

CENTRO PAULA SOUZA

FACULDADE DE TECNOLOGIA DE AMERICANA
Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas

FERNANDO CÉSAR HALFELD RENESTO

**Desenvolvimento de um sistema eletrônico financeiro e de tomada de decisão
para um site de compras coletivas**

Americana, SP

2015

CENTRO PAULA SOUZA

FACULDADE DE TECNOLOGIA DE AMERICANA
Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas

FERNANDO CÉSAR HALFELD RENESTO

fernandocesar.ti@hotmail.com

**Desenvolvimento de um sistema eletrônico financeiro e de tomada de decisão
para um site de compras coletivas**

**Trabalho Monográfico, desenvolvido em
cumprimento à exigência curricular do Curso
Superior de Tecnologia em Análise e
Desenvolvimento de Sistemas da Fatec-
Americana, sob orientação do Prof. Wladimir
da Costa.**

Americana, SP

2015

FICHA CATALOGRÁFICA – Biblioteca Fatec Americana - CEETEPS
Dados Internacionais de Catalogação-na-fonte

R328d	<p>Renesto, Fernando César Halfeld</p> <p>Desenvolvimento de um sistema eletrônico financeiro e de tomada de decisão para um site de compras coletivas. / Fernando César Halfeld Renesto – Americana: 2015. 92f.</p> <p>Monografia (Graduação em Tecnologia em Análise e Desenvolvimento de Sistemas). - - Faculdade de Tecnologia de Americana – Centro Estadual de Educação Tecnológica Paula Souza. Orientador: Prof. Me. Wladimir da Costa</p> <p>1. Desenvolvimento de software 2. Comércio eletrônico I. Costa, Wladimir da II. Centro Estadual de Educação Tecnológica Paula Souza – Faculdade de Tecnologia de Americana.</p> <p>CDU: 681.3.05 658.845</p>
-------	--

Fernando César Halfeld Renesto

**DESENVOLVIMENTO DE UM SISTEMA ELETRÔNICO
FINANCEIRO E DE TOMADA DE DECISÃO PARA UM SITE DE
COMPRAS COLETIVAS**

Trabalho de graduação apresentado como exigência parcial para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas pelo CEETEPS/Faculdade de Tecnologia – Fatec/ Americana.


Área de concentração: Análise de Sistemas.

Americana, 23 de junho de 2015.

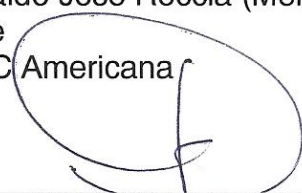
Banca Examinadora:



Wladimir da Costa (Presidente)
Especialista
FATEC Americana



Clerivaldo José Roccia (Membro)
Mestre
FATEC Americana



Antonio Alfredo Lacerda (Membro)
Especialista
FATEC Americana

“É preciso viver, não apenas existir”

Plutarco

AGRADECIMENTOS

Agradeço à minha família, em especial para meus pais, Paulo Sérgio Renesto e Sheila Escobar Halfeld Renesto, e minha irmã, Carolina Halfeld Renesto, por todo o apoio durante todos esses anos.

Agradeço à todos os amigos que conviveram comigo e sempre estiveram presente nas tristezas e felicidades, em especial para o Gustavo Suzigan, pela orientação do tema deste trabalho, ao amigo de longa data Maxihilian Polverini, por me motivar, e à todos os colegas de faculdade.

Agradeço ao Professor Wladimir da Costa pela paciência e disponibilidade de me orientar nesses últimos meses.

DEDICATÓRIA

Dedico esse trabalho aos meus pais pelo carinho e por investirem na minha educação, a minha irmã e aos meus amigos que fiz em toda a minha vida, em especial para o colega de faculdade Marcelo Kuhl *In Memoriam*.

RESUMO

O *e-commerce* cresce a cada dia que passa e, devido à esse sucesso, vem ganhando variações, como a compra coletiva. Desde a sua criação no formato em que conhecemos em meados de 2010, a compra coletiva é a forma mais rápida de adquirir ofertas de produtos e serviços na internet. No início esse tipo de *site* era exclusivo dos Estados Unidos, mas hoje até pequenas cidades de todo o mundo estão aderindo à esse negócio e, um exemplo claro dessa expansão é a presença de um *site* desse tipo em Americana no estado São Paulo. Como todo começo de negócio, a empresa está em constante crescimento e, para que essa euforia não atrapalhe esse processo, é preciso ter controle do caminho que a empresa está seguindo através de dados seguro e precisos sobre as ofertas. O projeto proposto nesse trabalho tem a intenção de demonstrar os passos no desenvolvimento de um sistema de apoio à tomada de decisões incorporada à um site de compra coletiva. Desenvolver o sistema eletrônico financeiro e de levantamento de dados visa desafogar setores de TI e de finanças permitindo otimizar as funções de todos os setores e proporcionando um diferencial estratégico/competitivo para a empresa.

Palavras Chave: compra coletiva; sistema financeiro; programação.

ABSTRACT

The e-commerce grows up each day and, because of this success, it is developing variations, as daily deals sites. Since your creation in the way we know nearby 2010, the daily deals is the fast way of buying offers of products and services in the internet. In the beginning this kind of sites, they were exclusive of United States, but today even small cities from the entire world are adhering to this business and, a clear example of this expansion it is in Americana, São Paulo, that have a site of daily deals. As all the beginnings of a business, the company is always growing up and to ensure that this euphoria won't disturb this process, it's needed to control the way the site is heading on through safe and precise data about the offers. The project proposed in this labor has the purpose of showing all the steps of developing a decision support system that will be attached to a daily deal site. Developing the financial electronic and gathering of data system should vent the IT and financial sectors providing a strategic/competitive differential to the company.

Keywords: daily deals; financial system; programming.

SUMÁRIO

INTRODUÇÃO	13
1. LEVANTAMENTO BIBLIOGRÁFICO	17
1.1 SISTEMA	17
1.1.2 SISTEMA DE INFORMAÇÃO	18
1.1.3 BENEFÍCIOS DE UM SISTEMA DE INFORMAÇÃO	20
1.1.4 SISTEMA ESTAÇÃO-CLIENTE X SISTEMA WEB	21
1.1.5 SISTEMA ESTAÇÃO-CLIENTE	21
1.1.5.1 VANTAGENS	22
1.1.5.2 DESVANTAGENS	23
1.1.6 SISTEMA <i>WEB</i>	23
1.1.6.1 VANTAGENS	24
1.1.6.2 DESVANTAGENS	24
1.2 COMPUTAÇÃO EM NUVEM	25
1.2.1 INFRAESTRUTURA	25
1.2.2 SERVIÇOS	27
1.2.3 <i>WEB BROWSERS</i>	28
1.3 COMPRA COLETIVA	28
2 UML	35
2.1 DIAGRAMA DE CASO DE USO	39
2.2 DIAGRAMA DE CLASSE	43
2.3 DIAGRAMA DE SEQUÊNCIA	50
2.4 DIAGRAMA DE ATIVIDADES	56
3. FERRAMENTAS DE TRABALHO	66
3.1 HTML	66

3.2 CSS.....	67
3.3 JAVASCRIPT	67
3.4 PHP	67
3.5 MySQL	68
4 ESTUDO DE CASO	69
4.1 DIFICULDADES	69
4.2 PROPOSTA DE SOLUÇÃO.....	70
4.3 DESENVOLVIMENTO	71
4.4 SISTEMA DE LEVANTAMENTO DE DADOS DE OFERTAS.....	73
4.5 SISTEMA DE LISTA DE PAGAMENTOS	74
CONCLUSÃO.....	79
REFERÊNCIAS.....	81
APÊNDICE	83

ÍNDICE DE FIGURAS

Figura 1 - Funções de um sistema de informação. (LAUDON & LAUDON, 2011)	20
Figura 2 - Exemplo de um diagrama de infraestrutura. (VELTE; VELTE; ESLSENPETER, 2010).....	26
Figura 3 - Etapas do ciclo de compra coletiva. (FELIPINI, 2011).....	31
Figura 4 - Diagrama de classe utilizando a propriedade de visibilidade. (GUEDES, 2011).....	38
Figura 5 - A Herança em sua forma mais complexa, a herança múltipla. (GUEDES, 2011).....	38
Figura 6 - Exemplo de polimorfismo. (GUEDES, 2011)	39
Figura 7 - Associação entre atores e um caso de uso. (GUEDES, 2011)	40
Figura 8 - Especialização/Generalização. (GUEDES, 2011)	41
Figura 9 - Inclusão em um sistema bancário (GUEDES, 2011).....	41
Figura 10 - Exemplo de extensão em um sistema bancário (GUEDES, 2011)....	42
Figura 11 - Exemplo de fronteira (GUEDES, 2011)	43
Figura 12 - Exemplo de estereótipo (GUEDES, 2011)	43
Figura 13 - Exemplo de um diagrama de classe. (GUEDES, 2011)	44
Figura 14 - Exemplo de Associação Unária (GUEDES, 2011).....	44
Figura 15 - Exemplo de Associação Binária (GUEDES, 2011).....	45
Figura 16 - Exemplo de Associação Ternária (GUEDES, 2011).....	45
Figura 17 - Exemplo de Agregação (GUEDES, 2011)	46
Figura 18 - Exemplo de Composição (GUEDES, 2011)	46
Figura 19 - Exemplo de Especificação/ Generalização (GUEDES, 2011).....	47
Figura 20 - Exemplo de Classe Associativa (GUEDES, 2011)	47
Figura 21 - Exemplo de Dependência/Realização (GUEDES, 2011).....	48
Figura 22 - Exemplo de interfaces e suas variações (GUEDES, 2011)	49
Figura 23 – Exemplo de restrição (GUEDES, 2011).....	49
Figura 24 – Exemplo de restrição de classe especializada (GUEDES, 2011).....	50
Figura 25 – Exemplo de diagrama de sequência (GUEDES, 2011)	52
Figura 26 – Exemplo de Mensagem Perdida e Mensagem Encontrada (GUEDES, 2011).....	53

Figura 27 – Exemplo de portas (GUEDES, 2011)	53
Figura 28 – Representação de usos de interação e de Operadores de Interação (GUEDES, 2011).....	54
Figura 29 – Representação de Interações (GUEDES, 2011)	55
Figura 30 – Exemplo de invariante de estado (GUEDES, 2011)	56
Figura 31 – Representação de Atividade (GUEDES, 2011)	57
Figura 32 – Exemplo Completo de um Diagrama de Atividades (GUEDES, 2011)	58
Figura 33 – Exemplo de Nó de Bifurcação com Repetição (GUEDES, 2011)	59
Figura 34 – Exemplo de Alfinetes (GUEDES, 2011).....	59
Figura 35 – Exemplo de Exceção (GUEDES, 2011)	59
Figura 36 – Exemplo de Ação de Evento de Aceitação (GUEDES, 2011)	60
Figura 37 – Exemplo de Ação de Evento de Tempo de Aceitação (GUEDES, 2011).....	60
Figura 38 – Exemplo de nós de buffer do tipo nó de repositório de dados (GUEDES, 2011).....	61
Figura 39 – Exemplo de nós de buffer do tipo nó de repositório de dados (GUEDES, 2011).....	61
Figura 40 – Exemplo de Ações de Chamada de Comportamento. (GUEDES, 2011).....	62
Figura 41 – Exemplo de Ações de Chamada de Operação. (GUEDES, 2011)	62
Figura 42 – Exemplo de Partição de atividade. (GUEDES, 2011)	63
Figura 43 – Exemplo de Região de Atividade Interrompível. (GUEDES, 2011) ..	63
Figura 44 – Exemplo de Nó de Atividade Estruturada. (GUEDES, 2011).....	64
Figura 45 – Exemplo de Região de Expansão. (GUEDES, 2011).....	64
Figura 46 – Exemplo de Código HTML. (NIXON, 2014).....	67
Figura 47 - Site de compra coletiva	71
Figura 48 - Tela de <i>login</i>	72
Figura 49– Demonstração da função utilizada para restringir acesso	72
Figura 50 - Painel de marketing.	73
Figura 51 - Página de dados de subofertas	73
Figura 52 - O método <i>Count</i>	74
Figura 53 - Opções para da lista	75
Figura 54 - <i>Tag select</i>	75

Figura 55 - Campos da opção Data Término da Oferta.....	75
Figura 56 - Código em JQuery que é acionado ao escolher uma das opções do <i>select</i>.....	76
Figura 57 - Definição do tipo de lista e preparação para os cálculos.....	76
Figura 58 - Resumo dos processos para gerar a lista de pagamentos	77
Figura 59 - Função para download da lista	77
Figura 60 - Lista gerada	78

LISTA DE ABREVIATURAS E SIGLAS

ABNT	Associação Brasileira de Normas Técnicas
CMS	<i>Content Management System</i>
CNPJ	Cadastro Nacional da Pessoa Jurídica
CSS	<i>Cascading Style Sheet</i>
DDoS	<i>Distributed Denial Service</i>
EUA	Estados Unidos da América
IBGE	Instituto Brasileiro de Geografia e Estatística
HTML	<i>HyperText Markup Language</i>
HTTP	<i>HyperText Transfer Protocol</i>
PHP	<i>HyperText Preprocessor</i>
POO	Programação Orientada à Objetos
RAD	<i>Rapid application Development</i>
SO	Sistema Operacional
TI	Tecnologia da Informação
UML	<i>Unified Modeling Language</i>
URL	<i>Uniform Resource Locator</i>
VPN	<i>Virtual Private Network</i>
XML	<i>eXtensible Markup Language</i>
SGBD	Sistema de Gerenciamento de Banco de Dados

INTRODUÇÃO

A criação da internet é um dos maiores pilares que contribuíram para o estilo de vida que as pessoas possuem atualmente. A internet sempre teve o compromisso de diminuir as barreiras da comunicação e, desde a sua criação, ela vem atingindo seu objetivo tornando-se a mais rápida e confiável ferramenta de troca de dados, no qual essa tecnologia permite que todo tipo de informação possa ir e vir em questões de milésimos de segundos em um espaço de milhares de quilômetros. Para quem nasceu em meados de 2000 no Brasil, não tem idéia de como era a vida sem a internet e serviços como o Mercado Livre, Facebook, Wikipedia, Google entre outros. Para realizar compras havia a necessidade das pessoas se dirigirem aos centros de compras ou shoppings, enquanto que as redes sociais na verdade eram clubes ou atividades sociais em geral, a Wikipedia era na verdade livros gigantes ordenados em ordem alfabética que eram atualizados anualmente e, para descobrir o telefone ou a programação do cinema, só eram possíveis através das, praticamente desnecessárias hoje em dia, listas amarelas. Logicamente todas essas formas alternativas citadas, que vieram antes da internet, ainda existem mas vem sendo adaptadas à essa era em que quase tudo pode ser feito por meios digitais.

O *e-commerce* é um dos setores dentro da internet que mais cresce no mundo pois, até pouco tempo, havia uma insegurança em fazer compras *online*, possibilitando que outros segmentos desta categoria, como as compras coletivas que já é sucesso no mundo todo desde meados de 2010. Esse novo tipo de negócio é bem visto por consumidores e comerciantes, até porque qualquer pessoa que tenha contato com a internet, já deve ter lido sobre ou até mesmo usufruído do serviço.

Algumas pessoas podem imaginar que as compras coletivas surgiram nos Estados Unidos (EUA) - lar de uma das maiores lojas de varejo do mundo, o Walmart - mas, a verdade, é que o seu berço foi na China. Seu formato atual surgiu após o estudo dos modelos chineses e, assim, adaptada ao mercado americano e à internet. A primeira grande empresa desse ramo foi o Groupon nos EUA, em meados de 2010, e assim o serviço se popularizou para o resto do mundo. (FELIPINI, 2011)

O projeto proposto nesse trabalho tem a intenção de demonstrar os passos

no desenvolvimento de um sistema de apoio à tomada de decisões incorporada à um site de compra coletiva. A empresa escolhida possui sede em Americana no estado de São Paulo e sua principal função é a de oferecer um serviço de venda de cupom de ofertas - ou de forma mais técnica, *marketing* agressivo - através de um sistema *web*. Essas ofertas são promoções de produtos ou serviços que só serão validadas se o número de interessados alcançar o mínimo de pessoas, definido pelo representante de vendas e o comerciante ofertante.

Já o **Problema** foi: O site de compra coletiva é um negócio em crescimento no Brasil e, para manter sua reputação entre parceiros e clientes, o site deve averiguar a procedência das ofertas e garantir que os pagamentos à parceiros seja realizado como combinado. Contudo, requisições à área de TI são regulares e em sua maioria eram sobre dados específicos de ofertas, como a quantidade de pessoas que compraram e os valores das comissões das ofertas. Esses dados são necessários para realizar rotinas administrativas, que atualmente são feitas manualmente e armazenadas em arquivos do Excel o que pode ser viável para pequenas empresas, mas com o crescimento da empresa isso se torna um empecílio.

Como **Pergunta** que se buscou responder: O sistema financeiro eletrônico integrado ao CMS do site é capaz de otimizar o tempo e eficiência dos setores da empresa?

As **Hipóteses** foram: Ao fim do projeto o sistema desenvolvido permitirá um desafogamento no setor financeiro e no de TI, permitindo a ampliação de outras atividades e, assim, poderá administrar melhor as finanças do site e liberar o máximo de tempo do programador para os projetos, diminuindo a dependência dos outros setores. Além do uso diário do setor financeiro existem outros dois setores que poderiam se beneficiar do sistema: o de marketing que, através do sucesso de vendas das ofertas conseguiria definir se os investimentos fariam jus à oferta ou não; e o de vendas que poderia verificar o custo/benefício da oferta e, assim, renovar contrato com o parceiro.

O **objetivo geral** consistiu em: Desenvolver um sistema eletrônico financeiro e de levantamento de dados visando desafogar os setores de TI e de finanças

permitindo otimizar as funções de todos os setores e proporcionando um diferencial estratégico/competitivo para a empresa.

Os **objetivos específicos** são: Desenvolver o sistema eletrônico com a mesma linguagem já usada no sistema do site, aprimorar a relação entre setores através de levantamentos de dados para todos os setores da empresa e gerar um relatório personalizado com os valores a serem pagos para cada oferta, diminuindo o trabalho e tempo do setor financeiro gasto com essa atividade.

Para auxiliar no desenvolvimento do trabalho foi utilizado como **metodologia** a pesquisa bibliográfica. "*A pesquisa bibliográfica é desenvolvida com base em material já elaborado, constituído principalmente de livros e artigos científicos.*", até porque, "*Os livros constituem as fontes bibliográficas por excelência.*" (GIL, 2002). À partir de dados históricos e documentados sobre como os sites de compra coletiva dos dois maiores consumidores do mundo, China e Estados Unidos, se adequaram à população, foi possível determinar algumas necessidades que os sistemas financeiro eletrônico e de e-commerce deveriam cumprir para satisfazer os usuários.

A partir de referências internas da empresa foi possível traçar as necessidades que o sistema financeiro eletrônico deveria suprir e que até mesmo aperfeiçoou algumas partes do sistema de compra coletiva à ser integrado, ou seja, foi aproveitada a pesquisa documental.

A pesquisa documental assemelha-se muito à pesquisa bibliográfica. A diferença essencial entre ambas está na natureza das fontes. Enquanto a pesquisa bibliográfica se utiliza fundamentalmente das contribuições dos diversos autores sobre determinado assunto, a pesquisa documental vale-se de materiais que não recebem ainda um tratamento analítico, ou que ainda podem ser reelaborados de acordo com os objetos da pesquisa. (GIL, 2002).

Como houve a necessidade do desenvolvimento de um sistema neste trabalho, a pesquisa experimental tornou-se necessária, já que novos cálculos e funcionalidades foram adicionados no processo.

[...] o experimento representa o melhor exemplo de pesquisa científica. Essencialmente, a pesquisa experimental consiste em determinar um objeto de estudo, selecionar as variáveis que seriam capazes de influenciá-lo, definir as formas de controle e de observação dos efeitos que a variável produz no objeto. (GIL, 2002).

O trabalho foi dividido em quatro capítulos mais a conclusão. O primeiro capítulo constitui o levantamento bibliográfico, e contém conhecimento teórico

necessário para entender o que são sistemas, computação em nuvem, compras coletivas. O segundo é uma exemplificação das funcionalidades presentes nos diagramas usados no projeto, o UML. O terceiro capítulo apresenta as ferramentas utilizadas para o desenvolvimento do sistema. O quarto capítulo é a parte prática, no qual é demonstrado os processos de desenvolvimento. Por último, a conclusão, que é sobre a qualidade do sistema desenvolvido e dificuldades.

1. LEVANTAMENTO BIBLIOGRÁFICO

1.1 SISTEMA

Sistemas sempre foram uma importante forma de, nós seres humanos, organizar, facilitar e manter o controle sobre atividades. Um sistema é a união de elementos que juntos podem finalizar uma determinada tarefa com o mínimo de esforço. Embora o estudo desse trabalho se baseie em um sistema informatizado para web - que une várias ferramentas para esse propósito como linguagens de programação, banco de dados entre outros - um sistema não precisa ser feito por computadores. Qualquer tipo de atividade que demande de passos e que use algum tipo de objeto para funcionar é um sistema. Uma varinha de bambu de pescar com anzol, bóia e isca pode ser considerado um sistema e através de técnicas de pesca e dos equipamentos citados é possível praticar a pesca.

Existem vários tipos de sistema, de manuais à computadorizados, mas a sua definição de um modo geral pode ser definida como: o conjunto de no mínimo dois ou mais elementos trabalhando em conjunto para alcançar um propósito. (SOMMERVILLE, 2007)

Alguns exemplos de sistemas utilizados no dia-a-dia:

- Sistema de entrega de encomendas (Correios);
- Sistema de Telecomunicações (Celulares, Internet e etc.);
- Sistema Elétrico;
- Sistema de Saneamento básico;

Como o trabalho se baseia em sistemas informatizados haverá, à partir desse ponto, foco nos sistemas desse tipo.

Os sistemas que utilizam algum tipo de software podem ser enquadrados em dois grupos: os sistemas técnicos baseados em computadores no qual são sistemas que possuem hardware e software mas não há uma utilidade específica para sua existência, como um editor de textos; e os sistemas sociotécnicos que podem ser constituído de um ou mais sistemas técnicos, que possuem processos pré-definidos

com o intuito de realizar uma atividade de maior importância, interagem com usuários e podem ser afetados por limitações legais ou organizacionais, como exemplo esse TCC que foi digitado em um editor de texto, mas que possui as regras da ABNT para padronizar as publicações do mesmo tipo de documento. (SOMMERVILLE, 2007)

Os sistemas sociotécnicos possuem finalidades empresariais e, portanto, quando são requisitadas visam encontrar uma forma melhor de organizar e agilizar processos e/ou aumentar os lucros. (SOMMERVILLE, 2007)

À partir desse ponto podemos dizer que o desenvolvimento desse sistema eletrônico financeiro é um sistema sociotécnico.

Como se sabe o negócio de compras coletivas é novo e, por isso, não possui um *modus operandis* concreto e finalizado, então qualquer diferencial é uma vantagem sobre os concorrentes. Embora essa ferramenta não seja usada por clientes ou parceiros, agilizar a produção dos empregados também é considerado um diferencial importante, permitindo que eles tenham tempo para realizar outras funções, dentro do que cabe a cada área. Um exemplo claro de como o sistema irá aumentar as vendas vem do sistema de levantamento de vendas de uma promoção que permite ao vendedor se vale ou não a pena investir naquele estabelecimento ou em um tipo específico de produto.

1.1.2 SISTEMA DE INFORMAÇÃO

Define-se sistema de informação a união de elementos que se entendem e trabalham juntos para coletar ou recuperar, processar, armazenar e distribuir dados de forma que eles se tornem informações úteis às organizações e que possibilitem ações decisivas, coordenativas e reguladoras. (LAUDON& LAUDON, 2011)

Os sistemas de informação que, qualquer pessoa tem acesso são os jornais, tanto os de papel, quanto os da televisão e os da internet. Uma notícia possui três etapas básicas, começando por um acontecimento que gera dados que sozinhos não fazem sentido algum, como os dados de um levantamento no Brasil de

aprovados em faculdades públicas por exemplo, e que, através de um jornalista, esses dados são processados de forma que divida os aprovados por etnias e, assim, gerar uma notícia do tipo "Branços são maioria nas faculdades públicas, segundo Instituto Brasileiro de Geografia e Estatística (IBGE)". Em alguns casos a notícia pode sofrer uma reformulação antes de ser publicada ou gerar uma errata após a publicação por causa de dados incompletos ou erros durante o processamento o que poderia alterar a notícia para: "Branços possuem 30% à mais de aprovação nas faculdades federais em relação à afro-decendentes, segundo IBGE", levando em consideração que a fonte dos dados se refere ao índice de aprovação no vestibular e não de alunos cursando. Lembrando que os dados citados foram criados apenas para exemplificar uma definição e, portanto, não representam a realidade.

O exemplo acima pode ser confirmado por (LAUDON & LAUDON) que cita os elementos, demonstrados na Figura 1, que qualquer sistema de informação deve ter:

- Entrada: são dados brutos que foram gerados ou coletados na empresa ou através de seus agentes externos (clientes, parceiros, distribuidores e etc.);
- Processamento: É a transformação dos dados da entrada em algo que possa ser entendido pelos usuários do sistema, ou seja, informação.
- Saída: É por ela que os responsáveis pelas tomadas de decisão irão receber as informações geradas.

Além desses processos podemos incluir o *feedback*. Processo importante para o aperfeiçoamento do sistema pois, através dele podem ser resolvidos problemas de entrada e de processamento, como exemplo alguma lei nova que altere a forma de executar uma conta ou processo de produção da organização.(LAUDON& LAUDON, 2011).

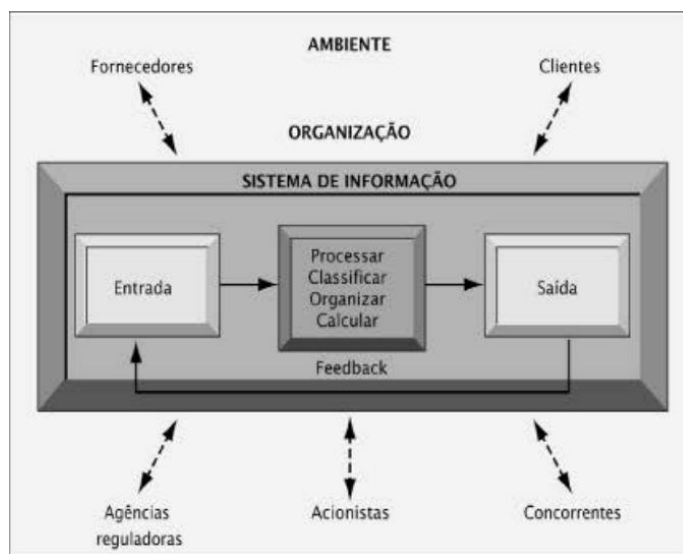


Figura 1 - Funções de um sistema de informação. (LAUDON & LAUDON, 2011)

1.1.3 BENEFÍCIOS DE UM SISTEMA DE INFORMAÇÃO

Os sistemas de informação foram criados para aperfeiçoar processos. Normalmente empresas que possuem algum tipo de sistema tem, por menor que seja, uma certa vantagem sobre seus concorrentes. As informações de mercado geradas por uma pesquisa ou até por um sistema, por exemplo, podem ajudar empresas a encontrarem falhas em seus outros sistemas e dessa forma permitem que superem grandes dificuldades ou até mesmo a torná-las ainda maior.

Alguns dos benefícios mais básicos de se ter um sistema de informação eficiente já foram citados: meio para melhorar decisões sobre o rumo da empresa e desenvolvimento ou aprimoramento de produtos. O caso da empresa Kia Motors é um exemplo claro desses benefícios. Ao tentar entrar no concorrido mercado norte-americano nos anos 90 a Kia conseguia vender seus produtos em uma quantidade razoável, mas que mantinha um número de defeito por veículo de 2,75, enquanto a média nacional dos Estados Unidos era de 1,3. Para aperfeiçoar seu produto, melhorar suas vendas e conseguir fidelizar seus clientes a, empresa contou com um sistema que recolhia dados de outros seis sistemas e organizava de forma a facilitar a visualização de queixas de clientes pelos vários meios de comunicação. Após a coleta de dados ser extraída, elas eram analisadas por um outro e emitiam alertas de picos de reclamação, agilizando possíveis recalls de seus carros ou apenas

peças. Graças ao novo sistema a empresa ficou em segundo lugar na categoria de carros compactos, com uma taxa de 1,4 problema por carro. (LAUDON & LAUDON, 2011)

Se a Kia precisou se ajustar apenas para enfrentar um mercado concorrido de uma população que gira em torno de 320 milhões (US Census, 2015), imagine uma empresa que tenha alcance mundial como o Wal Mart que possuem milhares de lojas, além da loja *online*, ao redor do mundo e precisa de suportar uma população que em 2011 atingiu a marca de 7 Bilhões (UNFPA, 2011):

[...]Em 2005, as vendas do Wal-Mart ultrapassaram os 285 bilhões de dólares - aproximadamente um décimo das vendas de varejo em todos os Estados Unidos - em grande parte graças a seu sistema RetailLink, que conecta digitalmente os fornecedores a cada uma das 5.289 lojas Wal-Mart espalhadas ao redor do mundo. Assim que um cliente compra um artigo o fornecedor fica sabendo que deve enviar um substituto para a prateleira. (LAUDON & LAUDON, 2011)

1.1.4 SISTEMA ESTAÇÃO-CLIENTE X SISTEMA WEB

Sistemas estação-cliente já foram amplamente usados há algum tempo atrás, mas com o advento da internet, outros tipos de sistemas se tornaram possíveis, aumentando a participação dos sistemas para *web* no mercado. Atualmente já existem tipos específicos de *softwares* para cada função desejada como cita (Sheriff, 2002), o tipo de sistema a ser desenvolvido depende do que o cliente dispõe, como se há um sistema já implementado e, caso a resposta seja afirmativa, qual banco de dados, qual linguagem usada entre outros. Além disso, deve ser levada em consideração qual a linguagem que o desenvolvedor conhece e qual a necessidade e utilidade do sistema a ser desenvolvido para o cliente.

1.1.5 SISTEMA ESTAÇÃO-CLIENTE

Nesse trabalho os sistemas *desktops* estarão incluídos na definição de estação-cliente assim como os sistemas móveis já que possuem muitas similaridades, diferenciando apenas pela capacidade e arquitetura do *hardware*.

1.1.5.1 VANTAGENS

De uma forma geral as vantagens desse tipo de sistema se resumem a qualidade e velocidade de comunicação que o *software* terá com o *hardware*. Em *softwares* que requerem alto rendimento de performance do hardware ainda não possuem *plugins* que funcionem de forma eficaz através de um intermediário. Um exemplo são os jogos *online* que rodam através de um navegador que, se comparados aos jogos de plataforma, possuem uma diferença significativa na qualidade de renderização. Segundo (Sheriff, 2002), um sistema *desktop*, permite ao desenvolvedor:

- **Comunicação da aplicação com o *hardware* do computador:** Permite utilizar o potencial do *hardware* que, normalmente, é superior e mais fácil de trabalhar, como no desenvolvimento de jogos, que necessita da utilização de placa de vídeo do usuário permitindo gráficos com complexidade maior que em um voltado para a *web*;
- **Melhor comunicação da aplicação com o banco de dados:** Sistemas *desktop* possuem requisições mais leves, pois o que muda de uma tela para outra, normalmente, são os dados apresentados, além de não depender de um provedor de internet em casos de processamento local;
- **Facilidade de comunicação com outros *hardwares* (impressoras, câmeras e etc.):** tornando seu sistema mais maleável e sem a necessidade de *softwares* de terceiros;
- **Uma aplicação para todos os computadores:** normalmente sistemas *desktops* não necessitam ser adaptados para várias plataformas, o que não acontece com sistemas *web* que precisam se adaptar para vários navegadores;

1.1.5.2 DESVANTAGENS

Embora o sistema estação-cliente possua vantagens de desempenho, nem sempre é a melhor escolha para o desenvolvimento, pois devem ser levadas em consideração outros pontos. As vezes um sistema pode ser rápido mas de difícil manuseio ou necessite de acesso rápido em diferentes sistemas operacionais e em diferentes locais. (Sheriff, 2002) cita os problemas e/ou barreiras encontradas em sistemas desse tipo:

- **Gasto com treinamentos:** pode causar confusão entre os usuários do sistema que não estarão acostumados ao novo sistema;
- **Dificuldades com aplicações que são remotamente operados:** Alguns problemas podem surgir em sistemas que utilizam essa funcionalidade como *firewalls*, conexão com a internet, softwares VPN entre outros;
- **Portabilidade:** caso o usuário necessite acessar o sistema de algum lugar diferente do comum ele terá que certificar de tê-lo instalado no novo computador;
- **Carga de acesso ao sistema:** Em caso de milhares de usuários a velocidade de resposta do servidor com os computadores requisitantes será maior e mais complexa que um acesso via HTTP;

1.1.6 SISTEMA WEB

Nos dias atuais em que a internet consegue alcançar os locais mais isolados, seja por um cabo ou via satélite, qualquer pessoa que tenha conhecimento da internet consegue definir o que é um sistema para *web*. Como o próprio nome diz é um sistema conectado por uma rede , ou em inglês *web*, permitindo que um servidor consiga fazer várias conexões com diferentes clientes. Define-se um sistema *web* da seguinte forma:

Uma aplicação *web* é qualquer aplicação que usa um *web browser* como cliente. A aplicação pode ser simples como um quadro de mensagem ou um registro de visitantes em um *website*, ou um complexo processador de texto. (Nations, tradução nossa)

1.1.6.1 VANTAGENS

Sistemas *web* possuem diversas vantagens, mas a principal continua sendo a mobilidade que ela permite, pois havendo um computador ou celular com acesso à internet é possível interagir com o *software* sem precisar de instalação ou sem exigir alguma configuração de *hardware*. As vantagens que (Shersiff, 2002) cita, são:

- **Conhecimento do ambiente:** usuários já possuem conhecimento do ambiente *web* e de como funciona;
- **Atualização centralizada:** sistemas *web* contam com a vantagem de, ao subir atualizações, só necessite realizá-la no servidor que se encontra hospedado;
- **Firewall e outros bloqueios:** O sistema *web* usa o protocolo HTTP e, portanto, diminui o risco dos dados serem barrados pelo *firewall*, o que não acontece com protocolos proprietários;
- **Processamento centralizado:** Algumas operações necessitam de um alto poder de processamento e , ao ter um servidor capaz de realizar o trabalho pesado, a empresa pode ter uma relação custo-benefício boa se tratando de *upgrades* em estações de trabalho;

1.1.6.2 DESVANTAGENS

Os sistemas *web* caminham para o tipo perfeito de sistema se encaixando em quase todos os tipos de funcionalidade. Mas ainda estão distantes dessa realidade já que os *plugins* as vezes não respondem como deveriam nos diferentes navegadores que existem dificultando a usabilidade do sistema. Além disso, por este tipo de sistema estar conectado à internet, facilita o acesso de pessoas estranhas à ele, aumentando as chances de ataques *hackers*, como o DDoS (*Distributed Denial of Service* ou Ataque Distribuído de Negação de Serviço) em que há um congestionamento de acessos, derrubando o sistema e prejudicando seu funcionamento. (Sheriff, 2002, tradução nossa) deixa isso claro ao citar "[...]desenvolver para *web* não é possuir uma tecnologia ilimitada.". Alguns dos problemas encontrados por (Sheriff, 2002), são:

- **HTML é irregular em diferentes navegadores:** As vezes os mesmo códigos HTML agem de forma diferente nos vários navegadores no mercado, podendo haver quebra de design.
- **Performance reduzida:** As requisições sempre serão, no mínimo, em grupos de duas partes, os dados e o design da página.
- **Uso de máscaras:** Em um sistema *web* o uso de máscaras – funcionalidade usada para pré-definir a entrada de dados, como o CNPJ que deve ter os pontos e hífen em locais específicos – terão que ser feitas com a ajuda de *scripts*, que são códigos que rodam no *browser* do usuário. Este problema acaba criando uma sobrecarga de código, já que a entrada do dado terá que ser testada duas vezes, uma pelo *script* local e outra pela linguagem *back-end*.

1.2 COMPUTAÇÃO EM NUVEM

O projeto, como já foi dito, será um sistema em que rodará dentro de um *site*, o que já pressupõe que será um sistema para a nuvem, ou como é mais conhecida, a internet. O *site* está hospedado em um *host*, ou seja, os arquivos que formam o site estão guardados dentro de um servidor na internet e toda vez que alguém acessar o site, através de uma URL ou endereço da *web*, esses arquivos serão executados pelo servidor e enviados até essa pessoa.

1.2.1 INFRAESTRUTURA

Em diagramas que exemplificam infraestruturas, como o da Figura 2, a internet é representada por uma nuvem, que significa tudo aquilo que não faz parte da estrutura estudada, mas possui algum tipo de relação com a estrutura diagramada. (VELTE; VELTE; ELSSEN PETER, 2010)

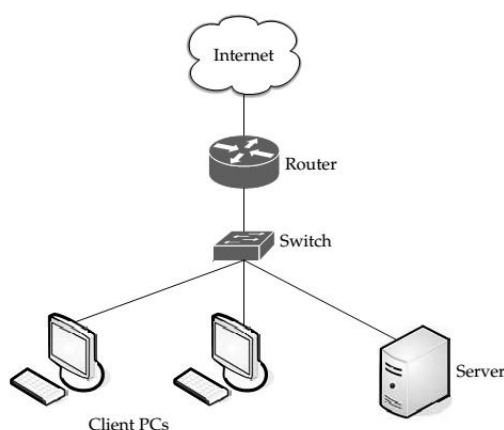


Figura 2 - Exemplo de um diagrama de infraestrutura. (VELTE; VELTE; ELSSENPETER, 2010)

Para desenvolver um sistema local para mais de uma pessoa, há gastos com manutenção, acessórios, pessoal especializado e atualizações que um servidor pode precisar para manter seu funcionamento em perfeito estado. Para (VELTE; VELTE; ELSSENPETER, 2010) essa é uma das maiores vantagens de um sistema na nuvem: não há gastos com infraestrutura, ao contratar uma empresa de hospedagem.

Segundo (VELTE; VELTE; ELSSENPETER, 2010) a computação em nuvem possui três elementos:

- Os clientes: são as ferramentas – computadores, celulares, *laptops* entre outros - usadas por qualquer usuário da internet para acessar o sistema hospedado na nuvem.
- Os *datacenter*, ou centro de dados: são os servidores que hospedam o sistema. Sistemas ou aplicativos na nuvem costumam não exigir muito do servidor físico e, por isso, muitas empresas vem utilizando servidores lógicos através da virtualização, ou seja, um servidor físico pode alocar vários servidores lógicos.
- Servidores distribuídos: são usados para garantir segurança e flexibilidade. Os serviços de hospedagem apostam em servidores em diferentes locais e, caso a área que um servidor esteja rodando uma instância de um sistema sofra um imprevisto, outro servidor em um local diferente pode manter o sistema no ar.

A infraestrutura da computação em nuvem possui variações que serão baseadas na necessidade que cada tipo de aplicação ou sistema irá precisar. (VELTE; VELTE; ESLSENPIETER, 2010) cita dois tipos de estruturas usadas em computação em nuvem:

- *Full virtualization*: ou virtualização completa, é a utilização de máquinas virtuais em servidores com o intuito de permitir ao cliente o uso do *hardware* presente nas máquinas servidoras.
- *Paravirtualization*: ou virtualização paralela, é uma alternativa ao modo *full*. Esta forma de virtualização é um modo de criar a máquina virtual dependendo da necessidade do cliente, podendo aproveitar as funcionalidades de cada SO.

1.2.2 SERVIÇOS

Segundo (VELTE; VELTE; ESLSENPIETER, 2010) há três de tipos de serviços que podem ser encontrados no mercado e, cada um deles, tem um propósito específico dentro de suas características. Eles são o SaaS (*Software as a Service* ou Programa com um Serviço), o PaaS (*Platform as a Service* ou Plataforma como um Serviço) e HaaS (*Hardware as a Service* ou Equipamento como serviço) que, dependendo do autor, também pode ser chamado de IaaS (*Infrastructure as a Service* ou Infraestrutura como um Serviço).

A compra coletiva é uma forma de SaaS, pois o cliente utiliza um serviço através da internet. Enquanto isso, a empresa de hospedagem que foi contratado pelo site é um HaaS, já que ela aluga todo um servidor, ou parte dele (como foi visto no capítulo anterior, sobre virtualização), para que o site possa estar *online*.

O site de compra coletiva utiliza o serviço da Amazon EC2 (*Amazon Elastic Compute Cloud* ou Computação Elástica em Nuvem) que, como comprova (VELTE; VELTE; ESLSENPIETER, 2010), é um serviço em que possui uma interface amigável ao cliente e tem, como principal característica, a flexibilidade. Caso o site receba muitos acessos ou precise usar mais capacidade de processamento do que o normal, o cliente pode alterar facilmente essas configurações.

1.2.3 WEB BROWSERS

Principal porta de acesso à computação em nuvem para usuários comuns, os *web browsers* são programas que foram desenvolvidos para traduzirem as mensagens dos servidores. Por um lado isso é uma vantagem, pois permite que servidores Linux se comuniquem com computadores clientes Windows ou MAC OS utilizando apenas uma linguagem de programação, por outro lado existem problemas de compatibilidade com os diferentes *web browsers* disponíveis. Páginas que apresentam quebra de *layout* ou *scripts* que não funcionem apropriadamente em um *browser* fazem com que os cliente procurem aquele serviço em outra empresa, por esse motivo é interessante verificar a compatibilidade com os diferentes *web browsers*, ou pelo menos os mais populares: Internet Explorer, Firefox, Safari e Chrome.

Essa ferramenta só é tão popular hoje em dia graças ao protocolo HTTP (Protocolo de Transferência de Hipertexto). Este protocolo é desprovido de estado, ou seja, não há necessidade de armazenar informações de usuários no servidor e, caso seja essa a vontade do desenvolvedor, eles podem utilizar *cookies*, que são variáveis criadas pelo programa rodando no *browser* do computador cliente. (VELTE; VELTE; ESLSENPETER, 2010)

1.3 COMPRA COLETIVA

Segundo (FELIPINI, 2011) define-se compra coletiva como um sistema *web* que possibilita que um grupo de pessoas possa adquirir produtos com desconto e, em alguns casos, há um mínimo estipulado de vendas daquela oferta e, se não alcançar aquele número, a oferta pode ser cancelada.

A compra coletiva é uma prática comum desde antes da internet, apenas conhecíamos por outro nome: compras por atacado. Na maioria dos casos, quem usava dessa estratégia eram os intermediários do comércio (lojistas em geral e distribuidores), quando há mais procura por um produto, mais o vendedor deveria ter em seu estoque e, portanto, mais a fábrica tinha de produzir para aquela loja e,

dessa forma, o próprio lojista poderia barganhar e preços mais baixos poderiam ser oferecidos aos seus clientes, naquele produto, já que o lucro por unidade será menor mas, no geral, o fabricante ainda sairá ganhando mais. (FELIPINI, 2011)

Isso acontece até hoje, mesmo que não seja um *site* de compra coletiva. Isso pode ser notado se compararmos lojas grandes com lojas regionais. Um exemplo de sistema que vem sendo introduzido na vida dos brasileiros são os supermercados por atacado (conhecidos como "atacadões") que, podem até ter uma maior variedade de produtos e possuir preços bem mais competitivos que um supermercado convencional. Obviamente há desvantagens, já que os "atacadões" normalmente ficam longe dos consumidores, pois necessitam de um grande espaço para estoque, enquanto que um supermercado convencional pode estar a menos de três quarteirões de casa.

Segundo (FELIPINI, 2011) para melhor entendermos como um *site* de compra coletiva funciona precisamos entender alguns termos que precisam ser de conhecimento de qualquer usuário do sistema, como:

- **Compradores:** Qualquer pessoa com poder aquisitivo. Todos podem comprar em um site de compra coletiva e se beneficiar de descontos de até 80% do valor de um produto. Existem algumas exceções, alguns *sites* são como clubes que tem preços abaixo do mercado, mas que possuem limitações de compras. Como não é o caso desse trabalho não iremos nos aprofundar.
- **Comerciantes/Parceiros:** Qualquer pessoa ou comerciante que queira oferecer um produto em busca de fidelizar clientes com a oferta de algum produto ou serviço, ou seja, oferecer algo por um valor abaixo do mercado esperando que o comprador goste e volte a fazê-lo com o valor normal.
- **Site de compra coletiva:** Empresa que utiliza o sistema *web* de compra coletiva que pode lucrar até 50% com o serviço.
- **Cupom:** Comprovante de compra com o código de compra e descrição do produto que fica ao dispor do comprador na área do usuário e permite baixar e/ou imprimir.
- **Ofertas:** Produto ou serviço que está disponível para compra no sistema.
- **Descrição da oferta:** Condições e descrições do produto em oferta no sistema.

- **Produto:** Qualquer tipo de produto, podendo ser um bem material ou um serviço ofertado no site.
- **Prazo de encerramento:** Tempo que o comprador tem para comprar. Não possui tempo mínimo ou máximo, mas normalmente cada oferta tem em torno de 12h à 24h para alcançar, se for o caso, o mínimo de vendas necessário para tornar válida a oferta.
- **Prazo de validade do cupom:** Tempo que o comprador tem para utilizar o cupom. Caso não o utilize no tempo, ele pode perder o direito de usar o cupom, dependendo da empresa.
- **Contrato de compra:** Documento que define os termos legais e o processo para utilizar a oferta. Normalmente conhecido como Termos e Condições de Uso.
- **Contrato de prestação de serviço:** Contrato que firma o acordo de oferta do produto entre a empresa ofertante e o site de compra coletiva.

O processo de compra começa com o responsável pelo setor de vendas que firma o contrato com o comerciante, que é enviado ao setor administrativo, que monta a oferta no sistema de criação definindo o horário em que a mesma irá entrar. Quando a oferta entrar no ar o cliente terá a possibilidade de se cadastrar, caso não tenha conta, e comprar a oferta. No dia do pagamento, como combinado no contrato, o setor administrativo se encarrega de fazer o pagamento. Assim que autorizado pelo site o cliente receberá uma mensagem, normalmente por e-mail, com o cupom para utilização do mesmo e, assim, se dirigir para o estabelecimento e adquirir o produto. O processo para compra coletiva pode ser ilustrado na Figura 3. (FELIPINI, 2011)

Etapas do ciclo de compra coletiva

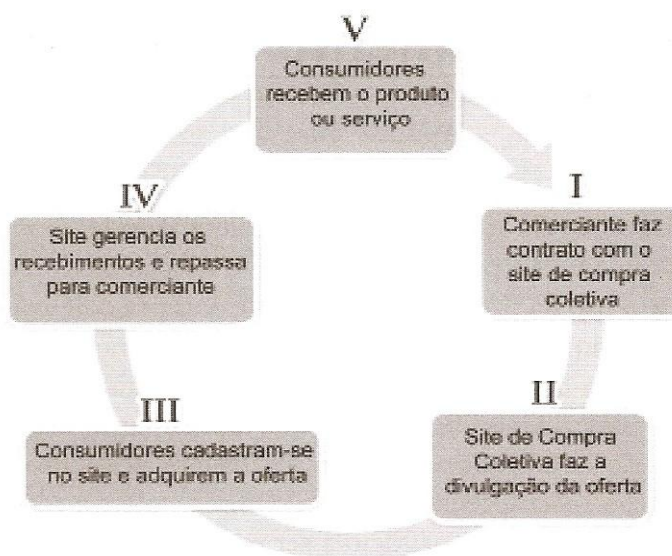


Figura 3 - Etapas do ciclo de compra coletiva. (FELIPINI, 2011)

As etapas descritas podem sofrer alterações, considerando que o negócio de compras-coletivas ainda é novo e pode ser adaptado a cada cultura em que é aplicada. Dentro da empresa em que este trabalho é desenvolvido há dois tipos disponíveis de ofertas, as que as compras são garantidas e aquelas que precisam de um número mínimo de vendas para liberar os cupons. Alguns *sites* ainda utilizam o sistema *web* para vender cupons que garantem ingressos ou combos de ingresso em shows e festas, aumentando a abrangência do site.

Existem três elementos que possuem interesses nos benefícios com as ofertas de um site de compra coletiva: o comprador, o comerciante e o empreendedor. (FELIPINI, 2011)

Para o cliente as vantagens podem ser diversas, mas ele deve estar ciente que os setores que possuem maior sucesso são os de serviço e os de alimentos. Na maioria dos casos, os compradores não precisam do que adquiriram, essa atitude é chamada de compra por impulso. Em resumo, o cliente irá adquirir um produto que, em situações normais, ele não adquiriria, como um restaurante caro, um produto com um desconto bom ou visitar um lugar que nem conhecia, mas que interessou ao visualizar a oferta. O cliente estará sempre informado da situação do produto que está adquirindo pois, na maioria dos sites deste tipo, há informações e fotos do local que, as vezes, podem transformar um oferta atrativa em algo sem utilidade, como

um restaurante que fica longe do trabalho do cliente e que não inclui entrega, tornando a oferta inviável. No caso do produto ser diferente do que aparentado no *site*, o cliente pode pedir a troca do produto ou ter o dinheiro de volta. Além desses fatores, alguns sites dão créditos à pessoas que indicam amigos para cadastro no site, que podem chegar até dez reais por pessoa indicada. (FELIPINI, 2011)

Os comerciantes tem o dever de preparar uma oferta que chame a atenção, tanto pelo valor quanto pelo tipo de oferta, sem se prejudicar a ponto de ter prejuízo e o de cumprir com o combinado sem menosprezar o cliente que está, de certa forma, pagando menos que os clientes que já se beneficiavam daquele produto.

Comerciantes que desejam anunciar em um site de compra coletiva pela primeira vez normalmente pensam que estarão vendendo o produto como em um mercado livre, o que não é verdade. Em média, o comerciante só recebe 25% da receita total das vendas da oferta. O real objetivo de se anunciar em um site de compra coletiva é o de angariar clientes novos, se o estabelecimento oferecer um produto que o cliente goste, há grandes chances de fidelizá-lo. Em resumo, o site de compra coletiva nada mais é que um *marketing* agressivo, no qual o pagamento é cobrado de acordo com a venda, pois o cliente terá que consumir pra conferir o produto ofertado. Por esse motivo que muitas vezes as melhores ofertas são na área de serviços porque, é mais fácil retirar o custo da mão-de-obra do que de uma mercadoria. Em alguns casos a situação é amenizada como em restaurantes, que ofertam um prato e podem até melhorar a margem de lucro com outros produtos, como bebidas ou até mesmo outros pratos que possam interessar ao consumidor. Outra forma que ameniza o investimento são os clientes que realizam a compra do cupom, mas perdem o prazo de validade ou perdem os sete dias previstos em lei para reembolso de um produto comprado na internet ou simplesmente não os utilizam e, muitas vezes nem pedem reembolso. Nos Estados Unidos há dados que confirmam que até 40% dos clientes não utilizam seus cupons. Isso é bom para o comerciante pois ele receberá o valor total da venda, independente se o usuário usar ou não o cupom. (FELIPINI, 2011)

O comerciante deve ainda se precaver de algumas situações em que o sistema de compra coletiva pode afetar negativamente, ao invés de melhorar sua relação com os novos clientes. Ao realizar a oferta o comerciante deve estar ciente

que irá ficar com pouco do lucro gerado, portanto não adianta criar uma oferta muito atrativa e atrasar pagamentos ou ter falta de produtos. Outro cenário é a venda de um produto visto como diferente, isso pode enfraquecer a marca, dando a sensação que não vale a pena pagar o preço original, além de que esse tipo de produto pode atrair compradores oportunistas, que são aqueles que, mesmo que goste do produto, não tem a intenção de voltar. Os produtos que tem pouca saída são os mais atrativos para esse tipo de mercado, já que podem atrair um mercado diferente que antes não alcançava. Uma dica importante é não tentar enganar o comprador com falsas promessas mostrando um produto e entregando outro, isso causa um marketing negativo. Outra situação importante é tomar cuidado com cupons de fraudadores que, com um simples programa de edição de imagens, pode tentar usar um mesmo cupom duas vezes. O último ponto que o comerciante deve se preocupar é com a reputação do site em que pretende fechar o contrato. *Sites* confiáveis são facilmente reconhecidos pela internet e com uma pequena pesquisa pode descobrir se o mesmo possui uma boa reputação. (FELIPINI, 2011)

Um *site* de compra coletiva é o sistema intermediário que deve prover ofertas de qualidade para seus clientes e garantir um preço justo com o parceiro. Infelizmente há a possibilidade de fraudes, dos dois lados, portanto as principais funções de um sistema desse tipo é o de oferecer um *marketing* eficiente, pagamento da parte dentro do prazo combinado em contrato e de informações básicas dos clientes que adquiriram o cupom, como nome do titular do cupom e o número do cupom. Para os clientes a obrigação se resume em ter acesso à todas as informações importantes do estabelecimento tais como endereço, meios de comunicação com o ofertante, fotos do local e do produto, documento da empresa ou do responsável e preço original e o valor do ofertado. Dentro desse comércio existe ainda um quarto componente que são as pessoas, que podem ser ou não clientes, que tiveram seus cartões clonados, pois, infelizmente, pelo motivo de sites de compras coletivas ofertarem produtos por um tempo curto, eles devem possuir sistemas de pagamentos que utilizem cartão de crédito e, algumas vezes, o processo de verificação do banco não é rápido suficiente para cancelar a compra a tempo da data de utilização.

O empreendedor, dono de um site de compra coletiva deve, segundo (FELIPINI, 2011), estar sempre atento a dois pontos críticos, se deseja ter sucesso nesse segmento:

- **Compradores:** são que os procuram um sistema que entregue tudo que o site tem a oferecer sem precisar correr atrás, seja recebendo um e-mail com as ofertas do dia, seja possuindo um site de fácil acesso e intuitivo.
- **Comerciante ou parceiro:** o representante da empresa deve mostrar claramente como é o funcionamento da empresa deixando claro como funciona um site de compra coletiva, melhorando ou alterando a oferta para que o retorno seja positivo para todos os três envolvidos.

2 UML

Para o desenvolvimento do sistema utilizaremos a UML – *Unified Modeling Language* ou linguagem unificada de modelagem – na qual (GUEDES, 2011) define como “[...] linguagem visual utilizada para modelar *softwares* baseados no paradigma de orientação a objetos”. Portanto, será utilizada uma ferramenta que permitirá, de forma clara e visual, visualizar quais as características que o sistema poderá ter como: requisitos, comportamentos, estruturas lógicas, dinâmica de processos e prováveis necessidades físicas (*hardware*), ou seja, prevenir possíveis falhas que poderão ocorrer durante o processo de desenvolvimento.

Como cita (GUEDES, 2011) muitas pessoas não veem a necessidade de modelar *software*, principalmente sistemas pequenos ou sistemas que tem um ou poucos envolvidos em seu desenvolvimento. Antigamente, quando os sistemas ainda eram menos complexos e pouco utilizados, talvez não houvesse tanta necessidade de utilizarmos essa ferramenta, mas como em 2005 - quando foi oficialmente lançada a versão 2.0 da linguagem de modelagem UML - já havia necessidade de planejar softwares, pode-se imaginar nos dias de hoje em que os sistemas mais são dinâmicos, ou seja, possuem a tendência de crescerem, seja por requisições de clientes, mudança nas estratégias das empresas ou até mesmo por novas leis que podem interferir na forma como um sistema trabalha. Essas documentações irão ajudar futuros desenvolvedores ou até mesmo o próprio desenvolvedor e, assim, diminuir as chances de surgirem novos erros durante a correção dos antigos e ao aplicar novas funções.

Segundo (GUEDES, 2011) durante o processo de desenvolvimento temos algumas fases de amadurecimento do projeto, que são:

- **Levantamento e Análise de Requisitos:** é nessa fase que determinamos, através de entrevistas e/ou questionários com o cliente, a primeira pergunta: "o que?", o que o software deve resolver e se é possível desenvolvê-lo. Nessa fase há apenas um problema, a comunicação entre ambas as partes pois, nem sempre o que o cliente pede é o que ele realmente quer ou pode haver até mesmo a falta de compreensão do sistema por parte do desenvolvedor. As principais características dessa fase são os requisitos

funcionais e não-funcionais. Os requisitos funcionais são os problemas que o sistema deve solucionar para o cliente. Já os requisitos não-funcionais são as restrições e condições que os sistemas não-funcionais devem ter.

- **Prototipação:** algumas aplicações, chamadas de RAD (Desenvolvimento Rápido de Aplicações), facilitam esse processo ao permitir a construção e alteração rápida de interfaces sem interferir em códigos a serem escritos ou já escritos. Dessa forma evita-se a descoberta tardia de falhas de planejamento ou de algum mal-entendido entre o desenvolvedor e o cliente na primeira fase do planejamento. O único problema da prototipação é a de passar uma falsa impressão de que o sistema está em estágio avançado de desenvolvimento.
- **Prazos e custos:** nessa fase deve-se ter um cuidado maior, pois é nela que se define a quantidade de desenvolvedores no projeto, tempo de desenvolvimento entre outros, caso contrário pode haver descontentamento por parte do cliente e prejuízo por parte do responsável do planejamento.
- **Projeto:** nessa parte procura-se responder como ele será trabalhado para desenvolver o sistema de forma rápida e eficiente. É nela que a modelagem começa a ser montada e estudada, com qual linguagem de programação e qual banco de dados a ser implementado.
- **Manutenção:** entende-se por manutenção o ato de reparar o que está errado, principalmente porque essa é uma das fases de maior trabalho e de maior custo durante um projeto mas, além do fato de haver alguma falha de projeto, podem ser criadas necessidades por elementos externos ao projeto inicial (leis, cliente e mercado), como já foi citado. Nessa fase que podemos perceber a importância da documentação no processo de desenvolvimento pois, sabemos que equipes podem mudar e, se não tivermos especificações bem definidas o sistema pode ser mais um empecílio do que uma solução, tanto para o cliente quanto para o desenvolvedor.

A linguagem de modelagem UML foi criada para *softwares* que serão desenvolvidos com Programação Orientada a Objetos (POO) possuindo vários diagramas adaptados para esse caso. O responsável pelo projeto poderá escolher um ou mais dentre todos esses diagramas, permitindo diferentes pontos de vista do

sistema e, assim, minimizar problemas de desenvolvimento e facilitar futuras alterações. (GUEDES, 2011)

Antes de estudar quaisquer dos diagramas que a linguagem UML possui há a necessidade de entender o que é a POO. A propriedade primária desse tipo de programação é a classe. Esta é uma propriedade de fácil compreensão, pois desde pequenos estamos acostumados a classificar coisas com definições abstratas. (GUEDES, 2011)

A classe dos carros, por exemplo, possui uma grande variedade de marcas e modelos. Sabe-se que essa classe deve ter a variável rodas (com o valor de quatro) e portas, ou seja, ao ver um carro com o nome Palio percebe-se que há diferenças com outros objetos da mesma classe carros e, ao criar a definição de Palio - ou criar a instância de um novo objeto da classe Palio - que definirá todo carro com aquelas variáveis e operações com aquele objeto. Pode-se aproveitar da classe carro a operação ou método carregar, mas nem todo carro serve apenas para carregar pessoas, mas também, outros tipos de carga. Dessa forma Palio pode ser uma classe abstrata e possui duas subclasses Adventure e Street no qual a variável caçamba pode ser definida como habilitada em um, e desabilitada em outro. Resumindo, temos a classe abstrata automóveis que possui a classe abstrata carros e dentro desta outra classe abstrata Palio que por fim tem duas classes Adventure e Street.

Outra propriedade importante em uma POO são as visibilidades dos métodos que são ao todo quatro: a privada, a protegida, a pública e a pacote. A visibilidade privada representa que o método ou atributo será utilizado apenas pelos objetos da classe ao qual pertence, seu símbolo é o (-). A visibilidade protegida permite o mesmo que a privada com a diferença que os objetos de sua(s) subclasse(s) também acessarão os seus elementos, seu símbolo é o (#). A visibilidade pública permite que qualquer objeto use os atributos ou métodos, seu símbolo é o (+). Por último a visibilidade pacote que permite que qualquer objeto tenha acesso dentro daquele pacote, seu símbolo é o (~). Um exemplo de como essas visibilidades são aplicadas em um diagrama de classe, que será melhor detalhada nos itens subsequentes, se encontra na Figura 4. (GUEDES, 2011)

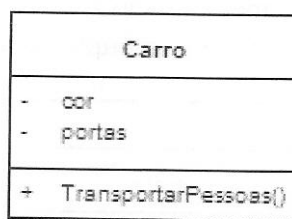


Figura 4 - Diagrama de classe utilizando a propriedade de visibilidade. (GUEDES, 2011)

A propriedade mais importante dentro da POO é a herança, que nada mais é a relação entre classes permitindo que uma subclasse use métodos ou atributos de uma classe-mãe (ou superclasse), evitando linhas de código desnecessárias, permitindo especializações, e diminuindo as dificuldades na manutenção. As especializações nada mais são que classes que partilham de algum(s) atributo(s) e/ou método(s) em comum. Algumas linguagens de POO permitem que uma classe herde elementos de mais de uma classe e este tipo de herança é chamado de herança múltipla. A propriedade pode ser melhor visualizada na Figura 5. (GUEDES, 2011)

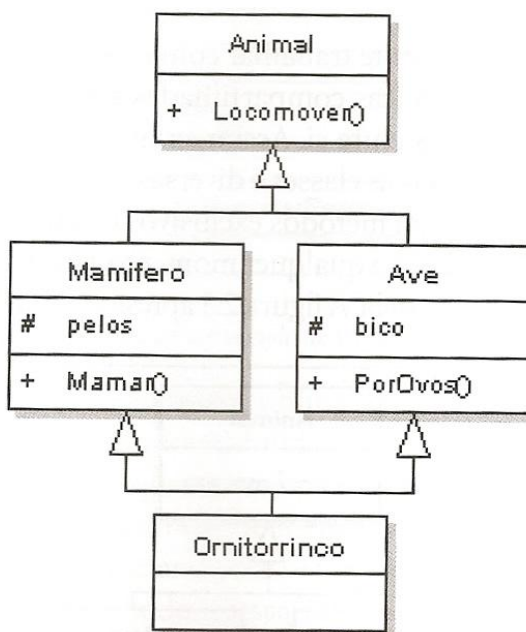


Figura 5 - A Herança em sua forma mais complexa, a herança múltipla. (GUEDES, 2011)

A herança ainda permite que duas subclasses, que compartilham de um método com nome em comum da superclasse, possam criar variações desse método dentro de sua própria classe sem que interfira um com o outro. Essa

propriedade é chamada de polimorfismo e, em algumas linguagens de POO, ela não está habilitada, por isso, antes de iniciar um projeto, é interessante verificar todas as ferramentas que a linguagem possui. Na Figura 6 há um exemplo dessa utilidade.

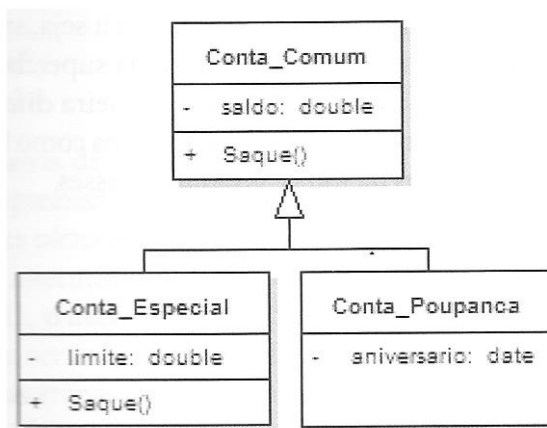


Figura 6 - Exemplo de polimorfismo. (GUEDES, 2011)

Nesse trabalho serão utilizados os diagramas de caso de uso, de classe, de sequência e de atividades que serão melhor apresentados a seguir.

2.1 DIAGRAMA DE CASO DE USO

Todo projeto deve começar das necessidades mais básicas e simples para que possa ser criada uma visão mais generalizada do sistema e, à partir desta visão criar esboços mais complexos baseados neste tipo mais simples.

Desta forma podemos dizer que o diagrama de caso de uso é bom para começar o planejamento de um sistema pois funciona como um visão geral do sistema, sem especificações técnicas, ou seja, é uma forma rápida e que permite visualizar e entender facilmente o propósito do projeto. E, por não descrever a parte técnica do sistema, o diagrama pode ser adicionado, junto de um protótipo, à pauta de reuniões com clientes. (GUEDES, 2011)

Esse diagrama possui dois itens essenciais para assim ser chamado de diagrama de caso de uso: Atores e Casos de Uso. Os atores são, como o próprio nome diz, os usuários que agem no sistema. Os casos de uso também possuem um nome bem intuitivo, são funções, pertencentes ao sistema, que atores poderão

utilizar, seja direta ou indiretamente. Os Casos de uso podem ser primários ou secundários. Os do tipo primário são os mais importantes dentro do sistema, como uma função de um sistema de compra coletiva relacionada ao pagamento, já os casos secundários seriam representadas como uma atualização de cadastro de usuários. Esses casos podem conter legendas, informando mais detalhadamente cada um. Para informar qual ação ou se há relacionamentos entre os elementos do caso de uso são usadas associações. Algumas situações permitem que o ator acione mais de uma vez um caso de uso e são chamados de multiplicidade. Para representar esses casos são utilizados o número 1 (um), para indicar que aquele usuário poderá acionar o caso de uso uma vez, e o símbolo asterisco, que simboliza que o ator não terá limites para usar o caso de uso. Na Figura 7 poderemos visualizar os dois clientes (atores) acionando (associação) uma função (caso de uso) para cadastro de sócio de um estabelecimento e, dessa forma, perceber que o ator "Funcionário" pode acionar mais vezes o caso de uso "Cadastrar Usuário" do que o ator "Sócio".(GUEDES, 2011)



Figura 7 - Associação entre atores e um caso de uso. (GUEDES, 2011)

Um caso de uso ou ator pode conter variações dentro de sua definição, podem ser chamados de especialização ou generalização. Ele é representado por uma seta interligando os elementos, como pode ser visualizado na Figura 8, no qual representa um ator nomeado de pessoa que tem duas especializações, pessoa física e pessoa jurídica. Sabemos que há poucas diferenças entre os dois atores, por isso pode ser utilizado um ator principal (pessoa) que representa as características que as duas especificações tem em comum. (GUEDES, 2011)

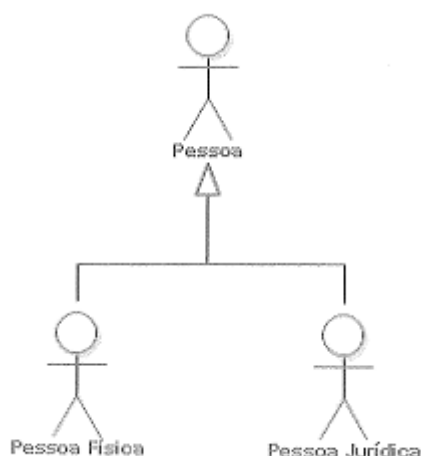


Figura 8 - Especialização/Generalização. (GUEDES, 2011)

Casos de uso ainda possuem outras duas conexões: inclusões e extensões. Ambas são representadas por setas que possuem linhas tracejadas e estão presentes quando há casos de uso que podem ser utilizadas em dois ou mais processos. As diferenças entre elas são a direção em que se direcionam e o nome que as representa. (GUEDES, 2011)

Quando um caso de uso possui uma inclusão aquele processo que será incluído será obrigatoriamente ativado. Na Figura 9 temos um exemplo de inclusão em um sistema bancário. (GUEDES, 2011)

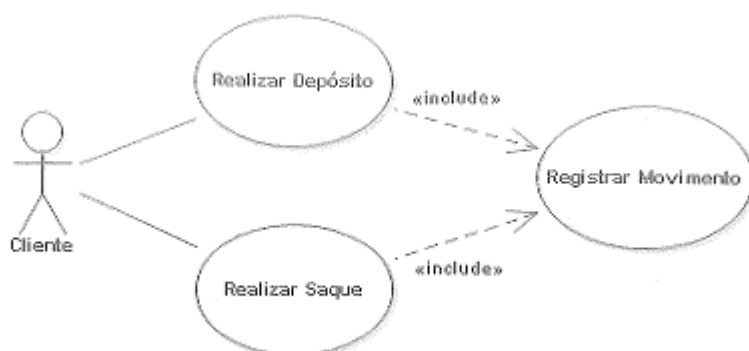


Figura 9 - Inclusão em um sistema bancário (GUEDES, 2011)

As extensões, ao contrário das inclusões, são casos de uso não obrigatórios e por isso, em alguns casos, necessitam de especificações que podem ser chamados de restrições e pontos de extensão. Pontos de extensão são situações prováveis que um caso de uso pode gerar e a partir desses pontos é que temos as restrições

para cada caso de uso estendido. Na Figura 10 podemos ver o que poderia ser a outra parte do caso de uso para o sistema bancário acompanhado de suas propriedades. (GUEDES, 2011)

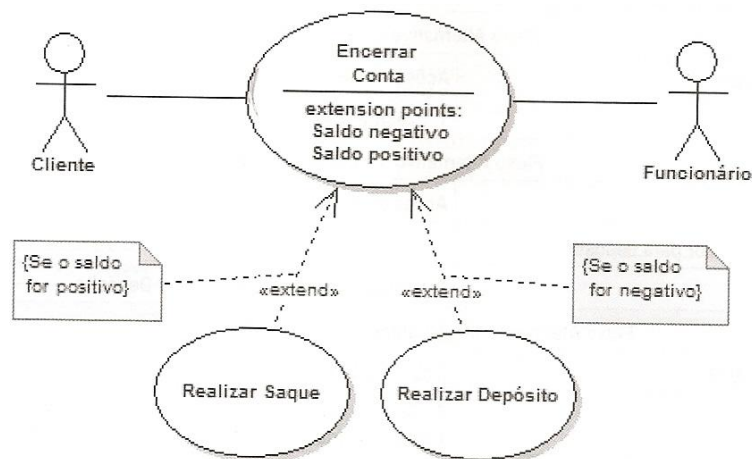


Figura 10 - Exemplo de extensão em um sistema bancário (GUEDES, 2011)

Outra propriedade de estrutura do caso de uso é a fronteira do sistema. Representadas por retângulos, as fronteiras como função dividir sistemas de subsistemas ou até mesmo de um sistema que interaja com outro, como um sistema de compra coletiva utilizando um sistema de pagamentos. Na Figura 11 há um exemplo da utilização da fronteira em um sistema de pesquisa na internet. (GUEDES, 2011)

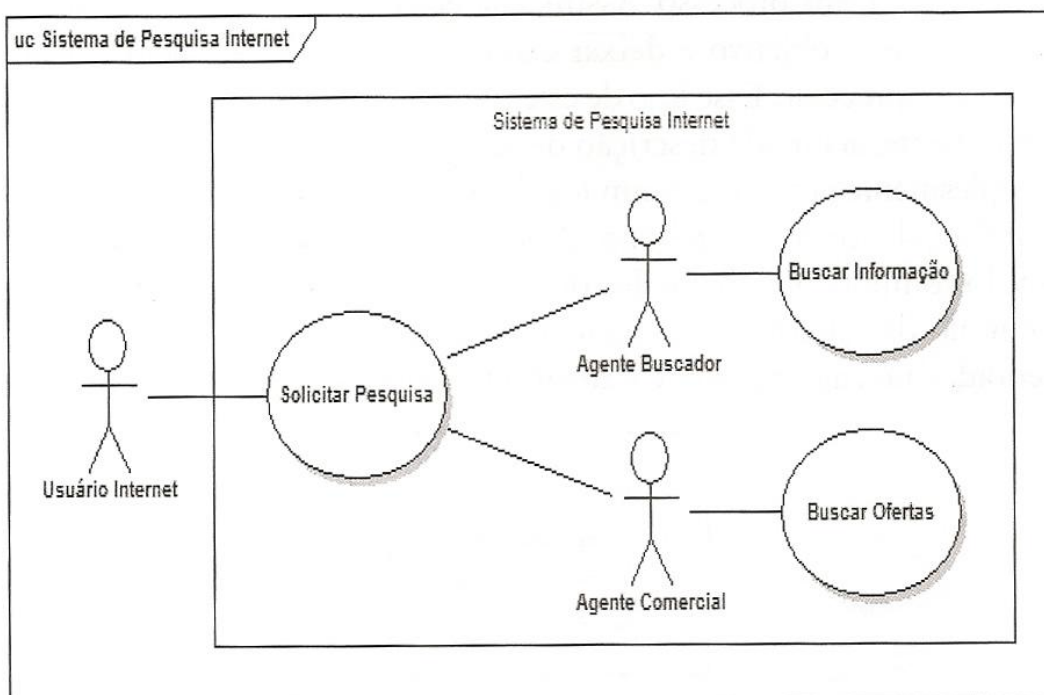


Figura 11 - Exemplo de fronteira (GUEDES, 2011)

Na Figura 12 temos o último elemento desse diagrama: o estereótipo. É utilizado em situações que o desenvolvedor pode especificar melhor a função de um caso de uso. Sua representação deve ficar a cima do nome do caso de uso e sua estrutura inicia o símbolo "<<" seguido pela nomenclatura e termina com o símbolo ">>". A nomenclatura fica à escolha do responsável. (GUEDES, 2011)



Figura 12 - Exemplo de estereótipo (GUEDES, 2011)

2.2 DIAGRAMA DE CLASSE

Após criar uma visão geral, o sistema deve começar a mostrar suas características mais claramente e, nesse passo da diagramação, inicia-se a introdução de termos técnicos. Neste diagrama existem algumas ferramentas criadas exclusivamente para uma ou outra linguagem de programação, portanto algumas funcionalidades do diagrama de classe podem não ter utilidade para este projeto.

O diagrama de classe tem como função definir as classes do projeto. É dele que se descobre o que o *software* precisa como variáveis e associações entre classes, ou seja, sua composição lógica. (GUEDES, 2011)

Na Figura 13 temos a representação desse tipo de diagrama que é formado por um retângulo que contém três partições: o nome da classe, as variáveis presentes naquela classe e as funções, ou métodos, que ela pode executar. Lembrando que, embora na Figura 13 vê-se as três partições, apenas a nomenclatura da classe é obrigatória. (GUEDES, 2011)

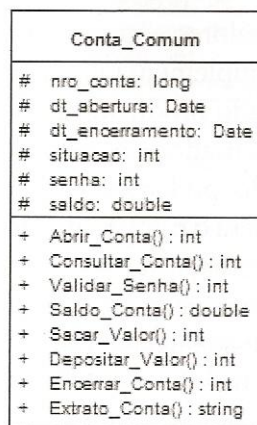


Figura 13 - Exemplo de um diagrama de classe. (GUEDES, 2011)

Assim como no diagrama de casos de uso, o diagrama de classes também possui associações que permitem visualizar as heranças em ação. Mas nesse caso temos três tipos de associações: unária ou reflexiva, binária e ternária ou N-ária. (GUEDES, 2011)

A do tipo unária, que pode ser visualizada na Figura 14, surge quando a própria classe interage com ela mesma, podendo conter limites de quantas vezes ela pode interagir com ela mesmo, pelo que chamamos de multiplicidade. Esta propriedade que pode ser reproduzida com n..n sendo a letra n um número natural e, caso seja infinita, deve ser utilizado o asterisco. Um exemplo 0..* , que informa que pode haver nenhuma ou muitas associações entre classes. Se não for informado no diagrama deve ser considerado a 1..1 que significa que apenas um objeto se relaciona com o objeto de outra classe. (GUEDES, 2011)

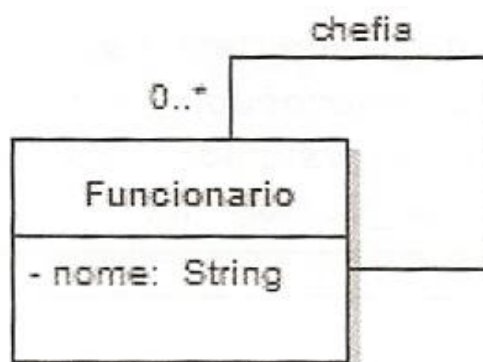


Figura 14 - Exemplo de Associação Unária (GUEDES, 2011)

No caso das binárias, que é representada na Figura 15, acontece quando duas classes diferentes se associam, mantendo a propriedade de multiplicidade e adicionando a de navegabilidade. Esta propriedade, que não é obrigatória, serve

para definir se alguma classe tem ou não poder de ativar métodos em outra. (GUEDES, 2011)

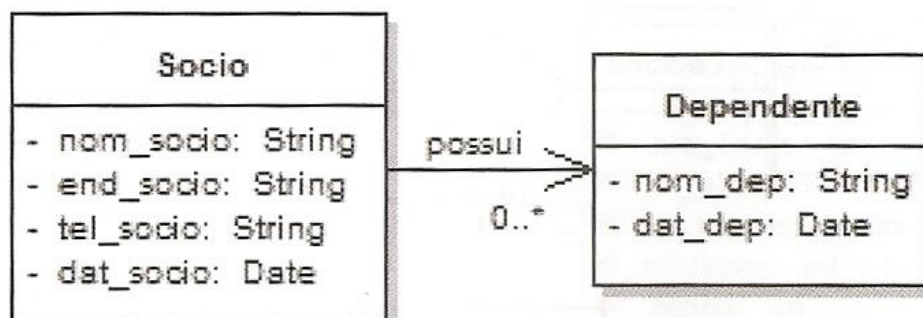


Figura 15 - Exemplo de Associação Binária (GUEDES, 2011)

À partir de três associações chamamos de ternária e utilizamos um losango para inteligarmos as associações como demonstra a Figura 16, na qual o professor pode lecionar em uma mesma turma. Estas associações existem para serem aplicadas mas, em alguns casos, tornam o entendimento difícil e por isso devem ser evitadas. (GUEDES, 2011)

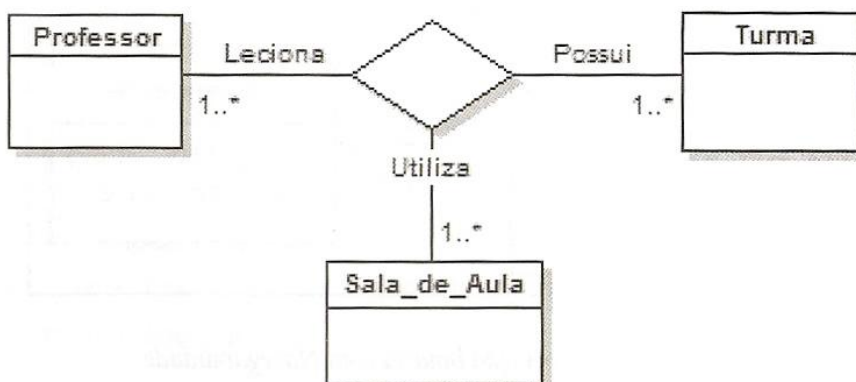


Figura 16 - Exemplo de Associação Ternária (GUEDES, 2011)

Há ainda a propriedade de agregação que é uma associação de complemento, ou seja, uma classe a irá complementar outra classe. Na Figura 17 vemos a representação de um diagrama usando agregação de uma conta conjunta em um banco qualquer. (GUEDES, 2011)

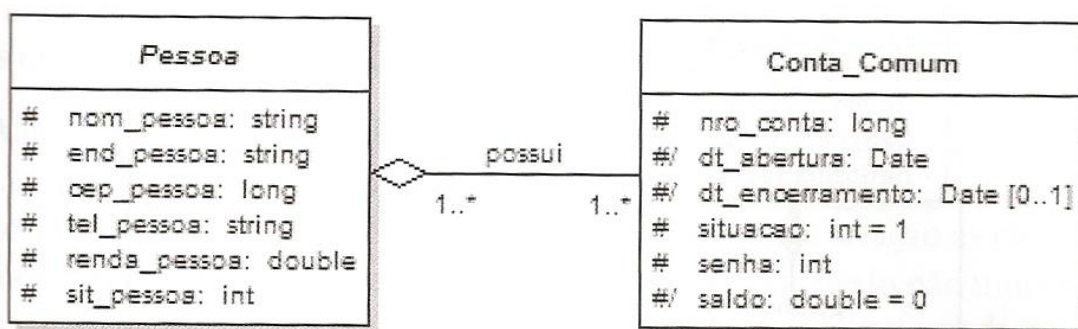


Figura 17 - Exemplo de Agregação (GUEDES, 2011)

A agregação possui uma variação chamada de composição. Esta variação é uma associação mais forte que força uma classe a se associar somente com a classe associada. Sua representação é de um losango preenchido. Na Figura 18 temos a representação de sua aplicação. (GUEDES, 2011)

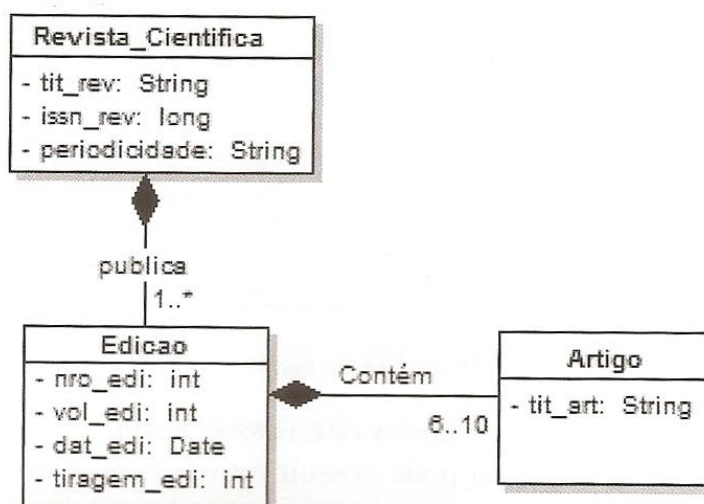


Figura 18 - Exemplo de Composição (GUEDES, 2011)

Dentro do diagrama de classes temos ainda a generalização e especialização que tem como objetivo de especificar o relacionamento entre as classes e defini-las como superclasse ou subclasse, dependendo de quem é herdeiro de quem. A Figura 19 deixa claro o funcionamento desse tipo de associação, a classe `conta_comum` é a superclasse e as outras as subclasses. (GUEDES, 2011)

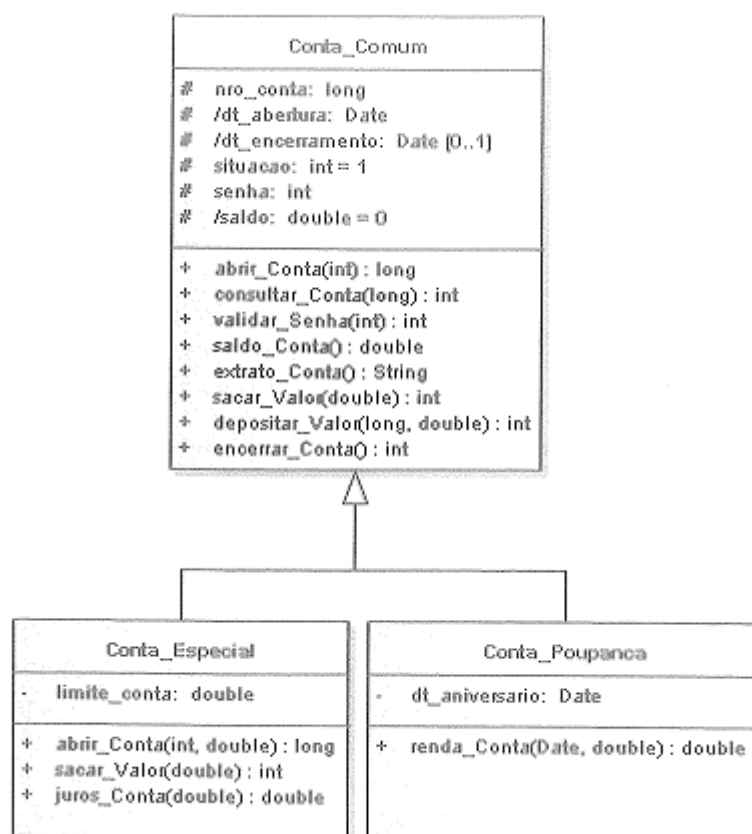


Figura 19 - Exemplo de Especificação/ Generalização (GUEDES, 2011)

Se ocorrer de duas classes possuírem a multiplicidade muitos em ambas as pontas da associação, deve ser usado uma classe chamada associativa, como na Figura 20. Esse tipo de classe serve para identificar qual classe está associada à outra.

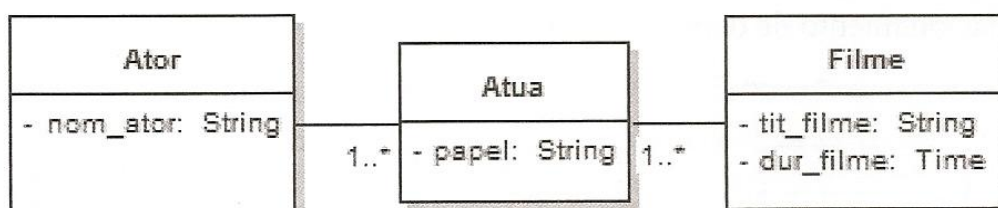


Figura 20 - Exemplo de Classe Associativa (GUEDES, 2011)

Como percebe-se, a associação entre classes é uma parte importante nesse tipo de diagrama. Em alguns casos há a necessidade de definir o quanto uma classe depende da outra e, para essas situações, há o relacionamento de dependência que é representada por uma linha pontilhada com uma seta na ponta que indica qual classe depende de qual. Em casos de uso já vimos um exemplo desse tipo de associação que são os estereótipos *include* e o *extend*. Esse tipo de

associação pode ser visualizada na Figura 21 na qual, uma classe chamada Placa-mae, depende em algum grau da classe iMonitor. (GUEDES, 2011)

Na linguagem Java há uma funcionalidade chamada *implements* que nada mais é que uma classe utilizando métodos de outra classe. Para suprir essa necessidade há o relacionamento de realização que junta características de generalização e de dependência. Sua representação é formada por um relacionamento pontilhada e na ponta uma seta vazia. Na Figura 21 à seguir podemos verificar como funciona um relacionamento de realização que indica que a classe Monitor implementa algum método da classe iMonitor. (GUEDES, 2011)

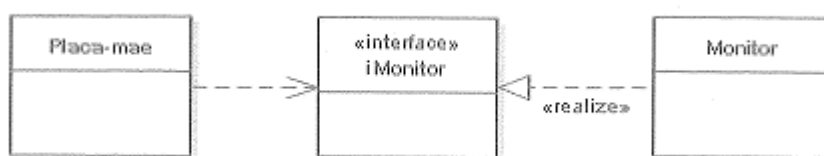


Figura 21 - Exemplo de Dependência/Realização (GUEDES, 2011)

Todo programador que estudou alguma linguagem de programação do tipo orientada à objetos, com certeza já encontrou dificuldades em entender o significado da palavra interface. Segundo (GUEDES, 2011) uma interface é “[...] um meio para descrever os serviços disponíveis de classes específicas.”, ou seja, utilidades que a classe implementada espera do ambiente em que está incluído. A classe ao implementar a interface deve respeitar as peculiaridades da interface, ou como cita (GUEDES, 2011) “Ao implementar uma interface, uma classe suporta o conjunto de características contidas por esta e obedece às suas restrições.”

Ela possui dois tipos as fornecidas e as requeridas. Sua representação, exemplificada na Figura 22, requer uma porta, representada por um quadrado, e uma linha que conecta à outra forma, que é o que define qual a variação da interface que ela representa. As interfaces do tipo fornecida são representadas por círculos e tem o objetivo de indicar o serviço implementado, ou como o próprio nome diz, fornece ao desenvolvedor a informação de qual serviço será necessário para a classe funcionar perfeitamente. Por outro lado a interface do tipo requeridas são representadas por semi-círculos abertos que mostram quais serviços serão usados pelo ambiente. Normalmente os dois tipos são representados juntos, já que um complementa o outro, como pode ser visualizado na Figura 22. (GUEDES, 2011)



Figura 22 - Exemplo de interfaces e suas variações (GUEDES, 2011)

Quando uma classe possui limites pode ser utilizada a ferramenta da UML chamada de restrição, embora seja recomendado usar com parcimônia já que existem outros diagramas que realizam essa função, além de que muitas restrições podem deixar esse tipo de diagrama muito complexo, de forma a perder seu objetivo. Dentro das restrições há o atributo estático, que é o atributo no qual seu valor não pode ser alterado, e para representar um atributo desse tipo basta sublinhá-lo. Para instruções mais específicas, como a necessidade de um cadastro em que a variável idade deve ter o valor de no mínimo 18 ou se há alguma limitação para uma classe interagir com outra, ela deve ser representada por um quadrado e a descrição da restrição, entre chaves, dentro dele, como mostra a Figura 23 abaixo sobre uma restrição que define que a Conta_Comum será ou de Pessoa_Física ou de Pessoa_Jurídica, nunca as duas ao mesmo tempo. (GUEDES, 2011)

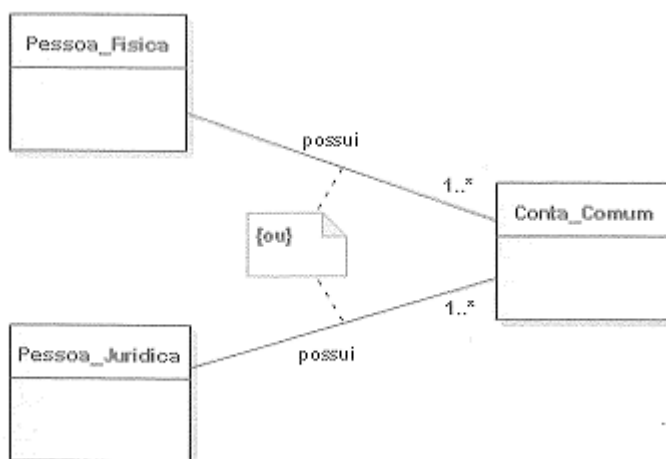


Figura 23 – Exemplo de restrição (GUEDES, 2011)

Outra restrição disponível para representação é a de coleção ordenada e possuem quatro elementos: *ordered* na qual informa que há algum tipo de ordem a ser seguida (alfabética, crescente, decrescente entre outras); *unique* que informa que dentro de uma coleção há pelo menos um atributo que não deve ser repetido e, para especificar qual atributo não deverá ser repetido, usa-se uma associação

qualificada que é representada por uma associação conectada à um retângulo que estará posicionado ao lado da classe e, dentro deste, o nome do atributo do tipo *unique*; *bag* que significa o contrário de *unique*, ou seja, permite que o elemento repita mais de uma vez; e *sequence* que é uma mistura de *bag* e *ordered*, ou seja, uma sequência. (GUEDES, 2011)

O último tipo de restrição define a relação que as classes especializadas possuem com sua classe derivada e as subclasses irmãs. As restrições desse tipo são quatro: Completa, Incompleta, Separada ou Disjunta e a Sobreposta. Quando uma classe possui todas as classes derivadas possíveis, podendo ser visualizada na Figura 24, ela é chamada de completa, caso contrário, chamamos de incompleta. Nos casos das subclasses irmãs em que são classes que não se relacionam entre si chamamos de separadas, ao contrário das restrições do tipo sobreposta em que um elemento possa pertencer a mais de uma subclasse. (GUEDES, 2011)

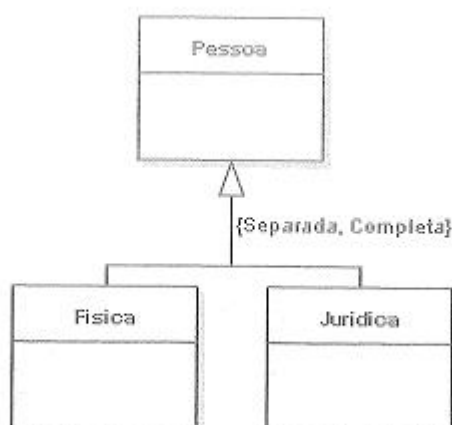


Figura 24 – Exemplo de restrição de classe especializada (GUEDES, 2011)

2.3 DIAGRAMA DE SEQUÊNCIA

Definir as classes, com seus métodos e atributos, viabiliza a criação do diagrama de sequência. Como o próprio nome diz, nesse passo irá ser definido a sequência em que as classes irão interagir entre si.

Este tipo de diagrama é do tipo comportamental pois ilustra como os métodos e atributos irão se comportar e como irão se relacionar com outras classes. O

diagrama de sequência deve ser baseado nos dois diagramas já citados, os de caso de uso e os de classes. (GUEDES, 2011)

Esse tipo de diagrama possui dois elementos principais: atores e *lifelines*. O diagrama é iniciado, normalmente, por atores, os mesmos que foram ilustrados no caso de uso, mas não são obrigatórios neste diagrama, embora sejam eles que, regularmente, iniciam um processo dentro do sistema. O outro elemento, *lifeline*, é a representação da instância de uma classe. Ao contrário dos atores, uma *lifeline* pode estar presente desde o começo do processo, como pode ser instanciada durante o processo. A representação para os atores são iguais aos casos de uso, bonecos tipo palito. Já a representação de uma *lifeline* é feita por um retângulo no qual possui no topo o nome da instância seguido do nome da classe atuante. Ambas as representações possuem linhas de vida que são linhas tracejadas saindo de suas bases e indicam, como o nome diz, o tempo de vida de um objeto, desde sua criação até o momento que o objeto é destruído e, quando isso ocorre, representamos com um “X”. As representações dos elementos desse diagrama podem ser visualizados na Figura 25. (GUEDES, 2011)

Para indicar o acionamento de um objeto no diagrama de sequência usa-se o foco de controle ou ativação. São elementos retangulares que sobrepõe a *lifeline* naquele espaço. Durante esse processo cria-se uma lacuna em branco, qual a ação realizada entre os dois objeto. Para isso usamos as mensagens ou estímulos que são setas que, com a descrição da atividade realizada em cima do corpo da mesma, indicam o sentido que a ação foi tomada seja de ator para ator, ator para objeto, objeto para objeto ou objeto para ator. Normalmente os estímulos conectam ativações com ativações mas, em caso de um objeto ser criado durante um processo, o estímulo conecta diretamente o foco de controle com o objeto ou, mais tecnicamente, o objeto é criado por um método construtor. A seta que representa a criação de instancias tem seu corpo alterado de contínuo para tracejado, como pode ser visto na Figura 25, na parte em que o Controlador_Banco aciona a instância da Conta_Comum. (GUEDES, 2011)

Quando um objeto é acionado, é normal que este retorne algum tipo de conteúdo, seja um simples *true* ou *false* ou uma mensagem. Para representar esse retorno, (GUEDES, 2011) indica o uso de mensagens de retorno que são setas

tracejadas que possuem a descrição do tipo e do nome da variável que será retornada e o sentido é contrário ao do primeiro acionamento.

Em alguns casos, o objeto não precisa utilizar um método externo por exemplo: um método presente na classe Pessoa_Fisica que verifica a autenticidade do CPF (Comprovante de Pessoa Física) e, por isso a chamada e o retorno não apontam para outro objeto e sim para ele próprio, portanto sua representação é a de uma seta com o corpo contínuo que aponta da *lifeline* para outra dentro da anterior. (GUEDES, 2011)

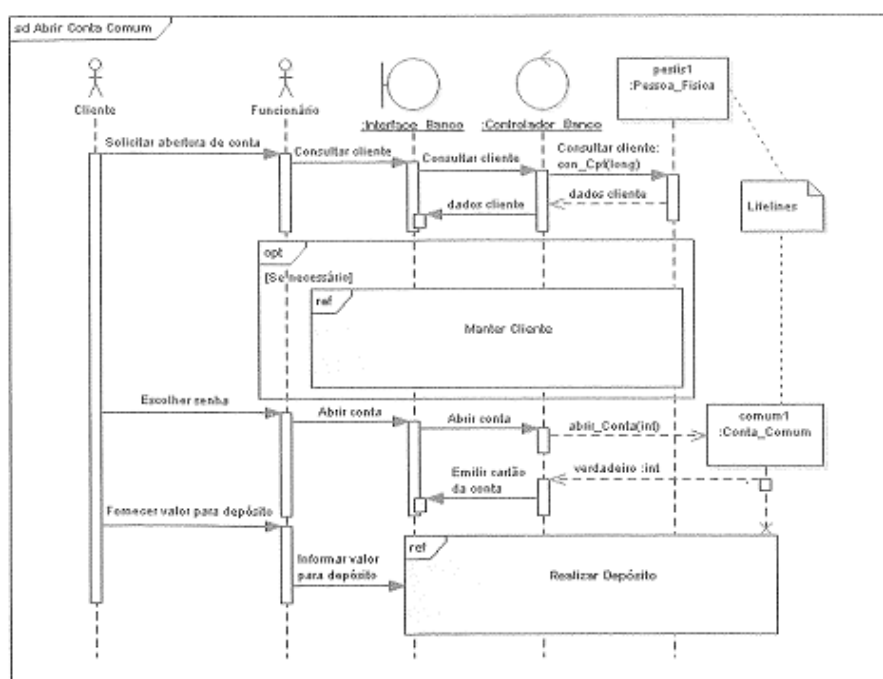


Figura 25 – Exemplo de diagrama de sequência (GUEDES, 2011)

Esse tipo de diagrama pode indicar tempo em que para demonstrar essa função no processo é adicionado uma mensagem, entre chaves, sobre o estímulo indicando a duração ou limite da ação. (GUEDES, 2011)

É muito comum em sistemas *web* ver mensagens de conexão perdida ou sobre o tempo de resposta atingir o limite definido. Isso acontece pois há duas estações que estão se comunicando entre si, mas não diretamente. Nesses casos (GUEDES, 2011) apresenta a ferramenta de mensagem perdida e a mensagem encontrada, que ambas são representadas por um círculo fechado e preenchido e possuem setas apontando para ele ou dele. Na Figura 26 é possível ver sua utilização.

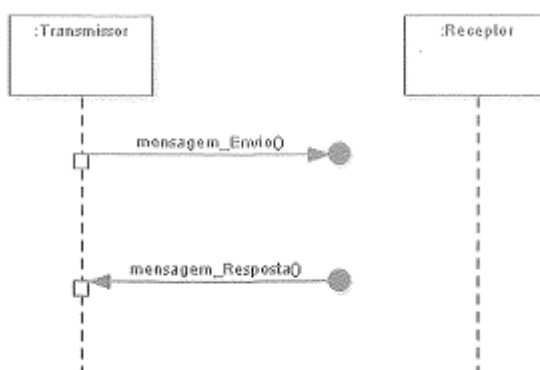


Figura 26 – Exemplo de Mensagem Perdida e Mensagem Encontrada (GUEDES, 2011)

Um objeto pode ter mais de uma *lifeline*, como nos casos em que uma classe necessita de uma interface, implementa-se o elemento porta. A representação dessas *lifelines* extras são semelhantes aos focos de controle mas não possuem conexões e ficam logo abaixo do objeto até uma certa altura, sendo que a continuação da porta é uma *lifeline*, como mostra a Figura 27. (GUEDES, 2011)

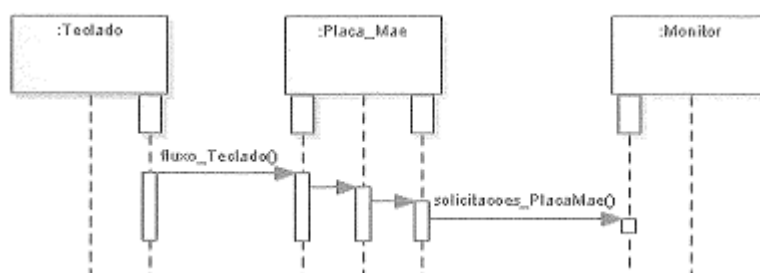


Figura 27 – Exemplo de portas (GUEDES, 2011)

Todo sistema que possua algum tipo de cadastro possuem processos que necessitam enviar e receber informações em partes. Quando temos esse tipo de interação (GUEDES, 2011) informa que são fragmentos de interação, ou seja, possuem mais de um foco de controle em uma *lifeline*.

Para que o diagrama de sequência não fique extenso e muito complexo utiliza-se usos de interação o que nada mais é que simplificar um processo representando-o com um quadrado e informando a qual subdiagrama aquele processo se refere. Representa-se esse elemento com um retângulo com o nome do processo e no canto esquerdo uma etiqueta com a nomenclatura de “ref”. Quando um processo é muito grande podem ser utilizados os fragmentos combinados em que possuem a mesma representação do uso de interação mas a etiqueta ganha a nomenclatura de “alt” e no seu conteúdo há pelo menos dois usos de interação

separados por uma linha tracejada, como pode ser visto na Figura 28.(GUEDES, 2011)

Em caso de um uso de interação for opcional utiliza-se o operador de opção, como visualizado na Figura 28, em que se o cliente possuir apenas uma conta ele irá acionar o processo Manter Cliente. (GUEDES, 2011)

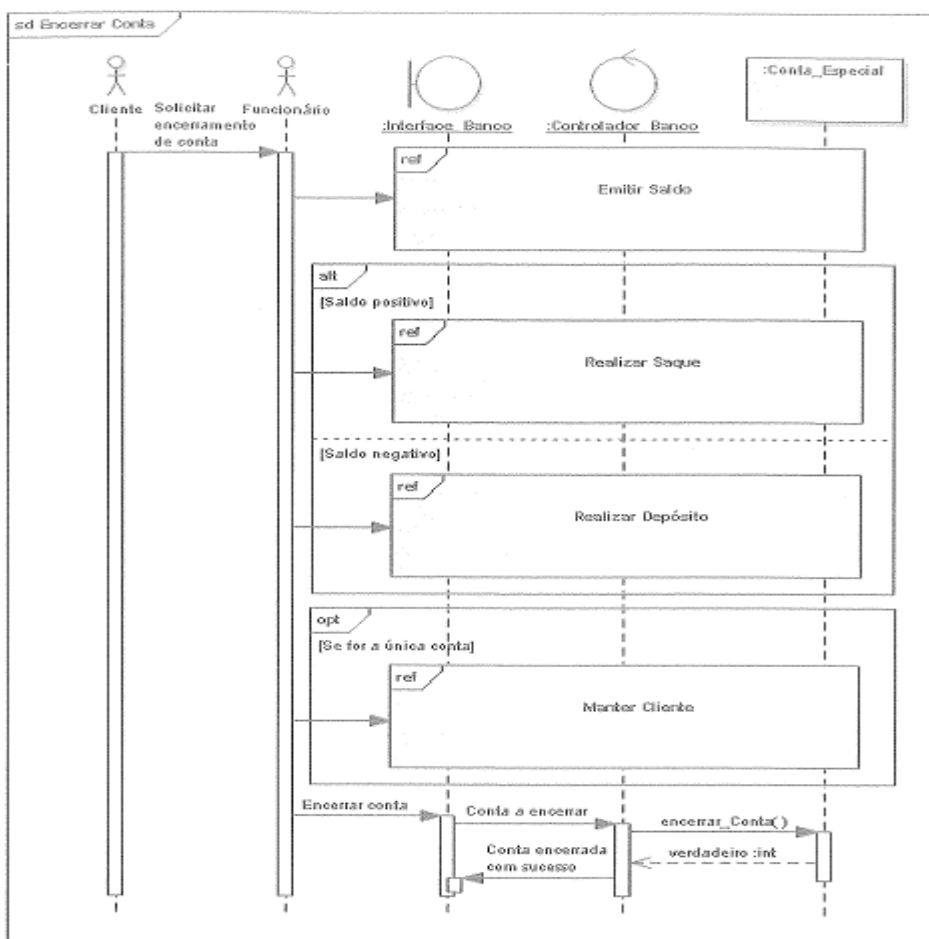


Figura 28 – Representação de usos de interação e de Operadores de Interação (GUEDES, 2011)

Para informar uma ação em conjunto utilizamos a interação paralela em que possui a mesma construção que as anteriores mas com a nomenclatura “par” e os processos que serão acionados juntos. (GUEDES, 2011)

Um site de compra coletiva pode ter o que é conhecido como carrinho. Um sistema de carrinho precisa verificar os produtos que foram adicionados pelo usuário. Para indicar essa funcionalidade, (GUEDES, 2011) cita o operador *loop* que indica que aquele processo dentro desse operador pode ser efetuado várias vezes até que, no caso do carrinho, cada produto seja verificado. É possível que haja

casos de exceções durante uma interação desse tipo e, portanto, será necessário parar o *loop*. Nesses casos recomenda-se usar o operador *break* e, assim, finalizar o processo. Se o processo do carrinho estiver presente durante a transação de pagamento é recomendado que o sistema não interrompa o processo de pagamento enquanto ele não finalizar completamente. Isso é chamado dentro da UML de região crítica e pode ser representado como as interações anteriores mas com a nomenclatura *critical*, tornando todos os processos dentro dele intocáveis por outros processos até o seu fim. Todos esse processos podem ser representados como na Figura 29.

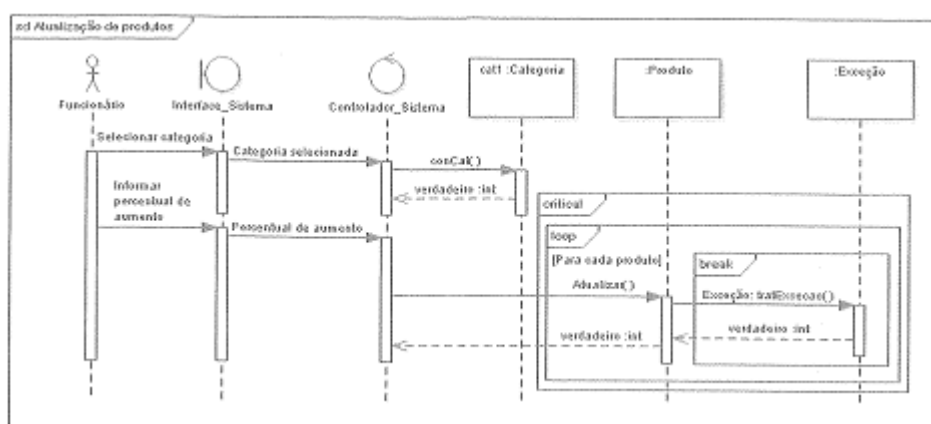


Figura 29 – Representação de Interações (GUEDES, 2011)

Segundo (GUEDES, 2011) há outros tipos de interações menos usadas como: a *neg*, que indicam eventos inválidos; a *assertion*, que age de forma contrária à *neg*, ou seja, informa eventos válidos; *ignore*, que indicam mensagens à serem ignoradas; *consider*, contrário ao *ignore*, indica mensagens válidas; *seq*, que são ocorrências de evento que mantêm as ordenações das ocorrências, qualquer ordem não afetará o processo se as ocorrências forem de *lifelines* diferentes e em caso que as ocorrências pertencentes à uma mesma *lifeline* serão ordenadas de acordo com a ordem do operando; *strict* que é a forma ordenada da *seq*.

O último elemento desse tipo de diagrama é o invariante de estado que, em resumo, se trata da representação de uma restrição dentro de processos. Este elemento é representado por um retângulo com pontas arredondadas e dentro a condição, entre chaves, para ativar a interação. A representação pode ser visualizada na Figura 30, em que define-se um limite mínimo de dez para todos os

produtos e, caso esteja abaixo desse limite, aciona o processo que faz um pedido para aquele produto até que chegue no mínimo requerido. (GUEDES, 2011)

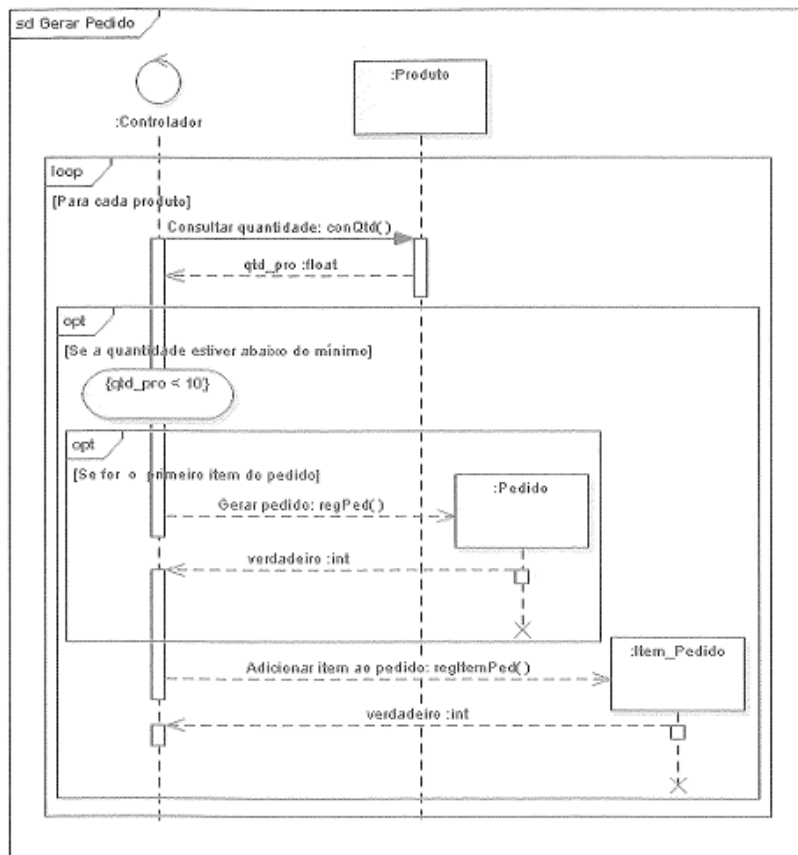


Figura 30 – Exemplo de invariante de estado (GUEDES, 2011)

2.4 DIAGRAMA DE ATIVIDADES

Entende-se por atividade como uma ação que possui o objetivo de resolver uma situação. O fluxograma é uma ferramenta que possui semelhanças com este diagrama pois, como o próprio nome já diz, ele descreve o fluxo de qualquer coisa, só que o diagrama de atividades é considerado mais completo. (GUEDES, 2011)

O principal elemento desse diagrama são as atividades. Elas são representadas por um retângulo com bordas arredondadas junto da descrição do que aquela atividade representa, como mostra a Figura 31. Ela tem um poder de representação bem ampla o que permite diversas representações como uma conta aritmética básica, uma manipulação de objetos, algumas ações relacionadas a comunicação e comportamentos. (GUEDES, 2011)

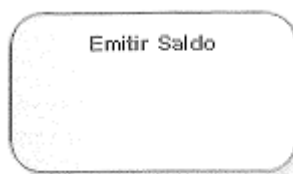


Figura 31 – Representação de Atividade (GUEDES, 2011)

Outro elemento que integra o diagrama de atividades é o nó de ação. Neste caso deve se ter um cuidado maior ao representá-lo, pois ele tem uma representação muito similar ao elemento atividade. É um retângulo com bordas arredondadas e com a descrição em seu interior, a diferença é o tamanho da representação e a impossibilidade de ser dividido. Na Figura 32 temos sete nós de ação, como o Receber número da conta, Consultar conta e outros cinco. (GUEDES, 2011)

O fluxo de controle é o elemento que une os nós de ação e é por ele que os nós de ação se comunicam. Como pode ser visto na Figura 32, existem fluxos de controle que interligam os diversos nós através de sua representação de seta, que indica qual a direção que o sinal será enviado, podendo conter alguma restrição como descrição. (GUEDES, 2011)

Percebe-se que na Figura 32 o diagrama possui um fluxo de controle conectando um círculo preenchido com o nó de ação Receber número da conta sendo o círculo o elemento indicador do início do diagrama, ou como é nomeado nó inicial. Por outro lado temos o nó final de atividade que é representada por um círculo vazio preenchido parcialmente por outro círculo completamente preenchido como a Figura 32 demonstra ao usar os fluxos de controle [Conta inválida] e [Senha inválida] para conectar um nó de decisão (losango), será compreendido a seguir, e o fim da atividade. (GUEDES, 2011)

Fluxos podem ter mais de um caminho e, nesses casos, é utilizado o nó de decisão. No diagrama ele é representado por um losango em que é o intermediário entre um nó de ação e os caminhos que este nó pode tomar, dependendo da restrição definida. Na Figura 32 uma possível consulta para login (nó de ação, Consultar conta) sobre uma conta pode resultar em conta válida ou conta inválida e cada uma dessas opções aciona o nó Solicitar senha ou finaliza o fluxo através de um nó de final de atividade. (GUEDES, 2011)

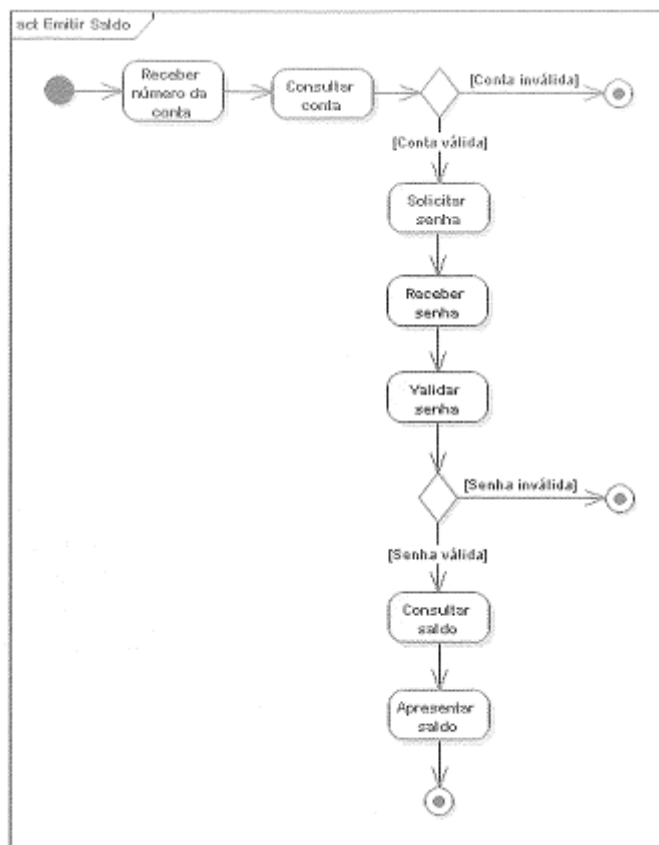


Figura 32 – Exemplo Completo de um Diagrama de Atividades (GUEDES, 2011)

Nós de ação podem ter mais de uma função funcionando ao mesmo tempo. Em um site de compra coletiva um cliente ao realizar um pagamento ativa o gatilho em que o sistema do site pode fazer o *check-in* do usuário no sistema de pagamento, enquanto calcula o valor a ser cobrado ao cliente. A ferramenta dentro de uma linguagem de programação, como PHP e JAVA, é comumente chamado de *thread*.

Em casos como esse usamos o nó de bifurcação, ou união, que são representados por uma barra vertical ou horizontal. Esse tipo de nó também pode ser usado para representar uma repetição e, portanto, há a necessidade de determinar algum símbolo para finalizar o processo, por isso usa-se um círculo com um “X” no seu interior para esse propósito. Esse tipo de nó é exemplificado na Figura 33. (GUEDES, 2011)

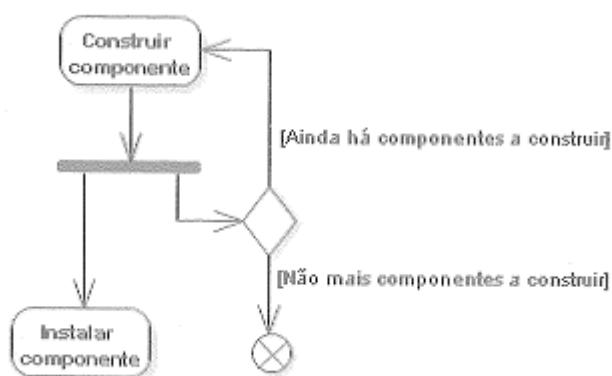


Figura 33 – Exemplo de Nó de Bifurcação com Repetição (GUEDES, 2011)

Representado por um retângulo, o nó de objeto, é a forma de demonstrar uma alteração, criação ou destruição de uma instância de uma classe. Esse processo pode ser chamado de fluxo de objetos. Para simplificar esse processo usam-se os nós alfinetes que são representados por pequenos retângulos ao lado dos nós de ação, como mostra a Figura 34. Alguns nós de objeto significam entrada ou saída de parâmetros dentro de uma atividade e são chamados de nó de parâmetro de atividade. (GUEDES, 2011)



Figura 34 – Exemplo de Alfinetes (GUEDES, 2011)

Tratamentos de exceções são comuns nas linguagens de hoje e são representados por setas em formato de raio, como é exemplificado na Figura 35. (GUEDES, 2011)

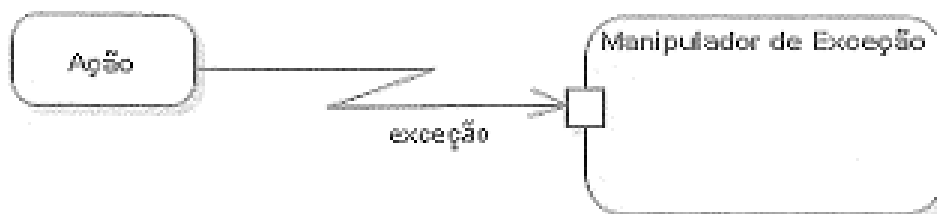


Figura 35 – Exemplo de Exceção (GUEDES, 2011)

Alguns casos requerem envio de sinais, tanto para objetos, quanto para ações. Existem dois tipos desse nó: a ação de evento de aceitação e a ação de tempo de aceitação. (GUEDES, 2011)

O primeiro tipo de envio de sinais é representado por um retângulo com o lado direito em formato de triângulo, demonstrado na Figura 36, e ocorre quando há a necessidade de uma confirmação ou aceitação, como o nome já supõe. (GUEDES, 2011)



Figura 36 – Exemplo de Ação de Evento de Aceitação (GUEDES, 2011)

O outro caso de envio de sinais, nomeado de ação de evento de tempo de aceitação, é uma variação do primeiro tipo e pode ser considerado uma *trigger* ou um *job*. A ativação de um objeto ou ação vai depender do tempo pré-definido. A representação desse tipo de envio de sinal é parecido com o de uma ampulheta, como pode ser confirmado na Figura 37. (GUEDES, 2011)

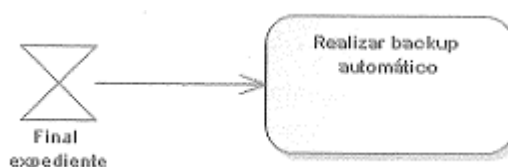


Figura 37 – Exemplo de Ação de Evento de Tempo de Aceitação (GUEDES, 2011)

Nós de *buffer* são utilizado em casos em que há fluxos sendo enviados de diferentes lugares e recebidos por vários outros. (GUEDES, 2011)

Um site de compra coletiva possui ofertas que são armazenadas em um banco de dados, nesse caso o MySQL. Essa atividade de armazenar os dados de uma oferta é um tipo de *buffer*.

Para o caso citado a cima, em que é usado uma ferramenta de armazenar dados, usa-se o nó de repositório de dados que é um nó objeto com o rótulo <<*datastore*>> seguido da descrição, como é representado na Figura 38. (GUEDES, 2011)



Figura 38 – Exemplo de nós de buffer do tipo nó de repositório de dados (GUEDES, 2011)

Quando há uma distância muito grande entre nós usa-se conectores. Sua representação se resume a pares de círculos com algum tipo de descrição sobre qual é a ponte criada entre os nós objetos, como visualizado na Figura 39. (GUEDES, 2011)



Figura 39 – Exemplo de nós de buffer do tipo nó de repositório de dados (GUEDES, 2011)

Como já foi citado antes, o sistema de qualquer e-commerce, incluindo o de compras coletivas, possui um sistema de pagamento responsável por gerenciar os pagamentos feitos pelos clientes. Normalmente o site de vendas não possui conhecimento do processo, no máximo que ele receberá o dinheiro e irá pagar ao site, retirando a parte presente em um acordo firmado em contrato. Para diagramar esse processo realizado pelo sistema de pagamento seria bem difícil, até impossível por questões de segurança e, portanto deve-se utilizar ações de chamada de comportamento.

Essas ações são utilizadas nos casos em que um diagrama já é conhecido ou não se sabe a procedência deste. Na Figura 40, há uma representação desse nó que se parece com um nó de objeto normal, embora haja uma diferença: o símbolo que deve estar presente no interior do retângulo. Este símbolo é o desenho de um tridente de ponta cabeça. (GUEDES, 2011)

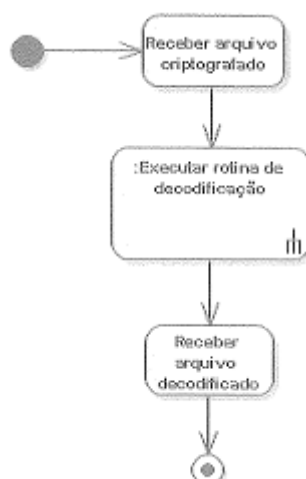


Figura 40 – Exemplo de Ações de Chamada de Comportamento. (GUEDES, 2011)

Acima vê-se a chamada de uma atividade, mas nos casos em que o diagrama necessita representar a chamada de um método e, para isso, usa-se a ação de chamada de operação. Sua representação, pode ser visualizada na Figura 41, e é basicamente um nó de objeto com a descrição do método utilizado entre parênteses, começando pela classe em que o método pertence e logo depois o método utilizado, em que os dois são separados por dois símbolos de dois pontos. (GUEDES, 2011)

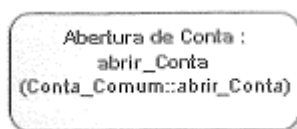


Figura 41 – Exemplo de Ações de Chamada de Operação. (GUEDES, 2011)

Dentro do processo de compra em um site de e-commerce há respostas que interagem com diferentes partes da empresa e com o cliente. Ao pagar com o cartão de crédito, por exemplo, o sistema de pagamento irá retornar para o site e para o cliente uma mensagem de confirmado ou rejeitado.

Para separar os setores afetados durante esse processo é utilizado partições ou partição de atividade. Como pode ser visualizado na Figura 42, essa partição é representada por etiquetas com a nomeação de cada parte. (GUEDES, 2011)

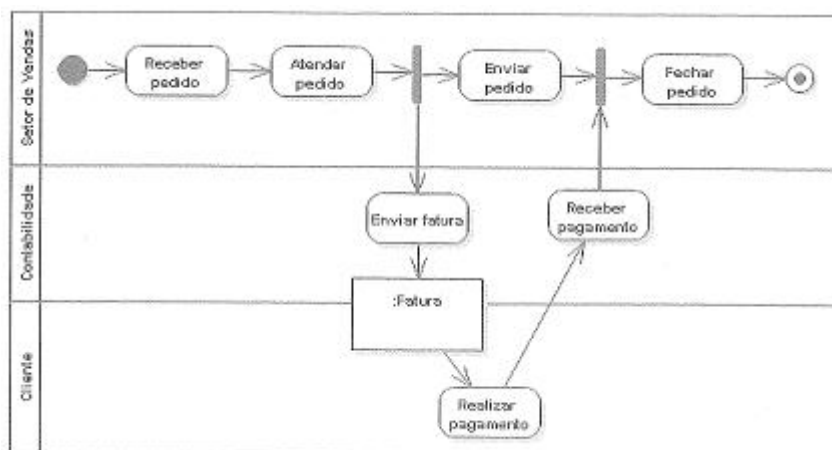


Figura 42 – Exemplo de Partição de atividade. (GUEDES, 2011)

É comum o cliente escrever reclamações ou sugestões diretamente com a administração do site de compras coletivas. Enquanto o usuário está na tela em que escreve a mensagem há a possibilidade de cancelar ou reescrever a mesma. Após clicar em enviar mensagem, não haverá uma forma direta do usuário cancelar o processo em que o sistema de gerenciamento de mensagens do site emitirá um aviso de novas mensagens ao responsável.

Para isso é utilizado a região de atividade interrompível, em que dentro da área definida o encerramento ou alteração de processos é possível. A área em que essa interrupção pode ocorrer deve ficar dentro de um retângulo com limites formados por uma linha tracejada, como mostra a Figura 43.

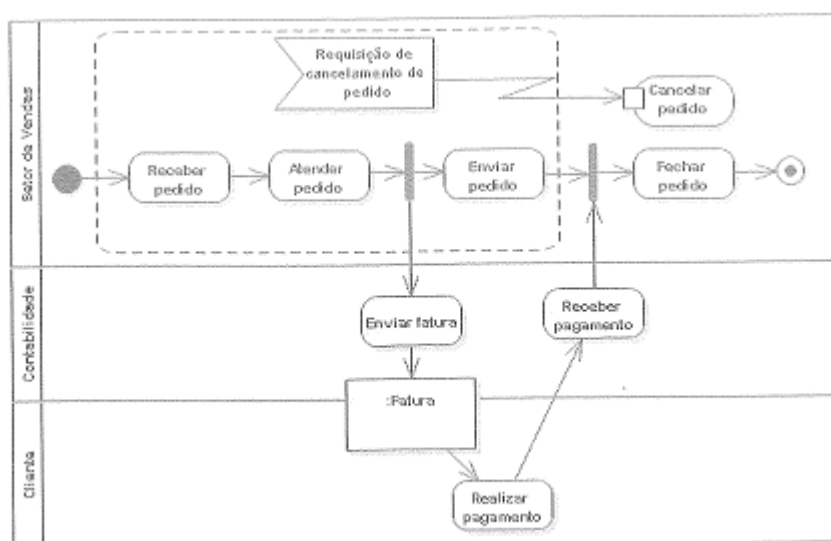


Figura 43 – Exemplo de Região de Atividade Interrompível. (GUEDES, 2011)

Um nó de atividade estruturada é um nó executável que não é dividida com outro nó estruturado e podem ter nós secundários, mas estes nós secundários não podem ser subordinados à outros nós do tipo atividade estruturada. Sua representação, como na Figura 44, é a de um nó de objeto, mas possui borda tracejada, o símbolo de diagrama de atividade em forma de oito deitado e o estereótipo <<structure>>. (GUEDES, 2011)

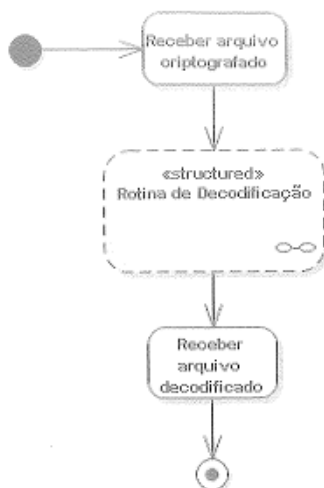


Figura 44 – Exemplo de Nó de Atividade Estruturada. (GUEDES, 2011)

Uma região de expansão funciona da mesma forma que o nó de atividade estruturada, mas com a diferença que o nó é executado de forma repetitiva pelo mesmo nó estruturado. Um exemplo prático é o cálculo fatorial como pode ser visto na Figura 45.

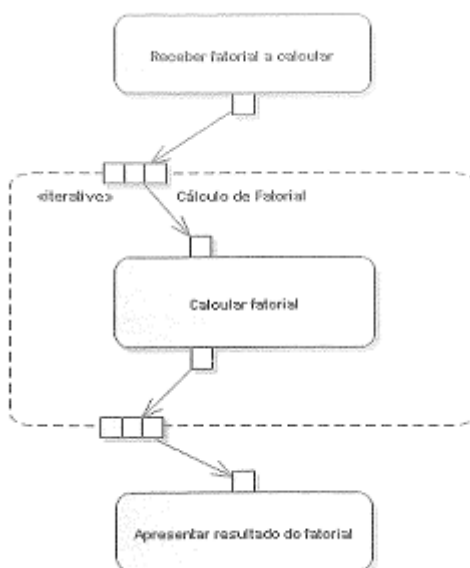


Figura 45 – Exemplo de Região de Expansão. (GUEDES, 2011)

O próximo capítulo irá além da diagramação e deve ser ter claramente as ferramentas necessárias para que o sistema rode em sincronia com o site.

3. FERRAMENTAS DE TRABALHO

O sistema será desenvolvido para funcionar junto do CMS (Sistema de Gerenciamento de Conteúdo) que já está implementado, portanto, todas as ferramentas foram escolhidas para que os sistemas possam funcionar em harmonia.

Segundo (NIXON, 2014) uma linguagem *front-end* ou *client-side* é toda a ferramenta que roda inteiramente no computador do usuário através de um navegador. Por outro lado existem as linguagens *back-end* ou *server-side* que são as ferramentas que rodam nos servidores em que o site está hospedado.

O projeto é composto por ambos os tipos de linguagem: o HTML, o javascript e o CSS que são do tipo *front-end*; do tipo *back-end* foi usado o PHP. Além dessas linguagens foi usado o MySQL que é o gerenciados de banco de dados nativo do PHP.

3.1 HTML

Baseado em XML, o HTML (Hypertext Markup Language ou Linguagem de Marcação de Hipertexto) é uma linguagem *front-end* responsável pela construção visual de um site. Sua estrutura é baseada em *tags*, ou etiquetas, e são representadas por colchetes e a descrição no seu interior, como mostra a Figura 46, em que há várias etiquetas representando o básico de uma página *web*. (NIXON, 2014)

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head><title>simple page</title></head>
  <body>
    <script type="text/javascript">
      document.write('Hello World!');
    </script>
    <noscript>
      <p>Your browser either does not support JavaScript, or you have
        JavaScript turned off.</p>
    </noscript>
  </body>
</html>
```

Figura 46 – Exemplo de Código HTML. (NIXON, 2014)

As *tags* HTML possuem duas formas de identificação: o *id* e a *class*. Ambos tem a função de tornar os elementos HTML maleáveis, seja usando CSS ou Javascript. Sua funcionalidade será melhor explorada à seguir. (NIXON, 2014)

3.2 CSS

O CSS (*Cascading Style Sheets* ou Folhas de Estilo em Cascata) é uma ferramenta *front-end* que está praticamente em todo projeto *web*, pois é através dela que um site ganha um formato amigável. Sua função, como o nome já diz, é a de estilizar o site, trabalhando e alterando as propriedades das etiquetas do HTML. (NIXON, 2014)

3.3 JAVASCRIPT

Javascript é uma ferramenta *front-end* que trás dinâmica para um site, seja movimentando uma imagem na tela, seja verificando um campo ou gerando uma janela de alerta no navegador. É uma linguagem fracamente tipada, ou seja, há a possibilidade de alterar as variáveis facilmente, tanto o valor quanto o tipo, embora a sintaxe seja muito parecida com a linguagem C, que é fortemente tipada. (NIXON, 2014)

O projeto contou com a ajuda de uma biblioteca dessa linguagem: o JQuery. Segundo o próprio site da ferramenta, é uma coleção de códigos que tem o objetivo de simplificar a codificação javascript, pois além de ter funções úteis prontas para a programação do dia-a-dia, conta com a vantagem que é compatível com os navegadores mais usados.

3.4 PHP

Assim como o javascript, o PHP, é uma linguagem que possui similaridades com a linguagem C e é fracamente tipada mas, ao contrário do javascript, o PHP é uma linguagem *back-end*, ou seja, uma linguagem que roda apenas no servidor. (NIXON, 2014)

3.5 MySQL

Um site de compra coletiva não é um simples aplicativo, é um sistema complexo que possui a necessidade de armazenar informações e o MySQL é o SGBD escolhido, pois, assim como o PHP, é uma ferramenta leve, potente, além de possuir escalabilidade e seu uso é gratuito. Segundo (NIXON, 2014) um banco de dados é um conjunto de dados organizados de forma que tenha um rápido tempo de resposta.

Para desenvolver esse sistema, o mesmo será dividido em duas partes: a análise e desenvolvimento. A análise é uma parte importante desse projeto pois, embora seja um projeto pequeno, sua funcionalidade é ampla e qualquer interferência negativa criada pelo sistema irá afetar toda a empresa. O desenvolvimento será baseado na análise minimizando as chances de que este possua deficiências críticas para a empresa.

4 ESTUDO DE CASO

O objetivo deste estudo de caso é a de permitir que todos os setores da empresa possam fiscalizar e definir rumos com base nos dados gerados através dos dois sistemas em estudo. É importante destacar que a empresa possui processos parecidos com os que serão desenvolvidos, mas ou são manuais ou necessitam de interferências de outras áreas, tornando qualquer processo que envolva dados demorado e pouco confiável.

Essas conclusões foram tiradas à partir da análise, que pode ser dividida em duas partes: as dificuldades e as soluções. Em ambas há um contato direto com os usuários, ou pelo menos com o responsável, na qual é feito o levantamento de requisitos e, assim, defini-se as propriedades do sistema.

4.1 DIFICULDADES

A empresa estudada situa-se em Americana/SP e é uma empresa de porte pequeno possuindo poucos responsáveis por área, ou seja, qualquer interferência faz uma grande diferença. Durante o levantamento de requisitos percebeu-se que diariamente os setores de vendas, administrativo, financeiro e, em menos casos, o de marketing, pediam dados da empresa que apenas quem tivesse acesso ao banco de dados teria acesso à eles, ou seja, o programador e o responsável pelo setor de TI, atrasando projetos que, as vezes, eram extremamente importantes para o crescimento da empresa.

O responsável por esse problema são as constantes alterações no sistema que buscam otimizar a relação cliente e site, mas que não possuem o acompanhamento adequado do sistema administrador. A alteração que mais afeta o sistema atual é o que permite que uma oferta tenha subofertas, como se fossem variações da oferta principal. Um exemplo seria uma oferta de um tênis, em que o cliente pode escolher a cor e o número que, no entanto, o sistema não informa o quanto foi vendido de cada suboferta, nem qual o valor, em créditos, que os clientes usavam entre outros dados importantes. Em resumo o site adquiriu uma ótima

ferramenta que solucionava problemas antigos com clientes, mas que criava outros para os setores da empresa.

Outro caso que ocorria diariamente e que requeria a atenção do setor de TI era o de pagamento de parceiros. Dentro desta operação haviam vários pontos que deviam ser contabilizados, tanto para prestar contas para as entidades governamentais quanto para a gerência. Se por acaso o responsável pelas finanças não conseguisse identificar quantos estornos ocorreram, a empresa acabaria ficando com prejuízo, sendo que o pagamento seria de acordo com os pedidos pagos e não de acordo com os pedidos concluídos. Além disso, todos os cálculos eram feitos manualmente e passados para planilhas do Excel pois, caso o parceiro discordasse do valor recebido, esta era forma de comprovar, junto dos cupons que ele recebeu, que o pagamento estava certo. Caso a reclamação persistisse, o setor de TI era acionado novamente para verificar se nenhum meio de pagamento deixou de enviar o código de confirmação de pagamento, uma falha no sistema ou até mesmo casos de fraude.

4.2 PROPOSTA DE SOLUÇÃO

Para solucionar os problemas, foram marcadas reuniões com os gerentes de cada setor junto de seus subordinados e, a solução encontrada, foi criar um sistema financeiro/administrativo auxiliar dividido em duas partes.

O primeiro era um projeto simples que consistia em criar uma página que informasse dados básicos de uma oferta e suas subofertas, como a quantidade do número de vendas, vendas no dia, vendas no total entre outros, ou seja, um acesso rápido e eficiente às informações importantes para a área de vendas, que teria agora uma visualização melhor, definindo a possibilidade de repetir ou não a oferta. Para a área de *marketing*, esse levantamento ajudaria a definir os valores gastos em *marketing* digital de acordo com o sucesso da oferta, como Facebook e GoogleAds.

A segunda parte do projeto é a mais crítica, pois envolve pagamentos, ou seja, qualquer erro pode causar desconfiança do serviço prestado pela empresa de compra coletiva, gerando um *marketing* negativo. O resultado dos cálculos serão

criados em um arquivo do Excel e disponibilizado para *download* para o usuário requisitante, podendo ele ser da área administrativa do site ou financeira da empresa.

O modelo de processo de desenvolvimento utilizado foi o de cascata e, embora (SOMMERVILLE, 2007) cite que não seja o melhor modelo para desenvolvimento de sistemas, ele também cita que este modelo pode ser usado em projetos que possuem requisitos bem especificados, que sejam apenas uma parte de um projeto maior e que não haja probabilidade de grandes alterações, ou seja, é exatamente a descrição desse projeto.

4.3 DESENVOLVIMENTO

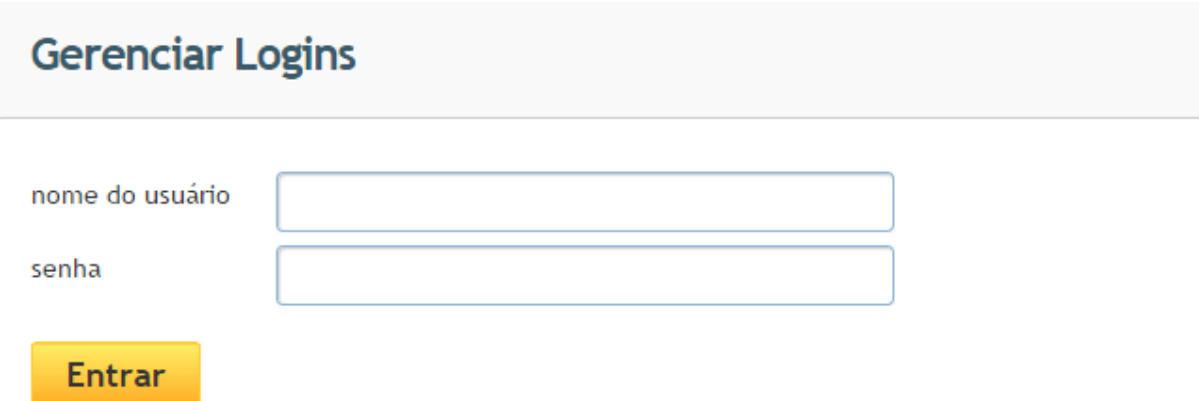
O site de compra coletiva estudado, representado na Figura 47, é baseado em uma empresa real. Para que o sigilo seja mantido, o nome e o *design* foram alterados.



Figura 47 - Site de compra coletiva

Foi definido em acordo com os setores envolvidos que a prioridade desse projeto seria o sistema de levantamento de dados de ofertas, que é mais simples e rápido de desenvolver. Após a conclusão da primeira fase todos os setores da empresa serão beneficiados com os dados sendo gerados quase em tempo real. Por fim será desenvolvido sistema de lista que precisará de um cuidado maior.

Para acessar o CMS do site deve ser digitado a URL : <http://www.compracoletiva.com.br/admin>. Aparecerá uma página, como na Figura 48, em que mostrará dois campos, um para nome do usuário e outro para senha. Ao inserir os dados, o sistema irá procurar a conta e caso o usuário tenha permissão o sistema irá ir para a página principal do CMS.



The image shows a web interface for managing logins. At the top, there is a header with the text "Gerenciar Logins". Below the header, there are two input fields. The first is labeled "nome do usuário" and the second is labeled "senha". Below these fields is a yellow button with the text "Entrar".

Figura 48 - Tela de *login*

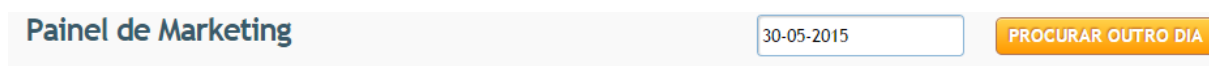
O responsável pelo site é responsável de atribuir permissões à essas contas e, além de permitir o acesso ao CMS, há ainda a possibilidade de limitar o acesso. Através da função demonstrada na Figura 49, o sistema verifica se a conta que está requisitando o acesso tem permissão para ver o conteúdo da página. No primeiro projeto todos os usuários da empresa tem acesso ao sistema, no entanto, no segundo sistema apenas o financeiro e a administração do site terão esse acesso. Na figura abaixo pode ser vista a palavra *market*, que significa que todos os que tiverem acesso à esse nível, poderão entrar na página. Para que esse trabalho não seja um risco à empresa estudada, a função não será melhor detalhada.

```
1 <?php
2 require_once (dirname(dirname(__FILE__))) . '/lib/require_auth.php';
3
4 // Verifica se o usuário tem permissão para acessar a página
5 need_auth('market');
6
7 // Continua a implementação da função
```

Figura 49– Demonstração da função utilizada para restringir acesso

4.4 SISTEMA DE LEVANTAMENTO DE DADOS DE OFERTAS

O objetivo desse sistema é facilitar a visualização de dados sobre ofertas, baseados na data que o usuário digitar. Essa página é chamada de Painel de Marketing, e é o local em que será mostrado o número de vendas da oferta, quanto a oferta faturou, a posição de importância e as datas de entrada e saída, como mostra a Figura 50. Ao carregar a página as ofertas que estão no ar, no dia em que o sistema é acessado, serão mostradas. Percebe-se que no topo da página há um campo com o formato de data e um botão com a descrição de Procurar Outro Dia e é esse campo que permite a visualização de ofertas de outros dias, basta escolher a data.



Painel indicador de marketing das ofertas online:

ID	Nome da oferta	Datas	Vendas	Faturado	Ordem
2	Deu a louca no gerente: Churrasco por apenas R\$ 5?!?!?! Valor: R\$5,00 Max: 0 >ver subofertas	Início: 29-05-2015 Fim: 30-12-2015	Total: 0 Hoje: 0	Hoje: R\$0 Total: R\$0	0
1	Deu a louca no gerente: Churrasco por apenas R\$ 5?!?!?! Valor: R\$5,00 Max: 0 >ver subofertas	Início: 29-05-2015 Fim: 30-12-2015	Total: 0 Hoje: 0	Hoje: R\$0 Total: R\$0	0

Figura 50 - Painel de marketing.

Outra parte importante é nos casos em que a oferta possui subofertas. Na figura acima existem dois links: um que dá acesso à oferta e outro que dá acesso aos dados das subofertas. Ao clicar no segundo link irá abrir a página, visualizada na Figura 51, com as mesmas informações da anterior, mas relativas às subofertas.

ID	Suboferta	Valor Unitário	Vendas	Faturado
4	Churrasco de maminha	5.00	0	Total: R\$0 .Créditos:R\$0
3	Churrasco de frango	5.00	0	Total: R\$0 .Créditos:R\$0

Figura 51 - Página de dados de subofertas

Para pesquisa no banco de dados é utilizado um método que busca os dados relacionados a quantidade de vendas e faturamento, e é chamada de *Count*. Esse método faz parte do objeto *Table* e, para que funcione perfeitamente, ele precisa de

três variáveis, o nome da tabela que será pesquisada, as condições em formato de *array* e o campo da tabela que será contada, como mostra a Figura 52.

```
$resultadd = Table::Count('tabela', array("atributo = valor"), 'campo');
```

Figura 52 - O método *Count*

4.5 SISTEMA DE LISTA DE PAGAMENTOS

O sistema de lista de pagamentos é o mais crítico do projeto por ter como objetivo gerar uma lista em Excel com os valores que o site de compra coletiva deve pagar aos parceiros. Essa funcionalidade só poderá ser acessada pela área de TI e de finanças e, portanto, a função que controla os acessos às páginas receberá o valor *financial*.

Para iniciar o projeto, foi desenvolvido a parte *front-end* do sistema, ou seja, as linguagens Javascript, HTML e CSS. Na primeira foi usada a *tag select* na qual permite ao usuário escolher apenas as opções que o sistema permite. Na Figura 53 percebe-se que há quatro opções, a ID, Data Término da Oferta, Data Pagamento dos 80% e Data Pagamento dos 20%. A primeira opção permite ao usuário escolher qual(is) oferta(s) a lista terá. Na segunda a lista será gerada com base na data em que as ofertas foram encerradas. A terceira e a quarta são baseadas na suposta data que a empresa de compra coletiva terá que pagar o parceiro, ressaltando que a empresa retém 20% do pagamento para possíveis reembolsos e, caso não isso não aconteça, o valor dos 20% é pago quando a data de utilização dos cupons se encerrar.

Procurar por

Escolha a opção ▾
Escolha a opção
ID
Data Término da Oferta
Data Pagamento dos 80%
Data Pagamento dos 20%

BAIXAR LISTA

Figura 53 - Opções para da lista

Como foi dito, usou-se a *tag select*, que é parte da linguagem de marcação HTML. Na figura 54 percebe-se que a *tag* possui um *id* com nome de searchFor que é a forma de identificar aquele elemento para as outras linguagens CSS e Javascript. Logo após vê-se que há outras cinco *tags* do tipo *option* e cada uma delas representa as opções da imagem acima.

```
<select id="searchFor">
  <option value="">Escolha a opção</option>
  <option value="c1">ID</option>
  <option value="c2">Data Término da Oferta</option>
  <option value="c3">Data Pagamento dos 80%</option>
  <option value="c4">Data Pagamento dos 20%</option>
</select>
```

Figura 54 - *Tag select*

Ao escolher uma opção, irá aparecer os campos que cada uma delas necessita para gerar a lista. Na Figura 55 a opção escolhida é a Data Término da Oferta e, portanto, precisa receber um intervalo de tempo para procurar as ofertas que terminaram naquele período.

Procurar por

Data Término da Oferta ▾

BAIXAR LISTA

De:

Até:

Figura 55 - Campos da opção Data Término da Oferta

Essa funcionalidade de gerar campos no HTML foi possível graças ao Javascript, mais especificamente o JQuery, como mostra a Figura 56.

```

4     var flag = <?php echo detect_ie()? 'true': 'false';?>;
5     $(document).ready(function(e) {
6         $('#downlist').live('click',function() {
7             switch(opt){
8                 case 'c...
45         }
46     });
47     $('#searchFor').live('change',function(){
48         opt = $(this).val();
49         $('.categoryDL').hide();
50         $('.'+opt).show();
51         jQuery('#downlist').removeAttr('disabled');
52     });
53 });
54 </script>

```

Figura 56 - Código em JQuery que é acionado ao escolher uma das opções do *select*

Após o usuário escolher a opção que melhor se encaixa na sua necessidade o sistema envia a requisição para um arquivo PHP, que possui todos os passos e cálculos necessários para gerar a lista de pagamentos. A primeira parte é a que contém uma sequência de passos para preparar a procura pelas ofertas escolhida e, em caso de comandos inválidos, ela retorna uma mensagem de erro, como mostra a Figura 57.

```

49 //Ofertas por ID
50 if(isset($_GET['ids'])){
51     $flagId...
64 }
65 //Ofertas por mês
66 else if(isset($_GET['from']) && isset($_GET['to']) && $_GET['type'] == 'end_time'){
67     $from =...
72 }
73 //Ofertas a serem pagas os 80%
74 else if(isset($_GET['from']) && isset($_GET['to']) && $_GET['type'] == '80p'){
75     $from =... |
88 }
89 //Ofertas a serem pagas os 20%
90 else if(isset($_GET['from']) && isset($_GET['to']) && $_GET['type'] == '20p'){
91     $from =...
101 }
102 //Erro no envio dos dados
103 else{
104     echo "Erro ao gerar relatório!";
105     return;
106 }

```

Figura 57 - Definição do tipo de lista e preparação para os cálculos

O segundo passo é a parte em que os cálculos são executados e organizados dentro de um *array*. Na figura 58 podemos ver um resumo do código em que é usada a função do PHP *foreach* que, à partir de um *array* contendo as ofertas

escolhidas, irá executar um *loop*, ou rotina de repetição. Percebe-se que mesmo que seja uma oferta simples o cálculo irá executar a parte das subofertas. Isso acontece por que o sistema permite que a oferta possa expandir, ou seja, caso o parceiro queira adicionar uma opção do seu produto após o contrato, é possível adicionar a opção sem ter a necessidade de refazer a oferta toda.

```

117 foreach( $steams as $n=>$steam ){
118     if($tea...
157     //Procura-se pedidos pagos
158     foreach($orders as $o){
159         if($o['...
169     }
170     //À partir dos pedidos é calculado os valores de pagamento individualmente
171     foreach($steamM as $n2=>$mt){
172         $cardpa...
184         //À partir desse ponto o cálculo e feito por suboferta
185         foreach($opt as $o){
186             if($isT...
301         }
302     }
303     //Depois é calculado os valores como se fossem uma só oferta
304     if(strl...
379 }

```

Figura 58 - Resumo dos processos para gerar a lista de pagamentos

Ao final desse processo o sistema gera a lista através da função `down_xls()`, exemplificada na Figura 59, que deve receber na ordem: a variável que contenha um *array* com todos os dados, uma variável com os nomes dos campos e o nome que descreverá a lista.

```

/
down_xls($steamEnd, $kn, $name);
?>

```

Figura 59 - Função para download da lista

Após esses processos é gerado um arquivo do Excel, contendo todos os dados que foram calculados, como exemplo da Figura 60, e, dessa forma, o usuário poderá enviar o relatório em um formato compatível com qualquer computador que possua o pacote Office. Percebe-se que ao gerar a lista os dados são organizados em subofertas e logo após a oferta correspondente, dessa forma o setor financeiro poderá visualizar claramente o valor real para cada promoção. Caso o parceiro discorde do repasse, o responsável poderá apresentar a tabela e esclarecer da melhor forma o valor de cada oferta.

The screenshot shows an Excel spreadsheet with the following data:

	A	B	C	D	E	F	G	H	I	J
1	ID	Parceiro	ID Oferta	Oferta	Valor Oferta	Comissão	Mês Fechamento	Oferta Começa	Pagamento 80%	Oferta Termina
2	1	Churrascaria	2	Churrasco por R\$ 5(Churrasco de frango)	R\$5.00	-	-	-	-	-
3	2	Churrascaria	2	Churrasco por R\$ 5(Churrasco de maminha)	R\$5.00	-	-	-	-	-
4	3	Churrascaria	2	Churrasco por R\$ 5	R\$5.00	30%	dez/15	29/mai/15	01/01/2016	30/12/2015
5	4	Churrascaria	1	Churrasco por R\$ 5(Churrasco de frango)	R\$5.00	-	-	-	-	-
6	5	Churrascaria	1	Churrasco por R\$ 5(Churrasco de maminha)	R\$5.00	-	-	-	-	-
7	6	Churrascaria	1	Churrasco por R\$ 5	R\$5.00	30%	dez/15	29/mai/15	01/01/2016	30/12/2015

Figura 60 - Lista gerada

CONCLUSÃO

O projeto teve duas situações que atrasaram o seu desenvolvimento. As duas tiveram relação direta com o tamanho da empresa, já que ela é uma empresa de porte pequeno e sempre funcionou com o efetivo baixo. A área de TI possuía dois responsáveis pela manutenção, sendo que ambos estavam na empresa desde o início do desenvolvimento do sistema, ou seja, conheciam bem o sistema e, por esse motivo, não haviam feito nenhum tipo de documentação. Perdeu-se muito tempo nesse processo de entender o sistema e, embora o prazo definido pelo contratante fosse suficiente para o desenvolvimento do sistema, não havia tempo sobrando para criar a documentação completa do site de compra coletiva. A segunda situação ocorreu durante o período de levantamento de dados e de testes em que havia a necessidade de marcar reuniões com os setores envolvidos nas alterações, sendo eles os setores administrativo, financeiro e de vendas. Como a empresa não tinha condições de ter vários funcionários por área, cada responsável agregava muitas funções sendo que, dificilmente possuíam horário livre para essas reuniões. A solução encontrada foi a de trocar mensagens por e-mail que, embora ainda não fosse a forma adequada de resolver o problema, adiantou o que poderia ser feito sem a presença dos mesmos.

Embora esses problemas tenham atrapalhado desde a análise até o desenvolvimento, o sistema funcionou como era esperado. Durante o período de uma semana cada área envolvida reportava o comportamento do sistema, havendo algumas alterações até chegar ao sistema descrito nesse trabalho.

Ao utilizar o sistema de levantamento de dados a área de *marketing* foi capaz de visualizar o número de vendas e definir os valores gastos com Google Ads e Facebook. Segundo o setor, os investimentos em publicidade foram melhor distribuídos, diminuindo gastos desnecessários.

O setor Administrativo pode informar aos superiores ou aos clientes sobre o andamento de ofertas com maior rapidez e sem precisar interromper o setor de TI, aumentando a produtividade e a satisfação dos parceiros.

Por fim, o sistema que gera listas de pagamentos permitiu ao setor financeiro a independência do setor de TI, agilizando a entrega e a confiabilidade desses relatórios.

REFERÊNCIAS

GIL, ANTÔNIO CARLOS - **Como Elaborar Projetos de Pesquisa**. 4ª Edição. São Paulo. Editora Atlas, 2002.

SOMMERVILLE, IAN. **Engenharia de Software**. 8ª Edição. São Paulo. Editora Pearson Education do Brasil, 2007.

LAUDON, Kenneth C.; LAUDON, Jane P.. **Sistema de Informações Gerenciais**, 9ª Edição. São Paulo. Editora Pearson Education do Brasil, 2011.

Estados Unidos da América. **Dado da população americana em Março 2015**. Disponível em: <http://www.census.gov/topics/population.html> . Acesso em 23 março 2015.

UNFPA. **Dado da população Mundial em 2011**. Disponível em: <http://www.unfpa.org/population-trends> . Acesso em 23 março 2015.

SHERIFF, Paul D.. **Designing for Web or Desktop?**. Disponível em: <https://msdn.microsoft.com/en-us/library/ms973831.aspx>. Acesso em 23 Março 2015.

NATIONS, Daniel. **What is a Web Application?**. Disponível em: http://webtrends.about.com/od/webapplications/a/web_application.htm. Acesso em 25 de Março 2015 .

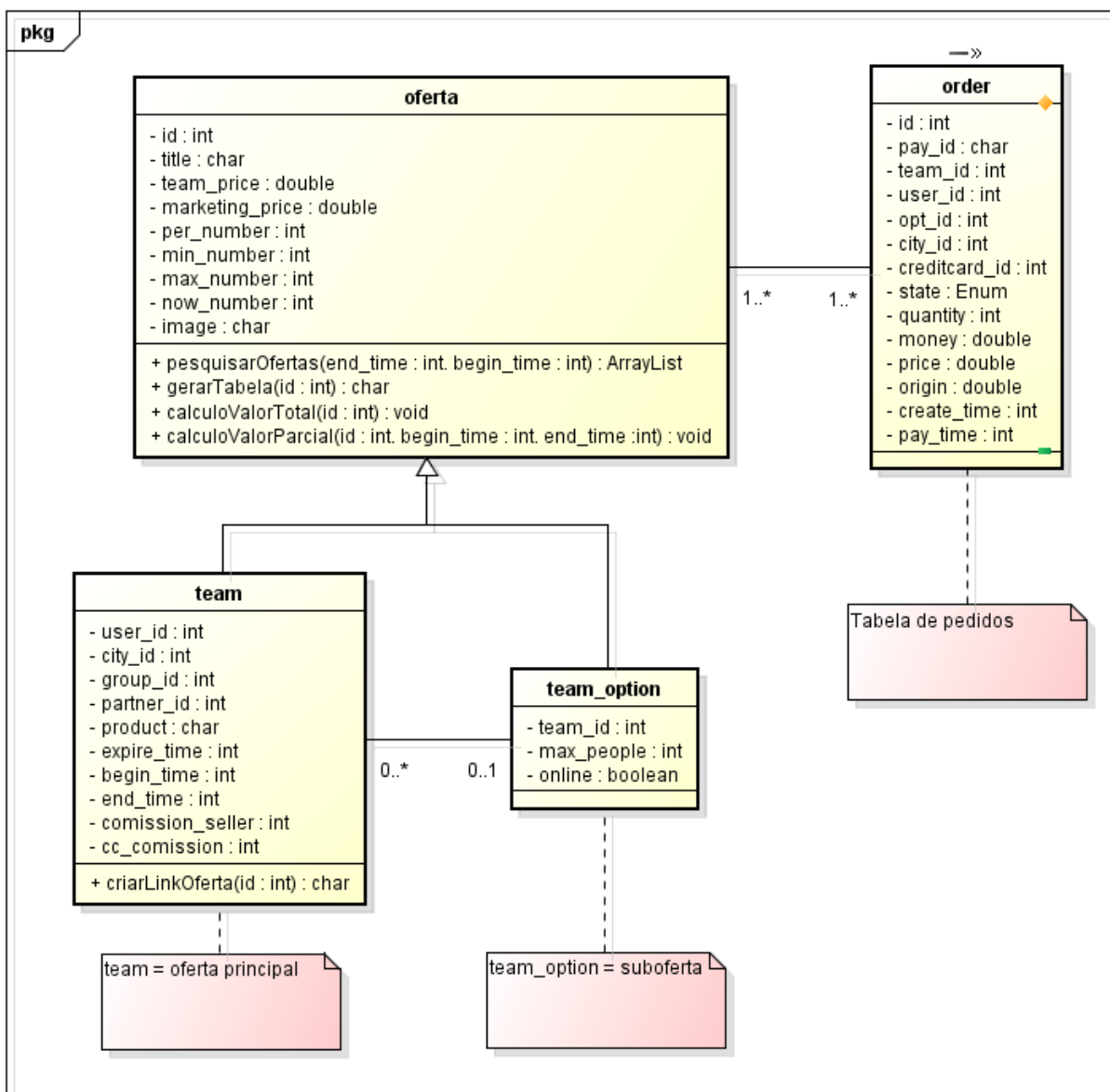
VELTE, Anthony T.; VELTE, Toby J., Ph.D.; ELSENPETER, Robert. **Cloud Computing – A Practical Approach**. Estados Unidos da América. Editora McGraw-Hill Companies, 2010.

GUEDES, Gilleanes T. A.. **UML 2 – Uma Abordagem Prática**. 2ª Edição. São Paulo. Editora Novatec, 2011.

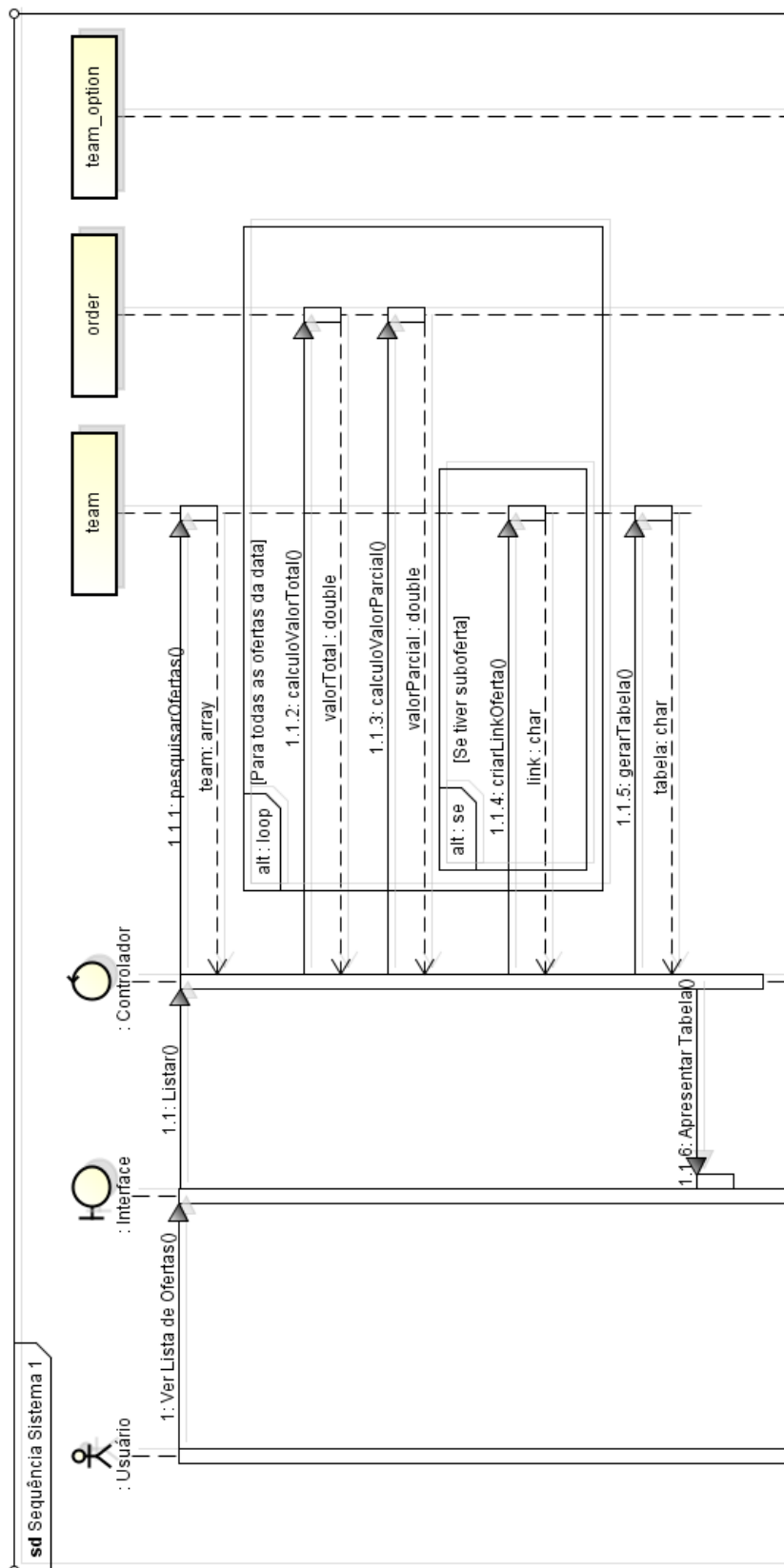
NIXON, Robin. **Learning PHP, MySQL, Javascript, CSS & HTML5**. 3^a Edição. Estados Unidos da América. Editora O'Reilly, 2014.

JQUERY. **What is JQuery?**. Disponível em: <https://jquery.com>. Acesso em 10 de maio 2015.

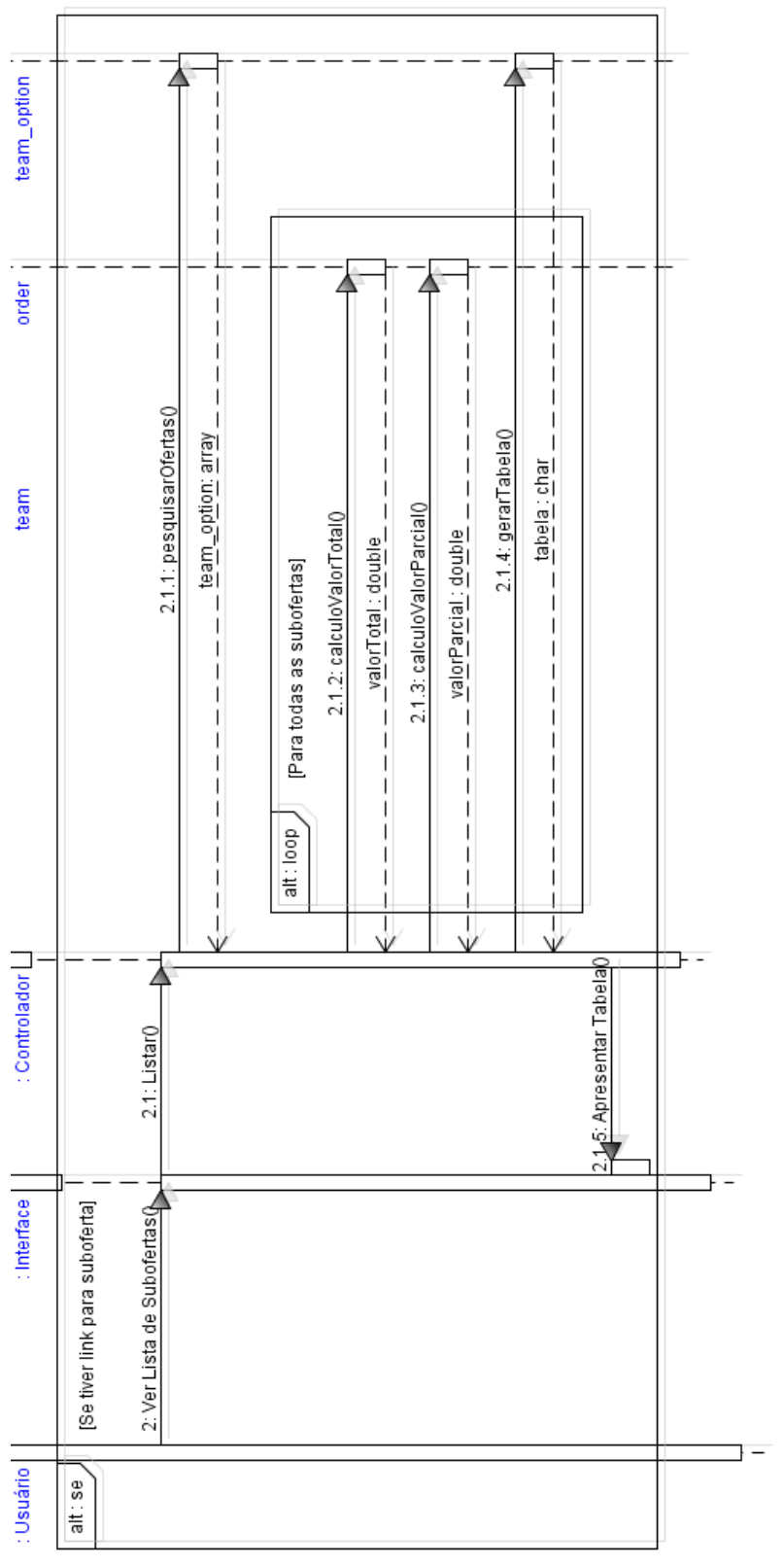
2. DIAGRAMA DE CLASSES DO SISTEMA DE LEVANTAMENTO DE DADOS



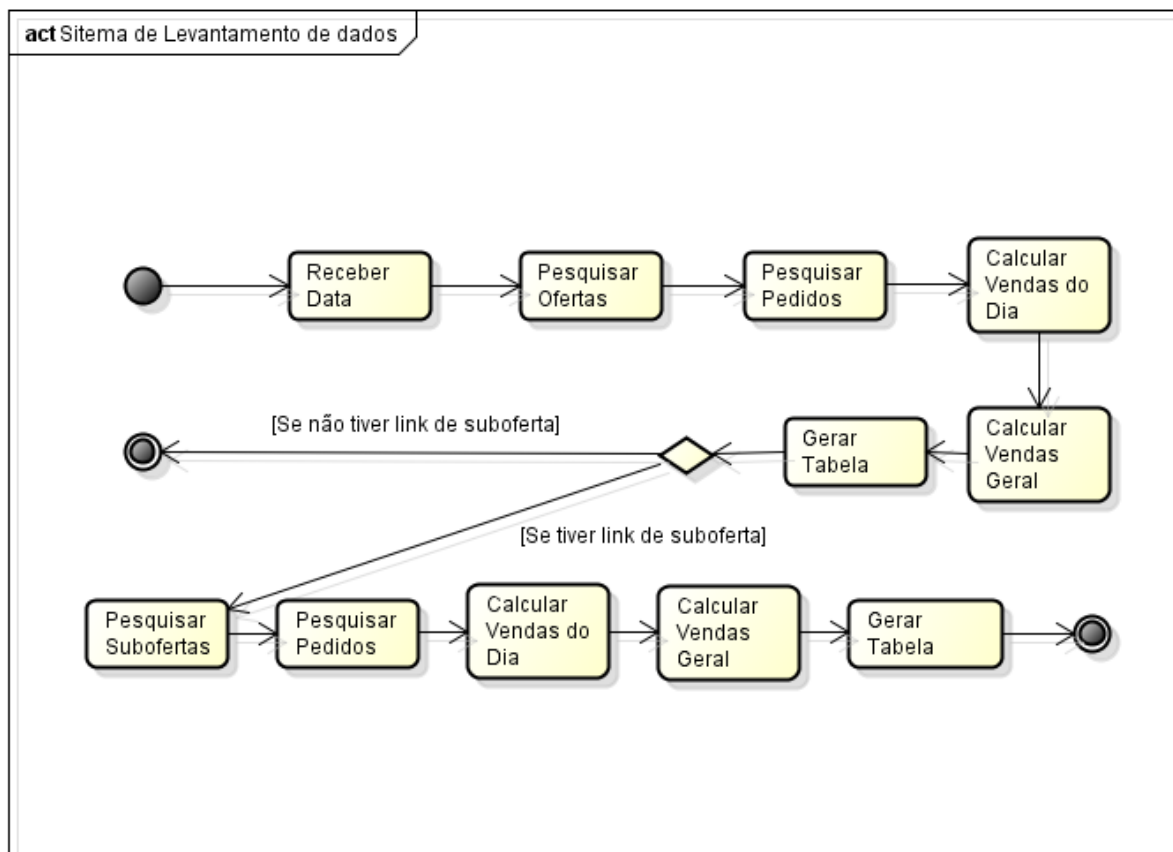
3. DIAGRAMA DE SEQUÊNCIA DO SISTEMA DE LEVANTAMENTO DE DADOS – PARTE I



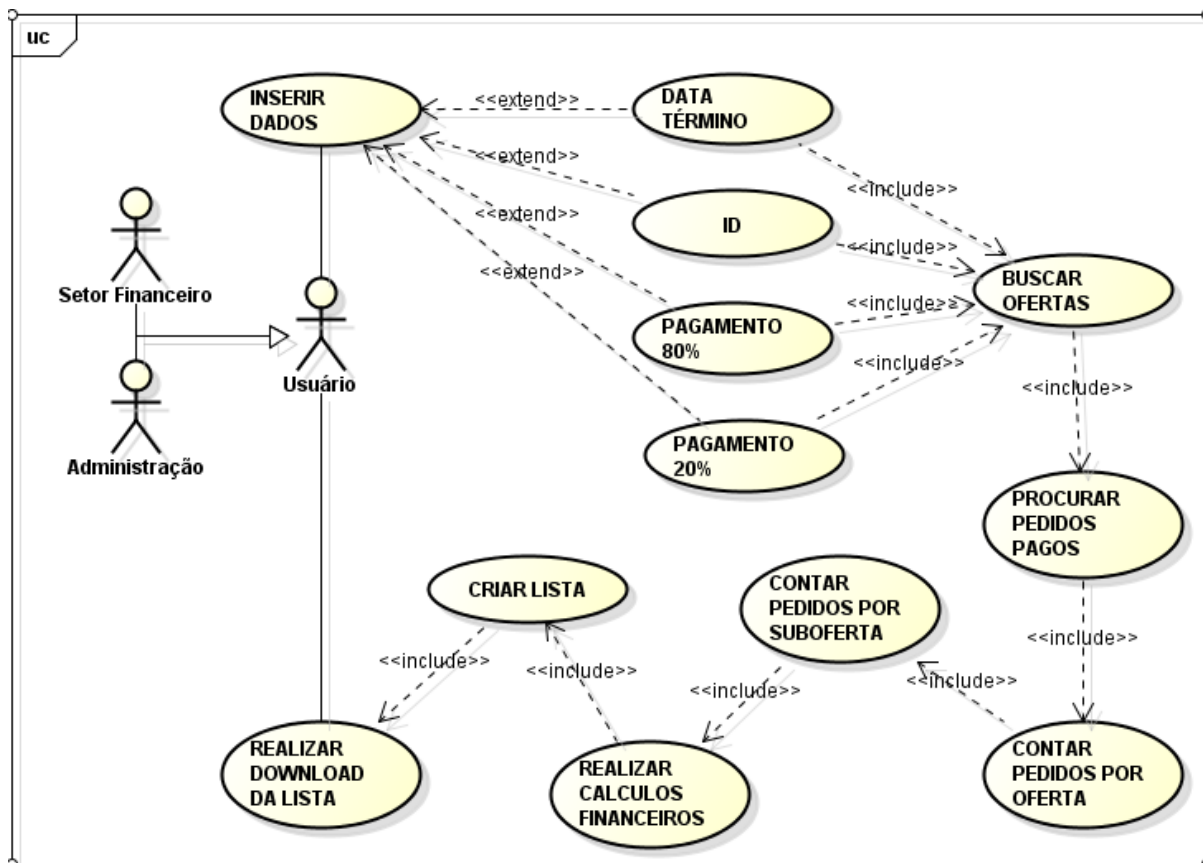
4. DIAGRAMA DE SEQUÊNCIA DO SISTEMA DE LEVANTAMENTO DE DADOS – PARTE II



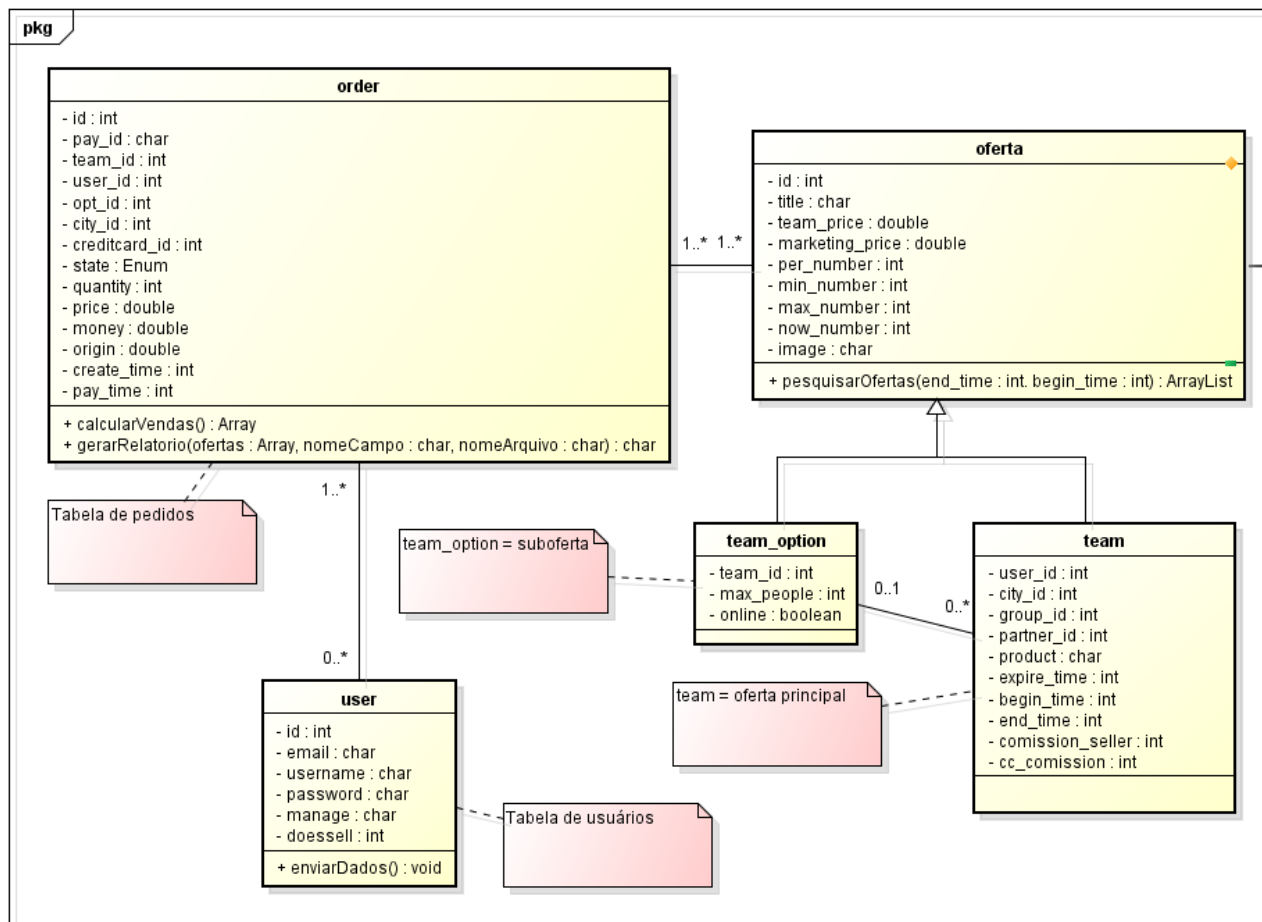
5. DIAGRAMA DE ATIVIDADES DO SISTEMA DE LEVANTAMENTO DE DADOS



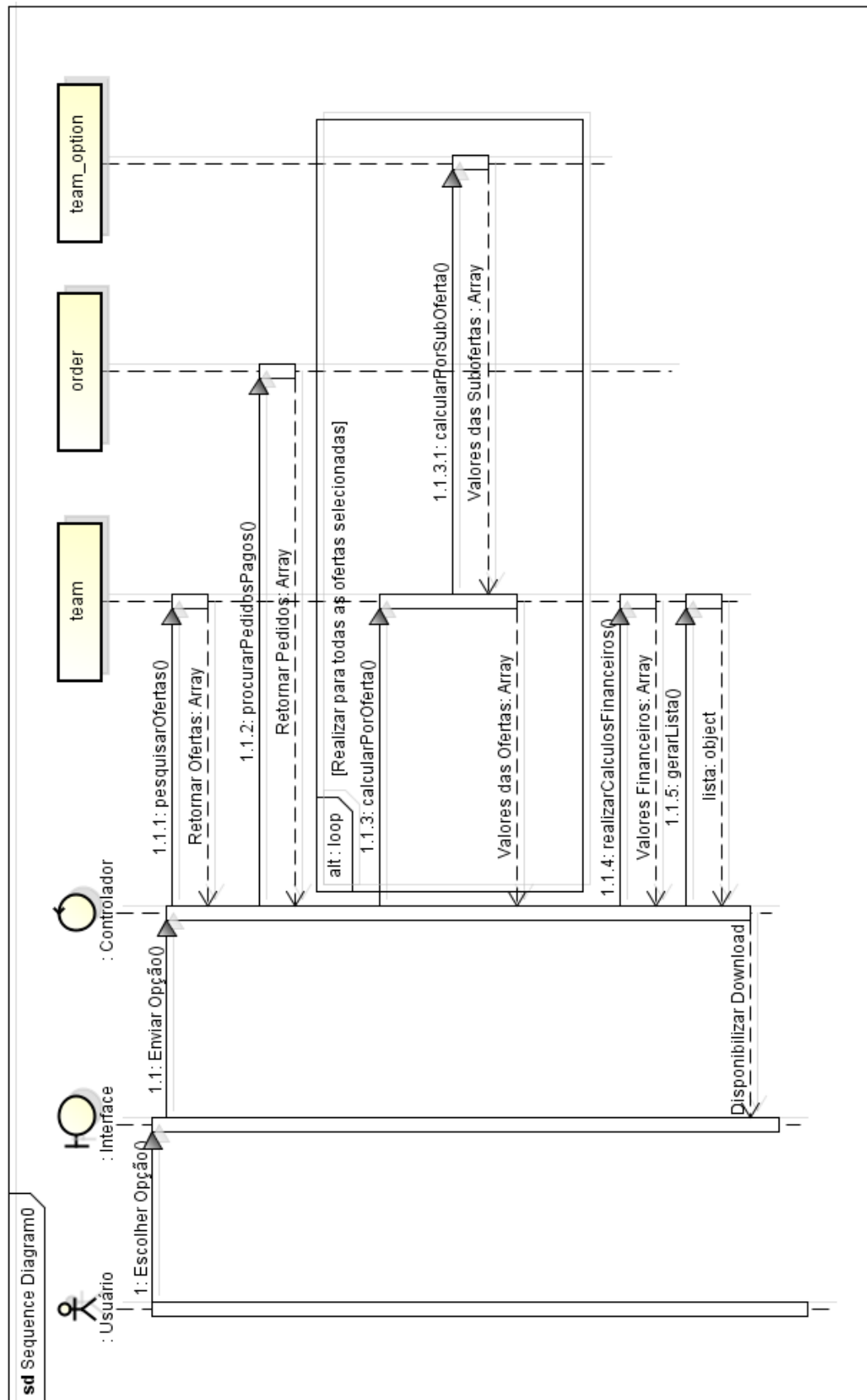
6. DIAGRAMA DE CASOS DE USO DO SISTEMA DE LISTA FINANCEIRA



7. DIAGRAMA DE CLASSES DO SISTEMA DE LISTA FINANCEIRA



8. DIAGRAMA DE SEQUÊNCIA DO SISTEMA DE LISTA FINANCEIRA



9. DIAGRAMA DE ATIVIDADES DO SISTEMA DE LISTA FINANCEIRA

