Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"

MATHEUS DOS SANTOS SOUZA

APLICATIVO PARA CONTROLE FINANCEIRO DE USO PESSOAL

Assis/SP
2016

Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"

MATHEUS DOS SANTOS SOUZA

APLICATIVO PARA CONTROLE FINANCEIRO DE USO PESSOAL

Projeto de pesquisa apresentado ao curso de Ciências da Computação do Instituto Municipal de Ensino Superior de Assis – IMESA e a Fundação Educacional do Município de Assis – FEMA, como requisito final à obtenção do Certificado de Conclusão.

Orientando(a): Matheus Santos Souza
Orientador (a): Prof Dr Luiz Ricardo Begosso

Assis/SP
2016

FICHA CATALOGRÁFICA

SOUZA, Matheus do Santos.
Aplicativo para controle financeiro de uso pessoal / Matheus dos Santos Souza. Fundação Educacional do Município de Assis –FEMA – Assis, 2016.
Número de páginas 60p.

1. Google Android. 2. Palavra-chave.

CDD: 001.6
Biblioteca da FEMA

APLICATIVO PARA CONTROLE FINANCEIRO DE USO PESSOAL:

MATHEUS DOS SANTOS SOUZA

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do Curso de Graduação, avaliado pela seguinte comissão examinadora:

Orientador: _____
Prof Dr Luiz Ricardo Begosso

Examinador: _____
Prof Fábio Eder Cardoso

Assis/SP
2016

Dedicatória

Dedico este trabalho primeiramente a Deus
por mais esse sonho realizado, a minha família
que me apoiaram nas horas difíceis, aos meus amigos
e a todos os professores pelo ensinamento durante esses anos.

Agradecimentos

Agradeço primeiramente a Deus, por ter iluminado meu caminho durante toda essa caminhada.

Agradeço a minha mãe Fatima, por sempre estar ao meu lado me dando forças e me apoiando nas horas difíceis, sempre me incentivando para que nunca desistisse.

Agradeço ao meu pai José por acreditar em mim durante essa caminhada, sempre estando ao meu lado e me incentivando a nunca desistir.

Agradeço a minha esposa Luana, que de forma especial e carinhosa me deu força todos esses anos, sempre estando ao meu lado me apoiando em todas as horas difíceis, me incentivando de todas as formas para que não desistisse dessa caminhada.

Agradeço aos meus filhos Izadora e Pedro, que embora não tenham conhecimento disto, mas iluminaram meus pensamentos de maneira que eu corresse atrás de adquirir novos conhecimentos.

Agradeço aos meus avós que sempre estiveram ao meu lado, que me apoiaram durante essa trajetória e incentivando a batalhar pelos meus sonhos, nunca deixando eu desistir daquilo que eu sonhava.

Agradeço também a todos os professores que me acompanharam durante a graduação e em especial ao meu orientador e Profº Begosso, responsável pela orientação e fornecendo conhecimento para que ocorresse o desenvolvimento deste projeto.

Aos meus amigos pelo apoio e incentivo, em especial Cláudio e Danilo que durante essa caminhada, sempre me ajudaram e de uma certa forma me deram “puxões de orelhas”.

RESUMO

O projeto será baseado na linguagem de programação Java para o projeto será realizado aplicativo de modelagem e desenvolvimento de uma aplicação baseada na plataforma Google Android .

Será utilizado como um desenvolvimento environmen tAndroid Studio e base de dados de armazenamento de dados SQLite .

O projeto tem como objetivo auxiliar na gestão de contas pessoais proporcionando ao usuário uma interface simples e fácil de manusear. O aplicativo separa seus gastos fornecendo relatórios detalhados de todas as despesas, podendo filtrar os relatórios escolhendo categoria, período e despesa total.

Palavras-chave: Google *Android*, Aplicativo, Gerenciador de Contas

ABSTRACT

The project will be based on the Java programming language for the project will be carried out modeling application and development of an application based on the Google Android platform.

It will be used as a environmentAndroid Studio development and SQLite data storage database.

The project is aimed at assisting in the management of personal accounts providing the user with a simple and easy to handle interface. The application separates your spending providing detailed reports of all expenses ,can filter the reports choosing category , period and total expenditure .

Keywords: Google *Android*, Application, Accounts Manager.

LISTA DE ILUSTRAÇÕES

Figura 1 Sequenciamento das atividades visando o desenvolvimento do projeto 15

Figura 2 Divisão da plataforma Google *Android* em três partes. 17

Figura 3 Subdivisão das camadas da plataforma Google *Android* 18

Figura 4 Diagrama de classes. Fonte: http://www.profissionaisti.com.br/2011/07/os-principais-diagramas-da-uml-resumo-rapido/. 24

Figura 5 Diagrama de objetos. Fontes: http://www.profissionaisti.com.br/2011/07/os-principais-diagramas-da-uml-resumo-rapido/. 25

Figura 6 Diagrama de componentes. Fonte: http://www.profissionaisti.com.br/2011/07/os-principais-diagramas-da-uml-resumo-rapido/. 25

Figura 7 Diagrama de estrutura composta. Fonte: http://docplayer.com.br/133677-Guia-pratico-2a-edicao-gilleanes-t-a-guedes-novatec.html. 26

Figura 8 Diagrama de pacote. Fonte: http://www.profissionaisti.com.br/2011/07/os-principais-diagramas-da-uml-resumo-rapido/. 26

Figura 9 Diagrama de implantação. 27

Figura 10 Diagrama casos de uso. Fonte: http://www.devmedia.com.br/o-que-e-uml-e-diagramas-de-caso-de-uso-introducao-pratica-a-uml/23408. 28

Figura 11 Diagrama de atividade. Fonte: http://www.profissionaisti.com.br/2011/07/os-principais-diagramas-da-uml-resumo-rapido/. 29

Figura 12 Diagrama de estado. Fonte: http://www.profissionaisti.com.br/2011/07/os-principais-diagramas-da-uml-resumo-rapido/. 30

Figura 13 Diagrama de sequência. Fonte: http://www.profissionaisti.com.br/2011/07/os-principais-diagramas-da-uml-resumo-rapido/. 31

Figura 14 Diagrama de visão geral. Fonte:http://gaia3.uel.br/projetos/oa_uml/Vis%C3%A3o%20Geral%20dos%20Diagramas%20da%20UML.htm. 32

Figura 15 Diagrama de tempo. Fonte: http://docplayer.com.br/133677-Guia-pratico-2a-edicao-gilleanes-t-a-guedes-novatec.html. 33

Figura 16 Diagrama de colaboração. Fonte: http://www.profissionaisti.com.br/2011/07/os-principais-diagramas-da-uml-resumo-rapido/. 34
Figura 17 Diagrama de Casos de Uso referente ao projeto.....35
Figura 18 Caso de uso da interação do usuário com cadastrar salario 36
Figura 19 Caso de uso usuário interagindo com Cadastrar Usuário. 37
Figura 20 Caso de uso usuário interagindo com Manter Contas. 38
Figura 21 Caso de Uso usuário interagindo com Manter Categoria. 39
Figura 22 Caso de Uso usuário interagindo com Manter Tipo de Conta. 40
Figura 23 Caso de Uso usuário interagindo com Relatório Gasto por Categoria 41
Figura 24 Caso de Uso usuário interagindo com Relatório Gasto Total 42
Figura 25 Caso de Uso usuário interagindo com Relatório gasto por período.....43
Figura 26 Casos de uso Manter cartão de crédito. 44
Figura 27 Caso de uso manter cheque. 45
Figura 28 Caso de uso manter compras parceladas. 46
Figura 29 Caso de uso manter economia. 47
Figura 30 Caso de uso visualizar cheques. 48
Figura 31 Caso de uso manter compras parceladas 48
Figura 32 Caso de uso visualizar contas pagas 49
Figura 33 Caso de uso visualizar contas atrasadas. 49
Figura 34 Caso de uso login. 50
Figura 35 Caso de uso autenticar 50
Figura 36 Caso de uso visualizar economia. 51
Figura 37 Caso de uso visualizar saldo disponível 51
Figura 38 WBS - WORK BREAKDOWN STRUTURE.....52
Figura 39 Diagrama entidade relacionamento. 53
Figura 40 Diagrama de classe. 54

LISTA DE TABELAS

Tabela 1 Versões *Android*. 20

Tabela 2 Descrição do utilizadores do *SQLite* e suas utilização.....22

Tabela 3 Cronograma do projeto. 56

SUMÁRIO

1. INTRODUÇÃO 12

1.1. JUSTIFICATIVA 13

1.2. METODOLOGIA 14

2. AMBIENTE DE DESENVOLVIMENTO 16

2.1. PLATAFORMA GOOGLE *ANDROID* 16

2.1.1. Historia 16

2.1.2. Arquitetura 17

2.1.3. Versões 19

2.2. *SQLITE* 21

3. UML 23

3.1. HISTÓRIA 23

3.2. ESTRUTURA 24

3.2.1. Descrição dos diagramas de estruturas 24

3.2.2. Descrição dos diagramas de comportamentos 28

3.2.3. Descrição dos Diagramas de Interação: 31

4. MODELAGEM 35

4.1. CASOS DE USO 35

4.1.1. Especificações Casos de Uso 36

4.2. WBS - WORK BREAKDOWN STRUTURE 52

4.3. DIAGRAMA ENTIDADE RELACIONAMENTO 53

4.4. DIAGRAMA DE CLASSE 54

5. CRONOGRAMA 55

6. CONCLUSÃO 57

REFERÊNCIAS 58

1. INTRODUÇÃO

Com o agravamento da crise econômica no Brasil, que vem causando uma certa preocupação na população Brasileira, a volta da inflação está sendo crucial para a economia, pois com ela os produtos encarecem, deixando cada vez mais desvalorizado o poder aquisitivo dos trabalhadores. A desvalorização do Real no mercado mundial é um fator inevitável, com isso observa-se que as moedas internacionais disparam no território Brasileiro. Empresários reduzem seus investimentos e começam a cortar gastos em suas empresas, para conseguir passar essa crise e com isso a taxa de desemprego está em expansão.

Com a evolução da crise e a alta da inflação que vem ocorrendo nos últimos meses, é fato que uma parcela da população entre em dívidas, por querer manter aquele mesmo padrão de vida que tinha antes do aparecimento da crise. Muitas pessoas tentam controlar seus gastos, realizando apenas compras necessárias, retirando de sua lista artigos considerados supérfluos. Um dos principais problemas é como realizar esse controle, não gastar mais do que o orçamento e como fazer para comprometer apenas uma parcela do orçamento pessoal ou até mesmo familiar em tempo de crise.

No mercado de tecnologia da informação pode-se encontrar várias opções para controle de gastos, algumas podendo ser gratuitas e outras pagas, onde determinadas aplicações pagas fornecem algumas versões gratuitas com uma certa limitação para o usuário, com o intuito de vender suas aplicações privadas como é o caso do Mobile que contém versões gratuitas e privadas, podendo ser acessadas pelo computador, Tablet ou Smartphone. O Guia de Bolso é um aplicativo gratuito que realiza a sincronização automática de todas as despesas, realiza também a sincronização das transações bancarias do cliente (de uma conta ou mais), bastando apenas que o usuário forneça a sua senha do Internet banking, para que ocorra a sincronização dos dados bancários com o aplicativo. O Guia de Bolso possui versões para iOS, *Android* e também para navegadores (RIBEIRO, 2015). O Moni é aplicativo gratuito também e que exige que o usuário realize os cadastros manualmente de todas as suas receitas e despesas, possui uma das mais simples e rápidas formas de cadastrar, disponível apenas para iOS (RIBEIRO, 2015).

Segundo a pesquisa *Mobile Report*, da Nielsen IBOPE, no primeiro trimestre de 2015 o total de pessoas que utiliza Internet pelo smartphone no Brasil chegou a marca de 68,4 milhões

de pessoas, por meio desses números é evidente que smartphones já são uma realidade na vida do Brasileiro, desde a classe baixa até a classe alta(NIELSEN.COM, 2015).

Com o uso intenso dos smartphones na vida dos Brasileiros, o gerenciador de contas pessoais para mobile, tem como finalidade realizar o gerenciamento de suas contas de forma simples e fácil, sem fugir da rotina, gerenciando suas contas através de cliques em seu smartphone.

Este projeto tem como finalidade o gerenciamento de gastos pessoal e familiares, auxiliando-os a controlar seus gastos, gerenciando seu salário de forma que haja uma divisão de seus gastos em categorias, controlando seu gasto desde os gastos essenciais que temos até aos gastos com estilo de vida, adotando um relatório de interface atrativa e fácil entendimento do usuário. O aplicativo pode ser uma excelente ferramenta durante essa crise financeira.

Este trabalho tem como objetivo realizar o desenvolvimento de um aplicativo mobile que realize o controle financeiro pessoal e familiar, com intuito de auxiliar no controle dos gastos mensal.

O projeto será desenvolvido utilizando como base a linguagem de programação *Java,* realizando o desenvolvimento de um aplicativo baseado na plataforma *Google Android*, o ambiente de desenvolvimento utilizado é *Android Studio* e para a estruturação do Banco de Dados será utiliza o *SQLite*.

1.1. JUSTIFICATIVA

A importância do desenvolvimento desse projeto se dá em relação à crise financeira que o país está vivendo atualmente e em relação a dificuldade que uma parte da população tem em gerenciar seus gastos, muitas pessoas tem dificuldade em realizar anotações de suas contas, chegando a um descontrole de suas finanças e com isso acaba acontecendo um acúmulo de suas contas.

O estudo ’Retratos da Sociedade Brasileira’ mostra que 57% da população precisou mudar seus hábitos de consumo ou planejamento financeiro, realizando corte de gastos desnecessários e os outro 21% pretendem mudar também, fazendo com que famílias mudem de residências, troque seus filhos de escolas privadas para públicas, muitas

famílias estão se aprofundando em dividas para cobrirem os gastos da casa, 37% afirmam ter adquirido dividas para cobrirem suas despesas (BRAZILIENSE, 2015).

Em abril as famílias Brasileiras comprometeram mais de 46% da renda anual com dividas em bancos, financeiras e com cartão(G1.COM, 2015). Muitos Brasileiros têm buscado trabalhos extras para conseguir cobrir todas suas despesas, pessoas que não trabalhavam estão voltando a buscar empregos para ajudar a cobrir gastos familiares (BRAZILIENSE, 2015). No primeiro trimestre de 2015, 68 milhões de pessoas utiliza Internet pelo Smartphone no Brasil, como podemos ver os Smartphone hoje em dia está uma febre e esse número tende a expandir mais (NIELSEN.COM, 2015).

As pessoas endividadas, tem dificuldade no relacionamento pessoal, familiar e profissional, podendo influenciar diretamente em nossas vidas. As pessoas que buscam alcançar determinado objetivo, realizam um planejamento financeiro pessoal, tendo suas finanças presente diariamente nas vidas delas, gerenciando seus gastos para que consiga realizar aquele desejado objetivo.

Um aplicativo que gerencie suas contas de forma fácil, simples e organizada pode facilitar muito a vida das famílias. Auxiliando-as comprometer apenas uma parcela de sua renda de forma segura, mostrando os gastos cadastrados daquele referente mês, saldo disponível, saldo gasto. Podendo ajudar as famílias se endividar menos e sempre ter um controle dos gastos mensal e ter um dinheiro para necessidades.

1.2. METODOLOGIA

O projeto começa a ser desenvolvido após um estudo sobre o assunto, avaliando quais são os requisitos essenciais para elaboração do projeto e como deverá ser desenvolvido, de forma que auxilie o entendimento dos usuários de forma simples e prática.

Após o levantamento dos requisitos será desenvolvida toda documentação do mesmo, em seguida, será realizada a validação dos requisitos, verificando se os requisitos levantados se enquadram no objetivo do projeto a ser desenvolvido. Ao término dos requisitos, serão criados a lista de eventos e o diagrama de Casos de Uso, seguidos de sua especificação.

Para que possa dar procedimento ao projeto será necessário realizar um estudo detalhado sobre a plataforma Google *Android*, tendo como base a linguagem de programação Java e sobre as ferramentas *Android* Studio e banco de dados *SQLite* visando o desenvolvimento

deste projeto, tendo como principal objetivo adquirir um certo conhecimento sobre o funcionamento e as funcionalidades das determinadas ferramentas.

Agora o projeto passa a parte de implementação onde realizaremos a prototipação das interfaces de telas, criação do diagrama entidade e relacionamento, criação do banco de dados, programação de cadastros, programação das movimentações, programação do relatórios e teste de unidades para finalizar será realizado o teste geral da integração.

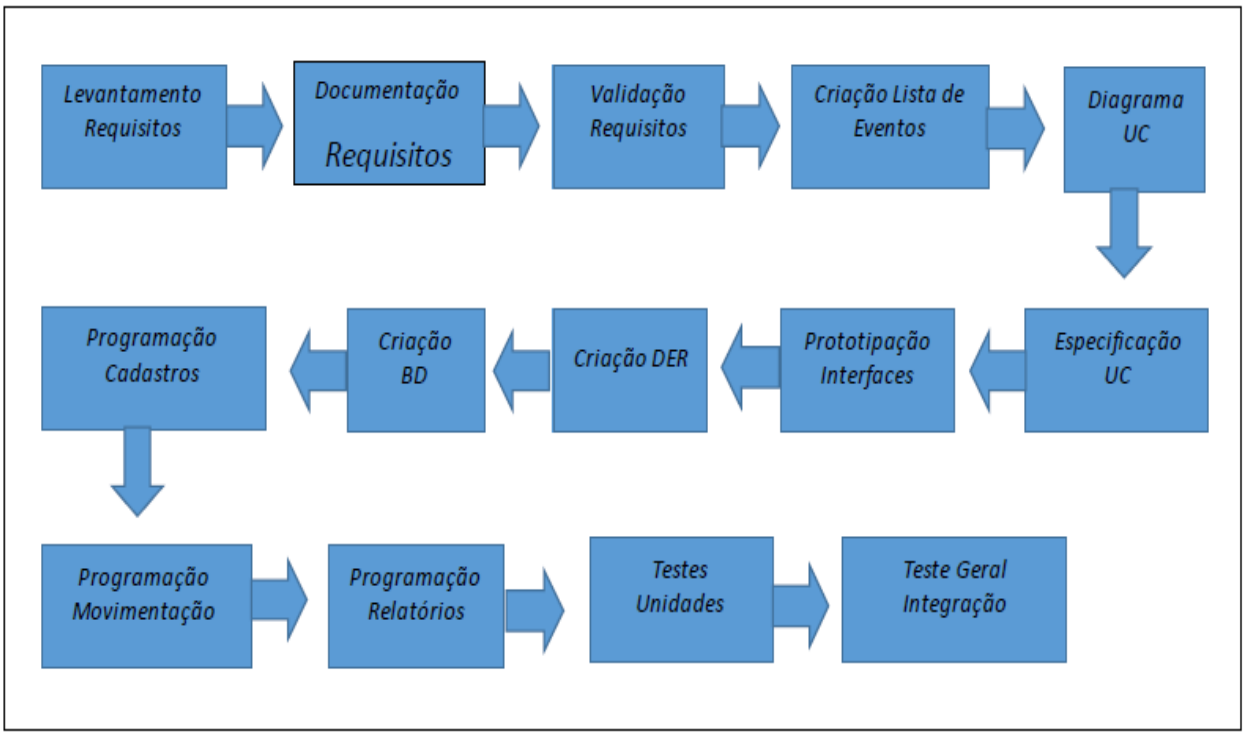

Figura 1 Sequenciamento das atividades visando o desenvolvimento do projeto. Fonte: Didática da aula de Engenharia de Software I

Este trabalho está estruturado da seguinte forma: o Capitulo 1 apresenta a introdução, justificativa e metodologia de realização. O Capitulo 2 apresenta a plataforma *GOOGLE ANDROID e o banco de dados SQLite, sobre o Android* será falado sobre sua historia, arquitetura e versões e para completar o capitulo apresenta um resumo sobre o banco de dados *SQLite.* O Capitulo 3 apresenta um estudo sobre a UML, falando um pouco de sua historia, estrutura descrevendo os diagramas. O Capitulo 4 apresenta a modelagem do projeto, diagramas de casos de uso, especificações dos casos de usos, *Work BreakDown Struture,* diagrama de entidade relacionamento e diagrama de classe. O Capitulo 5 apresenta o cronograma do projeto e o Capitulo 6 a conclusão do projeto.

2. AMBIENTE DE DESENVOLVIMENTO

2.1. PLATAFORMA GOOGLE *ANDROID*

2.1.1. Historia

O *Android* teve seu início em 2003 na cidade de Palo Alto na Califórnia, onde Andy Rubin, Rich Miner, Nick Sears e Cris White decidiram fundar a empresa *Android* Inc, que tinha como princípios desenvolver um inovador sistema para câmeras digitais. Por questão de o mercado não ser muito favorável, então decidiram migrar para um sistema mobile criando um sistema baseado no Kernel do Linux, com isso começaram a desenvolver softwares para celulares, seus projetos eram desenvolvidos de forma secreta. Em 2005 o Google adquiriu a empresa *Android* Inc, e decidiu entrar no mercado mobile, iniciando o desenvolvimento do que seria o novo *Android*, sendo baseado em Java e executado no Kernel 2.6 do Linux (MEYER, 2015).

Em novembro de 2007 foi criado o grupo Open Handset Alliance (OHA), que conta com gigantes de telefonia móvel e tecnologia lideradas pela Google. Entre os membros do grupo existem nomes consagrados como: LG, Asus, Dell, HTC, Samsung, Sony, Motorola entre outros, hoje esse grupo conta com 87 empresas que tem como intuito mudar a experiência móvel para os consumidores (MEYER, 2015).

O primeiro aparelho equipado com o *Android* foi o HTC G1, também conhecido como HTC DREAM, sendo vendido no mercado mundial. O aparelho era equipado com tela de 3,2 polegadas, processador de 528 MHz e 192 MB de memória RAM, suficiente para que pudesse executar o sistema operacional *Android* 1.0 e sua sucessivas versões até a versão 1.6. Seu lançamento oficial ocorreu em 22 de Outubro de 2008, com isso dando início ao remodelamento de todo o conceito de smartphones (MEYER, 2015).

Desde então o *Android*, passou a ser o sistema operacional de telefonia mais utilizado em todo o mundo, utilizando-o desde aparelhos mais simples até ao tops de linhas como: Samsung Galaxy S7 Edge, Lg G5, Samsung Galaxy S7, Samsung Galaxy Note 5, Motorola Moto X Force, Samsung Galaxy S6 Edge +, entre outras opções encontradas no mercado mundial.

Em 2009 os aparelhos equipados com o sistema operacional *Android* representava apenas 2,8% do aparelhos vendido em todo o mundo, já no próximo ano detinha 33% do mercado mundial, em 2011 já aderia a marca de 52,5%, obtendo um crescimento considerado nos próximos anos também, chegando em 2014 com uma marca de 81,5% do mercado mundial de smartphone, entre 2010 e 2014 já foram vendidos mais de 3 bilhões de smartphones com o sistema operacional *Google Android* e para os próximos anos esses números tendem a expandir mais.

Atualmente o *Android* não está disponível apenas em smartphones e Tablets, ele está presente também em netbooks, smartbooks, câmeras digitais, tocadores de mp3, Smart tvs, relógios inteligente, IoT e está chegando ao mercado automobilístico onde as montadoras Audi, General Motors, Hyundai e Honda se uniram para realizar o desenvolvimento do primeiro sistema operacional automobilístico.

2.1.2. Arquitetura

A arquitetura do sistema operacional *Android* é baseada na linguagem Java, sendo executado no Kernel 2.6 do Linux. A plataforma *Android* é dividida em três partes: Sistema Operacional, camada middleware e camada de aplicação, sucessivamente essas camadas são subdivididas em 5 camadas.

Figura 2 Divisão da plataforma Google Android em três partes.

A seguir uma breve descrição das camadas citadas na Figura 2:

Camada de Aplicação: Essa camada se divide em camada de aplicação e de suporte, nela se encontra todos os aplicativos nativos do sistema operacional e aplicações de terceiros.

Camada de Middleware: Essa camada é composta por framework da aplicação (aplication framework), conjunto de bibliotecas e *Android* Runtime.

- Framework da aplicação detém componente pré-instalados, permite a utilização de novas estruturas para futuras aplicações e prioriza a reutilização de códigos.
- O conjunto de bibliotecas é composto por bibliotecas nativas do *Android*, sendo todas escritas na linguagem c/c++, são divididas em quatros categorias: *bionic libc,* biblioteca de funções, bibliotecas de servidores nativos e bibliotecas de abstração de hardware.
- No *Android Runtime* é encontrado as principais bibliotecas Java e contém também a máquina virtual Dalvik (DVM). Os arquivos são escritos em Java e compilados como .class., o sistema *Android* irá converter esse arquivo para .dex para que seja executado pelo DVM.

Camada do Sistema Operacional: É responsável por gerenciar toda a parte de threads, gerenciamento de memória, processos, seguranças de arquivos e pastas, essa camada é composta por um conjunto de drivers que formam o kernel do Linux, atua como uma camada abstrata entre o hardware e o software

A Figura 3 representa a camada da plataforma *Google Android*, dividindo a camada em 5 partes, representando como é formada cada camada e seus respectivos aplicativos

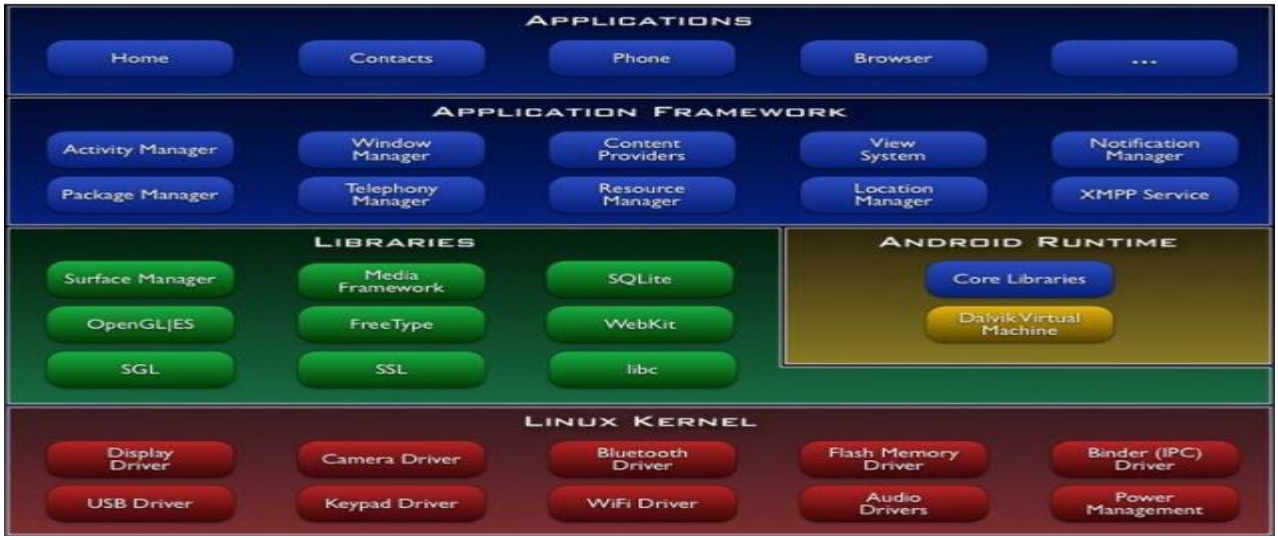

Figura 3 Subdivisão das camadas da plataforma Google *Android*. Fonte: developer.*Android*.com.

2.1.3. Versões

Atualmente o *Android* sofre várias atualizações, podendo variar desde a correção de simples bugs, incremento de novos aplicativos, troca de hardware e software, sua API só cresce quando ocorre mudanças mais significativas, podendo aderir até um novo nome.

Sua primeira versão surgiu em 2008, sendo implantada no celular HTC G1, sendo o primeiro celular a receber *Android* A primeira versão a receber um codinome foi a versão 1.5 denominada Cupcake ela surgiu em 2009, desde lá já foram lançadas mais 10 versões. Hoje se encontra na versão Marshmallow 6.0.

Abaixo um resumo das funcionalidades aderidas durante as atualizações das referidas versões.

Nome	Versão	Ano	Funcionalidade
Cupcake	1.5	2009	• Gravação de vídeos • Envio de fotos para Picasa • Envio de fotos para o Youtube
Donut	1.6	2009	• Galeria de fotos e vídeos melhorada • Busca por voz melhorada
Eclair	2.0 e 2.1	2009	• Novo aplicativo do google maps • Suporte para o HTML5 • Bluetooth 2.1
Froyo	2.2	2010	• Melhoria de *Perfomance* e estabilidade • Navegador recebeu *JavaScript 8* • Incremento da ferramenta tethering via usb e hospot

Gingerbread	2.3	2010	• Interface de usuário melhorada • Suporte a resolução de smartphones maiores • NFC
HoneyComb	3.0	2011	• Interface otimizada para Tablets • Videochat através da integração com o Google Talk
Ice Cream Sandwich	4.0	2011	• Desbloqueio por padrão facial • Otimização da Interface para smartphones e Tablets simultaneamente
Jelly Bean	4.1 até 4.3	2012	• Melhoria no desempenho • Novo design • Project Butter para enfrentar lags • Novas novidades • Introdução do Google Now
Kitkat	4.4	2013	• Melhoria de desempenho e segurança • Google Now Launcher • Integração das mensagens sms com o Hangouts • Mudanças sutis no design.
Lollipop	5.0	2014	• Suporte a dispositivos com arquitetura 64 bits • Suporte a diferentes tamanhos de telas • Mudanças visuais
Marshmallow	6.0	2015	• Mais segurança • Maior desempenho da bateria • Permissões de privacidade para apps

Tabela 1 Versões *Android*. Fonte: (VELOSO, 2015)

2.2. *SQLITE*

SQLite é um banco de dados desenvolvido em linguagem C, adotando como padrão o SQL integrado, diferentemente dos demais SQL o *SQLite* não tem um processo de servidor separado, realiza leitura e escrita para os arquivos de discos comuns(SQLITE.ORG, 2016).

Disponibiliza uma biblioteca compacta com todos recursos ativados, fornecendo uma biblioteca com tamanho inferior a 500KiB (SQLITE.ORG, 2016).

Necessita de pouca memória em tempo de execução, tornando-o popular para dispositivos com memórias restritas. Tem como característica realizar uso eficiente da memória, espaço em disco e banda de disco, altamente confiável, disponibiliza suporte para até dois terabytes de dados (SQLITE.ORG, 2016).

Fornece suporte para dados do tipo *TEXT, INTEGER* e *REAL*. Para que os demais tipos de dados possam ser armazenados, será necessário que realize uma conversão deles para alguns dos três tipos citados, pois o *SQLite* não realiza uma comparação, para saber se os dados enviados para armazenamento são iguais aos da coluna definida na tabela (SQLITE.ORG, 2016).

O *SQLite* é um banco de dados que fornece armazenamento local de dados para aplicações e dispositivos individuais, ao contrário dos demais banco de dados SQL cliente/servidor, que são projetados para viver dentro de datacenters(SQLITE.ORG, 2016).

Atualmente o *SQLite* está sendo muito utilizado em dispositivos embarcados, Internet das coisas, websites, análise de dados, cache de dados da empresa, banco de dados do lado do servidor, base de dados interna ou temporária, possibilitando outras diversas utilização (SQLITE.ORG, 2016).

É um ambiente que não precisa ser instalado e nem configurado antes de ser usado. Não necessita de administrador para criar instancia ou adicionar permissões de acesso. Os formatos de seus arquivos são multi-plataforma, podendo ser utilizados em uma máquina diferente com arquitetura diferente da máquina que ele se encontra, utiliza tipagem manifesto, oferece melhorias para a linguagem SQL (SQLITE.ORG, 2016).

Atualmente o *SQLite* está sendo utilizado por milhões de aplicações, se tornando o banco de dados mais implantado no mundo hoje. A seguir uma tabela mostrando alguns dos principais e mais conhecidos utilizadores do *SQLite* e aonde eles têm implantado o *SQLite* (SQLITE.ORG, 2016).

A Tabela 2 apresenta um breve levantamento das empresas que utilizam o banco de dados SQLite e suas respectivas implantações

Utilizadores	Implantações
Adobe	• Formato de arquivo para o Photoshop Lightroom. • Acrobat Reader • Em uma parte do Integrated Runtime Adobe.
Airbus	• Utilizado no software de voo para o A350 XWB.
Apple	• Dispositivos IOS • ITunes. • Aplicativos nativos do Mac desktop OS-X
Bosch	• Sistemas multimídia para veículos GM, Nissan e Suzuki.
Facebook	• Banco de dados em sua osquery.
Firefox	• Formato de armazenamento para o navegador Web Firefox. • Thunderbird, Leitor E-mail da Mozilla
Flame	• Chama, aplicativo espião.
Google	• *Android*. • Browser Chrome.
McAfee	• Utilizado nos programas antivírus.
Microsoft	• Utilizado como componente principal do Windows 10.
PHP	• Já vem com *SQLite*2 e *SQLite*3 embutido.
Python	• Já é utilizado pela linguagem desde sua versão 2.5
RPM	• RedHat Package Manager utiliza para controlar seu estado.

Tabela 2 Descrição do utilizadores do *SQLite* e suas utilizações.

O *SQLite* possui milhares de utilizadores não citados na tabela acima. Podendo ser implantado em diversos projeto

3. UML

UML é uma linguagem que tem como objetivo te auxiliar na especificação, visualização e documentação de sistemas orientado a objeto, de tal forma que atenda todos os requisitos necessários. Realizando a modelagem da aplicação, podendo ser de qualquer tipo e combinação de hardware, linguagem de programação, sistema operacional e aplicação que utilize qualquer middleware possibilitada por sua flexibilidade

(OMG.ORG, 2016).

Sua especificação consiste em duas partes: semântica que especifica a sintaxe abstrata, conceito da modelagem estática e a dinâmica dos objetos. Já a notação realiza a representação visual da semântica e especifica a notação gráfica do sistema. Detêm suporte para o desenvolvimento interativo e incremental realizando a divisão nos seguintes passos: Analise de requisitos, analise, projeto, implementação e testes (OMG.ORG, 2016).

3.1. HISTÓRIA

Seu surgimento ocorreu em meados da década de 90, com o surgimento de diversos métodos de análise e projeto no final da década de 80, todos esses métodos eram muito similares, entre eles três se destacaram: os métodos Booch desenvolvido pelo americano Grady Booch que se destacava na fase de projeto, OMT desenvolvido pelo também americano James Rumbaugh que se destacava na fase de análise de sistemas de informações e por último o OOSE criado pelo sueco Ivan Jacobson que se destacava nos casos de uso. Em 1995 durante o OOPLSA, ocorreu o lançamento do método unificado e Ivan Jacobson se integrou a equipe denominando-a de “ Três Amigos”, essa união trouxe o apoio da Rational Software que veio a financiar a junção dos três métodos. Com a junção deles em 1996 resultou na criação das versões 0.9 e 0.91 da UML. A partir disso várias empresas começaram a contribuir com o projeto, através dessas contribuições foi lançada a versão 1.0, era bem definida, expressiva, poderosa e aplicável. Em 1997 a UML foi adotada pela OMG como uma linguagem padrão de modelagem (ROCHA NETO, 2007).

3.2. ESTRUTURA

A Linguagem UML é composta por treze diagramas, onde são divididos em três grupos: diagramas de estruturas, diagramas de comportamento e diagrama de interações. Os diagramas têm como objetivo realizar a representação visual de um aspecto do sistema auxiliando no entendimento, colaboração e desenvolvimento de um software(OMG.ORG, 2016).

- **Diagramas de estruturas:** diagrama de classes, diagrama de objetos, diagrama de componentes, diagrama de estrutura composta, diagrama de pacote e diagrama de implantação.
- **Diagramas de comportamento:** Diagrama de casos de uso, diagrama de atividades e diagrama de estado.
- **Diagramas de Interação:** Diagrama sequência, diagrama de visão geral de interação, diagrama de tempo e diagrama de colaboração.

3.2.1. Descrição dos diagramas de estruturas

Diagrama de Classe: Mostra a estrutura estática das classes, representando as oque será manipulado pelo sistema, realiza a representação das classes mostrando seus atributos métodos e relacionamentos, considerado estático por sua estrutura representada ser válida durante qualquer parte do ciclo de vida do sistema(ETELG.COM.BR, 2016).

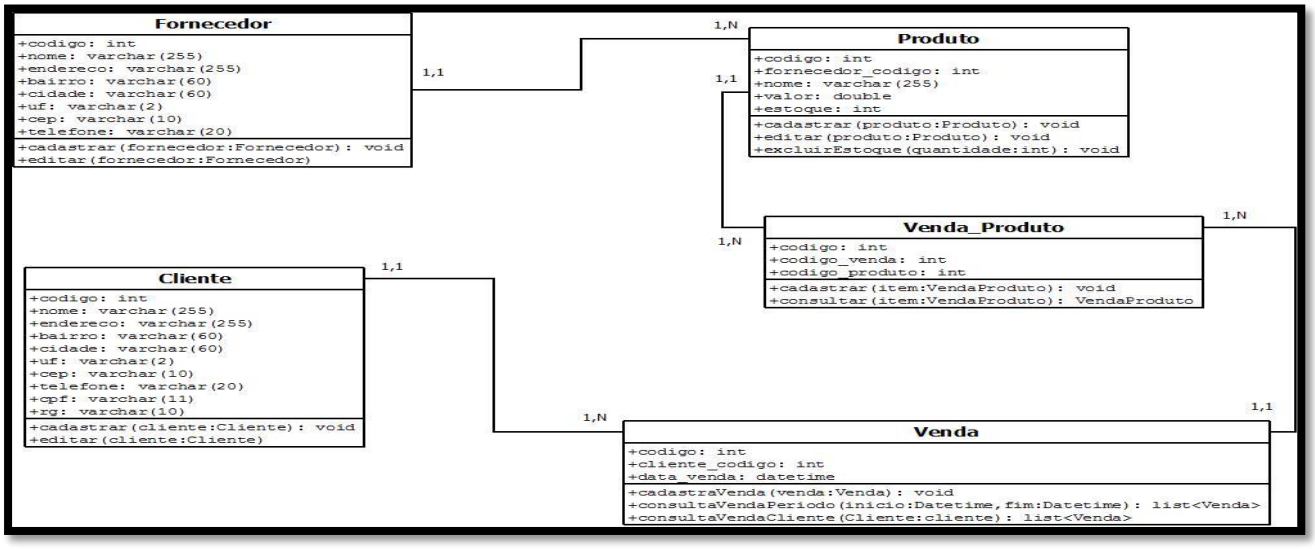

Figura 4 Diagrama de classes. Fonte: http://www.profissionaisti.com.br/2011/07/os-principais-diagramas-da-uml-resumo-rapido/.

Diagrama de Objetos: Utiliza quase a mesma anotação do diagrama de classes, mostrando os objetos estanciados da classe. Não é tão importante, mais por outros lados se tornam muito útil para ajudar na simplificação dos diagramas de classes. Sendo utilizados também nos diagramas de colaboração (ETELG.COM.BR, 2016).

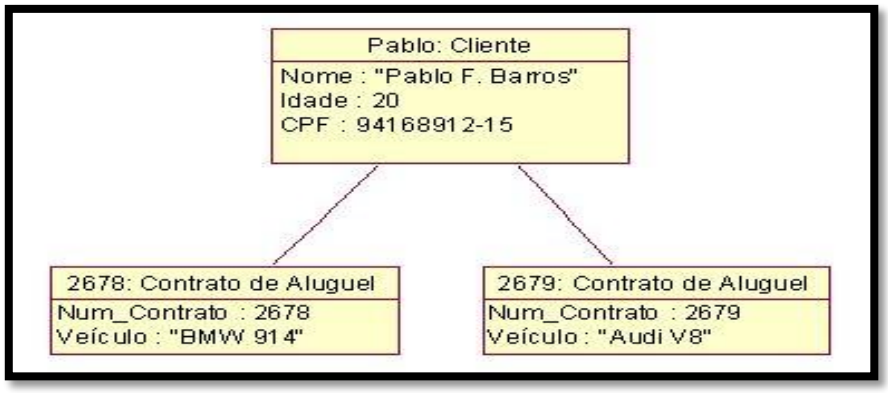

Figura 5 Diagrama de objetos. Fontes: http://www.profissionaisti.com.br/2011/07/os-principais-diagramas-da-uml-resumo-rapido/.

Diagrama de Componentes: O diagrama de componentes mostra o lado funcional do sistema, expondo as relações de seus componentes com os módulos. Realiza uma descrição dos componentes do software e suas dependências gerando uma representação da estrutura do código gerado. O diagrama é representado por uma elipse e 2 retângulos menores, do seu lado esquerdo, podendo ter seu nome escrito dentro ou abaixo do retângulo maior, sua dependência e representada por uma linha tracejada com uma seta na ponta, uma interface é representa com uma linha que parte de um componente e chega a um círculo em sua outra extremidade(ETELG.COM.BR, 2016).

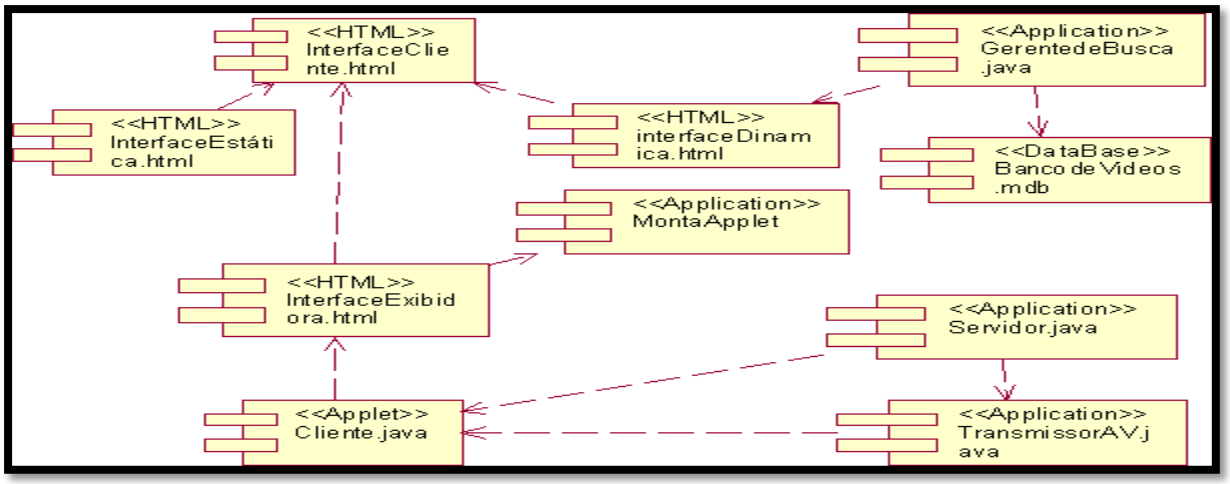

Figura 6 Diagrama de componentes. Fonte: http://www.profissionaisti.com.br/2011/07/os-principais-diagramas-da-uml-resumo-rapido/.

Diagrama de estrutura composta: Surgiu na segunda versão da UML realiza a definição da estrutura de um elemento focalizando nos detalhes, na construção e nos relacionamentos, foi introduzido a noção de “porto” onde pode ser associado interfaces e “colaboração” realizando a interligação dos portos para executar uma funcionalidade especifica (VARGAS, 2016).

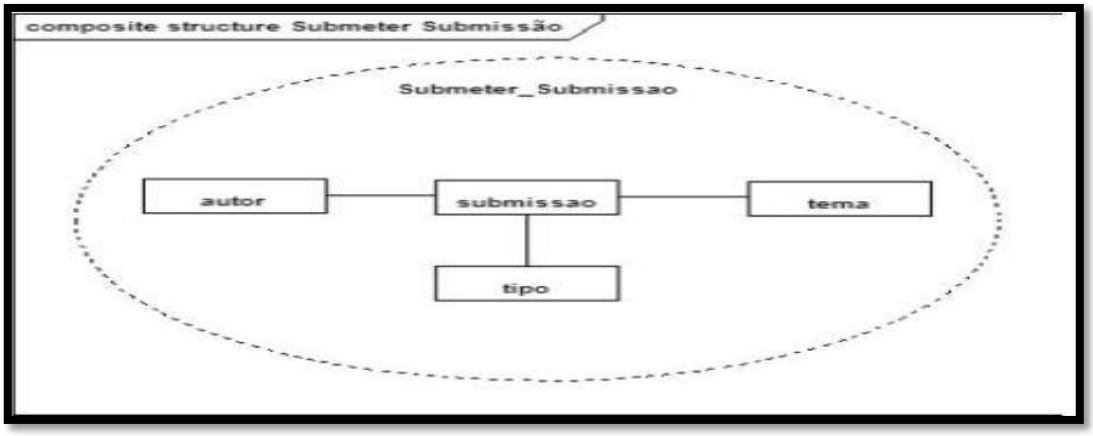

Figura 7 Diagrama de estrutura composta. Fonte: http://docplayer.com.br/133677-Guia-pratico-2a-edicao-gilleanes-t-a-guedes-novatec.html.

Diagrama de pacote: Foi introduzido na segunda versão do UML contendo pacotes e relacionamentos entre eles, tem como finalidade dividir a modelagem estrutural em divisões logicas e realizar a descrição entre eles (VARGAS, 2016).

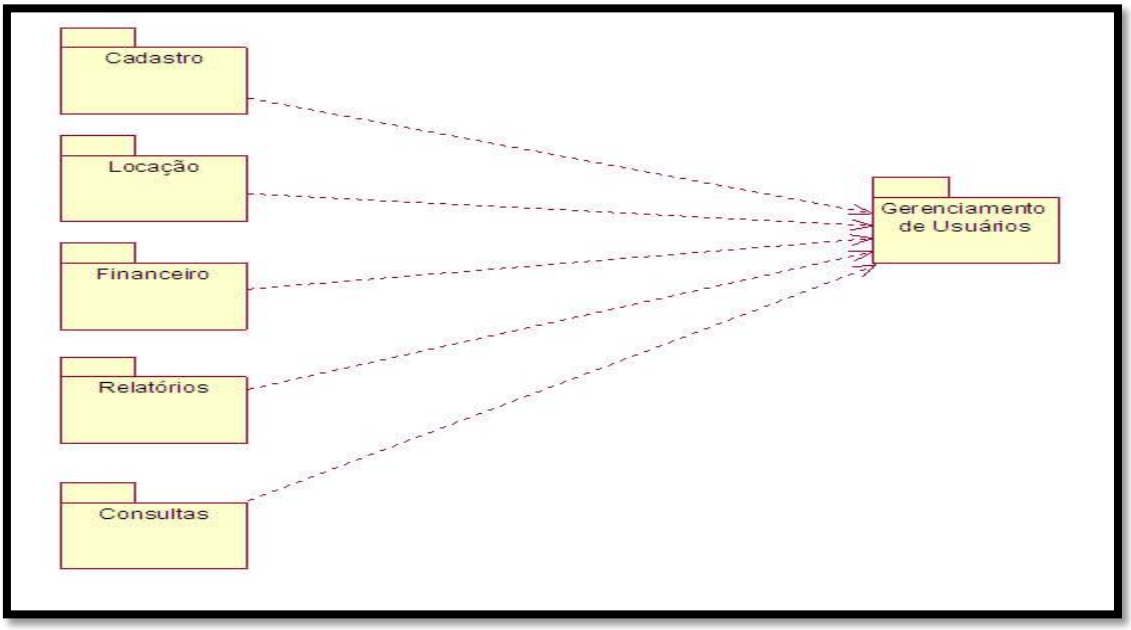

Figura 8 Diagrama de pacote. Fonte: http://www.profissionaisti.com.br/2011/07/os-principais-diagramas-da-uml-resumo-rapido/.

Diagrama de Implantação: Realiza a modelagem da parte física do sistema mostrando as configurações dos processadores, componentes físicos e de software que são utilizados no ambiente onde será implantado o sistema, tendo como principal elemento o “nodo” que tem como função realizar a representação de um recurso computacional. Esse diagrama é muito útil em projetos que tenham muitas dependências como impressoras, servidores, computadores entre outros (VARGAS, 2016).

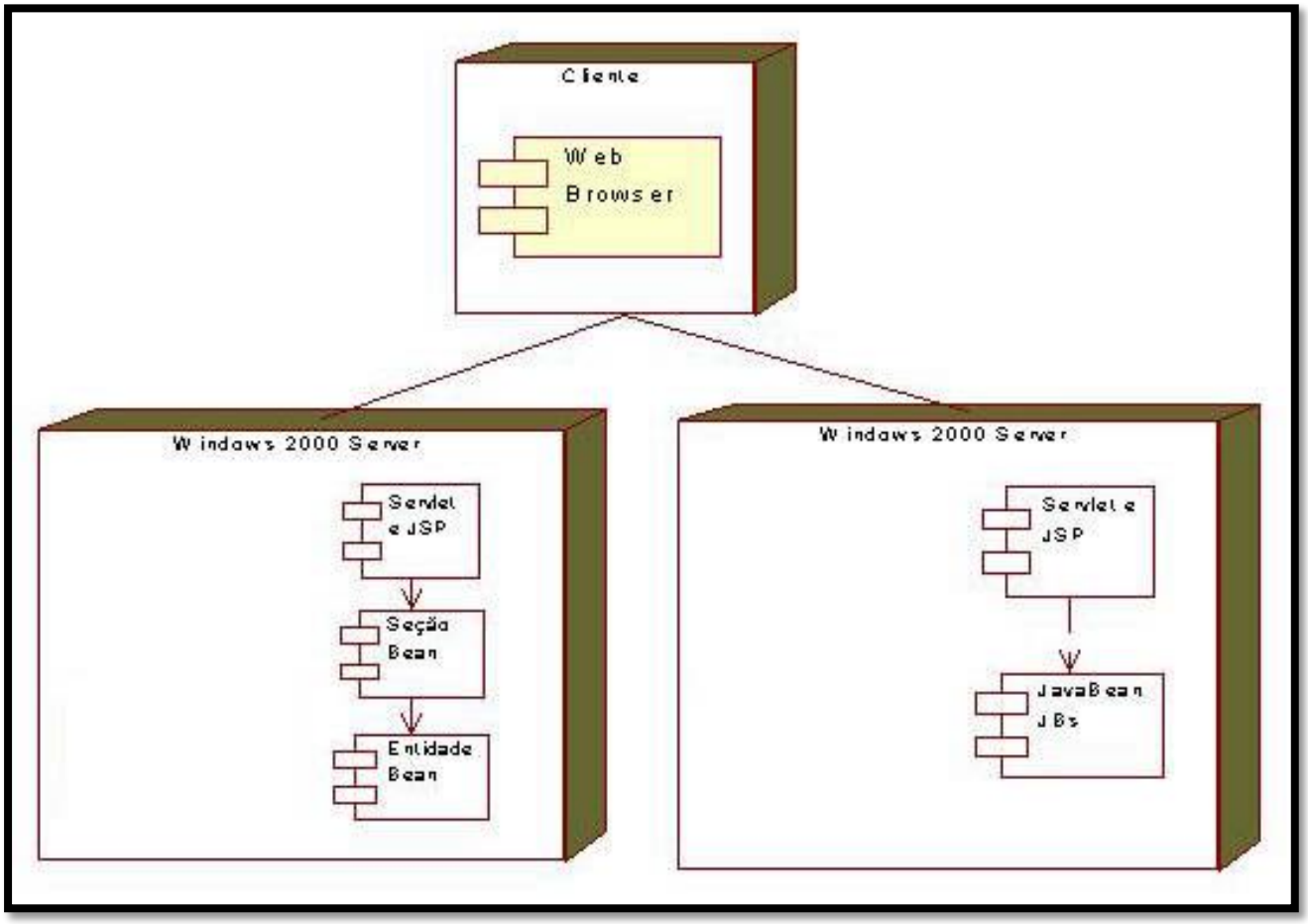

Figura 9 Diagrama de implantação. Fonte: http://homepages.dcc.ufmg.br/~amendes/GlossarioUML/glossario/conteudo/distribuicao/diagrama_de_distribuicao.html

3.2.2. Descrição dos diagramas de comportamentos.

Diagrama de casos de uso: O diagrama de casos de uso tem a finalidade de representar as funcionalidades do sistema aderindo o nome de “casos de uso”. A interação do sistema ocorre por meio do ator, podendo ser conectado com um ou mais casos de uso, devendo ter uma descrição mostrando seus relacionamento e interações, exemplificando como ocorre a interação para realizar determinada atividade (ETELG.COM.BR, 2016).

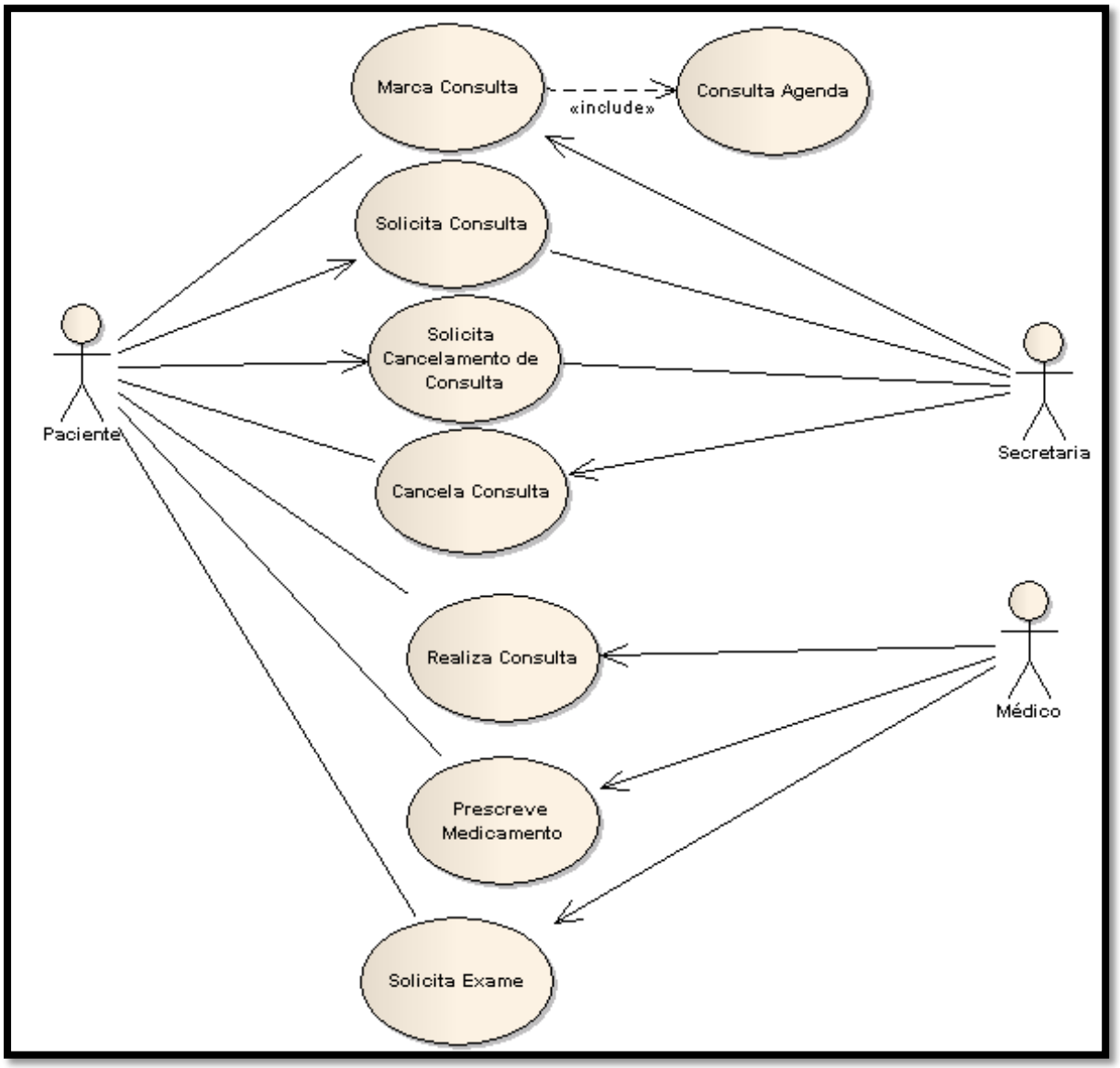

Figura 10 Diagrama casos de uso. Fonte: http://www.devmedia.com.br/o-que-e-uml-e-diagramas-de-caso-de-uso-introducao-pratica-a-uml/23408.

.

Diagrama de atividades: O diagrama de atividades tem o objetivo de representar a execução das ações e as transições necessária para conclusão de uma atividade. Diferentemente do diagrama de estado ele realiza a captura das ações durante a mudança de estado, mudam para um próximo estado quanto é executada uma ação, pode ser colocada com swimlanes.

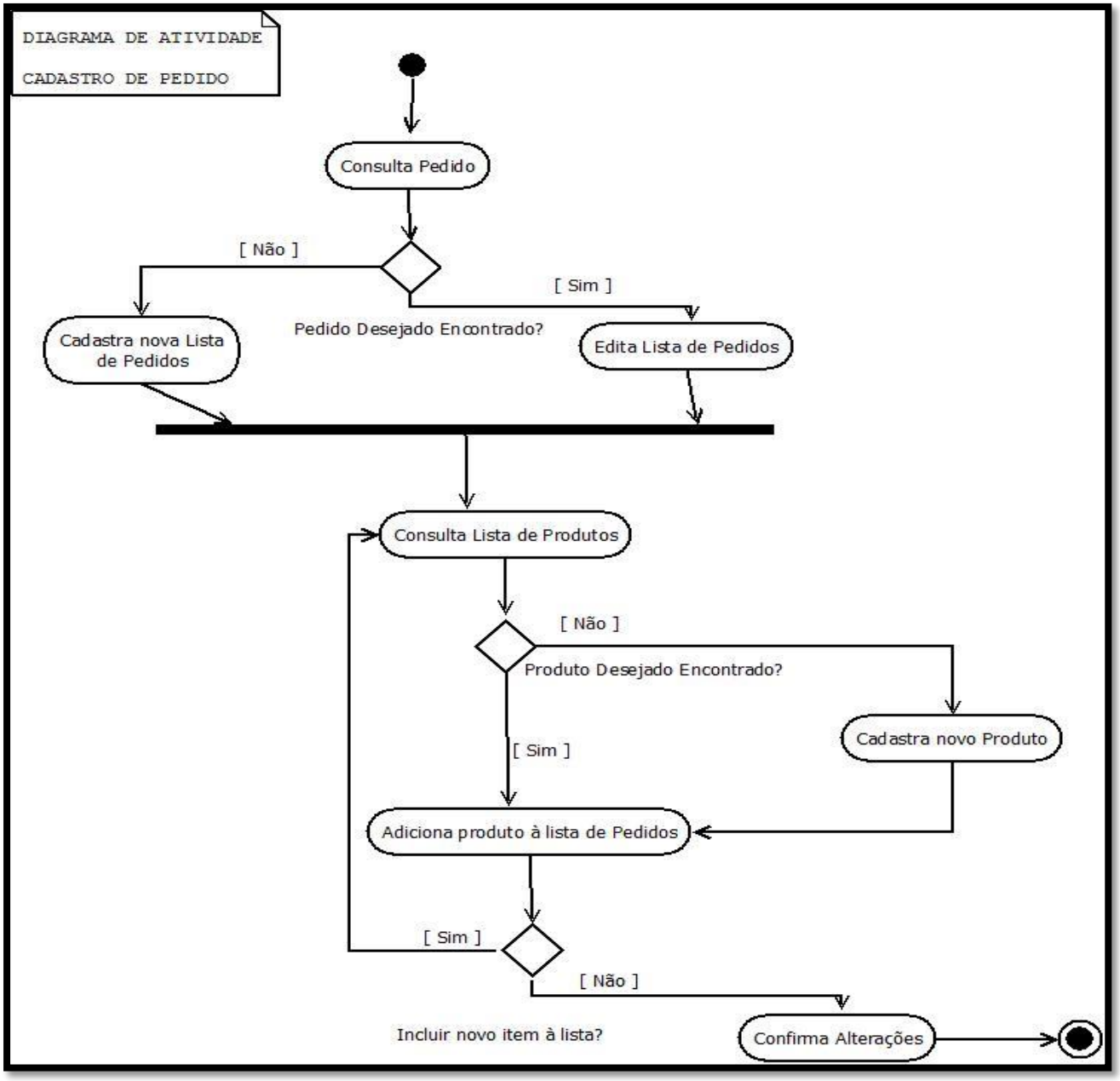

Figura 11 Diagrama de atividade. Fonte: http://www.profissionaisti.com.br/2011/07/os-principais-diagramas-da-uml-resumo-rapido/.

Diagrama de estado: É um complemento para a descrição da classe, mostrando todos os estados de objeto daquela classe e seus eventos responsável por aquela alteração, sendo escritos apenas para as classes que detêm um número definido de estado, sabendo aonde ocorrerá e qual será a mudança ocorrida naquele estado. O diagrama captura o ciclo de vida dos objetos, subsistema e sistema mostrando os estados que aquele determinado objeto pode possuir, e como os estados são afetados caso um evento passe do tempo. Possuem um estado inicial representado por um círculo totalmente preenchido e pode ter um ou mais estados finais sendo representados por dois círculos um dentro do outro.

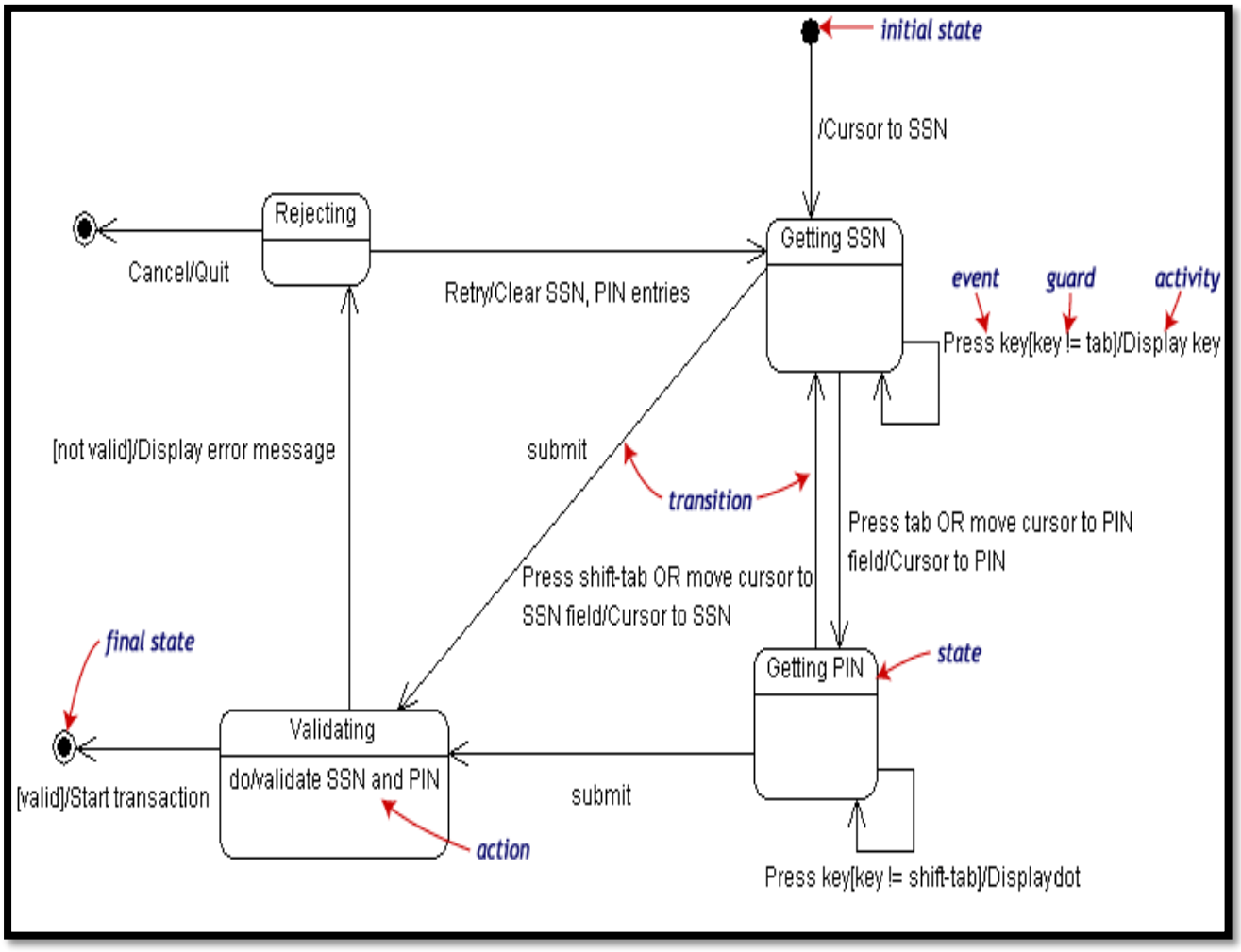

Figura 12 Diagrama de estado. Fonte: http://www.profissionaisti.com.br/2011/07/os-principais-diagramas-da-uml-resumo-rapido/.

3.2.3. Descrição dos Diagramas de Interação:

Diagrama sequência: O diagrama de sequência realizar a divisão do processo representando a troca de mensagens entre os objetos, mostrando a interação entre eles. O diagrama possui dois eixos, o vertical mostra o tempo e o horizontal mostra os objetos que são representados por um retângulo e um linha pontilhada na vertical denominada linha de vida, o tempo é observado no sentido vertical de cima para baixo, as mensagens são representadas por setas entres os objetos que tenham relacionamentos, podendo também ser numeradas com objetivo de exemplificar mais a sequência do diagrama (ETELG.COM.BR, 2016).

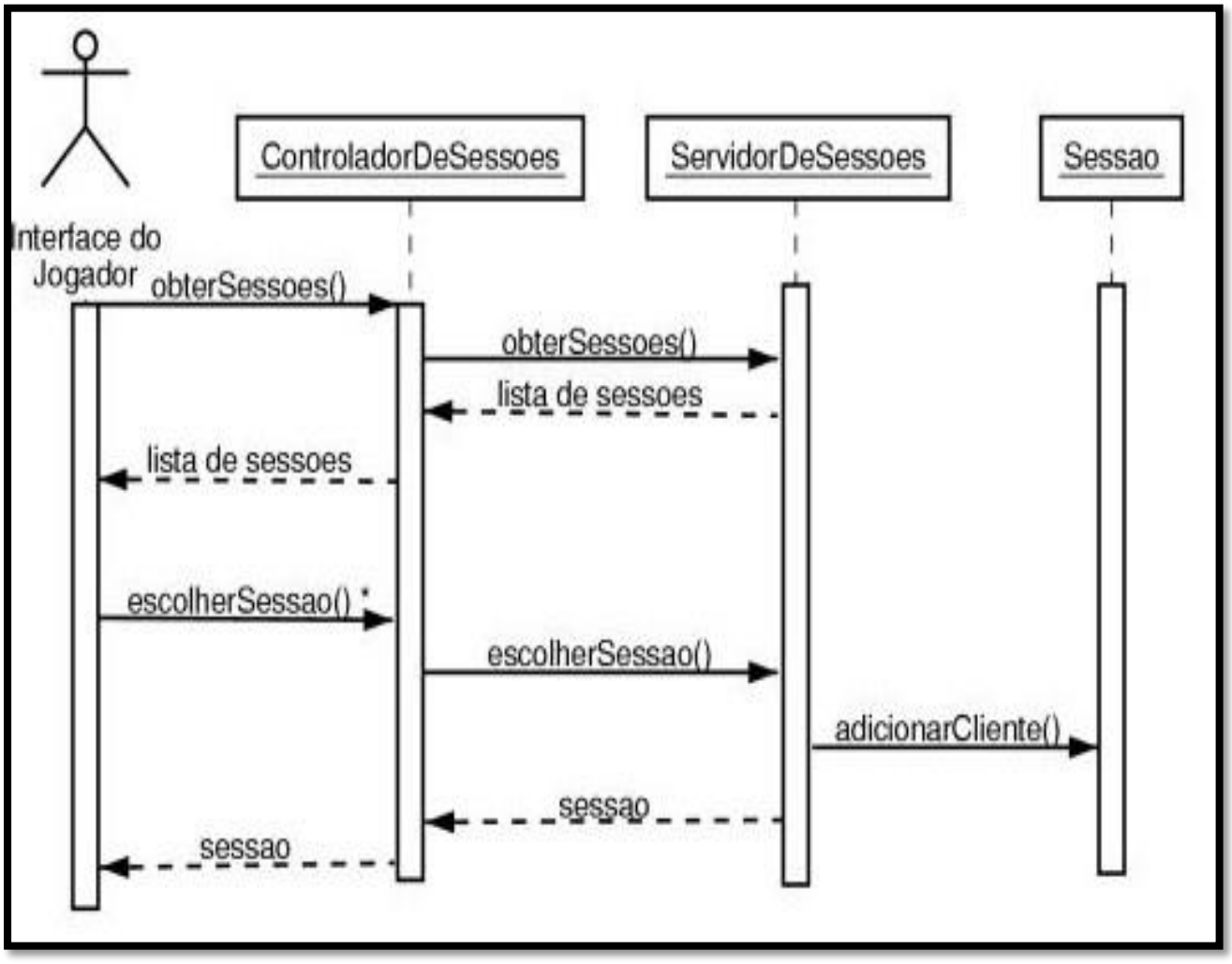

Figura 13 Diagrama de sequência. Fonte: http://www.profissionaisti.com.br/2011/07/os-principais-diagramas-da-uml-resumo-rapido/.

Diagrama de visão geral de interação: O diagrama de visão geral de interação representa o fluxo normal ou um fluxo alternativo, podendo mostrar os fluxos das atividades entre diagrama, tem semelhança com o diagrama de atividades e de sequência (IBM.COM, 2015).

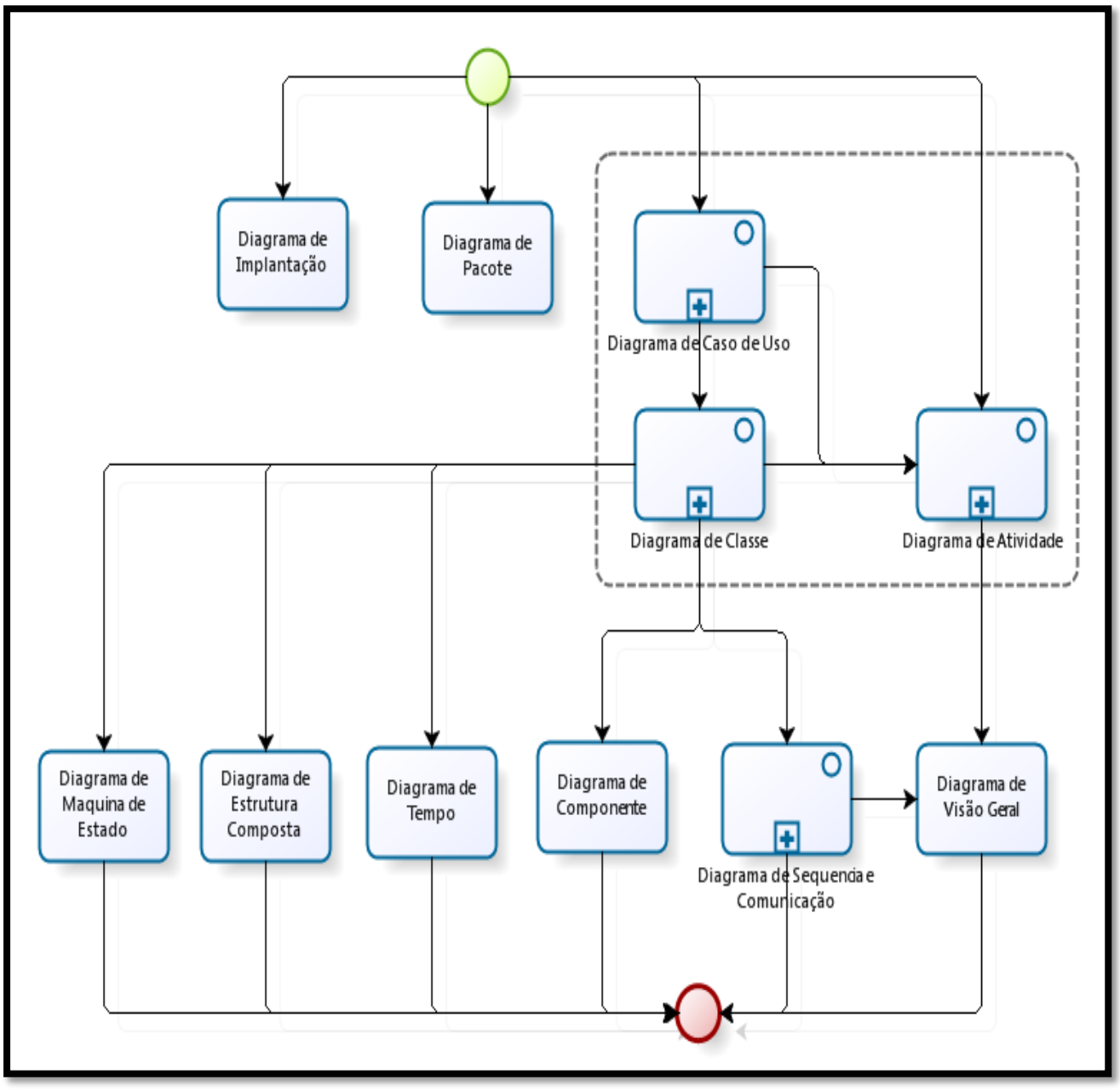

Figura 14 Diagrama de visão geral.

Fonte:http://gaia3.uel.br/projetos/oa_uml/Vis%C3%A3o%20Geral%20dos%20Diagramas%20da%20UML.htm.

Diagrama de tempo: Tem a finalidade de modelar as restrições temporais do software, podendo criar diversos diagramas focalizando cada interação, caso ocorra uma inclusão de algum elemento no diagrama de sequência, não poderá realizar a mesma inclusão no diagrama de tempo correspondente, quando for o contrário, poderá realizar a inclusão no diagrama de sequência, o diagrama de tempo representa uma parte da interação, não aderi todos os elementos do diagrama de sequência (IBM.COM, 2015).

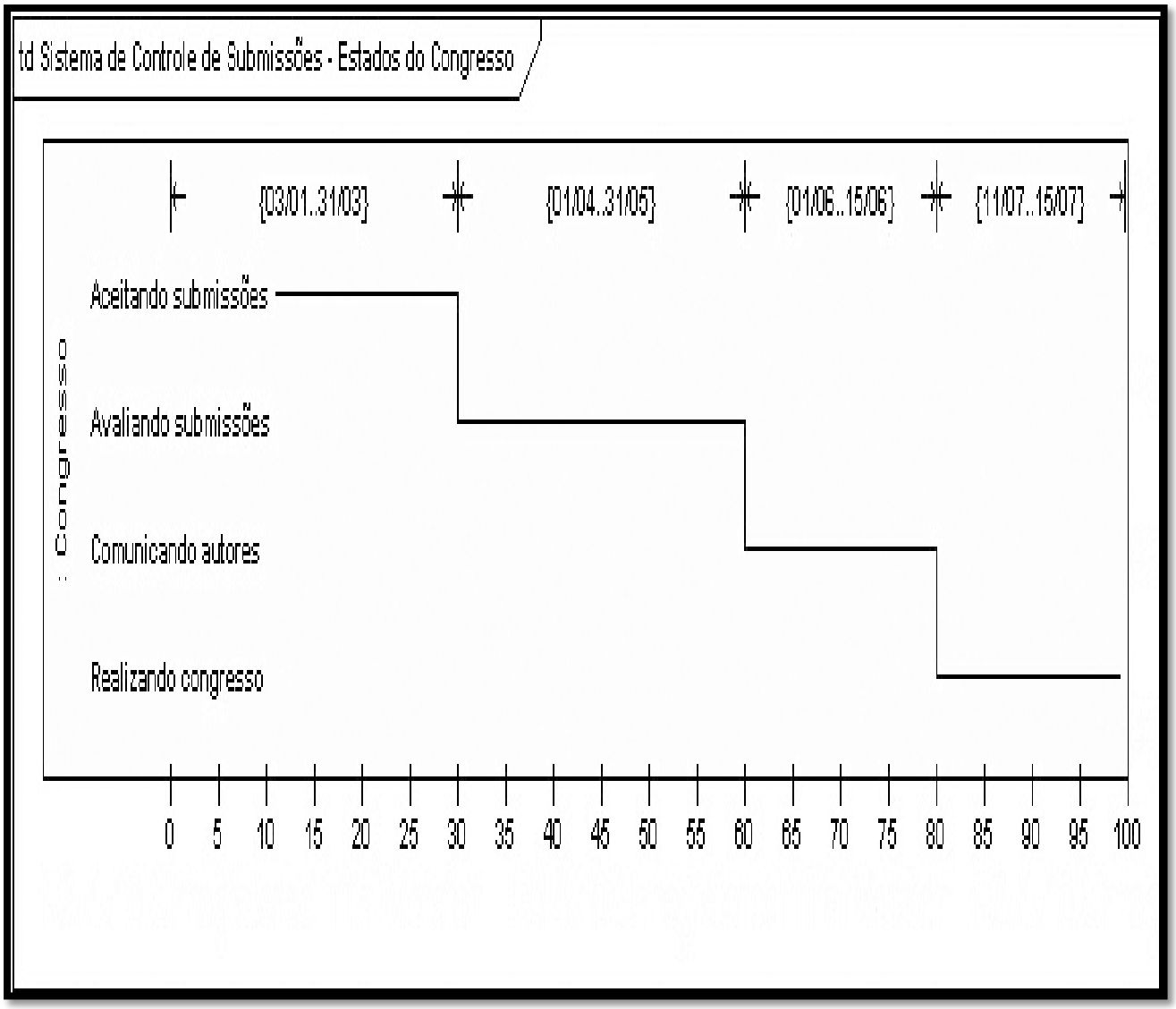

Figura 15 Diagrama de tempo. Fonte: http://docplayer.com.br/133677-Guia-pratico-2a-edicao-gilleanes-t-a-guedes-novatec.html.

Diagrama de colaboração: O diagrama de colaboração é muito semelhante ao de sequência, podendo realizar uma escolha de qual pretende usar, é representado como um diagrama de objeto mostrando os objetos juntamente com seus relacionamentos, as mensagens nomeadas mostrando sua ordem de execução, sendo representadas por setas (ETELG.COM.BR, 2016).

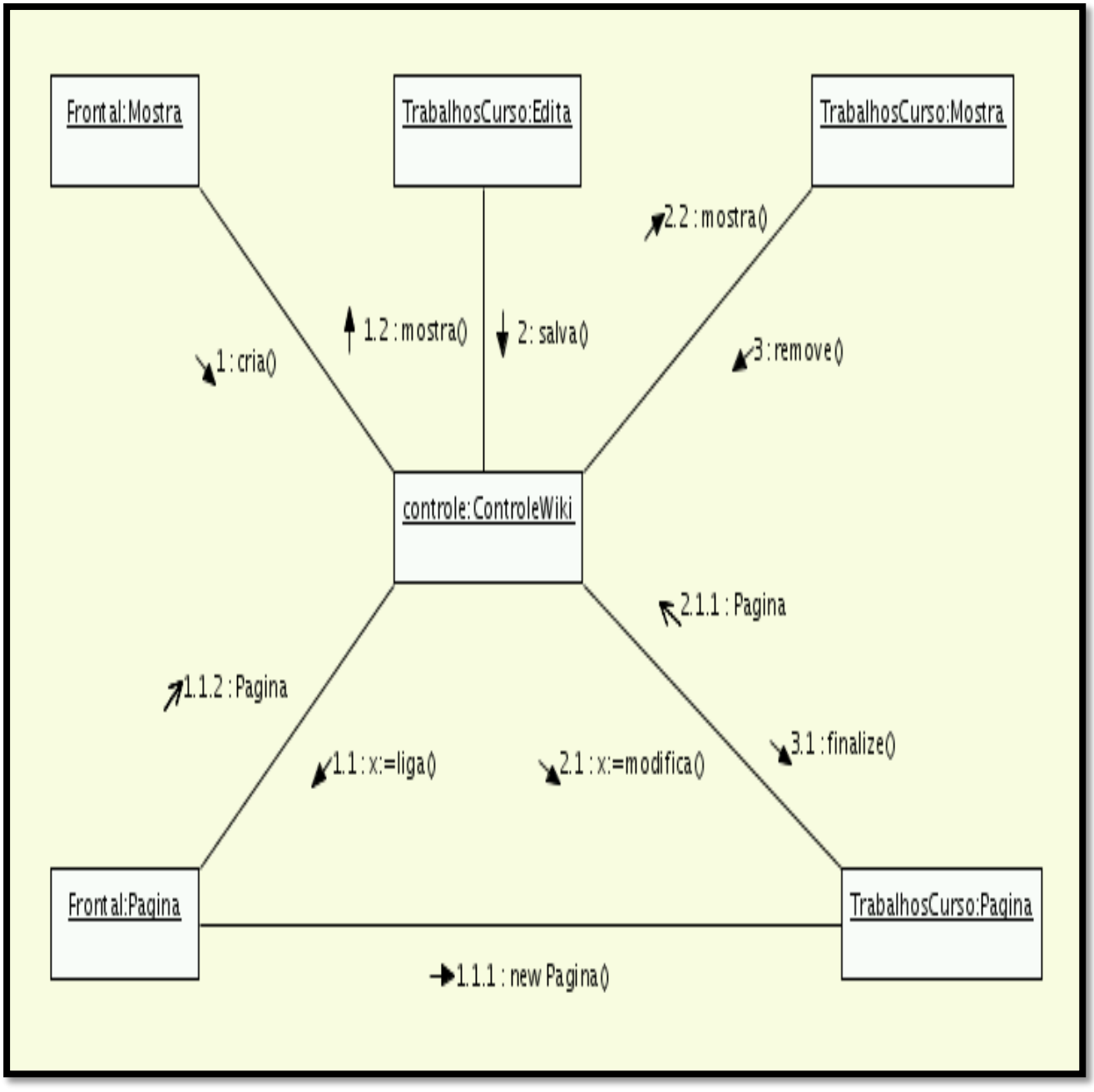

Figura 16 Diagrama de colaboração. Fonte: http://www.profissionaisti.com.br/2011/07/os-principais-diagramas-da-uml-resumo-rapido/.

4. MODELAGEM

4.1. CASOS DE USO

A figura a seguir tem como objetivo mostrar o diagrama de casos de uso do aplicativo de controle financeiro pessoal.

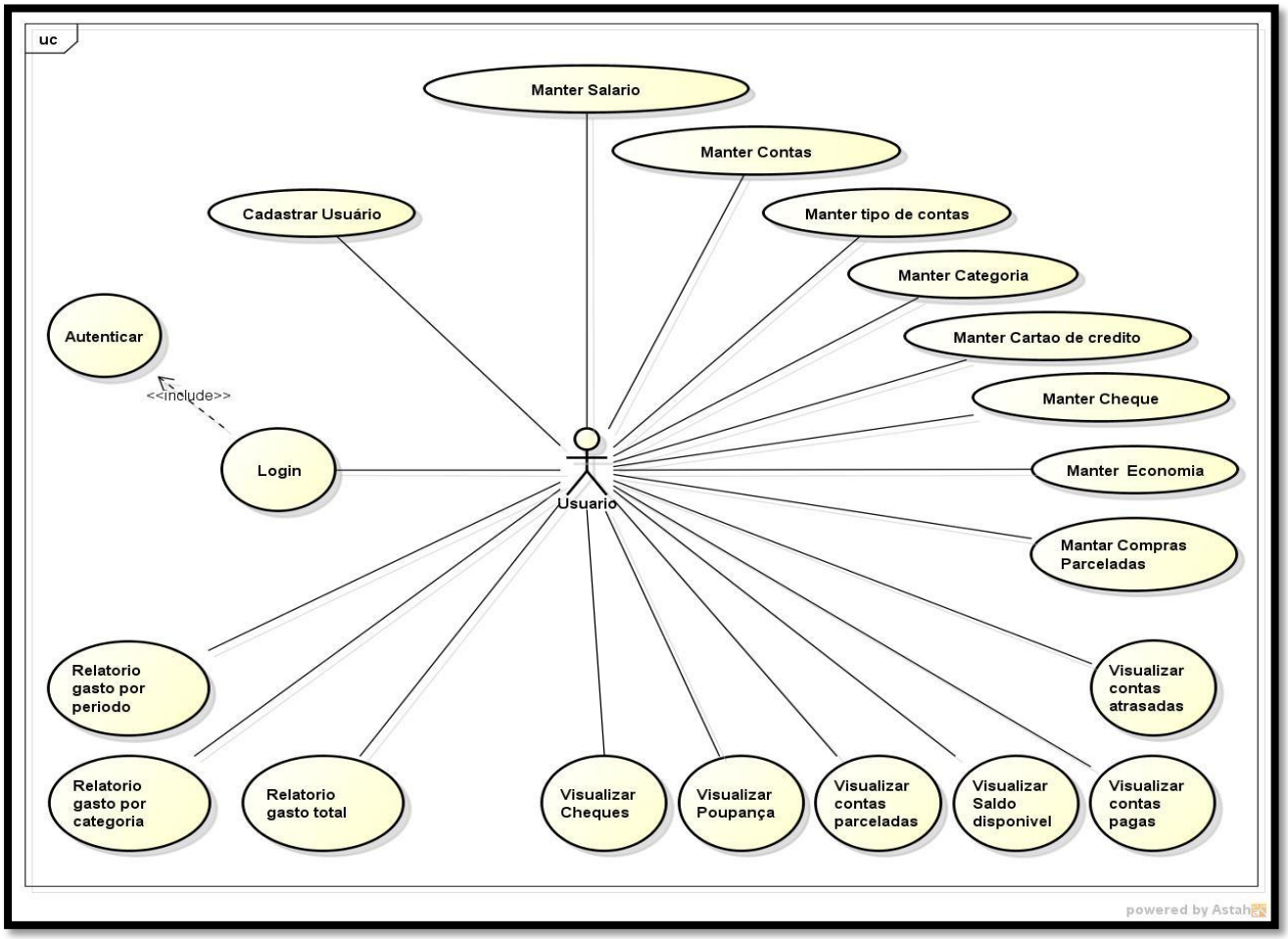

Figura 17 Diagrama de Casos de Uso referente ao projeto.

Como pode-se observar no diagrama o Usuário será o ator principal do aplicativo podendo realizar diversas interações com a aplicação.

Cada caso de uso mostrado, será basicamente a interação do usuário com o aplicativo, durante seu manuseio, a seguir será realizado uma especificação detalhada do funcionamento de cada caso de uso.

4.1.1. Especificações Casos de Uso

4.1.1.1. 1º Caso de Uso.

Caso de Uso: Manter Salário.

Descrição: O usuário pode realizar as seguintes funcionalidade: cadastrar, alterar, deletar. Para cadastrar, forneça as seguintes informações: Origem, valor.

Ator: Usuário.

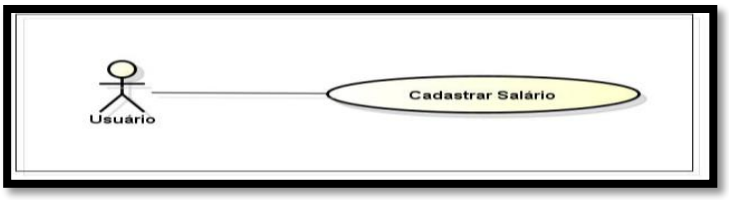

Figura 18 Caso de uso da interação do usuário com cadastrar salario

Fluxo Normal

1. Usuário realiza a iteração com manter salário.
2. O aplicativo retorna a tela de salários.
3. Usuário decide o que deve ser feito [A1], [A2], [A3].
4. O aplicativo retorna uma mensagem de confirmação.
5. O Usuário realiza a confirmação e se encerra o caso de uso.

Fluxo Alternativo A1

1. O usuário decide realizar o cadastro de um salário.
2. O aplicativo aguarda a entrada de dados.
3. Retorna para o fluxo normal de numeração 4.

Fluxo Alternativo A2.

1. O Usuário decide realizar alguma alteração em algum salário registrado.
2. O aplicativo aguarda a entrada de dados.
3. Retorna para o fluxo normal de numeração 4.

Fluxo Alternativo A3.

1. O usuário decide realizar a remoção de algum salário registrado.
2. O usuário seleciona o salário desejado.
1. Retorna para o fluxo normal de numeração 4.

2º Caso de Uso.

Caso de Uso: Cadastrar Usuário**.**

Descrição: Nesse caso de uso, será efetivado o cadastro de usuário, para quem não possua, podendo realizar a alteração de alguma informação dele.

Ator: Usuário.

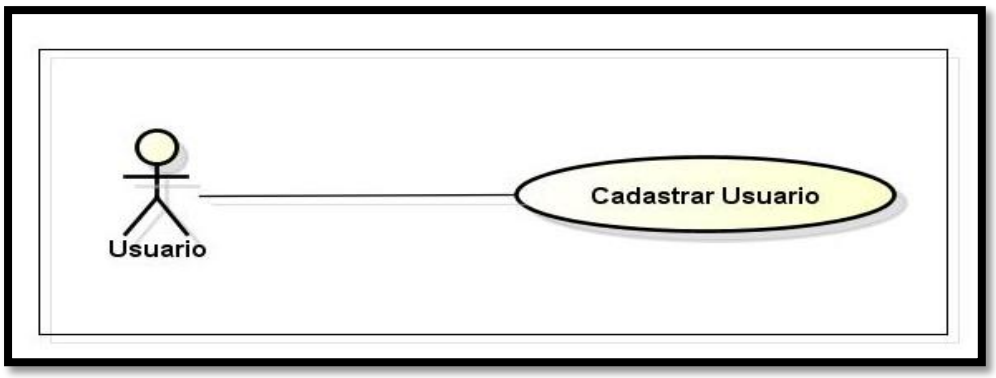

Figura 19 Caso de uso usuário interagindo com Cadastrar Usuário.

Fluxo Normal

1. Usuário realiza a iteração com Cadastrar Usuário.
2. O aplicativo retorna a tela de cadastro de usuário.
3. Usuário decide o que deve ser feito [A1], [A2], [A3].
6. O aplicativo retorna uma mensagem de confirmação.
7. O Usuário confirma.
8. Encerra o caso de uso.

Fluxo Alternativo A1

4. O usuário decide realizar o cadastro do usuário.
5. O aplicativo aguarda a entrada de dados.
6. Retorna para o fluxo normal de numeração 4.

Fluxo Alternativo A2.

4. O Usuário decide realizar alguma alteração de senha.
5. O aplicativo aguarda a entrada de dados.
6. Retorna para o fluxo normal de numeração 4.

3º Caso de Uso

Caso de uso: Manter contas

Descrição: O usuário pode realizar as seguintes funcionalidade: cadastrar, alterar, deletar e buscar. Para cadastrar, o aplicativo necessita das seguintes informações: descrição da conta, categoria, classificação, data de vencimento e de seu valor.

Ator: Usuário.

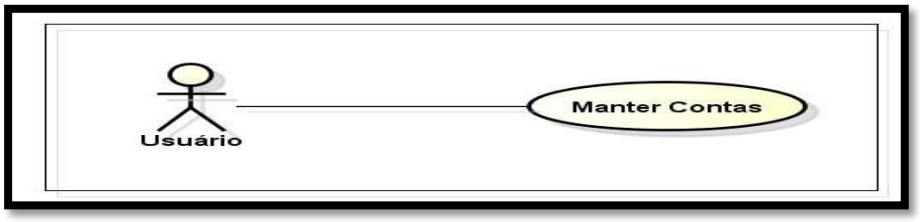

Figura 20 Caso de uso usuário interagindo com Manter Contas.

Fluxo Normal

1. Usuário realiza a iteração com manter contas.
2. O aplicativo retorna a tela de contas.
3. Usuário decide o que deve ser feito [A1], [A2], [A3].
4. O aplicativo retorna uma mensagem de confirmação.
5. O Usuário confirma.
6. Encerra o caso de uso.

Fluxo Alternativo A1

1. O usuário decide realizar o cadastro de uma conta.
2. O aplicativo aguarda a entrada de dados.
3. Retorna para o fluxo normal de numeração 4.

Fluxo Alternativo A2.

1. O Usuário decide realizar alguma alteração em alguma conta registrada.
2. O aplicativo aguarda a entrada de dados.
3. Retorna para o fluxo normal de numeração 4.

Fluxo Alternativo A3.

1. O usuário decide realizar a remoção de alguma conta registrada.
2. O usuário seleciona conta desejada.
3. Retorna para o fluxo normal de numeração 4.

4º Caso de Uso.

Caso de uso: Manter categoria.

Descrição: O usuário pode realizar as seguintes funcionalidade: cadastrar, alterar, deletar. Para cadastrar, o aplicativo necessita que seja informada sua descrição.

Ator: Usuário

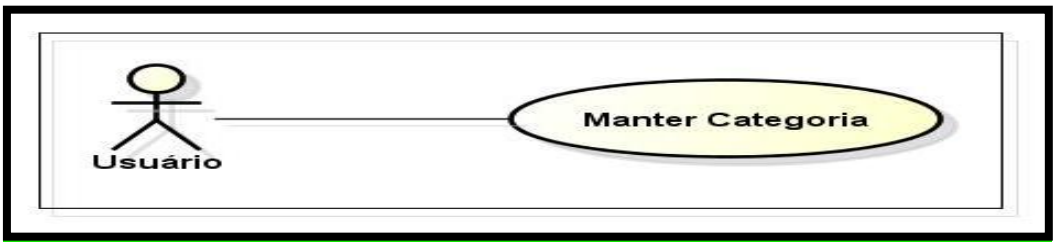

Figura 21 Caso de Uso usuário interagindo com Manter Categoria.

Fluxo Normal

1. Usuário realiza a iteração com manter categoria.
2. O aplicativo retorna a tela de categoria.
3. Usuário decide o que deve ser feito [A1], [A2], [A3].
4. O aplicativo retorna uma mensagem de confirmação.
5. O Usuário confirma.
6. Encerra o caso de uso.

Fluxo Alternativo A1

1. O usuário decide realizar o cadastro de uma categoria.
2. O aplicativo aguarda a entrada de dados.
3. Retorna para o fluxo normal de numeração 4.

Fluxo Alternativo A2.

1. O Usuário decide realizar alguma alteração em alguma categoria registrada.
2. O aplicativo aguarda a entrada de dados.
3. Retorna para o fluxo normal de numeração 4.

Fluxo Alternativo A3.

1. O usuário decide realizar a remoção de alguma categoria registrada.
2. O usuário seleciona categoria desejada.
3. Retorna para o fluxo normal de numeração 4

5º Caso de Uso

Caso de uso: Manter tipo de conta.

Descrição: O usuário pode realizar as seguintes funcionalidade: cadastrar, alterar, deletar. Para cadastrar, o aplicativo necessita das seguintes informações:

Ator: Usuário

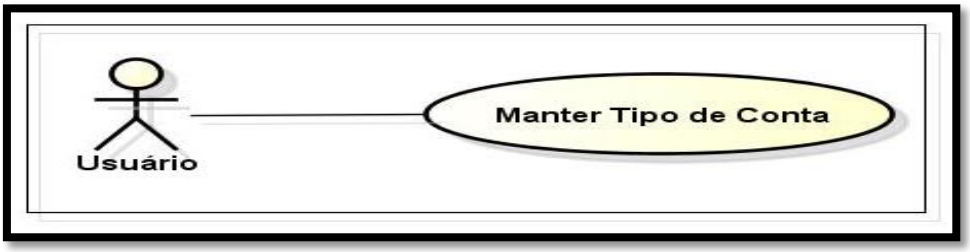

Figura 22 Caso de Uso usuário interagindo com Manter Tipo de Conta.

Fluxo Normal

7. Usuário realiza a iteração com manter tipo de conta.
8. O aplicativo retorna a tela de contas.
9. Usuário decide o que deve ser feito [A1], [A2], [A3].
10. O aplicativo retorna uma mensagem de confirmação.
11. O Usuário confirma.
12. Encerra o caso de uso.

Fluxo Alternativo A1

4. O usuário decide realizar o cadastro de um tipo de conta.
5. O aplicativo aguarda a entrada de dados.
6. Retorna para o fluxo normal de numeração 4.

Fluxo Alternativo A2.

4. O Usuário decide realizar alguma alteração em algum tipo de conta registrada.
5. O aplicativo aguarda a entrada de dados.
6. Retorna para o fluxo normal de numeração 4.

Fluxo Alternativo A3.

4. O usuário decide realizar a remoção de algum tipo de conta registrada.
5. O usuário seleciona conta desejada.
6. Retorna para o fluxo normal de numeração 4

6º Caso de Uso

Caso de uso: Relatório gasto por categoria.

Descrição: Neste caso de uso será realizado a emissão de um relatório mostrando para o usuário seus gastos separados por categoria, sendo um relatório detalhado de tudo o que foi gasto naquela categoria solicitada;

Ator: Usuário.

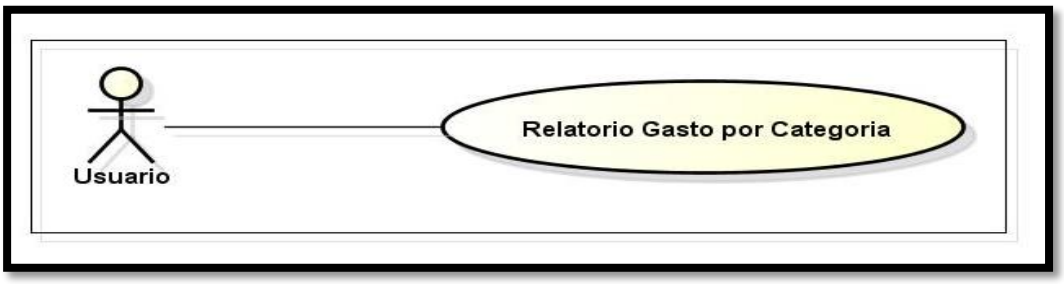

Figura 23 Caso de Uso usuário interagindo com Relatório Gasto por Categoria

Fluxo Normal

1. Usuário realiza a iteração com relatório gasto por categoria.
2. O aplicativo retorna uma tela para o usuário escolher a categoria.
3. Usuário escolhe a categoria.
4. O aplicativo retorna o relatório da categoria selecionada.
5. O usuário fecha o relatório.
6. O aplicativo retorna a tela de relatório.
7. O usuário fecha janela de relatório.
8. Encerra o caso de uso.

7º Caso de Uso

Caso de uso: Relatório gasto total.

Descrição: neste caso de uso será realizado a emissão de um relatório mostrando para o usuário seus gastos.

Ator: Usuário.

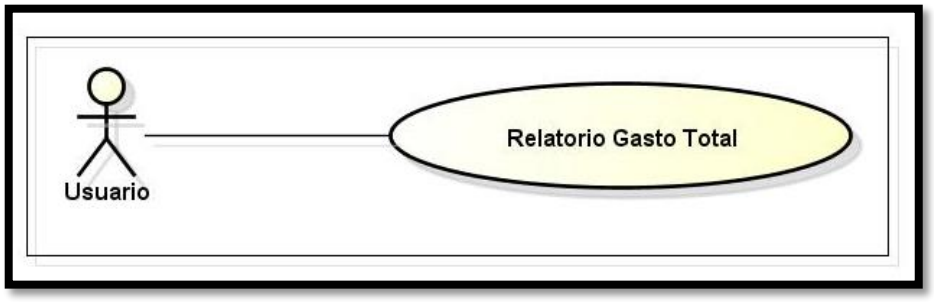

Figura 24 Caso de Uso usuário interagindo com Relatório Gasto Total

Fluxo Normal

1. Usuário realiza a iteração com o relatório gasto total.
2. O aplicativo retorna o relatório de gasto total.
3. Usuário fecha relatório.
4. Encerra o caso de uso.

8º Caso de uso

Caso de uso: Relatório gasto por período.

Descrição: Neste caso de uso será realizado a emissão de um relatório mostrando para o usuário qual foram seus gastos naquele determinado período, sendo um relatório detalhado de tudo o que foi gasto naquele período solicitado;

Ator: Usuário.

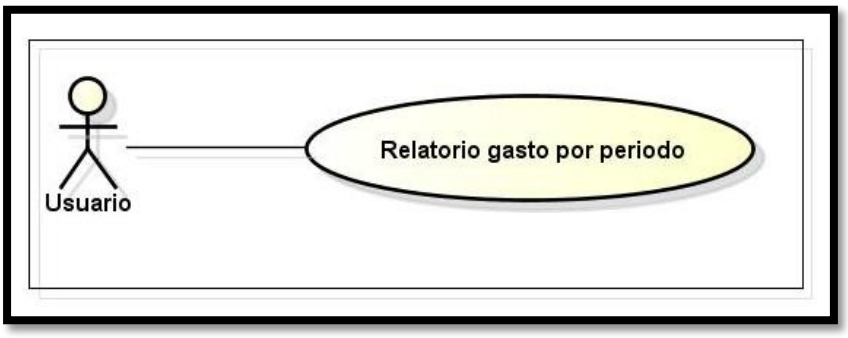

Figura 25 Caso de Uso usuário interagindo com Relatório gasto por período.

Fluxo Normal

1. Usuário realiza a iteração com relatório gasto por período.
2. O aplicativo retorna uma tela para o usuário escolha um período.
3. Usuário escolhe o período pretendido.
4. O aplicativo retorna o relatório do período solicitado.
5. O usuário fecha o relatório.
6. O aplicativo retorna a tela de relatório.
7. O usuário fecha janela de relatório.
8. Encerra o caso de uso.

9º Caso de Uso: Manter Cartão de Credito.

Descrição: Neste caso de uso usuário realizara o cadastramento de seus cartões de credito podendo realizar as seguintes funcionalidade: cadastrar, alterar, deletar. Para cadastrar, forneça as seguintes informações: Descrição, Bandeira, Limite, Vencimento.

Ator: Usuário.

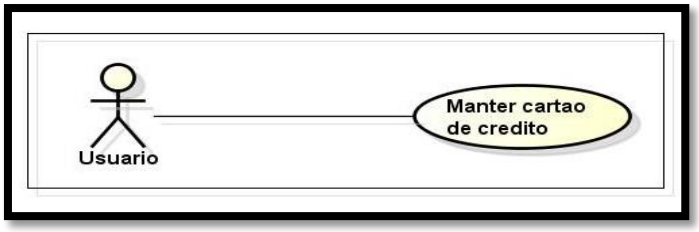

Figura 26 Casos de uso manter cartão de crédito.

Fluxo Normal

9. Usuário realiza a iteração com manter cartão de crédito.
10. O aplicativo retorna a tela de cartão de crédito.
11. Usuário decide o que deve ser feito [A1], [A2], [A3].
12. O aplicativo retorna uma mensagem de confirmação.
13. O Usuário realiza a confirmação e se encerra o caso de uso.

Fluxo Alternativo A1

7. O usuário decide realizar o cadastro de um cartão de crédito.
8. O aplicativo aguarda a entrada de dados.
9. Retorna para o fluxo normal de numeração 4.

Fluxo Alternativo A2.

7. O Usuário decide realizar alguma alteração em algum cartão registrado.
8. O aplicativo aguarda a entrada de dados.
9. Retorna para o fluxo normal de numeração 4.

Fluxo Alternativo A3.

2. O usuário decide realizar a remoção de algum cartão registrado.
3. O usuário seleciona o cartão desejado.
4. Retorna para o fluxo normal de numeração 4.

10º Caso de Uso: Manter cheque.

Descrição: Neste caso de uso usuário realizara o cadastramento de seus cheques podendo realizar as seguintes funcionalidade: cadastrar, alterar, deletar. Para cadastrar, forneça as seguintes informações: Descrição, Valor, Vencimento.

Ator: Usuário.

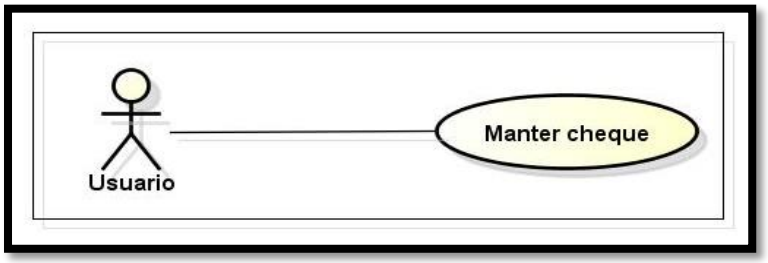

Figura 27 Caso de uso manter cheque.

Fluxo Normal

14. Usuário realiza a iteração com manter cheque.
15. O aplicativo retorna a tela de cheques.
16. Usuário decide o que deve ser feito [A1], [A2], [A3].
17. O aplicativo retorna uma mensagem de confirmação.
18. O Usuário realiza a confirmação e se encerra o caso de uso.

Fluxo Alternativo A1

10. O usuário decide realizar o cadastro de um cheque.
11. O aplicativo aguarda a entrada de dados.
12. Retorna para o fluxo normal de numeração 4.

Fluxo Alternativo A2.

10. O Usuário decide realizar alguma alteração em algum cheque registrado.
11. O aplicativo aguarda a entrada de dados.
12. Retorna para o fluxo normal de numeração 4.

Fluxo Alternativo A3.

5. O usuário decide realizar a remoção de algum cheque registrado.
6. O usuário seleciona o cheque desejado.
7. Retorna para o fluxo normal de numeração 4.

11º Caso de Uso: Manter Compras Parceladas.

Descrição: Neste caso de uso usuário realizara o cadastramento de suas contas parceladas podendo realizar as seguintes funcionalidade: cadastrar, alterar, deletar. Para cadastrar, forneça as seguintes informações: Descrição, Valor, Vencimento, Quantidade de parcelas.

Ator: Usuário.

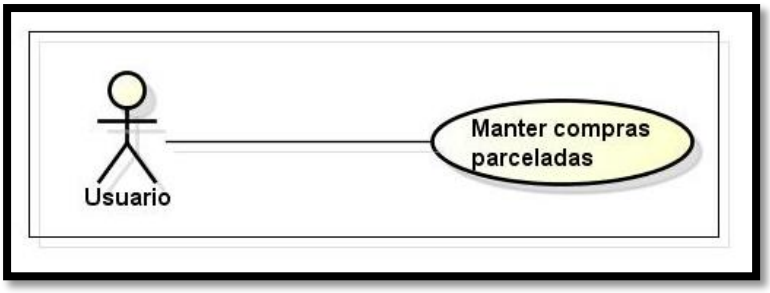

Figura 28 Caso de uso manter compras parceladas.

Fluxo Normal

19. Usuário realiza a iteração com manter compras parceladas.
20. O aplicativo retorna a tela de compras.
21. Usuário decide o que deve ser feito [A1], [A2], [A3].
22. O aplicativo retorna uma mensagem de confirmação.
23. O Usuário realiza a confirmação e se encerra o caso de uso.

Fluxo Alternativo A1

13. O usuário decide realizar o cadastro de uma compra parcelada.
14. O aplicativo aguarda a entrada de dados.
15. Retorna para o fluxo normal de numeração 4.

Fluxo Alternativo A2.

13. O Usuário decide realizar alguma alteração em alguma compra registrada.
14. O aplicativo aguarda a entrada de dados.
15. Retorna para o fluxo normal de numeração 4.

Fluxo Alternativo A3.

8. O usuário decide realizar a remoção de alguma compra registrada.
9. O usuário seleciona compra desejada.
10. Retorna para o fluxo normal de numeração 4.

12º Caso de Uso: Manter Economia.

Descrição: Neste caso de uso usuário realizara o cadastramento de suas economias, aonde o usuário poderá realizar as seguintes funcionalidade: depositar, modificar e retirar. Para depositar, forneça as seguintes informações: Valor, Dia;

Ator: Usuário.

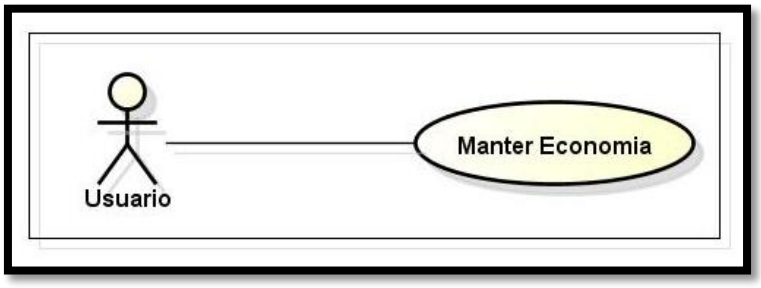

Figura 29 Caso de uso manter economia.

Fluxo Normal

24. Usuário realiza a iteração com manter economia.
25. O aplicativo retorna a tela de economia.
26. Usuário decide o que deve ser feito [A1], [A2], [A3].
27. O aplicativo retorna uma mensagem de confirmação.
28. O Usuário realiza a confirmação e se [encerra o caso de uso.

Fluxo Alternativo A1

16. O usuário decide realizar um deposito.
17. O aplicativo aguarda a entrada de dados.
18. Retorna para o fluxo normal de numeração 4.

Fluxo Alternativo A2.

16. O Usuário decide realizar alguma modificação em algum deposito.
17. O aplicativo aguarda a entrada de dados.
18. Retorna para o fluxo normal de numeração 4.

Fluxo Alternativo A3.

11. O usuário decide realizar uma retirada de sua economia.
12. O usuário informa o valor desejado.
13. Retorna para o fluxo normal de numeração 4.

13º Caso de Uso: Visualizar Cheque

Descrição: Neste caso de uso ocorre a visualização de todos os cheques que o usuário efetuou o cadastramento no aplicativo.

Ator: Usuário.

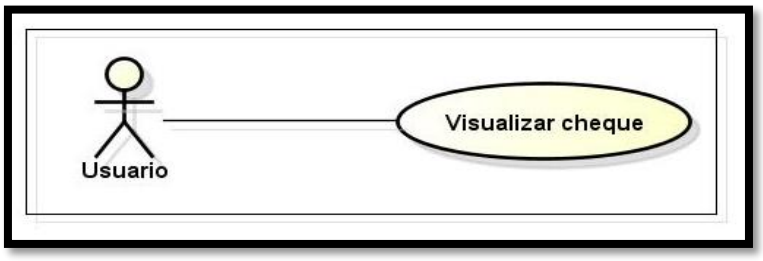

Figura 30 Caso de uso visualizar cheques.

Fluxo Normal

1. Usuário realiza a iteração com visualizar cheques.
2. O aplicativo retorna a tela de cheques.
3. O aplicativo oferece uma lista para o usuário constando todos seus cheques cadastrados
4. O usuário opta por fechar, encerrando-se o caso de uso

14º Caso de Uso: Visualizar Contas Parceladas

Descrição: Neste caso de uso ocorre a visualização de todas contas parceladas que o usuário efetuou o cadastramento no aplicativo.

Ator: Usuário.

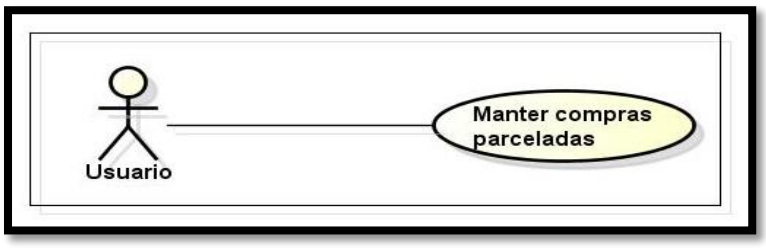

Figura 31 Caso de uso manter compras parceladas

Fluxo Normal

1. Usuário realiza a iteração com visualizar contas parceladas.
2. O aplicativo retorna uma tela com todas as contas parceladas.
3. O aplicativo oferece uma lista para o usuário com suas contas parceladas.
4. O usuário opta por fechar, encerrando-se o caso de uso

15º Caso de Uso: Visualizar contas pagas

Descrição: Neste caso de uso ocorre a visualização de todas contas pagas que o usuário efetuou o cadastramento no aplicativo.

Ator: Usuário.

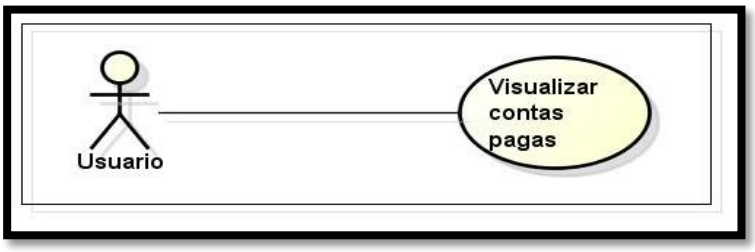

Figura 32 Caso de uso visualizar contas pagas

Fluxo Normal

1. Usuário realiza a iteração com visualizar contas pagas.
2. O aplicativo retorna a tela de contas pagas.
3. O aplicativo oferece uma lista para o usuário constando todas suas contas pagas cadastradas
4. O usuário opta por fechar, encerrando-se o caso de uso

16º Caso de Uso: Visualizar contas atrasadas

Descrição: Neste caso de uso ocorre a visualização de todas as contas atrasadas do usuário.

Ator: Usuário.

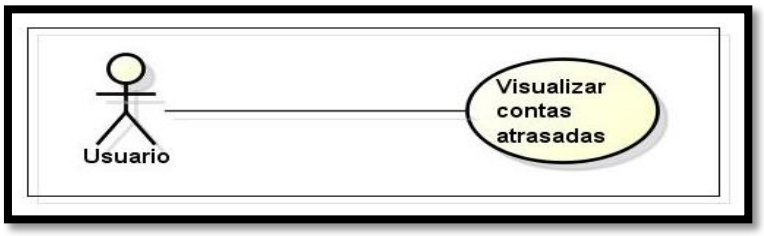

Figura 33 Caso de uso visualizar contas atrasadas.

Fluxo Normal

1. Usuário realiza a iteração com visualizar contas atrasadas.
2. O aplicativo retorna a tela de contas atrasadas.
3. O aplicativo oferece uma lista para o usuário constando todas suas contas em atraso.
4. O usuário opta por fechar, encerrando-se o caso de uso

17º Caso de Uso: Login

Descrição: Neste caso de uso ocorre quando o usuário já possua cadastro no aplicativo, caso ele não possua ainda deverá realizar o caso de uso de número 2.

Ator: Usuário.

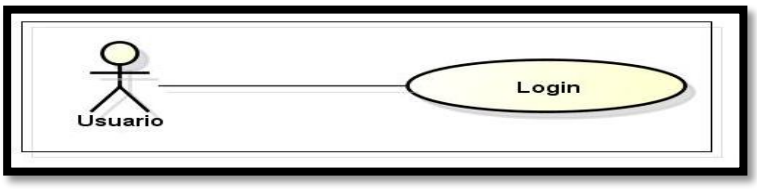

Figura 34 Caso de uso login.

Fluxo Normal

1. Usuário inicia o aplicativo
2. O aplicativo retorna a tela de login.
3. O usuário informa as informações para login e pressiona entrar.
4. Encerrando-se o caso de uso.

18º Caso de Uso: Autenticar

Descrição: Este caso de uso ocorre após o termino do caso de uso de login, onde será realizado a autenticação do usuário nesse caso de uso.

Ator: Usuário.

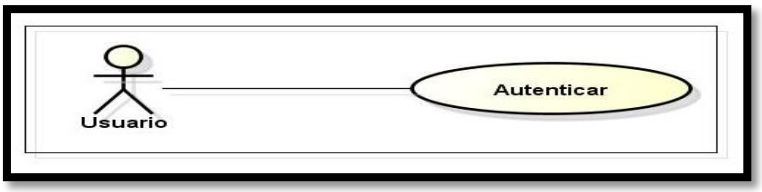

Figura 35 Caso de uso autenticar

Fluxo Normal

1. O aplicativo recebe os dados de login do usuário.
2. O aplicativo realiza uma autenticação no banco.
3. Após a autenticação é retornado a tela principal do aplicativo[A1].
4. Encerra-se o caso de uso.

Fluxo Alternativo A1

1. Caso não ocorra uma autenticação o sistema retorna uma mensagem.
2. Retorna ao passo 4.

19º Caso de Uso: Visualizar Economia

Descrição: Neste caso de uso ocorre a visualização das economias do usuário, informando para o usuário todos seus depósitos, retiradas e saldo atualizado.

Ator: Usuário.

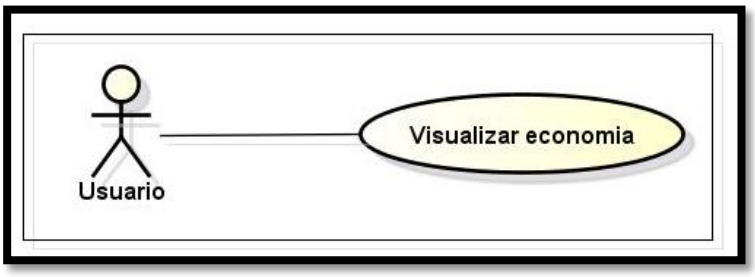

Figura 36 Caso de uso visualizar economia.

Fluxo Normal

1. Usuário realiza a iteração com visualizar economia.
2. O aplicativo retorna a tela de economia.
3. O aplicativo oferece uma lista para o usuário informando seus depósitos, retiradas e seus saldo atual
4. O usuário opta por fechar, encerrando-se o caso de uso

20º Caso de Uso: Visualizar saldo disponível

Descrição: Neste caso de uso é realizado um detalhamento ao usuário de suas despesas abatendo em seu salário informando o seu saldo disponível.

Ator: Usuário.

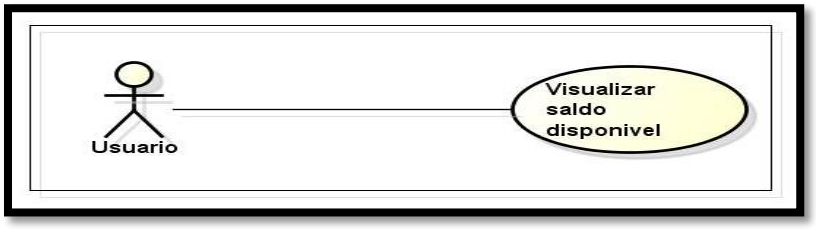

Figura 37 Caso de uso visualizar saldo disponível

Fluxo Normal

1. Usuário realiza a iteração com visualizar saldo disponível.
2. O aplicativo retorna a tela de saldo disponível.
3. O aplicativo oferece uma lista detalhando suas despesas e seu saldo.
4. O usuário opta por fechar, encerrando-se o caso de uso.

4.2. WBS - WORK BREAKDOWN STRUTURE

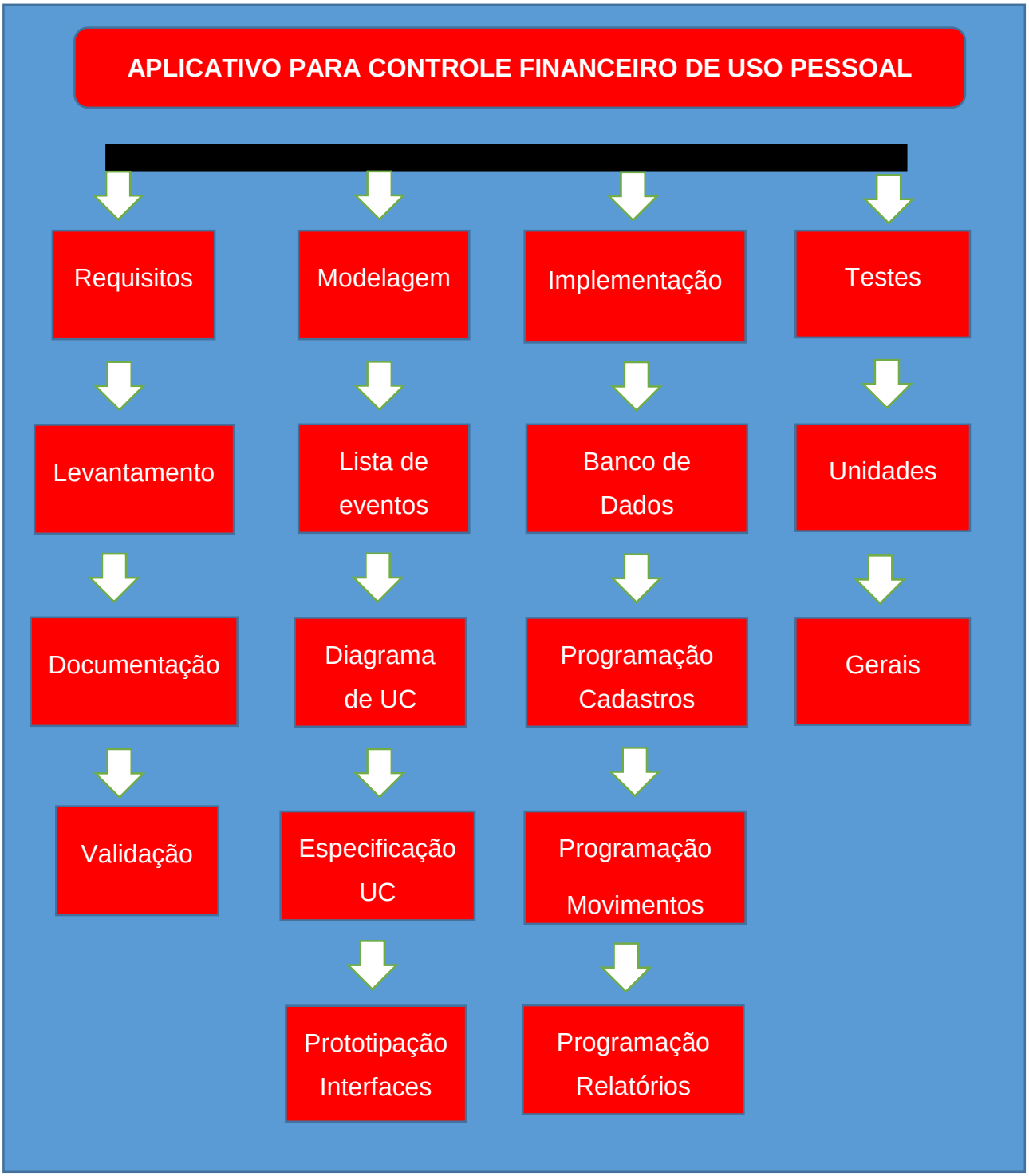

Figura 38 WBS - WORK BREAKDOWN STRUTURE

4.3. DIAGRAMA ENTIDADE RELACIONAMENTO

Tem como objetivo realizar a modelagem dos dados do projeto com alto nível de abstração, representando o modelo de negócio. Modelando as entidades e realizando a ligação de seus relacionamentos, mostrando suas dependências, simplificando a implementação no banco de dados.

A seguir a Figura 39 representa todas as tabelas do projeto com seus respectivos relacionamentos entre elas.

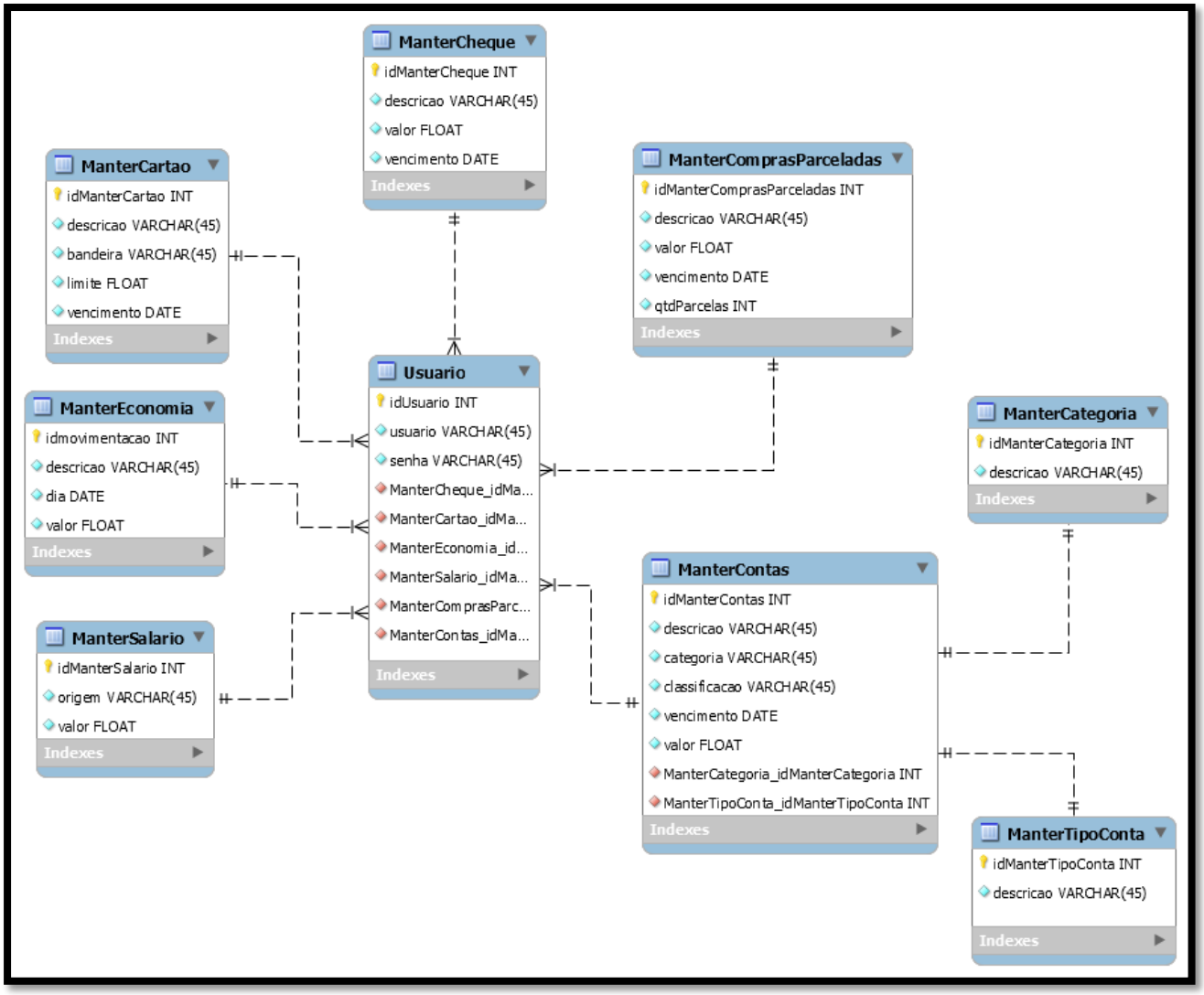

Figura 39 Diagrama entidade relacionamento.

4.4. DIAGRAMA DE CLASSE

A Figura 40 representa toda a estrutura de classe do projeto, especificando seus atributos e os métodos contidos em cada uma e seus respectivos relacionamentos.

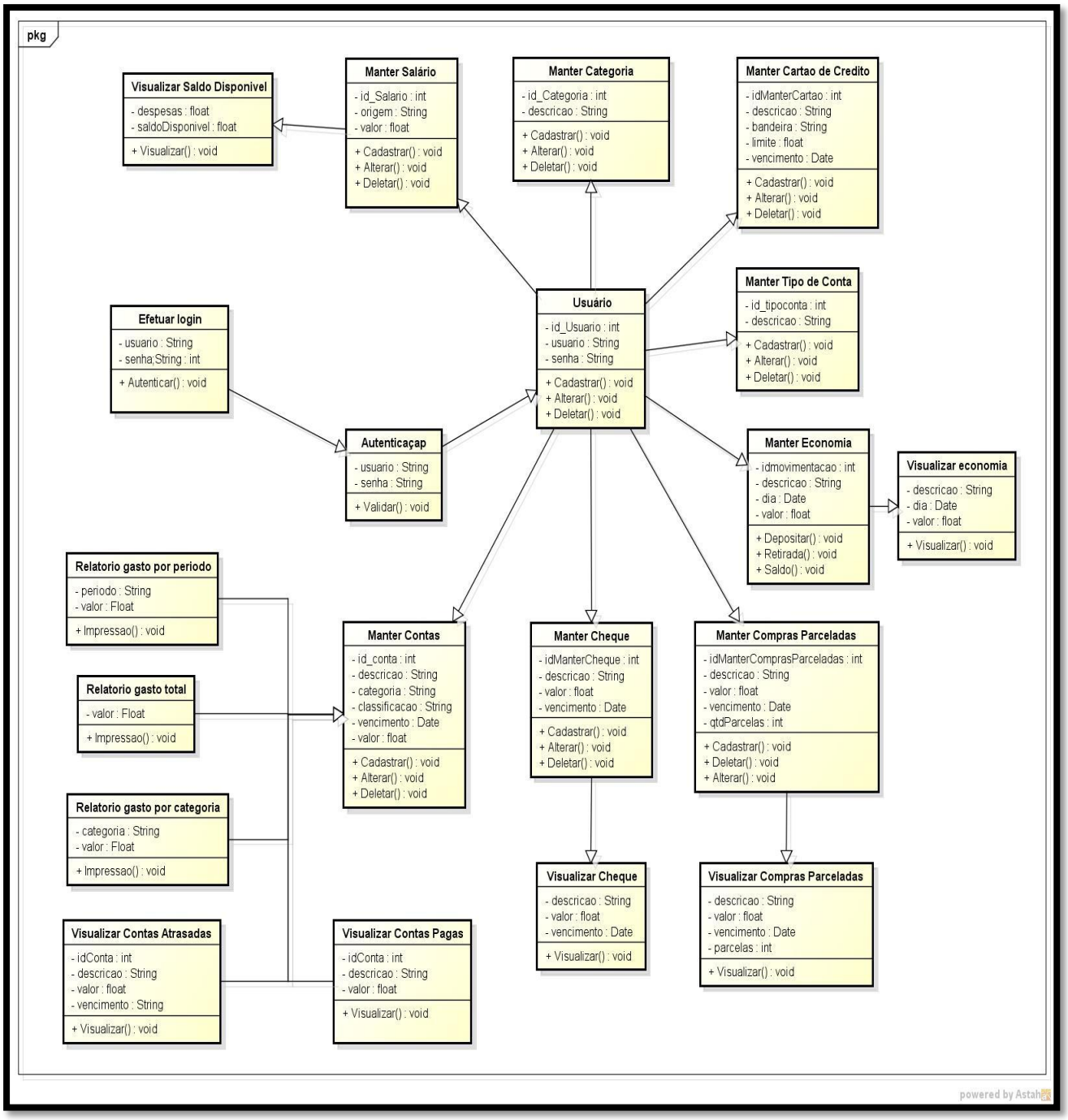

Figura 40 Diagrama de classe. Fonte: autor.

5. CRONOGRAMA

CRONOGRAMA										
Atividades	**Meses**									
	Nov	Dez	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago
Elaboração Pré-Projeto Entrega Pré-Projeto										
Desenvolvimento de estudo sobre o tema Leitura e levantamento de referência bibliográficas Levantamento de Requisitos										
Documentação de Requisitos Validação dos Requisitos Elaboração do Projeto										
Criação listas de eventos Diagrama de UC Especificação de UC										
Criação der Revisão documentação qualificação Entrega material para qualificação										
Banca de qualificação Correção pré-projeto definida pela banca Prototipação das Interfaces										

CONTINUAÇÃO DO CRONOGRAMA										
Atividades	**Meses**									
	Nov	Dez	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago
Banco de Dados Programação de Crud										
Programação das Movimentações Programação de relatórios Teste geral do projeto										
Revisão do material para a banca de defesa Entrega do material para a banca de defesa										
Banca de defesa Entrega monografia revisada Envio do TCC										

Tabela 3 Cronograma do projeto. Fonte: autor.

6. CONCLUSÃO

Durante o desenvolvimento do projeto, encontrei certas dificuldades por se tratar de um assunto novo, onde não tinha nenhum conhecimento sobre a plataforma *Google Android* e o Banco de Dados *SQLITE*, necessitando um estudo sobre ambos.

A modelagem do aplicativo foi de extrema importância para o desenvolvimento do aplicativo, desde o diagrama de caso de uso até o diagrama de classe, simplificando o entendimento de suas funcionalidades e a forma de como teria que ser implementada tal funcionalidade.

Com a fase final do projeto, posso dizer que o projeto me proporcionou um certo conhecimento sobre a tecnologias utilizadas em seu desenvolvimento, podendo facilitar o desenvolvimento de futuros projeto para a plataforma.

REFERÊNCIAS

BRAZILIENSE, Correio. **Crise gera perda do poder de compra e endividamento maior, diz CNI.** Disponível em: <http://www.correiobraziliense.com.br/app/noticia/economia/2015/09/09/internas_economia,497906/crise-gera-perda-do-poder-de-compra-e-e ndividamento-maior-diz-cni.shtml>. Acesso em: 20 out. 2015. Acessado em 20 out, 2015.

CAVALCANTI, Mayra. **Smartphone já está presente em 28% da população Brasileira. Mundo Bit.**

<http://blogs.ne10.uol.com.br/mundobit/2015/03/16/smartphone-ja-esta-presente-em-28-

da-populacao-Brasileira/>. Acesso em: 20out, 2015.

ETELG.COM.BR. **Linguagem de Modelagem Unificada.** Disponível em: <http://www.etelg.com.br/paginaete/downloads/informatica/apostila_uml.pdf>. Acesso em: 08 mar. 2016.

G1.COM. **Endividamento das famílias chega a 46,3%, o maior em 10 anos, mostra BC.** 2015. Disponível em: <http://g1.globo.com/economia/seu-dinheiro/noticia/2015/06/endividamento-das-familias-chega-463-o-maior-em-10-anos-mostra-bc.html>. Acessado em 20out, 2015.

IBM.COM. **Documentação do produto IBM Rational Software Architect.** Disponível em: <http://www.ibm.com/support/knowledgecenter/SS8PJ7_9.1.2/com.ibm.xtools.rsa_base.legal.doc/helpindex_rsa_base.html?cp=SS8PJ7_9.1.2/0&lang=pt-br>. Acesso em: 08 mar. 2016.

MEYER, Maximiliano. **A história do Android.** Disponível em: <https://www.oficinadanet.com.br/post/13939-a-historia-do-Android>. Acesso em: 05 mar. 2016.

NIELSEN.COM. **68 MILHÕES USAM A INTERNET PELO SMARTPHONE NO BRASIL.** Disponível em: <http://www.nielsen.com/br/pt/press-room/2015/68-milhoes-usam-a-Internet-pelo-smartphone-no-Brasil.html>. Acesso em: 20 out. 2015.

OMG.ORG. **Introduction to OMG's unified modeling language.** Disponível em: <http://www.omg.org/gettingstarted/what_is_uml.htm>. Acesso em: 08 mar. 2016.

RIBEIRO, Valdir. **9 apps para você controlar seus gastos**. Exame.

<http://exame.abril.com.br/tecnologia/noticias/9-apps-para-voce-controlar-seus-gastos-1#2>. Acesso em: 20 out, 2015.

ROCHA NETO, Elói. **Material sobre UML.** Disponível em: <http://www.dsc.ufcg.edu.br/~sampaio/cursos/2007.1/Graduacao/SI-II/Uml/historia_uml/historia_uml.htm>. Acesso em: 08 mar. 2016.

SQLITE.ORG. **About SQLite.** Disponível em: << http://www.SQLite.org/about.html>. Acesso em: 07 mar. 2016.

TUDOCELULAR.COM. **Android e iOS detêm 96,3% do mercado mundial de smartphones.** Disponível em: <www.tudocelular.com/Android/noticias/n50237/Android-ios-96-3-mercado-mundialsmartphones>. Acesso em: 20 out. 2015.

VARGAS, Thânia Clair de Souza. **A história de UML e seus diagramas.** Disponível em: <https://projetos.inf.ufsc.br/arquivos_projetos/projeto_721/artigo.tcc.pdf>. Acesso em: 08 mar. 2016.

VELOSO, Por Thássius. **Google Android: conheça todos os nomes, sabores e versões da plataforma móvel.**Disponível em: <https://tecnoblog.net/56850/google-android-versoes-nomes-sabores/>. Acesso em: 10 mar. 2015.