

FACULDADES INTEGRADAS DO VALE DO IVAÍ

Mantida pela Instituição Cultural e Educacional de Ivaiporã – ICEI

Recredenciada pela Portaria nº. 545 de 11/05/2012 – D.O.U. – 14/05/2012

Tecnologia em Análise e Desenvolvimento de Sistemas

CH CONTROLE

Controle de Finanças Pessoais para Aplicativo Móvel

CARLOS HENRIQUE VENCIGUERRA

CÉSAR AUGUSTO MALVEZI

Orientador

IVAIPORÃ
2013

Faculdades Integradas do Vale do Ivaí – Univale
Tecnologia em Análise e Desenvolvimento de Sistemas

CARLOS HENRIQUE VENCIGUERRA

CH CONTROLE
Controle de Finanças Pessoais para Aplicativo Móvel

Trabalho de conclusão de curso apresentado à disciplina de Estágio Supervisionado do curso de Tecnologia em Análise e Desenvolvimento de Sistemas das Faculdades Integradas do Vale do Ivaí, como requisito parcial para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador: Prof. César Augusto Malvezi

IVAIPORÃ
2013

CARLOS HENRIQUE VENCIGUERRA

CH CONTROLE
Controle de Finanças Pessoais para Aplicativo Móvel

Trabalho de conclusão de curso apresentado à disciplina de Estágio Supervisionado do curso de Tecnologia em Análise e Desenvolvimento de Sistemas das Faculdades Integradas do Vale do Ivaí, como requisito parcial para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas sob apreciação da seguinte comissão examinadora:

Prof. César Augusto Malvezi – Orientador
Faculdades Integradas do Vale do Ivaí

Prof.
Faculdades Integradas do Vale do Ivaí

Prof.
Faculdades Integradas do Vale do Ivaí

Ivaiporã, 30 de Novembro de 2013.

Dedico este trabalho a os meus amigos, familiares, minha esposa e meu filho que esta para nascer que mesmo com todas as dificuldades sempre estive do meu lado dando o maior apoio possível a todos meu obrigado.

AGRADECIMENTOS

Agradeço primeiramente a Deus que sem ele nada é possível, em segundo para meu orientador Professor César Augusto Malvezi que sempre esteve do meu lado dando maior força e incentivo para que este projeto desse certo, em terceiro todos os professores que me deram aulas nesses 3 anos de faculdade e souberam passar seu conhecimentos para que no final do curso usando essas disciplinas meu auxiliou para a concretização deste projeto.

Epígrafe

“Que os vossos esforços desafiem as possibilidades, lembrai-vos de que as grandes coisas do homem foram conquistadas do que parecia impossível.”

(Charles Chaplin)

VENCIGUERRA, Carlos. CH CONTROLE: Controle de Finanças Pessoais para Aplicativo Móvel – Curso de Tecnologia em Análise e Desenvolvimento de Sistemas, Faculdades Integradas do Vale do Ivaí, Ivaiporã, 2013.

RESUMO

Esse projeto consiste na criação de um aplicativo com base na plataforma Android que poderá ser executado na versão 2.2 com API nível 8, ou superior. Neste projeto faremos uso de uma linguagem destinada à programação de aplicativo para dispositivos Móveis com sistema Operacional Android. O projeto CH COTROLE aplicativo para Controle de Finanças, tendo como objetivo que cada indivíduo tenha uma vida financeira saudável. Este aplicativo móvel visa a melhorar a vida do cidadão que vive sem controle nenhum de seus gastos, pois em um mundo competitivo com o de hoje você saber se deve gastar quanto gastar e onde gastar pode ser de muito útil, com o uso correto este aplicativo pode se tornar uma ferramenta importante para a economia de cada pessoa que se beneficiar com este produto. Este projeto vai ser desenvolvido para controle pessoal de finanças, com tecnologia Android sendo abordados aqui algumas ferramentas que serão utilizadas no desenvolvimento, como o eclipse, *plugin* Adt, Sdk manager, *SQLite* xmltem como objetivo desenvolver um aplicativo para melhorar a situação financeira daquela pessoa que deseja ter uma vida financeira saudável, com controle total de seus gastos, usando um dispositivo móvel smartphone o usuário vai baixar o aplicativo cadastrar suas despesas e receitas no momento da ação e manter-se sempre atualizado sobre sua situação financeira.

Palavras-chave: Finanças, Aplicativos, Smartphone, Android.

LAST NAME, First Name; LAST NAME, First Name. Title: Subtitle – Graduation Course in Technology of Analysis and System Development, Integrated Colleges of Ivaí Valley, Ivaiporã, 2012.

ABSTRACT

This project is to create an application based on the Android platform that can run on version 2.2 with API level 8 or higher . This project will make use of a programming language for application for Mobile devices with Android operating system. The CH COTROLE project application to Control Finance , with the aim that each individual has a healthy financial life . This mobile application aims to improve the lives of citizens living without spending any of your control, because in a competitive world with today whether you should spend how much to spend and where to spend can be very useful , with the correct use this app can become an important tool for the economy of each person to benefit from this product . This project will be developed to control personal finances with Android technology being discussed here some tools that will be used in the development , as the eclipse Adt plugin , Sdk manager , SQLite and xml aims to develop an app to improve the financial situation of that person who wishes to have a healthy financial life , with total control of your spending using a mobile smartphone users will download the application registering your expenses and income at the time of action and keep up to date on your financial situation .

Keywords: Finance, Apps, Smartphone, Android.

LISTA DE ILUSTRAÇÕES

Figura 1: Controle de Finanças Pessoais	12
Figura 2: Controle de Finanças Pessoais	13
Figura 3: Orçamento	14
Figura 4: Ilustração da Tela do Finanças Pessoais	15
Figura 5: Ilustração da Tela do Money Wise	16
Figura 6: Ilustração da Tela do Money Lover	17
Figura 7: Arquitetura Android.....	19
Figura 8: Ciclo de vida da Activity	20
Figura 9: Fase do RUP	23
Figura 10: Caso de Uso	27
Figura 11: Diagrama de Classe	32
Figura 12: Diagrama de Atividade 1	33
Figura 13: Diagrama de Atividade 2	34
Figura 14: Diagrama de Sequência 1	35
Figura 15: Diagrama de Sequência 2	35
Figura 16: Diagrama de Sequência 3	36
Figura 17: Diagrama de Sequência 4	36
Figura 18: Diagrama de Sequência 5	37
Figura 19: DER	38
Figura 20: Tela Login.....	42
Figura 21: Tela Principal.....	42
Figura 22: Tela Registro.....	43
Figura 23: Index Developer.....	45
Figura 24: Licença para uso do Android SDK	45
Figura 25: Ícone Eclipse.....	46
Figura 26: Caminho Workspace.....	47
Figura 27: Início Developer Tools	47
Figura 28: Tela inicial do Android Developer Tools	48
Figura 29: Android SDK Manager (APIs)	49
Figura 30: Escolher a versão do API	50
Figura 31: Escolher API.....	51
Figura 32: Escolher Pacote para instalação	52
Figura 33: API instalada.....	52
Figura 34: Ícone do Android Virtual Device Manager	53
Figura 35: Tela do Android Virtual Device Manager	54
Figura 36: Tela configuração das (APIs)	55
Figura 37: AVD configurado	56
Figura 38: Criar um projeto de aplicativo Android.....	57
Figura 39: Tela de configuração do novo aplicativo Android.....	58
Figura 40: Tela de finalização de configuração	59
Figura 41: Ferramenta de visualização do projeto	60
Figura 42: Pasta do projeto.....	61
Figura 43: Classe Tela_Login.Java	62
Figura 44: Arquivo AndroidManifest	65
Figura 45: Arquivo R.Java	66
Figura 46: Comparação de funcionalidades	69

LISTA DE ABREVIATURAS E SIGLAS

ADT	<i>Android Development Tools</i>
API	<i>Application Programming Interface</i>
AVD	<i>Android Virtual Device</i>
SDK	<i>Software Development Kit</i>
XML	<i>eXtensible Markup Language</i>
RUP	<i>Rational Unified Process</i>
UML	<i>Unified Modeling Language</i>
S.O.	<i>Sistema Operacional</i>

SUMÁRIO

1	Introdução.....	10
1.1	CONTEXTUALIZAÇÃO	10
1.2	OBJETIVOS.....	11
1.2.1	Geral	11
1.2.2	Específicos	11
1.3	JUSTIFICATIVA	11
1.4	ESCOPO E DELIMITAÇÃO DO TRABALHO	12
2	Fundamentação Teórica	13
2.1	FINANÇAS PESSOAIS	13
2.2	ORÇAMENTO.....	14
2.3	DISPOSITIVOS SIMILARES.....	15
2.3.1	Finanças Pessoais.....	15
2.3.2	Money Wise	16
2.3.3	Money Lover	17
2.4	Android	18
2.4.1	Arquitetura Android	18
2.4.2	Runtime do android	19
2.4.3	Android SDK	19
2.4.4	Activity.....	20
2.4.5	Persistência de Dados	21
3	METODOLOGIA	21
3.1	SQLite	22
3.2	RuP (Rational Unified Process)	23
3.3	ASTAH COMMUNITY	24
3.4	Requisitos.....	24
3.4.1	Requisitos Funcionais.....	25
3.4.2	Requisitos Não Funcionais.....	25
3.5	Casos de Uso	26
3.5.1	Diagramas de casos de uso.....	27
3.5.2	Descrição dos casos de uso	27

3.6	Diagrama de classes	31
3.7	Diagrama Atividades	32
3.8	DAGRAMA DE SEGUENCIA.....	35
3.9	Banco de dados.....	38
3.9.1	Diagrama Entidade Relacionamento	38
4	PROJETO DO SISTEMA	40
4.1	Escopo	40
4.2	Regras de negócio.....	41
4.3	Interface Gráfica	42
5	Projeto de DESENVOLVIMENTO.....	44
5.1	Roteiro de instalação	44
6	CRONOGRAMA	67
6.1	CRONOGRAMA INICIAL	67
6.2	DESCRIÇÕES DAS TAREFAS DO CRONOGRAMA	67
7	projeto de implantação	68
7.1	Ambiente usuário	68
8	Resultados e discussão	69
9	Trabalhos futuros.....	70
10	Conclusão	71
	Referências Bibliográficas	72
	Referências Bibliográficas – SITES	73

1 INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO

Com a falta de tempo, as pessoas não conseguem controlar suas finanças, deixando sua situação financeira prejudicada. Através da ajuda de Aplicativos Móveis pode-se possuir um software que faça um controle detalhado de suas despesas e receitas podendo auxiliar na tomada de decisão em uma compra ou investimento com mais comodidade.

Criar um aplicativo para que as pessoas possam controlar melhor suas contas a pagar, seus gastos e usa-lo para acessar as informações de suas finanças pessoais ou empresariais de onde estiver, tendo apenas um Smartphone em mãos.

Contata-se então que nosso maior problema é controlar nossos gastos na hora que surge um tempo livre. Então que ferramenta criar para resolver esta questão de falta de tempo e oportunidade de melhorar nosso controle pessoal sobre nossas despesas.

Com planilhas de controle para finanças pessoais ou empresariais, fica difícil o indivíduo manipular ou consultar suas contas sempre que necessário. Com esta possível solução o usuário poderá ter mais controle sobre seus gastos ou investimentos. O desafio deste projeto é que a pessoa que utiliza este aplicativo esteja sempre sabendo detalhadamente sua situação financeira para que consiga tomar decisões certas na hora certa.

Este projeto é importante para pessoas que se preocupa em administrar suas economias ou sua pequena empresa sem ter que estar na frente de um computador ou livros de controle.

Com este aplicativo poderemos interagir com nossas finanças de qualquer lugar com um Smartphone com sistema operacional android, além de não atrasar contas a pagar eliminando o risco de pagar juros e comprometer seu orçamento mensal.

Pretende-se então por meio deste projeto criar um aplicativo que melhorará a qualidade de vida das pessoas por meio de programa e planejamento que ajustaram o tempo de vida econômica da pessoa que se beneficiará com este aplicativo.

1.2 OBJETIVOS

1.2.1 Geral

Desenvolver um protótipo de aplicação para dispositivos móvel de controle de finanças pessoais para que os usuários possam controlar melhor suas contas tendo uma situação financeira saudável.

1.2.2 Específicos

O protótipo terá mecanismo de sinalização e alertas, caso a despesa o receita no dia do vencimento, o usuário poderá visualizar todos seus gastos, obtendo um controle maior de contas a pagar e receber.

1.3 JUSTIFICATIVA

Esta pesquisa é importante para pessoas que se preocupa em administrar suas economias domesticas sem ter que estar na frente de um computador ou livros de controle.

Com este aplicativo poderemos interagir com nossas finanças de qualquer lugar, além de não atrasar contas a pagar, eliminando o risco de pagar juros e comprometer seu orçamento mensal.

Pretende-se então por meio desta pesquisa criar um aplicativo que melhorará a qualidade de vida das pessoas por meio de programa e planejamento que ajustaram o tempo de vida econômica da pessoa que se beneficiará com este aplicativo.

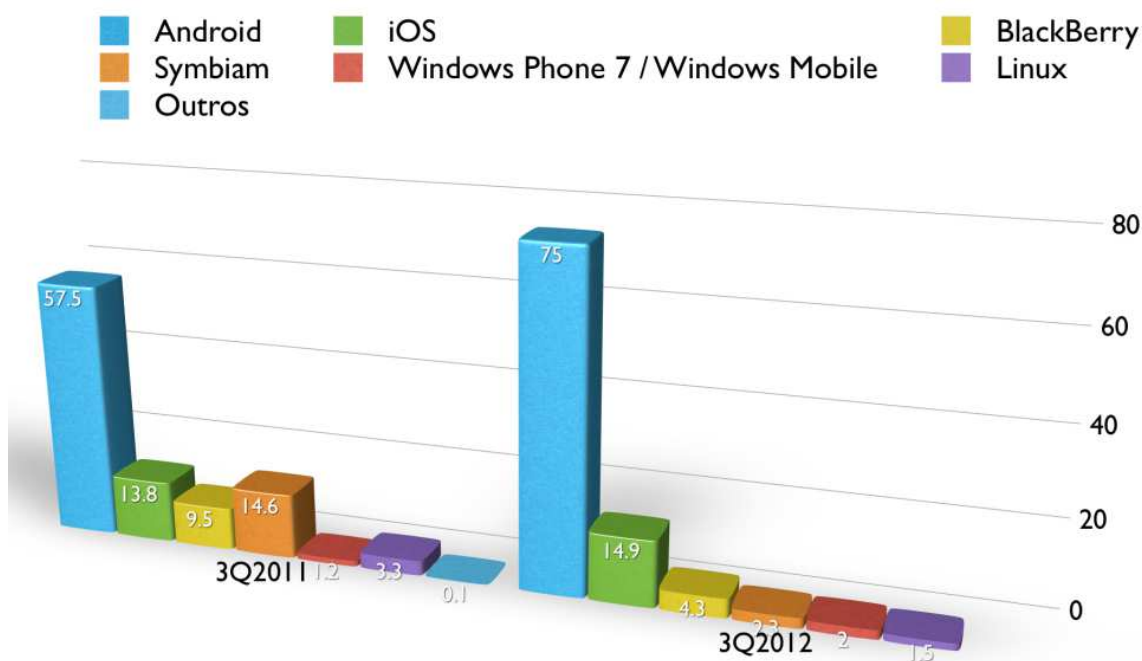
1.4 ESCOPO E DELIMITAÇÃO DO TRABALHO

Na atualidade, aplicativo móvel para controle de finanças existe milhares no mercado, “[...] Os Dispositivos Android se apresentam em todas as formas e tamanhos” (LEE, 2011, p.5).

Também são encontrados vários tipos de plataformas de desenvolvimento para dispositivos Móveis, como o Android, IOS, Symbian, Windows Phone, entre outras.

Fato é, que este projeto será um protótipo desenvolvido para SmartPhone com Sistema Operacional Android, pois esta plataforma possui todas as ferramentas open source além de um crescimento enorme de usuários em relação as outros S.O. Móvel. Com o uso correto, este aplicativo pode se tornar uma ferramenta importante para a economia de cada pessoa que se beneficiar com este produto (<http://campuseiros.com/temas/mobile/o-futuro-e-mobile-e-e-concorrido>).

Figura 1: gráfico SistemaOperacionai Móvel



Fonte: <http://campuseiros.com/temas/mobile/o-futuro-e-mobile-e-e-concorrido>

2 FUNDAMENTAÇÃO TEÓRICA

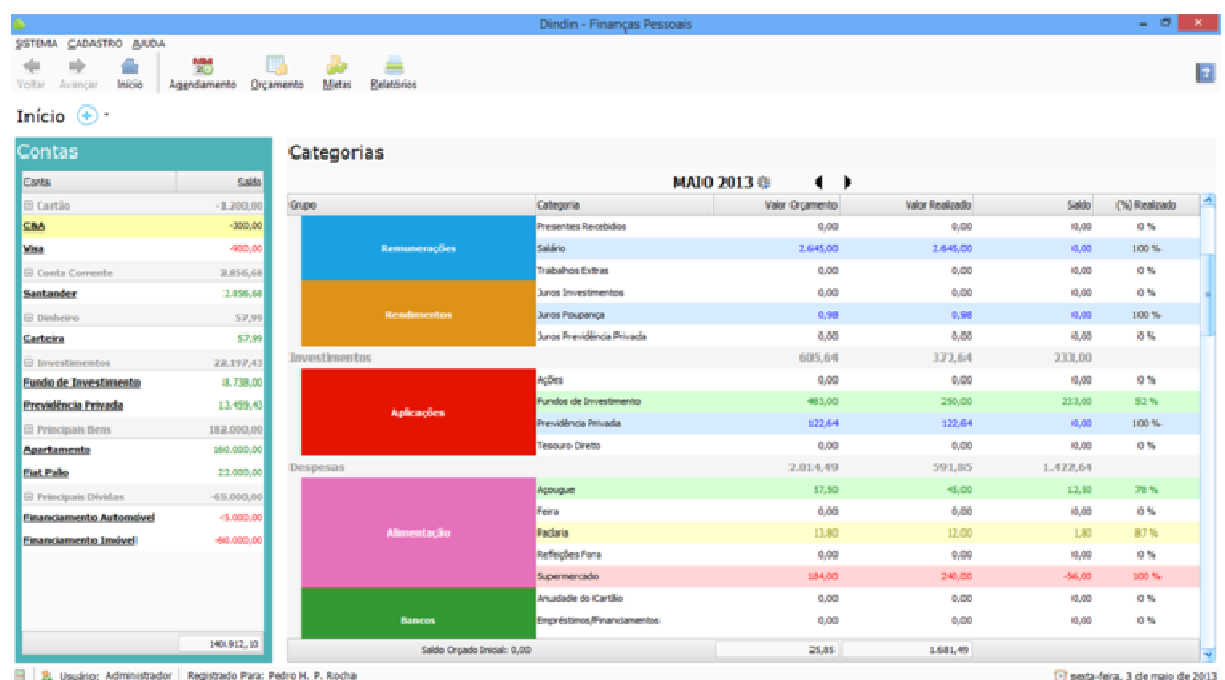
2.1 FINANÇAS PESSOAIS

Finanças pessoais é a ciência que estuda a aplicação de conceitos financeiros nas decisões financeiras de uma pessoa ou família. Em finanças pessoais são considerados os eventos financeiros de cada indivíduo, bem como sua fase de vida para auxiliar no planejamento financeiro (www.colegiomarquesdeolinda.com.br/artigos).

O objetivo principal das finanças pessoais é fazer com que cada pessoa tenha uma situação financeira tranquila e calma, para que se possa sair daquelas situações de riscos que passamos financeiramente quando administramos mal nossas finanças, além de conseguir realizar sonhos maiores como compra uma casa, fazer aquela viagem desejada, um carro ou até abrir seu próprio negócio.

Uma das tarefas mais importantes das finanças pessoais é o acompanhamento de gastos. Consiste em anotar todo o dinheiro recebido e todo o dinheiro gasto. A partir do controle financeiro é possível determinar se as contas estão deficitárias ou superavitárias (<http://cienciascontabeisiesia.blogspot.com.br/2010/09/educacao-financeira.html>).

Figura 2: controle finança pessoal



Fonte: <http://dindin.softonic.com.br/>

2.2 ORÇAMENTO

O orçamento nada mais é um plano financeiro estratégico de uma administração para determinado exercício. Para se alcançar metas pré-estabelecidas

Um orçamento é uma radiografia de seu estado financeiro; ele lhe permite ter um controle de suas finanças e identificar exatamente em que você está gastando seu dinheiro. Dessa maneira, você poderá definir prioridades e fazer as mudanças necessárias.

Para Torres(1995, p.45),“os economistas afirmavam desde que lançaram as bases científicas da economia, como ponto cardeal do orçamento, o equilíbrio financeiro”.

Figura 3: Orçamento

ORÇAMENTO DOMÉSTICO	
Receitas	
Salário	R\$ 2.500,00
Despesas	
Supermercado	R\$ 430,00
Água/luz	R\$ 100,00
Telefone	R\$ 50,00
Celular	R\$ 65,00
Condomínio	R\$ 175,00
Combustível	R\$ 140,00
Escola	R\$ 600,00
Plano de Saúde	R\$ 300,00
Outros	R\$ 300,00
TOTAL DAS DESPESAS	R\$ 2.160,00
SALDO LÍQUIDO	R\$ 340,00

Fonte: http://socialcarisma.blogspot.com.br/2012_05_01_archive.html

2.3 DISPOSITIVOS SIMILARES

Tendo como base no conteúdo acima citado, vi a oportunidade de estar desenvolvendo um aplicativo que tenha uma suma importância no contexto de controle de finanças pessoais com ênfase em sinalizações e alerta de vencimentos, coisas que os aplicativos existentes no mercado quando possuem essas ferramentas, só esta liberada para versão paga, como por exemplo, os aplicativos Finanças Pessoais, Money Wise e Money Lover que as versões gratuitas e pagas podem ser baixadas no Play Store (https://play.google.com/store?hl=pt_BR).

2.3.1 Finanças Pessoais

O aplicativo finanças pessoais funciona off-line, possui versões pagas e não pagas, possui cadastro de capitais que são riqueza adquirida, calendário, relatórios, importa e exporta, a interface é agradável, seu processamento é rápida, mais sua usabilidade é complicada não tem suporte para cliente e suas informações de ajuda não são suficientes, e backup só na versão paga (<https://play.google.com/store/apps>).

Figura 4: Ilustração Finanças Pessoais

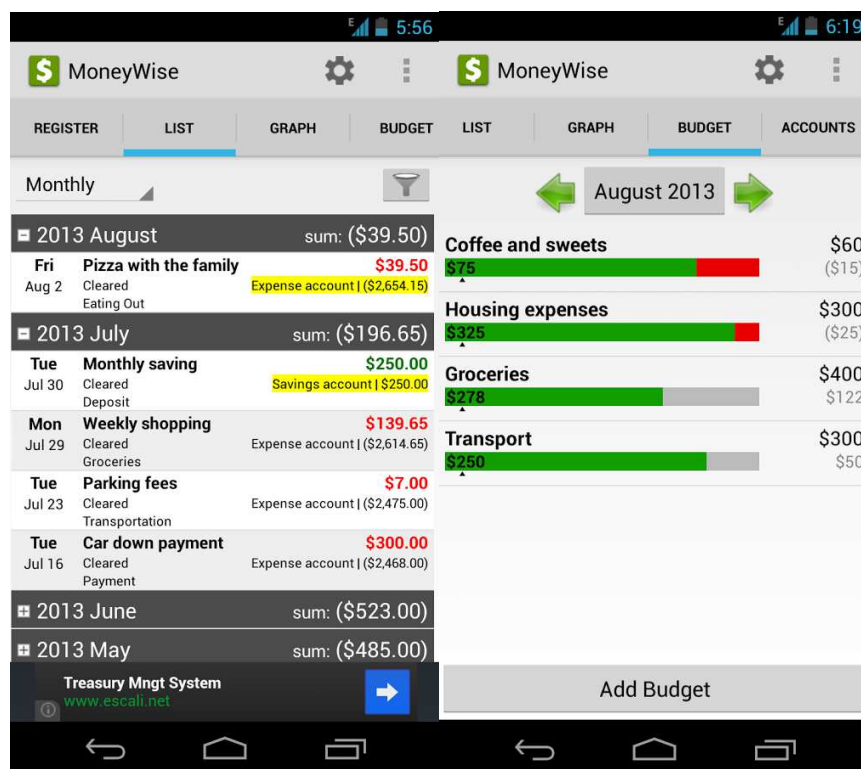


Fonte: <https://play.google.com/store/apps/details?id=incomeincloud.android>

2.3.2 Money Wise

O MoneyWise funciona off-line possui versão pagas e não pagas, sua usabilidade é mais simples e agradável de utilizar, ele registra despesas e ganhos do dia-a-dia e os exibe em forma de relatório e gráficos, possui cadastro de registros, orçamentos, contas e ajuda a controlar o fluxo do caixa, porém não consegue cadastrar categorias novas apenas editar (<https://play.google.com/store/apps>).

Figura 5: Ilustração Money Wise



Fonte: https://play.google.com/store/details?id=com.handynorth.moneywise_free&hl=pt_BR

MoneyWise foi nomeado "O Melhor Aplicativo de Monitoramento de Orçamento para Android" pelo LifeHacker, 2012.

2.3.3 Money Lover

O Money Lover é um aplicativo pessoal simples e funcional com versões paga e gratuita, ele é multilíngue, ele permite adicionar despesas e ganhos em categorias editáveis, organizando o fluxo de dinheiro.

Destacam-se como ponto positivo a ferramenta “gestão de dívidas”, que consegue usar dados da agenda telefônica para registrar credores, e os alertas automáticos de datas porem esta disponível apenas na versão paga. Já como ponto negativo, nota-se a falta de uma ferramenta de estatísticas e meios de pagamentos (<https://play.google.com/store/apps>).

Figura 6: Ilustração Money Lover



Fonte: https://play.google.com/store/apps/details?id=com.bookmark.money&hl=pt_BR

Como vocês podem ver estes aplicativos que possui no mercado, muitas das suas funcionalidades só funcionam nas ferramentas e outras funcionalidades importantes são encontradas separadamente em cada um desses aplicativos e nessa versões gratuita não possuem aleta de vencimentos gratuito que funciona com o aplicativo desligado.

2.4 ANDROID

O Android um sistema operacional móvel baseado em uma versão Linux. Foi desenvolvido pela OpenHandset Alliance, uma aliança entre várias empresas, dentre elas a Google. Como Android é de código aberto e esta disponível gratuitamente para que os fabricantes o personalizem, não há configurações fixas de hardware e software [LEE, 2011].

“A principal vantagem do Android é que ele oferece uma abordagem unificada para desenvolvimento de aplicativos.”

“Internamente suporta as seguintes Funcionalidades:”

- Armazenamento – usa o SQLite uma base de dados relacional leve, para armazenamento de dados;
- Conectividade – suporta WIFI, Bluetooth, GSM e outras;
- Troca de mensagens – suporta tanto SMS quanto MMS;
- Navegador Web – baseado no WebKit de código aberto, juntamente com JavaScript V8 do chrome;
- Suporte a meios – inclui MP3, MP4, MIDI, JPEG, ETC;
- Suporte a hardware – sensor de acelerômetro, câmara bússola digital, sensor de proximidade;
- Multitoques – suporte a telas multitoques;
- Multitarefa – suporte aplicativo multitarefa;
- Suporte flash – o Android 2.3 suporta o flash 10.1;
- Tethering – suporte ao compartimento de conexões de Internet como hotspot com e sem fios.

(LEE, Wei-Meng. Introdução ao Desenvolvimento de Aplicativos para Android. 1. ed. Rio de Janeiro: Editora Ciência Moderna Ltda, 2011).

2.4.1 Arquitetura Android

“A arquitetura Android é dividida em 5 seções em quatro camadas principais:”

- Kernel Linux;
- Bibliotecas;
- Runtime do Android;
- Estrutura de Aplicativos;
- Aplicativos.

(LEE, Wei-Meng. Introdução ao Desenvolvimento de Aplicativos para Android. 1. ed. Rio de Janeiro: Editora Ciência Moderna Ltda, 2011).

Figura 7: Arquitetura do Android



Fonte: <http://developer.android.com/about/versions/index.html>

2.4.2 Runtime do android

Runtime reúne varias bibliotecas centralizadas que permite o desenvolvimento de aplicativos para Android, usando linguagem Java, ele também possui uma máquina virtual chamada Dalvik para que cada aplicativo rode em seu próprio processo, com sua própria estância, esta máquina foi projetada especialmente para o Android [LEE, 2011].

2.4.3 Android SDK

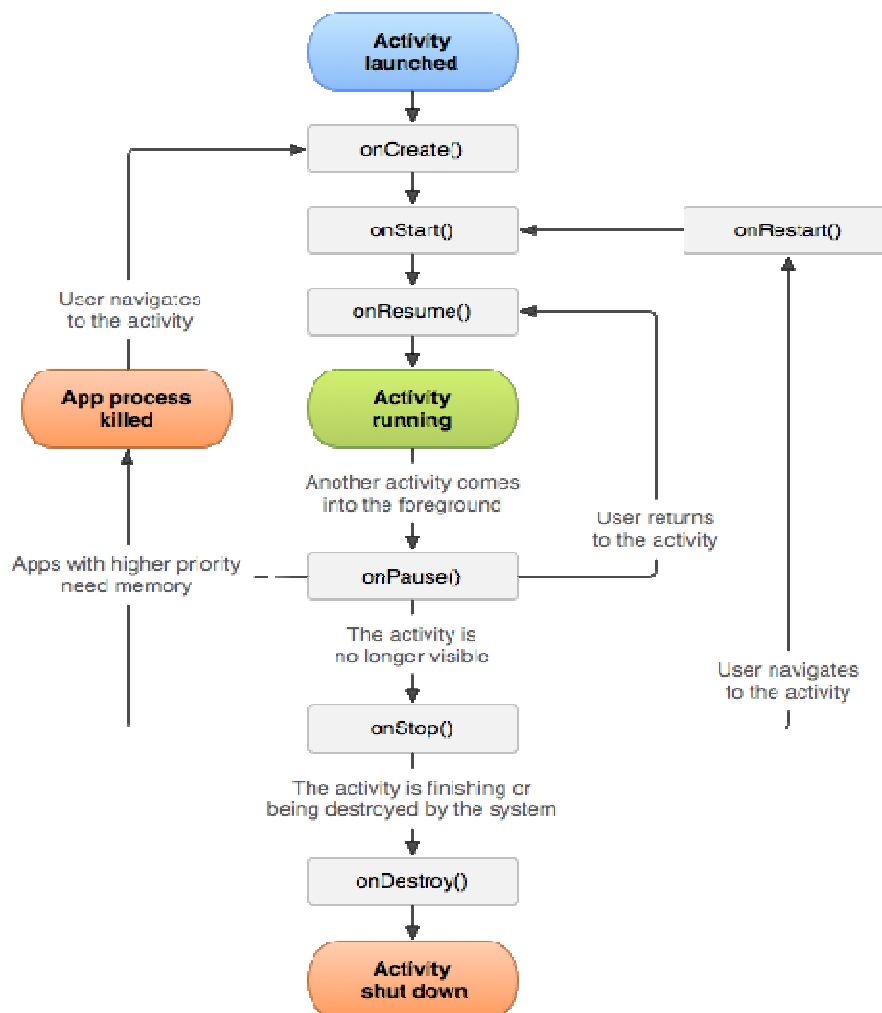
O Android SDK fornece as bibliotecas da API e ferramentas de desenvolvimento necessárias para construir, testar e depurar aplicativos para Android segundo LEE(2011, p.9).

2.4.4 Activity

Todo Aplicativo Android começa por uma activity, Para LEE(2011,p.31), activity ou atividade é uma janela que contém a interface de usuário de seus aplicativos. Um aplicativo pode ter zero ou mais atividades. Tipicamente, os aplicativos têm uma ou mais atividades, e o objetivo principal de uma atividade é interagir com o usuário.

As atividades passam por vários estágios durante sua execução, esse evento é conhecido como ciclo de vida de uma atividade.

Figura8: Ciclo de vida



Fonte: <http://developer.android.com/about/versions/index.html>

2.4.5 Persistência de Dados

“Persistência de dados é muito importante, para desenvolvimento Móvel, quando o usuário precisar usar esses dados que estão salvos. No Android temos varias maneira de fazer isso:”

- **SharedPreferences** – Armazenamento de dados primitivos provados em pares chave-valor;
- **InternalStorage** – Armazenamento de dados primitivos na memória interna do dispositivo movel;
- **ExternalStorage** - Armazenamento de dados primitivos no cartão de memória;
- **Network Connection** – Armazenam na Web ou seu servidor de rede.
- **SQLite** – gerencia base de dados relacionais.

(LEE, Wei-Meng. Introdução ao Desenvolvimento de Aplicativos para Android. 1. ed. Rio de Janeiro: Editora Ciência Moderna Ltda, 2011).

3 METODOLOGIA

Este projeto esta sendo desenvolvido para que se possa construir um protótipo que poderá ser executado em um smartphones que possua plataforma android 2.2 ou superior, esse projeto se faz uso de tecnologia APIsAndroid em nível 8, metodologias que constituirão o projeto do protótipo, o *AndroidDeveloper* usando pacote de ícones e menus para desenvolvimento do protótipo, *toolsplugin* (ADT) para o eclipse, fornecendo um ambiente de desenvolvimento de nível profissional para a construção de aplicativos Android. É uma completa Ide Java com recursos avançados para ajudar a construir, testar depurações de pacotes de seus aplicativos android com a persistencia de dados usando o SQLite que é o banco de dados nativo do android e se necessário SharedPreferences – armazenamento de dados primitivos provados em pares chave-valor, xml que é plenamente usado para fazer a interface do protótipo. E também o início deste projeto foi feito pesquisas em livros de contabilidade e de gerenciamento de software, para se obter um conhecimento sobre as regras de negócios e suas principais características. Foi efetuado um levantamento dos principais aplicativos existente no mercado, suas possíveis vantagens e desvantagens e suas funcionalidades, usaremos UML para modelar e especificar os requisitos que serão necessários para a construção deste aplicativo móvel em conjunto com modelo de desenvolvimento de software RUP (rational unified process).

3.1 SQLITE

SQLite é um mini SGBD, é uma ferramenta mais precisamente, uma biblioteca que permite com que desenvolvedores possam armazenar os dados de suas aplicações em tabelas e manipular esses dados através de comandos SQL conhecidos de outros bancos de dados.

Para LEE(2011, p.229), SQLite uma base de dados relacional do Android, quando esta base de dados é criada para um aplicativo ele só pode ser usado por ele mesmo. A base de dados SQLite ela é sempre armazenada na pasta /data/<nome do pacote>/databases.

“Algumas Características do SQLite:”

- Software gratuito, multiplataforma, desenvolvido em C padrão (ANSI);
- Todo o banco de dados é guardado localmente (junto com a aplicação), em um único arquivo que possui a extensão db;
- A base de dados pode ter tamanho superior a 2terabytes;
- Não necessita de instalação, configuração ou de administração;
- Suporta o uso de transações (COMMIT / ROLLBACK);
- Não oferece integridade referencial (chaves estrangeiras).
- Principais aplicações: programas locais, sites web, substituto de banco de dados em aulas ou demonstrações, substitui arquivos texto ou arquivos proprietários.

- Não deve ser usado nos seguintes casos: aplicações de alta concorrência e sistemas ou aplicações web de porte muito grande.
(LEE, Wei-Meng. Introdução ao Desenvolvimento de Aplicativos para Android. 1. ed. Rio de Janeiro: Editora Ciência Moderna Ltda, 2011).

3.2 RUP (RATIONAL UNIFIED PROCESS)

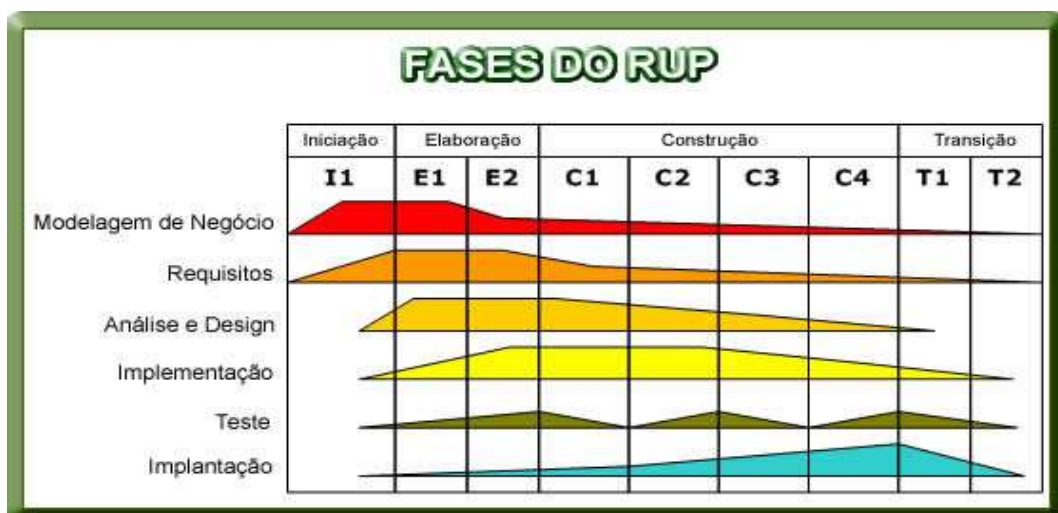
Processo Unificado Rational foi criado pela Rational Software, é um processo de engenharia de software criado pra ajudar no desenvolvimento orientado objeto, de forma sistemática para se ter vantagem usando UML.

O Principal objetivo é desenvolver um software de alta qualidade cumprindo um cronograma, ele define detalhadamente cada parte das tarefas a ser realizada informando responsabilidades, para que as metas sejam alcançadas pelos desenvolvedores no método iterativo e incremental.

Este Processo é feito para grandes equipes de desenvolvimento, porém por ser amplamente customizável, é possível que possa ser usado em projetos de qualquer tamanho.

O RUP organiza o desenvolvimento de software em quatro fases, onde são tratadas questões sobre planejamento, levantamento de requisitos, análise, implementação, teste e implantação do software (<http://www.infoescola.com/engenharia-de-software/rup/>).

Figura 9: Fase do RUP



Fonte: <http://www.infoescola.com/engenharia-de-software/rup/>

“Fase do RUP:”

- Iniciação: É feito a levantamento de requisitos, calculo de riscos, custos e prazo, contato com cliente e planejamento, analisar tudo e fazer reunião para ver se software vai ter continuidade ênfase no escopo do sistema.
- Elaboração: Abrange modelagem, revisando risco, custo e se o projeto é confiável, ênfase na arquitetura.
- Construção: é parte que acontece codificação, construção do desenvolvimento do software, ênfase no desenvolvimento.
- Transição: é a entrega do software e fase de teste treinamento do usuário, ênfase na implantação.

(LEE, Wei-Meng. Introdução ao Desenvolvimento de Aplicativos para Android. 1. ed. Rio de Janeiro: Editora Ciência Moderna Ltda, 2011).

3.3 ASTAH COMMUNITY

Astah, muito conhecido anteriormente como JUDE (*Java e UmlDevelopers*), é uma ferramenta de modelagem UML (*UnifiedModelngLanguage*) criado pela Vision empresa japonesa pelo Instituto Tecnologia da Informação, a base do AstahCommunity foram concebidos para ser fácil de usar, as seguintes funcionalidades estão agregadas na ferramenta comunitária, esta ferramenta nos auxilia para desenvolvermos da diagramação e documentação do software a ser criado (<http://astah.net/editions/community>).

3.4 REQUISITOS

Este protótipo ira rodar em Smartphone com sistema operacional Android que poderá ser executado na versão 2.2 com API nível 8, com armazenamento de dados no SQLite, irá trabalhar off-line e gratuito com todas sua funcionalidades.

O usuário vai abrir o aplicativo, se for a primeira vez ele vai cadastrar um usuário e senha, na primeira tela aparece o menu principal com uma lista de opções, na opção Registro o usuário encontrará uma tela de cadastro de registros onde o mesmo possa cadastrar seus gasto, compra, recebimento e outras ações que queira, na opção conta, o usuário irá cadastrar suas contas se desejar com saldo atual, no menu orçamento ele vai cadastrar um novo orçamento para aquele mês com aviso de quando este orçamento esta para ser estourado, na opção lista o usuário vai visualizar as contas pagas e não pagas, as receitas recebidas ou pendentes, as compras parcelas verificar quantas parcelas faltam, ver lista de despesas cadastradas com alerta de vencimentos, podendo observar onde esta gastando mais, se precisa

economizar, o usuário ainda pode entrar na opção Categorias e cadastrar ou editar categorias, na opção ajuda o cliente vai ter acesso um texto de ajuda informando como funciona o aplicativo com se fosse um manual do usuário, com essas funcionalidades a pessoa que estiver usando este aplicativo ficará sempre atualizado sobre sua situação financeira.

Requisitos são necessidades que os desenvolvedores e conjunto com o usuário precisam resolver para se obter um software adequado que atinja seu objetivo.

Para Bezerra (2007, p.22), O principal objetivo de Levantamento de requisitos é que usuários e desenvolvedores, juntamente com os clientes, tentam levantar e definir as necessidades dos futuros usuários do sistema a ser desenvolvido.

Os requisitos podem ser classificados em duas grandes Categorias: Requisitos funcionais que define as funcionalidades do sistema, ou seja, o que o sistema vai fazer e Requisitos não-funcionais que declara as características, restrições relacionada as suas funcionalidades.

3.4.1 Requisitos Funcionais

- Manter registro;
- Visualizar lista de contas a pagar e receber;
- Gerar alerta de vencimento;
- Manter categorias;
- Manter orçamento;
- Manter Contas;

Requisitos funcionais descrevem as funcionalidades que o aplicativo vai ser capaz de fazer, declaração de serviços que o software deve apresentar, como o sistema deve reagir e a se comportar em determinadas situações.

Para Wazlawick(2004, p.38), requisitos funcionais corresponde a listagem de tudo que o sistema deve fazer.

3.4.2 Requisitos Não Funcionais

- Segurança login e senha;
- Portabilidade Software em plataforma Java;
- Portabilidade Sistema Operacional Android;

- Desempenho as sistema de base e dados SQLite.

Definem as propriedades do sistema e restrições, como espaço em disco e tempo de resposta, restrições como tamanho dos dispositivos de entrada e saída, além de qual linguagem o aplicativo vai ser desenvolvido.

Para Wazlawick(2004, p.38), requisitos não-funcionais são restrições colocadas sobre como o sistema deve realizar seus requisitos funcionais.

3.5 CASOS DE USO

Descreve as principais funcionalidades, onde cada uma delas mostra a ligação do sistema com atores e agentes externos, para Bezerra (2007, p.54), um caso de uso representa um relato de uso de certa funcionalidade do sistema em questão, sem revelar a estrutura e o comportamento internos desse sistema.

3.5.1 Diagramas de casos de uso

Figura 10: Caso de uso



Fonte: autoria Própria

3.5.2 Descrição dos casos de uso

MANTER REGISTRO

Objetivo: o usuário vai cadastrar, editar, listar excluir campos na tela registros no software.

Ator: usuário.

Cenário Principal

- 1- O usuário vai informar usuário e senha.
- 2- O usuário seleciona no menu principal a opção Registro.
- 3- O usuário informa data da operação ou vencimento.
- 4- O usuário informa de qual conta vai fazer a operação.
- 5- O usuário informa a categoria do registro.
- 6- O usuário informa descrição do registro.
- 7- O usuário informa valor do registro.
- 8- O usuário seleciona informação do registro tipo de registro (receita ou despesa).
- 9- O usuário informa forma de pagamento (parcelada ou simples).
- 10- O usuário salva as informações.

Cenários alternativos

Categoria não cadastrada

- 5.a. se não existir categoria, voltar a página principal e escolher opção categorias.

MANTER CATEGORIAS

Objetivo: o usuário vai cadastrar, excluir, editar e listar suas categorias preferidas no software.

Ator: usuário.

Cenário Principal

- 1- O usuário seleciona no menu principal a opção categoria.
- 2- O usuário visualiza as categorias existentes já no software e escolhe se vai adicionar, editar ou excluir categorias.
- 3- O usuário seleciona opção que desejada.
- 4- O usuário adiciona nova categoria.
- 5- O usuário salva informações.

MANTER ORÇAMENTO

Objetivo: o usuário vai cadastrar, excluir, editar e listar seus orçamentos.

Ator: usuário.

Cenário Principal

- 1- O usuário seleciona no menu principal opção orçamento.
- 2- O usuário informa se vai adicionar, editar ou excluir orçamento.
- 3- O usuário seleciona o mês desejado.
- 4- O usuário informa o valor do orçamento.
- 5- O usuário salva informação.

MANTER CONTA

Objetivo: o usuário vai cadastrar, excluir, editar e listar suas contas bancárias no software.

Ator: usuário.

Cenário Principal

- 1- O usuário seleciona no menu principal a opção conta.
- 2- O usuário informa se vai adicionar, editar ou excluir a conta.
- 3- O usuário informa o nome do banco.
- 4- O usuário informa valor.
- 5- O usuário salva informações.

GERAR LISTA DE REGISTROS CADASTRADOS

Objetivo: o sistema mostra lista de registros cadastrados.

Ator: Sistema.

Cenário Gerar lista de registros

- 1- O sistema mostra despesas e receitas cadastradas.

GERAR ALERTA DE VENCIMENTOS

Objetivo: o sistema verifica o vencimento de cada conta a pagar e gera um alarme no celular no dia do vencimento com informações da conta vencida.

Ator: Sistema.

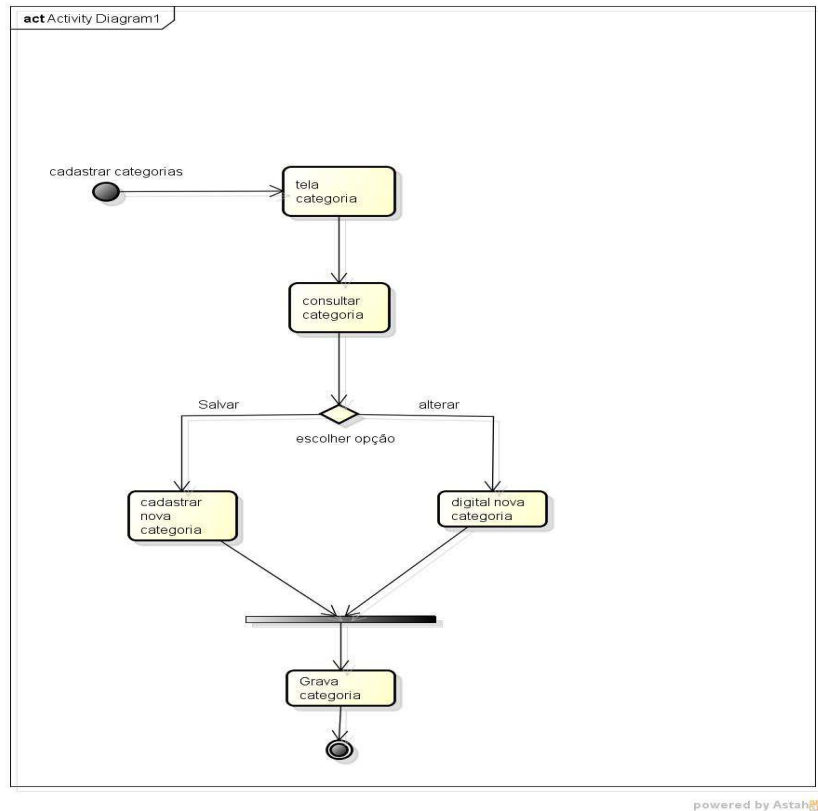
Cenário Alerta de Vencimentos

1- O sistema verifica se tem alguma conta vencida e informa o usuário com um alarme mostrando informações sobre esta conta.

É o diagrama que informa as atividades de um processo interno, ou seja, o que o sistema vai fazer em determinada operação internamente.

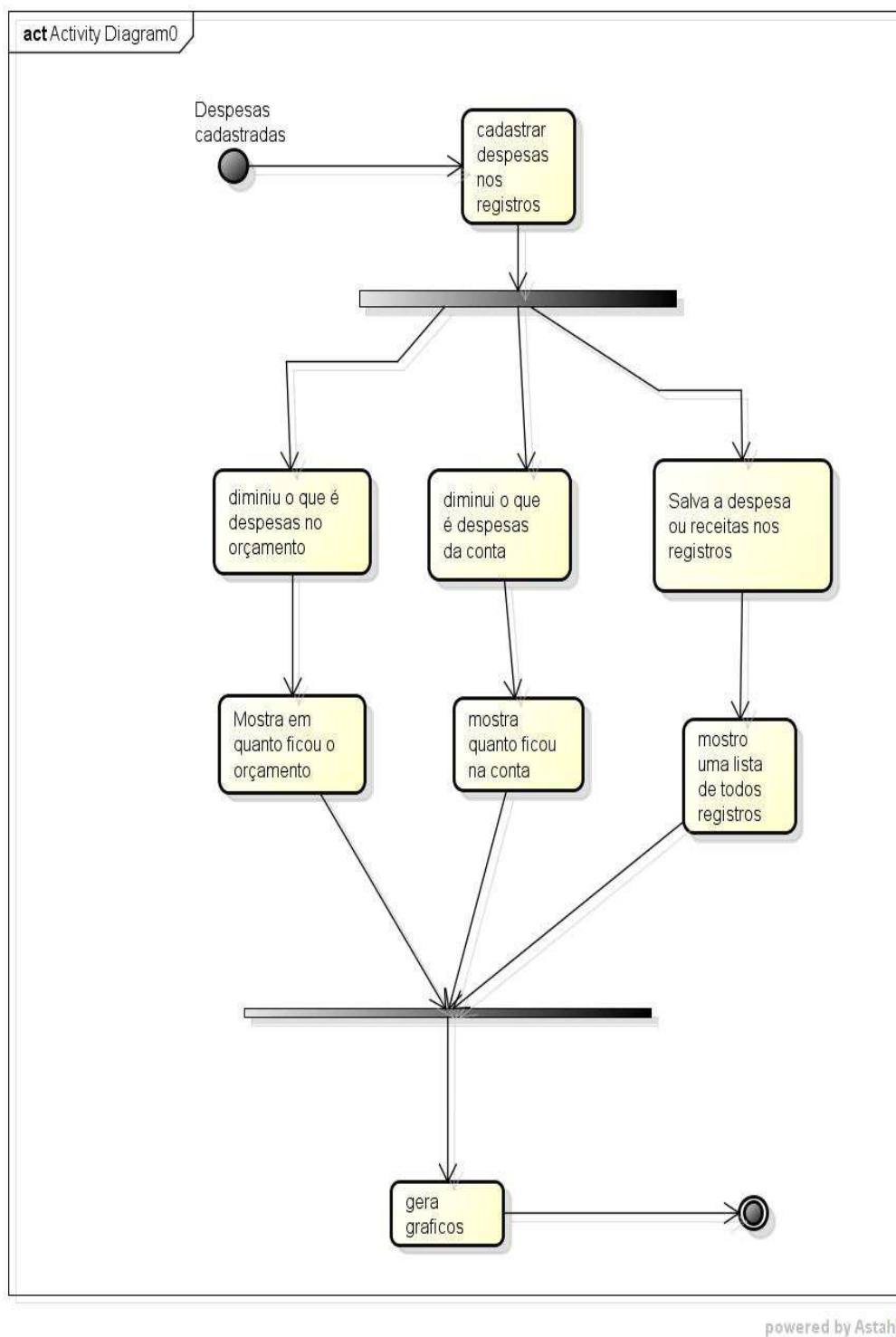
Para Melo (2010, p.199), este diagrama tem por propósito focalizar um fluxo de atividades que ocorrem internamente em um processo, dentro de um período de tempo.

Figura 12: Diagrama de atividades 1



Fonte: autoria própria

Figura 13: Diagrama de Atividades 2



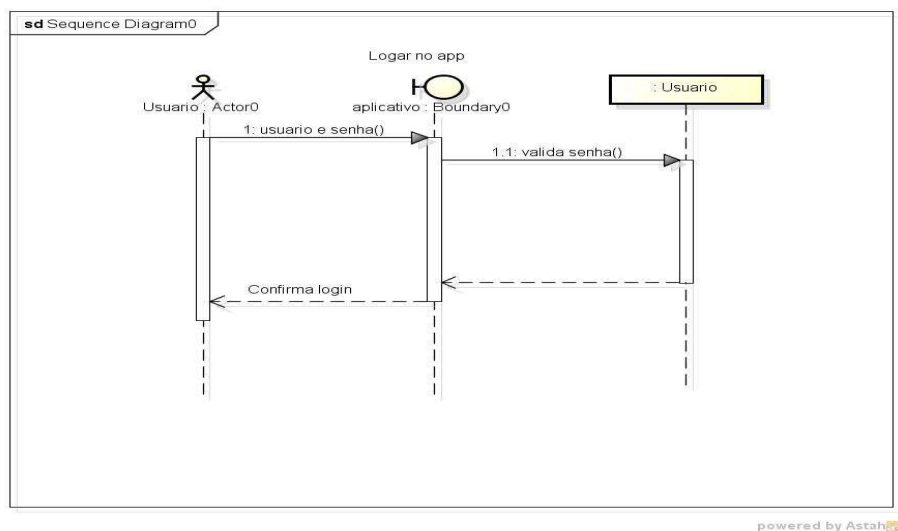
Fonte: autoria própria

3.8 DAGRAMA DE SEGUENCIA

Responsável por informar qual a sequência que cada função tem para cumprir determinada ação no software é uma das formas de diagrama de interação.

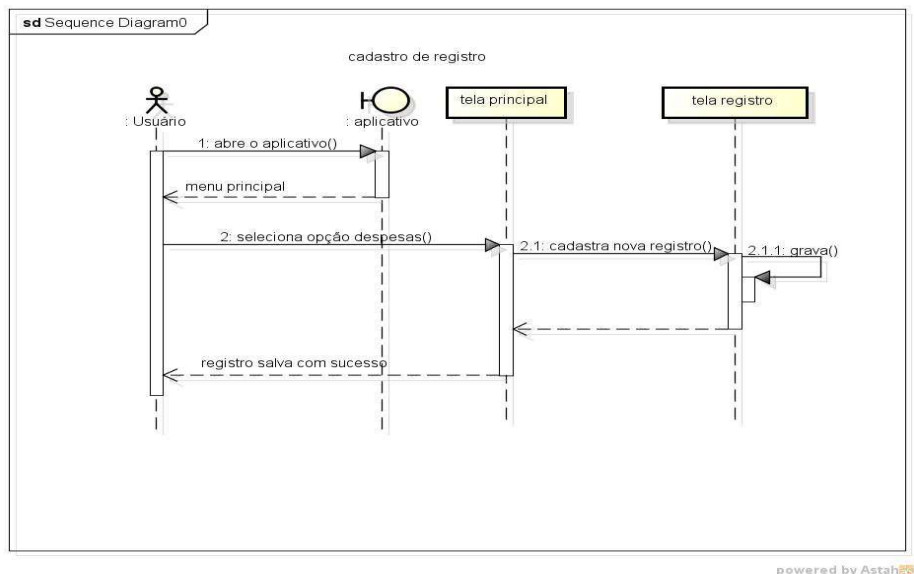
Para Melo(2010, p.156), Interação corresponde a um conjunto de mensagens trocadas entre objetos, com o objetivo de alcançar determinado propósito, respeitando-se o contexto do sistema.

Figura 14: Diagrama de sequência 1



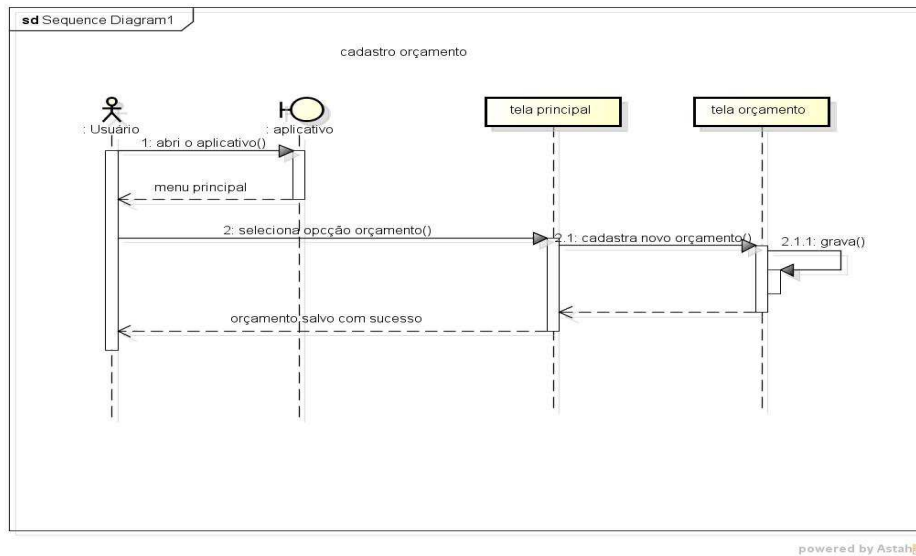
Fonte: autoria própria

Figura 15: Diagrama de sequência 2



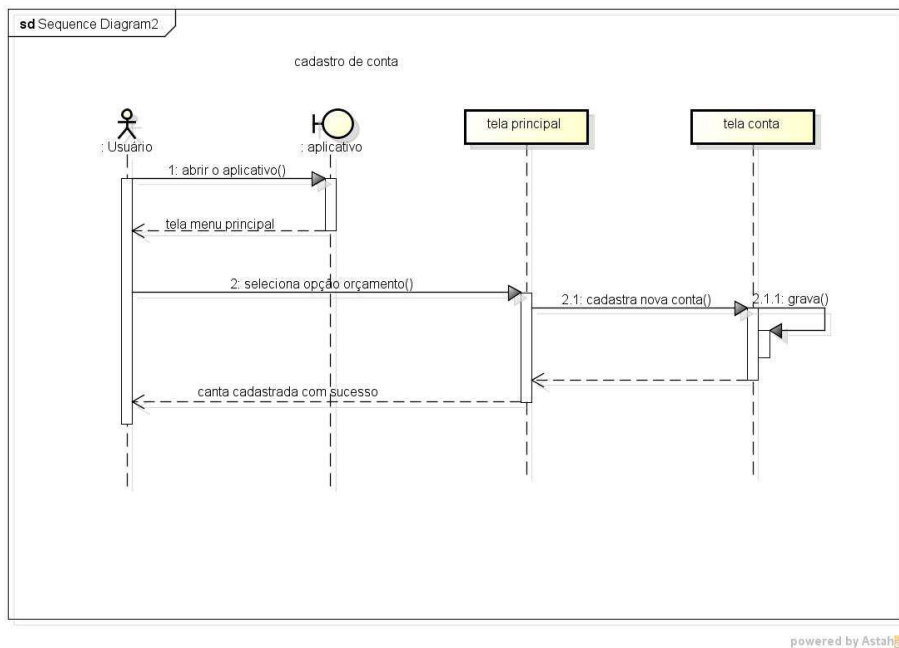
Fonte: autoria própria

Figura 16: Diagrama de sequência 3

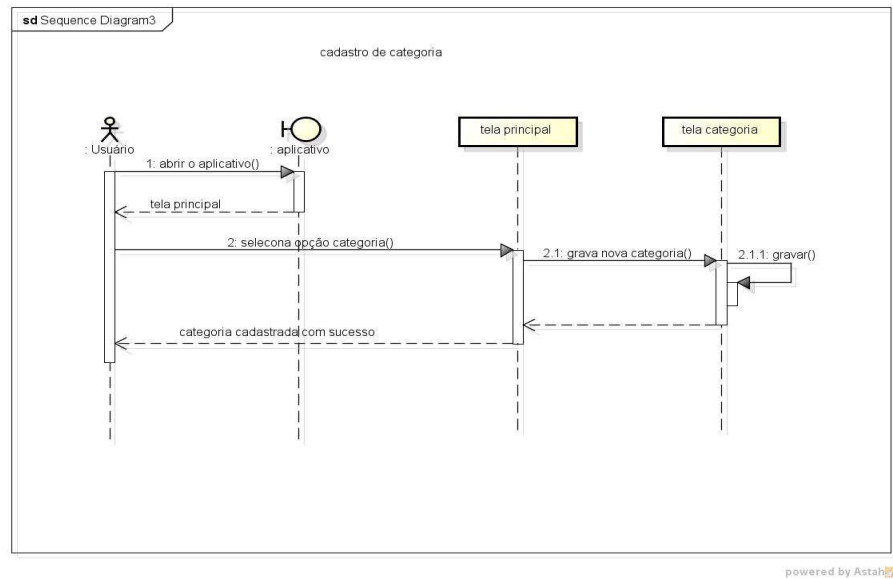


Fonte: autoria própria

Figura 17: Diagrama de sequência 4



Fonte: autoria própria

Figura 18: Diagrama de sequência 5

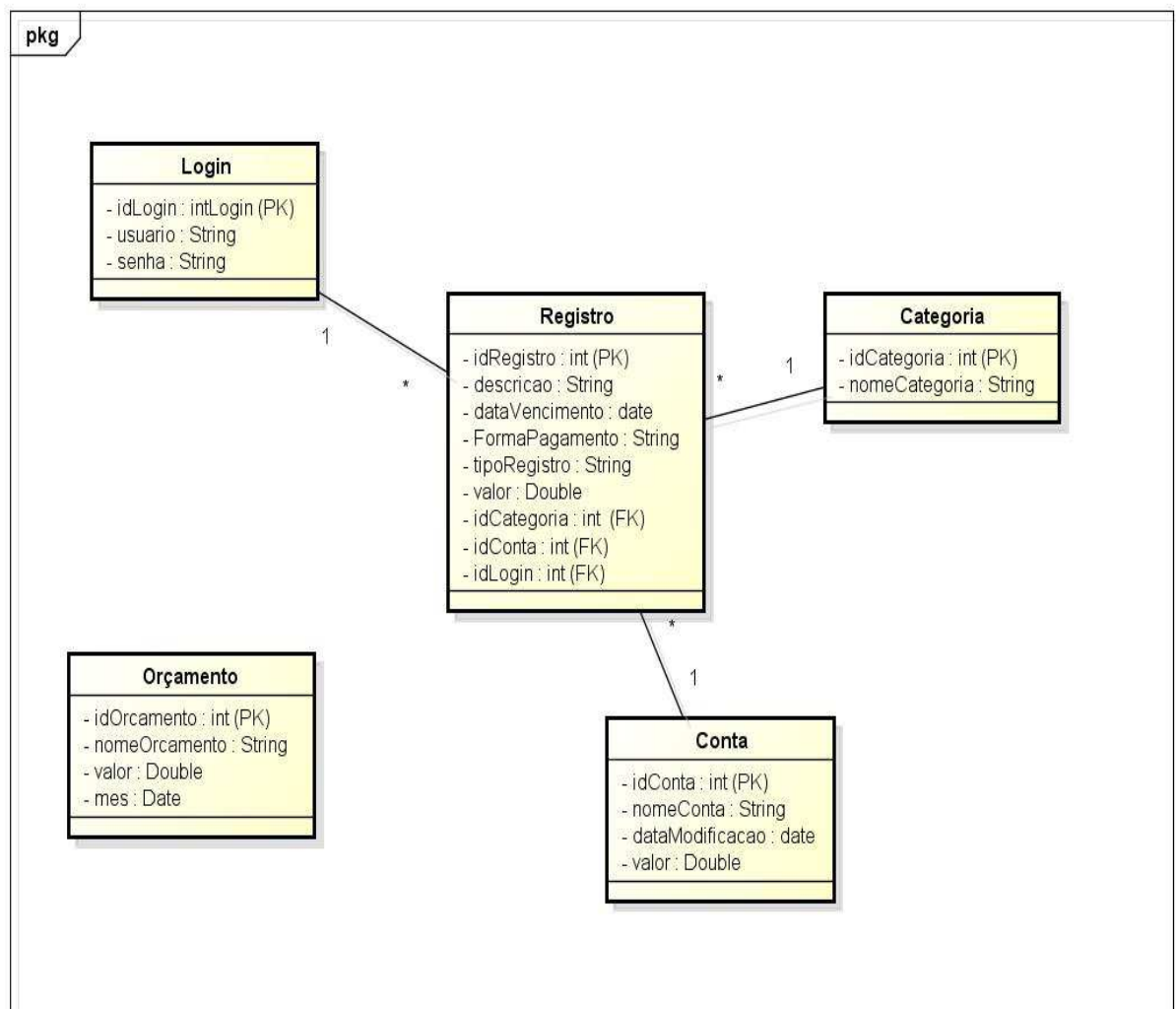
Fonte: autoria própria

3.9 BANCO DE DADOS

Este aplicativo móvel necessita de um banco de dados para armazenar na tabela Registro os campos descrição da despesa, categoria valor, status e tipo de registro, para mostrar na lista de registro, também salvar as contas cadastradas na tabela contas e também os campos da tabela Orçamento além da tabela categorias com seus respectivos campos onde PK significa chave primária e FK chave estrangeira.

3.9.1 Diagrama Entidade Relacionamento

Figura 19: DER



Essas estrutura de banco de dados serão criadas no aplicativo usando o SQLite que é um recurso que o Android proporciona para salvar informações do aplicativo a ser desenvolvido, esta persistência de dados é estruturada.

SQLite é uma biblioteca de software que implementa um autossuficiente , sem servidor , com configuração zero , transacional de banco de dados SQL. SQLite é o mais amplamente implantado motor de banco de dados SQL no mundo. O código fonte para SQLite está no domínio público (<http://www.sqlite.org>).

4 PROJETO DO SISTEMA

4.1 ESCOPO

O protótipo terá que ter função de controlar finanças pessoais, aplicando esta ferramenta para auxiliar o usuário em suas decisões financeiras, como ficar sempre em dia com suas contas a pagar e receber, também é importante sempre que tiver um gasto cadastrar no software e acompanhar com esta seu orçamento do mês e sua conta no banco com isso identificar onde esta gastando mais para começar a economizar e poupar dinheiro.

4.2 REGRAS DE NEGÓCIO

Quadro 1: Regras De Negócio

Requisitos	Descrição
RG001	O usuário terá que cadastrar um usuário para acessar o aplicativo.
RG002	O número máximo de registro de despesas é igual a 60.
RG003	O usuário é obrigado a escolher a categoria de cada despesa.
RG004	O usuário é obrigado a informar o valor da despesa ou receita
RG005	Todos os campos do menu despesas deveram ser preenchidos.
RG006	O usuário quando criar um orçamento tem que informar valor
RG007	Se orçamento estiver perto de estourar deve mandar um aviso.
RG008	Se despesa estiver pendente na data cadastrada deve gerar uma aviso o vencimento da despesa

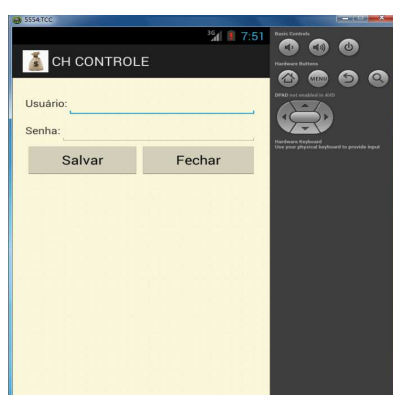
Fonte: Autoria Própria

4.3 INTERFACE GRÁFICA

O protótipo da aplicação é composto por oito telas sendo que todas as telas estão representadas por uma única atividade (*Activity*) cada.

A primeira tela a ser exibida é a tela login, na primeira vez que em que for executar o aplicativo precisa ser salvo um usuário com nome e senha, isto é uma questão de segurança nesta tela ainda possui botão de salvar e fechar.

Figura 20: Tela Login



Fonte: autoria própria

A segunda tela tem o menu principal onde o usuário consegue escolher a opção desejada como: registro, lista, conta, orçamento, categorias e configurações através de um *ImageButton*.

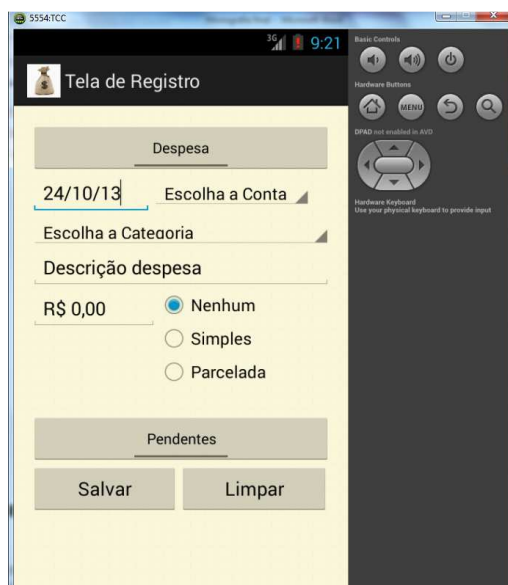
Figura 21: Tela Principal



Fonte: autoria própria

O usuário quando acessar o menu registro primeiramente um botãoToggleButton, depois escolher a data de vencimento em um data picker, escolher a conta em uma Spinner, a categoria do registro em outro Spinner, informar a descrição em um TextView, o valor em um TextView do tipo decimal, seguida se o tipo do registro se for simples ou parcelada em uma RadioGroup. Outro ToggleButton conta se esta pendente ou paga, por último um botão salvar e um botão para limpar.

Figura 22: Tela Registro



Fonte: autoria própria

5 PROJETO DE DESENVOLVIMENTO

5.1 ROTEIRO DE INSTALAÇÃO

Para o desenvolvimento do protótipo é necessário Fazer uso da ferramenta oferecida pela Google, que é livre, de código aberto, e é executado na maioria dos OS das plataformas principais e configurar o ambiente de desenvolvimento JDK e o SDK. O Android SDK fornece-lhe as bibliotecas da API e ferramentas de desenvolvimento necessárias para construir, testar e depurar aplicativos para o Android.

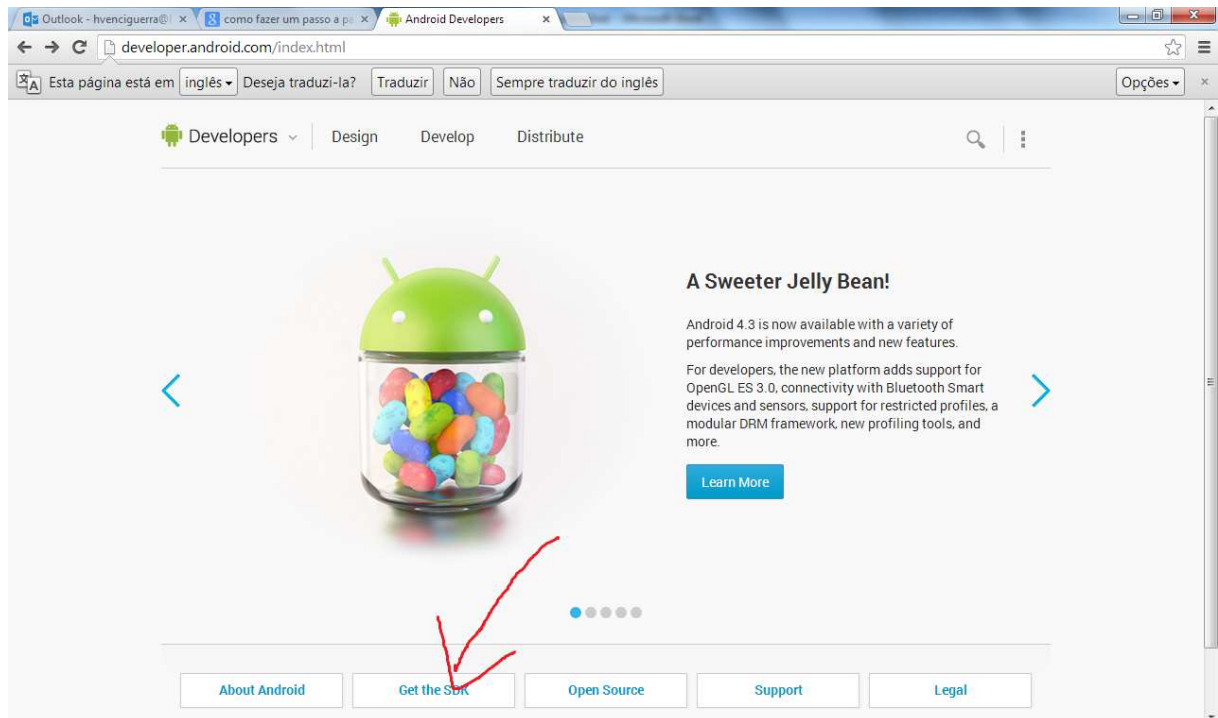
Iremos baixar o ADT (AndroidDeveloper Tools)[9].

Ele inclui os componentes essenciais para o início da programação de aplicativos Android nessa ferramenta vem incluindo:

- Eclipse + ADT Plugin
- Android SDK Tools;
- Plataforma Android-ferramentas;
- A mais recente plataforma Android;
- A imagem mais recente do sistema Android para o emulador.

Instalando as ferramentas de desenvolvimento de aplicativo android, primeiramente iremos precisar baixar o AndroidDeveloper Tools. Que fica no endereço <http://developer.android.com/index.html>, clicar na opção GET THE SDK, conforme a ilustração a baixo.

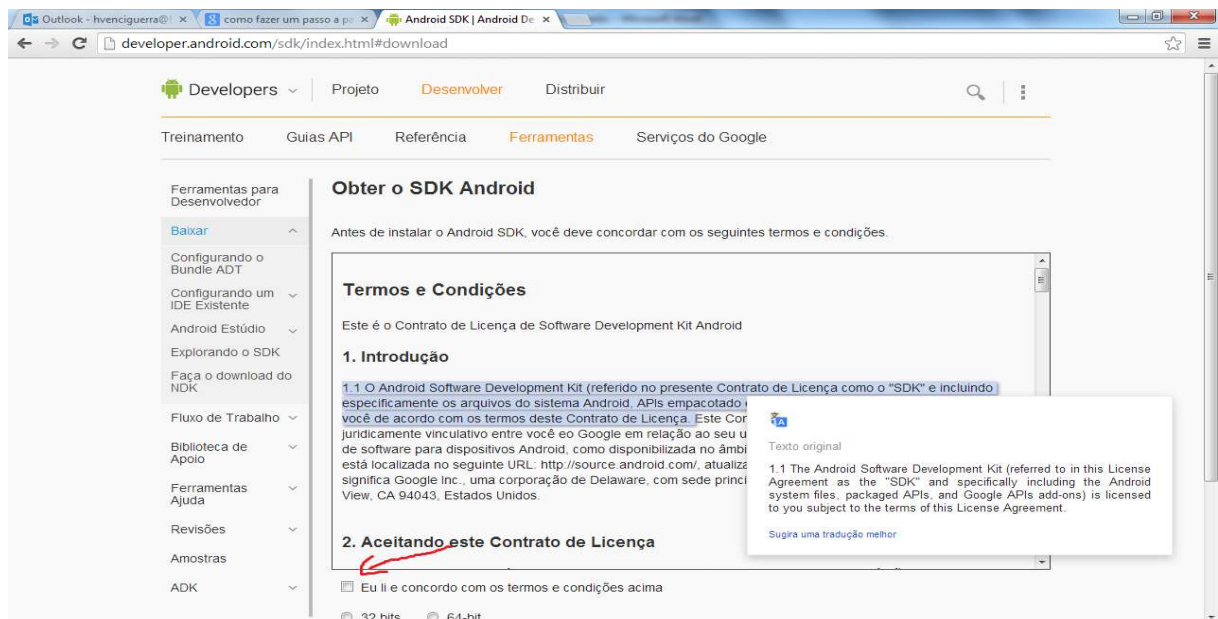
Figura 23: Index Developer



Fonte: <http://developer.android.com/about/versions/index.html>

Após escolhido a esta opção favor concordar com a licença e selecionar o a configuração do seu sistema operacional que vai ser usado.

Figura 24: Licença de uso do Android SDK



Fonte: <http://developer.android.com/about/versions/index.html>

Para que a ferramenta de desenvolvimento funcione, é necessário que você tenha instalado antes de tudo, a Máquina Virtual Java (de preferência a versão 7).

Para saber se você possui uma Máquina virtual Java, entre *noprompt* de comando de digite a seguinte linha: `Java -version`. Se mostrar algo parecido como:

- Java version "1.7.0_03";
- Java(TM) SE Runtime Environment (build 1.7.0_03-b05);
- Java HotSpot(TM) Client VM (build 22.1-b02, mixed mode, sharing).

Você possui uma máquina virtual Java instalada no seu computador, caso contrário, instale o JDK. Você pode fazer o download do JDK [17].

Procure saber como configurar as variáveis de ambiente do Windows ou outro sistema operacional que for usado.

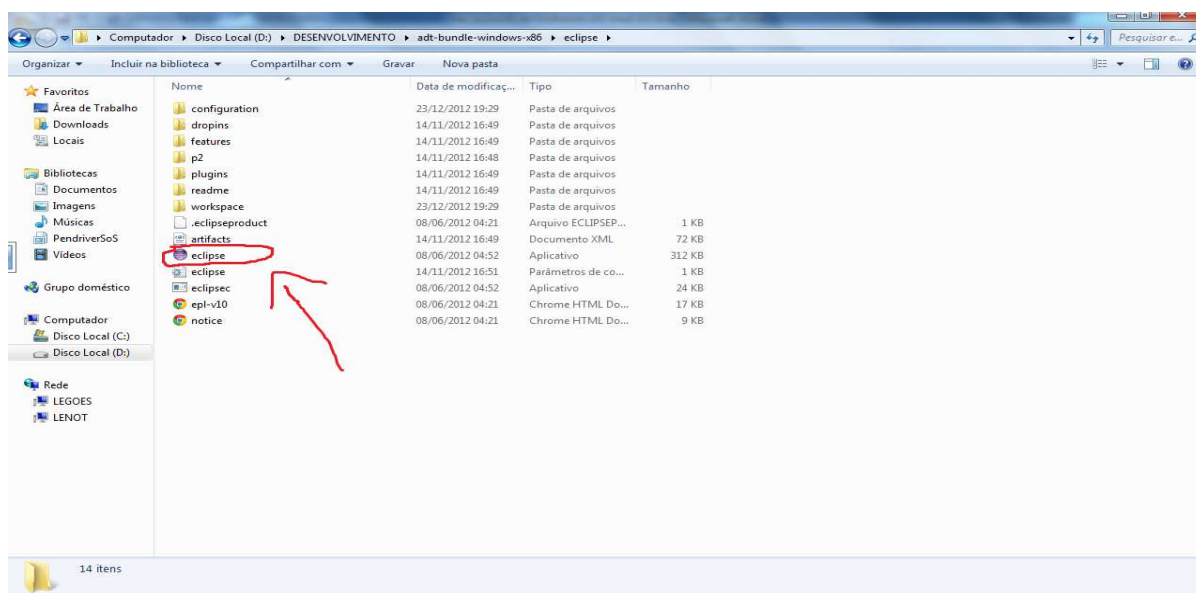
Após baixar de obter os programas eu recomendo que você descompacte o `adt-bundle-windows-x86` na unidade D: em seu computador

Após descompactar o arquivo na unidade (D:), vai ser criada uma pasta chamada `adt-bundle-windows-x86`, abra essa pasta.

Dentro da pasta de abrir pasta `adt-bundle-windows-x86` você encontrará outras duas pasta, uma com o eclipse e a outra com o SDK do Android. Abra a pasta eclipse.

Dentro da pasta eclipse você irá clicar duas vezes no ícone do aplicativo da ferramenta do eclipse.

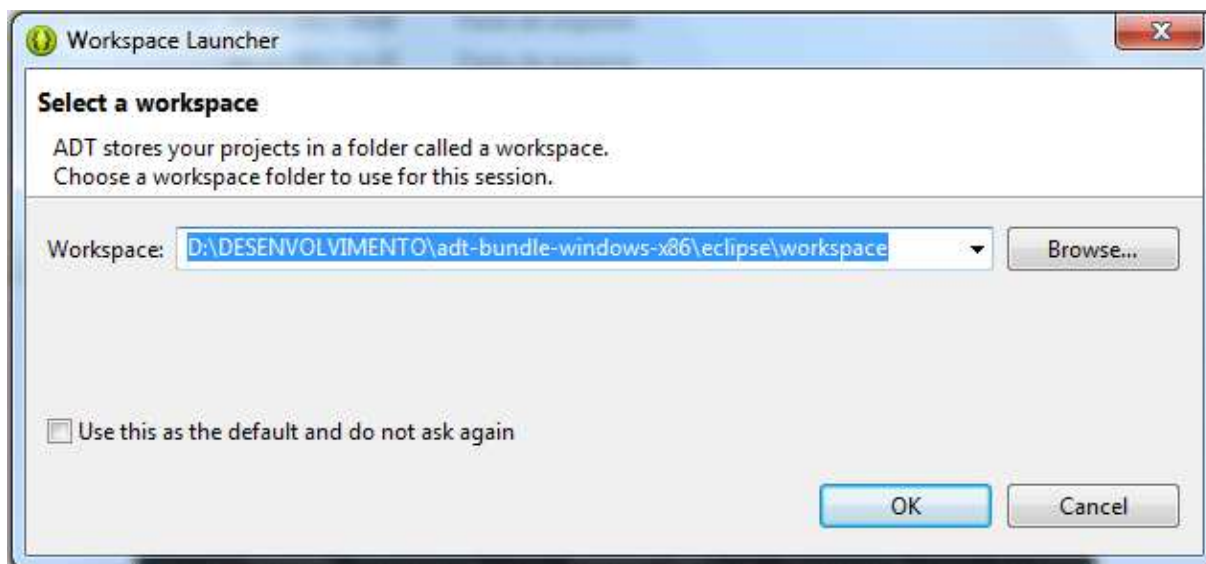
Figura 25: Ícone Eclipse



Fonte: eclipse

Troque o caminho padrão da workspace, use o caminho a onde se encontra o aplicativo do eclipse, assim você estará salvando todos os seus projetos junto com a instalação do eclipse.

Figura26: CaminhoWorkspace



Fonte: eclipse

Seu aplicativo AndroidDeveloper Tools irá iniciar trazendo com sigo todas as ferramentas necessárias para o desenvolvimento de aplicativos com base na plataforma Android, como é a nossa primeira inicialização, algumas configurações de atualizações deverão ser feita antes de começar a programar nosso aplicativo, tais como, Android SDK Manager e o Android Virtual Device Manager.

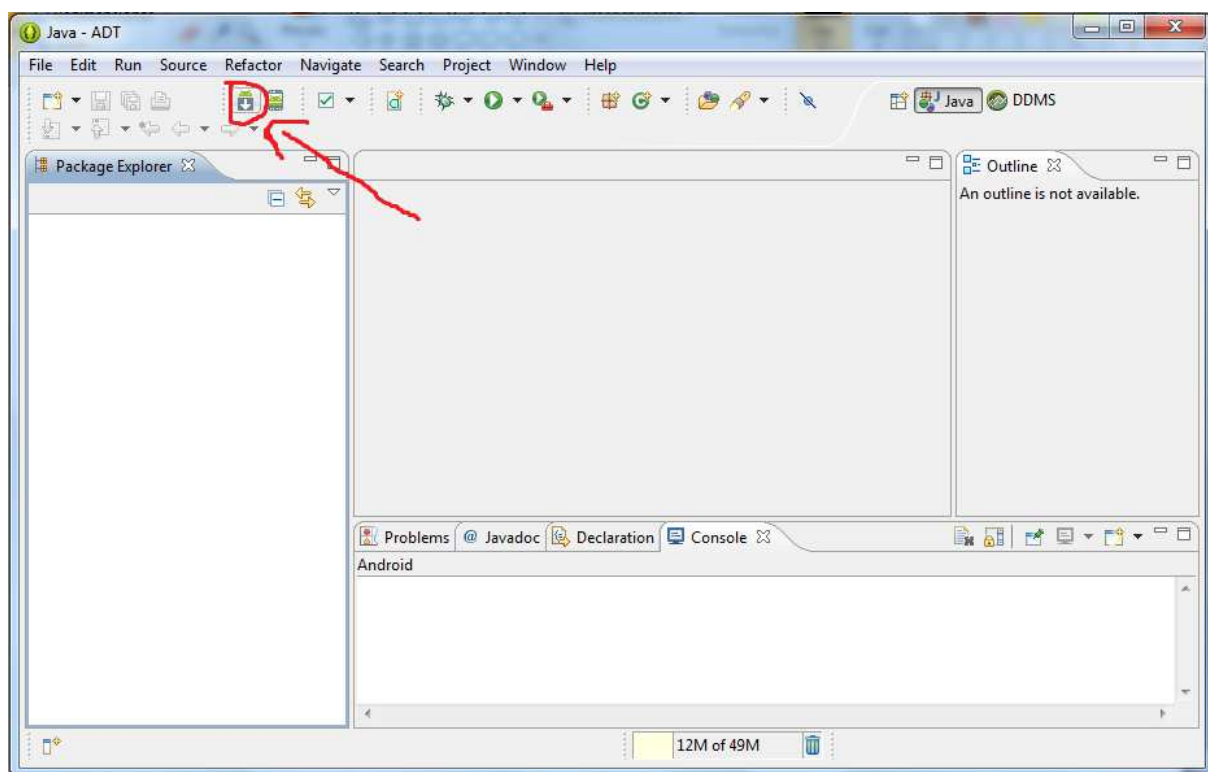
Figura27: Inicia AndroidDeveloper Tools



Fonte: eclipse

Como é a primeira vez que é inicializada a ferramenta é preciso clicar no ícone do Android SDK Manager, para baixar as APIs que desejamos trabalhar na construção do aplicativo, isso deverá ser feito porque, ele vem sem nem uma API pré-instalado.

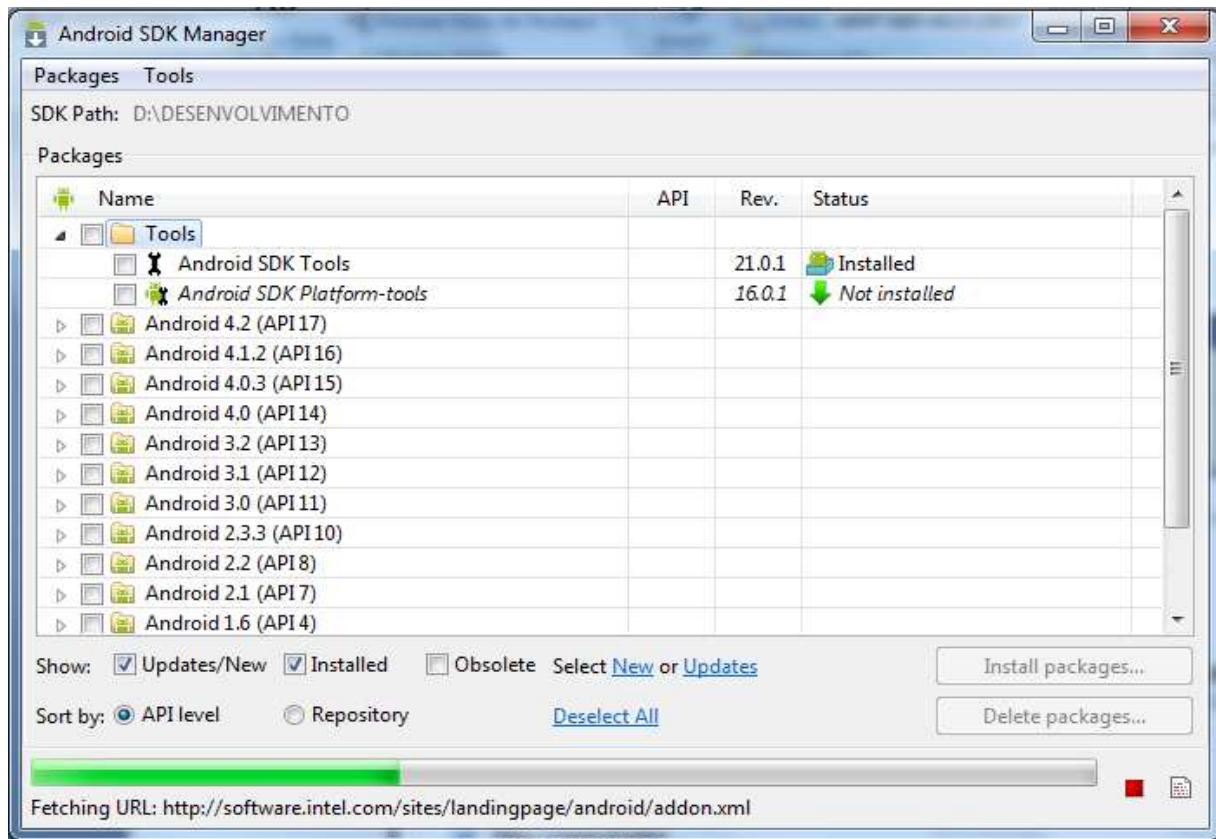
Figura28: Tela Inicial do AndroidDeveloper Tools



Fonte: eclipse

Ao executá-lo, ele irá mostrar as APIs disponíveis para ser baixado.

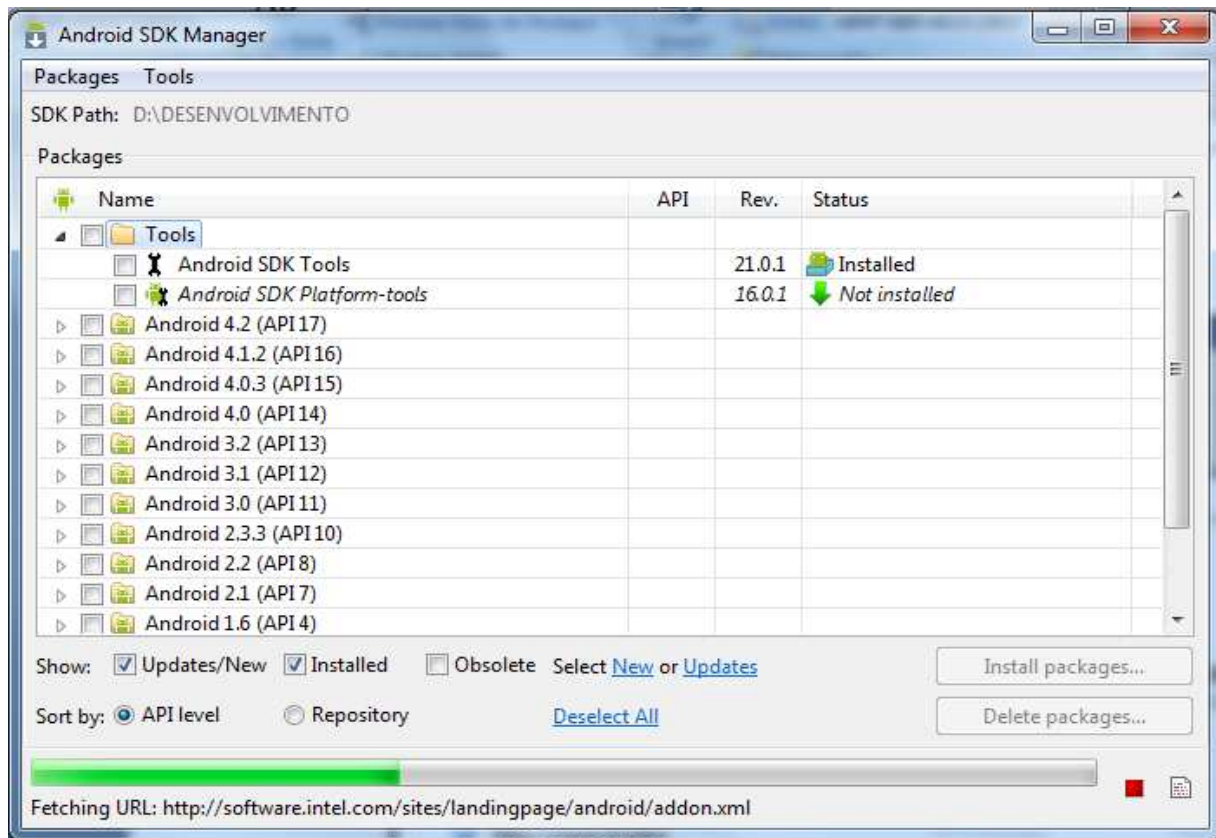
Figura 29:Android SDK Manager(APIs)



Fonte: eclipse

Você escolhe qual versão do API quer instalar.

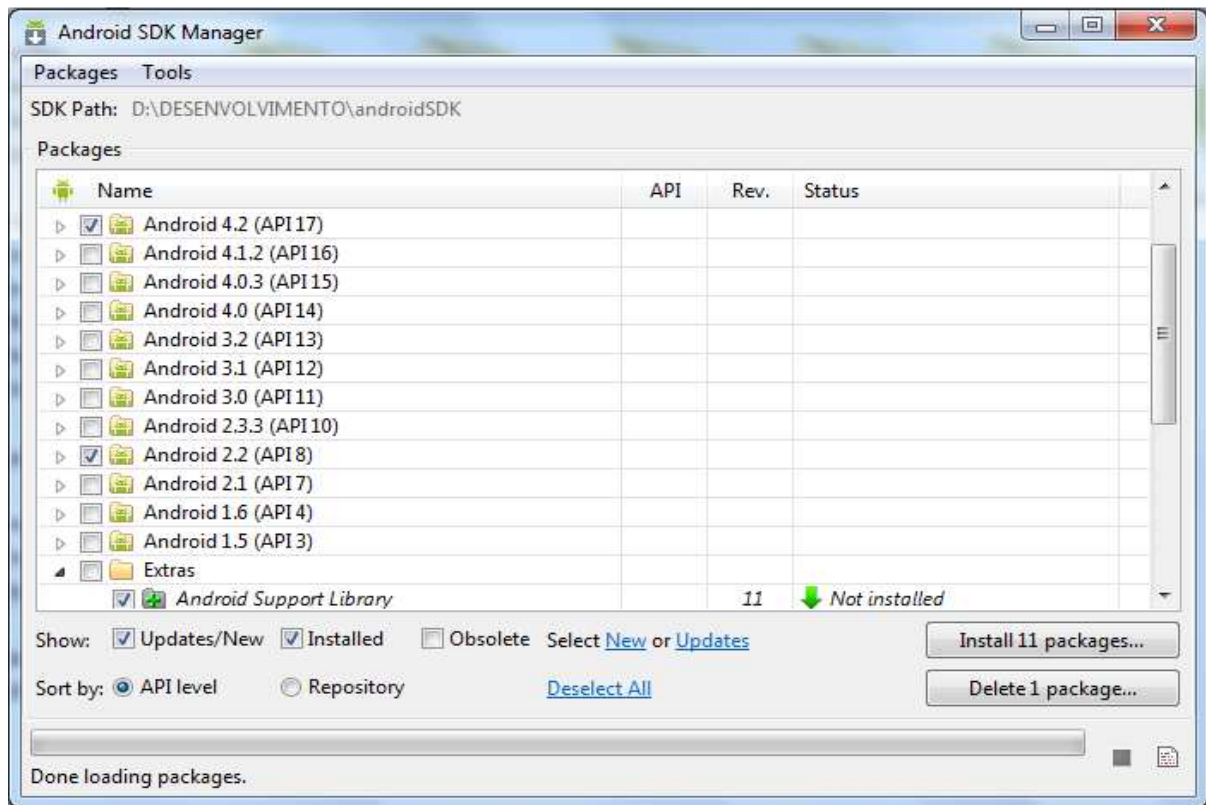
Figura 30: Escolhendo versão do API



Fonte: eclipse

Você escolhe qual versão do API quer instalar.

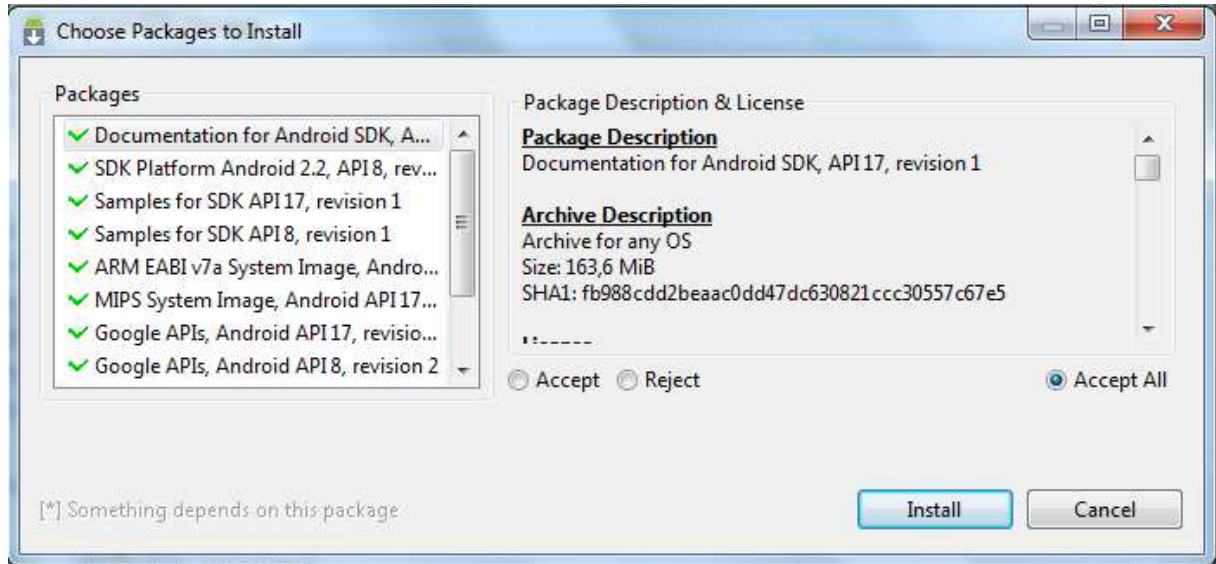
Figura 31: Escolhendo API



Fonte: eclipse

Escolher o pacote de instalação (instalar todos).

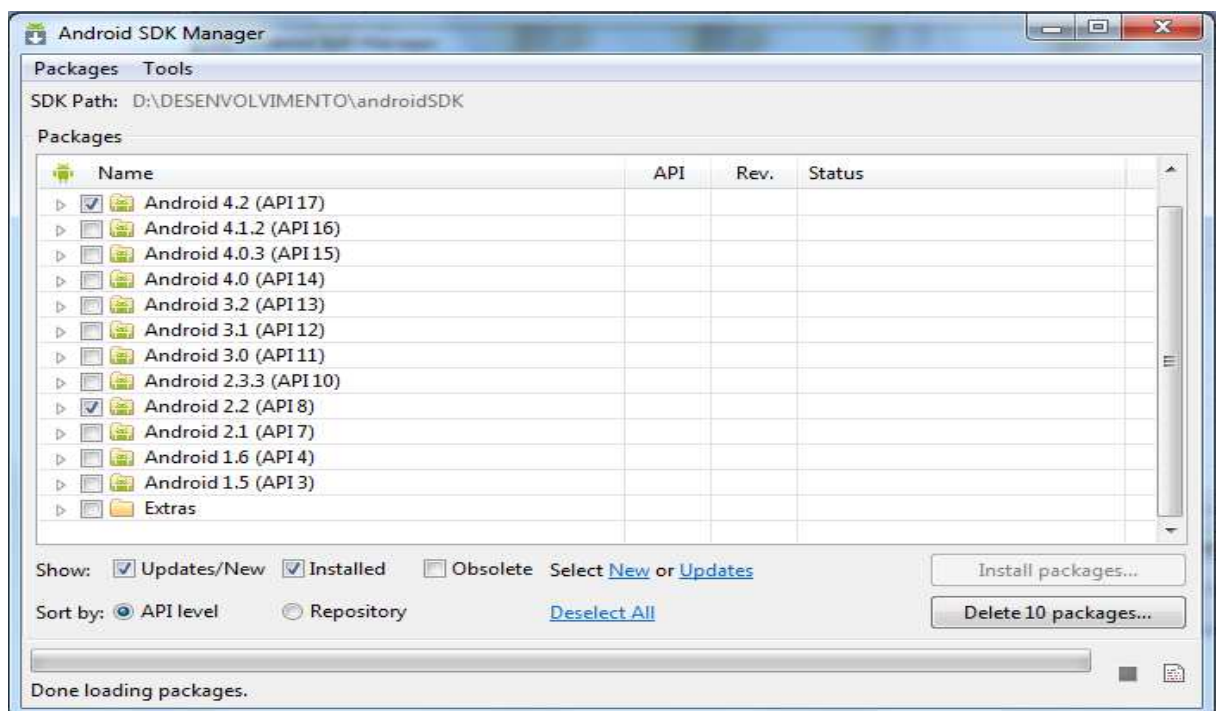
Figura 32: Escolher Pacote Para Instalação



Fonte: eclipse

Perceba de que agora a API 17 e a API 8 estão instaladas.

Figura 33: APIs Instaladas

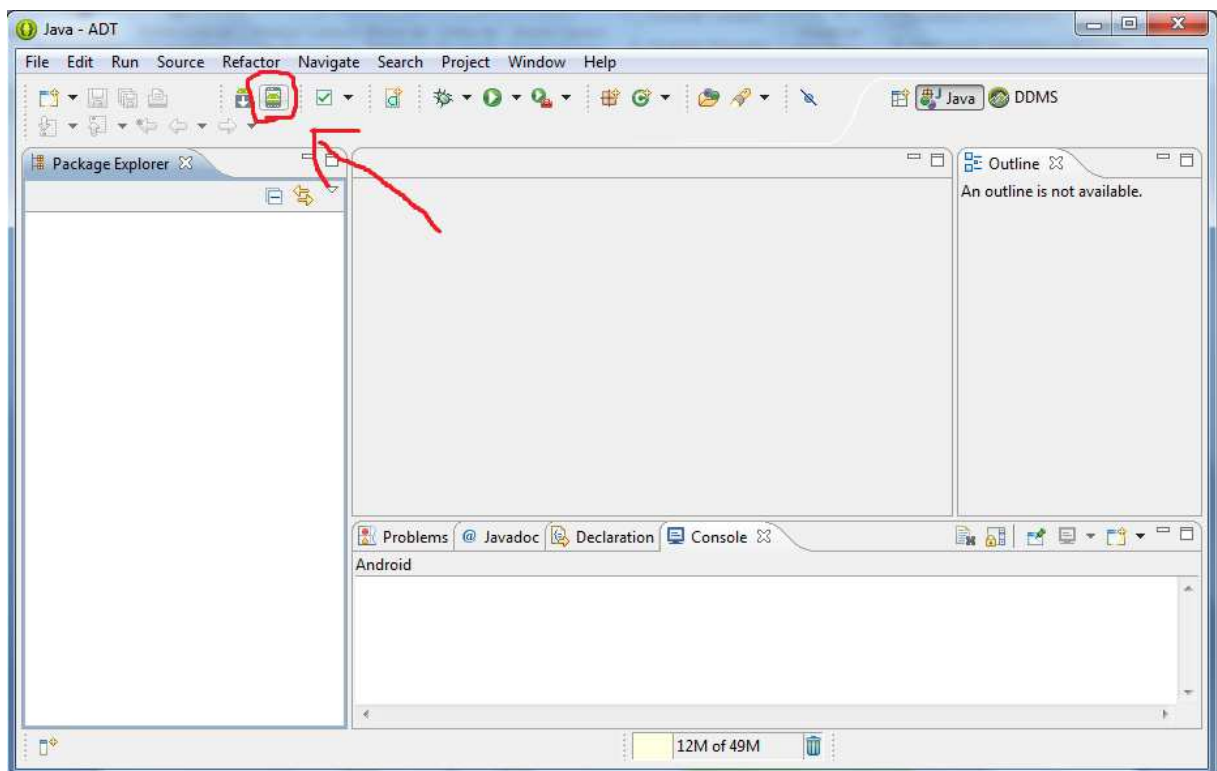


Fonte: eclipse

Bom, como já configuramos as APIs que iremos trabalhar, nos falta agora configurar o emulador, que no caso, iremos configurar o AndroidVirtualDevice Manager na versão 2.2 que será a API 8.

Clique no ícone doAndroidVirtualDevice para que possamos configurar o nosso emulador.

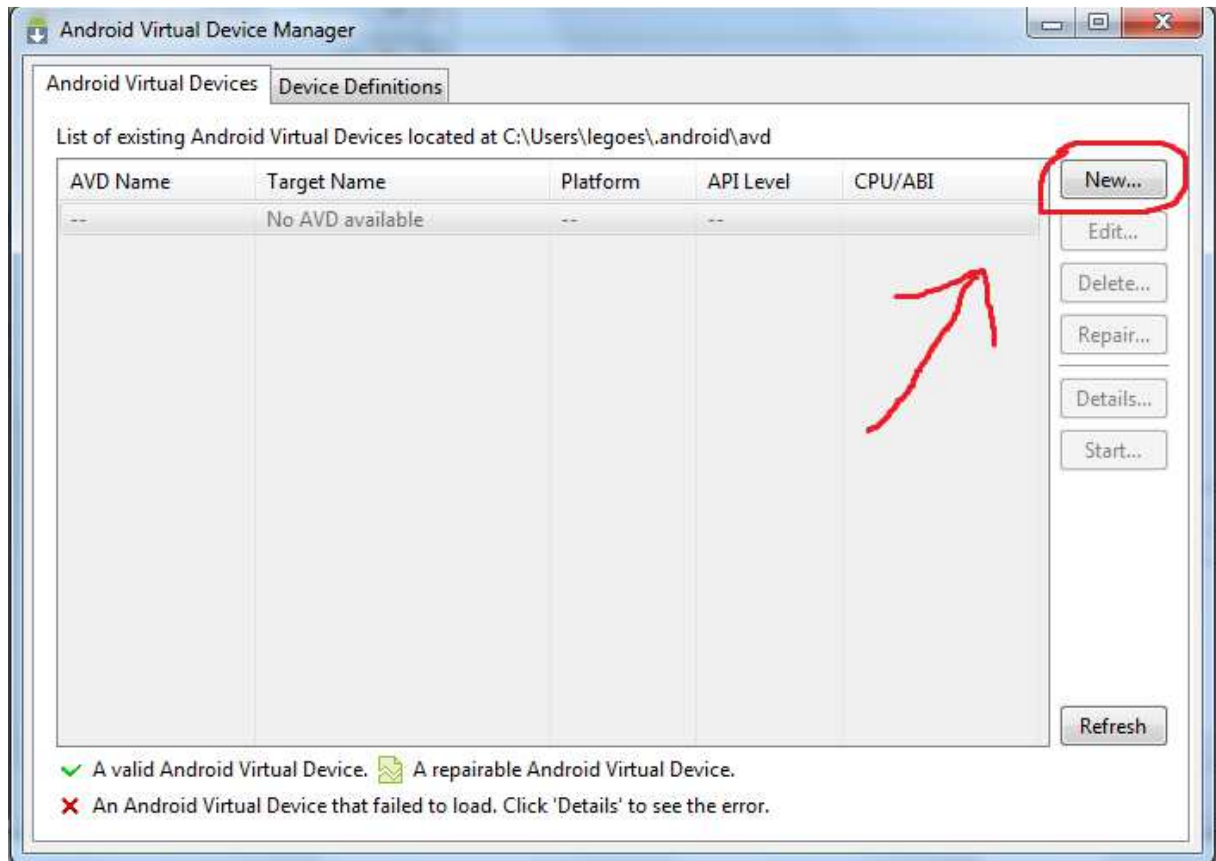
Figura 34: Ícone do Android Virtual Device Manager



Fonte: eclipse

Percebe que nosso AVD não contém nem uma configuração, clique em *New* para adicionar uma API.

Figura 35: Tela do Android Virtual Device Manager



Fonte: eclipse

Após ter clicado em *New*, vai abrir uma tela para você configurar, de nome a sua AVD no *AVD Name*, escolha o dispositivo *Device* ao qual você irá usar como emulador, escolha qual API você vai usar como objetivo *Target*.

Figura 36: Tela de Configuração das (APIs)

AVD Name: TCC

Device: Nexus S (4.0", 480 x 800: hdpi)

Target: Android 4.3 - API Level 18

CPU/ABI: ARM (armeabi-v7a)

Keyboard: ☒ Hardware keyboard present

Skin: ☒ Display a skin with hardware controls

Front Camera: None

Back Camera: None

Memory Options: RAM: 343 VM Heap: 32

Internal Storage: 200 MiB

SD Card: ☒ Size: MiB ☐ File: Browse...

Emulation Options: ☐ Snapshot ☐ Use Host GPU

☐ Override the existing AVD with the same name

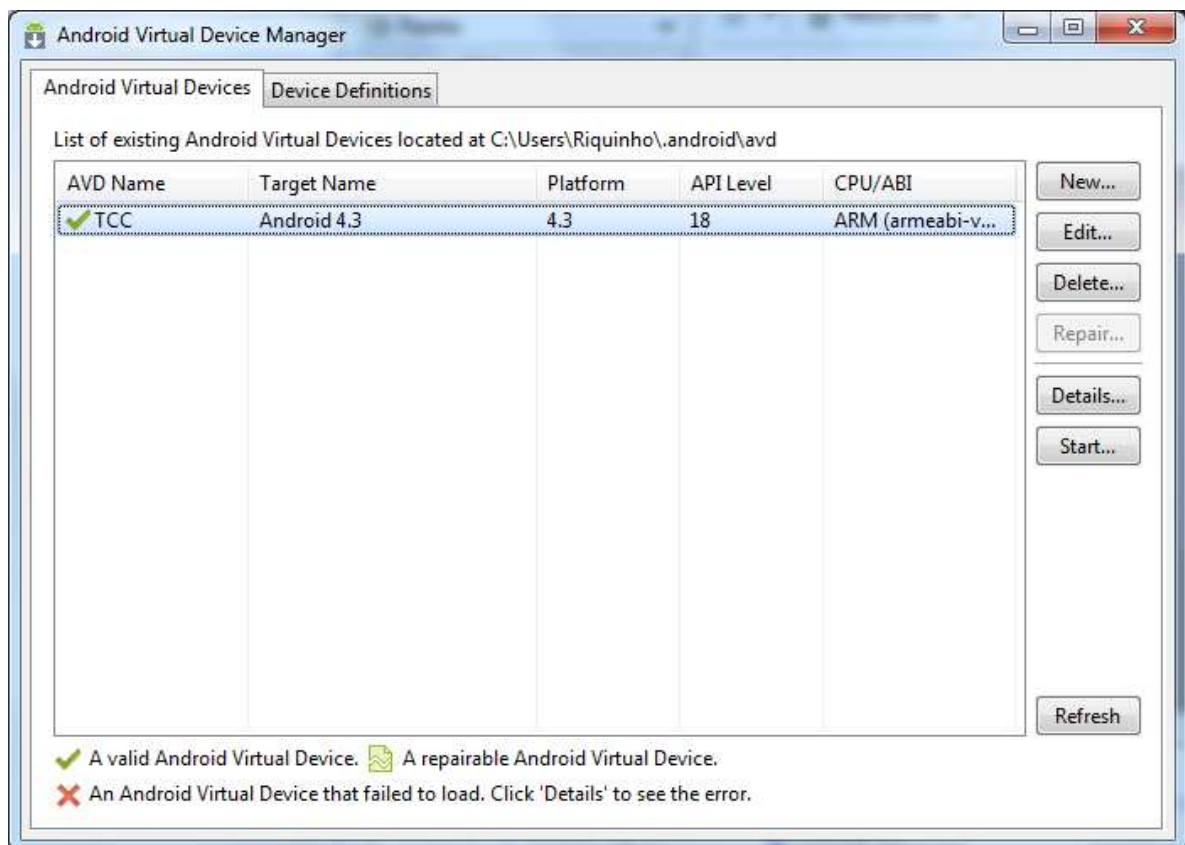
OK Cancel

Fonte: eclipse

Depois de ter preenchido os campos, clique em *OK*.

O Android Virtual Devices Manager agora tem uma configuração, a do Froyo ao qual se refere ao android 2.2 com API nível 8, caso queira ver como funciona o emulador, selecione a configuração que você inseriu no AVD e clique em *Start*, isso fará com que seu emulador entre em funcionamento.

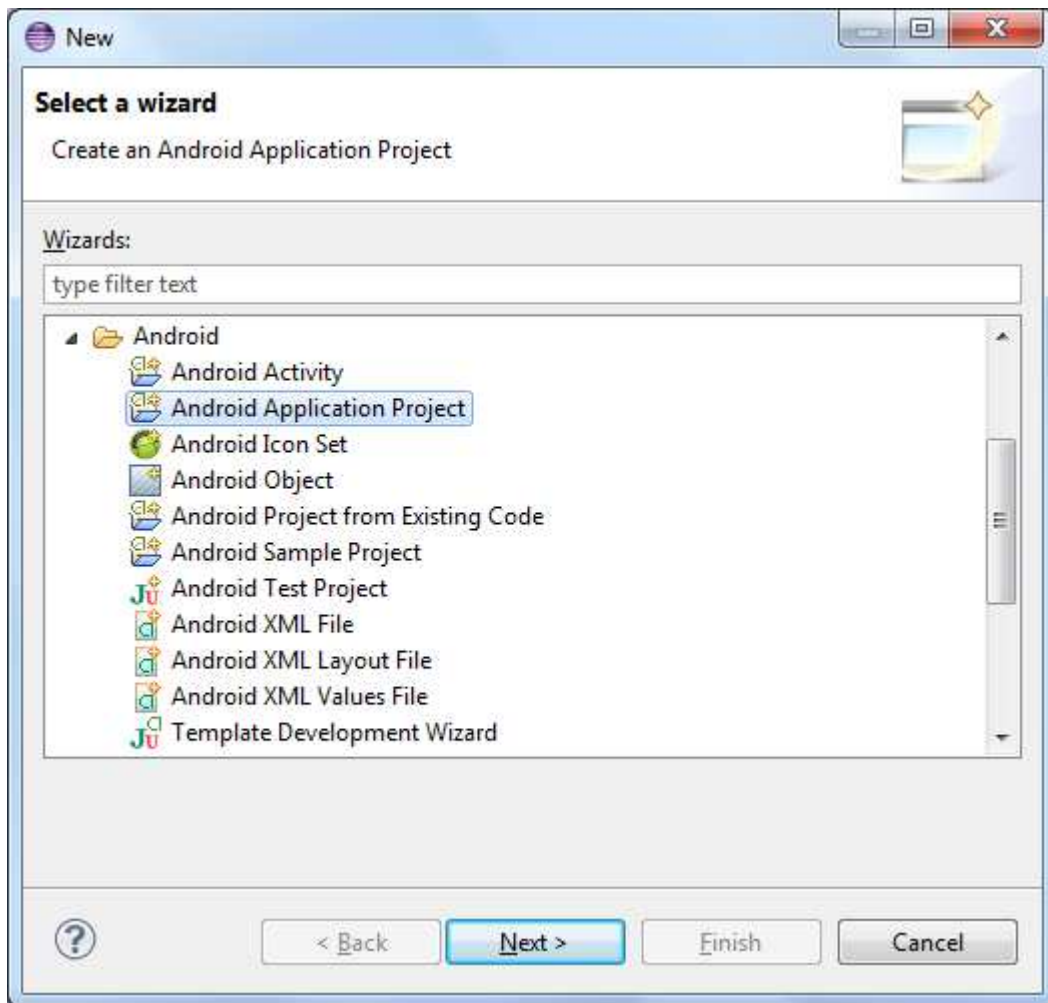
Figura 37: AVD Configurado



Fonte: eclipse

Agora vamos criar um projeto chamado CH CONTROLE, clique no menu *File / New / Other*. Selecione o projeto Android conforme figura abaixo. Depois de selecionar, clique em Next.

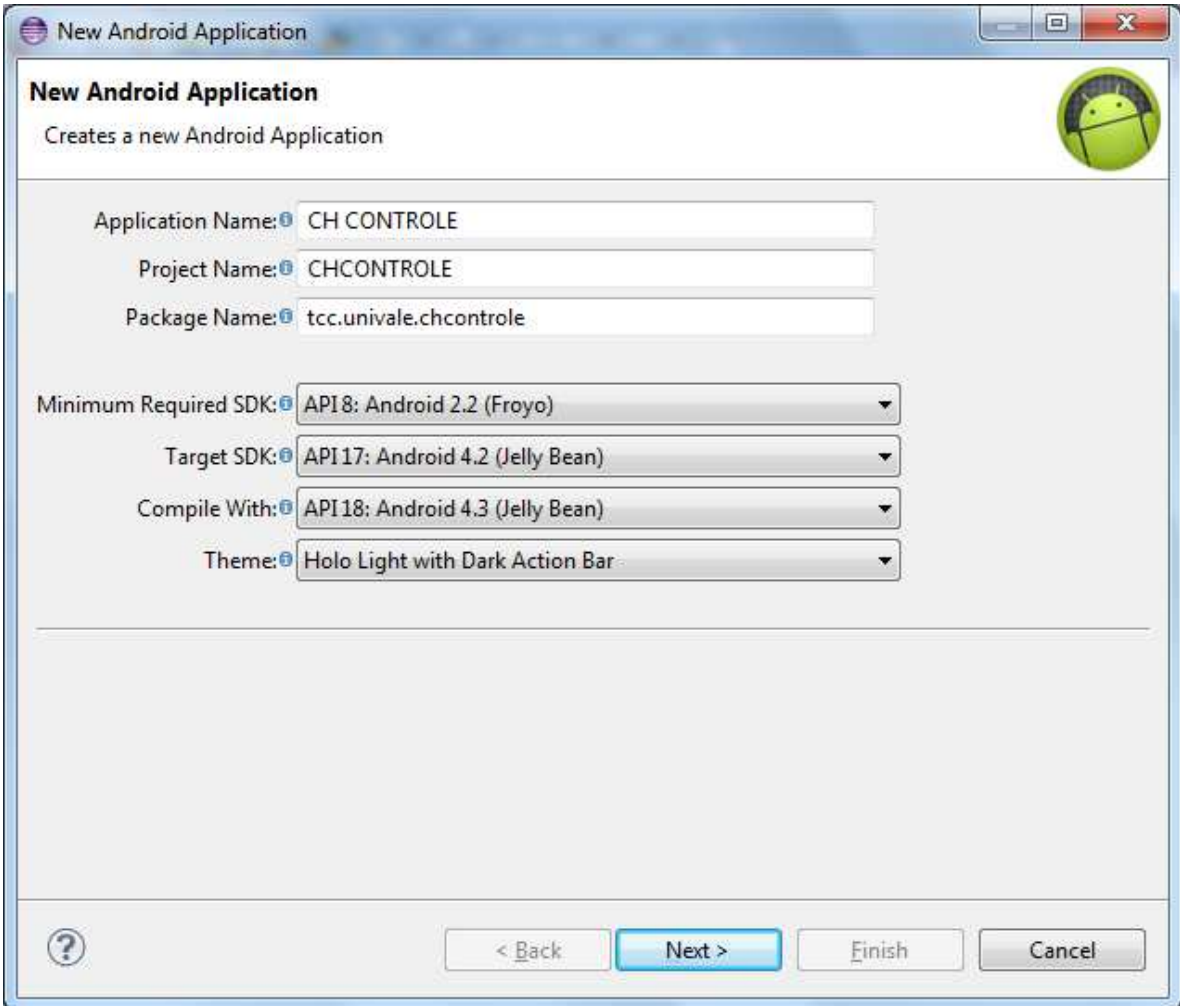
Figura 38: Criar um projeto de aplicativo Android



Fonte: eclipse

Preencha o nome do aplicativo, nome do projeto e o nome do pacote ao qual nosso aplicativo vai fazer uso. Nessa mesma tela temos que preencher os requisitos mínimo SDK e também o requisito máximo do objetivo maior no SDK, selecione qual API deseja compilar e depois é em *Next* até chegar na tela do *New BlankActivity*.

Figura 39: Tela de Configuração de Novo Aplicativo Android

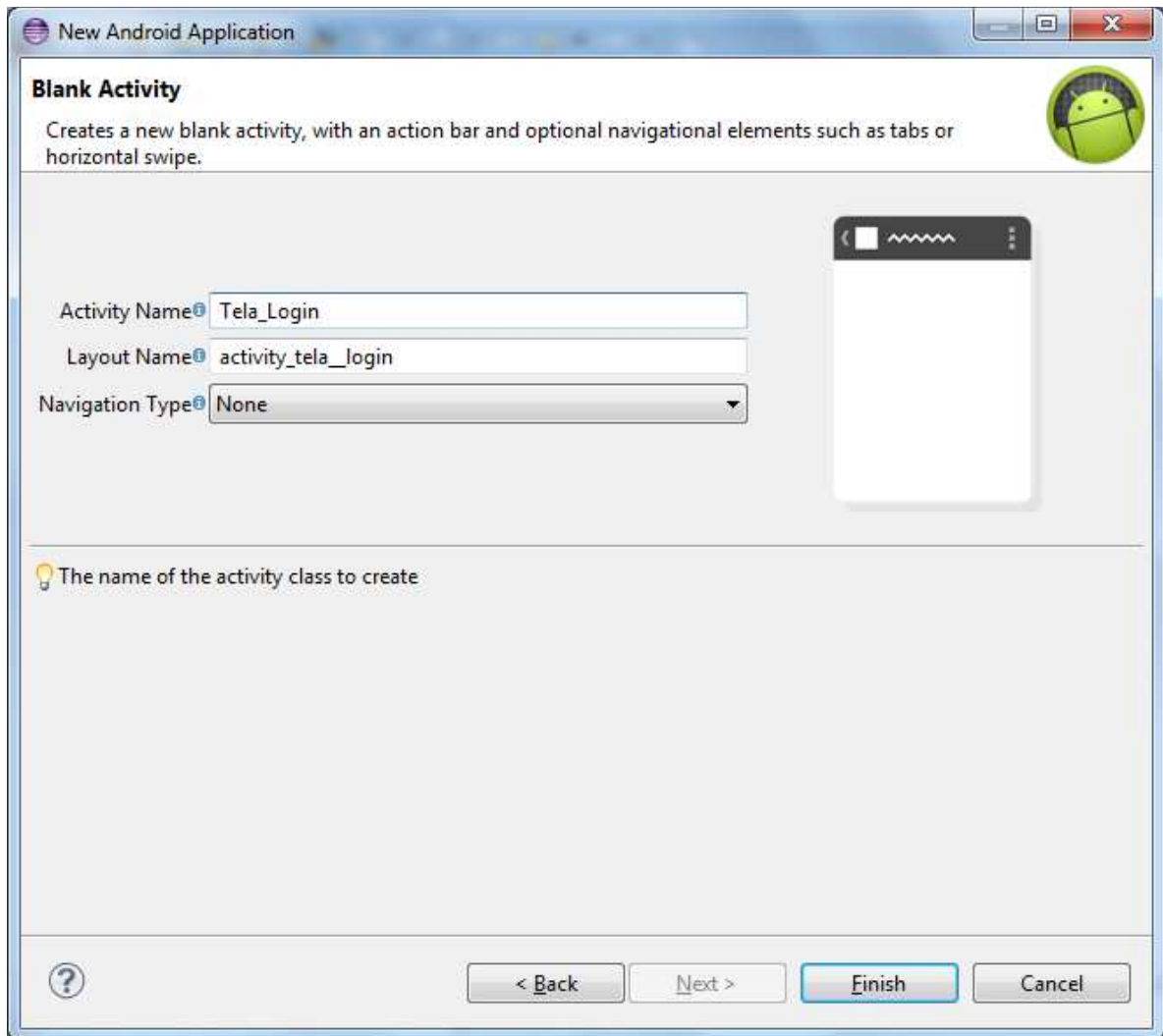


The screenshot shows the 'New Android Application' dialog box in the Eclipse IDE. The dialog has a title bar with the text 'New Android Application' and a standard Windows window control bar. Inside the dialog, the title 'New Android Application' is displayed in bold, followed by the subtitle 'Creates a new Android Application'. A green Android robot icon is located in the top right corner. The main area contains several input fields and dropdown menus: 'Application Name' with the value 'CH CONTROLE', 'Project Name' with the value 'CHCONTROLE', and 'Package Name' with the value 'tcc.univale.chcontrolle'. Below these are four dropdown menus: 'Minimum Required SDK' set to 'API8: Android 2.2 (Froyo)', 'Target SDK' set to 'API17: Android 4.2 (Jelly Bean)', 'Compile With' set to 'API18: Android 4.3 (Jelly Bean)', and 'Theme' set to 'Holo Light with Dark Action Bar'. At the bottom left is a help icon (a question mark in a circle). At the bottom right are four buttons: '< Back', 'Next >' (highlighted in blue), 'Finish', and 'Cancel'.

Fonte: eclipse

Nessa tela você precisa preencher os campos referentes à atividade em branco, preencha o nome da atividade, o nome do layout principal e o tipo da navegação.

Figura 40: Tela de Finalização de Configuração



Fonte: eclipse

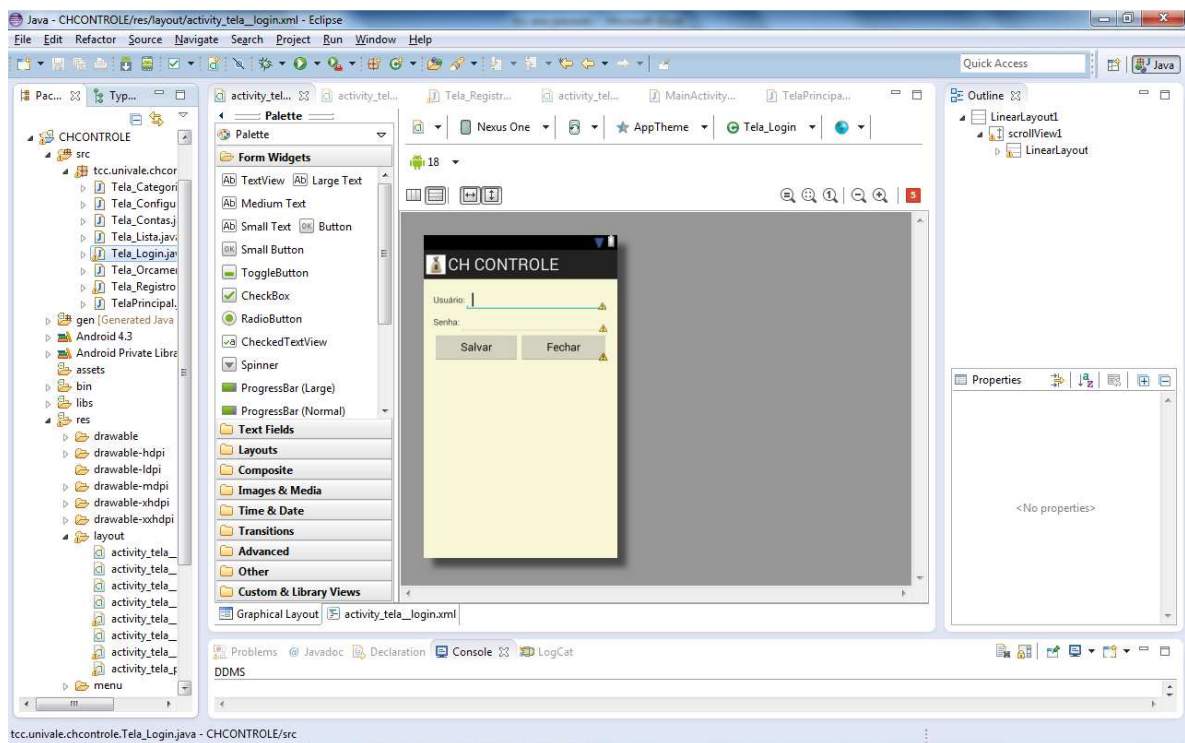
Depois é só clicar em *Next* até a tela BlankActivit.

Quando o projeto abre, a primeira visualização que aparece é a `activity_tela_login.xml` que é o Layout da Tela_Login.

Agora irei descrever a estrutura de um projeto CH CONTROLE. Observem que dentro da pasta *CHCONTROLE* existe uma pasta chamada *src* e dentro dela é que ficam os códigos fonte Java das aplicações. Observem que o arquivo *Tela_Login.java* se encontra dentro do pacote *tcc.univale.chcontroale*(Esse pacote também é uma pasta). Esse arquivo é a nossa aplicação Android.

Por padrão ele já vem com TextView com valor `hello world`.

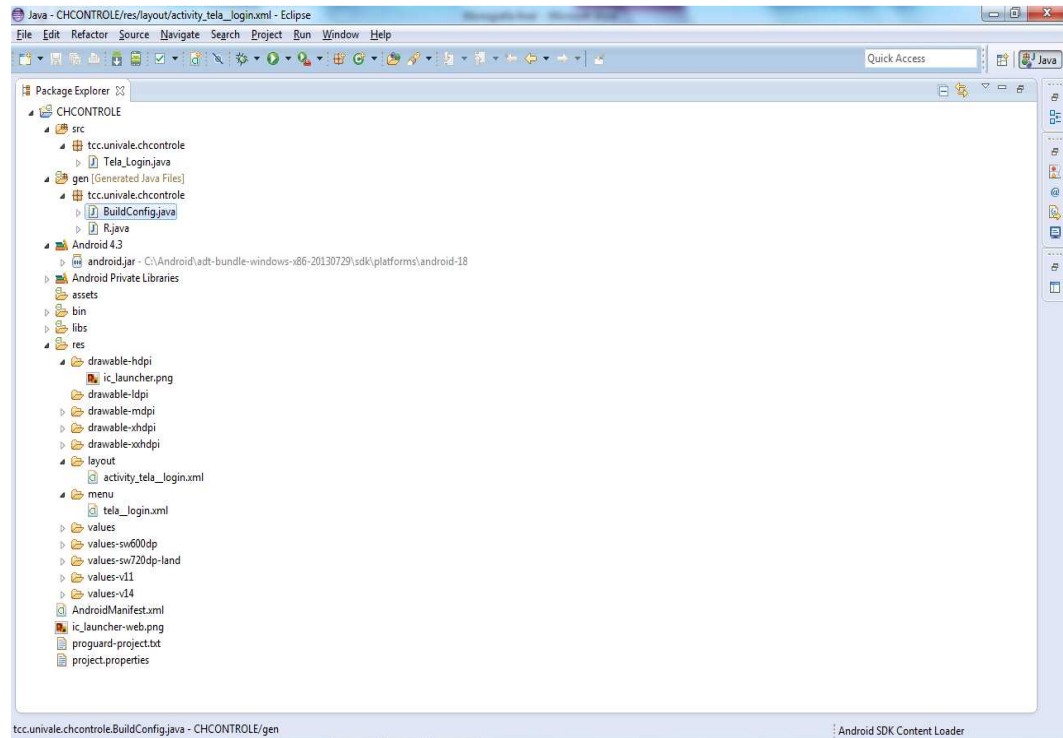
Figura 41: Ferramenta de Visualização do Projeto



Fonte: eclipse

Agora que já criamos o aplicativo CH CONTROLE vamos ver o que representa as principais pastas do projeto vista na figura abaixo:

Figura 42: Pastas do Projeto



Fonte: eclipse

As pastas são o seguinte:

- Src – contém os códigos .Java, neste exemplo a primeira tela do protótipo a *TelaLogin.Java*;
- Biblioteca Android 4.3 – Possui o arquivo android.jar, que contém todas as bibliotecas necessárias para o app;
- Gen – contém o arquivo R.Java, é responsável por mapear todos os recursos encontrados no projeto, não pode ser modificado;
- Res – contém os recursos usados no projeto com Layout: onde fica a interface gráfica, drawable: onde ficam as imagens e todas as resoluções separadas, e values: onde ficam as variáveis Strings;
- AndroidManifest.xml – onde são especificadas as permissões do projeto. E cada atividade que for criada tem que ser iniciada neste arquivo.

A classe já Java *Tela_Login.Java*, é onde se escreve o arquivo-fonte da atividade, onde esta classe herda da Activity e possui o método publico *OnCreate* que é o construtor e dentro dele encontra-se o outro método *setContentView* que responsável pela conexão da atividade com o interface gráfica *activity_tela_login*.

Figura 43: Classe Tela_Login.Java

```
package tcc.univale.chcontrole;

import android.os.Bundle;

public class Tela_Login extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_tela_login);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.tela_login, menu);
        return true;
    }

}
```

Fonte: eclipse

O arquivo *activity_tela_login.xml*, define com vai ser a interface do usuário, contendo vários componentes de telas como: TextView, EditText, Button, ImagemButon, Layout, etc. No exemplo a seguir vamos ver a interface do usuário com alguns componentes.

quadro2: Arquivo activity_tela_login.xml

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/hello_world"/>

<EditText
    android:id="@+id/edUsuario"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:ems="10"
    android:text="@string/edUsuario">
```

Fonte: eclipse

O @String é a constante String do seu aplicativo, localizada na pasta *res/values*. Neste arquivo pode-se adicionar uma nova String, atualizar e deletar. É de boa prática guardar todas as constantes String neste arquivo para fácil reutilização de código, manutenção e mudanças que este aplicativo possa sofrer.

Quadro 3: Arquivo String

```
<resources>

<stringname="app_name">CH CONTROLE</string>
<stringname="action_settings">Settings</string>
<stringname="hello_world">Hello world!</string>

</resources>
```

Fonte: eclipse

O Arquivo *AndroidManifest.xml*, para Lee (26) ele contém informações detalhadas sobre o aplicativo, em seguida algumas delas:

- Define o Nome do Pacote;
- Código e versão do aplicativo;
- Nome da versão;
- Aplicativo usa imagem de nome localizada na pasta *drawable*;
- Nome do Aplicativo na *Stringapp_nome*, definida na pasta *Strings.xml*.

Figura 44: Arquivo AndroidManifest

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="tcc.univale.chcontrole"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="17" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name="tcc.univale.chcontrole.Tela_Login"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

Fonte: eclipse

O arquivo R.java é gerado automaticamente pelo eclipse conforme for criado ou modificado porém se estiver erros ele não é gerado, ele é que faz referencia de todos os recursos encontrados no projeto.

Figura 45: arquivo R.Java

```
package tcc.univale.chcontrole;

public final class R {
    public static final class attr {
    }
    public static final class dimen {

        public static final int activity_horizontal_margin=0x7f040000;
        public static final int activity_vertical_margin=0x7f040001;
    }
    public static final class drawable {
        public static final int ic_launcher=0x7f020000;
    }
    public static final class id {
        public static final int action_settings=0x7f080000;
    }
    public static final class layout {
        public static final int activity_tela_login=0x7f030000;
    }
    public static final class menu {
        public static final int tela_login=0x7f070000;
    }
    public static final class string {
        public static final int action_settings=0x7f050001;
        public static final int app_name=0x7f050000;
        public static final int hello_world=0x7f050002;
    }
    public static final class style {
        public static final int AppBaseTheme=0x7f060000;

        public static final int AppTheme=0x7f060001;
    }
}
```

Fonte: eclipse

6 CRONOGRAMA

6.1 CRONOGRAMA INICIAL

Tabela 1: Cronograma Inicial

Tarefa	Mês									
	Mac.	Abri.	Mai.	Jun.	Jul.	Ago.	Set.	Out.	Nov.	Dez.
T1	X	X	X	X	X	X				
T2		X	X	X	X	X	X	X		
T3						X	X	X		
T4							X	X	X	X
T5								X	X	X

6.2 DESCRIÇÕES DAS TAREFAS DO CRONOGRAMA

Tabela 2: Descrição do Cronograma

Descrições das Tarefas	
T1	Realização de estudos sobre qual modelo de desenvolvimento de Software usar, estudos sobre finanças, orçamento, qual sistema operacional usar, e principais aplicativos existentes no mercado.
T2	Realização da coleta das informações (Requisitos) através de pesquisa para se obter os casos de uso, tais caso de uso serão usado a todo o momento durante a todo processo.
T3	Documentação e interface gráfica do software.
T4	Teste e análises serão feito durante o desenvolvimento do protótipo.
T5	A implantação e a finalização do projeto, produto final.

7 PROJETO DE IMPLANTAÇÃO

7.1 AMBIENTE USUÁRIO

O usuário para se beneficiar deste aplicativo terá que possuir um Smartphone com sistema operacional Android com versão acima de 2.2, estar conectado na Internet, acessar o aplicativo Play Store, se estiver no celular se não deve baixar este aplicativo, achar o App CHCONTROLE, baixar e instalar em seu Smartphone.

8 RESULTADOS E DISCUSSÃO

Após vários testes realizados com os três Aplicativos Móveis existentes no mercado, chegou-se a conclusão de queo MoneyWise leva vantagem, pois tanto na versão paga quanto na versão free ele obteve um desempenho de processamento superior ao demais, possuindo muitas funcionalidades uteis, ao contrario do Money Lover que depois da ultima atualização a versão free deixa a desejar nas funcionalidades e backup, por sua vez possui alerta automático de datas de vencimento, já o Finanças Pessoas fica um pouco atrás dos outros dois, pois não possui suporte para clientes e suas informações são confusas mais ao contrário do MoneyWise e MoneyLover possui cadastro de capitais.

Com o termino dos testes, chegou-se a conclusão de que o desenvolvimento de um novo aplicativo adotando as principais funcionalidades encontradas dentre os aplicativos analisados, juntamente com alguns novos requisitos levantados através de estudos de casos e leituras técnicas, ira proporcionaraousuário um grande auxílio no controle de suas finanças pessoais.

Figura46: Comparação de Funcionalidades

Funcionalidades	Finac. Pessoas	MoneyWise	MoneyLover
Versão paga	X	X	X
Versão gratuita	X	X	X
Cadastro Usuário			X
Status das Despesas		X	
Cadastro de Contas	X	X	
Off Line	X	X	X
Idioma Português	X	X	X
Rendas e Despesa		X	
Gráfico por categoria		X	X
Orçamentos		X	X
Contas	X	X	X
Saldo banco		X	
Exportar relatório	X	X	X
Importar relatório	X	X	X
Backup em cartão sd		X	X
Gestão de dividas			X

Fonte: autoria própria

9 TRABALHOS FUTUROS

Este projeto fica em aberto para implantação de algumas funcionalidades que no momento não é possível desenvolver neste aplicativo deixando-o mais completo.

Essas funcionalidades são:

- Geração de gráficos, onde com o cadastro de todos os registros será possível gerarum gráfico com varias formas identificando a porcentagem de cada categoria salva em seus respectivos registros, pra se ter uma visão geral de seu gastos;
- Tornar este aplicativo on-line, possibilitando o usuário ter acesso a seus dados de qualquer lugar mesmo sem esta com seu Smartphone;
- Implementar um ferramenta de estatística, onde o usuário é auxiliado por esta ferramenta sobre como tomar a melhor decisão na hora da compra;
- Importar e exportar tabelas, onde o usuário pode exportar e importar dados cadastrados para outro lugar;
- Configurações customizadas, para o usuário poder configurar funcionalidades e designer do jeito que quiser;
- Backup e cartão SD.

Sendo assim, proponho continuar a desenvolver este aplicativo com as funcionalidades existente nele, com mais essas possíveis melhorias citadas à cima.

10 CONCLUSÃO

Através de estudos feitos para este projeto conclui-se que as pessoas precisam ter uma vida financeira saudável, e para isso precisa se ter um controle detalhado de suas finanças assim conseguindo manter suas despesas e receitas equilibradas, para que em um momento de crise o mesmo possa supera-la com mais facilidade e nos momentos b nos poder alcançar um sonho grande.

Com ajuda da tecnologia cada vez mais presente em nossas vidas venho apresentar o protótipo para Smartphone com sistema operacional Android, o aplicativo CH CONTROLE que vai auxiliar com muita competência o usuário que se beneficiar dele, com ele o individuo vai poder anotar todos seus gastos e despesas na hora que acontecer, pois se tivermos sempre atualizado sobre sua situação financeira poderá tomar decisões acertadas para cada ocasião.

REFERÊNCIAS BIBLIOGRÁFICAS

BEZERRA, Eduardo. Princípios de Análise e projeto de Sistema com UML. 2. ed. Rio de Janeiro: Elsevier, 2007.

BOOCH, Grady; RUMBAUCH, James; JACOBSON, Ivar. Uml Guia do Usuário. 2. ed. Rio de Janeiro: Campus, 2005.

COSTA, Angelo Giuseppe Meira. Introdução ao Desenvolvimento de Aplicativos para Android. 1. ed. Rio de Janeiro: Ciência Moderna, 2011.

LEE, Wei-Meng. Introdução ao Desenvolvimento de Aplicativos para Android. 1. ed. Rio de Janeiro: Editora Ciência Moderna Ltda, 2011.

MELO, Ana Cristina. **Desenvolvendo aplicações com UML 2.2:** do conceitual á Implementação / Ana Cristina Melo. 3. Ed. Rio de Janeiro: Brasport, 2010.

MUCHOW, John w. Core J2ME: Tecnologia& MIDP. 1. ed. São Paulo: Pearson Makron Books, 2004.

SOMMERVILLE, Ian. Engenharia de Software. 9. ed. São Paulo: Pearson Prentice Hall, 2011.

TORRES, Ricardo Lobo. O Orçamento na constituição. 1. ed. Rio de Janeiro: Renovar, 1995.

WAZLAWICK, Raul Sidnei. Análise e projeto de sistemas de informação orientado a objetos/ Raul Sidney Wazlawinck. 9. ed. Rio de Janeiro: Elsevier, 2004.

REFERÊNCIAS BIBLIOGRÁFICAS – SITES

- [1] Site GOOGLE PLAY Aplicativo Money Lover. 2013;
https://play.google.com/store/apps/details?id=com.bookmark.money&hl=pt_BR, acessado em 03 de abril de 2013.
- [2] Site Google Play Aplicativo Money Wise, 2013;
https://play.google.com/store/apps/details?id=com.handynorth.moneywise_free&hl=pt_BR, acessado em 03 de abril de 2013
- [3] Site Ferramenta CASE Astah;
<http://astah.net/editions/community>, acessado em 16/08/2013.
- [4] Site Colegio Marques de Olinda;
http://www.colegiomarquesdeolinda.com.br/artigos/investimentos_em_educacao_e_financas_pessoais?id=8-5, 14 de março de 2013.
- [5] Site Ciência Contabeis IESA;
<http://cienciascontabeisiesa.blogspot.com.br/2010/09/educacao-financeira.html>, acessado em 20/04/2013.
- [6] Site info escola;
<http://www.infoescola.com/engenharia-de-software/rup/>, acessado em 08 de março de 2013.
- [7] Site TecmundoAndroid;
<http://www.tecmundo.com.br/android>, acessado em 09 de Setembro de 2013.
- [8] Site do AndroidDeveloper;
<http://developer.android.com/about/versions/index.html>, acessado em 20/09/2013.
- [9] Site do Android SDK;
<http://developer.android.com/sdk/index.html>, acessado em 20/09/2013.
- [10] Site AndroidActivity;
<http://developer.android.com/reference/android/app/Activity.html>, acessado em 21/09/2013.
- [11] Site Android UI;
<http://developer.android.com/guide/topics/ui/index.html>, acessado em 21/09/2013.
- [12] Site Androi data-storage;
<http://developer.android.com/guide/topics/data/data-storage.html>, acessado em 21/09/2013.
- [13] Site SQLite;
<http://www.sqlite.org>, acessado em 01 de outubro de 2013.
- [14] Site softonic;
<http://dindin.softonic.com.br/>, acessado em 06 de setembro de 2013.

[15] Site Social Carisma;
http://socialcarisma.blogspot.com.br/2012_05_01_archive.html, acessado em 10 de setembro de 2013.

[16] Site Google Play;
https://play.google.com/store/apps/details?id=incomeincloud.android.activity&hl=pt_BR, acessado em 06 de junho de 2013.

[17] Site SUN Java;
<http://java.sun.com/javase/downloads/index.jsp>, acessado em 16/10/2013.

[18] Site Google Play;
https://play.google.com/store?hl=pt_BR, acessado em 05/04/2013.

[19] site Campuseiros;
<http://campuseiros.com/temas/mobile/o-futuro-e-mobile-e-e-concorrido>, acessado em 29 de outubro de 2013.

