

LUIS GUSTAVO DE OLIVEIRA MAZZO

**IMPLEMENTAÇÃO DE APLICATIVO MÓVEL EM GOOGLE ANDROID
PARA GESTÃO INTELIGENTE DE CONTATOS E AGENDAMENTOS**

Assis

2013

LUIS GUSTAVO DE OLIVEIRA MAZZO

**IMPLEMENTAÇÃO DE APLICATIVO MÓVEL EM GOOGLE ANDROID
PARA GESTÃO INTELIGENTE DE CONTATOS E AGENDAMENTOS**

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do Curso Superior de Análise e Desenvolvimento de Sistemas.

Orientado: Prof. Esp. Guilherme de Cleve Farto

Área de Concentração: Informática

Assis
2013

FICHA CATALOGRÁFICA

MAZZO, Luis Gustavo de Oliveira

Implementação de Aplicativo Móvel em Google Android para Gestão Inteligente de Agendamentos de Eventos / Luis Gustavo de Oliveira Mazzo. Fundação Educacional do Município de Assis - FEMA - Assis, 2013.

57 Páginas.

Orientador: Prof. Esp. Guilherme de Cleva Farto

Trabalho de Conclusão de Curso - Instituto Municipal de Ensino Superior de Assis - IMESA.

1.Eventos 2.*SmartAgenda* 3.Aplicativo Móvel

CDD: 001.61
Biblioteca da FEMA

IMPLEMENTAÇÃO DE APLICATIVO MÓVEL EM GOOGLE ANDROID PARA GESTÃO INTELIGENTE DE AGENDAMENTOS DE EVENTOS

LUIS GUSTAVO DE OLIVEIRA MAZZO

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do Curso Superior de Análise e Desenvolvimento de Sistemas, analisado pela seguinte comissão examinadora:

Orientador: Prof. Esp. Guilherme de Cleva Farto

Analisador: Prof. Dr. Alex Sandro Romeo de Souza Poletto

Assis

2013

DEDICATÓRIA

Dedico este trabalho primeiramente a Deus, pois sem Ele nada seria. Dedico também a meus familiares e minha noiva, Karina de Souza Silva por estarem sempre ao meu lado e a meus amigos por sempre me apoiarem e ajudarem quando necessário.

AGRADECIMENTOS

Agradeço a Deus por me dar saúde, forças e sabedoria para percorrer esta longa estrada e parte da vida que é a formação no ensino superior e por estar sempre me guardando e protegendo, sem deixar que me abalasse nos momentos de dificuldade e situações adversas.

Agradeço também a minha família que tanto amo, minha mãe Simone Aparecida de Oliveira Mazzo, meu pai Luis Carlos Mazzo, meu irmão Fernando de Oliveira Mazzo e minha Noiva Karina de Souza Silva, por sempre me apoiarem mesmo nos momentos mais difíceis desta jornada para minha formação acadêmica.

Sou muito grato também a todos os meus amigos, Vitor Ribeiro Vidal, Rodolfo Pires, Maicon Alan Guizelini, Douglas Moret, Stella de Souza Silva, Diego Zaratini, Jonatas Chizzolini, Leonardo Rodrigues Bastelli, Gabriel Vieira Galli, Francisco Patta, Daniel de Souza Campos e todos os muitos outros amigos, que desde o início desta longa estrada estiveram juntos, batalhando, se esforçando para que nenhum fraquejasse e deixasse de lado este sonho que agora está para se concluir.

Eu agradeço a todos os Professores da Fundação Educacional do Município de Assis, FEMA, que sempre nos ajudaram no que foi preciso e tudo o mais que pudessem nos fornecer para nos preencher de conhecimento e um agradecimento especial a meu orientador, Guilherme de Cleve Farto que me auxiliou nos momentos de maior precisão para a finalização deste Trabalho de Conclusão de Curso.

RESUMO

Este projeto consiste no desenvolvimento de um aplicativo para dispositivos móveis, utilizando a plataforma de programação em ascensão, *Google Android*, que é baseada na linguagem Java. A ideia de implementação deste aplicativo originou-se da necessidade dos usuários em obter um maior controle sobre seus horários e compromissos, podendo manuseá-los de maneira mais simples e ágil, possibilitando uma visão geral dos eventos em seu cotidiano, com a geração de relatórios com informações dos eventos agendados. Além da tecnologia Google Android, foi utilizado o banco de dados SQLite que é uma tecnologia embarcada ao Android, facilitando a manipulação dos dados. O intuito na realização deste projeto é disseminar o uso das tecnologias utilizadas no mesmo e divulgar a área de softwares para agendamento de eventos.

Palavras chave: Google Android, SQLite, Aplicativo Móvel

ABSTRACT

This project consists of developing an application for mobile devices, using the programming language on the rise, Google Android, which comes from the Java language. The idea of implementing this application originated from the need for users to gain greater control over their schedules and commitments and can handle them in a more simple and flexible, allowing an overview of the events in their daily lives, with reporting with information of the scheduled events. Besides the aforementioned technology Google Android, will use the SQLite database, which is an embedded technology to Android, eliminating the need to use a Software Management Database, facilitating the manipulation of data. The aim in this project is to spread the use of the technologies used in the same area and disseminate software for scheduling events.

Keywords: Google Android, SQLite, Mobile App

LISTA DE FIGURAS

Figura 1 - Mapa Mental	25
Figura 2 - Diagrama de Caso de Uso - <i>SmartAgenda</i>	27
Figura 3 - Diagrama de Atividade - Cadastrar Cliente.....	36
Figura 4 - Diagrama de Atividade - Cadastrar Evento.....	36
Figura 5 - Diagrama de Atividade - Cadastrar Funcionário	37
Figura 6 - Diagrama de Atividade - Alterar Cliente	38
Figura 7 - Diagrama de Atividade - Alterar Evento	38
Figura 8 - Diagrama de Atividade - Alterar Funcionário	39
Figura 9 - Diagrama de Atividade - Excluir Cliente.....	39
Figura 10 - Diagrama de Atividade - Cancelar Evento	40
Figura 11 - Diagrama de Atividade - Excluir Funcionário	40
Figura 12 - Diagrama de Classes.....	41
Figura 13 - Diagrama de Entidade-Relacionamento	42
Figura 14 - Estrutura WBS do Projeto	44
Figura 15 - Tela Inicial do <i>SmartAgenda</i>	49
Figura 16 - Tela de Listagem dos Clientes.....	50
Figura 17 - Tela de Cadastro de Clientes.....	51
Figura 18 - Classe de Definição da Tabela no Banco de Dados	52
Figura 19 - Definição do Banco de Dados Utilizado	53
Figura 20 - Classe de Inicialização e Exclusão das Tabelas.....	54

LISTA DE TABELAS

Tabela 1 - Especificação de Caso de Uso - Parametrizar	27
Tabela 2 - Especificação de Caso de Uso - Listar Clientes.....	28
Tabela 3 - Especificação de Caso de Uso - Listar Funcionários	28
Tabela 4 - Especificação de Caso de Uso - Listar Eventos.....	29
Tabela 5 - Especificação de Caso de Uso - Cadastrar Funcionários	30
Tabela 6 - Especificação de Caso de Uso - Cadastrar Clientes.....	30
Tabela 7 - Especificação de Caso de Uso - Cadastrar Eventos.....	31
Tabela 8 - Especificação de Caso de Uso - Alterar Funcionários	32
Tabela 9 - Especificação de Caso de Uso - Alterar Clientes	32
Tabela 10 - Especificação de Caso de Uso - Alterar Eventos	33
Tabela 11 - Especificação de Caso de Uso - Excluir Funcionários	34
Tabela 12 - Especificação de Caso de Uso - Excluir Clientes.....	34
Tabela 13 - Especificação de Caso de Uso - Excluir Eventos.....	35
Tabela 14 - Orçamento	45

LISTA DE ABREVIATURAS E SIGLAS

IDE	<i>Integrated Development Environment</i> (Ambiente Integrado de Desenvolvimento)
API	<i>Application Programming Interface</i> (Interface de Programação de Aplicativos)
DDL	<i>Data Definition Language</i> (Linguagem de Definição de Dados)
DML	<i>Data Manipulation Language</i> (Linguagem de Manipulação de Dados)
SQL	<i>Structured Query Language</i> (Linguagem de Consulta Estruturada)
HTML	<i>HyperText Markup Language</i> (Linguagem de Marcação de Hipertexto)
TCP/IP	<i>Transmission Control Protocol/Internet Protocol</i> (Protocolo de controle de transmissão/Protocolo de Internet)
JVM	<i>Java Virtual Machine</i> (Máquina Virtual Java)
XML	<i>eXtensible Markup Language</i> (Linguagem de Marcação Extensível)
EUA	Estados Unidos da América
SDK	<i>Software Development Kit</i> (Kit de Desenvolvimento de Software)
MMS	<i>Multimedia Messaging Service</i> (Sistema de Mensagens Multimídia)
SMS	<i>Short Message Service</i> (Serviço de Mensagem Curta)
SGBD	Sistema de Gerencialmente de Banco de Dados
ACID	Atomicidade, Consistência, Isolamento e Durabilidade
UML	<i>Unified Modeling Language</i> (Linguagem de Modelagem Unificada)
WBS	<i>Work Breakdown Structure</i> (Estrutura Analítica de Projeto)

SUMÁRIO

1. INTRODUÇÃO.....	14
1.1. OBJETIVOS.....	15
1.2. JUSTIFICATIVAS	15
1.3. MOTIVAÇÃO	17
2. TECNOLOGIAS E FERRAMENTAS DE DESENVOLVIMENTO	19
2.1. TECNOLOGIA JAVA	19
2.2. TECNOLOGIA GOOGLE ANDROID	21
2.3. BANCO DE DADOS EMBARCADO SQLITE	22
2.3.1. Principais Funcionalidades	23
3. ANÁLISE E ESPECIFICAÇÃO DO SISTEMA.....	24
3.1. MAPA MENTAL	24
3.2. LISTA DE REQUISITOS.....	25
3.3. DIAGRAMA E ESPECIFICAÇÃO DE CASOS DE USO	26
3.4. DIAGRAMA DE ATIVIDADES	35
3.5. DIAGRAMA DE CLASSES.....	41
3.6. MODELO ENTIDADE-RELACIONAMENTO	42
4. ESTRUTURA DO PROJETO.....	43
4.1. ESTRUTURA ANALÍTICA DE TRABALHO [ou WORK BREAKDOWN STRUCTURE (WBS)]	43
4.2. ORÇAMENTO	45
5. IMPLEMENTAÇÃO DO PROJETO	46
5.1. ORGANIZAÇÃO ESTRUTURAL DO PROJETO GOOGLE ANDROID	46
5.2. INTERFACES DA APLICAÇÃO MÓVEL	47

5.3. TRECHOS DE IMPLEMENTAÇÃO.....	51
5.3.1. Implementação da Classe de Acesso e Definição do Banco de Dados	52
6. CONCLUSÃO.....	55
6.1. TRABALHOS FUTUROS	55
REFERÊNCIAS.....	56

1. INTRODUÇÃO

Em várias ocasiões não é possível comparecer aos eventos, reuniões e demais compromissos na hora agendada por variados motivos, o que pode ser e será mau visto por diversas pessoas.

Os atrasos são uma demonstração de desorganização na vida da pessoa, uma falta de planejamento na sua rotina, um descaso com as demais pessoas que compareceriam ao compromisso, a não ser que aconteçam por motivos que independem de si mesmo, como por exemplo, acidentes, doenças, dentre outros possíveis motivos, ou seja, os imprevistos.

De maneira contígua ao avanço da tecnologia mundial e a globalização, surge à falta de tempo para organizar os horários e eventos, devido ao fato de muitas pessoas não saberem como organizá-los e esquecerem-se de onde anotaram as informações que fazem referência aos mesmos.

Com base nas afirmações acima citadas, o aplicativo *SmartAgenda* surge com o intuito de findar os atrasos em eventos e também o esquecimento dos mesmos, proporcionando ao usuário uma interação rápida e prática com uma agenda virtual, para que o mesmo possa registrar todas as informações pertinentes aos seus compromissos, além de possuir dois relatórios, Eventos Agendados e Eventos Realizados, listando respectivamente os eventos que e as principais informações daqueles compromissos que estão para acontecer e também uma relação geral dos já realizados.

O *SmartAgenda* é um aplicativo móvel e consiste na disponibilização do agendamento de quaisquer tipos de eventos e compromissos, com divisões de intervalo de tempo a serem pré-definidas pelo usuário, variando de minutos a horas para uma melhor adequação à necessidade de quem estiver fazendo uso da aplicação e será descrito no item 2.3.1, aonde serão descritas e melhor explicadas as principais funcionalidades deste aplicativo.

1.1. OBJETIVOS

São muitos os objetivos deste trabalho, mas como principal pode ser ressaltado a intenção de que o software *SmartAgenda* está sendo desenvolvido para tornar-se líder no mercado de agendamento de eventos, baseando-se em soluções rápidas e práticas possibilitando um agendamento dos compromissos diários com uma maior praticidade e facilidade. O aplicativo irá proporcionar uma considerável diminuição nos atrasos, fazendo com que o este possa vir a tornar-se referência nesse segmento, por possuir uma interface simples e de fácil utilização e compreensão, possibilitando a qualquer usuário que possuir um Smartphone, fazer uso sem nenhuma complicação no entendimento dos recursos oferecidos e no entendimento dos relatórios gerados pelo mesmo.

Além deste objetivo principal, podem-se destacar diversos outros objetivos, que não deixam de ser importantes. Dentre estes demais objetivos, destacam-se:

- O estudo da plataforma Google Android.
- Modelar e Programar a arquitetura de uma aplicação móvel.
- Facilitar agendamento de eventos.

1.2. JUSTIFICATIVAS

A justificativa para a decisão de desenvolver o aplicativo *SmartAgenda* na plataforma móvel se dá pela considerável expansão do uso de Smartphones, os quais proporcionam um acesso mais rápido e mais fácil às necessidades que surgem em momentos inesperados no cotidiano das pessoas, dando aos aplicativos desenvolvidos nessa plataforma grandes chances de ganharem espaço no mercado de aplicativos móveis.

Segundo uma pesquisa feita pela empresa Ericsson, o mundo vai contar com cerca de 3,3 bilhões de Smartphones até 2018, sendo três vezes mais do que a atual situação mundial, atualmente os Smartphones representam cerca de apenas 15% a 20% dos celulares no mundo, parcela que aumentará consideravelmente após a triplicação de ‘Smart Users’ (Usuários Inteligentes) como são chamados os usuários de Smartphones.

Um recente estudo feito pela “Ipsos Media CT” comprova que apenas 14% da população brasileira possui Smartphones, 27 milhões de pessoas, número mesmo que pequeno, mas considerável tendo em vista que o primeiro Smartphone comercializado foi o T-Mobile G1 lançado em 2008, ou seja, em cinco anos os Smartphones atingem esse âmbito bastante considerável de usuários.

Esse aumento de usuários de Smartphone, somado ao aumento da comercialização e ao avanço tecnológico podem ser considerados exemplo explícitos de que o avanço da globalização aumenta gradativamente e exponencialmente a cada dia que se passa, causando a geração de novos empregos e novas oportunidades para que surjam novos e visionários desenvolvedores. Proveniente do intuito de findar a falta de tempo para organizar e manipular os horários, para acabar com o esquecimento de compromissos e proporcionar um melhor aproveitamento e melhor visualização dos compromissos, o *SmartAgenda* tem por finalidade gerar uma comodidade e uma facilidade para manipular os horários, fornecer auxílio para pessoas que não tem tempo ou entendimento de como manejar seus horários, além de fazer com que seu usuário torne-se mais pontual, fazendo com que se torne aparentemente mais confiável e responsável para os demais. Possuindo uma interface simples de entender e interagir, espera-se que essa aplicação ganhe espaço no mercado.

Em suma, o desenvolvimento do software *SmartAgenda* buscará atender as exigências do mercado para que possa se tornar em curto prazo o pioneiro neste mais novo nicho de mercado, as “AppStore”, lojas virtuais para a comercialização ou divulgação dos arquivos desenvolvidos nas diversas linguagens móveis de programação, sendo uma delas o foco deste projeto, a linguagem Google Android, que será descrita no item 2.2, trazendo a história e as principais informações da tecnologia Google Android.

1.3. MOTIVAÇÃO

A motivação para o desenvolvimento do software *SmartAgenda* em plataforma móvel se dá pela considerável expansão do uso de Smartphones, os quais proporcionam um acesso mais rápido e mais fácil às necessidades que surgem em momentos inesperados no cotidiano das pessoas, dando aos aplicativos desenvolvidos nesse ambiente, grandes chances de ganharem espaço no mercado.

Segundo uma pesquisa da empresa Ericsson, o mundo vai contar com cerca de 3,3 bilhões de Smartphones até 2018, sendo três vezes mais do que a atual situação mundial, atualmente os Smartphones representam cerca de apenas 15% a 20% dos celulares no mundo, parcela que aumentará consideravelmente após a triplicação de 'Smart Users' (Usuários Inteligentes) como são chamados os usuários de Smartphones.

Um recente estudo feito pela "Ipsos Media CT" comprova que apenas 14% da população brasileira possui Smartphones, 27 milhões de pessoas, número mesmo que pequeno, mas considerável tendo em vista que o primeiro Smartphone comercializado foi o T-Mobile G1 lançado em 2008, ou seja, em cinco anos os Smartphones atingem esse âmbito de usuários.

Com intuito de findar a falta de tempo para organizar e manipular horários, o esquecimento de compromissos e proporcionar um melhor aproveitamento do dia, o *SmartAgenda* tem por finalidade gerar uma comodidade e fornecer auxílio para pessoas que não tem tempo para manejar seus horários, além de fazer com que seu usuário torne-se mais confiável ao olhar dos demais. Possuindo uma interface simples de entender e interagir, espera-se que a aplicação ganhe espaço no mercado.

Além desse fator a tecnologia JAVA vem sendo a mais utilizada desde 2005 e cresce consideravelmente o uso da mesma a cada ano que se passa segundo pesquisa do índice TIOBE.

Em suma, o desenvolvimento do software *SmartAgenda* buscará atender as exigências do mercado para que possa se tornar em curto prazo um pioneiro neste

nicho de mercado, os 'Apps' (*Applications*) desenvolvidos com a linguagem JAVA Google Android.

2. TECNOLOGIAS E FERRAMENTAS DE DESENVOLVIMENTO

O *SmartAgenda* é um aplicativo móvel que será desenvolvido na linguagem de programação Google Android, que é uma nova tecnologia desenvolvida com o intuito de padronizar os sistemas operacionais móveis e proporcionar uma maior facilidade ao se tentar compartilhar e reutilizar aplicações e recursos entre diferentes aparelhos móveis foi comprado em 2005 pela Google da empresa “Android INC.” e tornou-se open source a partir de 2008 para justamente disseminar a padronização dos aplicativos. Seu ambiente de desenvolvimento será baseado na IDE Eclipse, que foi desenvolvida sobre a plataforma Java, originalmente desenvolvida pela IBM e que possui atualmente seu código aberto, é a plataforma mundialmente mais utilizada para desenvolvimento Java, pois facilita a programação por ter o desenvolvimento baseado em plug-ins, além de possibilitar a instalação de mais bibliotecas e API’s auxiliando na conclusão do projeto e na diversidade dos aplicativos desenvolvidos, mas mantendo uma padronização ao mesmo tempo.

O Banco de Dados escolhido para gerenciar o armazenamento das informações dos eventos é o SQLite Database, por ser embarcado a tecnologia e pelo mesmo possibilitar a utilização da de instruções SQL em Linguagem de Definição de Dados (DDL) e Linguagem de Manipulação de Dados (DML). Além deste ponto, pode-se ressaltar que esta funcionalidade embarcada não necessita de configuração alguma para ser utilizada e também possibilita a criação de Indexes e Triggers.

2.1. TECNOLOGIA JAVA

Java é uma linguagem de programação orientada a objetos desenvolvida em meados dos anos 90 por uma equipe de desenvolvedores liderada por James Gosling, na Sun Microsystems. Difere-se das linguagens convencionais por ser compilada para um bytecode e executada por uma máquina virtual. Em 1991 iniciou-se o Green Project, conhecido como o berço da tecnologia Java e seus mentores foram Patrick Naughton, Mike Sheridan e James Gosling.

Desde o lançamento da plataforma Java em 1995, tornou-se rapidamente a mais utilizada em comparação com qualquer outra linguagem de programação na história da computação. Em 2004 chegou à marca de três milhões de desenvolvedores por todo o mundo, e a tecnologia não parou por aí, continuou se desenvolvendo e hoje em dia é a pioneira no mercado de desenvolvimento de softwares. E foi o estático HTML dos navegadores que promoveu a rápida disseminação da tecnologia Java tornando-a popular pelo uso na internet. Possui hoje em dia seu ambiente de execução em navegadores, mainframes, sistemas operacionais, celulares, cartões inteligentes, tablets dentre outros.

A tecnologia Java foi projetada tendo por objetivo acoplar: a orientação a objetos; a portabilidade, com o intuito de programar uma única vez e poder ser executada em qualquer plataforma; recursos de rede, possuindo bibliotecas internas que facilitam a comunicação com os protocolos TCP/IP; segurança, por ser executada em sua própria máquina virtual, a JVM (Java Virtual Machine). Além disso, pode-se destacar sua facilidade na internacionalização, por suportar nativamente caracteres Unicode, que é um padrão que permite aos computadores representar os textos de qualquer sistema de escrita existente e também pela sua desalocação de memória automática executada por seu coletor de lixo, que vai destruindo o que não é mais utilizado liberando desta forma mais espaço na memória.

Atualmente a Sun Microsystems disponibiliza a maioria das distribuições Java gratuitamente, deixando para obter lucros com programas mais especializados.

Há ainda a possibilidade de utilizar frameworks para facilitar o desenvolvimento das aplicações. Dentre os frameworks mais conhecidos destacam-se o Hibernate, que facilita o mapeamento dos atributos entre uma base tradicional de dados relacionais e o modelo objeto de uma aplicação, mediante o uso de arquivos (XML) ou anotações Java; JUnit, que fornece suporte à criação de testes automatizados na linguagem de programação Java, onde seu intuito é basicamente verificar se cada método de uma classe funciona da forma esperada, exibindo possíveis erros ou falhas, além de o programador ter a possibilidade de usar esta ferramenta para criar um modelo padrão de testes, muitas vezes de forma automatizada.

É possível desenvolver aplicações em diversos ambientes de desenvolvimento integrado, que é um programa de computador que reúne características e

ferramentas de apoio ao desenvolvimento de software com o objetivo de otimizar o processo, visando a maior produtividade dos desenvolvedores. Dentre as IDE's disponíveis, destacam-se: Eclipse, um projeto aberto iniciado pela IBM; NetBeans, desenvolvido pela Sun Microsystems; JDeveloper, desenvolvido pela Oracle dentre outros.

2.2. TECNOLOGIA GOOGLE ANDROID

Android é um sistema operacional móvel baseado em um núcleo Linux e é desenvolvido pelo Open Headset Alliance, liderada pelo Google e outras empresas. Segundo pesquisa do Google cerca de 200 mil aparelhos celulares são vendidos a cada três dias.

Em Julho de 2005 o Google adquiriu a empresa Android Inc., uma empresa situada em Palo Alto, Califórnia, EUA. Inicialmente conduzida por Andy Rubin, uma plataforma móvel foi desenvolvida baseada em Linux, com o intuito de ser uma plataforma flexível, aberta e de fácil migração para os desenvolvedores. Em meados de 2008, a empresa disponibiliza o primeiro telefone com Android, o HTC Dream que enquanto elogiado por seu hardware robusto, a introdução do sistema operacional Android obteve algumas críticas pela falta de algumas funcionalidades e software de terceiros em relação a plataformas concorrentes, como Black Berry, iPhone e Symbian. Entretanto ainda assim foi considerado inovador, por ser gratuito e pelo sistema de notificações e integração pesada com os serviços do Google.

Com o lançamento do SDK, que é a sigla de Software Development Kit, ou seja, Kit de Desenvolvimento de Software ou Kit de Desenvolvimento de Aplicativos que são disponibilizados por empresas ou projetos opensource para que programadores externos tenham uma melhor integração com o software proposto, tornam as características e especificações para o Android facilmente distribuídas. A plataforma é adaptada tanto para VGA e gráficos em 2D como também para gráficos disponibilizados em 3D baseados em OpenGL ES, que é uma subseção da API da

biblioteca de gráficos tridimensionais OpenGL projetada para sistemas embarcados como telefones celulares, PDAs e consoles de vídeo games.

O Banco de Dados utilizado pela tecnologia Android é o SQLite, que vem embarcado a tecnologia e pelo mesmo possibilitar a utilização da de instruções SQL em Linguagem de Definição de Dados (DDL) e Linguagem de Manipulação de Dados (DML). Além deste fato, pode-se ressaltar que o mesmo não necessita de configuração alguma para ser utilizado e também possibilita a criação de Indexes e Triggers.

O padrão de envio de mensagens utilizado pode ser o MMS como também o SMS e o navegador utilizado são baseados no framework opensource WebKit.

O sistema multimídia suportado de áudio e vídeo são MPEG-4, H.264, MP3 e AAC, sendo também capaz de fazer uso de câmeras de vídeo, touchscreen, GPS.

2.3. BANCO DE DADOS EMBARCADO SQLITE

SQLite é uma biblioteca que implementa um banco de dados SQL embutido, os programas que usam a biblioteca SQLite podem ter acesso a banco de dados SQL sem executar um processo SGBD separado, processo este que é responsável por retirar da aplicação cliente a responsabilidade de gerenciar o acesso, manipulação e organização dos dados. SQLite não é uma biblioteca cliente usada para conectar com um grande servidor de banco de dados, mas sim o próprio servidor. A biblioteca SQLite lê e escreve diretamente o arquivo no banco de dados no disco.

O uso do SQLite é recomendado para aplicações que requerem mais simplicidade de administração, implementação e manutenção, aplicações essas que não necessitam de incontáveis recursos e funcionalidades dos SGBD's, onde estes por sua vez são voltados para aplicativos bem mais complexos. Alguns exemplos de onde o SQLite é aplicado são: Sites com menos de cem mil requisições diárias; sistemas embarcados; ferramentas de estatística e análise.

O SQLite é um software livre e multiplataforma, além de ser um mecanismo de armazenamento seguro com transações ACID, é a junção de Atomicidade, Consistência, Isolamento e Durabilidade, conceito este utilizado para garantir a integridade dos dados, não necessitando também de instalação, configuração ou administração. O SQLite não depende de nenhuma condição externa para sua execução, suporta bases de dados abaixo de dois terabytes e permite o armazenamento do banco de dados em um único arquivo.

Uma ferramenta muito utilizada para a manipulação do banco de dados SQLite é o Database Master - SQLite Manager, disponibilizado pela Nucleon Software, entretanto não é um software com licença gratuita.

2.3.1. Principais Funcionalidades

Dentre suas principais funcionalidades, destacam-se o fato de não necessitar de nenhuma ferramenta SGBD, tendo em vista que ele mesmo é o servidor, podendo desta forma ler e escrever dados diretamente no banco de dados. Além de possuir internamente a possibilidade de utilizar das instruções SQL em Linguagem de Definição de Dados (DDL) e Linguagem de Manipulação de Dados (DML).

3. ANÁLISE E ESPECIFICAÇÃO DO SISTEMA

A Análise é considerada por muitos estudiosos como a base de pensamentos com o objetivo de encontrar o caminho mais fácil, rápido e prático para o desenvolvimento do sistema, de modo que o código torne-se mais estruturado, tendo em vista que a principal função da Análise do Sistema é diminuir os possíveis erros e sanar a inconsistência dos dados.

Nos próximos tópicos as ferramentas mais utilizadas para analisar o sistema serão utilizadas e também explicadas, para que possa haver um melhor entendimento do modo com que o sistema irá agir.

3.1. MAPA MENTAL

Para uma visualização geral do projeto pode-se utilizar várias ferramentas que servem de auxílio ao desenvolvedor, um exemplo bastante comum é o “Mapa Mental” (Mind Map), ideia originalmente desenvolvida pelo britânico Tony Buzzan, dois de Junho de 1942.

O mapa mental serve de ferramenta para explanar melhor as ideias do projeto e serve de caminho a se seguir para o desenvolvimento, assim como as técnicas da linguagem padrão para auxílio na estruturação de um software, a UML (Unified Modeling Language), linguagem que será explicada logo abaixo.

Na imagem abaixo, o aplicativo *SmartAgenda* poderá ser visualizado e será mais bem descrito através do Mapa Mental.

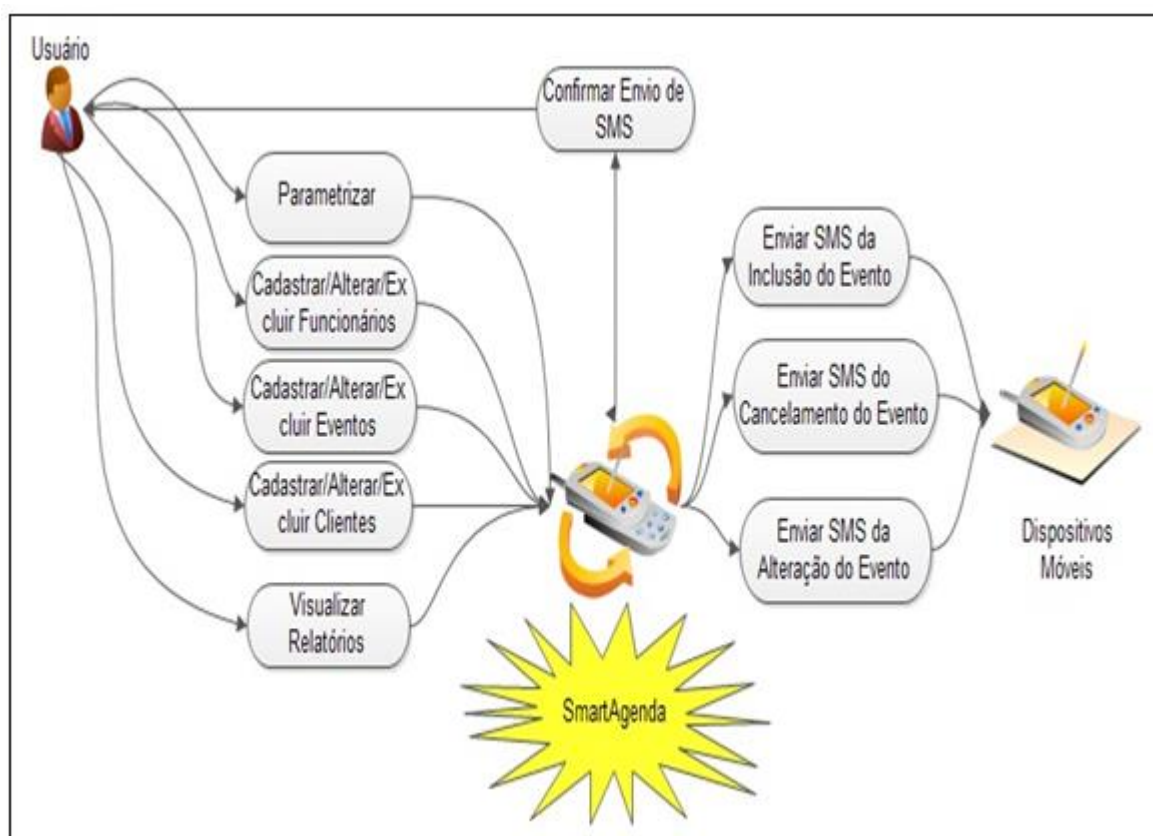


Figura 1 - Mapa Mental

3.2. LISTA DE REQUISITOS

O levantamento das solicitações dos usuários e a disposição das mesmas em forma de uma lista de requisitos é uma das partes mais cruciais da análise para o desenvolvimento de um sistema, tendo em vista que na mesma estarão descritos as partes ressaltadas pelo usuário, em um grau de necessidade enfatizada pelo mesmo.

Logo abaixo será apresentada a lista de requisitos do aplicativo *SmartAgenda*, serão dispostos em grau de importância, de cima para baixo.

- Inserir dados iniciais;
- Escolher horário do evento;
- Inserir dados do evento;

- Visualizar/Alterar os dados do evento;
- Exibir relatório de eventos realizados/cancelados/pendentes;
- Cadastrar Funcionários;
- SMS de inclusão/alteração/exclusão de eventos;
- Cadastrar Clientes;
- Utilização de Alarme para lembrar-se dos eventos;
- Exibir a confirmação do envio de SMS com uma notificação.

Sendo estes os requisitos do sistema, nos próximos tópicos serão apresentados diversos diagramas, para uma melhor disposição do comportamento do sistema.

3.3. DIAGRAMA E ESPECIFICAÇÃO DE CASOS DE USO

O Diagrama de Caso de Uso é uma disposição do Ator, interpreta o papel do usuário no sistema, e os casos de uso que representam os recursos disponíveis no sistema. É uma melhor representação visual de quais funcionalidades o usuário terá ao seu dispor no aplicativo.

Abaixo serão exibidos os casos de uso referente ao cenário do Usuário e o aplicativo *SmartAgenda*, onde as ações de cada serão expostas.

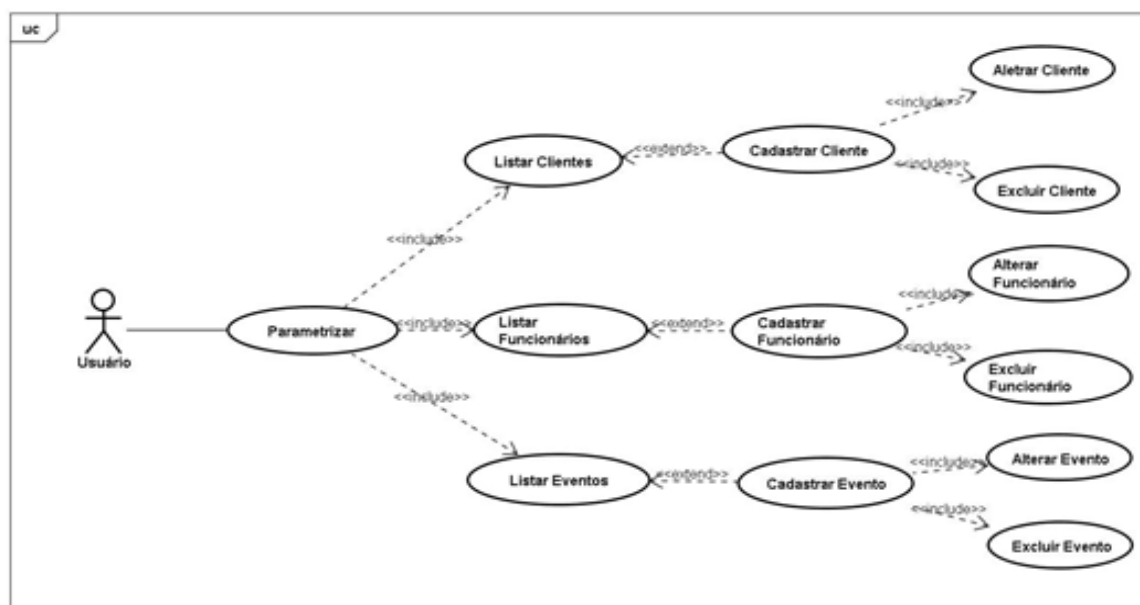


Figura 2 - Diagrama de Caso de Uso - *SmartAgenda*

Nome do Caso de Uso	Parametrizar
Finalidade	Tem por finalidade proporcionar ao usuário a oportunidade de inserir as configurações desejadas em seu sistema
Atores	Usuário
Pré-condições	Ter inserido seu usuário e senha no sistema
Evento Inicial	Acessar a opção no menu principal "Parametrização"
Fluxo Principal	<ol style="list-style-type: none"> 1- Instalar o aplicativo; 2- Executar o aplicativo; 3- Escolher a opção "Parametrização" no menu principal; 4- Confirmar as configurações inseridas; 5- FIM.
Fluxo Alternativo	Confirmar configurações inseridas: usuário deve confirmar as configurações que foram previamente definidas por ele mesmo, caso contrário o sistema retorna ao passo 4.
Fluxo de Exceção	Sistema deixa de responder no momento de inserir as configurações.
Pós-condições	Após a mensagem de configurações aplicadas com êxito, o sistema apresentará o menu principal
Casos de Teste	<ul style="list-style-type: none"> • Testar se configurações foram realmente aplicadas e salvas; • Testar se o usuário fechar e abrir o sistema o mesmo não solicitará as configurações já definidas.

Tabela 1 - Especificação de Caso de Uso - Parametrizar

Nome do Caso de Uso	Listar Clientes
Finalidade	Tem por finalidade proporcionar ao usuário uma listagem dos clientes cadastrados no sistema
Atores	Usuário
Pré-condições	Ter escolhido a opção para listar os clientes
Evento Inicial	Pressionar o botão “Clientes” no menu principal
Fluxo Principal	1- Instalar o aplicativo;
	2- Executar o aplicativo;
	3- Escolher a opção “Clientes” no menu principal;
	4- FIM.
Fluxo Alternativo	Não há.
Fluxo de Exceção	Sistema deixa de responder no momento em que o usuário pressiona o botão “Clientes” no menu principal.
Pós-condições	Após pressionar o botão “Clientes” o sistema exibe os já cadastrados.
Casos de Teste	<ul style="list-style-type: none"> •Testar se os clientes serão listados; •Testar um cliente poderá ou não ser excluído.

Tabela 2 - Especificação de Caso de Uso - Listar Clientes

Nome do Caso de Uso	Listar Funcionários
Finalidade	Tem por finalidade proporcionar ao usuário uma listagem dos funcionários cadastrados no sistema
Atores	Usuário
Pré-condições	Ter escolhido a opção para listar os funcionários
Evento Inicial	Pressionar o botão “Funcionários” no menu principal
Fluxo Principal	1- Instalar o aplicativo;
	2- Executar o aplicativo;
	3- Escolher a opção “Funcionários” no menu principal;
	4- FIM.
Fluxo Alternativo	Não há.
Fluxo de Exceção	Sistema deixa de responder no momento em que o usuário pressiona o botão “Funcionários” no menu principal.
Pós-condições	Após pressionar o botão “Funcionários” o sistema exibe os já cadastrados.
Casos de Teste	<ul style="list-style-type: none"> •Testar se os funcionários serão listados; •Testar um funcionário poderá ou não ser excluído.

Tabela 3 - Especificação de Caso de Uso - Listar Funcionários

Nome do Caso de Uso	Listar Eventos
Finalidade	Tem por finalidade proporcionar ao usuário uma listagem dos eventos cadastrados no sistema
Atores	Usuário
Pré-condições	Ter escolhido a opção para listar os eventos
Evento Inicial	Pressionar o botão “Eventos” no menu principal
Fluxo Principal	1- Instalar o aplicativo;
	2- Executar o aplicativo;
	3- Escolher a opção “Eventos” no menu principal;
	4- FIM.
Fluxo Alternativo	Não há.
Fluxo de Exceção	Sistema deixa de responder no momento em que o usuário pressiona o botão “Eventos” no menu principal.
Pós-condições	Após pressionar o botão “Eventos” o sistema exibe os já cadastrados.
Casos de Teste	<ul style="list-style-type: none"> •Testar se os eventos serão listados; •Testar um evento poderá ou não ser excluído.

Tabela 4 - Especificação de Caso de Uso - Listar Eventos

Nome do Caso de Uso	Cadastrar Funcionários
Finalidade	Tem por finalidade proporcionar ao usuário a opção de cadastrar seus Funcionários
Atores	Usuário
Pré-condições	Ter escolhido a opção para cadastrar os funcionários
Evento Inicial	Pressionar o botão “Menu” do dispositivo móvel e escolher a opção “Incluir”
Fluxo Principal	1- Instalar o aplicativo;
	2- Executar o aplicativo;
	3- Escolher a opção “Funcionários” no menu principal;
	4- Pressionar o botão “Menu” do dispositivo móvel e escolher a opção “Incluir”;
	5- Inserir informações do funcionário a ser cadastrado;
	6- Confirmar a inclusão do funcionário;
	7- FIM.
Fluxo Alternativo	O usuário pode optar por não confirmar a inclusão do registro e o sistema retorna ao passo 3;
Fluxo de Exceção	<p>Sistema deixa de responder no momento em que o usuário pressiona o botão “Funcionários” no menu principal e retorna ao passo 2;</p> <p>Sistema deixa de responder no momento em que o usuário confirma o registro a ser incluído e retorna ao passo 2;</p>

Pós-condições	Após confirmar a inclusão do registro o sistema exibe a lista dos funcionários.
Casos de Teste	<ul style="list-style-type: none"> •Testar se os funcionários serão listados; •Testar um funcionário será realmente incluído;

Tabela 5 - Especificação de Caso de Uso - Cadastrar Funcionários

Nome do Caso de Uso	Cadastrar Clientes
Finalidade	Tem por finalidade proporcionar ao usuário a opção de cadastrar seus Clientes
Atores	Usuário
Pré-condições	Ter escolhido a opção para cadastrar os clientes
Evento Inicial	Pressionar o botão “Menu” do dispositivo móvel e escolher a opção “Incluir”
Fluxo Principal	1- Instalar o aplicativo;
	2- Executar o aplicativo;
	3- Escolher a opção “Clientes” no menu principal;
	4- Pressionar o botão “Menu” do dispositivo móvel e escolher a opção “Incluir”;
	5- Inserir informações do cliente a ser cadastrado;
	6- Confirmar a inclusão do cliente;
	7- FIM.
Fluxo Alternativo	O usuário pode optar por não confirmar a inclusão do registro e o sistema retorna ao passo 3;
Fluxo de Exceção	Sistema deixa de responder no momento em que o usuário pressiona o botão “Clientes” no menu principal e retorna ao passo 2; Sistema deixa de responder no momento em que o usuário confirma o registro a ser incluído e retorna ao passo 2;
Pós-condições	Após confirmar a inclusão do registro o sistema exibe a lista dos clientes.
Casos de Teste	<ul style="list-style-type: none"> •Testar se os clientes serão listados; •Testar um cliente será realmente incluído;

Tabela 6 - Especificação de Caso de Uso - Cadastrar Clientes

Nome do Caso de Uso	Cadastrar Eventos
Finalidade	Tem por finalidade proporcionar ao usuário a opção de cadastrar seus Eventos
Atores	Usuário
Pré-condições	Ter escolhido a opção para cadastrar os eventos
Evento Inicial	Pressionar o botão “Menu” do dispositivo móvel e escolher a opção “Incluir”
Fluxo Principal	1- Instalar o aplicativo;
	2- Executar o aplicativo;
	3- Escolher a opção “Eventos” no menu principal;
	4- Pressionar o botão “Menu” do dispositivo móvel e escolher a opção “Incluir”;
	5- Inserir informações do evento a ser cadastrado;
	6- Confirmar a inclusão do evento;
	7- FIM.
Fluxo Alternativo	O usuário pode optar por não confirmar a inclusão do registro e o sistema retorna ao passo 3;
Fluxo de Exceção	Sistema deixa de responder no momento em que o usuário pressiona o botão “Eventos” no menu principal e retorna ao passo 2; Sistema deixa de responder no momento em que o usuário confirma o registro a ser incluído e retorna ao passo 2;
Pós-condições	Após confirmar a inclusão do registro o sistema exibe a lista dos eventos.
Casos de Teste	<ul style="list-style-type: none"> • Testar se os eventos serão listados; • Testar um evento será realmente incluído;

Tabela 7 - Especificação de Caso de Uso - Cadastrar Eventos

Nome do Caso de Uso	Alterar Funcionários
Finalidade	Tem por finalidade proporcionar ao usuário a opção de alterar seus funcionários previamente cadastrados
Atores	Usuário
Pré-condições	Ter escolhido a opção para alterar os funcionários
Evento Inicial	Pressionar o botão “Menu” do dispositivo móvel e escolher a opção “Alterar”
Fluxo Principal	1- Instalar o aplicativo;
	2- Executar o aplicativo;
	3- Escolher a opção “Funcionários” no menu principal;
	4- Pressionar o botão “Menu” do dispositivo móvel e escolher a opção “Alterar”;
	5- Inserir informações alteradas do funcionário cadastrado;

	6- Confirmar a alteração do funcionário; 7- FIM.
Fluxo Alternativo	O usuário pode optar por não confirmar a alteração do registro e o sistema retorna ao passo 3;
Fluxo de Exceção	Sistema deixa de responder no momento em que o usuário pressiona o botão “Funcionários” no menu principal e retorna ao passo 2; Sistema deixa de responder no momento em que o usuário confirma o registro a ser alterado e retorna ao passo 2;
Pós-condições	Após confirmar a alteração do registro o sistema exibe a lista dos funcionários.
Casos de Teste	•Testar se os funcionários serão listados; •Testar um funcionário será realmente alterado;

Tabela 8 - Especificação de Caso de Uso - Alterar Funcionários

Nome do Caso de Uso	Alterar Clientes
Finalidade	Tem por finalidade proporcionar ao usuário a opção de alterar seus clientes previamente cadastrados
Atores	Usuário
Pré-condições	Ter escolhido a opção para alterar os clientes
Evento Inicial	Pressionar o botão “Menu” do dispositivo móvel e escolher a opção “Alterar”
Fluxo Principal	1- Instalar o aplicativo;
	2- Executar o aplicativo;
	3- Escolher a opção “Clientes” no menu principal;
	4- Pressionar o botão “Menu” do dispositivo móvel e escolher a opção “Alterar”;
	5- Inserir informações alteradas do cliente cadastrado;
	6- Confirmar a alteração do cliente;
	7- FIM.
Fluxo Alternativo	O usuário pode optar por não confirmar a alteração do registro e o sistema retorna ao passo 3;
Fluxo de Exceção	Sistema deixa de responder no momento em que o usuário pressiona o botão “Clientes” no menu principal e retorna ao passo 2; Sistema deixa de responder no momento em que o usuário confirma o registro a ser alterado e retorna ao passo 2;
Pós-condições	Após confirmar a alteração do registro o sistema exibe a lista dos clientes.
Casos de Teste	•Testar se os clientes serão listados; •Testar um cliente será realmente alterado;

Tabela 9 - Especificação de Caso de Uso - Alterar Clientes

Nome do Caso de Uso	Alterar Eventos
Finalidade	Tem por finalidade proporcionar ao usuário a opção de alterar seus eventos previamente cadastrados
Atores	Usuário
Pré-condições	Ter escolhido a opção para alterar os eventos
Evento Inicial	Pressionar o botão “Menu” do dispositivo móvel e escolher a opção “Alterar”
Fluxo Principal	1- Instalar o aplicativo;
	2- Executar o aplicativo;
	3- Escolher a opção “Eventos” no menu principal;
	4- Pressionar o botão “Menu” do dispositivo móvel e escolher a opção “Alterar”;
	5- Inserir informações alteradas do evento cadastrado;
	6- Confirmar a alteração do evento;
	7- FIM.
Fluxo Alternativo	O usuário pode optar por não confirmar a alteração do registro e o sistema retorna ao passo 3;
Fluxo de Exceção	Sistema deixa de responder no momento em que o usuário pressiona o botão “Eventos” no menu principal e retorna ao passo 2; Sistema deixa de responder no momento em que o usuário confirma o registro a ser alterado e retorna ao passo 2;
Pós-condições	Após confirmar a alteração do registro o sistema exibe a lista dos eventos.
Casos de Teste	<ul style="list-style-type: none"> •Testar se os eventos serão listados; •Testar um eventos será realmente alterado;

Tabela 10 - Especificação de Caso de Uso - Alterar Eventos

Nome do Caso de Uso	Excluir Funcionários
Finalidade	Tem por finalidade proporcionar ao usuário a opção de excluir seus funcionários previamente cadastrados
Atores	Usuário
Pré-condições	Ter escolhido a opção para excluir os funcionários
Evento Inicial	Selecionar o funcionário e clicar no botão “Menu” do dispositivo móvel e escolher a opção “Excluir”
Fluxo Principal	1- Instalar o aplicativo;
	2- Executar o aplicativo;
	3- Escolher a opção “Funcionários” no menu principal;
	4- Selecionar o funcionário e clicar no botão “Menu” do dispositivo móvel e escolher a opção “Excluir”;

	5- Confirmar a exclusão do funcionário; 6- FIM.
Fluxo Alternativo	O usuário pode optar por não confirmar a exclusão do registro e o sistema retorna ao passo 4;
Fluxo de Exceção	Sistema deixa de responder no momento em que o usuário pressiona o botão “Funcionários” no menu principal e retorna ao passo 2; Sistema deixa de responder no momento em que o usuário confirma o registro a ser excluído e retorna ao passo 2;
Pós-condições	Após confirmar a exclusão do registro o sistema exibe a lista dos funcionários restantes.
Casos de Teste	•Testar se os funcionários restantes serão listados; •Testar o funcionário será realmente excluído;

Tabela 11 - Especificação de Caso de Uso - Excluir Funcionários

Nome do Caso de Uso	Excluir Clientes
Finalidade	Tem por finalidade proporcionar ao usuário a opção de excluir seus clientes previamente cadastrados
Atores	Usuário
Pré-condições	Ter escolhido a opção para excluir os clientes
Evento Inicial	Selecionar o cliente e clicar no botão “Menu” do dispositivo móvel e escolher a opção “Excluir”
Fluxo Principal	1- Instalar o aplicativo;
	2- Executar o aplicativo;
	3- Escolher a opção “Clientes” no menu principal;
	4- Selecionar o cliente e clicar no botão “Menu” do dispositivo móvel e escolher a opção “Excluir”;
	5- Confirmar a exclusão do cliente;
	6- FIM.
Fluxo Alternativo	O usuário pode optar por não confirmar a exclusão do registro e o sistema retorna ao passo 4;
Fluxo de Exceção	Sistema deixa de responder no momento em que o usuário pressiona o botão “Clientes” no menu principal e retorna ao passo 2; Sistema deixa de responder no momento em que o usuário confirma o registro a ser excluído e retorna ao passo 2;
Pós-condições	Após confirmar a exclusão do registro o sistema exibe a lista dos clientes restantes.
Casos de Teste	•Testar se os clientes restantes serão listados; •Testar o cliente será realmente excluído;

Tabela 12 - Especificação de Caso de Uso - Excluir Clientes

Nome do Caso de Uso	Excluir Eventos
Finalidade	Tem por finalidade proporcionar ao usuário a opção de excluir seus eventos previamente cadastrados
Atores	Usuário
Pré-condições	Ter escolhido a opção para excluir os eventos
Evento Inicial	Selecionar o evento e clicar no botão “Menu” do dispositivo móvel e escolher a opção “Excluir”
Fluxo Principal	1- Instalar o aplicativo;
	2- Executar o aplicativo;
	3- Escolher a opção “Eventos” no menu principal;
	4- Selecionar o evento e clicar no botão “Menu” do dispositivo móvel e escolher a opção “Excluir”;
	5- Confirmar a exclusão do evento;
	6- FIM.
Fluxo Alternativo	O usuário pode optar por não confirmar a exclusão do registro e o sistema retorna ao passo 4;
Fluxo de Exceção	Sistema deixa de responder no momento em que o usuário pressiona o botão “Eventos” no menu principal e retorna ao passo 2; Sistema deixa de responder no momento em que o usuário confirma o registro a ser excluído e retorna ao passo 2;
Pós-condições	Após confirmar a exclusão do registro o sistema exibe a lista dos eventos restantes.
Casos de Teste	<ul style="list-style-type: none"> •Testar se os eventos restantes serão listados; •Testar o evento será realmente excluído;

Tabela 13 - Especificação de Caso de Uso - Excluir Eventos

3.4. DIAGRAMA DE ATIVIDADES

Abaixo serão apresentados os diagramas de atividades, esses diagramas descrevem de qual forma a funcionalidade apresentada nos diagramas de Caso de Uso (3.3) irão se portar, quais serão os caminhos a seguir no sistema, sendo um modo visual da Especificação do Caso de Uso.

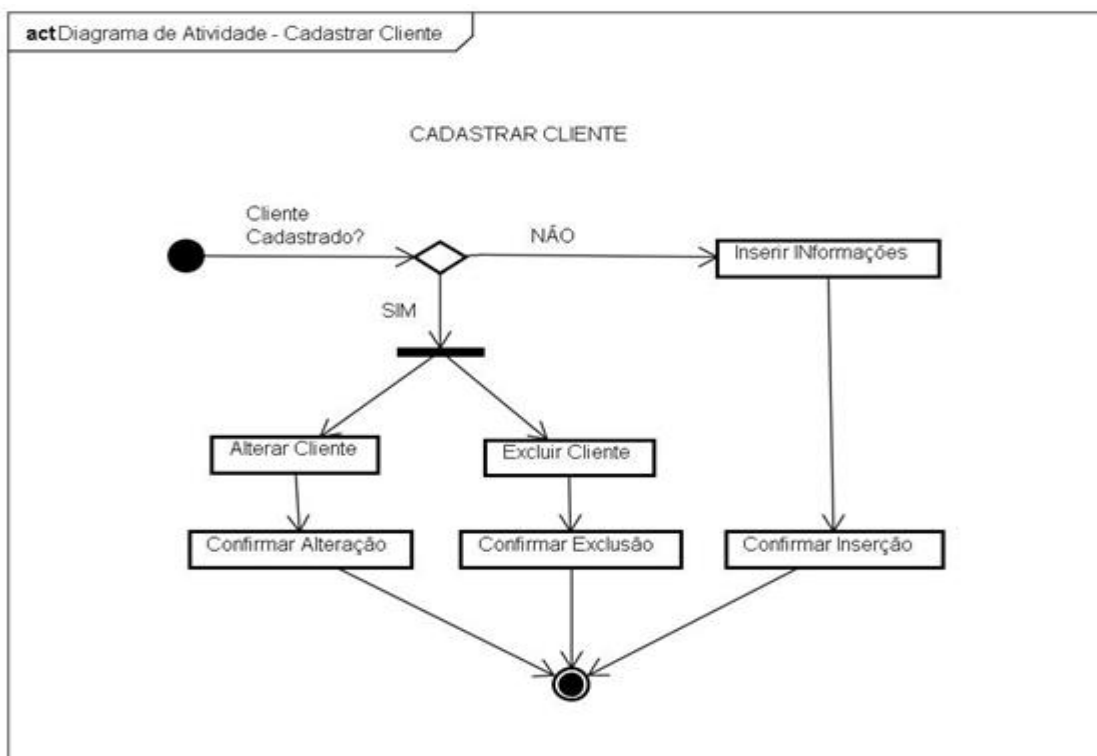


Figura 3 - Diagrama de Atividade - Cadastrar Cliente

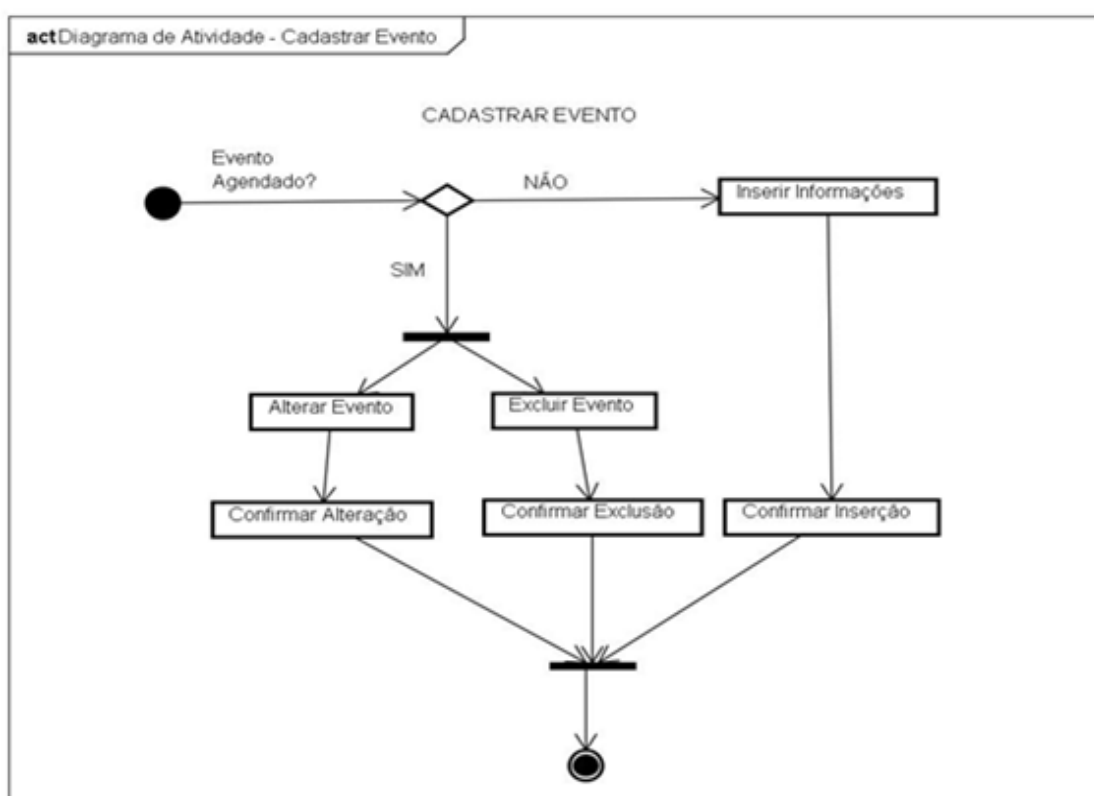


Figura 4 - Diagrama de Atividade - Cadastrar Evento

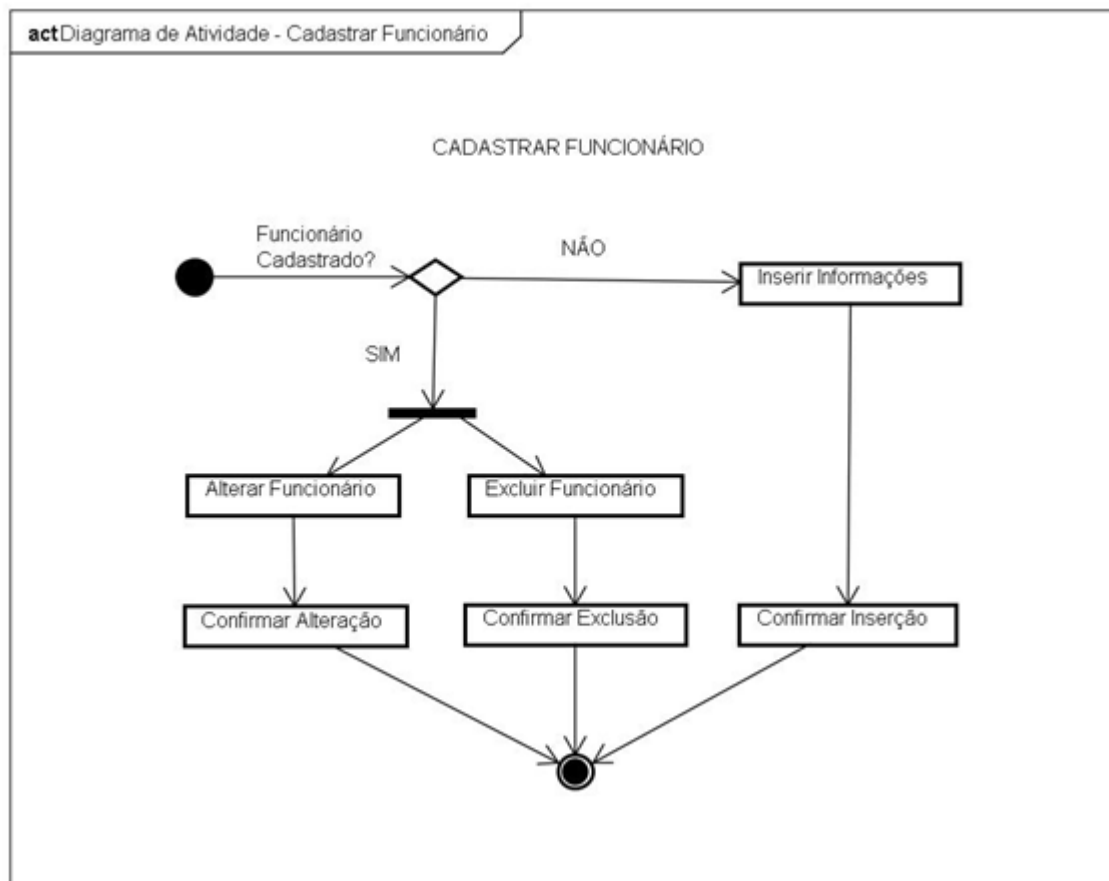


Figura 5 - Diagrama de Atividade - Cadastrar Funcionário

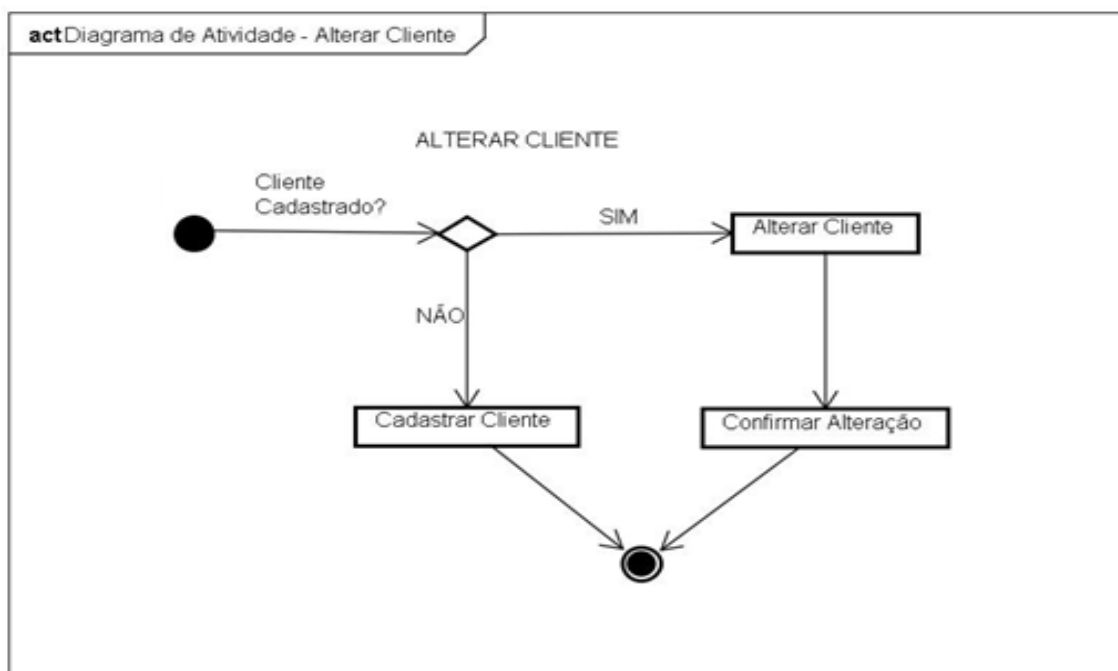


Figura 6 - Diagrama de Atividade - Alterar Cliente

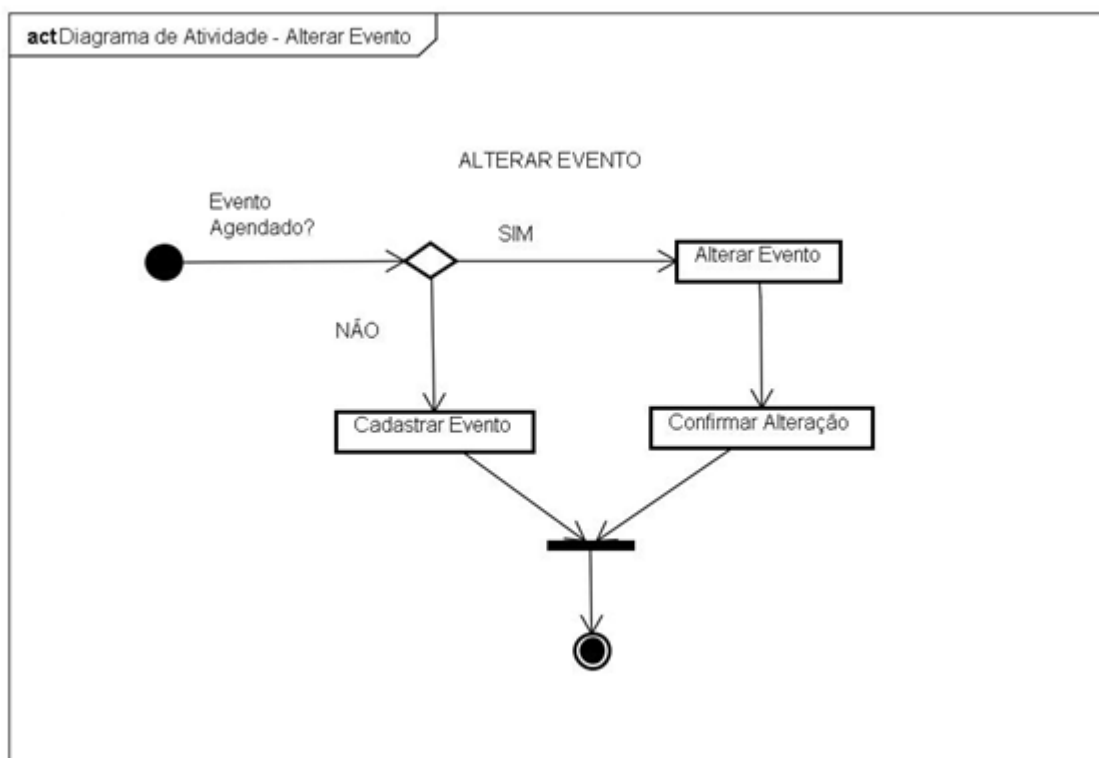


Figura 7 - Diagrama de Atividade - Alterar Evento

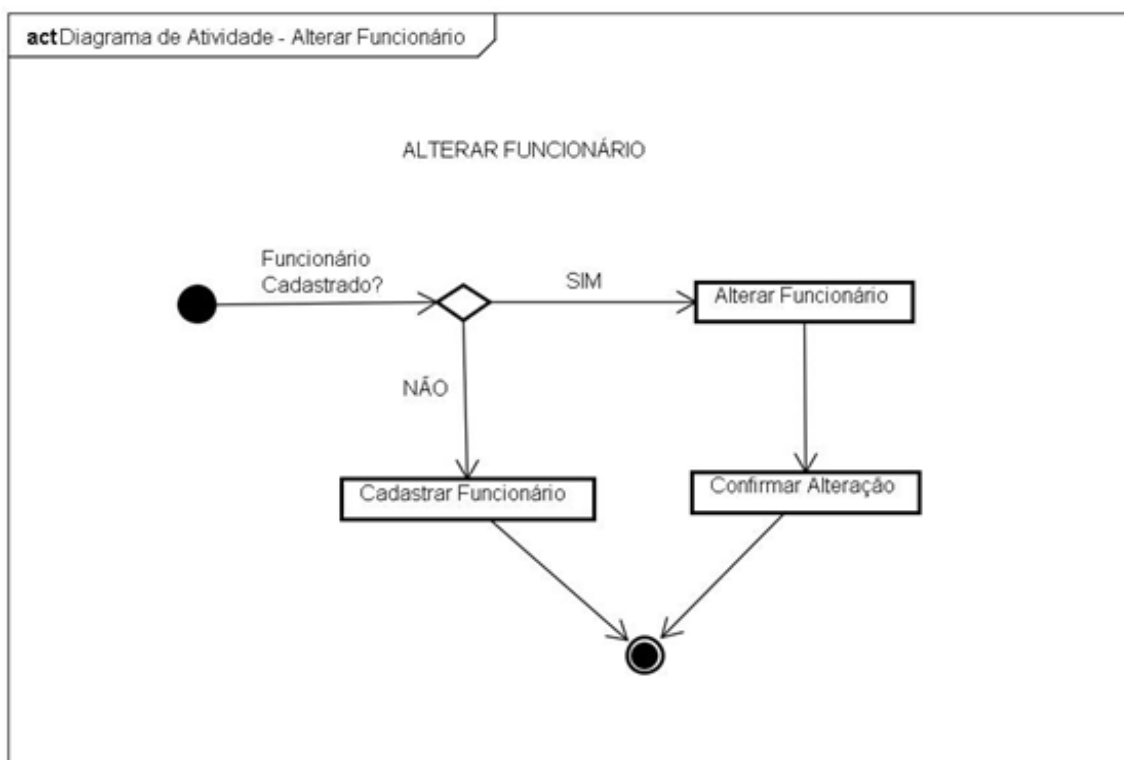


Figura 8 - Diagrama de Atividade - Alterar Funcionário

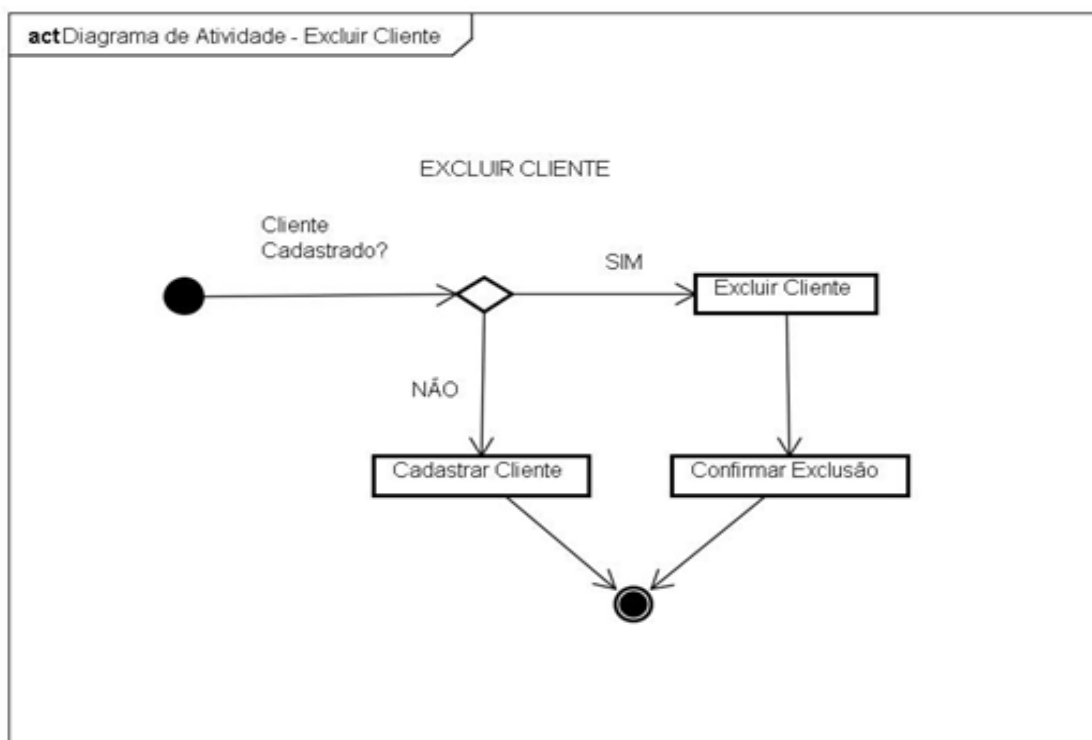


Figura 9 - Diagrama de Atividade - Excluir Cliente

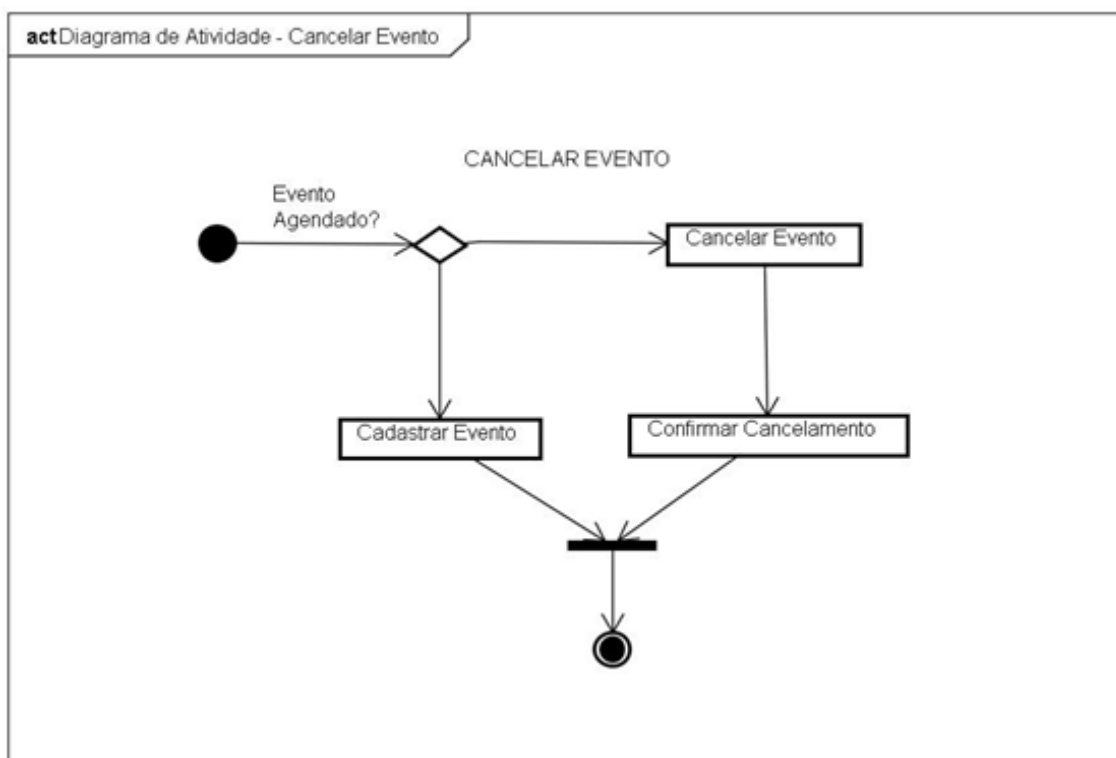


Figura 10 - Diagrama de Atividade - Cancelar Evento

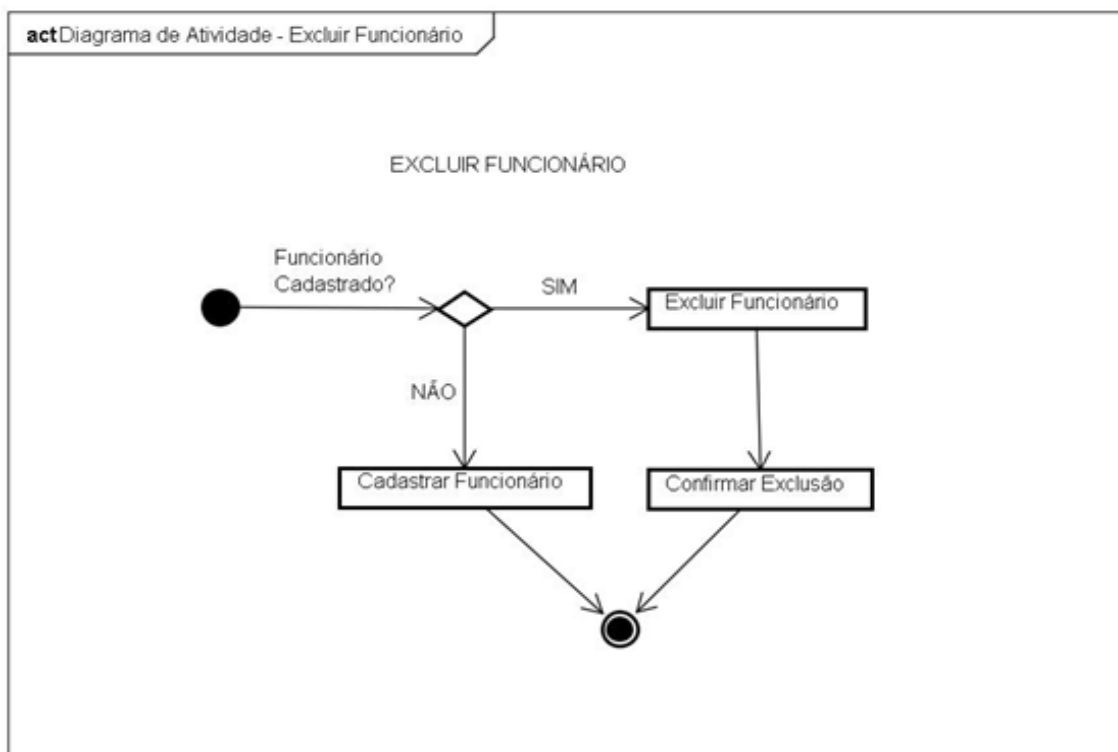


Figura 11 - Diagrama de Atividade - Excluir Funcionário

3.5. DIAGRAMA DE CLASSES

O Diagrama de Classes é uma maneira de apresentar visualmente a classe implementada no sistema, um modo geral de visualizar a ligação entre as classes e também visualizar seus objetos e métodos implementados.

A seguir a imagem abaixo irá apresentar o Diagrama de Classes deste projeto.

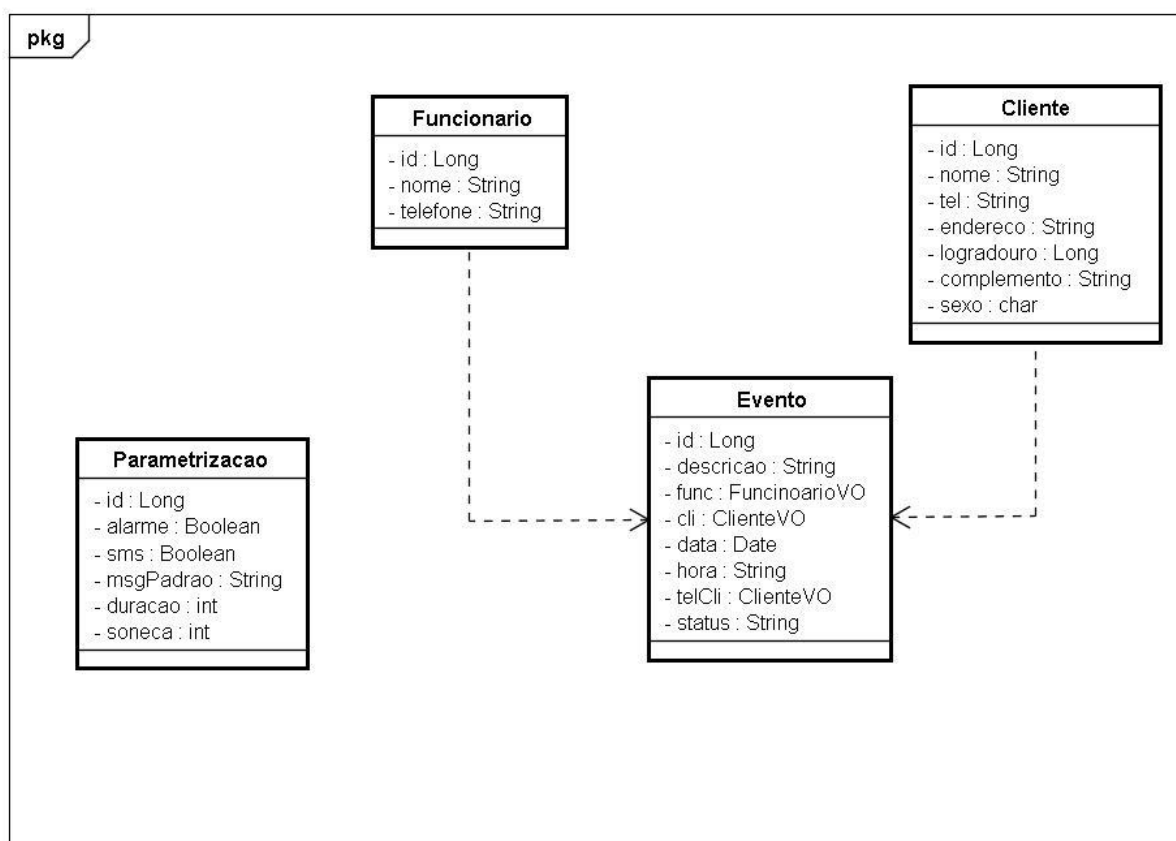


Figura 12 - Diagrama de Classes

A classe Parametrizacao acima exibida é responsável pela inserção das preferências no sistema, ou seja, o usuário irá definir de que modo deseja que o sistema se porte. Na classe Funcionario, o usuário irá definir os funcionários responsáveis pelo evento. A classe Cliente guarda as informações do cliente que fará parte do evento, como nome, endereço, telefone e demais informações. Por fim, a classe Evento, irá ser a classe onde o horário do evento e todas as demais

informações serão inseridas. Além das informações do evento, esta classe irá conter a utilização dos recursos do Alarme e do Envio de SMS.

3.6. MODELO ENTIDADE-RELACIONAMENTO

O Modelo Entidade-Relacionamento é um modo visual de como será implementado o banco de dados, quais tabelas irão existir e quais dados serão armazenados e de que modo poderão ser acessados. O Banco de Dados pode não ser a parte mais importante do aplicativo, mas é sim de suma importância para o sistema, tendo essa disposição física descrita no diagrama, torna-se mais simples o desenvolvimento do banco.

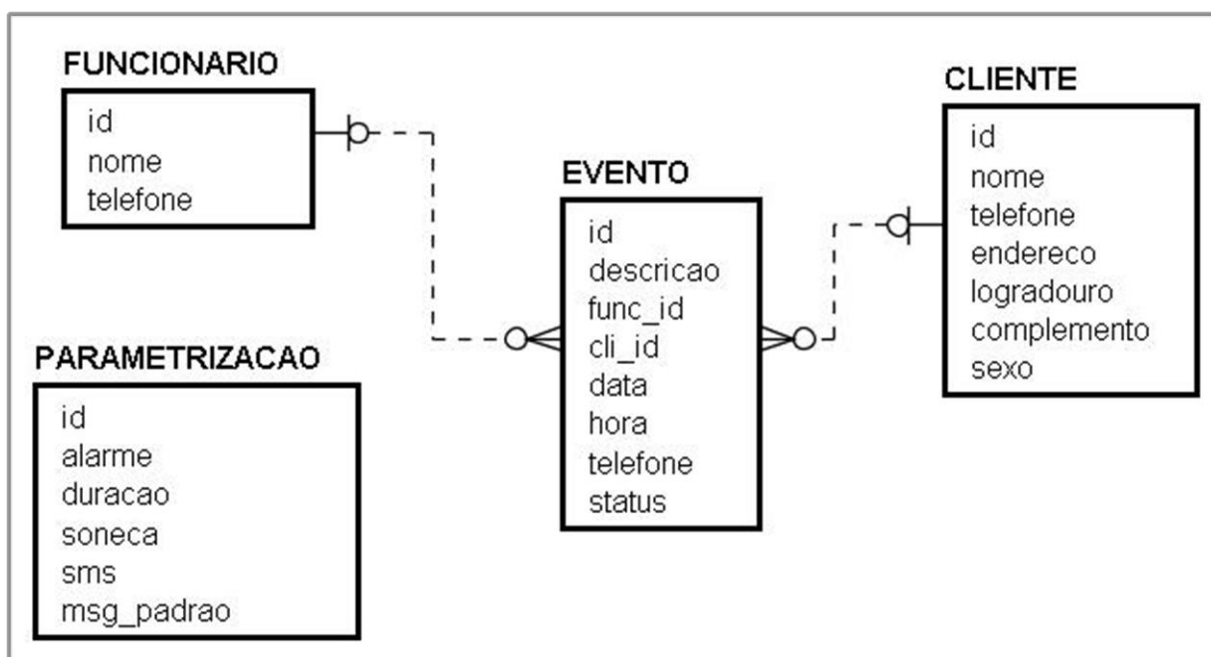


Figura 13 - Diagrama de Entidade-Relacionamento

4. ESTRUTURA DO PROJETO

4.1. ESTRUTURA ANALÍTICA DE TRABALHO [ou WORK BREAKDOWN STRUCTURE (WBS)]

Baseando-se em uma metodologia experimental, o desenvolvimento deste software será dividido em etapas para um melhor controle do planejamento. Para melhor estruturar essas etapas será utilizado o diagrama de Work Breakdown Structure (WBS) que faz uma representação da estrutura analítica do trabalho. A WBS é uma ferramenta que tem por objetivo principal decompor o projeto em partes menores, tornando-se mais fácil manejar e controlar o andamento do processo identificando os produtos, serviços e resultados a serem obtidos de um projeto. É uma ferramenta muito útil quando se tem um projeto de grande porte, dividindo-o em etapas e em projetos menores, evitando dessa maneira os possíveis erros e facilitando o encontro daqueles que existirem e a correção destes.

Esses processos menores que são gerados a partir do WBS podem também ser entendidos como processos. Um processo é um modo de determinar aonde inicia e aonde terminará o projeto, fazendo com que fique de maneira mais organizada e simplificada o entendimento.

Para melhor exemplificar as divisões de um projeto em processos, a figura logo abaixo apresentada, descreve as tarefas necessárias para a concretização do mesmo.

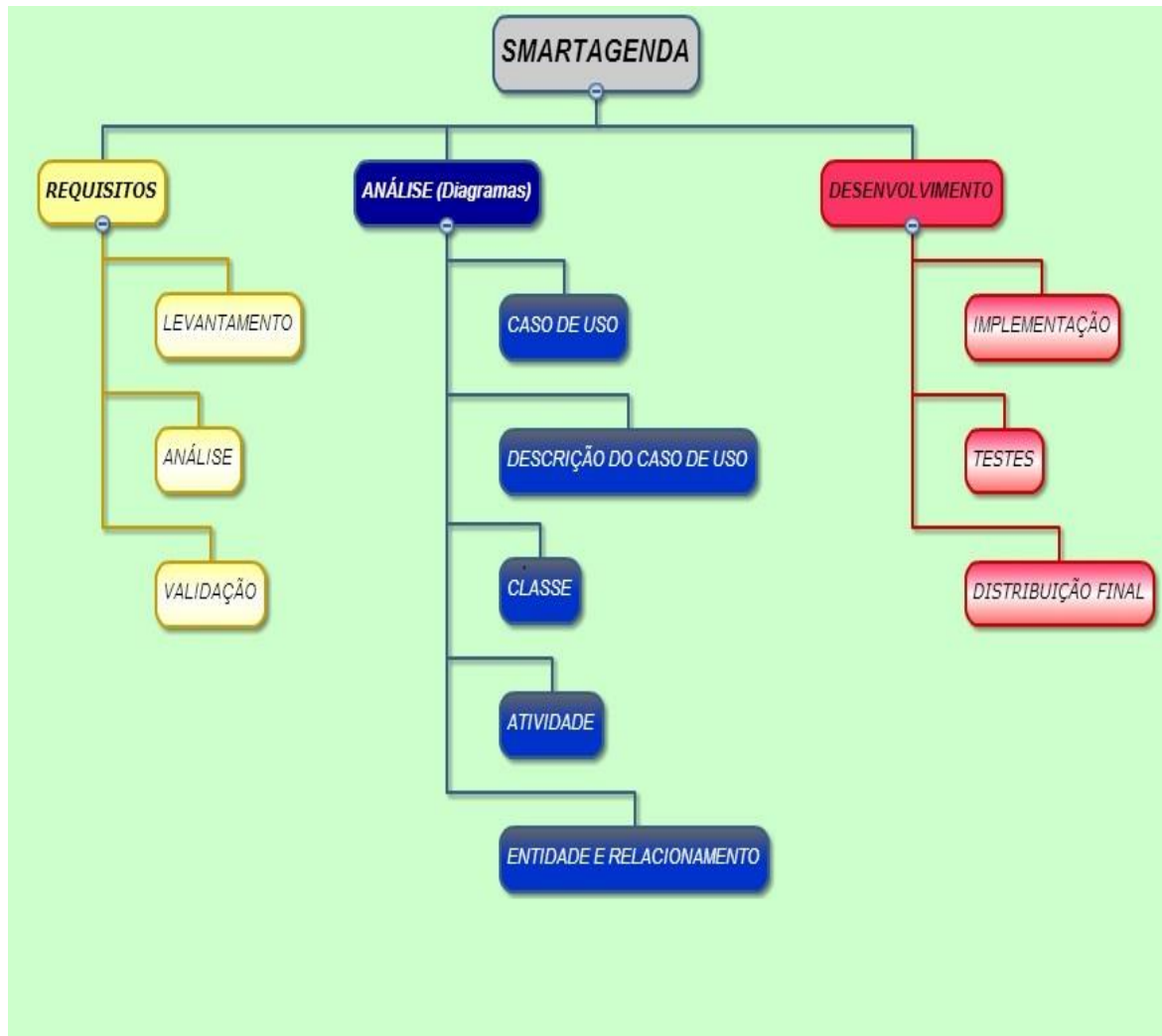


Figura 14 - Estrutura WBS do Projeto

4.2. ORÇAMENTO

Analista de Sistemas			
Analista	Quantidade Horas	Valor Hora	Total
Luis Gustavo de Oliveira Mazzo	150	R\$ 27,00	R\$ 4.050,00
Desenvolvedor			
Programador	Quantidade Horas	Valor Hora	Total
Luis Gustavo de Oliveira Mazzo	185	R\$ 30,00	R\$ 5.550,00
		Total	R\$ 9.600,00

Equipamentos				
Equipamento	Valor (inicial)	Vida Útil	Total	Valor (final)
Notebook	R\$ 2.850,00	R\$ 3,70	R\$ 1.239,50	R\$ 1.610,50
Dispositivo Móvel	R\$ 1.000,00	R\$ 0,65	R\$ 217,75	R\$ 782,25
		Total	R\$ 1.457,25	

Softwares			
Software	Quantidade Licenças	Valor Licença	Total
Eclipse Ganymede	1	R\$ -	R\$ -
SQLite Developer	1	R\$ -	R\$ -
Astah	1	R\$ -	R\$ -
Windows Seven	1	R\$ 750,00	R\$ 750,00
		Total	R\$ -

Valor Total do Projeto	
Mão de Obra	R\$ 9.600,00
Equipamentos	R\$ 1.457,25
Softwares	R\$ -
Total	R\$ 11.057,25

Tabela 14 - Orçamento

5. IMPLEMENTAÇÃO DO PROJETO

Assim como todo projeto de implementação de um aplicativo, em uma nova linguagem tal como Android não é diferente, existe um padrão a ser seguido já pré-definido como, por exemplo, a utilização de pacotes, como dispor os nomes das classes, dos objetos, das activities e demais objetos disponíveis para uso durante o desenvolvimento.

Este projeto faz uso de dez pacotes, *adapter*, *alarme_util*, *bo*, *dao*, *db*, *interfaces*, *sms_util*, *util*, *view*, *vo* onde cada um indica que tipo de classes estão relacionadas dentro de cada.

Para versão atualmente disponível do aplicativo não é feito uso algum de API's disponibilizadas pelo Google, porém em versões futuras a implementação de serviços disponibilizados pelo mesmo, tal como o *Google Maps* serão agregados ao sistema, para deixá-lo mais robusto e mais útil na localização dos eventos. Atualmente, a versão utilizada do Google Android pelo aplicativo *SmartAgenda* é a 2.3.3 – Gingerbread, que é a versão mais leve dentre as que oferecem todas as funcionalidades utilizadas pela aplicação.

5.1. ORGANIZAÇÃO ESTRUTURAL DO PROJETO GOOGLE ANDROID

O projeto da aplicação *SmartAgenda* foi dividido em pacotes para separar as partes transacionais, visuais e de regras de negócio, dividindo portanto em dez pacotes, onde cada um fica encarregado de agrupar um conjunto de classes relacionadas a diferentes tipos de classes, como por exemplo: *adapter* – pacote que agrupa as classes do controle visual das listas a serem exibidas; *alarme_util* – pacote encarregado de controlar a utilização de alarmes na aplicação; *bo* – pacote que uni as classes onde a finalidade das mesmas é efetuar o controle transacional, como por exemplo a inserção de dados e exclusão de dados do banco de dados; *dao* – é o pacote aonde a parte operacional da transação é realizada, é aonde os dados que serão manipulados pelas classes pertencentes ao pacote “*bo*” se encontram e são

enviados ao pacote citado; db – é o pacote responsável pela instanciação do banco de dados, definição das tabelas, inicialização do banco de dados e até mesmo a exclusão das tabelas; interfaces – é o pacote de definição das tabelas de banco de dados; sms_util – pacote encarregado de controlar a utilização de sms no aplicativo; util – é o pacote de classes de utilidades para o aplicativo, como por exemplo a conversão de dados de data e hora, de listas e demais possíveis utilidades necessárias; view – é o pacote que controla as telas previamente definidas e desenhadas, é o pacote responsável pela captura dos dados inseridos pelo usuário; vo – é o pacote aonde as classes, seus objetos, atributos e métodos são inseridos, ou seja, é um pacote comum de classes como nas demais linguagens.

Após estes pacotes serem criados pelo desenvolvedor, o Android também possui diversas pastas que são um padrão de qualquer aplicação a ser desenvolvida em Google Android, como a pasta “res” mais especificamente, pois é a mais utilizada assim como os pacotes, pois é nela aonde serão feitas as inserções de imagens que serão utilizadas no sistema, aonde será feita a definição das telas da aplicação, é aonde as telas do sistema são desenvolvidas e aonde o menu de seu aplicativo será descrito, ou seja, quando o usuário pressionar o “menu” de seu dispositivo móvel, o que será exibido para o mesmo e também aonde os apelidos de objetos, campos de texto e demais campos serão declarados, para facilitar o uso em todo o código, independente do pacote que irá utilizar.

5.2. INTERFACES DA APLICAÇÃO MÓVEL

O aplicativo *SmartAgenda* possui cerca de quinze páginas xml que tem seu conteúdo controlado diretamente por também quinze classes Java, onde dois pacotes estão diretamente envolvidos, o adapter e o view.

Cada tabela do banco de dados (parametrizacao, evento, cliente, funcionario) possui três páginas XML, uma para exibir o conteúdo armazenado no banco de dados, uma para controlar a forma com que essas informações armazenadas no banco de dados serão dispostas na tela e outra tela para capturar as informações inseridas pelo

usuário, com exceção da tabela parametrização, onde a mesma não terá uma tela que determine a disposição de seus dados ou uma tela que os exiba em forma de relatório, mas sim apenas uma tela para capturar as informações inseridas pelo usuário, onde a mesma quando for acessada novamente permitirá ao usuário alterar as informações que foram previamente inseridas.

Em sequência serão apresentadas algumas telas do sistema, para simplificar a visualização do sequenciamento dos processos, mostrando de forma mais ampla e completa a aplicação.

Logo abaixo será exibido o menu principal do sistema, onde o mesmo possibilita ao usuário acessar as telas que contem as listas de clientes, funcionários e eventos já cadastrados, sendo possível caso desejado, cadastrar, alterar ou até mesmo excluir um ou mais elementos já armazenados no banco de dados. Além destas opções, o usuário terá a possibilidade e em um primeiro instante a obrigação de parametrizar seu aplicativo, ou seja, inserir a forma que deseja que o aplicativo *SmartAgenda* irá se portar. As imagens serão dispostas em uma sequência de acesso no sistema:
Menu Principal - Botão cliente pressionado.

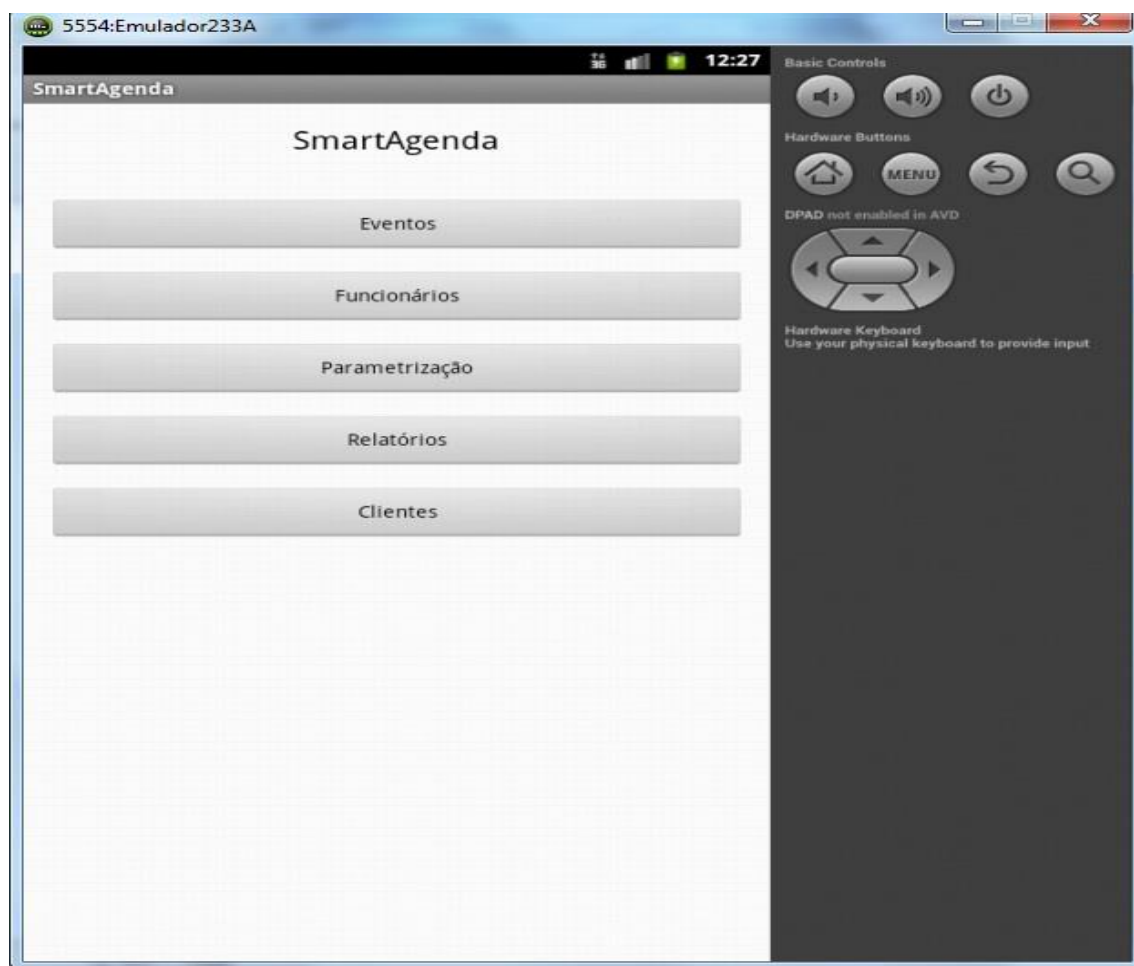


Figura 15 - Tela Inicial do *SmartAgenda*

A tela de Parametrização, onde o usuário irá optar por suas preferências.

A tela que exibirá a lista de clientes cadastrados no sistema, já com o botão “menu” do dispositivo móvel pressionado, disponibilizando as opções de adicionar, alterar e excluir, neste caso, os clientes.

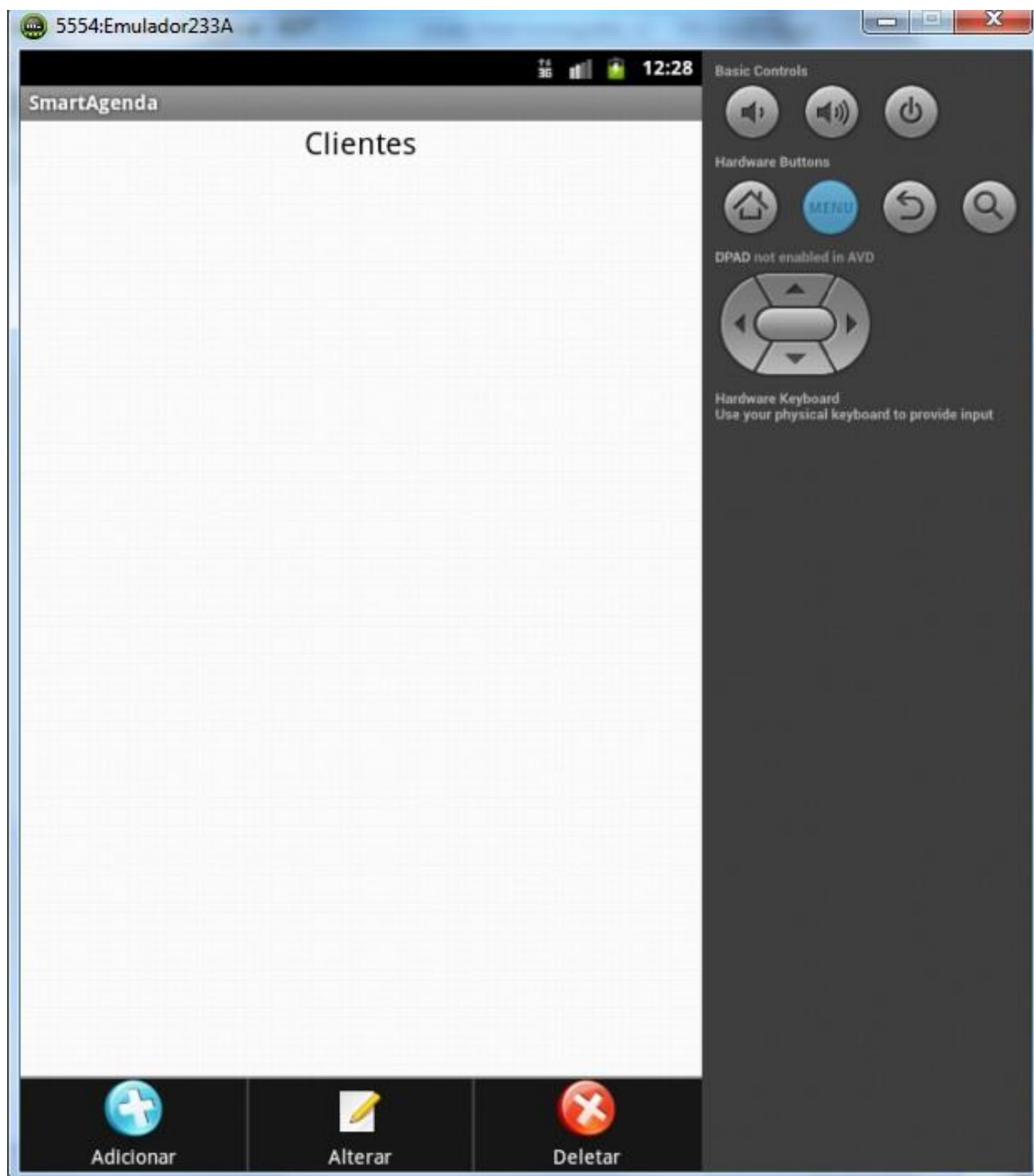


Figura 16 - Tela de Listagem dos Clientes

E por fim, a tela de cadastro de um novo cliente, com os campos preenchidos para mostrar o que cada um receberá, como por exemplo, o campo "Logradouro" receberá apenas números.

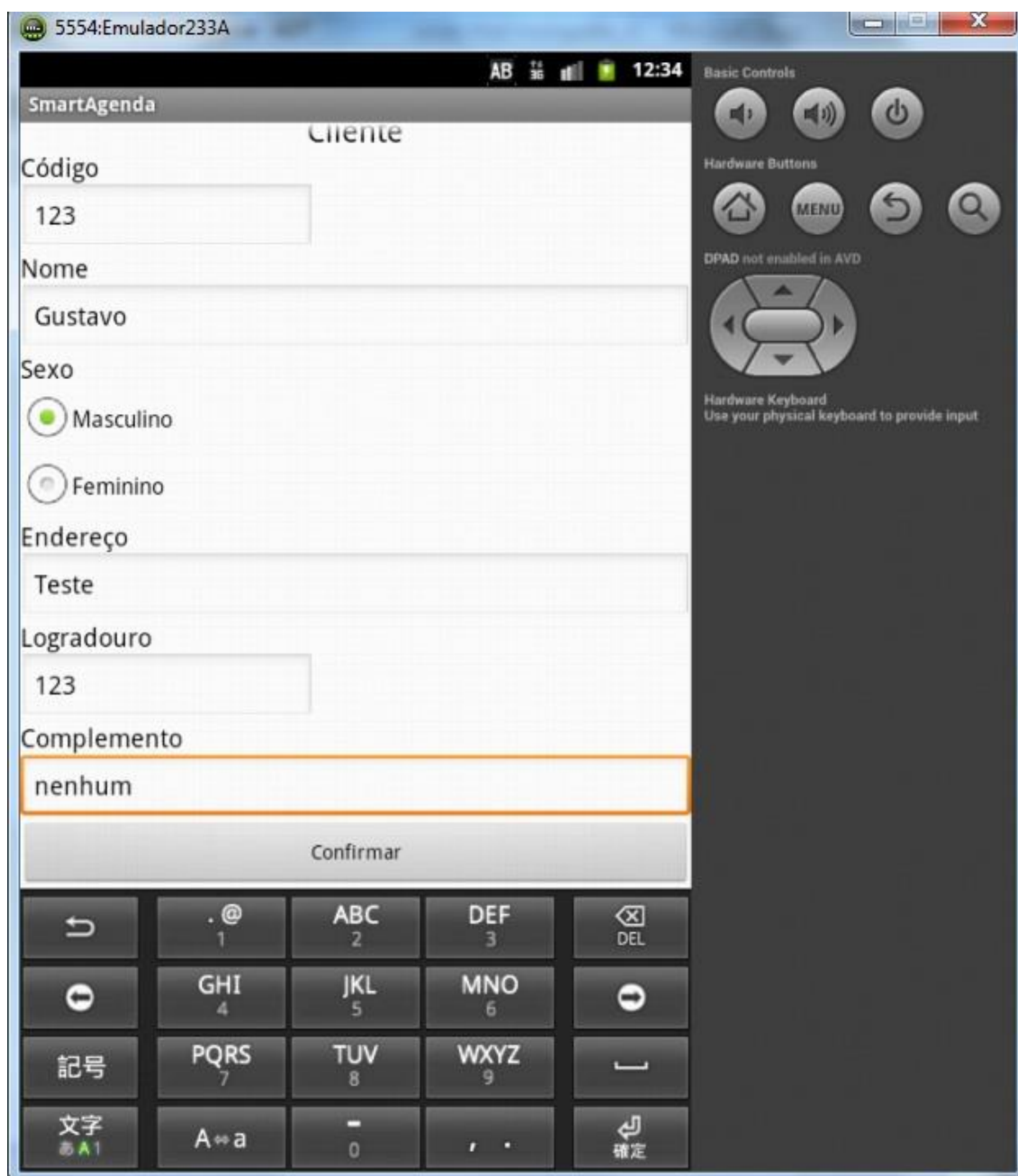


Figura 17 - Tela de Cadastro de Clientes

5.3. TRECHOS DE IMPLEMENTAÇÃO

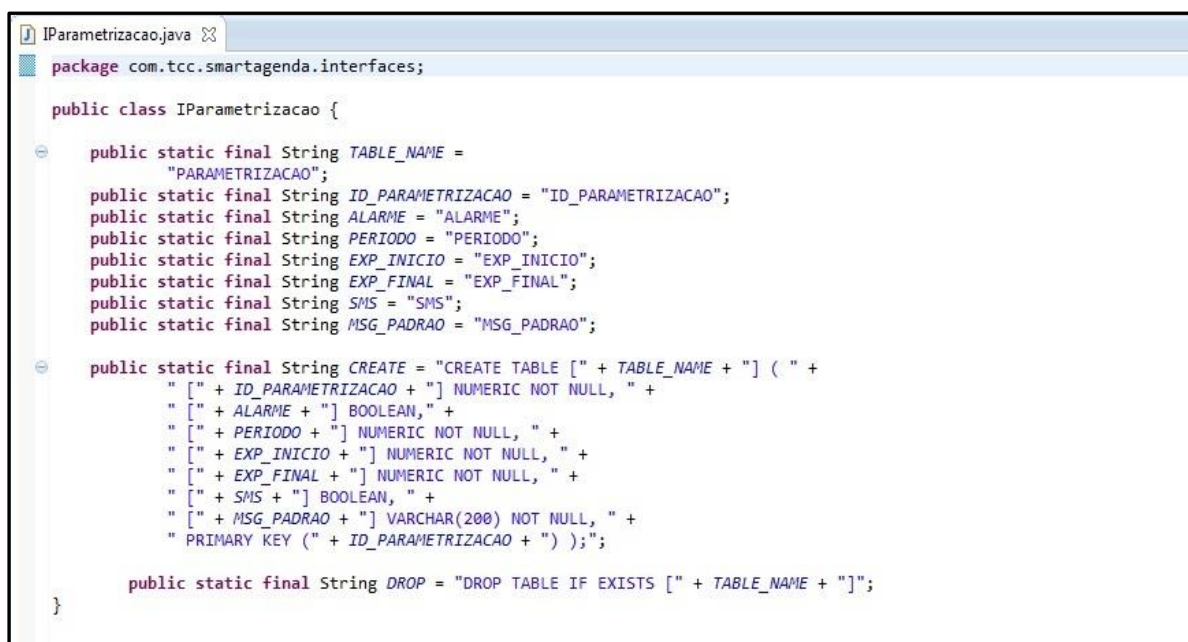
Em sequência serão exibidos alguns trechos do código do aplicativo *SmartAgenda*, onde os mesmos serão explicados, para proporcionar um entendimento da classe de definição do Banco de Dados, onde diversas

classes em mais de um pacote são utilizadas para realizar a criação do banco de dados e os demais procedimentos que envolvem o mesmo.

5.3.1. Implementação da Classe de Acesso e Definição do Banco de Dados

A definição do Banco de Dados é composta por um conjunto de classes, onde primeiro definimos o “CREATE” e o “DROP” de cada tabela, para ser utilizado em uma outra classe onde as tabelas serão iniciadas e/ou deletadas, além de uma outra classe para definirmos o nome do Banco de Dados.

As imagens abaixo exemplificam de forma mais clara e sucinta o que foi previamente dito. Nesta imagem a seguir, são realizadas as definições das constantes estáticas de criação da tabela, onde definimos os atributos, o nome da tabela, o “CREATE TABLE”, expressão que define a criação de tabelas e o “DROP TABLE”, expressão que exclui uma tabela de um banco de dados.



```
IParametrizacao.java
package com.tcc.smartagenda.interfaces;

public class IParametrizacao {

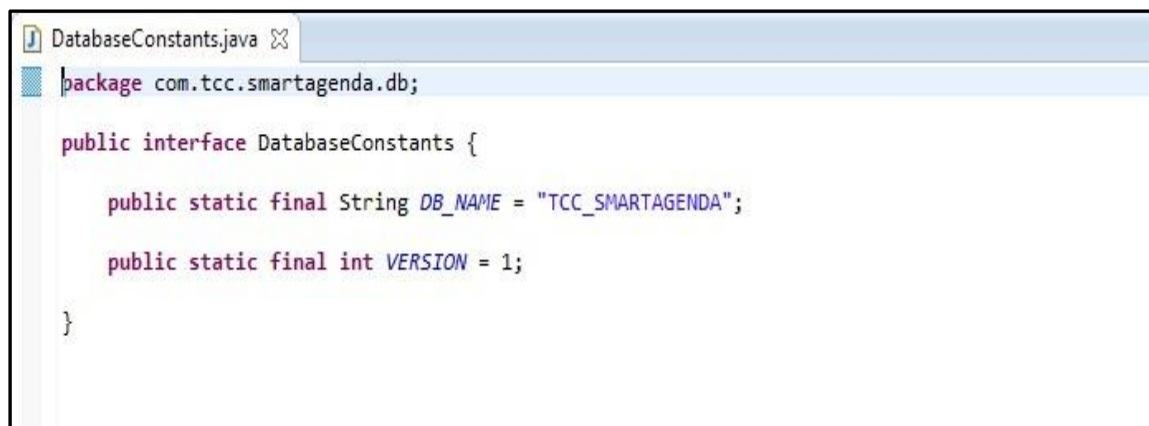
    public static final String TABLE_NAME =
        "PARAMETRIZACAO";
    public static final String ID_PARAMETRIZACAO = "ID_PARAMETRIZACAO";
    public static final String ALARME = "ALARME";
    public static final String PERIODO = "PERIODO";
    public static final String EXP_INICIO = "EXP_INICIO";
    public static final String EXP_FINAL = "EXP_FINAL";
    public static final String SMS = "SMS";
    public static final String MSG_PADRAO = "MSG_PADRAO";

    public static final String CREATE = "CREATE TABLE [" + TABLE_NAME + "] ( " +
        "[" + ID_PARAMETRIZACAO + "] NUMERIC NOT NULL, " +
        "[" + ALARME + "] BOOLEAN, " +
        "[" + PERIODO + "] NUMERIC NOT NULL, " +
        "[" + EXP_INICIO + "] NUMERIC NOT NULL, " +
        "[" + EXP_FINAL + "] NUMERIC NOT NULL, " +
        "[" + SMS + "] BOOLEAN, " +
        "[" + MSG_PADRAO + "] VARCHAR(200) NOT NULL, " +
        "PRIMARY KEY (" + ID_PARAMETRIZACAO + ") );";

    public static final String DROP = "DROP TABLE IF EXISTS [" + TABLE_NAME + "];";
}
```

Figura 18 - Classe de Definição da Tabela no Banco de Dados

Já nesta próxima imagem, definimos a versão para o banco de dados e o nome do mesmo, onde a versão pode ser alterada a cada atualização nas tabelas e demais conteúdo que envolve o banco de dados ou o sistema como um todo.



```
DatabaseConstants.java
package com.tcc.smartagenda.db;

public interface DatabaseConstants {

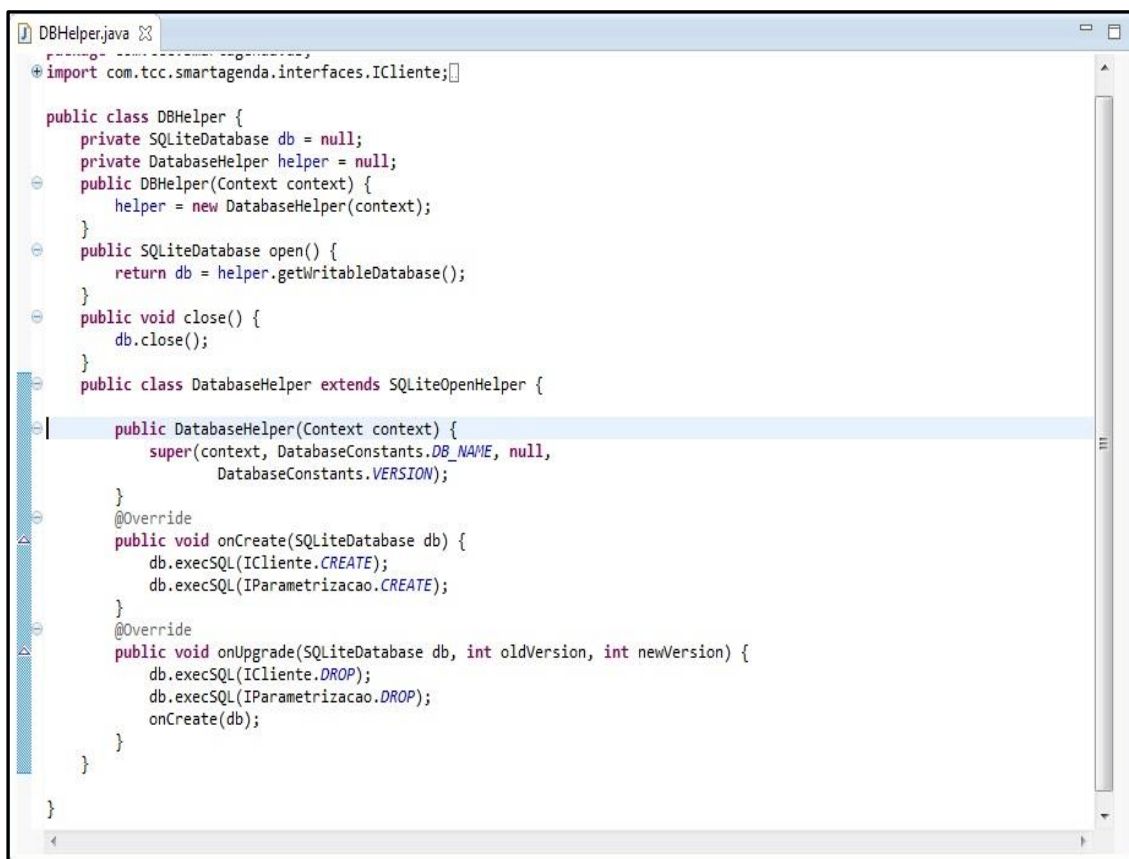
    public static final String DB_NAME = "TCC_SMARTAGENDA";

    public static final int VERSION = 1;

}
```

Figura 19 - Definição do Banco de Dados Utilizado

E nesta última imagem referente a utilização de um banco de dados em uma aplicação Android, é exibida a criação das tabelas e a exclusão das tabelas, utilizando as constantes estáticas criadas anteriormente no pacote interfaces. Além da utilização destas constantes, são criados métodos e subclasses, onde uma delas particularmente deve ser citada, a *“public class DatabaseHelper extends SQLiteOpenHelper”* pois esta se encarrega de criar a tabela se for necessário ser criada, abrir a tabela para uso da aplicação caso a mesma já tenha sido previamente criada ou até mesmo atualizá-la caso algo seja modificado e seja necessário uma atualização.



```
DBHelper.java
import com.tcc.smartagenda.interfaces.ICliente;

public class DBHelper {
    private SQLiteDatabase db = null;
    private DatabaseHelper helper = null;
    public DBHelper(Context context) {
        helper = new DatabaseHelper(context);
    }
    public SQLiteDatabase open() {
        return db = helper.getWritableDatabase();
    }
    public void close() {
        db.close();
    }
    public class DatabaseHelper extends SQLiteOpenHelper {
        public DatabaseHelper(Context context) {
            super(context, DatabaseConstants.DB_NAME, null,
                DatabaseConstants.VERSION);
        }
        @Override
        public void onCreate(SQLiteDatabase db) {
            db.execSQL(ICliente.CREATE);
            db.execSQL(IParametrizacao.CREATE);
        }
        @Override
        public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
            db.execSQL(ICliente.DROP);
            db.execSQL(IParametrizacao.DROP);
            onCreate(db);
        }
    }
}
```

Figura 20 - Classe de Inicialização e Exclusão das Tabelas

Em suma, diferentemente de outras linguagens de programação, o Google Android tem toda sua definição do banco de dados, inicialização do mesmo e atualização de suas tabelas e estruturas da aplicação, desenvolvido em classes nos pacotes dentro do código propriamente dito. Pode-se afirmar que desta maneira em que a estrutura de desenvolvimento do banco de dados em um aplicativo da linguagem Google Android é disposto, torna o processo de análise de erros, busca por falhas e melhorias a ser desenvolvida mais rápida e eficaz, tendo em vista que o código que está incorreto referente ao banco de dados é exibido no “LogCat”, a parte do ambiente de desenvolvimento que exhibe as possíveis inconsistências a serem encontradas no programa.

6. CONCLUSÃO

O *SmartAgenda* surge no intuito de auxiliar pessoas que necessitam de uma agenda de eventos e contatos mais interativa e que possa lhe ajudar não apenas com que o usuário não se esqueça de seus compromissos e tenha maior facilidade para controlar seus contatos, mas também de que os participantes daquele evento agendado possam ser alertados de seus compromissos, ou seja, mesmo sem estarem utilizando o aplicativo ou até mesmo um dispositivo móvel com um sistema operacional serão beneficiados com o aplicativo. Houve a necessidade da utilização de diversas bibliotecas previamente desenvolvidas do Google Android, para utilização de alarme do aparelho, para o envio de SMS e a utilização de notificações, onde proporcionou uma vasta gama de conhecimento nesta área de desenvolvimento móvel, preparando para o mercado de trabalho.

Em suma, muitos conhecimentos foram agregados com o desenvolvimento deste projeto, possibilitando a atuação em uma área de desenvolvimento que está em um crescente aumento e em constante avanço no mercado de trabalho.

6.1. TRABALHOS FUTUROS

Após a conclusão do projeto, pretende-se desenvolver uma atualização para que o aplicativo possa proporcionar ao usuário uma integração a algumas API's disponibilizadas pela Google, tal como a utilização do GOOGLE MAPS, agregando desta forma uma rota para chegar ao endereço do evento, além de atualizações para o avanço na captura de imagem de face dos contatos para que na listagem sejam exibidas suas fotos e não apenas imagens simbólicas.

Além destes pontos ressaltados, pretende-se submeter este projeto a uma possível inserção no Market do Google, disseminando desta forma o uso do aplicativo *SmartAgenda*.

REFERÊNCIAS

BARROS, Jussara. **Perdendo a Hora**. Disponível em: <<http://www.mundoeducacao.com.br/educacao/perdendo-hora.htm>>. Acesso em 15 de Março de 2013.

BORATTI, Isaias Camilo. **Programação Orientada a Objetos em JAVA**. 1ª Edição – Publicado em 2007.

BUZZAN, Tony. **Tony Buzzan Inventor of Mind Mapping**. Disponível em: <<http://www.tonybuzan.com>>. Acesso em 10 de Abril de 2013.

CHANGE VISION INC. **Change Vision Inc. astah Basic Operation Guide**. 2009. Edição Digital.

DEITEL, Paul; DEITEL, Harvey. **JAVA: Como Programar**. 8ª Edição – Publicado em 2010.

GAZZARRINI, Rafael. **Quantidade de smartphones no mundo vai ser três vezes maior até 2018**. Disponível em: <<http://www.tecmundo.com.br/celular/37416-quantidade-de-smartphones-no-mundo-vai-ser-tres-vezes-maior-ate-2018.htm>>. Acesso em 10 de Março de 2013.

FERREIRA, Rodrigo Lopes. **Introdução ao Google Android**. Produzido em 2011.

JAVA USER GROUP SWITZERLAND. **The Android GUI Framework**. Publicado em 2008.

JOBSTRAIBIZER, Flávia. **The Complete Guide to Google Android. Criação de aplicativos para celulares com Google Android**. Publicado em 2009.

LENCHETA, Ricardo R. **Google ANDROID: Aprenda a criar aplicações para dispositivos móveis com o Android SDK**. 2ª Edição – Publicado em 2010.

LOURENÇO, Michel. **Android Para Desenvolvedores**. 2ª Edição – Publicado em 2012.

MOTA, Kleber. **Introdução ao SQLite do Android: criando e usando um banco de dados baseado em sqlite3**. Publicado em 2011.

MURPHY, Mark L. **Beginning Android**. Publicado em 2009.

PRESSE, Da France. **Número de smartphones triplicará no mundo até 2018, diz pesquisa**. Disponível em: <http://g1.globo.com/tecnologia/noticia/2013/03/numero-de-smartphones-triplicara-no-mundo-ate-2018-diz-pesquisa.html?utm_source=twitterfeed&utm_medium=twitter>. Acesso em 21 de Março de 2013.

SALGADO, Diana. **História do Android**. Disponível em: <http://web.esad.ipleiria.pt/PDMIII/3080403/index_1.html>. Acesso em 13 de Abril de 2013.

VENTURA, Felipe. **No Brasil, só 14% tem smartphone**. Disponível em: <<http://www.gizmodo.com.br/no-brasil-so-14-tem-smartphone>>. Acesso em 05 de Junho de 2013.