

**UNIVERSIDADE FEDERAL DE SANTA CATARINA**

**CADESAÚDE: UM APLICATIVO PARA A PLATAFORMA ANDROID PARA  
LOCALIZAR ESTABELECIMENTOS DE SAÚDE PÚBLICOS E PRIVADOS NO  
ESTADO DE SANTA CATARINA**

**JUCEMAR DIMON**

Florianópolis/SC  
2016/2

**UNIVERSIDADE FEDERAL DE SANTA CATARINA  
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA  
CURSO DE SISTEMAS DE INFORMAÇÃO**

**CADESAÚDE: UM APLICATIVO PARA A PLATAFORMA ANDROID PARA  
LOCALIZAR ESTABELECIMENTOS DE SAÚDE PÚBLICOS E PRIVADOS NO  
ESTADO DE SANTA CATARINA**

**JUCEMAR DIMON**

Trabalho de conclusão de curso apresentado  
como parte dos requisitos para obtenção do  
grau de Bacharel em Sistemas de Informação  
da Universidade Federal de Santa Catarina.

Florianópolis/SC  
2016

JUCEMAR DIMON

CADESAUDE: UM APlicativo PARA A PLATAFORMA ANDROID PARA  
LOCALIZAR ESTABELECIMENTOS DE SAÚDE PÚBLICOS E PRIVADOS NO  
ESTADO DE SANTA CATARINA

Trabalho de conclusão apresentado como  
requisito para obtenção do grau de Bacharel,  
no curso de Sistemas de Informação da  
Universidade Federal de Santa Catarina, sob  
apreciação da seguinte Banca Examinadora:

---

Prof<sup>a</sup>. Dr<sup>a</sup>. Christiane Anneliese Gresse Von Wangenheim (UFSC)

---

Prof. Dr. Maurício Floriano Galimberti (UFSC)

---

Prof<sup>a</sup>. Dr<sup>a</sup>. Patricia Vilain (UFSC)

Aprovado em 09 de dezembro de 2016

---

Prof. Dr. Raul Sidnei Wazlawick (UFSC)  
Orientador

Dedico este trabalho a toda a população de Santa Catarina que precisa ou algum dia já necessitou de um atendimento médico ou hospitalar e, por muitas vezes, gastou muito tempo para localizar um simples telefone de contato ou endereço de uma clínica ou hospital que atendesse suas necessidades em um momento de enfermo.

## **AGRADECIMENTOS**

Agradeço a todos de minha família pela força e apoio na empreitada que é a admissão e conclusão de um curso de nível superior em uma universidade pública federal. Quero agradecer em especial a todos os amigos, colegas e conhecidos que se propuseram a instalar o aplicativo em seus dispositivos e ao respaldo dado, pois muitas questões foram repensadas e melhoradas a partir destes.

## RESUMO

Em Santa Catarina existe uma quantidade muito grande de estabelecimentos de saúde de diversos tipos, são clínicas especializadas, consultórios médicos, hospitais, postos de saúde, policlínica entre outras. Para conseguir informações sobre estes estabelecimentos, como um telefone por exemplo, é necessário recorrer a uma lista telefônica ou um catálogo de convênio médico que traz os estabelecimentos conveniados ou, então, a algum provedor de busca na internet que traz uma grande variedade de informações sobre estes estabelecimentos. Algumas delas, no pior dos casos, podem em algum momento tornarem-se desatualizadas e, devido a isto, podem não ser de utilidade para alguns usuários, principalmente em uma situação de urgência em saúde. Foi baseado neste contexto que se optou pela escolha do tema para este projeto de conclusão de curso. Ele envolveu a criação de uma solução para facilitar aos cidadãos e visitantes de Santa Catarina na busca por estabelecimentos de saúde dentro do nosso Estado. Utilizando-se dos dados dos estabelecimentos de saúde que estão na base federal do Cadastro Nacional dos Estabelecimentos de Saúde, o CNES, foi desenvolvido um aplicativo para *smartphones* que possuam a plataforma Android. A escolha pela base do CNES como fonte de dados foi baseada nas várias questões que garantem a veracidade dessas informações divulgadas pelo Ministério da Saúde, dentre elas, as diversas legislações que regem os procedimentos de cadastramento, podem garantir informações fidedignas sobre os estabelecimentos. O aplicativo desenvolvido possibilita a busca por estabelecimentos de saúde mais próximos de sua cidade ou nos arredores de sua posição geográfica atual. Este aplicativo também permite que o usuário faça a busca por um estabelecimento que possua um serviço, uma especialidade médica ou um profissional de seu interesse ou necessidade. Ele foi projetado para funcionar sem a necessidade de conexão com a internet e já se encontra disponível de forma gratuita na loja de aplicativos da Google.

**Palavras-chave:** CNES. Android. Saúde. Estabelecimentos de Saúde. Mobile.

## **ABSTRACT**

In Santa Catarina state there is a very large amount of health facilities of various kinds. Like Specialized Clinics, Doctors, Hospitals, Health Centers, among others. To get more information about these establishments, such as a phone for example, it is necessary to use a phone book or a catalog of medical "Health plans" or else a search provider on the Internet that provides information on these establishments. Some of them outdated may take additional time in search and harm the user especially in a medical urgency. The Brazilian Ministry of Health has a National registration system of health facilities (CNES) and disseminates free its database publicly on their specific portal. A number of legislative emplacements require that states and their municipalities carry out the registration of such establishments and keep this information up to date and reliable. Thus, we have a single database at the Federal level of all independent establishments if they are public or private. This course completion project involved the creation of solution to make it easier for citizens of Santa Catarina to search for health facilities within the state. Using "data" from health facilities that are in Federal CNES base, because they are public free information and there is legislation that standardizes the registration and upgrade processes, we developed an application for smartphones that have the Android platform that enables the search for the nearest health facilities in your town or geographic area. This application also allows the user to search for a property that has a service, a particular medical specialty or a professional interest or need. It is designed to work without need for connection to the internet and is now available free in the Google app store.

**Key words:** CNES. Android. Health. Health Facilities. Mobile.

## LISTA DE FIGURAS

Figura 1 - Portal de divulgação dos dados dos estabelecimentos.....	22
Figura 2 - Fluxo de dados dos estabelecimentos de saúde .....	23
Figura 3 - Fluxograma da contribuição no projeto .....	28
Figura 4 - Arquitetura da plataforma Android .....	32
Figura 5 - Página de detalhamento do estabelecimento .....	40
Figura 6 - Modelagem do banco de dados .....	41
Figura 7 - Diagrama de classes.....	42
Figura 8 - Classes que fazem a leitura de arquivos de texto.....	43
Figura 9 - Principais métodos da classe ConversorHtmlParaJava.....	44
Figura 10 - Árvore de links para navegação.....	45
Figura 11 - Site com problemas .....	46
Figura 12 - Diagrama de atividades .....	47
Figura 13 - Configuração do Estado a ser capturado.....	49
Figura 14 - Android Dashboards - Platform Versions .....	50
Figura 15 - Estrutura de diretórios: projeto Android X Emulador.....	53
Figura 16 - Classe gestora de conexões com o banco de dados.....	54
Figura 17 - A tela de apresentação do aplicativo .....	55
Figura 18 - Trecho de código fonte que inicia a tela de apresentação .....	56
Figura 19 - Tela de estabelecimentos favoritos .....	57
Figura 20 - Menu de configurações.....	58
Figura 21 - Código fonte da definição do estabelecimento como favorito .....	60
Figura 22 - Diretório no dispositivo onde é salvo o estabelecimento favorito .....	61
Figura 23 - Estrutura interna do arquivo de estabelecimentos favoritos.....	61
Figura 24 - método de execução em segundo plano .....	63
Figura 25 - Trecho de código que define a SearchView.....	64
Figura 26 - Layout do modelo de apresentação dos dados dos estabelecimentos ...	65
Figura 27 - Resultado do layout na interface gráfica .....	66
Figura 28 - Tela de configurações .....	68
Figura 29 - Declarações para geração do Layout da tela de configurações .....	70
Figura 30 - Tela de busca por GPS.....	71
Figura 31 - Filtros da tela de busca por GPS .....	73
Figura 32 - Aplicação de filtro sobre filtro .....	75

Figura 33 - Tela de busca por parâmetros .....	77
Figura 34 - Configuração da busca parametrizada .....	79
Figura 35 - Trecho de código que representa as consultas ao banco.....	81
Figura 36 - Tela de detalhes do estabelecimento.....	83
Figura 37 - Layout da visualização em abas da interface gráfica.....	84
Figura 38 - definição das Uses Feature no arquivo de manifesto .....	87
Figura 39 - Página do aplicativo na Google Play Store .....	88
Figura 40 - Feedback com relatório de erro .....	90
Figura 41 - Comparativo do consumo de memória RAM .....	93

## **LISTA DE QUADROS**

Quadro 1 - Plataforma de nomes de versões e níveis de API.....30

## SUMÁRIO

1 INTRODUÇÃO .....	12
2 OBJETIVOS .....	15
2.1 OBJETIVO PRINCIPAL.....	15
2.2 OBJETIVOS ESPECÍFICOS .....	15
2.3 JUSTIFICATIVAS.....	15
3 FUNDAMENTAÇÃO TEÓRICA.....	17
3.1 JSOUP .....	17
3.2 WEB CRAWLER .....	17
3.3 PADRÕES DE PROJETO.....	18
3.3.1 Data access object .....	18
3.3.2 Singleton .....	19
3.4 CADASTRO NACIONAL DOS ESTABELECIMENTOS DE SAÚDE .....	20
3.5 ANDROID.....	26
3.5.1 Histórico do Android .....	26
3.5.2 Arquitetura do Android.....	31
4 ESTADO DA ARTE .....	35
4.1 SISTEMA MÓVEL DE GEOLOCALIZAÇÃO E GEOPROCESSAMENTO PARA LOCAIS DE ATENDIMENTO EM SAÚDE.....	35
4.2 MEDSQUARE .....	37
5 DESENVOLVIMENTO DO PROJETO .....	39
5.1 A APLICAÇÃO DE CAPTURA DE DADOS.....	39
5.2 O APPLICATIVO PARA DISPOSITIVOS ANDROID .....	49
5.2.1 O banco de dados .....	51
5.2.2 A tela inicial .....	54
5.2.3 A tela de favoritos.....	56
5.2.4 A tela de configurações .....	67

5.2.5 A tela de busca por GPS .....	71
5.2.6 A tela de busca parametrizada.....	76
5.2.7 A tela de visualização do estabelecimento.....	83
5.2.8 A publicação na loja de aplicativos.....	86
5.2.9 Avaliação do aplicativo .....	88
5.2.10 Análise geral do uso da memória .....	91
6 CONCLUSÕES .....	95
6.1 TRABALHOS FUTUROS .....	96
6.1.1 Melhorias na ferramenta de captura de dados .....	96
6.1.2 Adaptação do aplicativo para o funcionar parcialmente online.....	96
6.1.3 Adaptação do aplicativo para o funcionar totalmente online .....	97
6.1.4 Reengenharia da interface gráfica .....	98
REFERÊNCIAS.....	99
APÊNDICES.....	104
APÊNDICE A - ESPECIFICAÇÃO DE REQUISITOS DE SOFTWARE .....	105
APÊNDICE B - CODIGO FONTE DO PROGRAMA DE CAPTURA DE DADOS ....	109
APÊNDICE C - CODIGO FONTE DO APLICATIVO ANDROID.....	204
APÊNDICE D - ARTIGO SOBRE O TCC .....	548
ANEXOS .....	564
ANEXO A - TIPOS DE ESTABELECIMENTOS/UNIDADES.....	565
ANEXO B - TIPOS DE SERVIÇOS ESPECIALIZADOS .....	569

## 1 INTRODUÇÃO

Com a difusão e popularização dos sites de anúncios para empresas na internet, muitos usuários quando necessitam encontrar alguma informação sobre algum estabelecimento comercial em algum ponto da cidade podem preferir buscar estas informações nestes repositórios ao invés de utilizar as antigas listas telefônicas de papel. Atualmente, quando se faz uma busca por uma clínica médica ou hospital em algum motor de busca na internet, o que se pode receber como resposta em alguns casos é uma infinidade de resultados. Dentre estes resultados poderemos achar: páginas do *MAPLink*<sup>1</sup>, páginas do Apontador<sup>2</sup>, do GuiaMais<sup>3</sup>, da Telelistas<sup>4</sup>, do *Foursquare*<sup>5</sup> e até mesmo do Google Mapas<sup>6</sup> e do Google Meu Negócio<sup>7</sup>, entre outras diversas, contendo telefones, endereços da empresa procurada ou similares a ela. Com o passar do tempo a empresa que anunciou seu negócio ou o usuário que cadastrou ou marcou um local em uma rede social, como o *Foursquare* por exemplo, podem perder seu interesse nessas informações. Isto pode ocasionar no esquecimento da atualização desses dados criados e assim essas informações podem tornar-se inúteis a quem necessita por estarem desatualizadas. De forma geral, no pior dos casos, pode-se perder tempo tentando encontrar a informação correta que se procura, como o telefone de contato, por exemplo. Situações como esta, por muitas vezes, podem forçar o internauta a gastar mais tempo do que o esperado, uma vez que tal recurso deveria servir para facilitar sua vida. Em uma situação emergencial quando se precisa buscar por um estabelecimento de saúde usando-se da metodologia descrita acima, a falta de informações precisas pode tornar a situação mais agravante.

As plataformas mobile tem crescido e se popularizado nos últimos anos. No ano de 2014 as vendas de dispositivos no mundo cresceram em torno de 23% (EXAME.COM, 2015). Atualmente os smartphones estão presentes no cotidiano de qualquer pessoa, pesquisas apontam que o seu uso ao longo do dia também tem

---

<sup>1</sup> <http://maplink.com.br/>

<sup>2</sup> <http://www.apontador.com.br/>

<sup>3</sup> <http://www.guiamais.com.br/florianopolis-sc>

<sup>4</sup> <http://www.telelistas.net/>

<sup>5</sup> <https://pt.foursquare.com/>

<sup>6</sup> <https://www.google.com.br/maps>

<sup>7</sup> <https://www.google.com.br/intl/pt-BR/business/>

aumentado cada vez mais (NAÍMA SALEH, 2015). Existe uma grande variedade de aplicações para auxiliar na resolução de problemas dos mais variados tipos e devido à mobilidade proporcionada por esses dispositivos a sua utilização não fica presa a um ambiente fixo, como acontece com os PCs. Baseando-se nesse contexto, desenvolvedores pesquisam sobre as dificuldades existentes na rotina da pessoa para projetar e promover soluções com o intuito de ganhar sua fatia no mercado de aplicativos para as plataformas IOS, Android e *Windows Phone*.

Muitas empresas de vários setores diferentes do mercado têm percebido a popularidade e a influência dos smartphones na vida dos seus clientes e vem apostando nessas tecnologias para ampliar seu canal de comunicação com seus consumidores (VOXAGE SERVIÇOS INTERATIVOS, 2016).

Empresas privadas prestadoras de serviços de saúde como clínicas especializadas, centros médicos integrados e hospitais tem apostado no mercado de aplicativos para as plataformas mobile como ferramenta útil para divulgação de seus serviços e de comunicação com seus clientes. Isto pode ser percebido em Clinica Privada de Especialidades Villa María (2015), em Social Advisors (2015) e em Xavier (2015).

Empresas de convênios médicos, odontológicos, e outros, investem na produção de aplicativos para smartphones para auxiliar seus clientes na localização de suas clínicas, filiais e profissionais conveniados (BUENO, 2016).

Empresas de software pesquisam as necessidades do mercado nas áreas da saúde e promovem soluções que são comercializadas, ou distribuídas livremente aos usuários. Um exemplo disto é o aplicativo *MedSquare* desenvolvido por WINTIMEIPC (2015) para a plataforma Android que é distribuído livremente na loja de aplicativos da Google.

Este projeto tem como foco o desenvolvimento de uma aplicação para dispositivos móveis compatível com a plataforma Android para auxiliar os cidadãos do estado de Santa Catarina na busca por uma clínica, hospital, laboratório ou consultório médico. Como base de informações para esse aplicativo serão utilizados dados do CNES (Cadastro Nacional dos Estabelecimentos de Saúde). Estes, por serem dados públicos disponibilizados oficialmente pelo Ministério da Saúde são informações fidedignas a respeito dos estabelecimentos. Existe uma grande quantidade de publicações de portarias e notas técnicas pelos órgãos controladores oficiais designando sobre a obrigatoriedade da manutenção destas informações, muitas delas

podem ser vistas em na seção de consulta à legislação em Cnes (2016). Sendo assim um dos grandes pontos positivos deste projeto é a confiabilidade dos dados disponibilizados no aplicativo.

## **2 OBJETIVOS**

Neste capítulo serão apresentados todos os objetivos: principal e específicos do projeto desenvolvido e sua justificativa para comprovação de sua importância de seu desenvolvimento.

### **2.1 OBJETIVO PRINCIPAL**

Desenvolver um aplicativo utilizando como base os dados públicos disponibilizados pelo Ministério da Saúde para fornecer uma forma prática e precisa de localizar um estabelecimento de saúde no Estado de Catarina.

### **2.2 OBJETIVOS ESPECÍFICOS**

Outros objetivos foram eleitos para este projeto, dentre eles destacam-se:

1. Produzir um documento de especificação de requisitos do aplicativo de busca por estabelecimentos;
2. Produzir uma ferramenta de captura de dados dos estabelecimentos de saúde que tenha como produto final de sua execução um arquivo de banco de dados para o SGBD *SQLite*, o qual será utilizado com base de consulta para o aplicativo;
3. Produzir um aplicativo para smartphones para consulta de estabelecimentos de saúde;
4. Publicar o aplicativo para consulta de estabelecimentos de saúde na loja de aplicativos da Google.

### **2.3 JUSTIFICATIVAS**

Ao realizar uma busca na loja de aplicativos do Google por opções que possibilitem aos usuários a localização de locais que prestam atendimento em saúde recebe-se muitas opções como resposta. Elas vão desde aplicações específicas para clientes de clinicas e convênios, como pode ser visto em Portal Unimed Ltda (2015), a simples aplicações para localizar locais que possuam atendimentos emergenciais,

como ocorre em DEVDECISION CONSULTORIA EM TECNOLOGIA (2015). Cada um desses aplicativos limita a quantidade de estabelecimentos mostrados. Podem limitar devido a alguns não serem conveniados com a empresa a qual pertence o aplicativo. Também limitam devido à necessidade de os próprios usuários serem os responsáveis pela alimentação da base de dados dos aplicativos. Pode ocorrer casos em que o aplicativo não é muito popular, sendo assim os usuários não contribuírem e a base de dados torna-se pequena.

O portal do DATASUS disponibiliza uma página onde é possível qualquer cidadão fazer uma busca por estabelecimentos cadastrados. Esses dados são públicos, porém pouco divulgados para a sociedade e suas páginas de busca possuem uma interface pouco muito amigável, uma vez que não seguem os padrões definidos pelo *World Wide Web Consortium*<sup>8</sup>(também conhecido como W3C). Pensando em uma forma de auxiliar esses usuários que precisam localizar um local que preste atendimento de saúde de forma rápida e precisa somado a praticidade e mobilidade fornecida pelos aplicativos mobile, definiu-se a criação de um aplicativo utilizando os dados do CNES como base de dados para consulta.

Espera-se que o projeto possibilite a produção de uma ferramenta que seja de utilidade pública para a população e que ela faça parte da rotina do cidadão do nosso estado.

---

<sup>8</sup> <http://www.w3.org>

### **3 FUNDAMENTAÇÃO TEÓRICA**

São apresentados neste capítulo alguns conceitos, técnicas e ferramentas que auxiliaram o autor na definição do tema e no desenvolvimento da solução projetada no objetivo deste trabalho.

#### **3.1 JSOUP**

Como afirma Jonathan Hedley (2015), *JSoup* é uma biblioteca Java que permite ao desenvolvedor manipular conteúdo HTML. Ela possui uma API (*Application Programming Interface*) que permite ao usuário extrair e manipular dados das páginas HTML utilizando DOM, CSS, e métodos do *jQuery-like*. Com ele é possível:

- Ler e converter HTML de uma URL, cadeira de URLs, arquivos ou *Strings*;
- Buscas dados das páginas HTML utilizando DOM ou seletores CSS;
- Manipular elementos HTML, atributos e textos;
- Organizar e formatar elementos HTML.

*JSoup* é um projeto *open source* distribuído livremente sob a MIT *license*<sup>9</sup>. O seu código fonte está disponível no *GitHub*<sup>10</sup>.

Para o projeto que foi desenvolvido, esta biblioteca Java foi utilizada na aplicação de captura de dados com o objetivo de conectar-se ao servidor do DATASUS para acessar as páginas do site do CNES para capturar suas *Tags* HTML que contém os dados dos estabelecimentos de saúde de interesse do projeto.

#### **3.2 WEB CRAWLER**

Um *web crawler* segundo Tavares e Lóscio (2013) é um agente de software que acessa a WEB de forma automatizada através de URLs para extrair dados. A esse processo dá-se o nome de *crawling*. O trabalho de Tavares, Lóscio e Oliveira (2012) complementa o conceito definindo que este componente automatizado visita as páginas HTML e navega pelo seu conteúdo capturando-o. Os links contidos nas

---

<sup>9</sup> <http://jsoup.org/license>

<sup>10</sup> <https://github.com/jhy/jsoup>

páginas são adicionados à sua fila de URLs que serão visitadas na sequência. Assim, um *web crawler* pode varrer toda a web de forma recursiva.

Foi baseado no conceito citado acima que se pensou em desenvolver uma aplicação específica que pudesse fazer a captura dos dados contidos nas páginas HTML do site do CNES. Neste projeto foi desenvolvido uma ferramenta de captura de dados nomeada de *JWebCrawlerCNES*. Esta ferramenta além de comportar-se como um *web crawler*, armazenando e visitando as URLs das páginas do site do CNES, realiza a seleção e captura dos dados das páginas cujas URLs visitadas utilizando a biblioteca JAVA *JSoup* e armazena esses dados em um banco de dados *SQLite* que também é criado pela aplicação utilizando a biblioteca Java *SQLite JDBC*.

### 3.3 PADRÕES DE PROJETO

Padrões de projeto podem ser vistos como uma solução que já foi testada para um problema. Desta forma, um padrão de projeto geralmente descreve uma solução ou uma instância da solução que foi utilizada para resolver um problema específico. Nas duas seções seguintes serão apresentados dois padrões que foram utilizados neste projeto.

#### 3.3.1 Data access object

O padrão de desenvolvimento conhecido como DAO, do inglês *Data Access Object*, segundo Taborda (2015), é um padrão introduzido no ambiente JEE para simplificar e desacoplar a interação das aplicações Java com a API JDBC. Praticamente todas as aplicações corporativas possuem algum mecanismo de persistência e o mais amplamente utilizado são os bancos de dados e a linguagem SQL é a mais popular para promover a comunicação com os gerenciadores de bancos de dados, a API JDBC (*Java Database Connectivity*) foi desenvolvida justamente para resolver essa questão. Antes da utilização de JAVA os sistemas corporativos eram escritos em linguagens orientadas a processos que continha muitos códigos SQL misturados. Quando estes sistemas foram feitos com JAVA, o padrão DAO tinha como objetivo a separação dessa grande quantidade de código SQL nos sistemas legados. No padrão DAO, todas as comunicações com o mecanismo de persistência são mediadas por um objeto escrito especificamente para mapear as informações dos

objetos específicos de transporte para a API de persistência e vice e versa. Toda a lógica de mapeamento e execução das instruções é deixada dentro do objeto DAO, desta forma, isolando a aplicação da API de persistência por completo. O objeto DAO é criado para operar e interagir com o mecanismo de persistência em nome da aplicação. Ele, geralmente, executa os quatro tipos de operações básicas: Criar; Recuperar; Alterar e Apagar (CRUD - do inglês *Create, Retrive, Update , Delete*). As ações de edição são invocadas simplesmente passando um objeto contendo as informações que se deseja editar.

No projeto desenvolvido tal padrão foi utilizado tanto na aplicação de captura de dados quanto no aplicativo para Android. O padrão foi seguido durante a implementação das classes que manipulam os objetos das classes de modelos de dados como: Estabelecimentos; Profissionais; Serviços. Tudo isso para auxiliar na manipulação das operações dessas classes e suas respectivas representações como tabelas no banco de dados.

### **3.3.2 Singleton**

*Singleton* é um tipo de padrão de projetos. É uma convenção que, se programada da forma correta, garante que um único objeto de uma classe de interesse será instanciado e que há apenas um ponto global de acesso a ela, ou seja, em qualquer ponto da aplicação de uma forma geral caso haja a intenção de uso do objeto cuja classe foi implementada usando este padrão, chama-se o construtor do objeto e caso ele já tenha sido criado, tem-se uma referência dele. Este tipo de padrão de projeto é muito útil quando os objetos possuem atributos de configuração que não podem se repetir. Um outro cenário onde ele é útil seria quando os objetos consomem muitos recursos do sistema, nesses casos evita-se ter mais de uma instância do mesmo na memória (FREEMAN et al., 2009, p. 132).

No projeto desenvolvido esse padrão foi utilizado na aplicação de captura de dados. O padrão foi seguido durante a implementação das classes que manipulam a comunicação com o banco de dados *SQLite* como um mecanismo para garantir uma correta manipulação dos processos de abertura e fechamento de conexões com o banco.

### **3.4 CADASTRO NACIONAL DOS ESTABELECIMENTOS DE SAÚDE**

O Ministério da Saúde define que todos os estabelecimentos, públicos e privados, que prestam algum tipo de serviço em saúde ao ser humano devem ser cadastrados em sua base de dados federal (BRASIL. Portaria nº 511, Diário Oficial da União, 2000). Esse cadastro é efetuado mediante ao preenchimento pela empresa de um formulário através do qual são coletados todos os dados de sua estrutura física, organizacional e humana, e a sua análise e aprovação pelo órgão responsável pelo cadastro que, geralmente, é uma secretaria municipal de saúde. Após o cadastro aprovado, o estabelecimento é registrado e passa a ser identificado na base de dados através do código conhecido como CNES (Cadastro Nacional de Estabelecimentos de Saúde). Todas as empresas cuja a atividade principal ou secundária é o fornecimento de serviços relacionados com a saúde humana precisam ter cadastro no Ministério da Saúde e os dados que são usados para o cadastro precisam ser verídicos e atualizados.

O cadastro nacional de estabelecimentos de saúde foi instituído pela Portaria nº 376, Diário Oficial da União (2000). Foi publicada no Diário oficial da União e permaneceu em consulta pública durante alguns meses onde os gestores municipais e estaduais e a população interessada contribuíram com suas experiências e sugestões. Após, esta consulta foi encerrada e definiu-se a Portaria nº 511, Diário Oficial da União (2000) que normatizaria o processo de cadastros dentro do território brasileiro. Todo esse sistema representa um desejo muito antigo de todos que utilizam as informações sobre saúde no país para seu trabalho diário. Os dados que são coletados sobre os estabelecimentos auxiliam os gestores públicos a visualizar o que há disponível em variedades de estabelecimentos no país para planejarem suas ações operacionais e gerenciais na área da saúde (MINISTÉRIO DA SAÚDE, 2006, p. 5). Estes dados coletados dos estabelecimentos, segundo explicado na seção sobre CNES em DATASUS (2016) são:

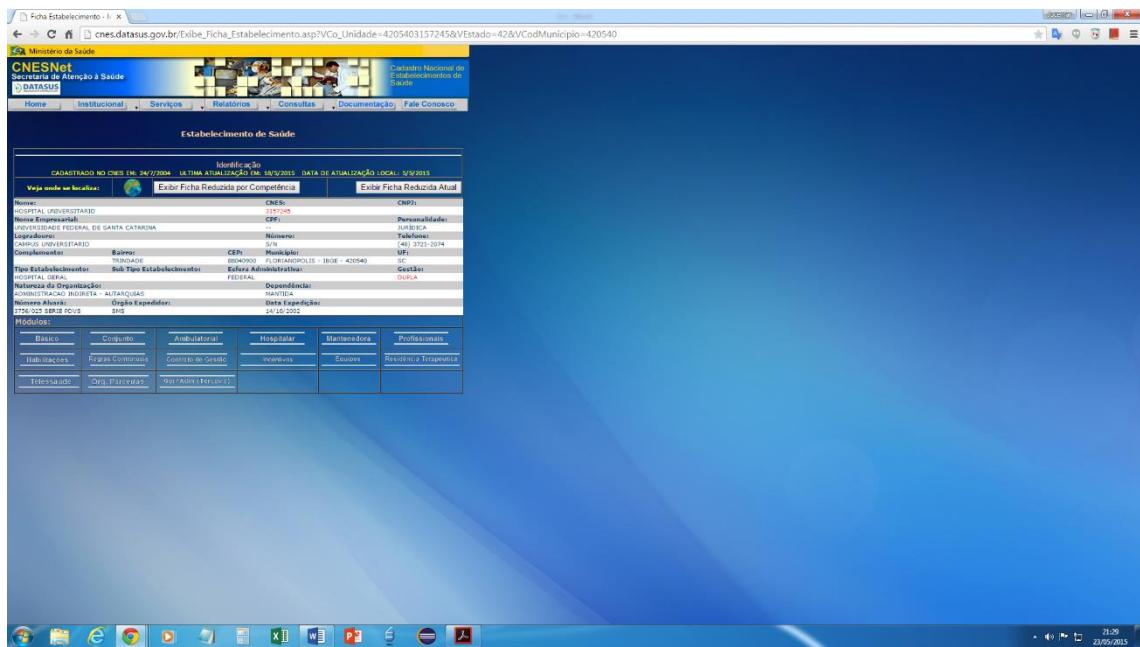
- Informações básicas gerais como nome fantasia, razão social;
- Endereçamento/Localização como: logradouro, bairro, CEP;
- Gestor responsável pela atualização do cadastro como: Secretaria Municipal de Saúde ou Secretaria Estadual de Saúde;
- Atendimentos prestados como: internação, atendimento ambulatorial, atendimento emergencial;

- Caracterização da empresa como: natureza jurídica, tipo de retenção de tributos;
- Equipamentos instalados como: RX, Tomógrafo, ultrassom;
- Serviços de Apoio como: SAME, S. Social, Lavanderia;
- Serviços Especializados como: Cardiologia, Nefrologia, Oftalmologia;
- Instalações Físicas como: leitos, salas, consultórios;
- Dados dos profissionais como: CBO, Carga horária, forma de vínculo;
- Equipes existentes como: ESF (Equipe de Estratégia de Saúde da Família);
- Cooperativas.

O empreendimento é gigantesco e os dados coletados são utilizados como base para operacionalizar todos os sistemas de informação em saúde do Ministério da Saúde que são a base para uma boa gestão pública, além, é claro, de adquirir conhecimento efetivo de como está formado o universo de estabelecimentos que cuidam da saúde da nossa população, desde nos grandes centros, até nas mais longínquas localidades, tornando visível esse cenário a toda sociedade, proporcionando ao gestor conhecer a rede assistencial existente e sua potencialidade, imprescindíveis nos processos de planejamento em saúde, regulação, avaliação, controle e auditoria, bem como dar maior visibilidade ao controle social para o melhor desempenho de suas funções DATASUS (2009b, p. 1).

A implementação efetiva do sistema ocorreu em 2003, desde então ações para sua manutenção e evolução vêm sendo tomadas para que o projeto mantenha uma roupagem mais moderna e amigável. Para tal, frequentemente, novas portarias vêm sendo emitidas pelo Ministério da Saúde para reforçar a importância das ações de cadastramento e atualização destas bases de dados, sempre lembrando das responsabilidades de cada parte e das devidas penalidades (BRASIL. Portaria nº 1.646, Diário Oficial da União, 2015). A criação de uma base de dados única e federal de todos os estabelecimentos de saúde do país é uma tarefa grande que deve envolver todas as esferas da administração pública Federal, Estadual e Municipal para que se possa manter uma base de dados com informações fidedignas para sistematicamente providenciar sua disseminação perante o território nacional. A Figura 1 exibe o portal principal utilizado para disseminar tais informações para a sociedade brasileira.

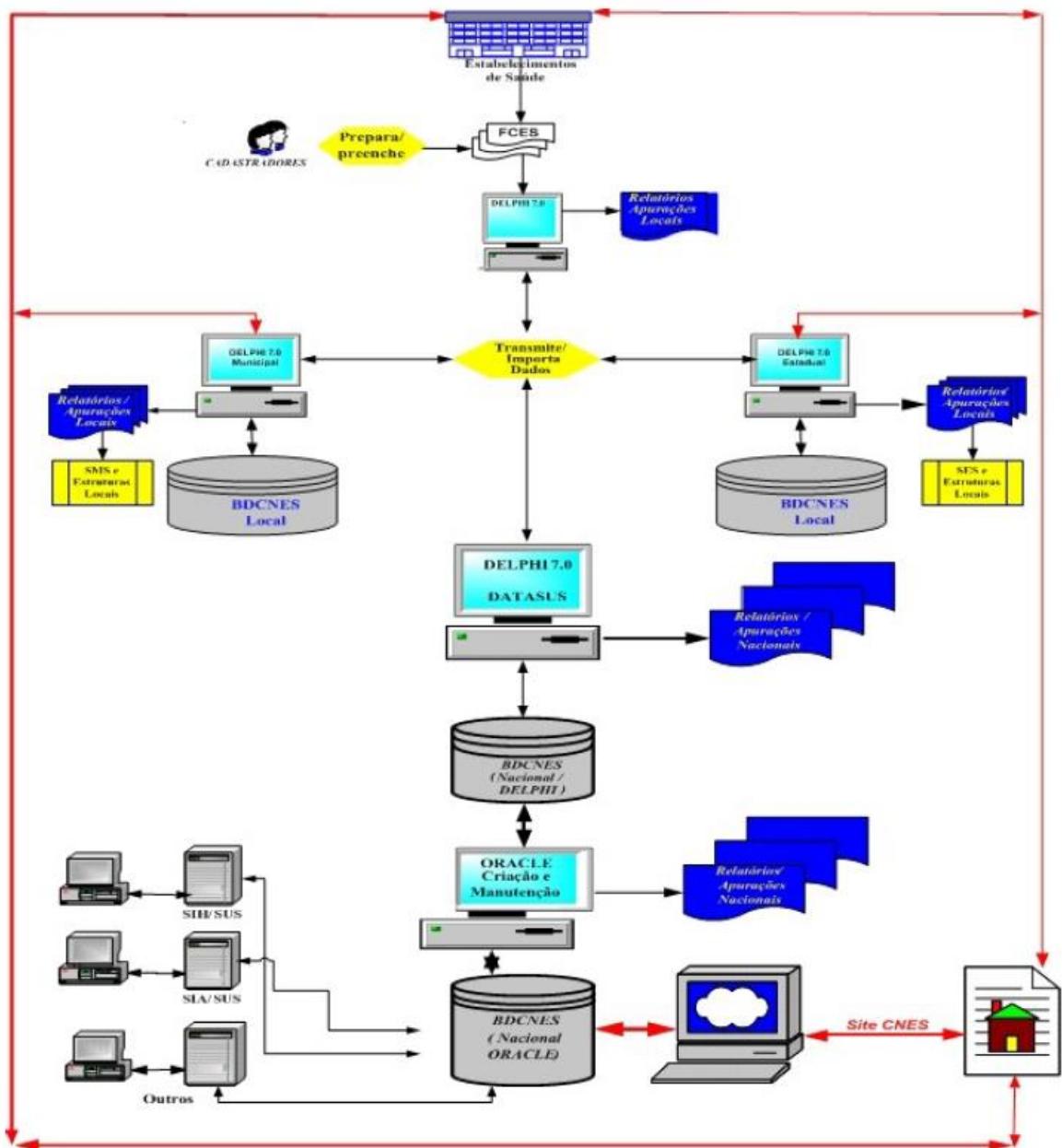
*Figura 1 - Portal de divulgação dos dados dos estabelecimentos*



Fonte: DATASUS (2015b)

Os responsáveis pela localização, cadastro, inspeção e atualização dos cadastros dos estabelecimentos de saúde dentro desse sistema são as secretarias de saúde municipais ou estaduais. Elas podem criar e gerenciar seu próprio fluxo de trabalho, porém estas devem sempre preconizar as normas e prazos estipulados pelas portarias, notas técnicas e normativas e as leis emitidas pelas esferas superiores. A Figura 2 abaixo apresenta o fluxo oficial de todos os processos que envolvem o sistema (DATASUS, 2009a, p. 3):

Figura 2 - Fluxo de dados dos estabelecimentos de saúde



Fonte: DATASUS (2009a, p. 9)

As secretarias municipais de saúde são as responsáveis pelo cadastramento dos estabelecimentos de saúde lotados em seus municípios. Elas devem buscar por estes estabelecimentos, solicitar a documentação necessária e promover seus cadastramentos em seu sistema de informação SCNES. Este software foi desenvolvido pelo DATASUS para que os gestores municipais e estaduais desses estabelecimentos pudessem enviar os dados dessas empresas de forma digital e eletrônica para o ministério da saúde, evitando assim mais burocracia e geração

desnecessária de formulário de papel. O software SCNES<sup>11</sup> pode ser adquirido pelo site do DATASUS e é de uso da administração pública, porém como é de livre distribuição na internet é possível a qualquer cidadão fazer o download e instalar em seu computador pessoal em sua casa, porém durante o processo de instalação é exigido que se crie um usuário e se escolha um perfil para este usuário. Segundo DATASUS (2009a, p. 26), cada perfil habilita funcionalidades diferentes. Alguns dos perfis disponíveis para a escolha são:

- Estabelecimento: onde o usuário será o próprio estabelecimento de saúde que pode usar o software para enviar suas informações para cadastro ou alteração para o órgão que é responsável pela gestão do seu estabelecimento, que geralmente é uma secretaria municipal de saúde;
- Secretaria Municipal de Saúde: quem escolhe esse perfil geralmente é uma secretaria municipal de saúde que é gestora dos estabelecimentos lotados em seu município, sendo este de pequeno porte, porém não possui gestão plena sendo assim subordinado à Secretaria Estadual de Saúde, logo todas as informações sobre seus estabelecimentos são transmitidas de forma digital para a Secretaria Estadual e é esta quem faz o envio das informações ao Ministério da saúde;
- Secretaria Municipal de Saúde com Gestão Plena: Este perfil é indicado para municípios de grande porte. Neste caso todas as informações dos estabelecimentos são transmitidas de forma digital diretamente para o Ministério da Saúde;
- Secretaria Estadual de Saúde: Este perfil deve ser selecionado pela Secretaria Estadual de Saúde de cada Estado pois nele existe funcionalidades que permitem receber os dados que são transmitidos pelos municípios de pequeno porte para serem repassados posteriormente ao Ministério da Saúde.

A seleção de um perfil é obrigatória durante a primeira execução do software SCNES e além da escolha do perfil pode-se criar um usuário e uma senha de acesso e existe um formulário no qual deve ser preenchido com os dados fidedignos do usuário que utilizará o programa, pois posteriormente serão validados. Sendo assim

---

<sup>11</sup> <http://cnes.datasus.gov.br/pages/downloads/aplicativos.jsp>

um cidadão comum e idôneo jamais poderia ter acesso oficial ao software uma vez que não existe um perfil específico criado para ele, pois não é de interesse do DATASUS que ele utilize.

As principais operações possíveis de execução no software SCNES são inclusão, alteração, exclusão e transmissão de informações sobre os estabelecimentos de saúde ao Ministério da saúde. O software possui um banco de dados local e off-line, porém ele é sincronizado mensalmente com o banco de dados federal do DATASUS onde são armazenados os dados desses estabelecimentos a nível Brasil através de atualizações via internet. As rotinas diárias e forma de utilização dos usuários deste software baseiam-se em legislações publicadas no Diário oficial da União. São portarias, notas técnicas, leis das quais os usuários precisam seguir e se adequar. Uma dessas regras definidas é a obrigatoriedade da transmissão de dados de sua base local para os servidores do Ministério da Saúde mensalmente (BRASIL. Portaria nº 682, Diário Oficial da União, 2010). Dessa forma, todas as alterações que são feitas na base de dados local são repassadas para o servidor do DATASUS para validação e aprovação e posterior publicação no portal de divulgação dos dados dos estabelecimentos à sociedade. Segundo disponibilizado por DATASUS (2015a) na seção sobre limites de competências, definiu-se um cronograma que todos os municípios devem cumprir em relação à transmissão de suas bases de dados locais para os servidores do Ministério da saúde. Analisando-se o cronograma vemos que os municípios podem transmitir seus dados ao Ministério da Saúde num determinado período dentro do mês e este período não é padrão pois varia todo mês. Dentro deste período os gestores podem transmitir seus dados locais uma vez por dia. Ao realizar o envio dos dados, o servidor recebe o arquivo, armazena, e durante a madrugada ele faz o processamento do arquivo recebido, caso não existam inconsistências, no dia seguinte os dados já ficam disponíveis para visualização no site de publicação das informações sobre os estabelecimentos. Porém, pode ocorrer o carregamento parcial das informações alteradas, pois durante este período do mês aberto para transmissões, cada gestor pode transmitir dados todos os dias fazendo várias alterações, sendo assim esse é um período transitório e incerto em relação a veracidade das informações sobre os estabelecimentos. Voltando a analisar o cronograma, percebe-se que existe um período dentro de cada mês onde não é permitido o envio de dados aos servidores, pois neste curto espaço de tempo os servidores do Ministério da Saúde, processam e fazem a consistência de todas as

informações recebidas. É durante esse período que se tem as informações oficiais e definitivas sobre estes estabelecimentos, pois nesse período o servidor não aceita requisições de envio de novos dados.

### **3.5 ANDROID**

Android é o sistema operacional open source que foi desenvolvido com a participação da empresa Google. Nesta seção do trabalho serão apresentados detalhes deste sistema. Primeiramente será apresentada um pouco de sua história de forma um pouco mais detalhada e na sequência serão mostradas as características de sua arquitetura.

#### **3.5.1 Histórico do Android**

A base do sistema operacional Android surgiu em 2003, na cidade de Palo Alto na Califórnia. Ele foi desenvolvido por Andy Rubin, Rich Miner, Nick Sears e Chris White. Estes empresários já iniciados no ramo da tecnologia, foram os fundadores da Android, Inc. A ideia principal desta empresa era a criação de dispositivos móveis mais inteligentes e que estivessem mais cientes das preferências e da localização dos seus usuário. No início a empresa desenvolvia todos os seus projetos de forma secreta. Em 2005 a Google adquiriu a empresa Android, Inc, nascendo assim a sua nova divisão de pesquisa em tecnologia móvel. Na mesma época os sistemas que dominavam o mercado de mobile eram o *Windows Mobile*, da Microsoft, e o iOS, da Apple e estas não se viram ameaçadas com os planos do Google e seus protótipos planejados juntamente com seus contratos de parcerias com algumas empresas da área de hardware. Foi somente a partir de 2007 que um grande passo foi dado e que marcou sua história fazendo outras empresas da mesma divisão de mercado começarem a se preocupar com suas posições no mercado, os grandes fabricantes como Samsung, Sony, HTC, operadoras como as americanas Sprint Nextel e T-Mobile, e fabricantes de hardware como Qualcomm e a Texas Instrumentes, além do próprio Google, reuniram-se em um consórcio de tecnologia e fundaram a OHA<sup>12</sup> (MEYER, 2016). Várias outras grandes empresas do ramo de telefonia, hardware, semicondutores e

---

<sup>12</sup> Open Handset Alliance - <http://www.openhandsetalliance.com>

de marketing e comércio participam e contribuem para a sua evolução. Todas essas empresas compartilham suas expectativas e *insights* nos congressos promovidos pela OHA(Open Handset Alliance). O principal objetivo deste consórcio é a busca de um padrão de dispositivo móvel aberto e que atenda as necessidades atuais de todos. Uma vez que os membros pertencem aos diversos setores que envolvem a concepção de um smartphone elas possuem conhecimento suficiente para saber o que é tendência no mercado e quais são as necessidades dos consumidores. Como descrito na seção sobre a filosofia de governança em ANDROID OPEN SOURCE PROJECT (2016), ao juntar todas as experiências dessas empresas pode-se manter o Android de uma forma que seja interessante tanto para os usuários finais como para as empresas que vão utilizá-los em seus dispositivos para comercializá-los. De uma certa forma esse consórcio é o responsável por ditar as regras de como o sistema operacional irá se comportar para atender as necessidades dos consumidores finais.

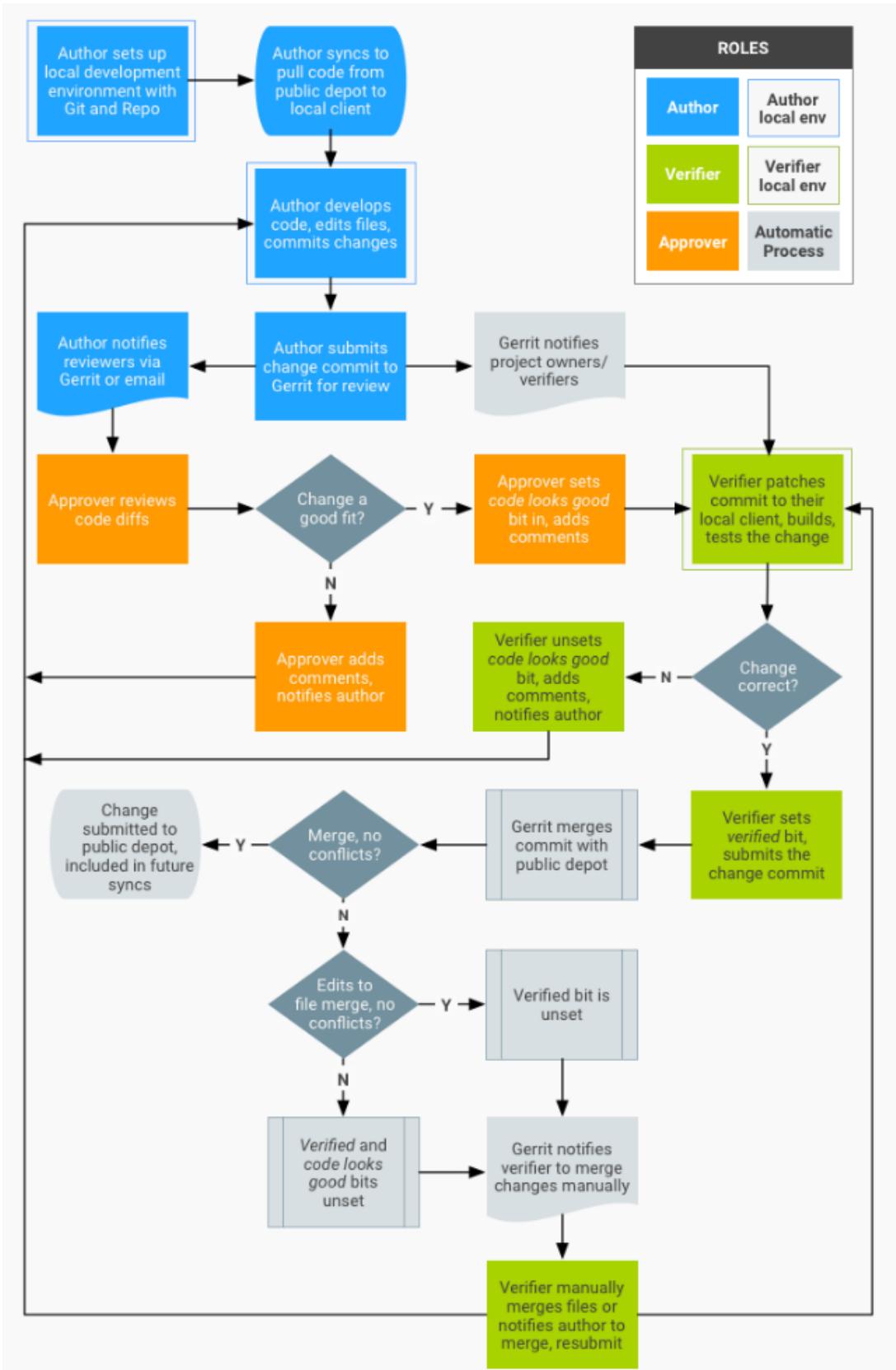
Um dos principais objetivos da Google ao participar na concepção do Android foi a criação de uma plataforma padrão na qual fosse utilizada por todos os dispositivos móveis. Através dessa iniciativa ela tinha a intenção de ganhar uma grande fatia no mercado de dispositivos. Para tal ela projetou uma plataforma cheia de atrativos com o objetivo de chamar a atenção tanto da comunidade de desenvolvedores quanto das grandes empresas fabricantes de dispositivos móveis demonstrando as infinitas possibilidades benéficas adquiridas na criação de dispositivos móveis compatíveis com sua plataforma. Quando se fala em compatibilidade entende-se a compatibilidade entre o hardware desenvolvido pela empresa interessada em lançar um novo modelo de dispositivo móvel. É para essa intenção do fabricante que o projeto da plataforma Android desenvolveu um sistema de validação de compatibilidade que pode ser visto na seção sobre compatibilidade do Android em ANDROID OPEN SOURCE PROJECT (2016). Podem-se citar algumas das principais características benéficas na adoção da plataforma Android como:

- A Linguagem usada na programação de aplicativos para a plataforma é o JAVA. Muito popular, bem difundida e bem documentada na atualidade o JAVA permitiu uma rápida adaptação e uma grande aceitação pelos desenvolvedores;
- A Plataforma Android é gratuita. Para os desenvolvedores que adotam esta plataforma para desenvolver suas aplicações os custos para tal são irrisórios;

- O código fonte da plataforma é livre e qualquer indivíduo pode adquiri-lo no portal do projeto oficial da plataforma na internet. Esse fato chama bastante a atenção das grandes fabricantes de *smartphones* e *tablets* pois os mesmos podem disponibilizar o Android em seus dispositivos vendidos sem a necessidade de aquisição de licença reduzindo assim os custos de sua produção, desde que sejam aprovados pelo programa de compatibilidade de dispositivos para Android;
- Possibilidade de um alto nível de personalização da plataforma. Tal característica permite que a plataforma possa ser adaptada e ajustada de acordo com o projeto no qual será inserida. Isto é um fato que pode atrair as grandes empresas pois elas podem tematizar e modificar a plataforma para se ajustar aos seus dispositivos físicos projetados.

Como dito, a concepção do sistema operacional Android não foi de exclusividade da empresa Google. Ela é a responsável pela gestão da plataforma, porém ela possui sua licença aberta e qualquer pessoa interessada pode participar para sua melhoria. Para isto basta o interessado procurar interagir com o AOSP (*The Android Open Source Project*) e seus subprojetos que somados garantem a evolução da plataforma. Como pode ser visto na seção de regras do projeto em ANDROID OPEN SOURCE PROJECT (2016), há uma série de regras e passos que podem ser acompanhados na Figura 3, que os interessados no projeto precisam adequar-se para poder contribuir.

*Figura 3 - Fluxograma da contribuição no projeto*



Fonte: ANDROID OPEN SOURCE PROJECT (2016)

A primeira versão do Android é de outubro de 2008. Como dito por Mercato

(2016), o primeiro smartphone disponibilizado comercialmente com esta versão foi o G1 fabricado pela empresa HTC. A partir de então, várias empresas começaram a adotar o Android como sistema padrão e sua popularização foi acontecendo e as empresas começaram a acompanhar e atualizar seus aparelhos lançados com as últimas versões oficiais lançadas. O Quadro 1 abaixo mostra o histórico completo de todas as versões lançadas até o presente momento da finalização deste projeto:

*Quadro 1 - Plataforma de nomes de versões e níveis de API*

<b>Code name</b>	<b>Version</b>	<b>API level</b>
Nougat	7.0	API level 24
Marshmallow	6.0	API level 23
Lollipop	5.1	API level 22
Lollipop	5.0	API level 21
KitKat	4.4 - 4.4.4	API level 19
Jelly Bean	4.3.x	API level 18
Jelly Bean	4.2.x	API level 17
Jelly Bean	4.1.x	API level 16
Ice Cream Sandwich	4.0.3 - 4.0.4	API level 15, NDK 8
Ice Cream Sandwich	4.0.1 - 4.0.2	API level 14, NDK 7
Honeycomb	3.2.x	API level 13
Honeycomb	3.1	API level 12, NDK 6
Honeycomb	3.0	API level 11
Gingerbread	2.3.3 - 2.3.7	API level 10
Gingerbread	2.3 - 2.3.2	API level 9, NDK 5
Froyo	2.2.x	API level 8, NDK 4
Eclair	2.1	API level 7, NDK 3
Eclair	2.0.1	API level 6
Eclair	2.0	API level 5
Donut	1.6	API level 4, NDK 2
Cupcake	1.5	API level 3, NDK 1
(no code name)	1.1	API level 2
(no code name)	1.0	API level 1

Fonte: ANDROID (2016b)

Durante a linha evolutiva das versões do Android, houve um fator que ramificou tal linha durante o desenvolvimento da versão 2.3, tal fato foi o surgimento dos *tablets*. Estes possuíam telas com resoluções e poder de processamento maiores do que os smartphones cujas versões do Android até então produzidas. Devido às grandes telas, os aplicativos produzidos para as versões 2.3 do Android para smartphones não se adaptavam muito bem aos *tablets*, ficando com sua interface gráfica deformada e diferente daquelas que haviam sido projetadas. Sendo assim para resolução destas questões, as versões da plataforma 2.3.1 e 2.3.2 em diante eram

destinadas a smartphone e as versões 3.0, 3.1 e 3.2 foram adaptadas, focando dispositivos como *tablets* que possuíam telas e resoluções maiores (DEITEL; DEITEL; WALD, 2016, p. 8).

Esta separação das plataformas, uma para *tablet* e uma para smartphone, fez com que houvesse uma preocupação quanto aos frameworks usados pelos desenvolvedores durante sua criação de aplicativos. Já que houve então uma separação os desenvolvedores deveriam se preocupar agora em ter de utilizar mais ferramentas para poderem desenvolver um mesmo aplicativo, deste modo teriam o trabalho dobrado. Pensando em solucionar esta questão a Google tomou a iniciativa de modificar a plataforma de uma maneira que se pudesse unificar novamente estas linhas divisórias paralelas no desenvolvimento de aplicativos. Para tal foi desenvolvido a versão do Android 4.0 cuja proposta era a criação de uma API unificada que pudesse ser utilizada tanto para criar aplicações para *smartphones* quanto para *tablets* (SCHMIDT, 2016).

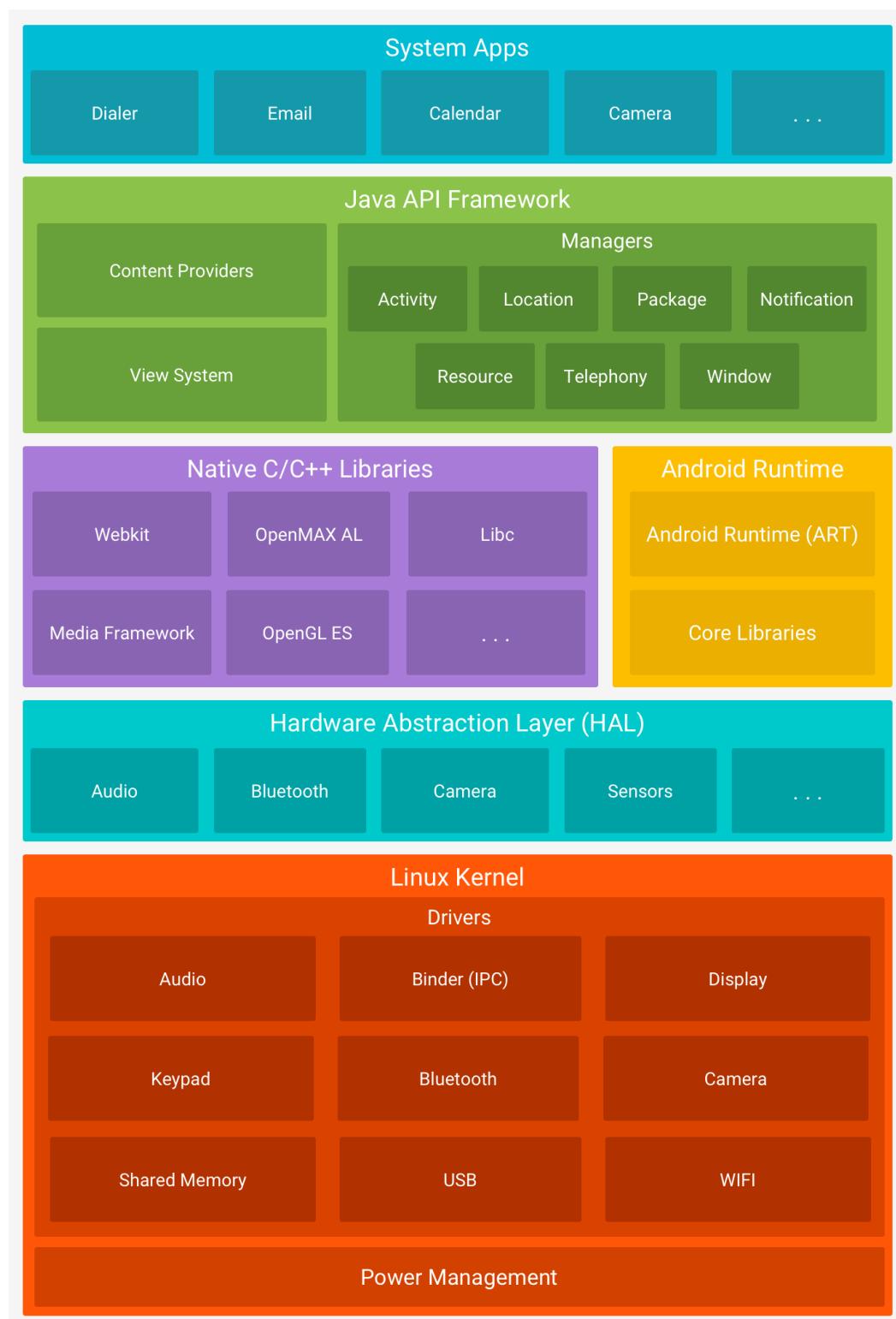
Ao longo das atualizações e melhoramentos na plataforma, novas evoluções tecnológicas foram surgindo no mercado como *SmartTvs*, sistemas operativos de multimídia para veículos automotivos e dispositivos vestíveis, como relógios e com elas a possibilidade de introduzirem a plataforma Android nesse novo universo. Dessa forma, a partir da versão 5.0 a possibilidade de desenvolvimento de aplicativos para essa nova gama de aparelhos tornou-se possível (ALVES, 2016).

### **3.5.2 Arquitetura do Android**

Segundo pode ser visto em ANDROID (2016b), a arquitetura da plataforma possui em seu nível mais baixo e mais perto do hardware, que inclusive é quem gerencia o mesmo, o Linux *kernel*. É ele quem está por trás do funcionamento do Android. Esse *Kernel* fornece os drives para a comunicação com o hardware, como por exemplo os drivers para comunicação com a câmera, com o teclado e com o autofalante. Ele também realiza o gerenciamento de processos, Threads e de memória. O Android é uma plataforma que executa sobre o sistema operacional Linux. Existem muitas praticidades na utilização de um sistema operacional *open source* como o Linux, uma delas é o fato de sua gratuidade, outras são questões como a praticidade de já se ter o gerenciamento de memória, processos, threads, segurança implementados e gerenciados. Dessa forma tira-se proveito inclusive da questão da

possibilidade de execução em multitarefa. A Figura 4 abaixo exibe todas as peças da arquitetura:

Figura 4 - Arquitetura da plataforma Android



Fonte: ANDROID (2016b)

Acima e em conjunto com o *kernel* do Linux existe a HAL<sup>13</sup>(*Hardware Abstraction Layer*). Esta camada é composta por vários módulos e interfaces que fornecem a capacidade de interação com uma linguagem de alto nível como o Java.

Logo acima da camada HAL, existem as bibliotecas que são nativas da plataforma Android e que são escritas normalmente na linguagem C ou C++. São elas que se comunicam com a camada HAL. Essas bibliotecas geralmente proveem funcionalidades como o suporte ao *SQLite*, ao motor de renderização de imagens 2D e imagens 3D, renderização de páginas WEB, entre outros.

Ao lado existe na estrutura da plataforma o *Android Runtime* que é composto pelas *Core Libraries* juntamente com a máquina virtual do Android. As *Core Libraries* são as responsáveis por disponibilizar ao desenvolvedor tudo que é utilizável pelo Java SE. A ART (*Android Runtime*) é a máquina virtual do Android utilizada para a execução dos aplicativos. Como os aplicativos são desenvolvidos e compilados em Java eles precisam de uma máquina virtual para poderem ser interpretados.

Até a versão 4.4 do Android uma outra máquina virtual era utilizada pelo Android, a *Dalvik*, esta era a máquina virtual padrão, porém o Android fornecia ao usuário a possibilidade de experimentar a ART que nesta versão era opcional (HAMANN, 2016). A partir da versão 5.0 a máquina virtual padrão tornou-se a ART.

Esta máquina virtual ART foi desenvolvida tendo como propósito melhorar a performance dos aplicativos na plataforma Android. Ela é capaz de executar os aplicativos mais rapidamente, pois em vez de trabalhar com o sistema “*Just-in-Time*” do *Dalvik*, é realizada uma compilação a frente do tempo. Os códigos são pré-compilados na linguagem de execução já durante a instalação dos aplicativos. Isso significa que a instalação dos softwares pode ser mais demorada, mas também representa mais rapidez na sua execução justamente pelo fato de a virtualização já ter acontecido e ter sido gravada na memória do dispositivo.

Diferentemente da máquina utilizada anteriormente, a *Dalvik*, a ART foi criada especificamente para o Android e para esse tipo de contexto.

A máquina Virtual do atual do Android não executa arquivos em formato *bytecode* com a extensão *.class*, assim como a JVM(Máquina Virtual do Java) na plataforma Java SE o faz, mas sim um arquivo em um formato *.dex* (*Dalvik Executable*). Na plataforma Android, o compilador, além de gerar um arquivo em

---

<sup>13</sup> <https://source.android.com/devices/index.html#Hardware Abstraction Layer>

formato de *bytecode* e extensão *.class*, executa um passo a mais utilizando como base esse arquivo para compilá-lo em um novo com formato *.dex* para que ele então possa ser executado pela ART.

Acima dessas camadas existe o *framework* de aplicações do Android. Neste *framework* estão as funcionalidades que vão permitir ao desenvolvedor criar seus aplicativos. Dentre esses frameworks podemos citar o *Telefony manager* que disponibiliza a funcionalidade de realizar ligações telefônicas, *Resource manager* que permite ao desenvolvedor o acesso a memória interna do dispositivo, além do *Location manager* que permite a possibilidade a utilização do GPS do dispositivo. Esta parte da plataforma provê todo o gerenciamento do Android pois ela faz a comunicação com as APIs de mais baixo nível da plataforma. Quando um desenvolvedor cria um aplicativo para a plataforma ele utiliza esse *framework* para tal. Todos os aplicativos nativos do Android também foram criados utilizando esse *framework*, dessa forma ambos os tipos de aplicações, nativas ou terceiras, estão no mesmo nível e podem ser substituídos sem causar prejuízos à plataforma.

No topo desta estrutura temos a camada de aplicações onde ficam as aplicações do Android, tanto as nativas como o aplicativo de telefonia, de contatos, de mensagens, quanto as aplicações desenvolvidas por terceiros. A plataforma Android percebe as aplicações desenvolvidas por terceiros assim como as nativas da mesma forma. A grande flexibilidade desta plataforma já mencionada anteriormente está no fato de os aplicativos criados pelos desenvolvedores estarem nessa mesma camada das aplicações nativas. Logo é possível desenvolver um aplicativo que substitua outro aplicativo nativo com o objetivo de personalizar toda a interface gráfica do usuário.

## **4 ESTADO DA ARTE**

Neste capítulo são apresentados dois sistemas que foram desenvolvidos focando o mesmo tema do trabalho desenvolvido. A busca pelo estado da arte em localização de estabelecimentos de saúde utilizou-se de vários repositórios de trabalhos acadêmicos na Internet. Para localizá-los mais rapidamente a ferramenta Google Acadêmico<sup>14</sup> foi utilizada. Os resultados obtidos tinham baixa compatibilidade principalmente pelo fato dos sistemas propostos ou desenvolvidos relatados nos trabalhos serem totalmente dependentes de uma conexão com a internet. Um sistema selecionado por possuir características próximas aos do atual será descrito na sequência. Ele foi selecionado principalmente pelo fato de se utilizar de um aplicativo como ferramenta de busca.

### **4.1 SISTEMA MÓVEL DE GEOLOCALIZAÇÃO E GEOPROCESSAMENTO PARA LOCAIS DE ATENDIMENTO EM SAÚDE**

O *Journal of Health Informatics* (*J. Health Inform.*) é uma publicação oficial da Sociedade Brasileira de Informática em Saúde. Na sua terceira edição do ano de 2014 os autores Sarrinho e Campos (2014) relataram sua experiência no desenvolvimento do projeto SAMob (Sistema Móvel de Geolocalização e Geoprocessamento para Locais de Atendimento em Saúde). O projeto SAMob tinha o objetivo de desenvolver um sistema de atendimento mobile capaz de identificar, compartilhar e qualificar centros de atendimento de saúde próximos à geolocalização atual de seus respectivos usuários via dispositivo móvel.

Com o desenvolvimento do sistema os autores esperavam fornecer aos seus usuários uma ferramenta para auxiliar na busca por locais que prestam atendimento à saúde não só em caráter emergencial, mas também local que prestam atendimento eletivo, dessa forma, aproveitando a mobilidade e praticidade dos dispositivos móveis. Os autores utilizaram duas plataformas para desenvolver o sistema: a plataforma mobile com sistema operacional Android na qual foi desenvolvido um cliente para que os seus usuários pudessem aproveitar as funcionalidades nativas dos dispositivos móveis atuais como naveabilidade GPS, onde seria possível realizar a busca e a

---

<sup>14</sup> <https://scholar.google.com.br/>

qualificação dos locais de atendimento e a plataforma web utilizando a *Google App Engine* na qual foi desenvolvido uma aplicação servidora e administradora que disponibilizaria a sincronização de dados com os cliente *mobile*. As funcionalidades da aplicação cliente foram definidas como: obter a qualificação do local de atendimento; qualificar o local de atendimento, possibilitando comentar ou tirar fotos do local onde se encontra; obter a melhor rota para chegar ao local de atendimento utilizando a visualização em mapa; obter dados do local de atendimento; buscar por geolocalização e parâmetros de filtragem. As funcionalidades da aplicação servidora seriam: manutenção de usuários; busca por usuários; manutenção dos locais de atendimento; busca por local de atendimento.

O artigo desenvolvido pelos autores apresenta algumas características da modelagem do sistema desenvolvido através de diagramas UML. São apresentados de forma resumida os diagramas de casos de uso, de classes e de sequência.

Com a conclusão do desenvolvimento do sistema, os autores obtiveram duas aplicações rápidas, simples e interativas. Para avaliar a usabilidade do sistema desenvolvido, foi feita uma pesquisa com 15 alunos do Universidade Estadual de Feira de Santana - UEFS, para tal o sistema foi apresentado aos entrevistados que também tiveram de responder um questionário de 30 questões referentes à usabilidade do sistema. Após o resultado do teste, os autores perceberam um alto nível de satisfação em relação aos vários quesitos analisados no teste. Ao final do artigo os autores concluem sua descrição do sistema classificando-o como: uma rede social que compartilha informações de atendimento médico-hospitalar de uma forma interativa e de baixo custo. Estes ainda acrescentam na conclusão que houve sucesso no projeto pois todas as funcionalidades previstas funcionam da forma que foram planejadas.

Como trabalhos futuros os autores sugerem: a adaptação da aplicação cliente para outras plataformas móveis iPhone, *Windows Phone* e *Firefox OS*; a adaptação da aplicação cliente para possibilitar a inclusão de sugestão de melhorias; integração e possibilidade de compartilhamento das opiniões sobre os estabelecimentos nas redes sociais e a integração do sistema com os sistemas corporativos de saúde da rede pública. Em relação a esta última sugestão do autor para trabalhos futuros, pode-se dizer que o aplicativo que foi desenvolvido neste projeto de conclusão de curso atende alguns quesitos, uma vez que utiliza em sua base de informação os dados oficiais retirados de um sistema corporativo de saúde pública, o CNES.

As buscas por trabalhos acadêmicos que relatavam sistemas de localização

de estabelecimentos de saúde não trouxeram grandes resultados. Para o aprofundamento das buscas por sistemas similares utilizou-se a loja de aplicativos da Google. Uma grande quantidade resultados foram obtidos, porém como são aplicativos de código fechado, não se obteve muitas informações referentes a sua arquitetura interna. O aplicativo que será descrito na sequência possui muitas características parecidas com o que está sendo desenvolvido pelo autor deste trabalho, principalmente em relação a suas funcionalidades. Devido a isto ele foi o selecionado dentre os vários encontrados.

## 4.2 MEDSQUARE

*MedSquare*, segundo Medsquare (2015), é um aplicativo desenvolvido pela empresa WINTIME IPC com o objetivo de melhorar a saúde e o atendimento dos pacientes nos estabelecimentos de saúde. Ele possui duas versões, uma para plataforma Android e outra para plataforma IOS. Suas funcionalidades permitem que o usuário: encontre um estabelecimento de saúde mais próximo de sua posição geográfica atual; busque estabelecimentos pelo nome; busque estabelecimentos por especialidade médica. Os usuários do aplicativo podem avaliar os estabelecimentos de saúde e podem registrar no aplicativo o tempo de espera por atendimento no estabelecimento que este se encontra. É possível ler os comentários e avaliações dos estabelecimentos e também conversar em tempo real com quem está num determinado estabelecimento. Para que todas estas funcionalidades possam ser contempladas, o aplicativo exige do dispositivo uma conexão com a internet pois o seu banco de dados fica em um servidor no qual ele se conecta para obter informações. O site institucional do aplicativo informa que os proprietários dos estabelecimentos podem solicitar seu cadastro na base de dados da aplicação mediante a um contato prévio com a empresa. Logo, o aumento do número de estabelecimentos cadastrados está diretamente ligado com o interesse dos seus proprietários em participar do processo de cadastro.

Em agosto de 2015 foi publicada na loja de aplicativos do Google a versão 2.0.2 do aplicativo *MedSquare*. Dentre o novo visual e algumas novas funcionalidades estava a forma de alimentação de seu banco de dados de estabelecimentos. Ele agora possuía informações de todos os estabelecimentos cadastrados na base federal do CNES (Cadastro Nacional dos Estabelecimentos de Saúde). Essa estratégia tornou

possível um aumento significativo na quantidade de resultados possíveis durante uma busca por algum estabelecimento.

Ao analisar os sistemas escolhidos para conhecer o estado da arte sobre o tema, percebe-se que ambos são sistema totalmente dependentes de uma conexão com a internet para proverem suas funcionalidades. Ambos são sistemas que usam a arquitetura cliente servidor, no qual o aplicativo, que é o cliente, solicita os dados sobre os estabelecimentos ao banco de dados que está armazenado em um servidor. E, ambos, se utilizam das várias possibilidades promovidas pelas plataformas mobile para promover facilidades aos usuários, como a possibilidade de utilizar o GPS do dispositivo, por exemplo. Percebeu-se também que a ação de inclusão de novos estabelecimentos ao banco de dados é totalmente dependente do interesse do próprio estabelecimento. Caso ele não se manifeste, ele não aparecerá no banco e os usuários não saberão de sua existência. No caso do aplicativo *MedSquare*, depois de sua última atualização esta situação deixa de ocorrer pois, segundo a própria empresa divulga, agora eles possuem em seu banco de dados todas as informações dos estabelecimentos de saúde que estão na base de dados do CNES. Entretanto, não há informações se estes dados são atualizados frequentemente.

## **5 DESENVOLVIMENTO DO PROJETO**

Neste capítulo são apresentadas as etapas que envolveram o desenvolvimento do projeto. Na primeira seção serão tratadas as questões relacionadas a ferramenta de captura de dados, nas próximas serão descritos todos os processos envolvidos no desenvolvimento do aplicativo para a plataforma Android, assim como suas dificuldades e resultados obtidos nesta etapa.

### **5.1 A APLICAÇÃO DE CAPTURA DE DADOS**

Foi desenvolvida uma aplicação utilizando a linguagem JAVA em sua versão 8. Como IDE, utilizou-se o *Netbeans*<sup>15</sup> em sua versão 8. Esta aplicação utilizou uma biblioteca Java HTML Parser JSOUP em sua versão 1.8.2. A utilização dessa biblioteca de classes permitiu importar para dentro do programa e manipular os elementos das páginas HTML retornadas nas consultas feitas as URLs do portal do CNESNet na seção de busca por estabelecimentos de saúde para que se pudesse extrair, limpar e converter os dados que foram usados na base de dados do aplicativo. A Figura 5 ilustra a página que é retornada ao requisitar ao portal uma busca.

---

<sup>15</sup> <https://netbeans.org/>

Figura 5 - Página de detalhamento do estabelecimento

The screenshot shows a web browser window titled "Ficha Estabelecimento - M x". The URL is "cnes.datasus.gov.br/Exibe\_Ficha\_Estabelecimento.asp?VCo\_Unidade=4205403157245&VEsta". The page is part of the "Ministério da Saúde" and "CNESNet Secretaria de Atenção à Saúde" websites. It features a banner for "Cadastro Nacional de Estabelecimentos de Saúde". The main navigation menu includes "Home", "Institucional", "Serviços", "Relatórios", "Consultas", "Documentação", and "Fale Conosco". The page title is "Estabelecimento de Saúde". Below the title, there is a header with the text "CADASTRADO NO CNES EM: 24/7/2004 ULTIMA ATUALIZAÇÃO EM: 18/5/2015 DATA DE ATUALIZAÇÃO LOCAL: 5/5/2015". There are three buttons: "Veja onde se localiza:", "Exibir Ficha Reduzida por Competência", and "Exibir Ficha Reduzida Atual". The main content area contains various fields for establishing identification, location, and operational details. At the bottom, there is a section titled "Módulos:" with a grid of buttons for different modules.

Nome:	CNES:	CNPJ:			
HOSPITAL UNIVERSITARIO	3157245				
Nome Empresarial:	CPF:	Personalidade:			
UNIVERSIDADE FEDERAL DE SANTA CATARINA	--	JURÍDICA			
Logradouro:	Número:	Telefone:			
CAMPUS UNIVERSITARIO	S/N	(48) 3721-2074			
Complemento:	CEP:	UF:			
	88040900	SC			
Tipo Estabelecimento:	Sub Tipo Estabelecimento:	Esfera Administrativa:	Gestão:		
HOSPITAL GERAL	TRINIDADE	FEDERAL	DUPLA		
Natureza da Organização:	Dependência:				
ADMINISTRACAO INDIRETA - AUTARQUIAS	MANTIDA				
Número Alvará:	Órgão Expedidor:	Data Expedição:			
3756/025 SERIE PDVS	SMS	14/10/2002			
Módulos:					
Básico	Conjunto	Ambulatorial	Hospitalar	Mantenedora	Profissionais
Habilidades	Regras Contratuais	Contrato de Gestão	Incentivos	Equipes	Residência Terapêutica
Telessaúde	Org. Parceiras	Gér/Adm (Terceiro)			

Fonte: DATASUS (2015b)

Após a conclusão do desenvolvimento desta ferramenta foram feitas as extrações e a montagem da base. A tecnologia da base de dados usada foi o *SQLite*.

*SQLite* é um banco autocontido, compacto, com suporte nativo no Android e sem necessidade de configuração ou instalação. Isto torna-o a escolha natural para um ambiente em que devemos prezar por desempenho, disponibilidade de memória e praticidade de uso (LUZZI, 2015).

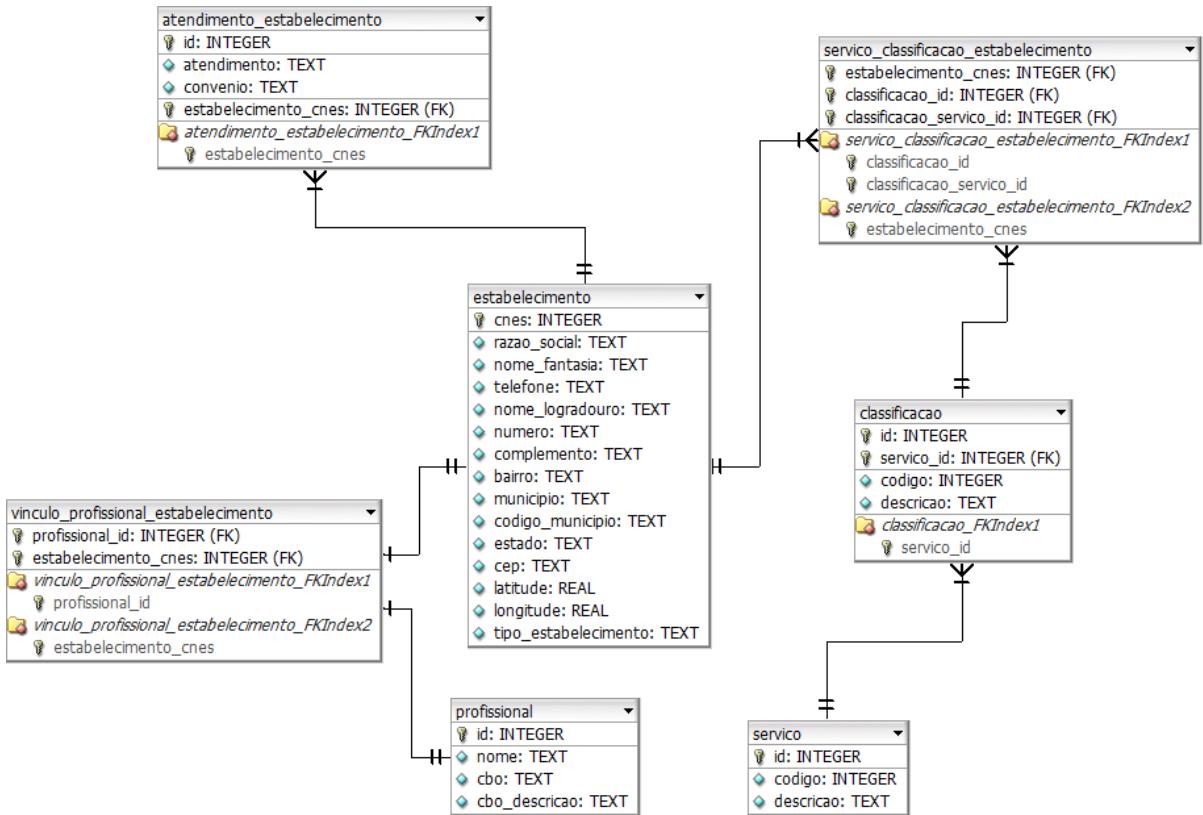
Foi utilizada a ferramenta *DB Designer Fork*<sup>16</sup> para auxiliar no processo de construção do banco de dados. Segundo Oliveira (2008), esta ferramenta *open source* é uma versão melhorada da ferramenta *fabFORCE DBDesigner 4*<sup>17</sup>, além de ser simples, prática e fácil de usar ela permite a configuração de uma conexão direta com

<sup>16</sup> <http://sourceforge.net/projects/dbdesigner-fork>

<sup>17</sup> <http://www.fabforce.net/dbdesigner4>

os sistemas gerenciadores de banco de dados: *MySQL*, *SQLite*, *Oracle* e *SQL Server*. A ferramenta permite a modelagem de forma visual das tabelas e relacionamentos além da geração de scripts de criação do banco de dados na linguagem SQL para diversos gerenciadores de banco de dados. A Figura 6 apresenta a modelagem das tabelas do banco de dados *SQLite*.

Figura 6 - Modelagem do banco de dados



Fonte: Elaborada pelo autor

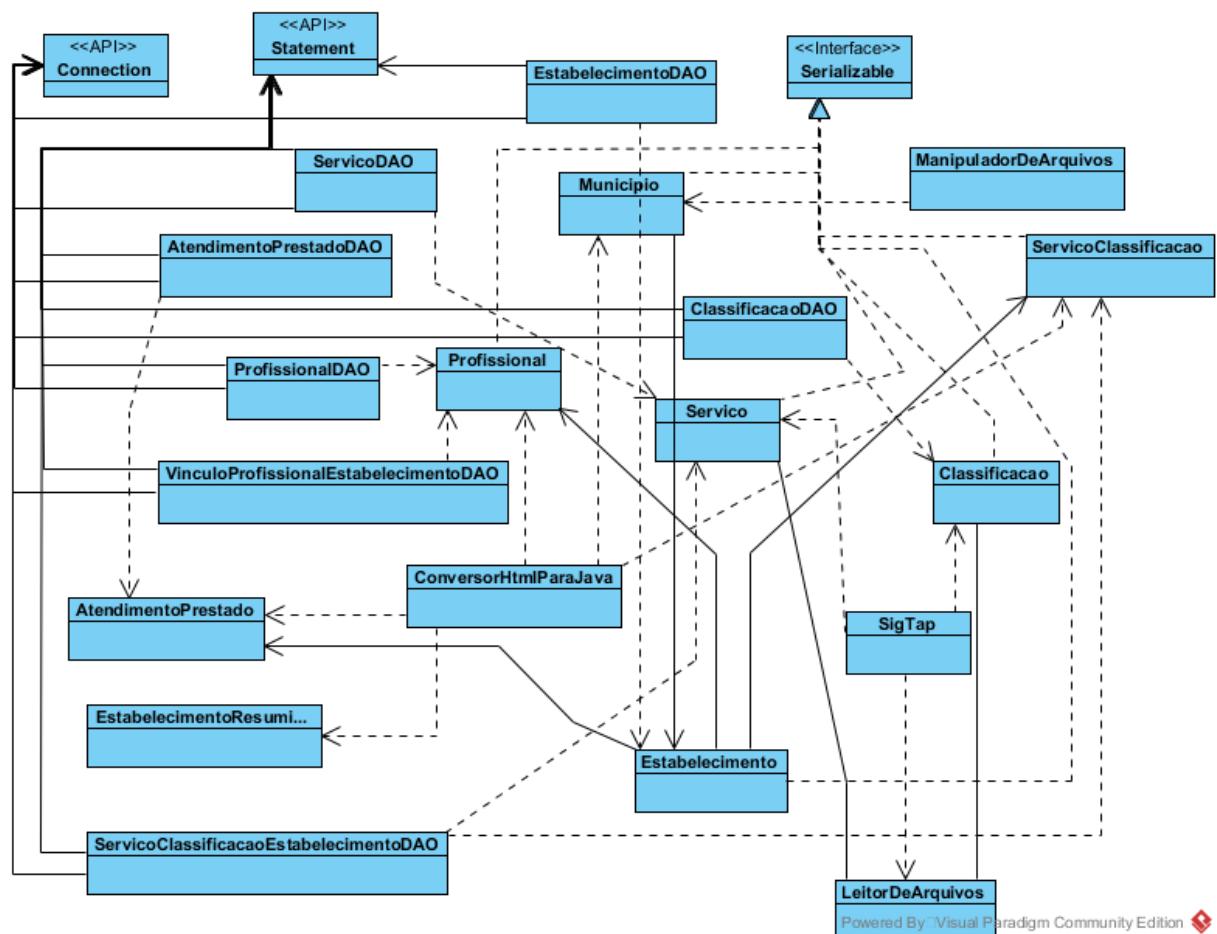
A tabela *estabelecimento* armazena os dados sobre os estabelecimentos de saúde assim como suas coordenadas geográficas. A tabela *profissional* armazena os dados dos profissionais que trabalham nos estabelecimentos, os dados armazenados são: o nome do profissional; o Código Brasileiro de Ocupação (CBO) e sua descrição. A tabela *vinculo\_profissional\_estabelecimento* registra quais profissionais trabalham em um estabelecimento formando um relacionamento muitos para muitos entre estas duas tabelas. A tabela *servico\_clasificacao\_estabelecimento* armazena os serviços oferecidos pelos estabelecimentos, ela cria um relacionamento muitos para muitos entre as tabelas *estabelecimento* e *classificacao*. As tabelas *servico* e *classificacao* armazenam os dados oficiais de todos os serviços e classificações cadastrados no

SIGTAP<sup>18</sup>(Sistema de Gerenciamento da Tabela de Procedimentos do SUS). Estas informações são disponibilizadas mensalmente e de forma atualizada ao cidadão pelo portal DATASUS do Ministério da Saúde através de arquivos no formato de texto.

Com a IDE *Netbeans* foi possível incluir a biblioteca *sqlite-jdbc*<sup>19</sup> no projeto para que as instâncias das classes responsáveis pela criação e conexão com bancos de dados *SQLite* pudessem ser utilizadas, assim, pôde-se focar o desenvolvimento na aquisição de informações através da captura dos dados das páginas HMTL e na importação dos dados advindos dos arquivos de texto baixados do site SIGTAP.

Para ter uma visão geral de todas as classes pensadas para a ferramenta de captura, foi desenvolvido um diagrama de classes em UML que pode ser visualizado na Figura 7.

*Figura 7 - Diagrama de classes*



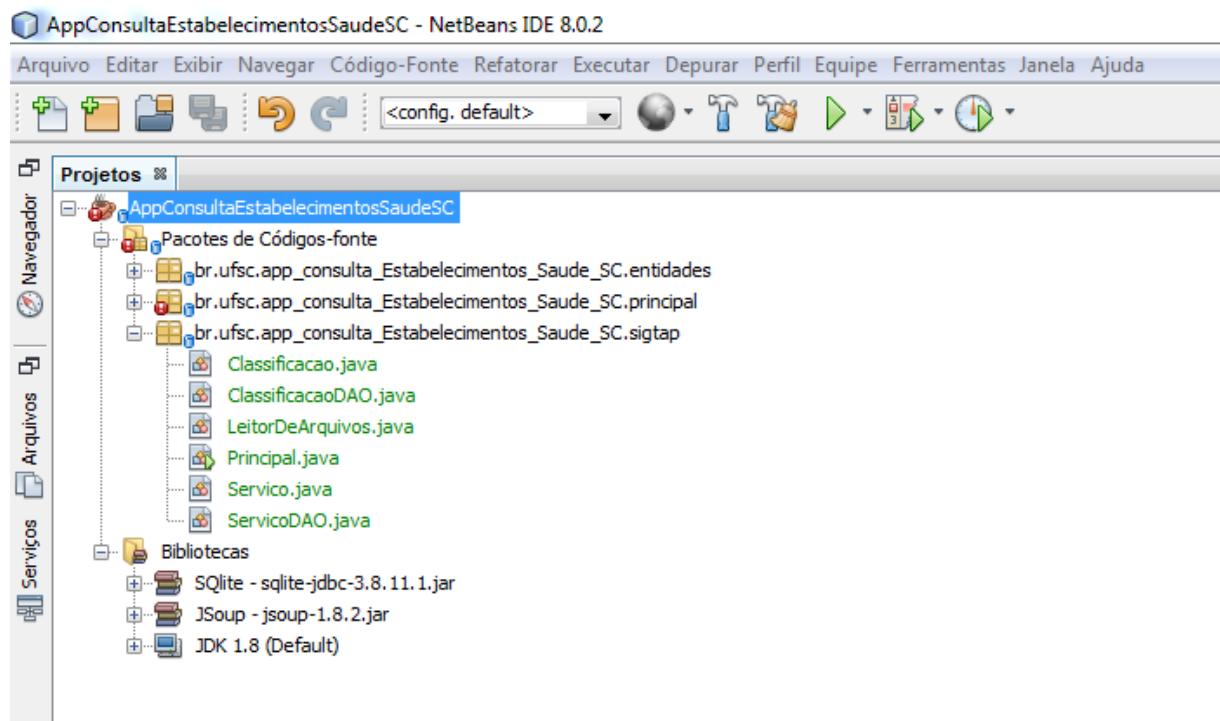
Fonte: Elaborada pelo autor

<sup>18</sup> <http://sigtap.datasus.gov.br/tabela-unificada/app/sec/inicio.jsp>

<sup>19</sup> <https://bitbucket.org/xerial/sqlite-jdbc>

Foram criadas algumas classes que compõem a funcionalidade de leitura de arquivos de texto desta aplicação desktop para realizar a extração dos dados contidos referentes as informações dos serviços e classificação dos SIGTAP para montagem da base de dados que será utilizada no aplicativo Android. A Figura 8 enumera as classes que servirão para este caso.

Figura 8 - Classes que fazem a leitura de arquivos de texto



Fonte: Elaborada pelo autor

Foram criadas as classes *Servico* e *Classificacao* que são o modelo de dados que auxiliam a classe *LeitorDeArquivos* a armazenar os serviços e suas classificações em memória para posteriormente serem persistidos em banco de dados com o auxílio das respectivas classes *ServicoDAO* e *ClassificacaoDAO*, desenvolvidas seguindo o padrão *Data Access Object* (DAO).

A classe *ConversorHtmlParaJava* pode ser considerada o coração da aplicação que faz a captura dos dados. Esta classe possui as instâncias dos objetos fornecidos pela biblioteca *JSoup* que se conectam ao servidor do DATASUS e manipulam o conteúdo das páginas HTML do site. Seus principais métodos podem ser visualizados na Figura 9.

Figura 9 - Principais métodos da classe ConversorHtmlParaJava

The screenshot shows the NetBeans IDE interface with the following details:

- Title Bar:** AppConsultaEstabelecimentosSaudeSC - NetBeans IDE 8.0.2
- Menu Bar:** Arquivo, Editar, Exibir, Navegar, Código-Fonte, Refatorar, Executar, Depurar, Perfil, Equipe, Ferramentas, Janelas, Ajuda.
- Toolbar:** Includes icons for New Project, Open Project, Save, Cut, Copy, Paste, Find, Replace, Run, Stop, and Help.
- Code Editor:** The main window displays Java code for a class named `Testando.java`. The code includes several methods with Javadoc comments and annotations like `@param` and `@return`.
- Project Explorer:** On the left, it shows a project structure with files like `ConversionParaJava.java`, `Testando.java`, and `main`.
- Properties and Tools:** On the right side, there are tabs for Properties, Services, and Analyzed.
- Status Bar:** Shows the line number (361), column number (150), file path (src\br\edu\ufsc\app\consulta\estabelecimentos\saude\service\impl\), and date (26/11/2010).

```
AppConsultaEstabelecimentosSaudeSC - NetBeans IDE 8.0.2
Arquivo Editar Exibir Navegar Código-Fonte Refatorar Executar Depurar Perfil Equipe Ferramentas Janelas Ajuda
[config, default] Pesquisar (Ctrl+F)
ConversionParaJava.java Testando.java
Código-Fonte Histórico + | - | Buscar | Refatorar | Executar | Depurar | Ferramentas | Sobre | Ajuda | Salvar | Fechar
64 * Método que retorna uma Collection do tipo ArrayList que armazenam dados
65 * dos municípios como Código IBGE, Nome, Quantidades de estabelecimentos que
66 * possuem e uma String que representa uma url para a página web onde é
67 * possível visualizar os estabelecimentos instalados no município.
68 *
69 * @return ArrayList com objetos do tipo Municipio
70 */
71 public ArrayList<Municipio> getMunicipios() {...68 linhas}
139
140 /**
141 * Método que retorna uma Collection do tipo ArrayList de objetos do tipo
142 * EstabelecimentoResumido que contém informações resumidas dos
143 * estabelecimentos do município
144 *
145 * @param municipio objeto do tipo Municipio com os dados resumidos sobre os
146 * estabelecimentos do município
147 * @return ArrayList de objetos do tipo EstabelecimentoResumido
148 */
149 public ArrayList<EstabelecimentoResumido> getEstabelecimentosDoMunicipio(Municipio municipio) {...72 linhas}
221
222 /**
223 * Método que retorna uma Collection do tipo ArrayList de objetos do tipo
224 * Estabelecimento que contém informações detalhadas do estabelecimento
225 *
226 * @param estabelecimentos ArrayList de objetos do tipo
227 * EstabelecimentoResumido
228 * @return ArrayList de objetos do tipo Estabelecimento
229 */
230 public ArrayList<Estabelecimento> getEstabelecimentosDoMunicipioDetalhados(ArrayList<EstabelecimentoResumido> estabelecimentos) {...158 linhas}
388
```

Fonte: Elaborado pelo autor

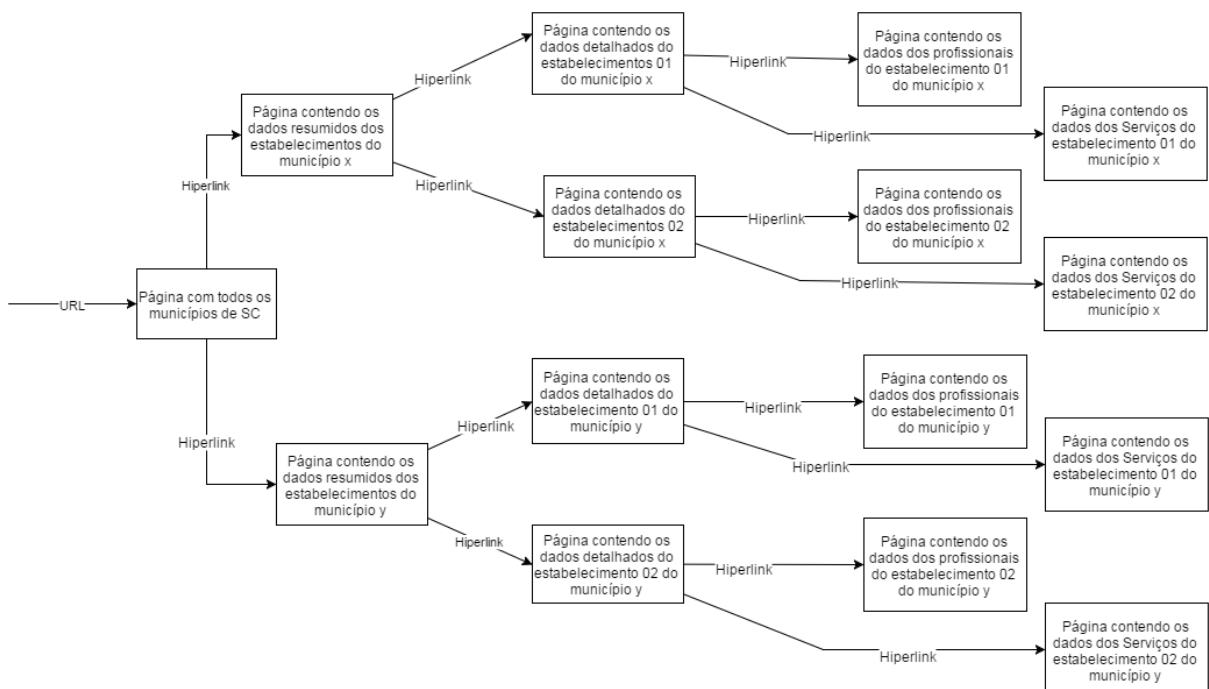
O método `getMunicípios` é o primeiro a ser executado. Ele faz uma requisição ao servidor através do endereço da página que lista todos os municípios de Santa Catarina e suas respectivas quantidades de estabelecimentos cadastrados, captura essas informações e salva em memória. Por sua vez, o método `getEstabelecimentosDoMunicípio` acessa a página que lista todos os estabelecimentos de cada município armazenando em memória dados como o nome do estabelecimento e o hiperlink que nos leva até a página detalhada de cada estabelecimento. Por fim o método `getEstabelecimentosDoMunicípioDetalhados` utiliza o `hiperlink` armazenado na etapa anterior para conectar a página que disponibiliza os dados detalhados de cada estabelecimento.

É importante ressaltar que existem vários tipos de estabelecimentos de saúde no banco de dados federal do MS. O Anexo A deste trabalho foi desenvolvido segundo (MINISTÉRIO DA SAÚDE, 2006) para descrever os tipo possíveis, porém para o escopo deste trabalho serão somente utilizados estabelecimentos dos tipos: Posto de Saúde; Centro de Saúde/Unidade Básica de Saúde; Policlínica; Hospital Geral; Hospital Especializado; Unidade Mista; Pronto Socorro Geral; Pronto Socorro Especializado; Consultório Isolado; Clínica Especializada/Ambulatório Especializado; Unidade de Serviço de Apoio de Diagnose e Terapia(SADT – clínicas que realizam

exames para diagnóstico); Centro de Parto Normal Isolado; Hospital / Dia- Isolado; Laboratório Central de Saúde Pública. Sendo assim o método `getEstabelecimentosDoMunicipioDetalhados` que faz uma iteração para cada estabelecimento, inicialmente analisa se o tipo do estabelecimento é compatível com o projeto, caso afirmativo ele inicia a sequência do processo de captura, caso negativo ele ignora o estabelecimento atual e passa para a próxima iteração.

Para que fosse possível conectar-se a uma página HTML utilizando as instâncias das classes disponíveis pela biblioteca `JSoup` foi importante definir um valor de parâmetro para o método `.timeout()` utilizado na chamada estática à classe que promove a conexão: `JSoup`. O parâmetro adicionado a este método define o tempo que o programa possui para tentar uma conexão bem-sucedida com o servidor de destino, caso esse tempo termine e a conexão não seja estabelecida, uma exceção é lançada informando a falha. Essa questão foi um grande desafio nesta etapa do projeto pois, precisou-se achar um valor ótimo que não fosse pequeno demais para que a exceção não chegasse a ser gerada, nem grande demais para evitar que o servidor fique com muitas requisições pendentes para atender.

Figura 10 - Árvore de links para navegação



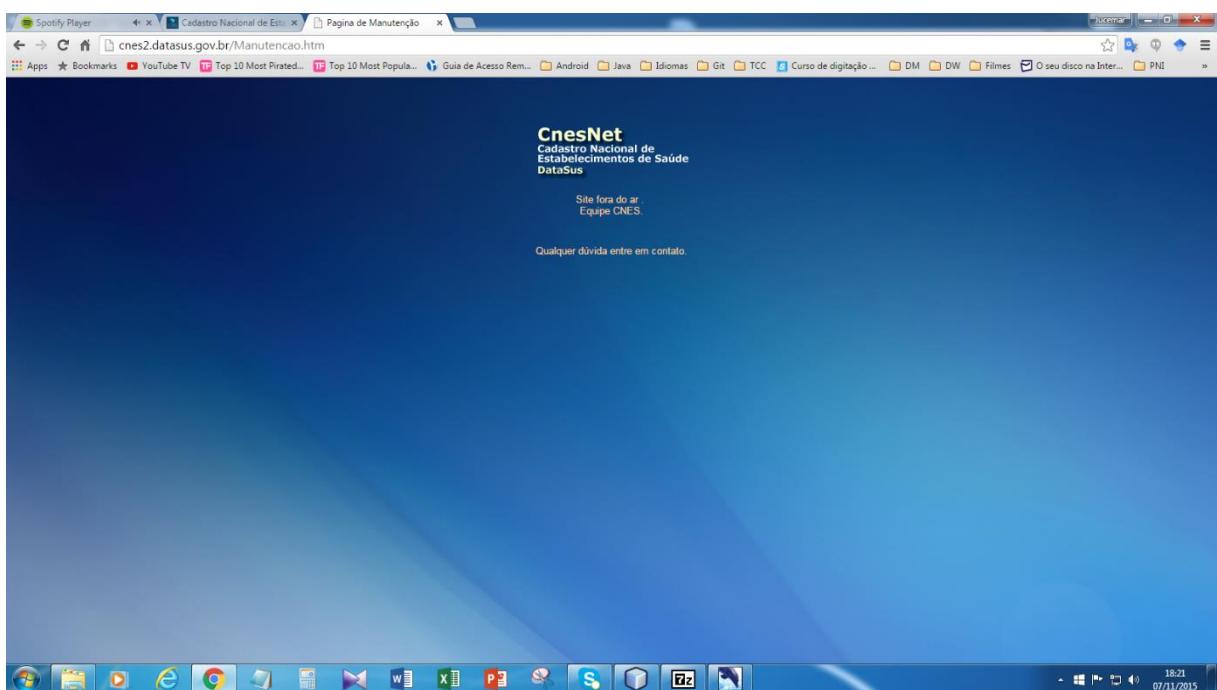
Fonte: Elaborado pelo autor

A Figura 10 demonstra a hierarquia dos links usados para navegar pelo site e

fazer as requisições ao servidor. Percebe-se claramente que a quantidade de requisições é grande. Logo, uma preocupação maior com o valor do tempo para cada requisição foi necessária para evitar que o mecanismo de segurança do servidor interprete as diversas requisições como um ataque do tipo negação de serviço (DDOS).

Várias tentativas foram feitas para encontrar um valor ótimo para o tempo de requisição, inicialmente definiu-se o valor como sendo 0. Neste contexto, ao iniciar a aplicação para realizar a captura dos dados percebeu-se que ao realizar 50% das requisições ao servidor, o mesmo começou a retornar, de forma esporádica e aleatória a página descrita na Figura 11.

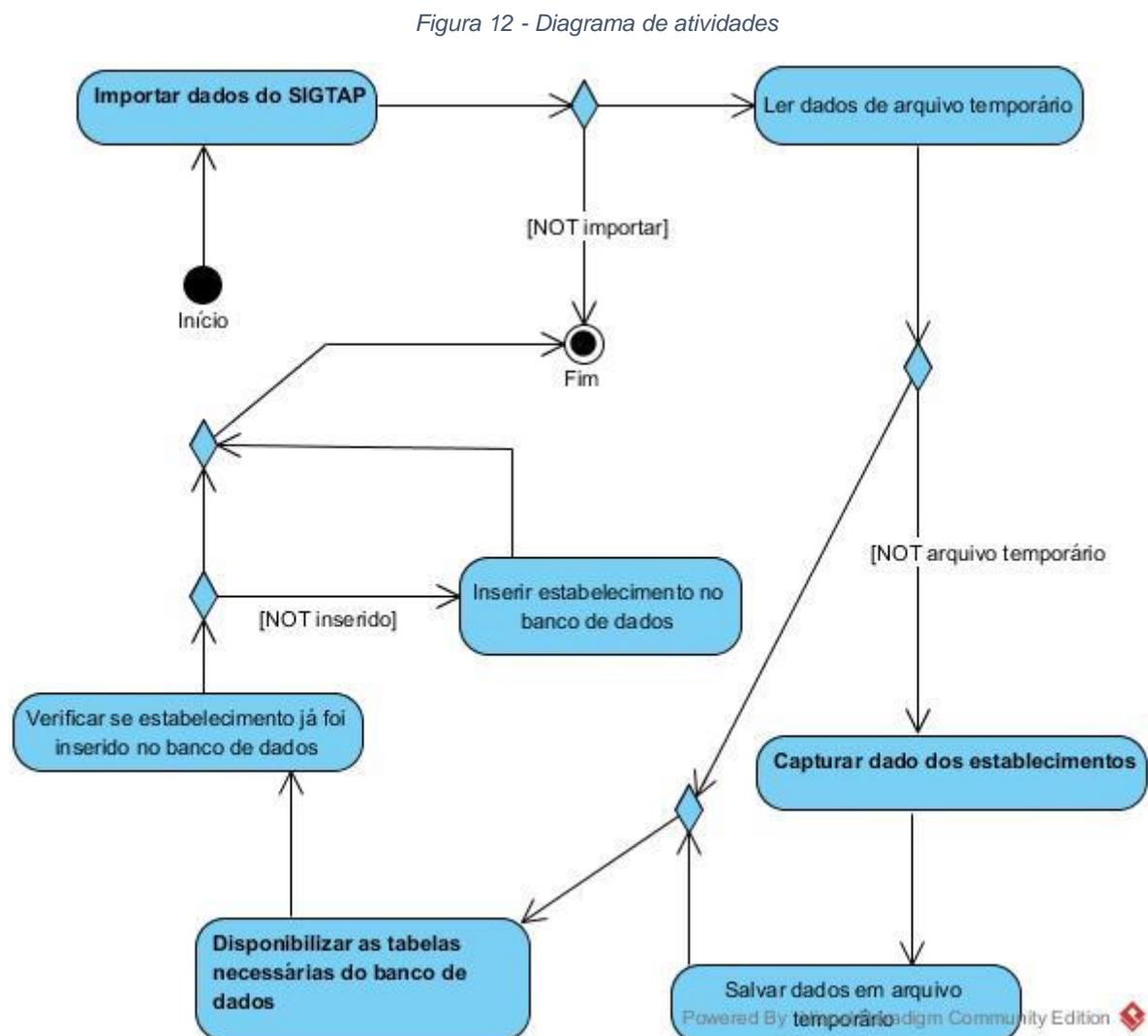
Figura 11 - Site com problemas



Fonte: Elaborado pelo autor

Após aumentar o valor do tempo de requisição para 200 milissegundos, a página mostrada na Figura 11 deixou de ser apresentada com muita frequência, porém, a quantidade de exceções lançadas em relação ao estouro do tempo de requisição aumentou significativamente, sendo assim, foi tratada a exceção fazendo com que se repetisse a requisição cujo tempo tenha estourado. Optou-se por aumentar o tempo de requisição para 350 milissegundos, com esse valor a quantidade de exceções geradas por estouro de tempo de requisição diminuiu.

A aplicação para Desktop desenvolvida para capturar os dados das páginas do site do CNES realiza vários procedimentos para gerar um banco de dados em formato *SQLite* 3. Para resumir todos os processos realizados pela ferramenta desenvolvida, foi criado um diagrama de atividades em UML, que pode ser visualizado na Figura 12, demonstrando o fluxo de execução da aplicação.



Fonte: Elaborado pelo autor

Para que fosse possível adquirir as coordenadas geográficas da localização dos estabelecimentos foram utilizadas duas fontes. A primeira e prioritária foi definida como sendo a *Open Street Maps* que através da ferramenta web *open-source* desenvolvida, *nominatim*, foi possível converter o endereço comercial de cada estabelecimento em sua coordenada geográfica equivalente. Isto foi possível através de seu serviço web que retorna um arquivo no formato XML, contendo várias

informações sobre o endereço enviado como parâmetro, sendo uma delas, as coordenadas geográficas. Não existe um limite de requisições que podem ser feitas a essa ferramenta gratuita, contudo a precisão e o sucesso na busca por um endereço são muito baixos. A segunda fonte utilizada como sendo a secundária foi a API de mapas do Google. Essa ferramenta possui uma grande precisão nos resultados obtidos. Ela funciona de forma semelhante a primeira fonte utilizada, porém há um limite de 2.500 requisições por dia, sendo assim, criou-se um método principal `getGeocod` na classe `ConversorHtmlParaJava` que depende de outros dois métodos internos para funcionar: `getGeocodOpenStreetMaps` e `getGeocodGoogleMaps`. Inicialmente o método `getGeocod` é chamado e recebe como parâmetro o endereço do estabelecimento, esse endereço é submetido ao método `getGeocodOpenStreetMaps` e caso não retorne os dados solicitados, o método `getGeocodGoogleMaps` é acionado devido ao seu alto grau de precisão. E assim a aplicação foi adquirindo as coordenadas geográficas variando a consulta nesses dois servidores e caso nenhum retorne a resposta solicitada o método `getGeocod` retorna um valor nulo identificando assim que não conseguiu localizar coordenadas com o endereço passado como parâmetro.

Esta ferramenta desenvolvida é capaz de capturar os estabelecimentos de todos os estados do país, para tal basta alterar a URL da página inicial que contém os municípios do estado desejado, pois o site do CNES está organizado da seguinte forma: Página contendo os Estados; página contendo os Municípios e página contendo os estabelecimentos do Município. Neste projeto a ferramenta de captura foi configurada para iniciar a coleta dos estabelecimentos do Estado de SC então a URL que está dentro do programa é a que leva a páginas de municípios de SC e ao final do processo temos como resultado um arquivo de banco de dados `SQLite` com estabelecimentos de SC. Pode-se simplesmente informar a URL que leva a um outro estado como o Paraná, por exemplo e ao final do processo tem-se um arquivo de banco de dados `SQLite` com estabelecimentos do PR. Sendo assim pode-se gerar um arquivo para cada Estado, caso queira. A Figura 13 exemplifica o que foi comentado.

Figura 13 - Configuração do Estado a ser capturado

```
26
27     /**
28      * Classe responsável pela extração dos dados das páginas html do portal do
29      * CNEsnet
30      *
31      * @author Jucemar
32      * @version 1.0
33      */
34  public class ConversorHtmlParaJava {
35
36      private static final String FUNCAO_PROFFISIONAIS_ESTABELECIMENTO = "Mod_Profissional.asp?VCo_Unidade=";
37      private static final String FUNCAO_SERVICO_CLASSIFICACAO = "Mod_Conj_Informacoes.asp?VCo_Unidade=";
38      private static final String FUNCAO_ATENDIMENTOS_PRESTADOS = "Mod_Bas_Atendimento.asp?VCo_Unidade=";
39      private static final String URL_PRINCIPAL = "http://cnes2.datasus.gov.br/";
40
41      private ArrayList<String> estabelecimentosFisicos;
42      int estabelecimentosValidos = 0;
43      int coordLocalizadasOSM = 0;
44      int coordLocalizadasGM = 0;
45
```

Fonte: Elaborado pelo autor

## 5.2 O APlicativo para Dispositivos ANDROID

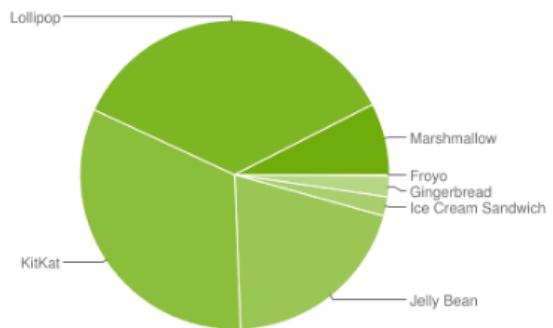
A Google monitora todas as versões existentes e ativas em todos os aparelhos celulares através da sua loja de aplicativos, a *Google Play Store*. A partir do momento que um aplicativo é instalado em um dispositivo ou que uma conta de email é vinculada ao smartphone, a *Google play Store* captura a informação sobre qual a versão do Android existente naquele aparelho e contabiliza e armazena todos esses dados. Dessa forma ela pode acompanhar o crescimento e a abrangência de seu sistema nos celulares ativos hoje no mercado e divulgar ao mundo a grande fatia de mercado adquirida através de sua página de divulgação a *Dashboard*<sup>20</sup> que possui dados atualizados semanalmente como pode ser visto na Figura 14:

---

<sup>20</sup> <https://developer.android.com/about/dashboards/index.html#Platform>

Figura 14 - Android Dashboards - Platform Versions

Version	Codename	API	Distribution
2.2	Froyo	8	0.1%
2.3.3 - 2.3.7	Gingerbread	10	2.2%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	2.0%
4.1.x	Jelly Bean	16	7.2%
4.2.x		17	10.0%
4.3		18	2.9%
4.4	KitKat	19	32.5%
5.0	Lollipop	21	16.2%
5.1		22	19.4%
6.0	Marshmallow	23	7.5%



Data collected during a 7-day period ending on May 2, 2016.

Any versions with less than 0.1% distribution are not shown.

Fonte: <https://developer.android.com/about/dashboards/index.html#Platform>

O desenvolvimento de um aplicativo para esta plataforma força o desenvolvedor a escolher qual a cobertura de versões do sistema da Google será focada em seu projeto. Dessa forma, quanto menor a versão focada maior será sua compatibilidade com os aparelhos ativos no mercado e maior será a sua cobertura entre os usuários. Baseando-se nesse contexto a versão escolhida para o desenvolvimento do aplicativo proposto nesse projeto foi a 4.0 (*Ice Cream Sandwich*) por ser totalmente compatível com versões superiores do Android e por atender a 97,7% de todos os *smartphones* ativos no mundo no momento da escrita deste documento.

A ideia para o desenvolvimento de um aplicativo para auxiliar na localização de estabelecimentos de saúde (clínicas médicas, clínicas especializadas, consultórios médicos especializados, clínicas que realizam exames de raios-x, exames de laboratórios) partiu da possibilidade do uso da funcionalidade de geolocalização dos atuais telefones celulares. Sendo assim, em uma situação de levantamento de requisitos o primeiro requisito funcional seria permitir ao usuário do aplicativo localizar os estabelecimentos de saúde nas proximidades. O primeiro pré-requisito não funcional poderia exigir que o dispositivo no qual o aplicativo será instalado e utilizado

deverá prover a localização geográfica (latitude e longitude) atual do usuário através do GPS e sem a necessidade de conexão com qualquer serviço de localização obtido via internet.

Para vislumbrar todas as possíveis funcionalidades do aplicativo foi desenvolvido um documento descritivo, que pode ser observado no Apêndice A. Tais requisitos foram eleitos baseando-se na ideia de facilitar a forma com que os usuários do site de divulgação dos estabelecimentos de saúde cadastrados no MS visualizam as informações. Foi pensando em uma visão sob a perspectiva de um cidadão sem conhecimentos técnicos ou avançados na área da saúde que resolveu-se reaproveitar as informações contidas nesse portal de divulgações e organizá-las de forma que pudessem auxiliar qualquer pessoa durante a sua rotina diária.

Após definidas todas as funcionalidades que iriam ser disponibilizadas pelo aplicativo no documento de requisitos criado, iniciou-se o processo de desenvolvimento por telas. Da seção 4.2.2 até a seção 4.2.7 serão apresentadas, de forma detalhada, as experiências no processo de desenvolvimento de cada uma delas. Para cada uma delas pensou-se inicialmente como ficaria a sua interface gráfica em relação ao posicionamento de botões, menus e textos e quais funcionalidades ela forneceria aos usuários. Na sequência a etapa de programação do código foi realizada. A ordem de desenvolvimento das telas segue a ordem apresentada nas seções a seguir. Esta ordem foi seguida pois cada tela desenvolvida é acessada respectivamente pela tela anterior.

O aplicativo foi desenvolvido focando-se na praticidade. Nele foram utilizados alguns dos componentes visuais mais atuais na data do desenvolvimento deste projeto. Tais componentes foram utilizados seguindo o guia de utilização do *Material Design* visto em GOOGLE DESIGN (2016), este guia aconselha como devem ser utilizados os elementos de interface gráfica dentro do aplicativo para que ele fique dentro dos novos padrões do Android.

### **5.2.1 O banco de dados**

O aplicativo conta com um banco de dados interno. Trata-se de um arquivo único que é gerado na etapa de captura dos dados dos estabelecimentos usando o gerenciador de banco de dados *SQLite* em sua versão 3. Este arquivo é a base para as consultas que são feitas no aplicativo que se utiliza das soluções nativas para

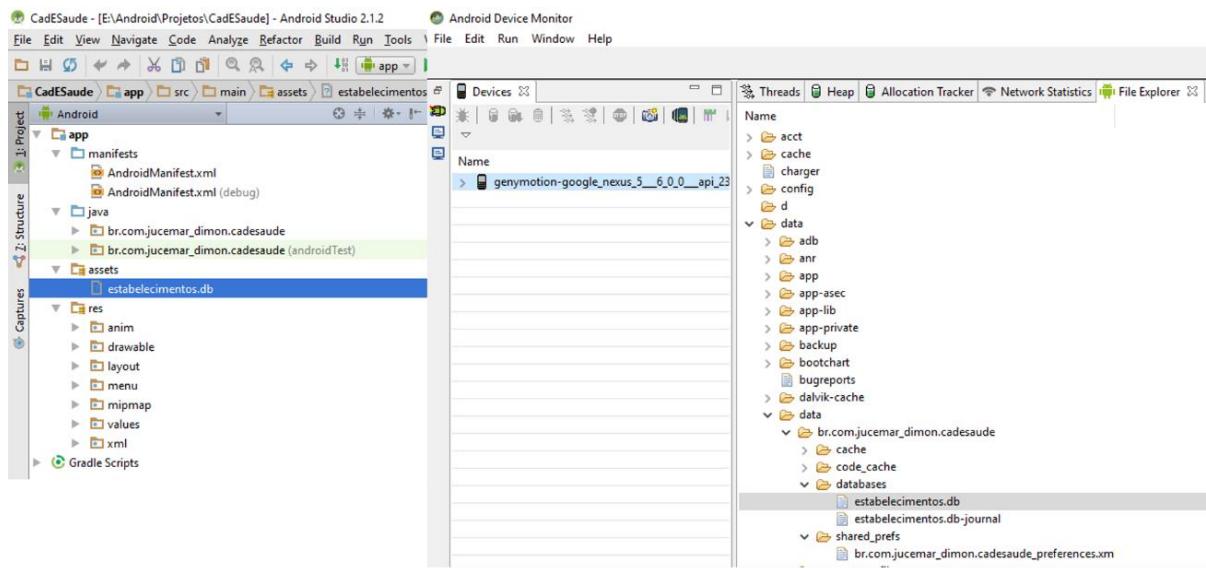
manipulação de bancos e dados *SQLite*<sup>21</sup> no SDK do Android. Esse arquivo de banco de dados é acessado pelas classes desenvolvidas para manipular seus dados. Na árvore de diretórios do projeto do aplicativo no software *Android Studio* existem pastas específicas para armazenamentos de arquivos de tipos específicos, segundo o guia do usuário apresentado em ANDROID STUDIO (2016), existe um diretório no projeto do aplicativo no qual pode-se armazenar um arquivo num formato qualquer que o mesmo não é alterado no processo de compilação do código fonte e criação do arquivo APK utilizado para instalação do aplicativo no dispositivo, o diretório *assets*. Este diretório mantém os arquivos intactos e possibilita a sua leitura através de *streams* de *bytes*. O arquivo contendo o banco de dados do aplicativo foi salvo neste diretório. Tal decisão foi tomada pois o aplicativo não criará um arquivo de banco de dados durante sua primeira execução, como geralmente é feito ao utilizar-se as classes para o gerenciamento de banco de dados nativos no android como *SQLiteOpenHelper*<sup>22</sup>. Neste contexto já existe um banco de dados criado e preenchido, sendo assim é necessário armazená-lo dentro do dispositivo para que se possa utilizá-lo conectando-se a ele e realizando consultas. Como afirmado, o conteúdo da pasta *assets* é armazenado dentro do arquivo no formato APK utilizado na instalação do aplicativo no dispositivo, sendo assim a técnica utilizada para transferir o banco de dados para o dispositivo foi o armazenamento e transferência para pasta final onde o aplicativo instalado fica armazenado dentro da estrutura de diretórios de arquivos da plataforma Android. A Figura 15 demonstra ao lado esquerdo a localização do arquivo de banco de dados no projeto no *Android Studio* e ao lado direito a sua localização no diretório do aplicativo instalado dentro do dispositivo.

---

<sup>21</sup> <https://developer.android.com/reference/android/database/sqlite/package-summary.html>

<sup>22</sup> <https://developer.android.com/reference/android/database/SQLiteDatabase.html>

Figura 15 - Estrutura de diretórios: projeto Android X Emulador



Fonte: Elaborado pelo autor

Após a autorização da instalação do aplicativo pelo usuário, os arquivos de instalação contidos no arquivo APK de instalação são transferidos para o dispositivo e durante a primeira execução do aplicativo, o processo de armazenamento do arquivo de banco de dados em seu local de destino na pasta `data/data/br.com.jucemar_dimon.cadesaude/databases` é completado deixando o mesmo disponível para conexões. Todo o gerenciamento da conexão com o banco de dados é realizado através de uma classe nomeada de `BancoDeDados` que herda todas as características da classe nativa da plataforma Android para manipulação de conexões com bancos de dados `SQLite`, a classe `SQLiteOpenHelper`<sup>23</sup>. Utilizando os métodos desta classe é possível criar, atualizar, abrir e fechar conexões com o banco. Para este aplicativo criou-se um método personalizado para permitir que a estratégia aqui desenvolvida fosse aplicável. A Figura 16 demonstra algumas linhas de código usadas na solução.

<sup>23</sup> <https://developer.android.com/reference/android/database/sqlite/SQLiteOpenHelper.html>

Figura 16 - Classe gestora de conexões com o banco de dados

```
--  
17 public class BancoDeDados extends SQLiteOpenHelper {  
18     public static String BD_CAMINHO = "/data/data/br.com.jucemar.dimon.cadesaude/databases/";  
19     public static String BD_NOME = "estabelecimentos.db";  
20     public static final int BD_VERSAO = 201607;  
21     private static BancoDeDados instancia;  
22     private Context contexto;  
23  
24     public static BancoDeDados getInstancia(Context contexto) {  
25         if (instancia == null) {  
26             instancia = new BancoDeDados(contexto);  
27             String myPath = BD_CAMINHO + BD_NOME;  
28             File banco = new File(myPath);  
29             if (banco.exists() && !banco.isDirectory()) {  
30                 banco.delete();  
31             }  
32             string caminho = "/data/data/br.com.jucemar.dimon.cadesaude/databases/";  
33             File pasta = new File(caminho);  
34             pasta.mkdirs();  
35             File bancoPronto = new File(pasta, BD_NOME);  
36             try {  
37                 bancoPronto.createNewFile();  
38                 Log.e("BancoDeDados", "arquivo de banco de dados não existe, ele será copiado");  
39                 instancia.copiarBancoDeDados();  
40             } catch (IOException e) {  
41                 Log.e("BancoDeDados", "Houve algum erro durante a criação ou cópia do banco de dados");  
42             }  
43         }  
44         return instancia;  
45     }  
46  
47     private BancoDeDados(Context contexto) {...}  
51  
52     @Override  
53     public void onCreate(SQLiteDatabase db) {...}  
56  
57     public void copiarBancoDeDados() {...}
```

Fonte: Elaborado pelo autor

Como visto na figura acima, a classe usa o padrão de projetos *Singleton* (FREEMAN et al., 2009, p. 125) na qual somente uma instância da classe *BancoDeDados* é disponibilizada por vez. A cada solicitação de uma instância da classe é feita uma verificação em relação à existência e acessibilidade do arquivo de banco de dados. No caso da não existência do mesmo, como no contexto da primeira execução do aplicativo logo após a instalação, uma cópia dele é feita para o diretório *databases* para que a conexão possa ser criada.

### 5.2.2 A tela inicial

A primeira tela visualizada pelo usuário do aplicativo é tela de apresentação. Ela é muito simples pois contém apenas o logotipo, a descrição e a versão do software, porém ela tem um papel fundamental pois além de apresentar o aplicativo atraindo a atenção do usuário, permite que um processamento seja feito em segundo plano para verificar se o aplicativo já foi utilizado alguma vez ou se esta é a sua primeira

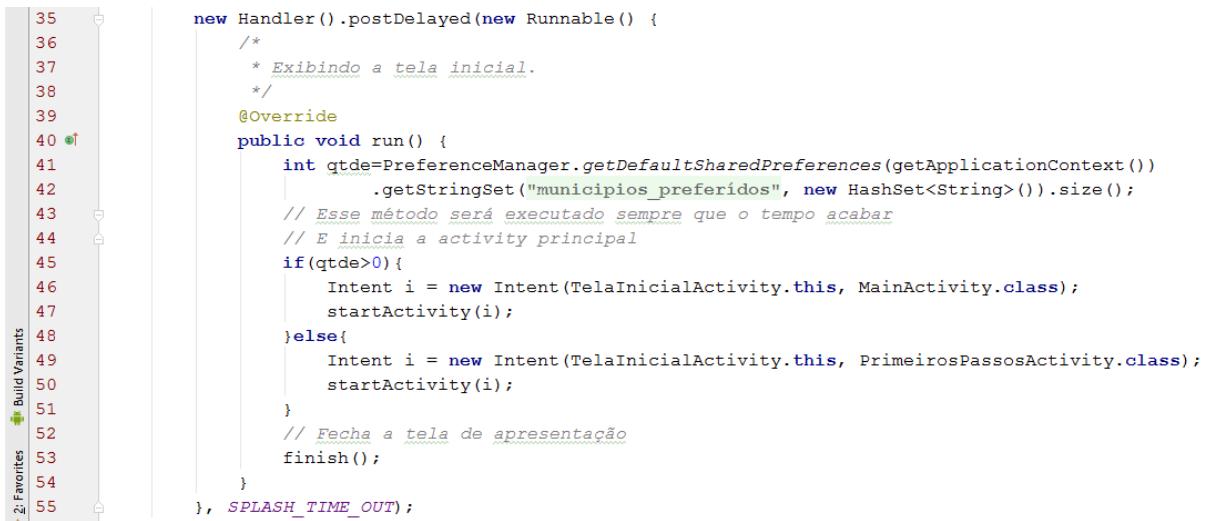
execução. Caso seja a sua primeira execução, o usuário é levado à tela de configuração inicial do aplicativo, caso contrário a tela principal é apresentada ao mesmo. A Figura 17 ilustra a tela e a Figura 18 contém um trecho do código que executa esta ação.

Figura 17 - A tela de apresentação do aplicativo



Fonte: Elaborado pelo autor

Figura 18 - Trecho de código fonte que inicia a tela de apresentação



```
35     new Handler().postDelayed(new Runnable() {
36         /*
37          * Exibindo a tela inicial.
38          */
39     @Override
40     public void run() {
41         int qtde=PreferenceManager.getDefaultSharedPreferences(getApplicationContext())
42             .getStringSet("municípios_preferidos", new HashSet<String>()).size();
43         // Esse método será executado sempre que o tempo acabar
44         // E inicia a activity principal
45         if(qtde>0){
46             Intent i = new Intent(TelaInicialActivity.this, MainActivity.class);
47             startActivity(i);
48         }else{
49             Intent i = new Intent(TelaInicialActivity.this, PrimeirosPassosActivity.class);
50             startActivity(i);
51         }
52         // Fecha a tela de apresentação
53         finish();
54     }
55 }, SPLASH_TIME_OUT);
```

Fonte: Elaborado pelo autor

Segundo a documentação oficial disponibilizada pelo site *Developer Android*<sup>24</sup>, um *Handler* é um mecanismo que permite enviar uma mensagem a *thread* principal do aplicativo, a *UI thread*. Somente a *thread* principal faz o processamento dos elementos da interface gráfica dos aplicativos na plataforma Android, logo, um *Handler* avisa esta *thread* de que ela precisa realizar certo processamento. É isto que está sendo feito na Figura 18. Um *Handler* executa as linhas de código dentro do método *run()*, após o tempo de 2000 milissegundos que foi definido pela constante *SPLASH\_TIME\_OUT*. A variável *qtde* armazena a quantidade de municípios que o usuário definiu em sua configuração. Caso ela seja menor ou igual a zero a tela de configurações é aberta com auxílio da classe *Intent*<sup>25</sup>.

### 5.2.3 A tela de favoritos

A tela de favoritos, vista na Figura 19, possui uma área na qual os estabelecimentos anteriormente definidos como favoritos pelo usuário aparecem listados para que o mesmo tenha seu acesso rápido.

<sup>24</sup> <https://developer.android.com/reference/android/os/Handler.html>

<sup>25</sup> <https://developer.android.com/reference/android/content/Intent.html>

Figura 19 - Tela de estabelecimentos favoritos



Fonte: Elaborado pelo autor

Essa área do aplicativo possui dois botões de ação flutuante que se destacam visualmente do restante da interface gráfica justamente para chamar a atenção do usuário. Segundo a seção sobre botões de ação flutuante em GOOGLE DESIGN (2016), um botão desta categoria é o componente visual que conterá as ações principais do aplicativo ou de uma tela em questão. Seguindo esta boa prática definida pela Google, tal categoria de botão foi adicionado a esta tela do aplicativo. Foram adicionados dois botões, cada um deles com uma ação específica: um para permitir que o usuário seja direcionado para a tela de busca por estabelecimentos utilizando suas coordenadas geográficas e outro para levar o usuário à tela que permitirá ao mesmo a busca por estabelecimentos por parâmetros de interesse.

Há ainda nesta tela, como visto na Figura 20, um menu de opções padrão localizado no canto superior direito, dentro de um componente da interface gráfica

denominado *ActionBar*<sup>26</sup>, o qual possibilita o armazenamento e gerenciamento de componentes como menus de ação e buscas entre outros. Neste menu foram incluídas algumas ações que possibilitam ao usuário: ir à área de configurações central do aplicativo; ir até a tela de ajuda e visualizar informações gerais sobre a aplicação.

Figura 20 - Menu de configurações



Fonte: Elaborado pelo autor

No desenvolvimento destas telas das Figuras 19 e 20, utilizou-se o componente de interface gráfica *RecyclerView*<sup>27</sup>. Este componente fornece uma lista que pode armazenar e exibir uma quantidade bastante grande de itens na tela da aplicação. Nesta tela, todos os estabelecimentos os quais o usuário visualizou e

<sup>26</sup> <https://developer.android.com/reference/android/app/ActionBar.html>

<sup>27</sup> <https://developer.android.com/reference/android/support/v7/widget/RecyclerView.html>

marcou como favorito são listados dentro da *RecyclerView*. Este por sua vez permite acesso direto à visualização dos dados do estabelecimento com um simples toque interativo na área onde fica localizado o título deste estabelecimento dentro de um item na lista da *RecyclerView*.

Os estabelecimentos podem ser marcados como preferidos pelo usuário durante a visualização de seus dados detalhados. O registro do estabelecimento marcado como favorito é salvo na memória interna do aplicativo com o auxílio da interface *SharedPreferences*<sup>28</sup> disponibilizada pelo SDK do Android. Esta API facilita a programação do processo de salvamento dos dados na memória do dispositivo. A Figura 21 demonstra como o processo de salvamento pode ser simplesmente feito através de poucas linhas de código. Com a execução do método descrito na figura, os dados contidos na estrutura de dados do tipo *Set* de *Strings*<sup>29</sup> são salvos em um arquivo no formato XML. A Figura 22 revela o diretório dentro da memória interna do dispositivo onde os dados são salvos caso se utilize a interface *SharedPreferences* para realizar tal ação. Para que fosse possível a visualização deste diretório fez-se necessário a utilização de uma ferramenta contida na IDE *Android Studio* onde é possível visualizar informações sobre os recursos do dispositivo que está sendo emulado num determinado momento, a ferramenta em questão chama-se *Android Device Monitor*<sup>30</sup>. Tal ferramenta foi utilizada nesse contexto para visualizar a árvore de diretórios gerada pela instalação do aplicativo no sistema operacional Android para comprovar o funcionamento da estratégia utilizada para armazenar a informação sobre estabelecimentos favoritos.

---

<sup>28</sup> <https://developer.android.com/reference/android/content/SharedPreferences.html>

<sup>29</sup> <https://developer.android.com/reference/java/lang/String.html>

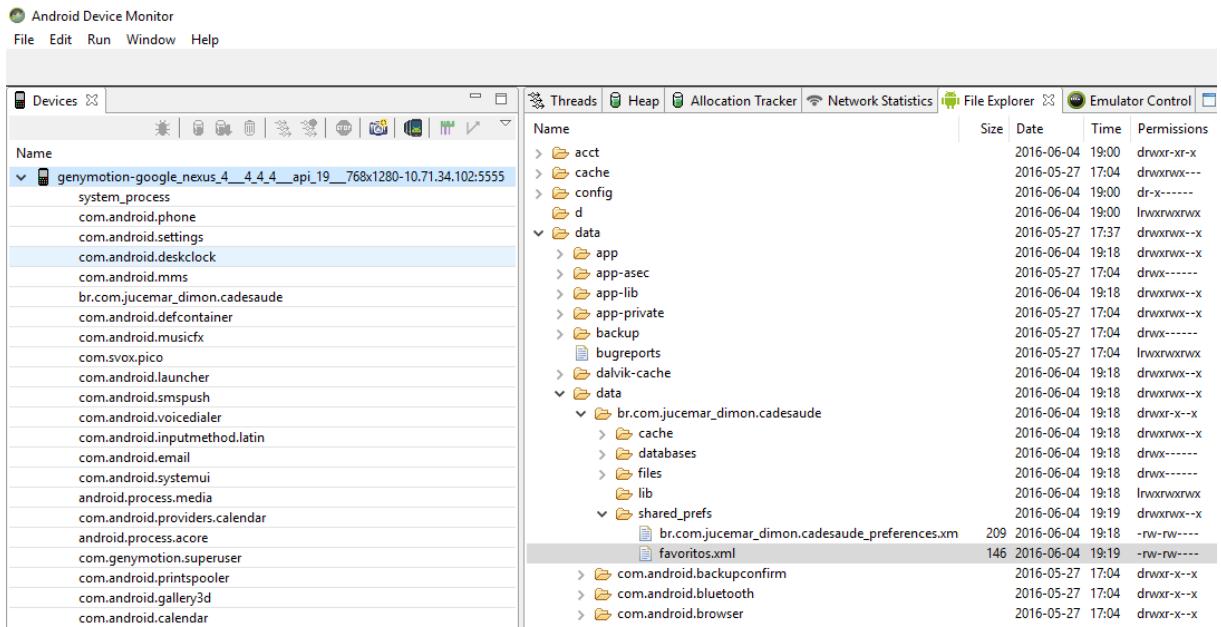
<sup>30</sup> <https://developer.android.com/studio/profile/monitor.html>

Figura 21 - Código fonte da definição do estabelecimento como favorito

```
1 package br.com.jucemar_dimon.cadesaude;
2
3 import android.content.Intent;
4 import android.content.SharedPreferences;
5 import java.util.ArrayList;
6 import java.util.HashSet;
7 import java.util.List;
8 import java.util.Set;
9
10 public class EstabelecimentoActivity extends AppCompatActivity {
11     private ArrayList<ServicoClassificacao> listaDeServicos;
12     private SharedPreferences.Editor editorConfig;
13     private Estabelecimento estabelecimento;
14
15     @Override
16     protected void onCreate(Bundle savedInstanceState) {
17         super.onCreate(savedInstanceState);
18         listaDefavoritos=new HashSet<String>();
19         estabelecimento = getIntent().getParcelableExtra("estabelecimento");
20     }
21
22     private void toggleFavorito() {
23         if(listaDefavoritos.contains(String.valueOf(estabelecimento.getNome())))) {
24             listaDefavoritos.remove(String.valueOf(estabelecimento.getNome()));
25         }else{
26             listaDefavoritos.add(String.valueOf(estabelecimento.getNome()));
27         }
28         editorConfig.putStringSet("listaDeFavoritos", listaDefavoritos);
29         editorConfig.apply();
30     }
31 }
```

Fonte: Elaborado pelo autor

Figura 22 - Diretório no dispositivo onde é salvo o estabelecimento favorito



Fonte: Elaborado pelo autor

Para armazenar os dados sobre os favoritos marcados utilizou-se um arquivo no formato XML, como visto na Figura 22. Optou-se por armazenar o código CNES de cada estabelecimento definido como favorito para que a sua posterior recuperação no banco de dados pudesse se tornar mais eficiente. A Figura 23 mostra a estrutura interna do arquivo que mapeia os códigos CNES dos estabelecimentos definidos como favoritos.

Figura 23 - Estrutura interna do arquivo de estabelecimentos favoritos

```

<?xml version='1.0' encoding='utf-8' standalone='yes'?>
<map>
  <set name="listaDeFavoritos">
    <string>19275</string>
    <string>6808190</string>
    <string>3327353</string>
  </set>
</map>

```

Fonte: Elaborado pelo autor

A classe principal da aplicação que contém a tela de favoritos é chamada de *MainActivity*. Ela é uma classe que herda da classe *AppCompatActivity*<sup>31</sup> que é a classe que pertence à biblioteca de suporte desenvolvida pela empresa, que permite que certas funcionalidades de versões mais recentes do Android funcionem em versões mais antigas. A classe *MainActivity* possui uma classe interna chamada de *CarregadorDeDados* que herda da classe *AsyncTask*<sup>32</sup>. A *AsyncTask* é uma classe pertencente a biblioteca nativa do SDK do Android que permite abstrair o processo de trabalhar com *multithreads* no desenvolvimento, esta classe permite configurar uma tarefa para que seja executada em um *thread* separada da *thread* principal, a UI Thread, do aplicativo e faz isso de forma assíncrona impedindo que a interface gráfica do aplicativo que é gerenciado pela *thread* principal gaste seu tempo de processamento com outras tarefas secundárias. A classe privada *CarregadorDeDados* executa suas tarefas em uma *thread* em segundo plano através do método *doInBackground*<sup>33</sup>, este, após o término da execução das linhas de códigos contidas, chama o método *onPostExecute*<sup>34</sup> o qual faz a comunicação entre as ações realizadas na *thread* de segundo plano com a *thread* de primeiro plano permitindo assim ambas interagirem. O processo de carregamento dos estabelecimentos cujo usuário definiu como favorito é feito através da instanciação da classe *CarregadorDeDados* e execução do método que inicia o processo a partir da classe *MainActivity*. Na sequência, é extraído da memória interna a lista de estabelecimentos favoritos através do objeto da classe *PreferenceManager*<sup>35</sup>. Em posse desta lista e de uma instância da classe *EstabelecimentoDAO*, que é a responsável por abstrair a comunicação com a tabela *estabelecimento* do banco de dados, executa-se o método *listaEstabelecimentosFavoritos* passando como parâmetro a lista, tem-se como resultado uma coleção de estabelecimentos, caso haja algum favorito definido anteriormente. Após o final desses processos o método *onPostExecute* é chamado e a *RecyclerView* tem seu adaptador preenchido com estes dados e logo ela está pronta para apresentar os estabelecimentos na tela. Na Figura 24 pode-se visualizar o código fonte mencionado.

---

<sup>31</sup> <https://developer.android.com/reference/android/support/v7/app/AppCompatActivity.html>

<sup>32</sup> <https://developer.android.com/reference/android/os/AsyncTask.html>

<sup>33</sup> [https://developer.android.com/reference/android/os/AsyncTask.html#doInBackground\(Params...\)](https://developer.android.com/reference/android/os/AsyncTask.html#doInBackground(Params...))

<sup>34</sup> [https://developer.android.com/reference/android/os/AsyncTask.html#onPostExecute\(Result\)](https://developer.android.com/reference/android/os/AsyncTask.html#onPostExecute(Result))

<sup>35</sup> <https://developer.android.com/reference/android/preference/PreferenceManager.html>

Figura 24 - método de execução em segundo plano

```
254     private class CarregadorDeDados extends AsyncTask<Void, Void, Void> {
255         private Context contexto;
256
257         public CarregadorDeDados(Context contexto) {
258             this.contexto = contexto;
259         }
260
261         @Override
262         protected Void doInBackground(Void... params) {
263             listaDeFavoritos = config.getStringSet("listaDeFavoritos", new HashSet<String>());
264             listaDeMunicipios = PreferenceManager.getDefaultSharedPreferences(contexto)
265                 .getStringSet(getString(R.string.pref_municipios_preferidos_key), new HashSet<String>());
266             estabDAO = EstabelecimentoDAO.getInstance(getApplicationContext());
267             estabelecimentos = estabDAO.listaEstabelecimentosFavoritos(listaDeFavoritos);
268             //Log.e("search", dadosDeBuscaNaTela.trim().length() > 0 ? "SearchView" + " 1" : "Nada" + " 1");
269             if (dadosDeBuscaNaTela.trim().length() > 0) {
270                 bkpEstabelecimentos=new ArrayList<>();
271                 bkpEstabelecimentos = estabelecimentos;
272                 ArrayList<Estabelecimento> estabelecimentosTemp = new ArrayList<~>();
273                 for (Estabelecimento e : estabelecimentos) {
274                     if (e.getNomeFantasia().toLowerCase()
275                         .contains(dadosDeBuscaNaTela.toLowerCase())
276                         || e.getRazaoSocial().toLowerCase()
277                         .contains(dadosDeBuscaNaTela)) {
278                         estabelecimentosTemp.add(e);
279                     }
280                 }
281                 estabelecimentos = estabelecimentosTemp;
282             }
283             dadosDeBuscaNaTela = "";
284             return null;
285         }
286     }
```

Fonte: Elaborado pelo autor

A tela de favoritos ainda conta com a funcionalidade de busca pelo nome de algum estabelecimento que está armazenado dentro na *RecyclerView*. A lupa localizada no canto superior direito da tela ao ser tocada expande abrindo um campo de busca de texto. A base dessa funcionalidade é fundamentada através do framework de busca oficial do Android, e através da *SearchView*<sup>36</sup> foi possível adicionar tal funcionalidade ao aplicativo. Um dos passos para configuração do framework de *search* é a definição no arquivo *Android Manifest*<sup>37</sup> de qual será a *Activity*<sup>38</sup> que será responsável por processar as buscas. Como as buscas serão feitas na lista de estabelecimentos favoritos, a configuração foi feita nas propriedades da *Activity MainActivity* no arquivo de configurações *Android Manifest*. A Figura 25 detalha como foi definida a funcionalidade de busca nesta tela do aplicativo.

<sup>36</sup> <https://developer.android.com/reference/android/widget/SearchView.html>

<sup>37</sup> <https://developer.android.com/guide/topics/manifest/manifest-intro.html>

<sup>38</sup> <https://developer.android.com/guide/components/activities.html>

Figura 25 - Trecho de código que define a SearchView

```
162     @Override
163     public boolean onCreateOptionsMenu(Menu menu) {
164         MenuInflater menuInflater = getMenuInflater();
165         menuInflater.inflate(R.menu.menu_main, menu);
166         MenuItem searchItem = menu.findItem(R.id.action_search);
167         SearchManager searchManager = (SearchManager) MainActivity.this
168             .getSystemService(Context.SEARCH_SERVICE);
169         SearchView searchView = null;
170         if (searchItem != null) {
171             searchView = (SearchView) searchItem.getActionView();
172         }
173         if (searchView != null) {
174             searchView.setSearchableInfo(searchManager.getSearchableInfo(MainActivity.this
175                 .getComponentName()));
176         }
177         MenuItemCompat.OnActionExpandListener expandListener = new MenuItemCompat
178             .OnActionExpandListener() {
179                 @Override
180                 public boolean onMenuItemActionCollapse(MenuItem item) {
181                     carregaDados();
182                     return true;
183                 }
184
185                 @Override
186                 public boolean onMenuItemActionExpand(MenuItem item) {
187                     // Fazer algo quando o menu é expandido
188                     return true;
189                 }
190             };
191             MenuItemCompat.setOnActionExpandListener(searchItem, expandListener);
192         return super.onCreateOptionsMenu(menu);
193     }
```

Fonte: Elaborado pelo autor

O método `onCreateOptionsMenu`<sup>39</sup> é um método padrão de uma *Activity* que é executado pela plataforma do Android seguindo o ciclo de vida de uma *Activity*. Neste método é definido como o menu e seus itens serão apresentados na interface gráfica do aplicativo. Como a funcionalidade de busca é ativada através do toque de um item de menu, a sua programação deve ser feita dentro deste método. A *SearchView* foi definida e foi adicionado um *listener*<sup>40</sup> para avisar sobre o momento em que o usuário toca no botão de busca. Com isto o campo de busca expande-se para que o usuário possa digitar a palavra a ser buscada e com o aviso do *listener* acione-se o método que realiza o procedimento de busca em segundo plano, como mencionado na Figura 24. Essa busca verifica se algum termo foi digitado no campo de busca da tela, caso afirmativo é feita uma verificação na lista de estabelecimentos adquiridos do banco de dados dos estabelecimentos que estão atualmente na lista de

<sup>39</sup> <https://developer.android.com/guide/topics/ui/menus.html>

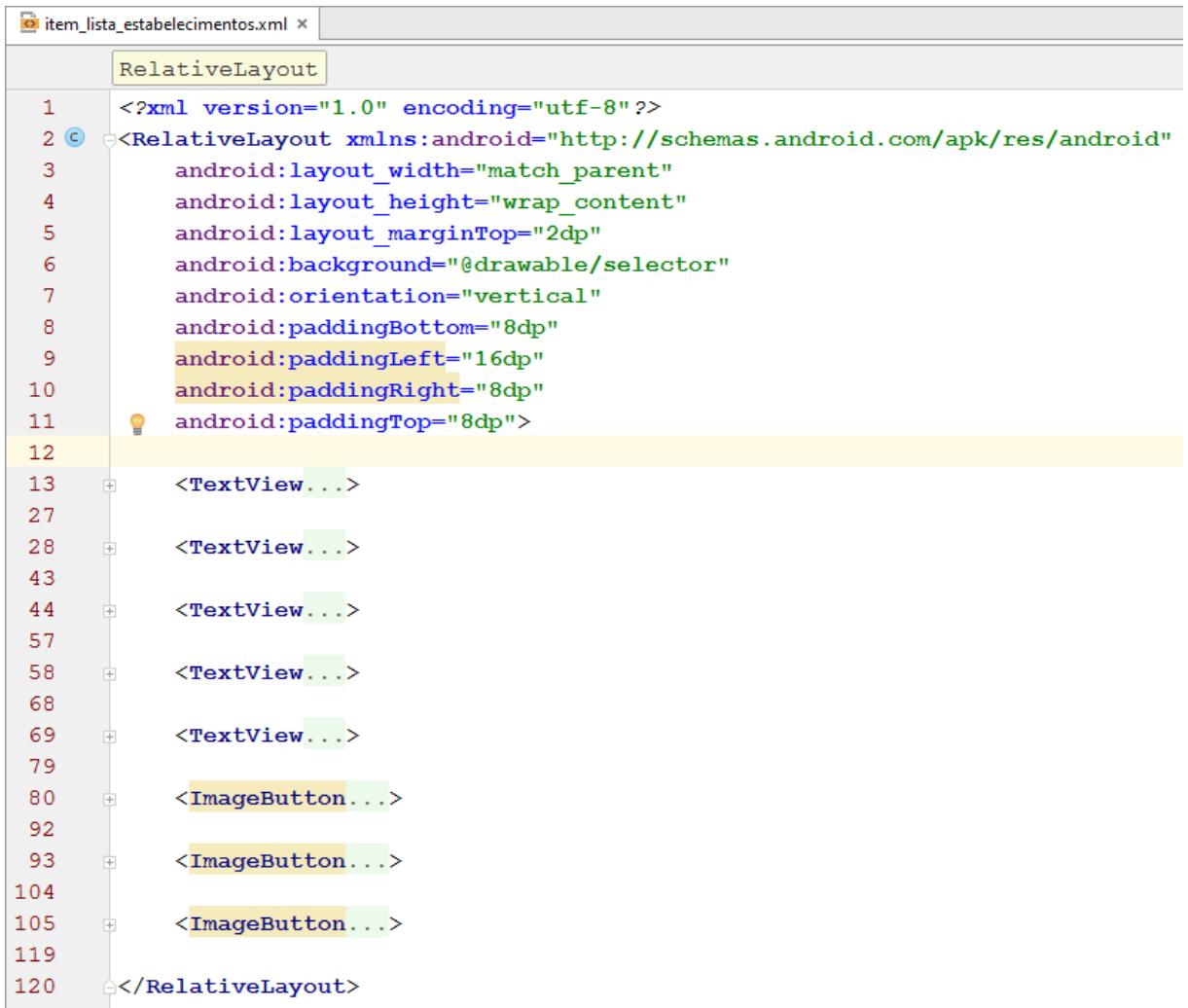
<sup>40</sup>

[https://developer.android.com/reference/android/view/MenuItem.html#setOnActionExpandListener\(android.view.MenuItem.OnActionExpandListener\)](https://developer.android.com/reference/android/view/MenuItem.html#setOnActionExpandListener(android.view.MenuItem.OnActionExpandListener))

favoritos atual e que possuem este termo em sua razão social ou nome fantasia, caso afirmativo esses estabelecimentos são postos em uma outra lista, a qual será apresentada na tela através de uma outra *RecyclerView*.

Para organizar as informações que serão exibidas sobre cada estabelecimento de saúde que será adicionado dentro da *RecyclerView* foi criado um adaptador de dados personalizado estendendo a classe *RecyclerView.Adapter*<sup>41</sup>. Estendendo esta classe, reescrevendo os seus métodos para que os elementos de interface gráfica que foram definidos através de um arquivo XML que pode ser visualizado na Figura 26, obteve-se como resultado os elementos de interface gráfica contidos da figura 27.

Figura 26 - Layout do modelo de apresentação dos dados dos estabelecimentos



```
<?xml version="1.0" encoding="utf-8" ?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="2dp"
    android:background="@drawable/selector"
    android:orientation="vertical"
    android:paddingBottom="8dp"
    android:paddingLeft="16dp"
    android:paddingRight="8dp"
    android:paddingTop="8dp">

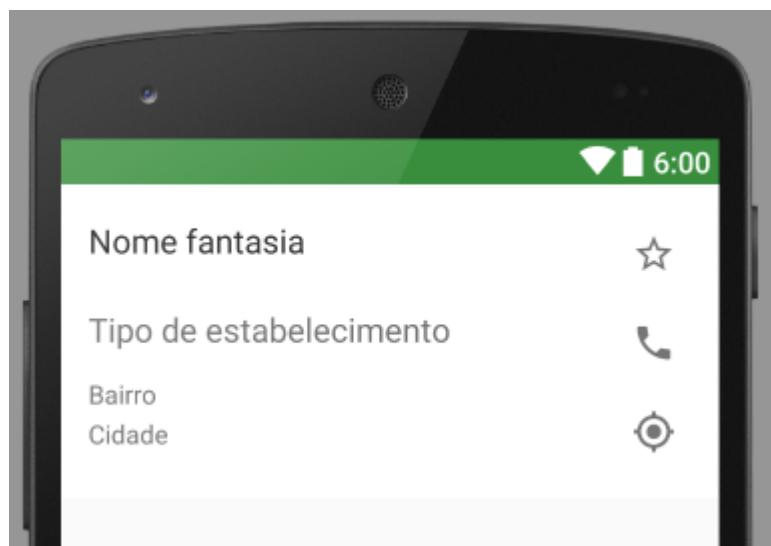
    <TextView...>
    <TextView...>
    <TextView...>
    <TextView...>
    <TextView...>
    <TextView...>
    <ImageButton...>
    <ImageButton...>
    <ImageButton...>

</RelativeLayout>
```

Fonte: Elaborado pelo autor

<sup>41</sup> <https://developer.android.com/reference/android/support/v7/widget/RecyclerView.Adapter.html>

Figura 27 - Resultado do layout na interface gráfica



Fonte: Elaborado pelo autor

Esse modelo de lista de estabelecimentos de saúde é utilizado em todos as telas que apresentam listas no aplicativo. Ele apresenta as informações mais relevantes sobre os dados de um estabelecimento de saúde. A direita existe uma coluna com três ícones e cada um deles dispara uma função. Estes ícones são *ImageButtons*<sup>42</sup> que possuem métodos que são chamados por um *listener* caso haja um *click* em um deles:

- O ícone de estrela ao ser pressionado, define o estabelecimento como um dos favoritos pelo usuário;
- O ícone de telefone realiza uma chamada telefônica para o número de telefone cadastrado no estabelecimento, utilizando o aplicativo padrão de ligações do *smartphone* onde ele está instalado;
- O ícone em formato de GPS, abre o aplicativo padrão de visualização de mapas do *smartphone* e faz a marcação do local no mapa da localização deste estabelecimento utilizando as coordenadas geográficas cadastradas no estabelecimento.

A decisão pela terceirização da execução de ações como fazer uma ligação telefônica e mostrar a posição geográfica num mapa foi tomada pensando na estrutura da plataforma do Android e em suas camadas de aplicações. Segundo dito na seção

---

<sup>42</sup> <https://developer.android.com/reference/android/widget/ImageButton.html>

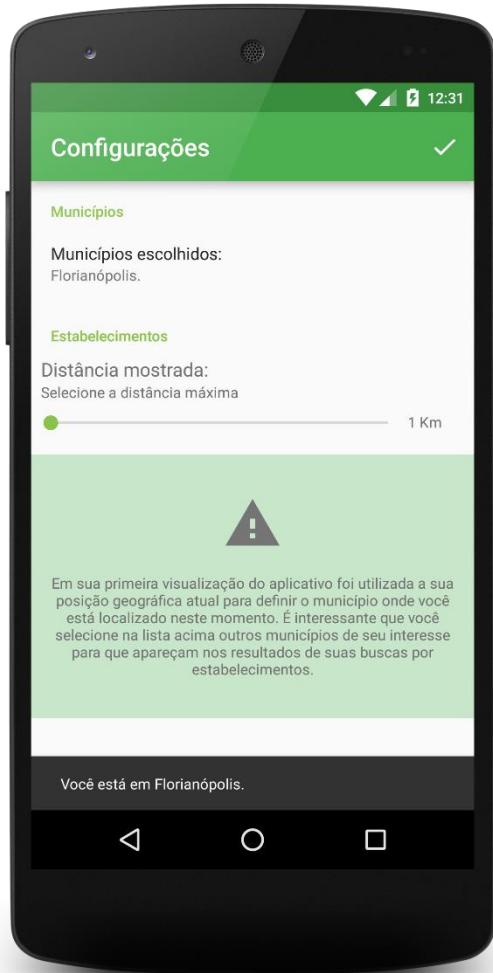
sobre o sistema de aplicativos em ANDROID (2016b), os aplicativos nativos ou desenvolvidos por terceiros que estão instalados no *smartphone* podem ter algumas de suas partes reaproveitados por outros aplicativos. Sendo assim, foram aproveitadas as soluções prontas e mais elaboradas e especializadas para realização de tais ações descritas acima.

#### **5.2.4 A tela de configurações**

A tela de configurações é utilizada para duas ações que influenciam no funcionamento central do aplicativo. É nela que o usuário deverá definir qual a distância máxima em quilômetros os seus resultados de busca por localização geográfica irão retornar. Nela também são definidos quais os municípios do Estado de Santa Catarina deverão aparecer nos resultados das buscas por parâmetros.

Todas as configurações do aplicativo podem ser alteradas acessando a tela principal do aplicativo onde encontra-se os estabelecimentos favoritos. Sua configuração é obrigatória para o funcionamento do aplicativo, tanto que durante a primeira execução do aplicativo no *smartphone* ela é a primeira tela que surge para o usuário após a tela de apresentação, como pode ser visto na Figura 28.

Figura 28 - Tela de configurações



Fonte: Elaborado pelo autor

Nesta primeira execução do aplicativo a configuração inicial é feita automaticamente para o usuário. São utilizadas as coordenadas geográficas do *smartphone* para descobrir qual o município atual ele se localiza e essa informação é definida nas configurações. A distância máxima para busca por estabelecimentos via GPS é definida por padrão como sendo 1 quilometro. O usuário pode alterar todas estas informações a qualquer momento, inclusive neste momento inicial.

Para adquirir as coordenadas geográficas da localização atual do usuário nesta primeira tela de configuração inicial, utilizou-se a classe *PrimeiraConfiguracaoActivity*. Esta classe herdou as características da classe *AppCompatActivity*<sup>43</sup>, que pertence ao pacote da API de compatibilidade com telefones mais antigos, e implementa as interfaces: *ConnectionCallbacks*<sup>44</sup>,

<sup>43</sup> <https://developer.android.com/reference/android/support/v7/app/AppCompatActivity.html>

<sup>44</sup>

`GoogleApiClient.OnConnectionFailedListener`<sup>45</sup>, `LocationListener`<sup>46</sup>,  
`ResultCallback<LocationSettingsResult>`<sup>47</sup>. Todas essas interfaces auxiliam na instânciação do objeto `GoogleApiClient`<sup>48</sup> cuja principal função é a verificação se o smartphone possui o Google Play Services instalado, se ele está atualizado para uma versão que seja compatível com a versão do SDK que está sendo utilizado no desenvolvimento do aplicativo, se ele possui o serviço de localização no dispositivo, através de um GPS físico ou via internet, se este serviço está ativo e se o aplicativo possui autorização para utilizá-lo. Para cada uma dessas verificações o objeto executa uma ação, um método de *call-back* caso o resultado seja negativo para alguma delas. Caso todas as etapas sejam aprovadas o Android inicializa o processo de captura das coordenadas geográficas e através do método de *call-back* `onLocationChanged(Location location)`<sup>49</sup> ele entrega ao desenvolvedor um objeto do tipo `Location`<sup>50</sup> que possui atributos como latitude e longitude. Este método é executado repetidas vezes durante um espaço de tempo que podemos definir. Como são necessárias apenas as coordenadas atuais, foi configurado para que ele seja executado apenas uma vez. Com as coordenadas do usuário, uma chamada a um método privado da classe principal que executa operações em segundo plano é feita. A classe `CarregadorDeDados` que herda funcionalidades de `AsyncTask`<sup>51</sup> realiza as seguintes ações para descobrir qual o município o usuário se encontra atualmente:

- Realiza uma chamada ao banco de dados retornando todos os estabelecimentos cadastrados;
- Calcula a distância em metros de cada um deles através do método `Location.distanceBetween(latitude,longitude,lat,lon,distancia)` que recebe como parâmetro dois pares de coordenadas, calcula a distância entre elas em métodos e armazena tal informação em uma atributo de cada objeto

---

<sup>45</sup> <https://developers.google.com/android/reference/com/google/android/gms/common/api/GoogleApiClient.ConnectionCallbacks>

<sup>46</sup> <https://developers.google.com/android/reference/com/google/android/gms/location/LocationListener>

<sup>47</sup> <https://developers.google.com/android/reference/com/google/android/gms/common/api/ResultCallback>

<sup>48</sup> <https://developers.google.com/android/reference/com/google/android/gms/common/api/GoogleApiClient>

<sup>49</sup> <https://developer.android.com/reference/android/location/LocationListener.html>

<sup>50</sup> <https://developer.android.com/reference/android/location/package-summary.html>

<sup>51</sup> <https://developer.android.com/guide/components/processes-and-threads.html>

Estabelecimento;

- A lista de objetos do tipo estabelecimento é organizada em ordem crescente através do Classe *Collections*, que provê estruturas de dados do tipo *ArrayLists*, baseando-se no atributo distância.
- O primeiro elemento desta lista organizada é o que possui o município no qual o usuário está nesse momento.
- O nome do município é definido na tela de configuração inicial.

Com apenas um município já definido o usuário pode salvar os dados clicando no botão no canto direito superior da tela. Assim ele poderá retornar a tela inicial.

A tela de configuração utiliza componentes prontos contidos no *Preference Framework*<sup>52</sup> disponibilizados pelo SDK do Android, na qual o componente da interface gráfica prontos para utilizar e montar telas padronizadas podem ser utilizadas para criar uma tela de configuração para o aplicativo. Ao utilizar esse framework percebe-se que ele é um pouco limitado em relação aos componentes pré-programados, porém é possível desenvolver componentes mais avançados como por exemplo uma *SeekBar*<sup>53</sup> que possa ser acoplada ao *framework* para que funcione de forma automatizada como os outros componentes. Toda a funcionalidade de salvamentos dos valores das configurações é transparente ao usuário do *framework*, por exemplo quando o usuário desliza a *seekbar* para definir a distância máxima que deseja visualizar estabelecimentos buscados via GPS, esse valor é salvo automaticamente pelo *framework* pois ele utiliza a mesma interface *SharedPreferences* que foi utilizada de forma separada na funcionalidade de adição de estabelecimentos de saúde ao arquivo XML de favoritos que é salvo na memória interna reservada ao aplicativo dentro da memória principal do *smartphone* como pode ser visto na Figura 22. Já na Figura 29 é apresentado *layout* que gera a interface gráfica da tela de configuração do aplicativo.

Figura 29 - Declarações para geração do Layout da tela de configurações

---

<sup>52</sup> <https://developer.android.com/guide/topics/ui/settings.html>

<sup>53</sup> <https://developer.android.com/reference/android/widget/SeekBar.html>

```

PreferenceScreen
1 <?xml version="1.0" encoding="utf-8"?>
2 <PreferenceScreen xmlns:android="http://schemas.android.com/apk/res/android"
3     android:title="Configurações">
4
5     <PreferenceCategory android:title="Municípios">
6         <MultiSelectListPreference
7             android:defaultValue="@array/municipios_preferidos_default"
8             android:dialogTitle="selecione os municípios"
9             android:entries="@array/nomes_municípios_view"
10            android:entryValues="@array/códigos_municípios"
11            android:key="municípios_preferidos"
12            android:selectable="true"
13            android:summary="Não há municípios selecionados"
14            android:title="Municípios escolhidos:" />
15     </PreferenceCategory>
16
17     <PreferenceCategory android:title="Estabelecimentos">
18         <br.com.jucemar_dimon.cadesaude.SeekBarPreference
19             android:defaultValue="1"
20             android:key="@string/pref_distancia_estabelecimentos_key" />
21     </PreferenceCategory>
22
23     <br.com.jucemar_dimon.cadesaude.InfoPreference
24         android:key="info"
25         android:selectable="false"
26         android:enabled="false" />
27
28 </PreferenceScreen>

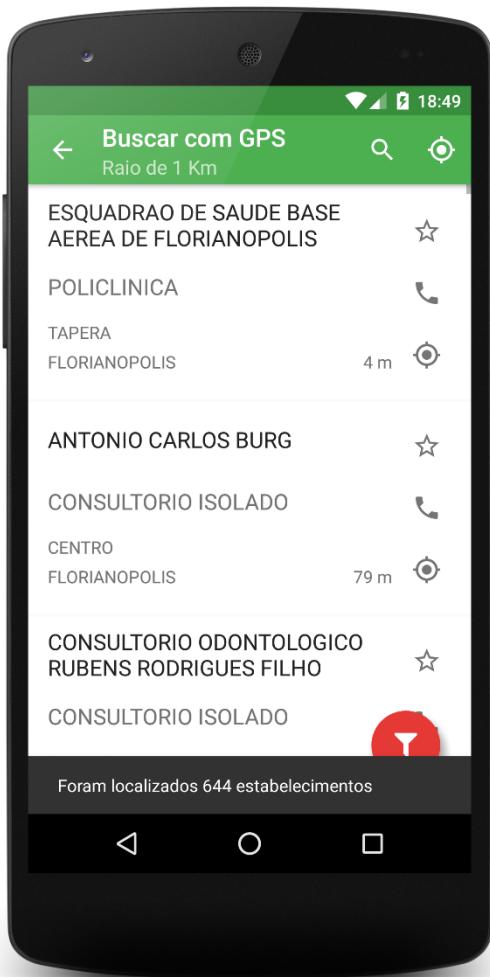
```

Fonte: Elaborado pelo autor

### 5.2.5 A tela de busca por GPS

A tela de busca por estabelecimentos através da posição geográfica do usuário é a funcionalidade principal do aplicativo. As buscas feitas nesta funcionalidade levam em consideração somente a distância máxima definida pelo usuário na tela de configurações do aplicativo para que o estabelecimento seja mostrado ou não nos resultados de busca. A Figura 30 faz a ilustração da interface gráfica da tela.

Figura 30 - Tela de busca por GPS



Fonte: Elaborado pelo autor

A tela mostra em seu cabeçalho qual o raio de distância predefinido nas configurações do aplicativo. Possui dois botões de ações na barra superior de menu: o filtro por nome e a nova busca por GPS para que o usuário possa atualizar sua pesquisa caso tenha mudado de posição geográfica. Assim como nas outras telas do aplicativo o componente visual utilizado para apresentar o resultado da busca pelos estabelecimentos é uma *RecyclerView*. Dentro dela são adicionados os estabelecimentos e seus dados básicos, como nome fantasia, bairro e cidade são apresentados juntamente com os botões de ações já explicado em seções anteriores. Nesta tela existe um botão de ação flutuante que se expande como um menu revelando outras diversas opções. Este componente visual é denominado de *FloatActionMenu*<sup>54</sup> e foi utilizado para auxiliar na criação de um filtro dos resultados encontrados. O *FloatActionMenu* é uma implementação personalizada que agrupa,

<sup>54</sup> <https://github.com/Clans/FloatingActionButton#floating-action-menu>

cria animações e interatividade entre os *FloatActionButton* que estão disponíveis como classes nativas do SDK do Android. O *FloatActionMenu* está disponível na biblioteca *FloatActionMenu* desenvolvida por Dmytro Tarianyk<sup>55</sup> a qual foi utilizada neste projeto por permitir de uma forma muito transparente e prática a criação de menus flutuantes. As opções para filtrar os estabelecimentos são montadas de acordo com as características dos estabelecimentos retornados da busca. A Figura 31 demonstra todos os tipos de filtros possíveis.

Figura 31 - Filtros da tela de busca por GPS



Fonte: Elaborado pelo autor

Cada um dos tipos de filtros possui seus valores a serem filtrados baseando-se nas características dos estabelecimentos que resultaram na busca. Caso o ícone do tipo de filtro fique desabilitado subentende-se que só exista um valor de opção

---

<sup>55</sup> <https://github.com/Clans>

possível para ele, logo não existe sentido utilizá-lo.

O processo de busca por localização geográfica utiliza também a *GoogleApiClient*. Inicialmente é feita a verificação no dispositivo se existe a aplicação Google API Service instalada, isto é feito através do comando *isGooglePlayServicesAvailable*<sup>56</sup> que retorna um código que representa o sucesso ou fracasso, caso seja fracasso uma janela é apresentada ao usuário solicitando que ele instale tal serviço. A segunda etapa é construir um objeto com auxílio da classe *GoogleApiClient* e suas interfaces e métodos de *callback*:

- *onConnected*: faz a conexão do *GoogleApiClient* com o serviço de localização do dispositivo.
- *onRequestPermissionsResult*: Caso o usuário não tenha permissão para utilizar o serviço de localização no dispositivo, esse método abre uma janela solicitando que o usuário autorize o aplicativo a usar o serviço.
- *adquirirLocalizacao*: Esse método é chamado caso todas as etapas anteriores tenham sido aceitas, ele inscreve o aplicativo ao serviço de *Location* do Android para ser informado sobre as coordenadas a cada período que foi definido no objeto instanciado através da *GoogleApiClient*.
- *onLocationChanged*: quando este *callback* método é chamado ele já traz um objeto do tipo *Location* que contém a latitude e a longitude do atual smartphone.

Após adquiridas as coordenadas geográficas atuais do dispositivo, é utilizado uma classe que executa uma tarefa em segundo plano chamada através da classe *CarregadorDeDadosBGPSA* que herda as propriedades da classe *AsyncTask* permitindo executar a tarefa em uma *thread* separada da *UI thread*. Neste método é adquirida uma instância da classe *EstabelecimentosDAO* que utiliza os padrões de projeto DAO e Singleton para facilitar o acesso ao banco de dados da aplicação. Tendo esta instância única, o método *buscaEstabelecimentosPorGPS* que recebe com parâmetros as variáveis *localizacao* e *distanciaMaxima* é utilizado para retornar uma coleção do tipo *ArrayList* de objetos do tipo *Estabelecimento*. Esta coleção de objetos é gerada utilizando a seguinte sequência de passos:

---

<sup>56</sup>

<https://developers.google.com/android/reference/com/google/android/gms/common/GoogleApiAvailability>

- Uma consulta ao banco de dados que retorna todos os estabelecimentos que possuem valores válidos nos campos Latitude e Longitude é realizada;
- Armazena-se estes dados em um *ArrayList* de objetos do tipo *Estabelecimento*, porém temporário;
- O método estático da classe *Location* da API do Android é utilizado para calcular a distância entre as coordenadas geográficas do estabelecimento e as coordenadas geográficas do dispositivo.
  - *Location.distanceBetween(localizacaoAtual.getLatitude(),localizacaoAtual.getLongitude(),latEstab,lonEstab,distancia);*
- O valor dessa distância é armazenado no atributo distância de cada objeto da classe estabelecimento contido na lista temporária.
- Este processo é feito para todos os estabelecimentos do banco e logo após eles são organizados em ordem de distância do menor para o maior.
- Na sequência é feita uma verificação de quais estabelecimentos possuem sua distância dentro da distância limite definida nas configurações do aplicativo e as que ficam dentro dos limites são armazenadas em uma terceira lista temporária que será retornada pelo método.

Com a lista de estabelecimentos configurada, a sua adição dentro da *RecyclerView* é realizada através de um adaptador. Todo esse processo é feito dentro do *call-back* método *onPostExecute* que é reescrito no método para que se possa criar uma comunicação com a *UI Thread* da interface gráfica e adicionar os estabelecimentos dentro da *RecyclerView*.

Quanto aos filtros possíveis de serem utilizados sobre os resultados, eles são aplicados filtros sobre filtros. Aplicando um filtro, o resultado deste na lista de estabelecimentos torna-se a base para a aplicação do próximo, e assim sucessivamente, até esgotarem-se as possibilidades de opções de filtros como pode ser visualizado na Figura 32.

*Figura 32 - Aplicação de filtro sobre filtro*



Fonte: Elaborado pelo autor

O processo de filtragem dos estabelecimentos é feito dentro do método *doInBackground* da classe *CarregadorDeDadosBGPSA*. Os estados de todos os filtros são salvos em listas do tipo *ArrayList* e todas são salvas no contexto do aplicativo através do método *onSaveInstanceState*<sup>57</sup>, para caso alguma outra tela do próprio aplicativo ou de outra aplicação do Android passe a ocupar o topo da tela do dispositivo. Todo o estado dos filtros aplicados são recuperados quando a tela de busca com GPS volta a ser visualizada através da instância do objeto da classe *Bundle* que pode ser acessada através do método de call-back *onSaveInstanceState* que recebe como parâmetro um objeto deste tipo.

### 5.2.6 A tela de busca parametrizada

<sup>57</sup>

[https://developer.android.com/reference/android/app/Activity.html#onSaveInstanceState\(android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onSaveInstanceState(android.os.Bundle))

A tela de busca parametrizada é acessada através da tela principal do aplicativo tocando no botão de ação flutuante contendo o ícone de lupa. Esta funcionalidade do aplicativo leva em consideração no processo de busca por estabelecimentos os municípios predefinidos pelo usuário na tela de configuração do aplicativo. Tal decisão foi tomada pensando na performance das buscas no banco de dados pois utilizando-se do município como parâmetro de busca retornou-se uma quantidade menor de resultados nas consultas. Isto tornou a manipulação dos dados na memória mais prática. O objetivo de desenvolver esta funcionalidade foi pensado para permitir ao usuário uma busca de forma rápida por algum parâmetro específico que não possa ser buscado na função de busca por GPS. Um contexto que exemplifica a necessidade de tal funcionalidade seria: um usuário que consultou em uma data passada com um profissional médico e alguns meses depois essa paciente descobre que este profissional não atende mais na clínica na qual ocorreu o atendimento anterior, porém este paciente prefere por insistência consultar com este mesmo profissional por ter sentido satisfação em seu trabalho. Utilizando a função de busca pelo nome do profissional nesta funcionalidade do aplicativo é possível descobrir todos os estabelecimentos que este profissional está trabalhando no momento. Este é apenas um exemplo de contexto de como essa funcionalidade do aplicativo pode ser útil, uma outra muito interessante seria a busca de uma clínica por parte do nome fantasia ou razão social do estabelecimento. A Figura 33 ilustra a tela de busca por parâmetros.

*Figura 33 - Tela de busca por parâmetros*



Fonte: Elaborada pelo autor

Esta é a tela qual o usuário fará sua busca por estabelecimentos utilizando parâmetros que serão selecionáveis através da janela de diálogo que será apresentada no momento em que o botão de ação flutuante com ícone de lupa for tocado. A partir de então uma série de configurações de busca pode ser feita. Os tipos de buscas possíveis das quais uma delas deve ser selecionada inicialmente na janela de diálogo são:

- Tipos de convênios: São as possíveis formas de pagamentos pelo atendimento prestado pelo estabelecimento, esses valores são predefinidos e podem ser:
  - Particular;
  - Plano / Seguro próprio;
  - Plano / Seguro terceiro;
  - Plano de saúde privado;
  - Plano de saúde público;

- SUS.
- Formas de atendimento: São as formas com as quais os estabelecimentos trabalham para atender aos clientes segundo sua estrutura física e estrutura e profissional:
  - Ambulatorial (Atendimentos com pouca urgência);
  - Internação (Atendimentos com pouca ou muita urgência que necessitem de acompanhamento monitorado);
  - SADT (Serviço de exames para diagnósticos: Exames laboratoriais, Raios-X, eletrocardiogramas, entre outros);
  - Urgência (Atendimentos que envolvem risco de vida).
- Nome do profissional: Modalidade de busca pelo nome do profissional. O termo buscado deverá ser digitado pelo usuário;
- Especialidades: Neste tipo de busca será retornada à janela de diálogo uma lista completa com todos os tipos de especialidades médicas cadastradas nos estabelecimentos de saúde localizados nos municípios definidos pelo usuário para realizar a busca;
- Nome do estabelecimento: Modalidade de busca na qual o usuário deverá digitar um termo a ser buscado;
- Serviços oferecidos: Modalidade de busca onde o usuário deverá selecionar um tipo de serviço especializado oferecido pelos estabelecimentos dos municípios listados para serem inclusos nos resultados de busca na tela de configurações do aplicativo. Os tipos de serviços possíveis podem ser vistos no Anexo B.

Após realizada a configuração da busca o usuário iniciará a busca e os resultados serão listados na tela. A Figura 34 demonstra um exemplo.

*Figura 34 - Configuração da busca parametrizada*



Fonte: Elaborada pelo autor

Esta tela é processada pela classe *BuscaGeralActivity* que herda da classe nativa do Android *AppCompatActivity* e implementa uma interface personalizada que foi desenvolvida para criar um canal de comunicação entre esta classe e a janela de diálogo que servirá para configurar os parâmetros de busca. A interface criada foi nomeada de *InterfaceBuscaGeralPorString* e a mesma possui o método *informarTermoParaConsulta* que recebe como parâmetro um uma constante inteira chamada de *tipoBusca* que conterá o tipo de busca a ser feita e uma variável do tipo *String* chamada de termo que armazenará qual o valor a ser buscado no banco de dados. Esse método *call-back* é chamada pelo Android no momento que a janela de configurações das pesquisas fecha.

As técnicas utilizadas para poder produzir ao usuário as configurações de todas essas funcionalidades são baseadas na classe *BuscaGeralDialog*. Esta classe herda as características da classe nativa do Android, a *DialogFragment* e implementa uma interface que permite a interação e vinculação entre os dois campos de lista

suspensa do tipo *Spinner*<sup>58</sup> nos quais os tipos possíveis de buscas são apresentados, a *AdapterView.OnItemSelectedListener*<sup>59</sup>. Esta interface permitiu que os dois elementos de interface gráfica do tipo *Spinner*: *comboBoxTipoBusca* e *comboBoxBusca* pudessem interagir de forma que dependendo do valor que o usuário selecionou em um influenciasse nas opções carregadas no outro. Todas estas opções carregadas nestes elementos, exceto as opções de tipos de busca que estão armazenadas em um simples *Array* de *Strings*, são advindas do banco de dados. Dessa forma para que o carregamento desses dados não interferissem nos processos que a UI *Thread* está executando, criou-se uma classe privada nomeada e *CarregadorDeDados*. Esta classe herda as propriedades da classe nativa do Android, a *AsyncTask*, que executa os procedimentos definidos em seus métodos em uma *thread* em segundo plano e de forma assíncrona. A interface *AdapterView.OnItemSelectedListener* foi vinculada aos elementos objetos *comboBoxTipoBusca* e *comboBoxBusca*. Dessa forma quando o usuário interage e define um tipo de busca em *comboBoxTipoBusca* o Android se utiliza da interface que foi implementada para chamar o método *call-back onItemSelected*. Neste método a seleção feita pelo usuário no *comboBoxTipoBusca* é capturada e então uma instância da classe *CarregadorDeDados* é criada para executar os métodos que farão as consultas no banco de dados. A Figura 35 ilustra esta execução.

Figura 35 - Trecho de código que representa as consultas ao banco

---

<sup>58</sup> <https://developer.android.com/guide/topics/ui/controls/spinner.html>

<sup>59</sup> <https://developer.android.com/reference/android/widget/AdapterView.OnItemSelectedListener.html>

```

226
227     @Override
228     protected Void doInBackground(Void... params) {
229         listaDeMunicipios = PreferenceManager.getDefaultSharedPreferences(getActivity())
230             .getStringSet("municipios_preferidos", new HashSet<String>());
231         switch (tipoDeBusca) {
232             case SPINNER_CONVENIOS_ACEITOS:
233                 AtendimentoDAO atendimentoDAOConv = AtendimentoDAO
234                     .getInstancia(contexto.getApplicationContext());
235                 dadosSpinner02 = atendimentoDAOConv.listarTiposConvenios(listaDeMunicipios);
236                 break;
237             case SPINNER_ESPECIALIDADES:
238                 ProfissionaisDAO profissionaisDAO = ProfissionaisDAO
239                     .getInstancia(contexto.getApplicationContext());
240                 dadosSpinner02 = profissionaisDAO.listarTiposProfissionais(listaDeMunicipios);
241                 break;
242             case SPINNER_FORMAS_DE_ATENDIMENTO:
243                 AtendimentoDAO atendimentoDAOAtend = AtendimentoDAO
244                     .getInstancia(contexto.getApplicationContext());
245                 dadosSpinner02 = atendimentoDAOAtend.listarTiposAtendimento(listaDeMunicipios);
246                 break;
247             case SPINNER_SERVICOS_OFERECIDOS:
248                 ServicoClassificacaoDAO servicoClassificacaoDAO = ServicoClassificacaoDAO
249                     .getInstancia(contexto.getApplicationContext());
250                 dadosSpinner02 = servicoClassificacaoDAO.listarTiposServicos(listaDeMunicipios);
251                 break;
252             }
253         }
254     }

```

Fonte: Elaborado pelo autor

Na figura acima percebe-se que a variável *listaDeMunicipios* armazena os municípios recuperados que foram definidos pelo usuário nas configurações do aplicativo. Na sequência verifica-se com uma estrutura de seleção o tipo de busca definido pelo usuário e dependendo desse valor é adquirida uma instância de um tipo específico de classe que manipula um tipo específico de dados no banco de dados do aplicativo. Após o resultado do método que traz as opções do tipo específico de consulta eles são carregados na interface gráfica através do método *onPostExecute* que é o responsável por solicitar a UI *Thread* que execute os comando contidos nele para que o *comboBoxBusca* possa ser preenchido com as opções carregadas do banco de dados.

Definidos os parâmetros de busca, o usuário toca no botão confirmar então a janela de dialogo fecha. Nesse momento, o método *informarTermoParaConsulta* da interface *InterfaceBuscaGeralPorString* é chamado e o foco passa a ser a tela principal de busca pois a janela de diálogo será fechada.

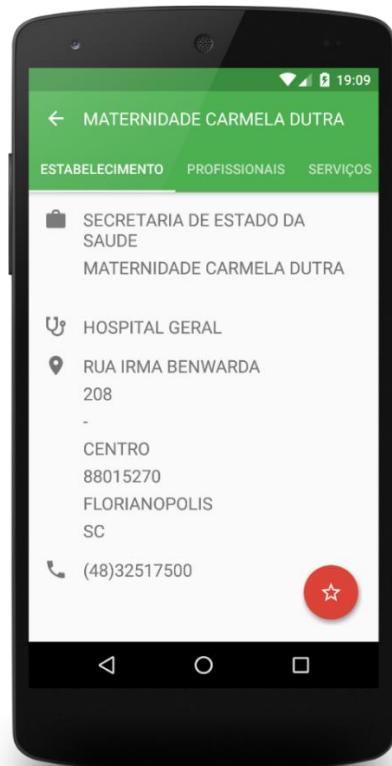
A partir desse momento já existe a informação sobre o tipo de busca e o valor a ser buscado nos estabelecimentos. Neste momento que o método *call-back* da interface é chamado, usa-se uma classe para executar em segundo plano a busca por esses estabelecimentos que possuem esse parâmetro em alguma de suas características. Após a busca feita, um *ArrayList* de objetos do tipo *Estabelecimento* é

retornado e repassado ao adaptador da *RecyclerView* para que esses estabelecimentos sejam mostrados na tela. Estes estabelecimentos mostrados dentro da *RecyclerView* possuem todas as funcionalidades das listas das outras telas como: funcionalidades de adição do estabelecimento a lista de favoritos, funcionalidade de visualização da posição geográfica em um aplicativo de mapa terceiro e a funcionalidade realizar ligações para o estabelecimento.

### 5.2.7 A tela de visualização do estabelecimento

A tela de visualização dos dados dos estabelecimentos é acionada a partir do toque em algum estabelecimento contido na *RecyclerView*. Quando um item dessa estrutura é tocado uma nova tela é aberta contendo todos os dados contidos no banco de dados sobre o estabelecimento em questão: Nome fantasia; Razão social; Endereço; Telefone; Profissionais que atuam no estabelecimento; Serviços; Categorias de atendimento e Convênios. Todas estas informações são apresentadas na tela numa estrutura de guias deslizáveis. A Figura 36 ilustra esta tela.

Figura 36 - Tela de detalhes do estabelecimento



Fonte: Elaborado pelo autor

A guia Estabelecimentos que mostra todos os dados principais referentes ao

estabelecimento ainda possui algumas funcionalidades para facilitar o uso pelo usuário. Ele também pode ter acesso à função de ligação direta, clicando no ícone do telefone, visualização da localização do estabelecimento em um mapa ao clicar no ícone de marcação de mapa e também pode adicionar o estabelecimento a sua lista de favoritos clicando no botão de ação flutuante com ícone de estrela.

As técnicas que permitem a visualização desta tela aos usuários foram desenvolvidas utilizando com base a classe *EstabelecimentoActivity*. Esta classe personalizada herda as características da classe nativa do Android chamada *AppCompatActivity*. A estrutura de guias foi desenvolvida baseando-se na criação de uma classe personalizada nomeada de *VisualizadorDePaginasAdapter* que herda as características da classe *FragmentStatePagerAdapter*<sup>60</sup>. Essa classe permite de forma simples a inserção de *Fragments*<sup>61</sup>, que são na verdade uma forma simplificada de telas as quais podem ser agrupadas e apresentadas na tela do dispositivo de várias formas diferentes. Para permitir essa estrutura de guias foram criadas várias classes que herdam as características desses *Fragments*:

- *AtendimentoFragment*;
- *EstabelecimentoFragment*;
- *ProfissionaisFragment*;
- *ServicosFragment*.

Cada um desses fragmentos é adicionado ao *VisualizadorDePaginasAdapter* e através da guia nomeada com o título das informações contidas nela, o usuário navega através delas simplesmente tocando no título da guia ou arrastando as telas do aplicativo, pois a classe *FragmentStatePagerAdapter* providencia todos mecanismos necessários para tal. A Figura 37 mostra o arquivo de layout em formato XML que auxilia na criação a tela em questão.

Figura 37 - Layout da visualização em abas da interface gráfica

---

<sup>60</sup> <https://developer.android.com/reference/android/support/v4/app/FragmentStatePagerAdapter.html>

<sup>61</sup> <https://developer.android.com/reference/android/app/Fragment.html>

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <android.support.design.widget.CoordinatorLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:app="http://schemas.android.com/apk/res-auto"
4      xmlns:fab="http://schemas.android.com/apk/res-auto"
5      xmlns:tools="http://schemas.android.com/tools"
6      android:layout_width="match_parent"
7      android:layout_height="match_parent"
8      android:fitsSystemWindows="true"
9      tools:context="br.com.jucemar_dimon.cadesaude.EstabecimentoActivity">
10
11     <android.support.design.widget.AppBarLayout
12         android:layout_width="match_parent"
13         android:layout_height="wrap_content"
14         android:theme="@style/MeuTemaMaterial.AppBarOverlay">
15
16         <android.support.v7.widget.Toolbar ...>
17
18         <android.support.design.widget.TabLayout
19             android:id="@+id/tabs"
20             android:layout_width="match_parent"
21             android:layout_height="wrap_content"
22             app:tabGravity="fill"
23             app:tabIndicatorColor="@color/icons"
24             app:tabMode="scrollable"
25             app:tabSelectedTextColor="@color/icons"
26             app:tabTextColor="@color/primary_light" />
27
28     </android.support.design.widget.AppBarLayout>
29     <android.support.v4.view.ViewPager
30         android:id="@+id/viewpager"
31         android:layout_width="match_parent"
32         android:layout_height="match_parent"
33         android:background="@color/background_material_light"
34         android:visibility="gone"
35         app:layout_behavior="@string/appbar_scrolling_view_behavior" />
36
37 </android.support.design.widget.CoordinatorLayout>
38
39
40
41
42
43
44
45
46
47
48
49
50

```

Fonte: Elaborado pelo autor

O elemento *TabLayout*<sup>62</sup> trata de gerar as guias nomeadas usadas pelo usuário para a abertura de cada uma das telas e o elemento *ViewPager*<sup>63</sup> é o responsável por armazenar e exibir cada uma das telas de *Fragments* a medida que forem chamadas.

Para o carregamento dos dados dos estabelecimentos em cada uma das telas é utilizado um processo em segundo plano disponibilizado através da classe privada *CarregadorDeDados* que herda as características da classe *AsyncTask* cujos comandos armazenados no método *doInBackground* são executados em uma *thread* separada da *UI Thread*. Para cada classe do tipo *Fragment* uma classe do banco de dados é utilizada:

- *AtendimentoFragment*: Utiliza a classe *AtendimentoDAO* para extrair do banco de dados um *ArrayList* de objetos do tipo *Atendimento* que será passado como argumento para que futuramente seja recuperado e apresentado na lista de atendimento quando o fragmento é exibido;
- *ProfissionaisFragment*: Utiliza a classe *ProfissionaisDAO* para extrair do

---

<sup>62</sup> <https://developer.android.com/reference/android/support/design/widget/TabLayout.html>

<sup>63</sup> <https://developer.android.com/reference/android/support/v4/view/ViewPager.html>

banco de dados um *ArrayList* de objetos do tipo *Profissional* que será passado como argumento para que futuramente seja recuperado e apresentado na lista de profissionais quando o fragmento é exibido;

- *ServicosFragment*: Utiliza a classe *ServicosDAO* para extrair do banco de dados um *ArrayList* de objetos do tipo *ServicoClassificacao* que será passado como argumento para que após seja recuperado e apresentado na lista de atendimento quando o fragmento é exibido;

O objeto da classe *Estabelecimento* que contém as informações básicas é extraído do objeto *Bundle*<sup>64</sup> da *intent* da *EstabelecimentoActivity* que por sua vez recebe tal objeto da lista de estabelecimentos armazenados pela *RecyclerView*. Tendo o objeto da classe *Estabelecimento* e a lista contendo os convênios e as formas de atendimento, a lista contendo os profissionais vinculados e a lista contendo os serviços especializados, todos os campos da tela que descreve o estabelecimento podem ser completados seus gráficos e textuais.

### **5.2.8 A publicação na loja de aplicativos**

Para facilitar ao usuário o processo de aquisição de novos aplicativos, a plataforma Android possui um sistema de filtros que é transparente aos usuários. Caso o desenvolvedor do aplicativo siga corretamente as instruções que são divulgadas na documentação oficial do Android acerca das configurações desses filtros em seu arquivo APK que é armazenado na loja de aplicativos do Google, esse sistema de filtros atingirá seu objetivo e somente permitirá aos usuários da *Google Play Store* instalar os aplicativos que são compatíveis com o seu aparelho. Caso o desenvolvedor não dispenda atenção suficiente sobre esta questão, pode haver um aumento significativo dos cenários em que o usuário instala um aplicativo através da *Google Play Store* e ele não funciona corretamente.

Esses sistemas de filtros são basicamente as declarações de elementos XML contendo as configurações desejadas dentro do arquivo *AndroidManifest.XML*. Os elementos utilizados para criação de filtros são baseados nas propriedades possíveis das declarações *<uses-feature>*<sup>65</sup>. Com estas declarações podem ser definidos quais

---

<sup>64</sup> <https://developer.android.com/reference/android/os/Bundle.html>

<sup>65</sup> <https://developer.android.com/guide/topics/manifest/uses-feature-element.html>

os componentes de hardware e software um dispositivo deve ter para ser compatível com o aplicativo que se está declarando. Sendo assim, somente se este dispositivo tiver tais características ele poderá receber tal aplicativo na *Google Play Store*. Para o aplicativo que está sendo desenvolvido formam declaradas as características obrigatórias definidas na Figura 38.

Figura 38 - definição das *Uses Feature* no arquivo de manifesto

```
<uses-feature  
    android:name="android.hardware.screen.portrait"  
    android:required="true" />  
<uses-feature  
    android:name="android.hardware.telephony"  
    android:required="true" />  
<uses-feature  
    android:name="android.hardware.location.gps"  
    android:required="true" />
```

Fonte: Elaborado pelo autor

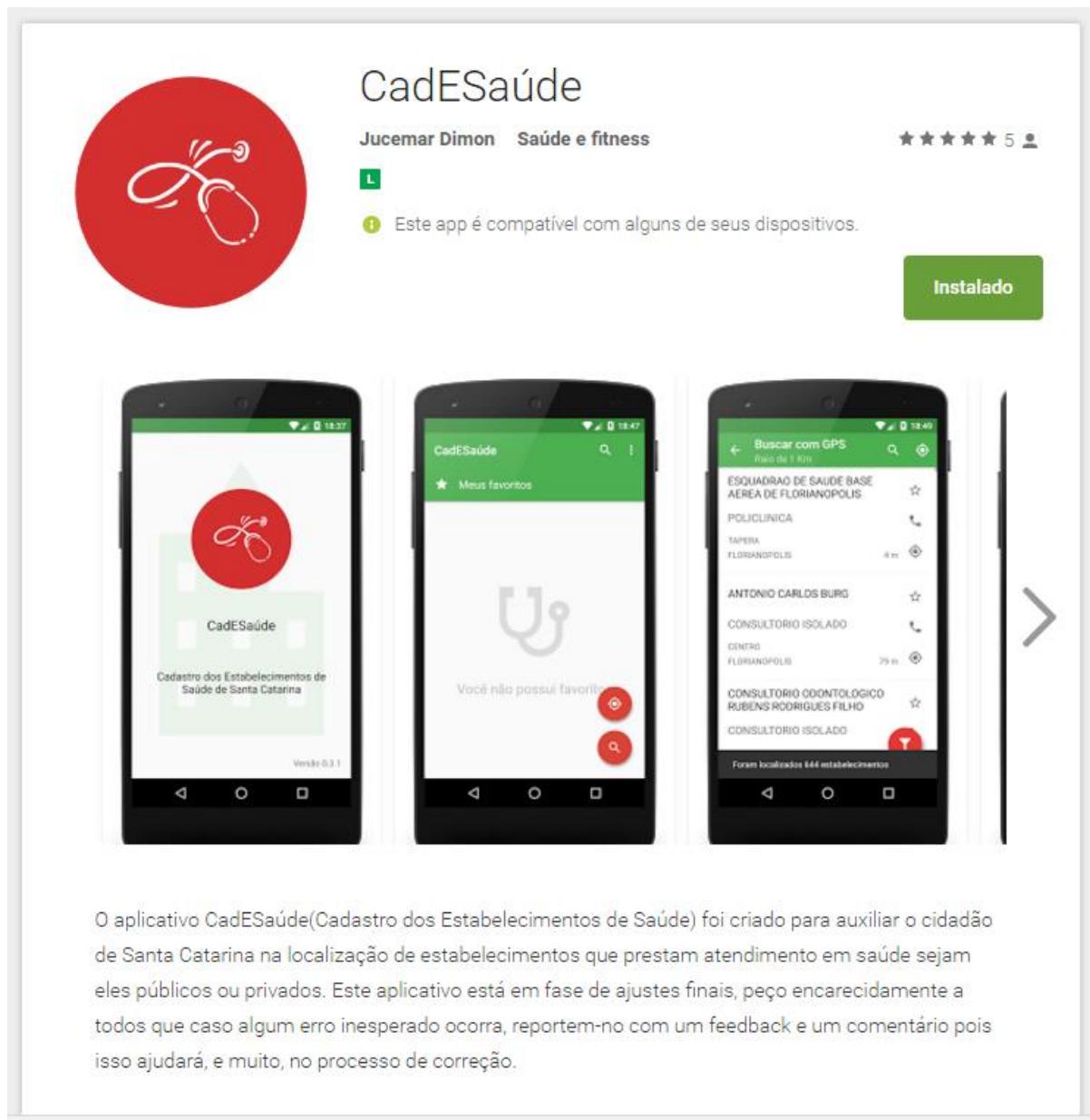
Segundo declarado na figura acima, para que o aplicativo seja compatível com o dispositivo ele somente necessita possuir um GPS, possuir a capacidade de realizar ligações telefônicas e poder utilizar a tela do dispositivo em modo vertical. Porém outras preocupações foram tomadas para garantir que além de compatível segundo os requisitos declarados, o aplicativo também funcionasse de forma correta.

Uma grande questão que poderia auxiliar no bom funcionamento do aplicativo no dispositivo no qual ele será instalado são os testes. A IDE *Android Studio* Fornece diversas possibilidades de testes de aplicativos, uma das mais interessantes foi utilizada durante o desenvolvimento do projeto: *Android Virtual Device*. Esta funcionalidade permite criar dispositivos virtuais simulando grande parte das funcionalidades de dispositivos reais. Dessa forma, pode-se testar o comportamento do aplicativo nas mais diversas versões do Android nas mais diversas características de hardware de dispositivos como tamanho e resolução de tela e poder de processamento. Para os testes no aplicativo foram utilizados emuladores com as seguintes versões do Android: 4.0; 4.1.1; 4.3; 4.4.4; 5.0; 5.1 e 6.0.

Após a realização de todas as etapas de desenvolvimento, preparação do projeto, geração e assinatura digital do arquivo APK, a sua publicação na loja de

aplicativos do Google através do portal *Google Play Developer Console* foi realizada e o aplicativo foi disponibilizado ao público. A primeira publicação do aplicativo foi feita no dia 08 de maio de 2016. Em sua versão inicial 0.01. A partir de então a cada nova alteração ou correções de bugs uma nova versão do aplicativo era disponibilizada. Na Figura 39 pode ser vista a página de apresentação do aplicativo.

Figura 39 - Página do aplicativo na Google Play Store



O aplicativo CadESaúde(Cadastro dos Estabelecimentos de Saúde) foi criado para auxiliar o cidadão de Santa Catarina na localização de estabelecimentos que prestam atendimento em saúde sejam eles públicos ou privados. Este aplicativo está em fase de ajustes finais, peço encarecidamente a todos que caso algum erro inesperado ocorra, reportem-no com um feedback e um comentário pois isso ajudará, e muito, no processo de correção.

Fonte: Elaborado pelo autor

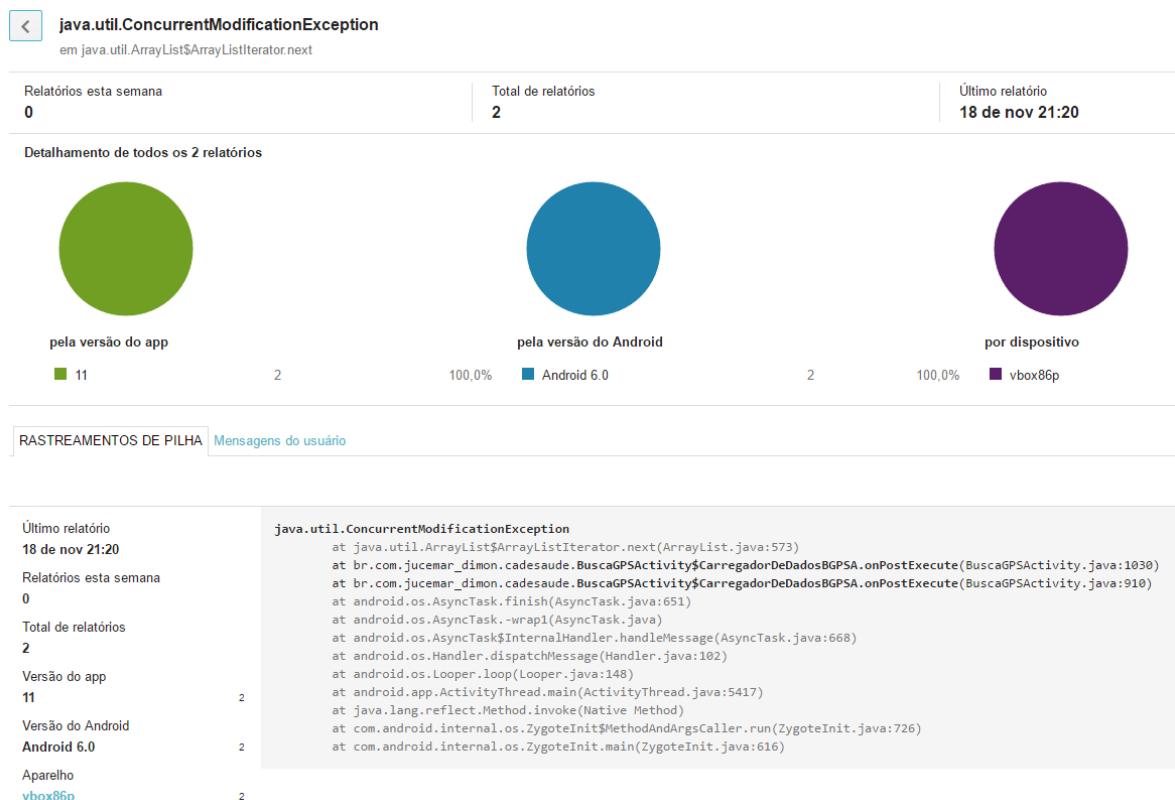
### 5.2.9 Avaliação do aplicativo

A avaliação do aplicativo pelos usuários foi feita de forma informal. Não foi feita uma divulgação em massa em nenhum veículo de comunicação ou qualquer outra do gênero sobre sua criação. Não se utilizou de nenhuma metodologia de avaliação da usabilidade da interface gráfica para testar o aplicativo. Foi feito apenas um simples teste na qual um pequeno grupo e pessoas foi convidado a realizar os simples atos de: entrar na página de divulgação do aplicativo, baixar, instalar, usar e reportar de suas impressões sobre o mesmo. As pessoas convidadas foram selecionadas segundo alguns requisitos:

- Selecionou-se pessoas com formações em áreas da saúde: Médicos, enfermeiras ou técnicos de enfermagem. Dessa forma caso algum termo técnico voltado a área da saúde fosse apresentado erroneamente, um respaldo seria dado;
- Pessoas com conhecimentos técnicos na área de desenvolvimento de aplicativos para a plataforma Android. Assim, caso alguma implementação de alguma ação dentro do aplicativo fugisse totalmente das práticas atuais utilizadas no desenvolvimento da plataforma um respaldo sobre sua correção o melhoramento seria oferecido;
- Pessoas com muita afinidade com a plataforma Android, que são usuários frequentes de smartphones e de seus aplicativos. Com isto, eles próprios poderiam realizar algum comparativo com outros aplicativos que usam frequentemente e reportar se o aplicativo CadESaúde foge aos padrões dos demais aplicativos.

Os convidados instalaram o aplicativo e retornaram suas impressões de forma informal. Alguns de forma verbal, outros através da avaliação do aplicativo em sua página de divulgação através de comentários ou classificações positivas. Foi possível também receber um retorno dos usuários contra falhas no aplicativo durante sua utilização pois a plataforma Android fornece ao usuário a opção de reportar ao responsável pelo aplicativo um erro ocorrido durante sua execução. O responsável pelo aplicativo pode visualizar o relatório de erros através do console do desenvolvedor da loja de aplicativo do Google. A Figura 40 ilustra um reporte de um caso de erro.

Figura 40 - Feedback com relatório de erro



Fonte: Elaborado pelo autor

Ao longo do desenvolvimento do aplicativo, foram publicadas 10 versões novas contendo as correções de vários itens: erros reportados pelos convidados; novas ideias sugeridas; correções de erros encontrados pelo autor durante o desenvolvimento; atualizações dos dados do banco de dados; entre outros.

Da forma como foi projetado, o aplicativo exige uma nova versão a cada mês para que se possa atualizar os dados contidos no banco de dados do aplicativo para que fiquem de acordo com o que é divulgado no site do CNES<sup>66</sup>. Quanto à estratégia de lançamento das novas versões, pensou-se em ser feita logo após o quinto dia útil de cada mês. Essa decisão foi tomada baseado no cronograma do CNES<sup>67</sup>, pois, geralmente, esse período é a data limite para que os municípios enviem ao banco de dados nacional as alterações feitas nos cadastros. Sendo assim uma única atualização mensal seria necessária para tanto corrigir bugs descobertos quanto a atualização do banco de dados. A atualização do banco de dados consiste em

66

[http://cnes2.datasus.gov.br/Lista\\_Tot\\_Es\\_Municipio.asp?Estado=42&NomeEstado=SANTA%20CATARINA](http://cnes2.datasus.gov.br/Lista_Tot_Es_Municipio.asp?Estado=42&NomeEstado=SANTA%20CATARINA)

<sup>67</sup> <http://cnes.datasus.gov.br/pages/acesso-rapido/cronograma.jsp>

executar a ferramenta de captura de dados na data estipulada, gerar uma nova base de dados, substituir a anterior pela atual no projeto da IDE *Android Studio*, preparar o projeto para exportação com a nova versão e fazer o upload do arquivo APK no console de desenvolvedor da Google para que o mesmo promova a notificação sobre uma nova atualização aos usuários que possuem o aplicativo instalado.

### **5.2.10 Análise geral do uso da memória**

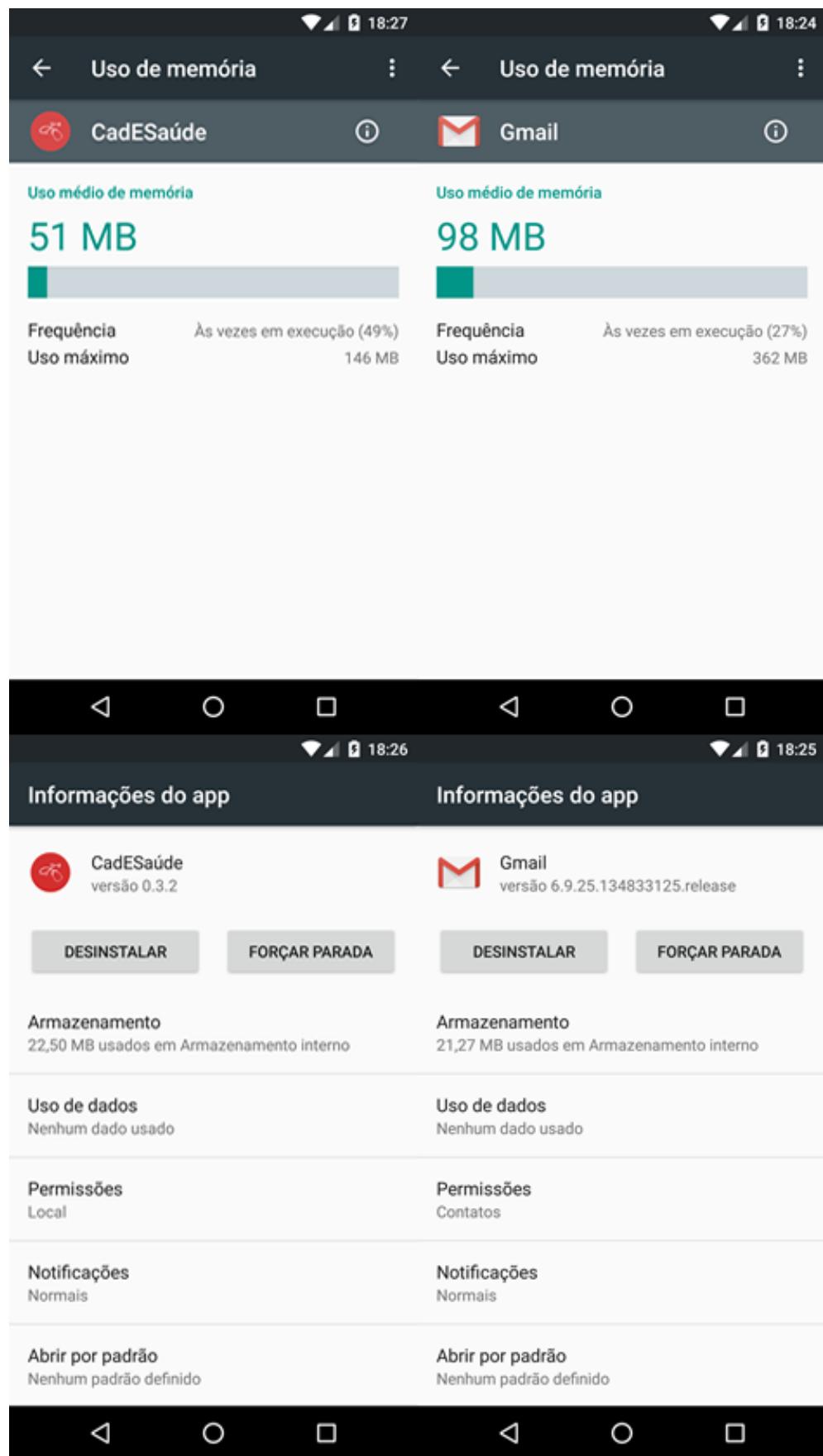
Uma das últimas versões da plataforma Android lançada, o Android 6.0 conhecida popularmente como Android *Marshmallow*, trouxe algumas novas ferramentas, e o melhoramento de diversas outras, para gerenciamento e controle do dispositivo. Uma destas ferramentas melhoradas foi a responsável pelo monitoramento da memória RAM dos aplicativos em execução, sua nova versão armazena o registro do uso de memória RAM de cada aplicativo possibilitando assim o monitoramento do gasto de memórias das aplicações em separado e também como um todo podendo assim saber o quanto de memória livre o dispositivo ainda possui (JUNIOR, 2016).

O objetivo desta seção é realizar um teste básico de memória do aplicativo desenvolvido. O aplicativo é totalmente off-line e possui um banco de dados interno frequentemente utilizado durante sua execução. Dessa forma são utilizados muitos dos recursos da memória RAM do dispositivo para auxiliar nos processos de consulta ao banco. Com esse pequeno teste pode-se verificar a situação dos picos de consumo da memória RAM do aplicativo no dispositivo. Assim, caso os picos de consumo de memória cheguem a valores que façam o aplicativo parar de responder, outras estratégias em relação a forma como os resultados das consultas ao banco que são armazenadas em memória deverão ser utilizadas.

Para que fosse possível verificar o comportamento do aplicativo em relação ao seu consumo de memória foi utilizado um emulador com as características do *smartphone LG NEXUS 5* que podem ser conferidas no site da própria fabricante em LG (2016). A memória RAM que este modelo de telefone possui é de 2GB e para este teste de memória foi utilizado o gerenciador de memória do Android *Marshmallow*. O aplicativo *CadESaúde*, desenvolvido neste trabalho, foi aberto no emulador e ficou ativo durante 3 horas e em tempos aleatórios o seu uso era feito. Para que se pudesse ter algum parâmetro para comparação, utilizou-se um aplicativo popular para que

também participasse do teste. O aplicativo utilizado foi o *Gmail*. Este aplicativo foi escolhido pois ele é nativo da plataforma Android. Ele foi desenvolvido pela própria Google que também é responsável pela plataforma. Pode-se interpretar que foi bem desenvolvido e que se preocupa com questões como o consumo de memória RAM de recursos de processamento. Logo, pareceu ser a melhor opção de aplicativo a se comparar. Sendo assim, se os resultados da comparação mostrarem que o CadESaúde possui valores maiores do que o *Gmail* em relação ao consumo de memória RAM, alguma ação corretiva deveria ser tomada. Ambos os aplicativos foram iniciados e ficaram ativos durante todo o tempo do teste. A Figura 41 abaixo exibe o resultado.

Figura 41 - Comparativo do consumo de memória RAM



Fonte: Elaborado pelo autor

Visualizando os valores nas telas capturadas no emulador na parte superior da Figura percebe-se que o *CadESaúde* possui um consumo médio de memória menor que o *Gmail*. Ambos aplicativos testados não são tipos de aplicativo nos quais são utilizados durante horas seguidas. O *Gmail*, por exemplo, é um aplicativo para enviar e receber e-mails, possivelmente um usuário recebe uma notificação de nova mensagem, abre o aplicativo lê a mensagem e na sequência fecha o mesmo. O *CadESaúde* também é um tipo de aplicativo no qual o usuário vai abri-lo, procurar o estabelecimento que necessita e na sequência, possivelmente, irá fechar o aplicativo. Durante o seu uso podem acontecer picos de utilização da memória RAM como foi observado na figura, o aplicativo *CadESaúde* teve um uso máximo de memória em algum momento do tempo em que foi realizado o teste, porém sua média de utilização é menor se comparada com o aplicativo do *Gmail*.

Observando os dados das telas capturadas do emulador percebe-se na parte inferior da figura os valores dos armazenamentos internos de cada um dos aplicativos. *CadESaúde*, apesar de possuir um banco de dados interno contendo aproximadamente 15.000 estabelecimentos de saúde armazenado em um arquivo, ocupa somente 22,50MB da memória interna do aplicativo.

## 6 CONCLUSÕES

O projeto desenvolvido finda com a produção da aplicação de captura de dados e com a publicação do aplicativo de busca por estabelecimentos na loja de aplicativos da Google.

O desenvolvimento de uma ferramenta de captura de dados para adquirir as informações sobre os estabelecimentos de saúde que foi apresentado na seção 1 do capítulo 4. As experiências vividas nesta etapa do projeto relembram a questão da importância da qualidade dos dados. Mesmo com todo o embasamento legislativo e suporte aos sistemas de saúde no DATASUS, ainda assim, foram encontrados casos de dados mal formatados, textos e campos de textos que não seguem um padrão de escrita. Um exemplo desse fato foi a falta do número de contato telefônico de alguns estabelecimentos. Uma informação como esta é de vital importância não só no contexto deste trabalho. Também tiveram casos em que o campo de telefone continha um número, porém ele, ou estava mal formatado, ou faltando um dígito. Outra questão que penalizou o resultado do trabalho foi quantidade de erros nos nomes dos logradouros de alguns estabelecimentos. Tal fato produziu discrepâncias nos resultados de busca das coordenadas geográficas de alguns estabelecimentos. Isto causou divergências quanto a sua localização exata durante sua visualização em um mapa. Contudo, pode-se considerar atingido o objetivo específico número 2, pois mesmo produzindo informações um pouco imprecisas dos estabelecimentos, a ferramenta cumpre o papel que se propôs.

Os demais objetivos específicos podem ser definidos como atingidos. O objetivo específico 1, que tratou da criação de uma especificação de requisitos para o aplicativo, pode ser visualizado no Apêndice I do documento. O objetivo específico 3, que tratou do desenvolvimento de um aplicativo, pode ser visto em detalhes na seção 2 do capítulo 4. E, por fim, o objetivo específico 4, que trata da publicação do aplicativo na *Google Play Store*, pode ser visto na subseção 2.8 do capítulo 4.

Pode-se dizer que o objetivo geral do projeto também foi atingido, pois produziu-se algo que fica muito próximo do projetado. O atingimento dos objetivos específicos somados, foi capaz de produzir uma solução para auxiliar os cidadãos do Estado de Santa Catarina na busca por estabelecimentos de saúde. O fato de o aplicativo não necessitar de uma conexão com a internet, retirou dele um fator

limitador em aplicativos que a exigem, pois existem áreas interioranas onde a rede da operadora de telefonia não oferece uma cobertura adequada. Dessa forma o aplicativo poderia se tornar inoperante. Contudo, questões como o tempo de execução do processo da captura dos dados da ferramenta de captura e a falta de um padrão no formato dos dados dos estabelecimentos deixaram a desejar.

Os temas e tecnologias aqui abordados acrescentaram grande conhecimento para seu autor, pois uma vasta pesquisa da plataforma Android foi necessária para o desenvolvimento do aplicativo. O foco principal das técnicas necessárias para o seu desenvolvimento foram a manipulação da API de localização para obter acesso aos dados do GPS e da API de banco de dados para a realização de consultas. Um conhecimento aprofundado em HTML foi adquirido neste trabalho. Tal aquisição foi necessária pois a biblioteca JSOUP que foi utilizada na construção da ferramenta de captura de dados, utiliza a manipulação direta dos elementos dessa linguagem. Este conhecimento fez a diferença no momento da criação dos métodos que navegam nas páginas do portal do CNES pois estas não seguem o padrão W3C.

## **6.1 TRABALHOS FUTUROS**

Esta seção fornece ao leitor algumas questões que podem ajudá-lo a identificar possibilidades de continuar os trabalhos desenvolvidos aqui, melhorando de forma geral o sistema em todos os seus aspectos.

### **6.1.1 Melhorias na ferramenta de captura de dados**

Sugere-se a melhoria da ferramenta de captura de dados pois em seu estado atual ela utiliza-se de apenas uma *Thread* para realizar todos os processos de gerenciamento do banco de dados, requisições ao servidor do DATASUS, consumo de webservices para aquisição de geocoding de endereços dos estabelecimentos e leitura de arquivos de texto. Dessa forma, a realização de todas essas tarefas automatizadas demanda muito tempo.

### **6.1.2 Adaptação do aplicativo para o funcionar parcialmente online**

Sugere-se a adequação do sistema para que possibilite a utilização do

aplicativo em todo o território nacional. Para tal deverá ser pesquisada uma técnica que possibilite tal alteração de forma que a performance não seja afetada. As alterações necessárias para o aplicativo Android trabalhar com Estados do Brasil diferentes seria a troca do arquivo de banco de dados. Sendo assim não há como criar a possibilidade de uma opção de “configurar” o aplicativo para outros estados pois no arquivo APK de instalação não caberiam todos os estados. A estrutura do sistema hoje só permite que haja um arquivo de banco de dados no aplicativo contendo dados de um estado, para que se pudesse criar uma opção de configuração de trocar o Estado, teria que se adotar a estrutura de sistema na qual o aplicativo teria de usar uma conexão com a internet para baixar o arquivo de banco de dados do estado solicitado pelo usuário e isso envolveria não somente uma conexão com a internet mas também a necessidade da existência de um servidor onde seriam armazenados todos os arquivos de cada estado juntamente com algum webservice para controlar todas essas requisições de downloads.

### **6.1.3 Adaptação do aplicativo para o funcionar totalmente online**

O foco do aplicativo aqui desenvolvido é o de ser uma ferramenta prática que não exija uma conexão com a internet para funcionar, pois todos os dados dos estabelecimentos estão armazenados em um banco de dados interno. Tanto os processos de consulta aos dados quanto a sua exibição utilizam dos recursos do dispositivo para serem executadas. Sendo assim para um cenário onde tem-se estabelecimentos de um único estado do país, tal técnica poderia funcionar relativamente bem. Porém, em um cenário onde o mesmo projeto fosse adaptado para comportar dados de estabelecimentos de outros estados do Brasil, a técnica utilizada aqui deveria ser abandonada. Provavelmente, esta técnica geraria um tamanho de arquivo que ultrapassaria os limites permitidos pela loja de aplicativos da Google, mesmo tal limite de tamanho ter sido dobrado de 50MB para 100MB com afirmado por Farinaccio (2016), ainda assim a questão do processamento das consultas poderia causar possíveis gargalos.

Para uma adaptação onde se teriam informações de todos os estabelecimentos do país, abandonar-se-ia a ideia de aplicativo off-line. Dever-se-ia adotar uma estrutura onde um servidor online armazenaria os dados dos estabelecimentos disponibilizá-los-ias através de um webservice. O aplicativo, através

de conexão com a internet consumiria tal serviço solicitando consultas ao banco de dados armazenado no servidor. Dessa forma o custo computacional para as consultas ficaria a cargo do servidor e o aplicativo seria focado na realização de requisições ao servidor e apresentação dos dados recebidos ao usuário.

#### **6.1.4 Reengenharia da interface gráfica**

O projeto desenvolvido tinha como foco principal a criação de um aplicativo para localizar estabelecimentos de saúde dentro do Estado de Santa Catarina. Todos os esforços gastos foram utilizados para atingir este objetivo. E tal objetivo pode ser considerado atingido, porém, o resultado de todo esse trabalho gerou algumas questões que podem ser repensadas em relação a usabilidade do aplicativo. Dessa forma, propõem-se como trabalhos futuros um estudo aprofundado utilizando as mais modernas técnicas de engenharia da usabilidade visando a criação de novas telas, mais intuitivas e de simples de se utilizar.

## REFERÊNCIAS

ALVES, Paulo. **A evolução do Android: do Cupcake ao Marshmallow, conheça todas as versões.** Disponível em:

<<http://www.techtudo.com.br/noticias/noticia/2015/10/a-evolucao-do-android-do-cupcake-ao-marshmallow-conheca-todas-as-versoes.html>>. Acesso em: 31 ago. 2016.

ANDROID DEVELOPER. **Dashboards.** Disponível em:

<<https://developer.android.com/about/dashboards/index.html#Platform>>. Acesso em: 20 mai. 2016.

ANDROID OPEN SOURCE PROJECT. **Android Compatibility.** Disponível em:

<<http://source.android.com/compatibility/index.html>>. Acesso em: 30 ago. 2016.

ANDROID OPEN SOURCE PROJECT. **Governance Philosophy.** Disponível em:

<<http://source.android.com/source/index.html#governance-philosophy>>. Acesso em: 30 ago. 2016.

ANDROID OPEN SOURCE PROJECT. **Life of a Patch.** Disponível em:

<<http://source.android.com/source/life-of-a-patch.html>>. Acesso em: 30 ago. 2016.

ANDROID OPEN SOURCE PROJECT. **Project Roles.** Disponível em:

<<http://source.android.com/source/roles.html>>. Acesso em: 30 ago. 2016.

ANDROID STUDIO. **User Guide.** Disponível em:

<<https://developer.android.com/studio/projects/index.html>>. Acesso em: 21 set. 2016.

ANDROID. **Codenames, Tags, and Build Numbers.** 2016a. Disponível em:

<<http://source.android.com/source/build-numbers.html#platform-code-names-versions-api-levels-and-ndk-releases>>. Acesso em: 30 ago. 2016.

ANDROID. **Platform Architecture.** 2016b. Disponível em:

<<https://developer.android.com/guide/platform/index.html>>. Acesso em: 1 set. 2016.

BRASIL. MINISTRO DE ESTADO DA SAÚDE. Portaria nº 1.646, de 02 de outubro de 2015. **Diário Oficial da União**, Brasília, 2 out. 2015, v. , n. 190, p. 669-670.

BRASIL. MS-SAS - Secretaria de Atenção a Saúde. Portaria nº 376, de 03 de outubro de 2000. **Diário Oficial da União**, Brasília, 3 out. 2000, v. , n. 192-E, p. 19-39.

BRASIL. MS-SAS: Secretaria de Assistência a Saúde . Portaria nº 511, de 29 de dezembro de 2000. **Diário Oficial da União**, Brasília, 4 jan. 2001, v. , n. 3-E, p. 11-20.

BRASIL. MS-SAS - Secretaria de Atenção a Saúde. Portaria nº 682, de 10 de dezembro de 2010. **Diário Oficial da União**, Brasília, 10 dez. 2010, v. , n. 237, p. 52-53.

BUENO, Juliano Martins. **Ciro Radiologia Odontológica**. 2016. Disponível em: <<https://play.google.com/store/apps/details?id=com.embarcadero.ciro>>. Acesso em: 06 dez. 2016.

CLINICA PRIVADA DE ESPECIALIDADES VILLA MARÍA. **Clinica Especialidades**. Disponível em: <<https://play.google.com/store/apps/details?id=clinicaespecialidades.myapp.name>>. Acesso em: 25 out. 2015.

CNES. **CONSULTA LEGISLAÇÃO**. 2016. Disponível em: <<http://cnes.datasus.gov.br/pages/legislacoes.jsp>>. Acesso em: 06 dez. 2016.

DATASUS. **CNES - Cadastro Nacional de Estabelecimentos de Saúde**. Disponível em: <<http://datasus.saude.gov.br/sistemas-e-aplicativos/cadastros-nacionais/cnes>>. Acesso em: 6 set. 2016.

DATASUS. **Consultas - Limites de Competências**. 2015a. Disponível em: <<http://200.214.44.204/transmissor-web/pages/consultaLimiteCompetencia.jsp?cmbNomeSistema=CNES>>. Acesso em: 21 nov. 2015.

DATASUS. **Estabelecimento de saúde**: Portal CNESnet. 2015b. Disponível em: <[http://cnes2.datasus.gov.br/Exibe\\_Ficha\\_Estabelecimento.asp?VCo\\_Unidade=4205403157245&VEstado=42&VCo;](http://cnes2.datasus.gov.br/Exibe_Ficha_Estabelecimento.asp?VCo_Unidade=4205403157245&VEstado=42&VCo;)>. Acesso em: 23 maio 2015.

DATASUS. **SCNES: Sistema de Cadastro Nacional de Estabelecimentos de Saúde - Manual de Instalação e Administração**. 3 ed. Rio de Janeiro: MS/DATASUS/CTI, 2009a.

DATASUS. **SCNES: Sistema de Cadastro Nacional de Estabelecimentos de Saúde - Manual de Operação**. 3 ed. Rio de Janeiro: MS/DATASUS/CTI, 2009b.

DEITEL, Paul; DEITEL, Harvey; WALD, Alexander. **Android 6 para programadores : uma abordagem baseada em aplicativos**. 3 ed. Porto Alegre: Bookman, 2016.

DEVDECISION CONSULTORIA EM TECNOLOGIA. **SIPS - Guia Hospitais**. Disponível em: <<https://play.google.com/store/apps/details?id=com.sipsv2>>. Acesso em: 24 mai. 2015.

EXAME.COM. **Vendas mundiais de smartphones crescem 23% em 2014**. Disponível em: <<http://exame.abril.com.br/tecnologia/noticias/vendas-mundiais-de-smartphones-crescem-23-em-2014>>. Acesso em: 23 mai. 2015.

FARINACCIO, Rafael. **Limite de tamanho de apps na Google Play Store sobe de 50 MB para 100 MB**. Disponível em: <<http://www.tecmundo.com.br/google-play/87155-limite-tamanho-apps-google-play-store-de-50-mb-100-mb.htm>>. Acesso em: 4 out. 2016.

FREEMAN, Eric; FREEMAN, Elisabeth; SIERRA, Kathy; et al. **Use a Cabeça:**

**Padrões e Projetos.** 2 ed. Rio de Janeiro: Alta Books, 2009.

**GLOBALDEV - SISTEMAS WEB E APlicativos MOBILE. Clinica Santa Cecilia.** Disponível em: <<https://play.google.com/store/apps/details?id=br.clinicasantacecilia>>. Acesso em: 25 out. 2015.

**GOOGLE DESIGN. Material Design Guidelines.** Disponível em: <<https://www.google.com/design/spec/material-design/introduction.html>>. Acesso em: 31 mai. 2016.

**HAMANN, Renan. Android: entenda as diferenças entre o ART e o Dalvik.** Disponível em: <<http://www.tecmundo.com.br/android/54387-android-entenda-diferencias-entre-art-o-dalvik.htm>>. Acesso em: 1 set. 2016.

**HEDLEY, Jonathan. JSoup: Java HTML Parser.** Disponível em: <<https://jsoup.org/>>. Acesso em: 26 out. 2015.

**JUNIOR, Alessandro A.. Android Marshmallow: veja quais apps estão consumindo memória RAM.** Disponível em: <<http://www.techtudo.com.br/noticias/noticia/2015/12/android-marshmallow-veja-quais-apps-estao-consumindo-memoria-ram.html>>. Acesso em: 12 out. 2016.

**LG. LG NEXUS 5: ESPECIFICAÇÃO TÉCNICA.** Disponível em: <<http://www.lg.com/br/celulares/lg-D821>>. Acesso em: 12 out. 2016.

**LUZZI, Luciano. Android – Persistência de Dados usando SQLite.** Disponível em: <<http://www.mobiltec.com.br/blog/index.php/android-persistencia-de-dados-usando-sqlite>>. Acesso em: 29 out. 2015.

**MEDSQUARE. medsquare.** Disponível em: <<http://www.medsquare.com.br>>. Acesso em: 25 out. 2015.

**MERCATO, Mattia. A doce história do Android.** Disponível em: <<http://www.androidpit.com.br/historia-do-android>>. Acesso em: 31 ago. 2016.

**MEYER, Maximiliano. A história do Android.** Disponível em: <<https://www.oficinadanet.com.br/post/13939-a-historia-do-android>>. Acesso em: 31 ago. 2016.

**MINISTÉRIO DA SAÚDE. MANUAL TÉCNICO DO CADASTRO NACIONAL DE ESTABELECIMENTOS DE SAÚDE - VERSÃO 2.** 1 ed. Brasília: Ministério da Saúde, 2006.

**NAÍMA SALEH. Revista Crescer. Mais de 70% dos pais exageram no uso de smartphones, aponta pesquisa.** 2015. Portal GLOBO.COM. Disponível em: <<http://revistacrescer.globo.com/Familia/Rotina/noticia/2015/08/mais-de-70-dospais-exageram-no-uso-d>>. Acesso em: 06 dez. 2016.

**OLIVEIRA, Caio. DBDesigner Fork.** Disponível em: <<http://www.caiooliveira.com.br/?p=66>>. Acesso em: 30 mai. 2015.

**PORTAL UNIMED LTDA. Guia Médico Nacional Unimed para smartphone.**

Disponível em:

<<https://play.google.com/store/apps/details?id=com.unimed.guamdico>>. Acesso em: 23 mai. 2015.

SARINHO, Victor Travassos; CAMPOS, Luan Rios. SAMob - Sistema Móvel de Geolocalização e Geoprocessamento para Locais de Atendimento em Saúde.

**Journal of Health Informatics.** São Paulo; v. 6, n. 3, set. 2014. p. 96-101.

Disponível em: <<http://www.jhi-sbis.saude.ws/ojs-jhi/index.php/jhi-sbis/article/view/304>>. Acesso em: 18 out. 2015.

SCHMIDT, Roberto Hammer. **Linha do Tempo: por dentro da evolução do Android.** Disponível em: <<http://www.tecmundo.com.br/android/82344-linha-tempo-dentro-evolucao-do-sistema-android.htm>>. Acesso em: 31 ago. 2016.

**SOCIAL ADVISORS. Clínica Bicentenario.** Disponível em:

<[https://play.google.com/store/apps/details?id=biz.app4mobile.app\\_4bb5870e058a4ae9b067d271d142c996.app](https://play.google.com/store/apps/details?id=biz.app4mobile.app_4bb5870e058a4ae9b067d271d142c996.app)>. Acesso em: 25 out. 2015.

TABORDA, Sérgio Manuel Marcos. **DAO.** Disponível em:

<<https://sergiotaborda.wordpress.com/desenvolvimento-de-software/java/patterns/dao>>. Acesso em: 30 mai. 2015.

TAVARES, Alberto Trindade; LÓSCIO, Bernadette Farias. Processamento de consultas na Web de Dados: uma abordagem para busca de fontes de dados relevantes. In: SIMPÓSIO BRASILEIRO DE BANCO DE DADOS. 28., 2013, Recife.

**Short Papers...** Recife: SBC, 2013. Disponível em:

<<http://sbbd2013.cin.ufpe.br/Proceedings/artigos/pdfs/SBBD-Proceedings-Complete.pdf>>. Acesso em: 25 out. 2015.

TAVARES, Alberto Trindade; LÓSCIO, Bernadette Farias; OLIVEIRA, Hélio Rodrigues de. Buscando Fontes de Dados Relevantes para Aplicações Linked Data. In: SIMPÓSIO BRASILEIRO DE SISTEMAS MULTIMÍDIA E WEB. 18., 2012, São Paulo. **Anais IX Workshop de Trabalhos de Iniciação Científica...** São Paulo: SBC, 2012. Disponível em:

<[http://sws2012.ime.usp.br/download/WebMedia2012\\_Anais.pdf](http://sws2012.ime.usp.br/download/WebMedia2012_Anais.pdf)>. Acesso em: 25 out. 2015.

**VOXAGE SERVIÇOS INTERATIVOS. MEATENDE.** Disponível em:

<<http://www.voxage.com.br/meatende/>>. Acesso em: 06 dez. 2016.

**WINTIMEIPC. MedSquare.** Disponível em:

<[https://play.google.com/store/apps/details?id=br.com.medsquare.medsquare&hl=pt\\_BR](https://play.google.com/store/apps/details?id=br.com.medsquare.medsquare&hl=pt_BR)>. Acesso em: 24 mai. 2015.

**XAVIER, Renan. Clinica Santa Cecilia.** Disponível em:

<<https://play.google.com/store/apps/details?id=br.clinicasantacecilia>>. Acesso em: 25 out. 2015.



## **APÊNDICES**

## **APÊNDICE A - ESPECIFICAÇÃO DE REQUISITOS DE SOFTWARE**

# **Especificação de Requisitos do Aplicativo CadESaúde**

### **Projeto: CadESaúde**

Versão	Autor	Data	Ação
1.0	Jucemar Dimon	15/12/2015	Estabelecimento de Requisitos

### **Conteúdo:**

1. Introdução
2. Visão Geral
3. Requisitos de Software

### **Introdução:**

**Objetivo:** Desenvolvimento de um aplicativo para a plataforma Android que permita ao usuário portando um smartphone com o programa devidamente instalado localizar um estabelecimento de saúde de forma mais eficiente.

**Descrição:** CadESaúde é um aplicativo para smartphone que será desenvolvido com o intuito de auxiliar o cidadão, morador ou turista, do Estado de Santa Catarina a localizar um estabelecimento que presta atendimento em saúde seja ele privado ou público. Entende-se por estabelecimento que presta atendimento em saúde como: clínicas ou consultórios médicos, clínicas ou consultórios odontológicos, policlínicas, hospitais, entre outros. Portando de um smartphone com o aplicativo instalado e o serviço de localização ativo, será possível visualizar uma lista de todos os estabelecimentos de saúde próximos dentro de um raio de circunferência que deve ser pré-definido pelo usuário nas configurações do aplicativo. Em posse dessa lista de locais o usuário terá a opção de utilizar diversos filtros para auxiliar na visualização do que procura no momento. Um filtro pode ser aplicado um em sobreposição a outro para aumentar a eficácia da busca. As opções filtros vão de tipo de estabelecimento, passando por serviços oferecidos até os tipos de convênios aceitos. Além da busca por localização geográfica que auxiliará o usuário a encontrar estabelecimentos independente da cidade onde estiver dentro do estado de Santa Catarina, há ainda a possibilidade de encontrar estabelecimentos dentro das cidades pré-definidas nas configurações do aplicativo. Nessa categoria de busca será possível realizar buscas tabuladas por tipos de estabelecimentos, por tipos de especialidades médicas existentes nestes além da possibilidade de buscas por nome. O aplicativo será desenvolvido e sua versão final e utilizável será publicada na loja de aplicativos da Google.

### **Visão Geral:**

**Paradigma do Programa:** Programa Orientado a Objetos.

**Premissas de Desenvolvimento:** O aplicativo será desenvolvido na linguagem Java utilizando a Android API 4.0.3 – API Level 15 - e deverá ser compatível com aparelhos smartphones que utilizem sistema operacional Android em suas versões 4.0.3 - Ice Cream Sandwich ou superiores. A interface gráfica com o usuário será desenvolvida utilizando os artefatos mais modernos nativos da versão mais atualizada da API do Android que no momento do desenvolvimento deste projeto seria a 6.0 juntamente com os componentes visuais fornecidos pelo padrão visual Material Design com sua biblioteca de compatibilidade com versões antigas.

## Requisitos de Software:

### Requisitos Funcionais:

**Requisito funcional 1 – Definir raio de distância:** A tela de configurações do aplicativo deverá exibir ao usuário a opção de definir um raio de distância de sua preferência. Tal informação será utilizada na funcionalidade de busca por estabelecimentos através de localização geográfica.

**Requisito funcional 2 – Definir municípios de preferência:** A tela de configurações do aplicativo deverá exibir ao usuário uma lista de municípios de Santa Catarina na qual este poderá selecionar o seu município de residência e outros adjacentes de seu interesse. Tais informações serão obrigatórias para o funcionamento do aplicativo pois através delas serão possíveis realizar as buscas tabuladas onde os estabelecimentos resultantes serão aqueles instalados nos municípios pré-definidos no aplicativo.

**Requisito funcional 3 – Visualizar estabelecimentos favoritos:** O aplicativo deve permitir ao usuário visualizar uma lista contendo todos os estabelecimentos de saúde que foram marcados como favoritos em uma situação anterior.

**Requisito funcional 4 – Buscar estabelecimentos pelas redondezas:** O aplicativo deverá permitir ao seu utilizador usar a sua posição geográfica atual, adquirida através do GPS do smartphone, para visualizar os estabelecimentos nas redondezas até um limite de distância pré-definido anteriormente pelo usuário no menu de configurações do aplicativo.

**Requisito funcional 5 – Filtrar resultados de buscas por estabelecimentos:** O aplicativo deverá permitir que filtros possam ser aplicados aos resultados das buscas por estabelecimentos nas localidades com o intuito de facilitar a identificação de uma característica específica de interesse do usuário. Deve haver a possibilidade de aplicação de filtro sobre filtro para tornar o processo mais eficiente. Os filtros devem ter relação com os dados disponíveis no banco de dados referenciando cada estabelecimento. Os filtros a serem disponibilizados aos usuários serão:

- Tipos de convênios;
- Tipos de atendimentos;
- Tipos de profissionais;
- Tipos de serviços;
- Tipos de estabelecimentos.

**Requisito funcional 6 – Visualizar os detalhes de um estabelecimento:** Deverá ser possível ao usuário visualizar os dados detalhados do estabelecimento encontrado. Entende-se por dados detalhados todas as informações relacionadas a este estabelecimento que estão armazenadas no banco de dados do aplicativo. As principais informações que deverão ser mostradas na tela são:

- Informações empresariais: nome fantasia, razão social;
- Informações sobre a localização onde está instalado o negócio;
- Informações sobre o tipo de estabelecimento: se ele é uma clínica, um hospital, uma policlínica, entre

- outros;
- Telefones de contato;
  - Informações sobre os profissionais que prestam atendimento no local;
  - Informações sobre os serviços oferecidos pela empresa;
  - Informações sobre os tipos de atendimento realizado no local: se faz atendimento de urgência, se faz internação do paciente, se realiza exames no estabelecimento, entre outros;
  - Informações sobre os tipos de convênios que aceita: SUS, particular, privado, entre outros.

**Requisito funcional 7 – Visualizar mapa com a localização geográfica do estabelecimento:** O aplicativo deve permitir ao usuário a possibilidade de visualização de um mapa contendo a localização geográfica do estabelecimento que está sendo visualizado em detalhes. Tal mapa não será fornecido junto com o aplicativo. Este somente fará a comunicação com o sistema operacional Android e repassará alguns dados que possibilitem ao aplicativo que foi definido pelo usuário como aplicativo padrão para visualização de mapas para que este desempenhe suas tarefas de marcar o estabelecimento no mapa, definir rotas até este ponto, entre outros.

**Requisito funcional 8 – Definir estabelecimento como favorito:** O aplicativo deve permitir ao usuário optar por marcar ou desmarcar como favorito um estabelecimento de saúde que esteja sendo visualizado de forma detalhada na tela do celular.

**Requisito funcional 9 – Realizar uma ligação telefônica para um estabelecimento:** O aplicativo deve permitir ao usuário optar por fazer uma ligação telefônica para o estabelecimento que está tendo seus dados detalhados visualizados. O aplicativo não fará por conta própria a ligação, ele simplesmente solicitará ao sistema operacional Android que abra o aplicativo padrão de ligações telefônicas e faça uma ligação para o número de telefone e a partir deste ação será concluída.

**Requisito funcional 10 – Buscar estabelecimentos de forma generalizada:** O aplicativo deve permitir ao usuário buscar estabelecimentos de acordo com alguma característica específica ou comum entre os estabelecimentos. As possibilidades de busca serão:

- Busca por estabelecimentos que trabalham com tipo de convenio específico. Neste caso o usuário deverá escolher uma dentre uma relação pré-definida de tipos de convênio;
- Busca por estabelecimentos que possuam uma forma de atendimento específica. Nesta situação o usuário deverá escolher uma dentre as possíveis disponíveis para realizar sua busca;
- Busca por estabelecimentos que possuam em seu quadro de funcionários um profissional com um nome equivalente ao informado pelo usuário;
- Busca por estabelecimentos que possuam um profissional do tipo escolhido pelo usuário de uma lista de possibilidades;
- Busca por estabelecimentos que possuam em seu nome fantasia ou razão social um termo a ser informado pelo usuário;
- Busca por estabelecimentos que possuam um tipo de serviço listado e escolhido pelo usuário do aplicativo durante o processo de busca.

Os resultados das buscas realizadas nesta funcionalidade são limitados de acordo com o município de instalação do estabelecimento, dessa forma somente aparecerão nos resultados as empresas localizadas dentro dos municípios pré-definidos pelo usuário na tela de configuração da aplicação.

**Requisito funcional 11 – Fornecer ajuda ao usuário do aplicativo:** O aplicativo deverá fornecer ao usuário um menu onde ele poderá tirar algumas dúvidas em relação aos termos técnicos utilizados nas nomenclaturas das áreas da saúde e medicina.

## Requisitos Não-Funcionais:

**Requisito não-funcional 1 – Sistema de localização:** O smartphone no qual será instalado o aplicativo deverá possuir um sistema de localização baseado em GPS e este deverá fornecer a localização geográfica do aparelho de forma rápida e precisa com risco de comprometer o desempenho final do aplicativo caso este não atenda aos requisitos definidos.

**Requisito não-funcional 2 – Independência de conexão com a internet:** O aplicativo desenvolvido não depende

da conexão com a internet para executar suas funções básicas planejadas, porém algumas funções adicionais possíveis de serem realizadas pelo mesmo como a instalação e atualização do aplicativo necessitam que o aparelho no qual ele esteja instalado ou irá ser instalado pela primeira vez, possua um conexão ativa com a internet pois é através dela que os avisos sobre novas versões lançadas serão enviados.

**Requisito não-funcional 3 – Aplicativo de visualização de mapas:** O smartphone no qual o CadESaúde será instalado deverá possuir um aplicativo padrão para visualização de mapas. Este aplicativo poderá exigir ou não uma conexão com a internet, não existe restrição quanto a isso.

**Requisito não-funcional 4 – Tipo de dispositivo:** CadESaúde é compatível somente com smartphones, logo, tablets, TVs, wearables e autos, entre outros, não serão compatíveis. Ele funcionará somente em modo retrato (*portrait*) será instalado deverá possuir um aplicativo padrão para visualização de mapas. Este aplicativo poderá exigir ou não uma conexão com a internet, não existe restrição quanto a isso.

```
1 package br.com.jucemar_dimon.jwebcrawlers.entidades;
2
3 import java.io.Serializable;
4
5 /**
6 * @author Jucemar
7 */
8 public class AtendimentoPrestado implements Serializable {
9
10    private String tipoAtendimento;
11    private String convenio;
12
13    public AtendimentoPrestado() {
14    }
15
16    public String getTipoAtendimento() {
17        return tipoAtendimento;
18    }
19
20    public void setTipoAtendimento(String tipoAtendimento) {
21        this.tipoAtendimento = tipoAtendimento;
22    }
23
24    public String getConvenio() {
25        return convenio;
26    }
27
28    public void setConvenio(String convenio) {
29        this.convenio = convenio;
30    }
}
```

## APÊNDICE B - CODIGO FONTE DO PROGRAMA DE CAPTURA DE DADOS



E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar\_dimon/jwebcrawlers/entidades/Estabelecimento.java

```
1 package br.com.jucemar_dimon.jwebcrawlers.entidades;
2
3 import java.io.Serializable;
4 import java.util.ArrayList;
5
6 /**
7  * @author Jucemar
8 */
9 public class Estabelecimento implements Serializable {
10
11     private int cnes;
12
13     private String razaoSocial;
14     private String nomeFantasia;
15     private String logradouro;
16     private String numero;
17     private String complemento;
18     private String bairro;
19     private String cep;
20     private String municipio;
21     private String estado;
22     private String telefone;
23     private String tipoEstabelecimento;
24     private String codigoMunicipio;
25     private Double longitude;
26     private ArrayList<Profissional> profissionaisVinculados;
27     private ArrayList<ServicoClassificacao> servicosOferecidos;
28     private ArrayList<AtendimentoPrestado> atendimentosPrestados;
29
30     public Estabelecimento() {
31 }
```

## E:/Projetos Netbeans/JWebCrawlerCNES/src/com/jucemar\_dimon/jwebcrawlers/entidades/Estabelecimento.java

```
32     this.profissionaisVinculados = new ArrayList<Professional>();
33     this.servicosOferecidos = new ArrayList<ServiçoClassificado>();
34     this.atendimentosPrestados = new ArrayList<AtendimentoPrestado>();
35 }
36
37     public Estabelecimento(int cnes, String razaoSocial, String nomeFantasia, String nomeLogradouro, String numero,
38     String complemento, String bairro, String cep, String município, String codigoMunicípio, String estado, String telefone,
39     String tipoEstabelecimento, Double latitude, Double longitude) {
40         this.cnes = cnes;
41         this.razaoSocial = razaoSocial;
42         this.nomeFantasia = nomeFantasia;
43         this.logradouro = nomeLogradouro;
44         this.numero = numero;
45         this.complemento = complemento;
46         this.bairro = bairro;
47         this.cep = cep; this.municipio =
48         this.codigoMunicípio; this.estado = estado;
49         this.telefone = telefone;
50         this.tipoEstabelecimento = tipoEstabelecimento;
51         this.latitude = latitude;
52         this.longitude = longitude;
53     }
54     public String getCodigoMunicípio() {
55         return codigoMunicípio;
56     }
57
58     public void setCodigoMunicípio(String codigoMunicípio) {
59         this.codigoMunicípio = codigoMunicípio;
60 }
```

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar\_dimon/jwebcrawlers/entidades/Estabelecimento.java

```
61      }
62
63     public ArrayList<ServicoClassificacao> getServicosOferecidos() {
64         return servicosOferecidos;
65     }
66
67     public void setServicosOferecidos(ArrayList<ServicoClassificacao> ServicosOferecidos) {
68         this.servicosOferecidos = ServicosOferecidos;
69     }
70
71     public ArrayList<Profissional> getProfissionaisVinculados() {
72         return profissionaisVinculados;
73     }
74
75     public void setProfissionaisVinculados(ArrayList<Profissional> profissionaisVinculados) {
76         this.profissionaisVinculados = profissionaisVinculados;
77     }
78
79     public int getCnes() {
80         return cnes;
81     }
82
83     public void setCnes(int cnes) {
84         this.cnes = cnes;
85     }
86
87     public String getLogradouro() {
88         return logradouro;
89     }
90
91     public void setLogradouro(String nomeLogradouro) {
```

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar\_dimon/jwebcrawlers/entidades/Estabelecimento.java

```
92     this.logradouro = nomeLogradouro;  
93 }  
94  
95 public String getEstado() {  
96     return estado;  
97 }  
98  
99 public void setEstado(String estado) {  
100     this.estado = estado;  
101 }  
102  
103 public Double getLatitude() {  
104     return latitude;  
105 }  
106  
107 public void setLatitude(Double latitude) {  
108     this.latitude = latitude;  
109 }  
110  
111 public Double getLongitude() {  
112     return longitude;  
113 }  
114  
115 public void setLongitude(Double longitude) {  
116     this.longitude = longitude;  
117 }  
118  
119 public String getRazaoSocial() {  
120     return razaoSocial;  
121 }  
--
```

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar\_dimon/jwebcrawlers/entidades/Estabelecimento.java

```
123  public void setRazaoSocial(String razaoSocial) {  
124      this.razaoSocial = razaoSocial;  
125  }  
126  
127  public String getNomeFantasia() {  
128      return nomeFantasia;  
129  }  
130  
131  public void setNomeFantasia(String nomeFantasia) {  
132      this.nomeFantasia = nomeFantasia;  
133  }  
134  
135  public String getNumero() {  
136      return numero;  
137  }  
138  
139  public void setNumero(String numero) {  
140      this.numero = numero;  
141  }  
142  
143  public String getComplemento() {  
144      return complemento;  
145  }  
146  
147  public void setComplemento(String complemento) {  
148      this.complemento = complemento;  
149  }  
150  
151  public String getBairro() {  
152      return bairro;  
153  }
```

```
154     public void setBairro(String bairro) {  
155         this.bairro = bairro;  
156     }  
157  
158     public String getCep() {  
159         return cep;  
160     }  
161  
162     public void setCep(String cep) {  
163         this.cep = cep;  
164     }  
165  
166     public String getMunicipio() {  
167         return municipio;  
168     }  
169  
170     public void setMunicipio(String municipio) {  
171         this.municipio = municipio;  
172     }  
173  
174     public String getTelefone() {  
175         return telefone;  
176     }  
177  
178     public void setTelefone(String telefone) {  
179         this.telefone = telefone;  
180     }  
181  
182     public String getTipoEstabelecimento() {  
183         return tipoEstabelecimento;  
184     }
```

```
185      }
186
187  public void setTipoEstabelecimento(String tipoEstabelecimento) {
188      this.tipoEstabelecimento = tipoEstabelecimento;
189  }
190
191  public ArrayList<AtendimentoPrestado> getAtendimentosPrestados() {
192      return atendimentosPrestados;
193  }
194
195  public void setAtendimentosPrestados(ArrayList<AtendimentoPrestado> atendimentosPrestados) {
196      this.atendimentosPrestados = atendimentosPrestados;
197  }
198
199 }
```

```
1 package br.com.jucemar_dimon.jwebcrawlers.entidades;
2
3 import java.io.Serializable;
4
5 /**
6 * @author Jucemar
7 */
8 public class EstabelecimentoResumido implements Serializable {
9
10    private String nome;
11    private String cnes;
12    private String codigoMunicipio;
13    private String idConsulta;
14    private String linkDaPagina;
15
16    public EstabelecimentoResumido(String nome, String cnes, String codigoMunicipio, String linkDaPagina) {
17        this.nome = nome;
18        this.cnes = cnes;
19        this.codigoMunicipio = codigoMunicipio;
20        this.idConsulta = codigoMunicipio + cnes;
21        this.linkDaPagina = linkDaPagina;
22    }
23
24    public String getCodigoMunicipio() {
25        return codigoMunicipio;
26    }
27
28    public void setCodigoMunicipio(String codigoMunicipio) {
29        this.codigoMunicipio = codigoMunicipio;
30    }
31}
```

```
32     public String getIdConsultaO {
33         return idConsulta;
34     }
35
36     public void setIdConsulta(String idConsulta) {
37         this.idConsulta = idConsulta;
38     }
39
40     public String getNameO {
41         return nome;
42     }
43
44     public void setNameO(String nome) {
45         this.nome = nome;
46     }
47
48     public String getCnesO {
49         return cnes;
50     }
51
52     public void setCnes(String cnes) {
53         this.cnes = cnes;
54     }
55
56     public String getLinkDaPaginaO {
57         return linkDaPagina;
58     }
59
60     public void setLinkDaPagina(String linkDaPagina) {
61         this.linkDaPagina = linkDaPagina;
62     }
63 }
```

63  
64 }

## E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar\_dimon/jwebcrawlers/entidades/Municipio.java

```
1 package br.com.jucemar_dimon.jwebcrawlers.entidades;
2
3 import java.io.Serializable;
4 import java.util.ArrayList;
5
6 /**
7 * @author Jucemar
8 */
9 public class Municipio implements Serializable {
10
11     private String codigo;
12     private String nome;
13     private int qtdeEstabelecimentos;      String
14     private LinkDaPagina;    ArrayList<Estabelecimento>;
15     private estabelecimentos;
16
17     public Municipio(String codigo, String nome, int qtdeEstabelecimentos, String linkDaPagina) {
18         this.codigo = codigo;
19         this.nome = nome;
20         this.qtdeEstabelecimentos = qtdeEstabelecimentos;
21         this.linkDaPagina = linkDaPagina;
22         estabelecimentos = null;
23     }
24
25     public ArrayList<Estabelecimento> getEstabelecimentos() {
26         return estabelecimentos;
27     }
28
29     public void setEstabelecimentos(ArrayList<Estabelecimento> estabelecimentos) {
30         this.estabelecimentos = estabelecimentos;
31     }
32 }
```

```
32
33     public String getCodigoO {
34         return codigo;
35     }
36
37     public void setCodigo(String codigo) {
38         this.codigo = codigo;
39     }
40
41     public String getNomeO {
42         return nome;
43     }
44
45     public void setNome(String nome) {
46         this.nome = nome;
47     }
48
49     public int getQtdEstabelecimentosO {
50         return qtdEstabelecimentos;
51     }
52
53     public void setQtdEstabelecimentos(int qtdEstabelecimentos) {
54         this.qtdEstabelecimentos = qtdEstabelecimentos;
55     }
56
57     public String getLinkDaPaginaO {
58         return linkDaPagina;
59     }
60
61     public void setLinkDaPagina(String linkDaPagina) {
62         this.linkDaPagina = linkDaPagina;
63     }
```

```
63      }
64
65 }
```

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar\_dimon/jwebcrawlers/entidades/Profissional.java

```
1 package br.com.jucemar_dimon.jwebcrawlers.entidades;
2
3 import java.io.Serializable;
4
5 /**
6 * @author Jucemar
7 */
8 public class Profissional implements Serializable {
9
10    private String nome;
11    private String cbo;
12    private String cbоДescricao;
13
14    public Profissional() {
15    }
16
17    public String getName() {
18        return nome;
19    }
20
21    public void setName(String nome) {
22        this.nome = nome;
23    }
24
25    public String getCbo() {
26        return cbo;
27    }
28
29    public void setCbo(String cbo) {
30        this.cbo = cbo;
31    }
32}
```

```
32     public String getCboDescricaoO {  
33         return cboDescricao;  
34     }  
35  
36     public void setCboDescricao(String cboDescricao) {  
37         this.cboDescricao = cboDescricao;  
38     }  
39 }  
40 }  
41 }
```

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar\_dimon/jwebcrawlercnes/entidades/ServicoClassificacao.java

```
1 package br.com.jucemar_dimon.jwebcrawlercnes.entidades;
2
3 import java.io.Serializable;
4
5 /**
6 * @author Jucemar
7 */
8 public class ServicoClassificacao implements Serializable {
9
10    private int servico;
11    private int classificacao;
12
13    public ServicoClassificacao(int servico, int classificacao) {
14        this.servico = servico;
15        this.classificacao = classificacao;
16    }
17
18    public ServicoClassificacao() {
19
20    }
21
22    public int getServico() {
23        return servico;
24    }
25
26    public void setServico(int servico) {
27        this.servico = servico;
28    }
29
30    public int getClassificacao() {
31        return classificacao;
32    }
33}
```

```
32      }
33
34     public void setClassificacao(int classificacao) {
35         this.classificacao = classificacao;
36     }
37
38 }
```

E:\Projetos Netbeans\JWebCrawlerCNES\src\br\com\jucemar\_dimon\jwebcrawlercnes\principal\AtendimentoPrestadoDAO.java

```

1 package br.com.jucemar_dimon_jwebcrawlernes.principal;
2
3 import br.com.jucemar_dimon_jwebcrawlernes.entidades.AtendimentoPrestado;
4 import java.sql.Connection;
5 import java.sql.DriverManager;
6 import java.sql.SQLException;
7 import java.sql.Statement;
8 import java.util.logging.Level;
9 import java.util.logging.Logger;
10
11 /**
12 * @author Jucemar
13 */
14 public class AtendimentoPrestadoDAO {
15
16     private Connection c = null;
17     private Statement stmt = null;
18
19     public AtendimentoPrestadoDAO() {
20
21         Class.forName("org.sqlite.JDBC");
22         c = DriverManager.getConnection("jdbc:sqlite:estabelecimentos.");
23
24         System.out.println("Conexão com o banco de dados estabelecida com sucesso!");
25
26         try {
27
28             String tabelaAtendimento = "CREATE TABLE IF NOT EXISTS atendimentos (
29                     + "id INTEGER PRIMARY KEY AUTOINCREMENT,"
30                     + "atendimento TEXT NOT NULL,"
31                     + "convenio TEXT NOT NULL,"
32                     + "data DATE NOT NULL,"
33                     + "valor DECIMAL(10,2) NOT NULL,"
34                     + "observacao TEXT NOT NULL,"
35                     + "status TEXT NOT NULL,"
36                     + "criado_em DATE NOT NULL,"
37                     + "atualizado_em DATE NOT NULL,"
38                     + "deletado_em DATE NOT NULL,"
39                     + "ativo BOOLEAN NOT NULL"
40                 )";
41
42             stmt.executeUpdate(tabelaAtendimento);
43
44         } catch (SQLException ex) {
45             Logger.getLogger(AtendimentoPrestadoDAO.class.getName()).log(Level.SEVERE, null, ex);
46         }
47     }
48
49     public void inserir(AtendimentoPrestado atendimento) {
50
51         try {
52             String sql = "INSERT INTO atendimentos (id, atendimento, convenio, data, valor, observacao, status, criado_em, atualizado_em, deletado_em, ativo) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)";
53
54             PreparedStatement ps = c.prepareStatement(sql);
55
56             ps.setInt(1, atendimento.getId());
57             ps.setString(2, atendimento.getAtendimento());
58             ps.setString(3, atendimento.getConvenio());
59             ps.setDate(4, atendimento.getData());
60             ps.setDouble(5, atendimento.getValor());
61             ps.setString(6, atendimento.getObservacao());
62             ps.setString(7, atendimento.getStatus());
63             ps.setDate(8, atendimento.getCriadoEm());
64             ps.setDate(9, atendimento.getAtualizadoEm());
65             ps.setDate(10, atendimento.getDeletadoEm());
66             ps.setBoolean(11, atendimento.isAtivo());
67
68             ps.executeUpdate();
69
70         } catch (SQLException ex) {
71             Logger.getLogger(AtendimentoPrestadoDAO.class.getName()).log(Level.SEVERE, null, ex);
72         }
73     }
74
75     public void alterar(AtendimentoPrestado atendimento) {
76
77         try {
78             String sql = "UPDATE atendimentos SET atendimento = ?, convenio = ?, data = ?, valor = ?, observacao = ?, status = ?, criado_em = ?, atualizado_em = ?, deletado_em = ?, ativo = ? WHERE id = ?";
79
80             PreparedStatement ps = c.prepareStatement(sql);
81
82             ps.setString(1, atendimento.getAtendimento());
83             ps.setString(2, atendimento.getConvenio());
84             ps.setDate(3, atendimento.getData());
85             ps.setDouble(4, atendimento.getValor());
86             ps.setString(5, atendimento.getObservacao());
87             ps.setString(6, atendimento.getStatus());
88             ps.setDate(7, atendimento.getCriadoEm());
89             ps.setDate(8, atendimento.getAtualizadoEm());
90             ps.setDate(9, atendimento.getDeletadoEm());
91             ps.setBoolean(10, atendimento.isAtivo());
92             ps.setInt(11, atendimento.getId());
93
94             ps.executeUpdate();
95
96         } catch (SQLException ex) {
97             Logger.getLogger(AtendimentoPrestadoDAO.class.getName()).log(Level.SEVERE, null, ex);
98         }
99     }
100
101    public void excluir(AtendimentoPrestado atendimento) {
102
103        try {
104            String sql = "DELETE FROM atendimentos WHERE id = ?";
105
106            PreparedStatement ps = c.prepareStatement(sql);
107
108            ps.setInt(1, atendimento.getId());
109
110            ps.executeUpdate();
111
112        } catch (SQLException ex) {
113            Logger.getLogger(AtendimentoPrestadoDAO.class.getName()).log(Level.SEVERE, null, ex);
114        }
115    }
116
117    public List<AtendimentoPrestado> listar() {
118
119        List<AtendimentoPrestado> lista = new ArrayList<AtendimentoPrestado>();
120
121        try {
122            String sql = "SELECT * FROM atendimentos";
123
124            PreparedStatement ps = c.prepareStatement(sql);
125
126            ResultSet rs = ps.executeQuery();
127
128            while (rs.next()) {
129                AtendimentoPrestado atendimento = new AtendimentoPrestado();
130
131                atendimento.setId(rs.getInt("id"));
132                atendimento.setAtendimento(rs.getString("atendimento"));
133                atendimento.setConvenio(rs.getString("convenio"));
134                atendimento.setData(rs.getDate("data"));
135                atendimento.setValor(rs.getDouble("valor"));
136                atendimento.setObservacao(rs.getString("observacao"));
137                atendimento.setStatus(rs.getString("status"));
138                atendimento.setCriadoEm(rs.getDate("criado_em"));
139                atendimento.setAtualizadoEm(rs.getDate("atualizado_em"));
140                atendimento.setDeletadoEm(rs.getDate("deletado_em"));
141                atendimento.setAtivo(rs.getBoolean("ativo"));
142
143                lista.add(atendimento);
144            }
145
146        } catch (SQLException ex) {
147            Logger.getLogger(AtendimentoPrestadoDAO.class.getName()).log(Level.SEVERE, null, ex);
148        }
149
150        return lista;
151    }
152
153    public AtendimentoPrestado buscarPorId(int id) {
154
155        AtendimentoPrestado atendimento = null;
156
157        try {
158            String sql = "SELECT * FROM atendimentos WHERE id = ?";
159
160            PreparedStatement ps = c.prepareStatement(sql);
161
162            ps.setInt(1, id);
163
164            ResultSet rs = ps.executeQuery();
165
166            if (rs.next()) {
167                atendimento = new AtendimentoPrestado();
168
169                atendimento.setId(rs.getInt("id"));
170                atendimento.setAtendimento(rs.getString("atendimento"));
171                atendimento.setConvenio(rs.getString("convenio"));
172                atendimento.setData(rs.getDate("data"));
173                atendimento.setValor(rs.getDouble("valor"));
174                atendimento.setObservacao(rs.getString("observacao"));
175                atendimento.setStatus(rs.getString("status"));
176                atendimento.setCriadoEm(rs.getDate("criado_em"));
177                atendimento.setAtualizadoEm(rs.getDate("atualizado_em"));
178                atendimento.setDeletadoEm(rs.getDate("deletado_em"));
179                atendimento.setAtivo(rs.getBoolean("ativo"));
180
181            }
182
183        } catch (SQLException ex) {
184            Logger.getLogger(AtendimentoPrestadoDAO.class.getName()).log(Level.SEVERE, null, ex);
185        }
186
187        return atendimento;
188    }
189
190    public void fecharConexao() {
191
192        try {
193            c.close();
194        } catch (SQLException ex) {
195            Logger.getLogger(AtendimentoPrestadoDAO.class.getName()).log(Level.SEVERE, null, ex);
196        }
197    }
198}
```

E:/Projetos Netbeans/JWebCrawlerCNES/src/com/jucemar\_dimon/jwebcrawlercnes/principal/AtendimentoPrestadoDAO.java

```
32          + " establecimiento_cnes INTEGER NOT NULL,"  
33          + " FOREIGN KEY(estabelecimento_cnes)"  
34          + " REFERENCES estabelecimento(cnes));";  
35  
36      String index1 = "CREATE INDEX IF NOT EXISTS atendimento_estabelecimento_PKIndex ON  
atendimento_estabelecimento(id);";  
37      String index2 = "CREATE INDEX IF NOT EXISTS atendimento_estabelecimento_FKIndex ON  
atendimento_estabelecimento (estabelecimento_cnes);";  
38  
39      stmt = c.createStatement();  
40      stmt.executeUpdate(tabelaAtendimento);  
41      stmt.executeUpdate(index1);  
42      stmt.executeUpdate(index2);  
43      stmt.close();  
44      c.close();  
45      catch (Exception e) {  
46          System.out.println(e.getClassName().getName() + ":" + e.getMessage());  
47      }  
48      System.out.println("Tabela de Atendimento_prestado criada com sucesso");  
49      } catch (ClassNotFoundException ex) {  
50          Logger.getLogger(EstabelecimentoDAO.class.getName()).log(Level.SEVERE, null, ex);  
51      } catch (SQLException ex) {  
52          Logger.getLogger(EstabelecimentoDAO.class.getName()).log(Level.SEVERE, null, ex);  
53      }  
54  }  
55  public void inserirAtendimento(int cnesEstabelecimento, AtendimentoPrestado atendimento) {  
56  
57      try {  
58          c = DriverManager.getConnection("jdbc:sqlite:estabelecimentos.db");  
59          this.stmt = this.c.createStatement();  
60      }
```

```
61
62     try {
63
64         String sql = "INSERT INTO atendimento_estabelecimento(" +
65             "atendimento, convenio, estabelecimento_cnes) " +
66             "VALUES(" +
67             "atendimento.getTipoAtendimento() + " , " +
68             "atendimento.getConvenio() + " , " +
69             "cnesEstabelecimento +
70             "); ";
71         System.out.println(sql);
72         this.stmt.executeUpdate(sql);
73
74     } catch (SQLException e) {
75         e.printStackTrace();
76     }
77     stmt.close();
78     c.close();
79     } catch (SQLException ex) {
80         Logger.getLogger(EstabelecimentoDAO.class.getName()).log(Level.SEVERE, null, ex);
81     }
82
83     }
84 }
```

E:/Projetos Netbeans/JWebCrawlerCNES/src/com/jucemar\_dimon/principal/ConversorHtmlParaJava.java

```
1 package br.com.jucemar_dimon.principal;
2
3 import br.com.jucemar_dimon.jwebcrawlerses.entidades.AtendimentoPrestado;
4 import br.com.jucemar_dimon.jwebcrawlerses.entidades.Estabelecimento;
5 import br.com.jucemar_dimon.jwebcrawlerses.entidades.EstabelecimentoResumido;
6 import br.com.jucemar_dimon.jwebcrawlerses.entidades.Municipio;
7 import br.com.jucemar_dimon.jwebcrawlerses.entidades.Profissional;
8 import br.com.jucemar_dimon.jwebcrawlerses.entidades.ServicoClassificacao;
9 import java.io.IOException;
10 import java.util.ArrayList;
11 import java.util.logging.Level;
12 import java.util.logging.Logger;
13 import org.jsoup.Connection;
14 import org.jsoup.Connection.Response;
15 import org.jsoup.Jsoup;
16 import org.jsoup.nodes.Document;
17 import org.jsoup.select.Elements;
18 /**
19 * Classe responsável pela extração dos dados das páginas html do portal do
20 * CNESNet
21 *
22 * @author Jucemar
23 * @version 1.0
24 */
25 public class ConversorHtmlParaJava {
26
27     private static final String FUNCAO_PROFESSIONAIS_ESTABELECIMENTO = "Mod_Profissional.asp?VCo_Unidade=";
28     private static final String FUNCAO_SERVICO_CLASSIFICACAO = "Mod_Conj_Informacoes.asp?VCo_Unidade=";
29     private static final String FUNCAO_ATENDIMENTOS_PRESTADOS = "Mod_Bas_Atendimento.asp?VCo_Unidade=";
30     private static final String URL_PRINCIPAL = "http://cnes2.datasus.gov.br/";
```

```

32     private static final String FUNCAO_ESTABELECIMENTOS_MUNICIPIOS =
33         "Lista_Tot_Es_Municipio.asp?Estado=42&NomeEstado=SANTA%20CATARINA";
34     private String googleApiKey = "AIzaSyB4EXy3SiTgmWrGtBiHHSz8rQmT GhCO-Yg";
35     private ArrayList<String> estabelecimentosFisicos;
36     int estabelecimentosValidos = 0;
37     int coordLocalizadasOSM = 0;
38     int coordLocalizadasGM = 0;
39
39     /**
40      * Simples construtor padrão
41     */
42     public ConversorHtmlParaJava() {
43         estabelecimentosFisicos = new
44             DE ORGAOS ESTADUAL";
45         estabelecimentosFisicos.add("CENTRAL DE ATENCAO HEMOTERAPIA E OU HEMATOLOGICA");
46         estabelecimentosFisicos.add("CENTRO DE APOIO A SAUDE DA FAMILIA");
47         estabelecimentosFisicos.add("CENTRO DE ATENCAO PSICOSSOCIAL");
48         estabelecimentosFisicos.add("CENTRO DE PARTE NORMAL - ISOLADO");
49         estabelecimentosFisicos.add("CENTRO DE SAUDE/UNIDADE BASICA");
50         estabelecimentosFisicos.add("CLINICA/CENTRO DE ESPECIALIDADE");
51         estabelecimentosFisicos.add("CONSULTORIO ISOLADO");
52         estabelecimentosFisicos.add("HOSPITAL ESPECIALIZADO");
53         estabelecimentosFisicos.add("HOSPITAL GERAL");
54         estabelecimentosFisicos.add("HOSPITAL/DIA - ISOLADO");
55         estabelecimentosFisicos.add("LABORATORIO CENTRAL DE SAUDE PUBLICA LACEN");
56         estabelecimentosFisicos.add("LABORATORIO DE SAUDE PUBLICA");
57         estabelecimentosFisicos.add("POLICLINICA");
58         estabelecimentosFisicos.add("POSTO DE SAUDE");
59         estabelecimentosFisicos.add("PRONTO ATENDIMENTO");
60         estabelecimentosFisicos.add("PRONTO SOCORRO ESPECIALIZADO");
61         estabelecimentosFisicos.add("PRONTO SOCORRO GERAL");
62     }

```

E:/Projetos Netbeans/JWebCrawlerCNES/src/com/jucemar\_dimon/jwebcrawlercnes/principal/ConversorHtmlParaJava.java

```
62     estabelecimentosFisicos.add("UNIDADE DE APOIO DIAGNOSE E TERAPIA (SADT ISOLADO');");
63
64 }
65
66 /**
67 * Método que retorna uma Collection do tipo ArrayList que armazena dados
68 * dos municípios como Código IBGE, Nome, Quantidades de estabelecimentos que
69 * possuem e uma String que representa uma url para a página web onde é
70 * possível visualizar os estabelecimentos instalados no município.
71 *
72 * @return ArrayList com objetos do tipo Municipio
73 */
74 public ArrayList<Municipio> getMunicipios() {
75     int statusCode = -1;
76     int qtdeTentativasRequisicaoM = 0;
77     int timeOut = 1000;
78     Response respostaServidor = null;
79     ArrayList<Municipio> resultado = new ArrayList<Municipio>();
80     int qtdeTentativasRequisicao = 0;
81     boolean fim = false;
82     while (fim == false) {
83         Document paginaHtmlMunicipios = null;
84         try {
85             respostaServidor = (Response) Jsoup
86                 .connect(URL_PRINCIPAL + FUNCAO_ESTABELECIMENTOS_MUNICIPIOS)
87                 .ignoreContentType(true)
88                 .userAgent("Mozilla/5.0 (Windows NT 6.1; WOW64; rv:40.0) Gecko/20100101 Firefox/40.1")
89                 .referrer("http://www.google.com")
90                 .timeout(timeOut)
91                 .followRedirects(true)
92                 .execute();
93         } catch (IOException e) {
94             e.printStackTrace();
95         }
96         if (respostaServidor != null) {
97             String conteudo = respostaServidor.body();
98             if (conteudo != null) {
99                 Document doc = Jsoup.parse(conteudo);
100                ...
101            }
102        }
103    }
104 }
```

E:/Projetos Netbeans/JWebCrawlerCNES/src(br/com/jucemar\_dimon/jwebcrawlercnes/principal/ConversorHtmlParaJava.java

```

93     statusCode = respostaServidor.statusCode;
94     if (statusCode == 200) {
95         timeout = 500; flag = true;
96         qtdTentativasRequisicao = 1;
97         System.out.println("Conexão estabelecida com a página de municípios do estado
98                         de Santa
99                         Catarina.");
100    paginaHtmlMunicípios = respostaServidor.parse();
101    Elements tables = paginaHtmlMunicípios.getElementsByTag("table");
102    if (tables.size() < 8) {
103        System.out.println("Erro - página de Municípios de Santa Catarina fora do ar. Uma nova
tentativa será realizada");
104    } else {
105        Elements trs = tables.get(4).getElementsByTag("tr");
106        for (int i = 0; i < trs.size(); i++) {
107            Elements tds = trs.get(i).getElementsByTag("td");
108            if (tds.size() < 3) {
109                qtdTentativasRequisicao++;
110                System.err.println("Erro - QTDE de TDS: " + trs.get(i).select("td").size());
111                System.err.println(trs.get(i).select("td"));
112                System.err.println("Erro - Problemas ao carregar município - Linha: " + i);
113                System.err.println("Erro - Tentativa " + qtdTentativasRequisicao);
114                i--;
115            if (qtdTentativasRequisicao > 5) {
116                System.err.println("Impossível ler a Linha: " + i + ", ela será descartada.");
117                qtdTentativasRequisicao = 0;
118                i++;
119            }
120        } else {
121            resultado.add(new Município(

```

E:/Projetos Netbeans/JWebCrawlerCNES/src/com/jucemar\_dimon/jwebcrawlercnes/principal/ConversorHtmlParaJava.java

```
122     tds.get(0).textO,
123     tds.get(1).textO,
124     Integer.parseInt(tds.get(2).textO),
125     URL_PRINCIPAL + tds.get(1).getElementsByTag("a").attr("href")));
126   }
127 }
128 }
129 }
130 } catch (IOException ex) {
131     qtdeTentativasRequisicao++;
132     System.err.println("-----");
133     System.err.println("StatusCode : " + statusCode);
134     System.err.println("Erro - tentativa com erro " + qtdeTentativasRequisicao + " - Tempo limite de
135 requisicao atingido.");
136     System.err.println("Não foi possível conectar a página de municípios do Estado de Santa Catarina.");
137     if (qtdeTentativasRequisicao > 4) {
138         qtdeTentativasRequisicao = 0;
139         timeOut += 100;
140         System.err.println("Atenção! Verifique sua conexão com a internet");
141         System.err.println("Time out aumentado para " + timeOut);
142         System.err.println("Aguardando 5s até a próxima tentativa.");
143         try {
144             Thread.sleep(5000);
145         } catch (InterruptedException ex1) {
146             Logger.getLogger(ConversorHtmlParaJava.class.getName()).log(Level.SEVERE, null, ex1);
147         }
148     }
149 }
150 }
```

```
152     }
153     return resultado;
154 }
155 /**
156 * Método que retorna uma Collection do tipo ArrayList de objetos do tipo
157 * EstabelecimentoResumido que contém informações resumidas dos
158 * estabelecimentos do município
159 *
160 * @param município objeto do tipo Municipio com os dados resumidos sobre os
161 * estabelecimentos do município
162 * @return ArrayList de objetos do tipo EstabelecimentoResumido
163 */
164 public ArrayList<EstabelecimentoResumido> getEstabelecimentosDoMunicipio(Municipio municipio) {
165     String url = municipio.getLinkDaPagina();
166     ArrayList<EstabelecimentoResumido> resultado = new ArrayList<EstabelecimentoResumido>();
167     int statusCode = -1;
168     boolean fim = false;
169     Response respostaServidor = null;
170     int qtdTentativasRequisicao = 0;
171     int qtdTentativasRequisicaoE = 0;
172     int timeOut = 1000;
173     while (fim == false) {
174         Document paginaHtmlEstabelecimentosMunicipio = null;
175         try {
176             respostaServidor = Jsoup
177                 .connect(url)
178                 .ignoreContentType(true)
179                 .userAgent("Mozilla/5.0 (Windows NT 6.1; WOW64; rv:40.0) Gecko/20100101 Firefox/40.1")
180                 .referrer("http://www.google.com")
181                 .timeout(timeOut)
182         } catch (IOException e) {
183             e.printStackTrace();
184             if (qtdTentativasRequisicao < qtdTentativasRequisicaoE) {
185                 qtdTentativasRequisicao++;
186             } else {
187                 fim = true;
188             }
189         }
190         if (!fim) {
191             resultado.add(new EstabelecimentoResumido(paginaHtmlEstabelecimentosMunicipio));
192             qtdTentativasRequisicaoE++;
193         }
194     }
195     return resultado;
196 }
```

### E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar\_dimon/jwebcrawlercnes/principal/ConversorHtmlParaJava.java

```
183                                         .followRedirects(true)
184
185                                         .executeO;
186
187                                         statusCode = respostaServidor.statusCodeO;
188                                         if (statusCode == 200) {
189                                         timeOut = 500;
190                                         qtdTentativasRequisicao = 1;
191                                         fim = true;
192                                         System.out.println("Conexão estabelecida com a página de estabelecimentos do município de: " +
municipio.getNameO);
193                                         paginaHtmlEstabelecimentosMunicipio = respostaServidor.parseO;
194                                         Elements tables = paginaHtmlEstabelecimentosMunicipio.getElementsByTag("table");
195                                         if (tables.sizeO < 6) {
196                                         fim = false;
197                                         System.out.println("Erro - página de estabelecimentos do município de " + municipio.getNameO)
+ " fora do ar. Uma nova tentativa será realizada");
198                                         } else {
199                                         Elements trs = tables.get(4).getElementsByTag("tr");
200                                         for (int i = 0; i < trs.sizeO; i++) {
201                                         Elements tds = trs.get(i).getElementsByTag("td");
202                                         if (tds.sizeO < 4) {
203                                         qtdeTentativasRequisicaoE++;
204                                         System.out.println("Erro - QTDE de TDS: " + tds.sizeO);
205                                         municipio " + municipio.getNameO);
206                                         System.out.println("Erro - Tentativa " + qtdeTentativasRequisicaoE);
207                                         i--;
208                                         if (qtdeTentativasRequisicaoE > 5) {
209                                         System.out.println("Impossível ler a linha: " + i + ", ela será descartada.");
210                                         qtdeTentativasRequisicaoE = 0;
211                                         i++;
212                                         }
213                                         }
214                                         }
215                                         }
216                                         }
217                                         }
218                                         }
219                                         }
220                                         }
221                                         }
222                                         }
223                                         }
224                                         }
225                                         }
226                                         }
227                                         }
228                                         }
229                                         }
230                                         }
231                                         }
232                                         }
233                                         }
234                                         }
235                                         }
236                                         }
237                                         }
238                                         }
239                                         }
240                                         }
241                                         }
242                                         }
243                                         }
244                                         }
245                                         }
246                                         }
247                                         }
248                                         }
249                                         }
250                                         }
251                                         }
252                                         }
253                                         }
254                                         }
255                                         }
256                                         }
257                                         }
258                                         }
259                                         }
260                                         }
261                                         }
262                                         }
263                                         }
264                                         }
265                                         }
266                                         }
267                                         }
268                                         }
269                                         }
270                                         }
271                                         }
272                                         }
273                                         }
274                                         }
275                                         }
276                                         }
277                                         }
278                                         }
279                                         }
280                                         }
281                                         }
282                                         }
283                                         }
284                                         }
285                                         }
286                                         }
287                                         }
288                                         }
289                                         }
290                                         }
291                                         }
292                                         }
293                                         }
294                                         }
295                                         }
296                                         }
297                                         }
298                                         }
299                                         }
300                                         }
301                                         }
302                                         }
303                                         }
304                                         }
305                                         }
306                                         }
307                                         }
308                                         }
309                                         }
310                                         }
311                                         }
312                                         }
313                                         }
314                                         }
315                                         }
316                                         }
317                                         }
318                                         }
319                                         }
320                                         }
321                                         }
322                                         }
323                                         }
324                                         }
325                                         }
326                                         }
327                                         }
328                                         }
329                                         }
330                                         }
331                                         }
332                                         }
333                                         }
334                                         }
335                                         }
336                                         }
337                                         }
338                                         }
339                                         }
340                                         }
341                                         }
342                                         }
343                                         }
344                                         }
345                                         }
346                                         }
347                                         }
348                                         }
349                                         }
350                                         }
351                                         }
352                                         }
353                                         }
354                                         }
355                                         }
356                                         }
357                                         }
358                                         }
359                                         }
360                                         }
361                                         }
362                                         }
363                                         }
364                                         }
365                                         }
366                                         }
367                                         }
368                                         }
369                                         }
370                                         }
371                                         }
372                                         }
373                                         }
374                                         }
375                                         }
376                                         }
377                                         }
378                                         }
379                                         }
380                                         }
381                                         }
382                                         }
383                                         }
384                                         }
385                                         }
386                                         }
387                                         }
388                                         }
389                                         }
390                                         }
391                                         }
392                                         }
393                                         }
394                                         }
395                                         }
396                                         }
397                                         }
398                                         }
399                                         }
400                                         }
401                                         }
402                                         }
403                                         }
404                                         }
405                                         }
406                                         }
407                                         }
408                                         }
409                                         }
410                                         }
411                                         }
412                                         }
413                                         }
414                                         }
415                                         }
416                                         }
417                                         }
418                                         }
419                                         }
420                                         }
421                                         }
422                                         }
423                                         }
424                                         }
425                                         }
426                                         }
427                                         }
428                                         }
429                                         }
430                                         }
431                                         }
432                                         }
433                                         }
434                                         }
435                                         }
436                                         }
437                                         }
438                                         }
439                                         }
440                                         }
441                                         }
442                                         }
443                                         }
444                                         }
445                                         }
446                                         }
447                                         }
448                                         }
449                                         }
450                                         }
451                                         }
452                                         }
453                                         }
454                                         }
455                                         }
456                                         }
457                                         }
458                                         }
459                                         }
460                                         }
461                                         }
462                                         }
463                                         }
464                                         }
465                                         }
466                                         }
467                                         }
468                                         }
469                                         }
470                                         }
471                                         }
472                                         }
473                                         }
474                                         }
475                                         }
476                                         }
477                                         }
478                                         }
479                                         }
480                                         }
481                                         }
482                                         }
483                                         }
484                                         }
485                                         }
486                                         }
487                                         }
488                                         }
489                                         }
490                                         }
491                                         }
492                                         }
493                                         }
494                                         }
495                                         }
496                                         }
497                                         }
498                                         }
499                                         }
500                                         }
501                                         }
502                                         }
503                                         }
504                                         }
505                                         }
506                                         }
507                                         }
508                                         }
509                                         }
510                                         }
511                                         }
512                                         }
513                                         }
514                                         }
515                                         }
516                                         }
517                                         }
518                                         }
519                                         }
520                                         }
521                                         }
522                                         }
523                                         }
524                                         }
525                                         }
526                                         }
527                                         }
528                                         }
529                                         }
530                                         }
531                                         }
532                                         }
533                                         }
534                                         }
535                                         }
536                                         }
537                                         }
538                                         }
539                                         }
540                                         }
541                                         }
542                                         }
543                                         }
544                                         }
545                                         }
546                                         }
547                                         }
548                                         }
549                                         }
550                                         }
551                                         }
552                                         }
553                                         }
554                                         }
555                                         }
556                                         }
557                                         }
558                                         }
559                                         }
560                                         }
561                                         }
562                                         }
563                                         }
564                                         }
565                                         }
566                                         }
567                                         }
568                                         }
569                                         }
570                                         }
571                                         }
572                                         }
573                                         }
574                                         }
575                                         }
576                                         }
577                                         }
578                                         }
579                                         }
580                                         }
581                                         }
582                                         }
583                                         }
584                                         }
585                                         }
586                                         }
587                                         }
588                                         }
589                                         }
590                                         }
591                                         }
592                                         }
593                                         }
594                                         }
595                                         }
596                                         }
597                                         }
598                                         }
599                                         }
600                                         }
601                                         }
602                                         }
603                                         }
604                                         }
605                                         }
606                                         }
607                                         }
608                                         }
609                                         }
610                                         }
611                                         }
612                                         }
613                                         }
614                                         }
615                                         }
616                                         }
617                                         }
618                                         }
619                                         }
620                                         }
621                                         }
622                                         }
623                                         }
624                                         }
625                                         }
626                                         }
627                                         }
628                                         }
629                                         }
630                                         }
631                                         }
632                                         }
633                                         }
634                                         }
635                                         }
636                                         }
637                                         }
638                                         }
639                                         }
640                                         }
641                                         }
642                                         }
643                                         }
644                                         }
645                                         }
646                                         }
647                                         }
648                                         }
649                                         }
650                                         }
651                                         }
652                                         }
653                                         }
654                                         }
655                                         }
656                                         }
657                                         }
658                                         }
659                                         }
660                                         }
661                                         }
662                                         }
663                                         }
664                                         }
665                                         }
666                                         }
667                                         }
668                                         }
669                                         }
670                                         }
671                                         }
672                                         }
673                                         }
674                                         }
675                                         }
676                                         }
677                                         }
678                                         }
679                                         }
680                                         }
681                                         }
682                                         }
683                                         }
684                                         }
685                                         }
686                                         }
687                                         }
688                                         }
689                                         }
690                                         }
691                                         }
692                                         }
693                                         }
694                                         }
695                                         }
696                                         }
697                                         }
698                                         }
699                                         }
700                                         }
701                                         }
702                                         }
703                                         }
704                                         }
705                                         }
706                                         }
707                                         }
708                                         }
709                                         }
710                                         }
711                                         }
712                                         }
713                                         }
714                                         }
715                                         }
716                                         }
717                                         }
718                                         }
719                                         }
720                                         }
721                                         }
722                                         }
723                                         }
724                                         }
725                                         }
726                                         }
727                                         }
728                                         }
729                                         }
730                                         }
731                                         }
732                                         }
733                                         }
734                                         }
735                                         }
736                                         }
737                                         }
738                                         }
739                                         }
740                                         }
741                                         }
742                                         }
743                                         }
744                                         }
745                                         }
746                                         }
747                                         }
748                                         }
749                                         }
750                                         }
751                                         }
752                                         }
753                                         }
754                                         }
755                                         }
756                                         }
757                                         }
758                                         }
759                                         }
760                                         }
761                                         }
762                                         }
763                                         }
764                                         }
765                                         }
766                                         }
767                                         }
768                                         }
769                                         }
770                                         }
771                                         }
772                                         }
773                                         }
774                                         }
775                                         }
776                                         }
777                                         }
778                                         }
779                                         }
779                                         }
780                                         }
781                                         }
782                                         }
783                                         }
784                                         }
785                                         }
786                                         }
787                                         }
788                                         }
789                                         }
789                                         }
790                                         }
791                                         }
792                                         }
793                                         }
794                                         }
795                                         }
796                                         }
797                                         }
798                                         }
799                                         }
799                                         }
800                                         }
801                                         }
802                                         }
803                                         }
804                                         }
805                                         }
806                                         }
807                                         }
808                                         }
809                                         }
809                                         }
810                                         }
811                                         }
812                                         }
813                                         }
814                                         }
815                                         }
816                                         }
817                                         }
818                                         }
819                                         }
819                                         }
820                                         }
821                                         }
822                                         }
823                                         }
824                                         }
825                                         }
826                                         }
827                                         }
828                                         }
829                                         }
829                                         }
830                                         }
831                                         }
832                                         }
833                                         }
834                                         }
835                                         }
836                                         }
837                                         }
838                                         }
839                                         }
839                                         }
840                                         }
841                                         }
842                                         }
843                                         }
844                                         }
845                                         }
846                                         }
847                                         }
848                                         }
849                                         }
849                                         }
850                                         }
851                                         }
852                                         }
853                                         }
854                                         }
855                                         }
856                                         }
857                                         }
858                                         }
859                                         }
859                                         }
860                                         }
861                                         }
862                                         }
863                                         }
864                                         }
865                                         }
866                                         }
867                                         }
868                                         }
869                                         }
869                                         }
870                                         }
871                                         }
872                                         }
873                                         }
874                                         }
875                                         }
876                                         }
877                                         }
878                                         }
879                                         }
879                                         }
880                                         }
881                                         }
882                                         }
883                                         }
884                                         }
885                                         }
886                                         }
887                                         }
888                                         }
889                                         }
889                                         }
890                                         }
891                                         }
892                                         }
893                                         }
894                                         }
895                                         }
896                                         }
897                                         }
898                                         }
899                                         }
899                                         }
900                                         }
901                                         }
902                                         }
903                                         }
904                                         }
905                                         }
906                                         }
907                                         }
908                                         }
909                                         }
909                                         }
910                                         }
911                                         }
912                                         }
913                                         }
914                                         }
915                                         }
916                                         }
917                                         }
918                                         }
919                                         }
919                                         }
920                                         }
921                                         }
922                                         }
923                                         }
924                                         }
925                                         }
926                                         }
927                                         }
928                                         }
928                                         }
929                                         }
930                                         }
931                                         }
932                                         }
933                                         }
934                                         }
935                                         }
936                                         }
937                                         }
938                                         }
939                                         }
939                                         }
940                                         }
941                                         }
942                                         }
943                                         }
944                                         }
945                                         }
946                                         }
947                                         }
948                                         }
949                                         }
949                                         }
950                                         }
951                                         }
952                                         }
953                                         }
954                                         }
955                                         }
956                                         }
957                                         }
958                                         }
959                                         }
959                                         }
960                                         }
961                                         }
962                                         }
963                                         }
964                                         }
965                                         }
966                                         }
967                                         }
968                                         }
969                                         }
969                                         }
970                                         }
971                                         }
972                                         }
973                                         }
974                                         }
975                                         }
976                                         }
977                                         }
978                                         }
978                                         }
979                                         }
979                                         }
980                                         }
981                                         }
982                                         }
983                                         }
984                                         }
985                                         }
986                                         }
987                                         }
988                                         }
988                                         }
989                                         }
989                                         }
990                                         }
991                                         }
992                                         }
993                                         }
994                                         }
995                                         }
996                                         }
997                                         }
998                                         }
999                                         }
999                                         }
1000                                         }
```

## E:/Projetos Netbeans/JWebCrawlerCNES/src/com/jucemar\_dimon/jwebcrawlercnes/principal/ConversorHtmlParaJava.java

```
211
212     }
213     } else {
214         resultado.add(new EstabelecimentoResumido(
215             tds.get(0).text(),
216             tds.get(1).text(),
217             municipio.getCodigo(),
218             URL_PRINCIPAL + tds.get(0).getElementsByTag("a").attr("href")));
219     }
220 }
221 }
222 } catch (IOException ex) {
223     qtdeTentativasRequisicao++;
224     System.out.println("-----");
225     System.out.println("StatusCode : " + statusCode);
226     System.out.println("Erro - tentativa com erro " + qtdeTentativasRequisicao + " - Tempo limite de
requisição atingido.");
227     System.out.println("Não foi possível conectar a página de estabelecimentos do município de: " +
municipio.getNome());
228     if (qtdeTentativasRequisicao > 4) {
229         qtdeTentativasRequisicao = 0;
230         timeOut += 100;
231         System.out.println("Atenção! Verifique sua conexão com a internet");
232         System.out.println("Time out aumentado para " + timeOut);
233         System.out.println("Aguardando 5s até a próxima tentativa.");
234         try {
235             Thread.sleep(5000);
236         } catch (InterruptedException ex1) {
237             System.out.println("Atenção! Algo deu errado e não será possível Aguardando 5s até a próxima
tentativa.");
238         }
239     }
240 }
```

```
239         }
240     }
241 }
242     return resultado;
243 }
244 /**
245 * Método que retorna uma Collection do tipo ArrayList de objetos do tipo
246 * Estabelecimento que contém informações detalhadas do estabelecimento
247 *
248 * @param estabelecimentos ArrayList de objetos do tipo
249 * EstabelecimentoResumido
250 * @return ArrayList de objetos do tipo Estabelecimento
251 */
252 public ArrayList<Estabelecimento> getEstabelecimentosDoMunicipioDetalhados(ArrayList<EstabelecimentoResumido>
estabelecimentos) {
253     ArrayList<Estabelecimento> resultado = new ArrayList<Estabelecimento>();
254     int qtdTentativasRequisicao = 0;
255     int timeOut = 1000;
256     for (int i = 0; i < estabelecimentos.size(); i++) {
257         int statusCode = -1;
258         Document paginaHtmlEstabelecimentoDetalhado = null;
259         Response respostaServidor = null;
260         try {
261             respostaServidor = (Response) Jsoup
262                 .connect(estabelecimentos.get(i).getLinkDaPagina())
263                 .ignoreContentType(true)
264                 .userAgent("Mozilla/5.0 (Windows NT 6.1; WOW64; rv:40.0) Gecko/20100101 Firefox/40.1")
265                 .referrer("http://www.google.com")
266                 .timeout(timeOut)
267                 .followRedirects(true)
268         } catch (IOException e) {
269             e.printStackTrace();
270         }
271         if (statusCode != 200) {
272             continue;
273         }
274         String conteudo = respostaServidor.html();
275         if (paginaHtmlEstabelecimentoDetalhado != null) {
276             paginaHtmlEstabelecimentoDetalhado.append(conteudo);
277         } else {
278             paginaHtmlEstabelecimentoDetalhado = Jsoup.parse(conteudo);
279         }
280         resultado.add(new EstabelecimentoResumido(paginaHtmlEstabelecimentoDetalhado));
281     }
282     return resultado;
283 }
```

```

269         .executeO;
270
271         statusCode = respostaServidor.statusCodeO;
272         System.out.println("Iteração: " + i);
273         System.out.println("Estabelecimento: " + estabelecimentos.get(i).getNomeO);
274         System.out.println("Estabelecimento: " + estabelecimentos.get(i).getLinkDaPaginaO);
275         System.out.println("StatusCode : " + statusCode);
276
277         if (statusCode == 200) {
278             timeOut = 500; qtdTentativasRequisicao = 0;
279             paginaHtmlEstabelecimentoDetalhado = respostaServidor.parseO;
280             Elements tables = paginaHtmlEstabelecimentoDetalhado.getElementsByTag("table");
281             System.out.println("Tamanho da tabela: " + tables.sizeO);
282             Elements trs = null;
283             String tipoEstab = null;
284             boolean incluirEstabelecimento = true;
285
286             if (tables.sizeO < 7) {
287                 int iVelho = i;
288                 if (i == iVelho) {
289                     System.out.println("-----");
290                     trs = tables.get(0).getElementsByTag("tr");
291                     System.out.println(trs.get(1).textO);
292                     System.out.println("Erro iteracão " + i + " tentativa " + qtdTentativasRequisicao + " - A");
293                     System.out.println("A página do estabelecimento " + estabelecimentos.get(i).getNomeO + " "
294                     System.out.println("-----");
295                     if (qtdTentativasRequisicao > 5) {
296                         System.out.println("Atenção! Aguardando 5s até a próxima tentativa.");
297                     try {

```

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar\_dimon/jwebcrawlercnes/principal/ConversorHtmlParaJava.java

```
298         Thread.sleep(5000);
299     }
300     catch (InterruptedException ex1) {
301     }
302     i--;
303   } else {
304     if (tables.size() > 7) {
305       trs = tables.get(5).getElementsByTag("tr");
306       tipoEstab = trs.get(9).getElementsByTag("td").get(0).text();
307       isEstablecimientoFisico(tipoEstab); System.out.println("Tipo establecimiento: " +
308       tipoEstab); System.out.println(isEstablecimientoFisico(tipoEstab));
309       'Establecimiento válido' :
310     "Establecimiento inválido");
311     if (incluirEstablecimiento == false) {
312       System.out.println("-----");
313     } else {
314       trs = tables.get(4).getElementsByTag("tr");
315       tipoEstab = trs.get(9).getElementsByTag("td").get(0).text();
316       isEstablecimientoFisico(tipoEstab); System.out.println("Tipo establecimiento: " +
317       tipoEstab); System.out.println(isEstablecimientoFisico(tipoEstab));
318       'Establecimiento válido' :
319     "Establecimiento inválido");
320     if (incluirEstablecimiento == false) {
321       System.out.println("-----");
322     } else {
323       if (incluirEstablecimiento) {
324         estabelecimentosValidos++;
325         Establecimiento establecimiento = new Establecimiento();
326       }
327     }
328   }
329 }
```

## E:/Projetos Netbeans/JWebCrawlerCNES/src/com/jucemar\_dimon/jwebcrawlercnes/principal/ConversorHtmlParaJava.java

```
327     estabelecimento.setCnes(Integer.parseInt(estabelecimento.get(i).getCnes()));
328     String nomeFantasia = trs.get(1).getElementsByTagName("td").get(0).getText();
329     estabelecimento.setNomeFantasia(nomeFantasia.length() == 0 ? "-" : nomeFantasia);
330     String razaoSocial = trs.get(3).getElementsByTagName("td").get(0).getText();
331     estabelecimento.setRazaoSocial(razaoSocial.length() == 0 ? "-" : razaoSocial); String
332     nomeLogradouro = trs.get(5).getElementsByTagName("td").get(0).getText();
333     estabelecimento.setLogradouro(nomeLogradouro.length() == 0 ? "-" : nomeLogradouro);
334     String numero = trs.get(5).getElementsByTagName("td").get(1).getText();
335     estabelecimento.setNumero(numero.length() == 0 ? "S/N" : numero);
336     String telefone = trs.get(5).getElementsByTagName("td").get(2).getText();
337     estabelecimento.setTelefone(telefone.length() == 0 ? "-" : telefone);
338     String complemento = trs.get(7).getElementsByTagName("td").get(0).getText();
339     estabelecimento.setComplemento(complemento.length() == 0 ? "-" : complemento);
340     String bairro = trs.get(7).getElementsByTagName("td").get(1).getText();
341     estabelecimento.setBairro(bairro.length() == 0 ? "-" : bairro);
342     String cep = trs.get(7).getElementsByTagName("td").get(2).getText();
343     estabelecimento.setCep(cep.length() == 0 ? "-" : cep);
344     String municipio = trs.get(7).getElementsByTagName("td").get(3).getText(); municipio =
345     ImparNomeMunicipio(municipio); estabelecimento.setMunicipio(municipio.length() ==
346     0 ? "-" : municipio);
347     estabelecimento.setCodigoMunicipio(estabelecimento.get(i).getCodigoMunicipio());
348     String estado = trs.get(7).getElementsByTagName("td").get(4).getText();
349     estabelecimento.setEstado(estado.length() == 0 ? "-" : estado);
350     String tipoEstabelecimento = trs.get(9).getElementsByTagName("td").get(0).getText();
351     tipoEstabelecimento,
352     System.out.println("Detalamentos _____");
353     System.out.println("Cnes: " + estabelecimento.getCnes()); System.out.println("Razão
354     social: " + estabelecimento.getRazaoSocial()); System.out.println("Nome fantasia: " +
355     estabelecimento.getNomeFantasia()); System.out.println("Tipo estabelecimento: " +
estabelecimento.getTipoEstabelecimento());
```

## E:/Projetos Netbeans/JWebCrawlerCNES/src/com/jucemar\_dimon/jwebcrawlercnes/principal/ConversorHtmlParaJava.java

```
357     System.out.println("Logradouro: " + estabelecimento.getLogradouro());
358     System.out.println("Número: " + estabelecimento.getNumero());
359     System.out.println("Bairro: " + estabelecimento.getBairro());
360     System.out.println("Complemento: " + estabelecimento.getComplemento());
361     System.out.println("CEP: " + estabelecimento.getCep());
362     System.out.println("Cidade: " + estabelecimento.getMunicipio());
363     System.out.println("Telefone: " + estabelecimento.getTelefone());
364     System.out.println("Estado: " + estabelecimento.getEstado());
365
366     if (estabelecimentos.get(i).getLinkDaPagina().toString().length() == 117) {
367
368         estabelecimento.setProfissionaisVinculados(getProfissionaisDoEstabelecimento(estabelecimentos.getIdConsulta()));
369
370         estabelecimento.setServicosOferecidos(getServicosDoEstabelecimento(estabelecimentos.getIdConsulta()));
371
372         estabelecimento.setAtendimentosPrestados(getAtendimentosDoEstabelecimento(estabelecimentos.getIdConsulta()));
373
374         System.out.println("Estabelecimento desativado e com link bloqueado");
375
376         Double[] coordenadas = getGeocod(nomeLogradouro, numero, bairro, cep, municipio, estado);
377         estabelecimento.setLatitude(coordenadas[0]);
378         estabelecimento.setLongitude(coordenadas[1]);
379         resultado.add(estabelecimento);
380
381     } catch (IOException ex) {
382         int iVelho = i;
383         if (i == iVelho) {
384             qtdTentativasRequisicao++;
```

## E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar\_dimon/jwebcrawlercnes/principal/ConversorHtmlParaJava.java

```
385      }
386      System.err.println("-----");
387      System.err.println("StatusCode : " + statusCode);
388      System.err.println("Erro ao conectar a página de informações sobre o estabelecimento");
389      System.err.println("Erro iteração " + i + " tentativa com erro " + qtdeTentativasRequisicao + " - "
Tempo limite de requisição atingido.");
390      System.err.println("-----");
391      if (qtdeTentativasRequisicao > 5) {
392          timeOut += 100;
393          System.err.println("Timeout aumentado para: " + timeOut);
394          System.err.println("Atenção! Aguardando 5s até a próxima tentativa.");
395          try {
396              Thread.sleep(5000);
397          } catch (InterruptedException ex1) {
398              System.err.println("Atenção! Algo deu errado e não será possível Aguardando 5s até a próxima
tentativa.");
399          }
400      }
401      i--;
402  }
403  }
404  System.err.println("Estabelecimentos válidos: " + estabelecimentosValidos);
405  System.err.println("Coordenadas válidas OSM: " + coordLocalizadasOSM);
406  System.err.println("Coordenadas válidas GM: " + coordLocalizadasGM);
407  System.err.println("Estabelecimentos sem coordenadas geográficas: " + (estabelecimentosValidos -
(coordLocalizadasGM + coordLocalizadasOSM)));
408
estabelecimentosValidos = 0;
410  coordLocalizadasOSM = 0;
411  coordLocalizadasGM = 0;
412  return resultado;
```

```
413 }
414 /**
415 * Método interno que retorna um booleano que identifica se um
416 * estabelecimento de saúde é físico, fixo e presta algum tipo de
417 * atendimento pessoal e local
418 *
419 * @param String contendo o tipo de estabelecimento a ser analisado pelo
420 * método
421 * @return boolean que representa se o estabelecimento atende os parametros
422 * pre-definidos
423 */
424 private boolean isEstabelecimentoFisico(String tipo) {
425     boolean fim = false;
426     boolean resultado = false;
427     int i = 0;
428     while (!fim && i < estabelecimentosFisicos.size()) {
429         if (estabelecimentosFisicos.get(i).equalsIgnoreCase(tipo)) {
430             resultado = true;
431             fim = true;
432         }
433         i++;
434     }
435     return resultado;
436 }
437
438 private String isProfissionalValido(String cbo) {
439     String resultado = "";
440     if (cbo.length() < 7) {
441         System.out.println(cbo);
442     } else {
443     }
444 }
```

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar\_dimon/jwebcrawlercnes/principal/ConversorHtmlParaJava.java

```
444     String familiaCBO = cbo.trim().substring(0, 6);
445     if (familiaCBO.startsWith("2212")) {
446         || familiaCBO.startsWith("2231")
447         || familiaCBO.startsWith("2232")
448         || familiaCBO.startsWith("2236")
449         || familiaCBO.startsWith("2237")
450         || familiaCBO.startsWith("2238")
451         || familiaCBO.startsWith("2239")
452         || familiaCBO.startsWith("2241")
453         || familiaCBO.startsWith("2251")
454         || familiaCBO.startsWith("2252")
455         || familiaCBO.startsWith("2253")
456         || familiaCBO.startsWith("2261")
457         || familiaCBO.startsWith("2263")
458         || familiaCBO.startsWith("2515"))
459     resultado = familiaCBO;
460 }
461     return resultado;
462 }
463 }
464 private String limparNomeMunicipio(String nomeSuje) {
465     String nomeLimpo = null;
466     int i = 0;
467     int idApostrofo = 0;
468     boolean ok = false;
469     while (!ok) {
470         if (nomeSuje.charAt(i) == '·') {
471             ok = true;
472             nomeLimpo = nomeSuje.substring(0, i - 1);
473         }
474     }
475 }
```

```

475      i++;
476    }
477    idApostrofo = nomeLimpo.indexOf("•");
478    if (idApostrofo > 0) {
479      nomeLimpo = nomeLimpo.replace("•", " ");
480    }
481    return nomeLimpo;
482  }

483  private ArrayList<Professional> getProfissionaisDoEstabelecimento(String idConsulta) {
484    int qtdeTentativasRequisicao = 1;
485    int qtdeTentativasRequisicaoP = 1;
486    int statusCode = -1;
487    boolean fim = false;
488    int timeOut = 1000;
489    ArrayList<Professional> resultado = new ArrayList<Professional>();
490    Document paginaHtmlProfissionais = null;
491    Response respostaServidor = null;
492    System.out.println("Conectando a página dos profissionais++++++");
493    while (fim == false) {
494      try {
495        respostaServidor = (Response) Jsoup
496          .connect(URL_PRINCIPAL + FUNCAO_PROFISSIONAIS_ESTABELECIMENTO + idConsulta)
497          .ignoreContentType(true)
498          .userAgent("Mozilla/5.0 (Windows NT 6.1; WOW64; rv:40.0) Gecko/20100101 Firefox/40.1")
499          .referrer("http://www.google.com")
500          .timeout(timeOut)
501          .followRedirects(true)
502          .execute();
503        statusCode = respostaServidor.statusCode();
504        System.out.println("StatusCode : " + statusCode);
505      }

```

E:/Projetos Netbeans/JWebCrawlerCNES/src/com/jucemar\_dimon/jwebcrawlercnes/principal/ConversorHtmlParaJava.java

```
506     if (statusCode == 200) {  
507         timeOut = 500; qtdeTentativasRequisicao = 1;  
508         paginaHtmlProfissionais = respostaServidor.parseO;  
509         Elements trs = paginaHtmlProfissionais.select(".gradeA");  
510  
511         if (trs.sizeO == 0) {  
512             System.out.println("Não há profissionais cadastrados");  
513             fim = true;  
514         } else {  
515             Elements tds = trs.select("td");  
516             System.out.println("Tamanho da tabela de profissionais: " + tds.sizeO);  
517             if (paginaHtmlProfissionais.select("table").sizeO < 6) {  
518                 fim = false;  
519                 System.err.println("Erro - Site fora do ar. Uma nova tentativa será realizada");  
520             } else {  
521                 for (int i = 0; i < trs.sizeO; i++) {  
522                     qtdeTentativasRequisicaoP++;  
523                     if (trs.get(i).select("td").sizeO < 6) {  
524                         System.err.println("Erro - QTDE de TDS: " + trs.get(i).select("td").sizeO);  
525                         System.err.println(trs.get(i).select("td"));  
526                         System.err.println("Erro - Problemas ao carrega profissional - Linha: " + i);  
527                         System.err.println("Erro - Tentativa " + qtdeTentativasRequisicaoP);  
528                         i--;  
529                     if (qtdeTentativasRequisicaoP > 5) {  
530                         System.out.println("Impossível ler a linha: " + i + ", ela será descartada.");  
531                         qtdeTentativasRequisicaoP = 0;  
532                         i++;  
533                     } else {  
534                         String r1 = trs.get(i).select("td").get(5).textO;  
535                         if (r1.lengthO > 9) {  
536                         }  
537                     }  
538                 }  
539             }  
540         }  
541     }  
542 }
```

## E:/Projetos Netbeans/JWebCrawlerCNES/src/com/jucemar\_dimon/jwebcrawlercnes/principal/ConversorHtmlParaJava.java

```
537     String cbo = isProfissionalValido(r1);
538     String nome = trs.get(i).select("td").get(0).text();
539     String descCBO = r1.substring(9);
540     if (cbo.length() > 0) {
541         System.out.println("Linha: " + i + " Profissional: " + cbo + " | " +
542             nome);
543
544     Profissional p = new Profissional();
545     p.setCbo(cbo);
546     p.setNome(nome);
547     p.setCboDescricao(descCBO);
548     resultado.add(p);
549     } else {
550         System.err.println("Erro de leitura na linha: " + i + "ela será descartada");
551     }
552     }
553     fim = true;
554 }
555 }
556 }
557 } catch (IOException ex) {
558     qtdeTentativasRequisicao++;
559     System.out.println("StatusCode : " + statusCode);
560     System.err.println("-----");
561     System.err.println("Erro ao conectar a página de profissionais do estabelecimento");
562     System.err.println("Tentativa " + qtdeTentativasRequisicao + " - Tempo limite de requisição
563 atingido.");
564     System.out.println("-----");
565     if (qtdeTentativasRequisicao > 5) {
566         System.out.println("-----");
567     }
568 }
```

### E:/Projetos Netbeans/JWebCrawlerCNES/src/com/jucemar\_dimon/jwebcrawlercnes/principal/ConversorHtmlParaJava.java

```
566         timeOut += 100;
567         System.out.println("Atenção! Aguardando 5s até a próxima tentativa.");
568         System.out.println("Timeout aumentado para: " + timeOut);
569         qtdeTentativasRequisicao = 0;
570         try {
571             Thread.sleep(5000);
572         } catch (InterruptedException ex) {
573             System.out.println("Atenção! Algo deu errado e não será possível Aguardando 5s até a próxima
tentativa.");
574         }
575     }
576 }
577 }
578 return resultado;
579 }

580 private ArrayList<ServicoClassificacao> getServicosDoEstabelecimento(String idConsulta) {
581     int qtdeTentativasRequisicao = 1;
582     int qtdeTentativasRequisicaoS = 1;
583     int statusCode = -1;
584     boolean fim = false;
585     int timeOut = 1000;
586     ArrayList<ServicoClassificacao> resultado = new ArrayList<ServicoClassificacao>();
587     Document paginaHtmlServicos = null;
588     Response respostaServidor = null;
589     System.out.println("Conectando a página Serviços XXXXXXXXXXXXXXXXXX");
590     String servClass0Id = "";
591     while (fim == false) {
592         try {
593             respostaServidor = (Response) Jsoup
594                 .connect(URL_PRINCIPAL + FUNCAO_SERVICO_CLASSIFICACAO + idConsulta)
595         }
```



E:/Projetos Netbeans/JWebCrawlerCNES/src/com/jucemar\_dimon/jwebcrawlercnes/principal/ConversorHtmlParaJava.java

```
627     i--;
628     if (qtdTentativasRequisicoes > 5) {
629         System.out.println("Impossível ler a linha: " + i + ", ela será descartada.");
630         qtdTentativasRequisicoes = 0;
631         i++;
632     }
633 } else {
634     String servClas = trs.get(i).select("td").get(0).text();
635     if (servClassId.equals(ignoreCase(servClas))) {
636         System.out.println("Serviço e Classificação em duplicidade: " + servClas +
e será descartado");
637     } else {
638         ServicoClassificacao s = new ServicoClassificacao();
639         String descServ = trs.get(i).select("td").get(1).text();
640         String descClass = trs.get(i).select("td").get(2).text();
641         String serv = servClas.substring(0, 3);
642         String classif = servClas.substring(6, 9);
643         System.out.println("Serv: " + serv);
644         System.out.println("Class: " + classif);
645         System.out.println("Desc serv: " + descServ);
646         System.out.println("Desc class: " + descClass);
647         s.setServico(Integer.parseInt(serv));
648         s.setClassificacao(Integer.parseInt(classif));
649         resultado.add(s);
650     }
651     servClassId = servClas;
652 }
653     fim = true;
654 }
655 }
```

### E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar\_dimon/jwebcrawlercnes/principal/ConversorHtmlParaJava.java

```
657         }
658     } catch (IOException ex) {
659         qtdeTentativasRequisicao++;
660         System.out.println("-----");
661         System.out.println("StatusCode : " + statusCode);
662         System.out.println("Erro ao conectar a página de Serviços do estabelecimento");
663         System.out.println("Tentativa " + qtdeTentativasRequisicao + " - Tempo Limite de requisição
atingido.");
664         System.out.println("-----");
665         if (qtdeTentativasRequisicao > 5) {
666             timeOut += 100;
667             System.err.println("Atenção! Aguardando 5s até a próxima tentativa.");
668             System.err.println("Timeout aumentado para: " + timeOut);
669             qtdeTentativasRequisicao = 0;
670             try {
671                 Thread.sleep(5000);
672             } catch (InterruptedException ex1) {
673                 System.err.println("Atenção! Algo deu errado e não será possível Aguardando 5s até a próxima
tentativa.");
674             }
675         }
676     }
677 }
678     return resultado;
679 }
680
681 private ArrayList<AtendimentoPrestado> getAtendimentosDoEstabelecimento(String idConsulta) {
682     int qtdeTentativasRequisicao = 1;
683     int qtdeTentativasRequisicaoA = 1;
684     int statusCode = -1;
685     boolean fim = false;
```

E:/Projetos Netbeans/JWebCrawlerCNES/src/com/jucemar\_dimon/jwebcrawlercnes/principal/ConversorHtmlParaJava.java

```
686     int timeOut = 1000;
687     ArrayList<AtendimentoPrestado> resultado = new ArrayList<AtendimentoPrestado>();
688     Document paginaAtendimentosPrestados = null;
689     Response respostaServidor = null;
690     System.out.println("Conectando a página de Atendimentos prestados %%%%%%%%%%%%%%%");
691     while (fim == false) {
692         try {
693             respostaServidor = (Response) Jsoup
694                 .connect(URL_PRINCIPAL + FUNCAO_ATENDIMENTOS_PRESTADOS + idConsulta)
695                 .ignoreContentType(true)
696                 .userAgent("Mozilla/5.0 (Windows NT 6.1; WOW64; rv:40.0) Gecko/20100101 Firefox/40.1")
697                 .referrer("http://www.google.com")
698                 .timeout(timeOut)
699                 .followRedirects(true)
700                 .execute();
701             statusCode = respostaServidor.statusCode();
702             System.out.println("StatusCode : " + statusCode);
703             System.out.println("Link da consulta: " + URL_PRINCIPAL + FUNCAO_ATENDIMENTOS_PRESTADOS +
idConsulta);
704             if (statusCode == 200) {
705                 timeOut = 500;
706                 paginaAtendimentosPrestados = respostaServidor.parse();
707                 Elements tabelas = paginaAtendimentosPrestados.select("table");
708                 System.out.println("Qtde de tabelas: " + tabelas.size());
709                 if (tabelas.size() == 1) {
710                     fim = false;
711                     System.err.println("Erro - Site fora do ar. Uma nova tentativa será realizada");
712                 } else if (tabelas.size() == 3) {
713                     System.out.println("Não possui Tipos de atendimentos cadastrados.");
714                     fim = true;
715                 } else {
716                 }
717             }
718         }
719     }
720 }
```

## E:/Projetos Netbeans/JWebCrawlerCNES/src/com/jucemar\_dimon/jwebcrawlercnes/principal/ConversorHtmlParaJava.java

```
716      qtdeTentativasRequisicao = 1;
717      Elements trs = tabelas.get(5).select("tr");
718      for (int i = 0; i < trs.size(); i++) {
719          qtdeTentativasRequisicao++;
720          Elements tds = trs.get(i).select("td");
721          if (tds.size() < 2) {
722              System.err.println("Erro - QTDE de TDS: " + trs.get(i).size());
723              System.err.println(trs.get(i).select("td"));
724              System.err.println("Erro - Problemas ao carrega atendimento - Linha: " + i);
725              System.err.println("Erro - Tentativa " + qtdeTentativasRequisicaoA);
726              i--;
727              if (qtdeTentativasRequisicaoA > 5) {
728                  System.out.println("Impossível ler a linha: " + i + ", ela será descartada.");
729                  qtdeTentativasRequisicaoA = 0;
730                  i++;
731              }
732          } else {
733              AtendimentoPrestado a = new AtendimentoPrestado();
734              String tipoAtdm = tds.get(0).text();
735              String conven = tds.get(1).text();
736              a.setTipoAtendimento(tipoAtdm);
737              a.setConvenio(conven);
738              resultado.add(a);
739              System.out.println("Tipo de atendimento: " + tipoAtdm + " | Convênio: " + conven);
740          }
741          fim = true;
742      }
743  }
744  } catch (IOException ex) {
745      qtdeTentativasRequisicao++;
746  }
```

## E:/Projetos Netbeans/JWebCrawlerCNES/src/com/jucemar\_dimon/jwebcrawlercneshprincipal/ConversorHtmlParaJava.java

```
747     System.out.println("-----");
748     System.out.println("StatusCode : " + statusCode);
749     System.out.println("Erro ao conectar a página de Atendimentos prestados do estabelecimento");
750     System.out.println("Tentativa " + qtdeTentativasRequisicao + " - Tempo Limite de requisição
atingido.");
751     System.out.println("-----");
752     if (qtdeTentativasRequisicao > 5) {
753         timeOut += 100;
754         System.out.println("Atenção! Aguardando 5s até a próxima tentativa.");
755         System.out.println("Timeout aumentado para: " + timeOut);
756         qtdeTentativasRequisicao = 0;
757         try {
758             Thread.sleep(5000);
759         } catch (InterruptedException ex) {
760             System.out.println("Atenção! Algo deu errado e não será possível Aguardando 5s até a próxima
tentativa.");
761         }
762     }
763 }
764 }
765 return resultado;
766 }
767
768 private Double[] getGeocodGoogleMaps(String endereco) {
769     int qtdeTentativasRequisicao = 1;
770     int statusCode = -1;
771     boolean fim = false;
772     int timeOut = 1000;
773     Double[] resultado = new Double[2];
774     Document xmldoc = null;
775     Connection.Response respostaServidor = null;
```

## E:Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar\_dimon/jwebcrawlercnes/principal/ConversorHtmlParaJava.java

```
776 endereco = endereco.replace(" ", "+");
777 System.out.println(endereco);
778 while (fim == false) {
779     try {
780         respostaServidor = (Connection.Response) Jsoup
781             .connect("https://maps.googleapis.com/maps/api/geocode/xml?address=" + endereco + "&key=" +
782             googleApiKey)
783             .ignoreContentType(true)
784             .userAgent("Mozilla/5.0 (Windows NT 6.1; WOW64; rv:40.0) Gecko/20100101 Firefox/40.1")
785             .referrer("http://www.google.com")
786             .timeout(timeOut)
787             .followRedirects(true)
788             .execute();
789         statusCode = respostaServidor.statusCode();
790         if (statusCode == 200) {
791             timeOut = 500;
792             fim = true;
793             xmlDoc = respostaServidor.parse();
794             Elements latLon = xmlDoc.select("location");
795             System.out.println("Quantidade de resultados: " + latLon.size());
796             if (latLon.size() > 0) {
797                 if (latLon.size() > 1) {
798                     resultado[0] = Double.parseDouble(latLon.get(0).select("lat").text());
799                     resultado[1] = Double.parseDouble(latLon.get(0).select("lng").text());
800                 } else {
801                     resultado[0] = Double.parseDouble(latLon.select("lat").text());
802                     resultado[1] = Double.parseDouble(latLon.select("lng").text());
803                 }
804             }
805         }
806     }
807 }
```

### E:/Projetos Netbeans/JWebCrawlerCNES/src/com/jucemar\_dimon/jwebcrawlercneshprincipal/ConversorHtmlParaJava.java

```
806
807         }
808     } catch (IOException ex) {
809         qtdeTentativasRequisicao++;
810         System.out.println("-----");
811         System.out.println("StatusCode : " + statusCode);
812         System.out.println("Erro ao conectar a página de Geocoding do Google Mapas");
813         System.out.println("Tentativa " + qtdeTentativasRequisicao + " - Tempo Limite de requisição
atingido.");
814         if (qtdeTentativasRequisicao > 5) {
815             timeOut += 100;
816             System.err.println("Atenção! Aguardando 5s até a próxima tentativa.");
817             System.err.println("Timeout aumentado para: " + timeOut);
818             qtdeTentativasRequisicao = 0;
819             try {
820                 Thread.sleep(5000);
821             } catch (InterruptedException ex1) {
822                 System.err.println("Atenção! Algo deu errado e não será possível Aguardando 5s até a próxima
tentativa.");
823             }
824         }
825     }
826     }
827     return resultado;
828 }
829 }

830 private Double[] getGeocodOpenStreetMaps(String endereco) {
831     int qtdeTentativasRequisicao = 1;
832     int statusCode = -1;
833     boolean fim = false;
```

### E:/Projetos Netbeans/JWebCrawlerCNES/src/com/jucemar\_dimon/jwebcrawlercnes/principal/ConversorHtmlParaJava.java

```
835     int timeOut = 1000;
836     Double[] resultado = new Double[2];
837     Document xmldoc = null;
838     Connection.Response respostaServidor = null;
839     endereco = endereco.replace(" ", "%20");
840     System.out.println(endereco);
841     while (fim == false) {
842         try {
843             respostaServidor = (Connection.Response) Jsoup
844                 .connect("http://nominatim.openstreetmap.org/search/" + endereco.toLowerCase() +
845                     "?format=xml")
846                     .ignoreContentType(true)
847                     .userAgent("Mozilla/5.0 (Windows NT 6.1; WOW64; rv:40.0) Gecko/20100101 Firefox/40.1")
848                     .referrer("http://www.google.com")
849                     .timeout(timeout)
850                     .followRedirects(true)
851                     .execute();
852             statusCode = respostaServidor.statusCode;
853             if (statusCode == 200) {
854                 timeout = 1000;
855                 fim = true;
856                 xmldoc = respostaServidor.parse();
857                 latlon =
858                     System.out.println("Quantidade de resultados: " + latlon.size());
859                 if (latlon.size() > 0) {
860                     if (latlon.size() > 1) {
861                         resultado[0] = Double.parseDouble(latlon.get(0).attr("lat"));
862                         resultado[1] = Double.parseDouble(latlon.get(0).attr("lon"));
863                     } else {
864                         resultado[0] = Double.parseDouble(latlon.attr("lat"));
865                     }
866                 }
867             }
868         } catch (Exception e) {
869             e.printStackTrace();
870         }
871     }
872 }
```

## E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar\_dimon/jwebcrawlercnes/principal/ConversorHtmlParaJava.java

```
865         resultado[1] = Double.parseDouble(latLon.getAttribute("lon"));
866     }
867     System.out.println("GeocodOpenStreetMaps " + resultado[0] + " " + resultado[1]);
868 }
869 } catch (IOException ex) {
870     qtdeTentativasRequisicao++;
871     System.out.println("StatusCode : " + statusCode);
872     System.out.println("Erro ao conectar a página de Geocoding do Open Street Maps");
873     System.out.println("Tentativa " + qtdeTentativasRequisicao + " - Tempo limite de requisição
atingido.");
874     System.out.println("-----");
875     if (qtdeTentativasRequisicao > 5) {
876         timeOut = timeOut + 1000;
877         System.out.println("Atenção! Aguardando 5s até a próxima tentativa.");
878         System.out.println("Timeout aumentado para: " + timeOut);
879         qtdeTentativasRequisicao = 0;
880         try {
881             Thread.sleep(5000);
882         } catch (InterruptedException ex1) {
883             System.out.println("Atenção! Algo deu errado e não será possível Aguardando 5s até a próxima
tentativa.");
884         }
885     }
886     }
887     }
888 }
889 }
890 return resultado;
891 }
892 private String[] extraTipologadouro(String logradouro) {
```

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar\_dimon/jwebcrawlercnes/principal/ConversorHtmlParaJava.java

```
894     String[] resultado = new String[2];
895     boolean ok = false;
896     int i = 0;
897     while (!ok && i < logradouro.length()) {
898         if (logradouro.charAt(i) == '•') {
899             resultado[0] = logradouro.substring(0, i);//tipo de logradouro
900             resultado[1] = logradouro.substring(i + 1, logradouro.length());//nome do logradouro
901             ok = true;
902         }
903         i++;
904     }
905     if (resultado[0] == null) {
906         resultado[0] = "";
907         resultado[1] = logradouro.trim();
908     } else {
909         resultado[0].trim();
910         resultado[1].trim();
911     }
912     return resultado;
913 }
914     private Double[] getGeocod(String logradouro, String numero, String bairro, String cep, String municipio, String
915     estado) {
916         Double[] resultado = new Double[2];
917         String[] logradouroFormatado = extrairTipologradouro(logradouro);
918         String enderecoOSM = numero + " " + logradouroFormatado[0] + " , " + municipio
919         + " , " + estado + " , " + cep;
920         String enderecoGM = Logradouro + " " + numero + " " + bairro + " " + cep + " " + municipio + " " + estados;
921         resultado = getGeocodOpenStreetMaps(enderecoOSM);
922         if (resultado[0] != null) {
923             System.out.println("Origem das coordenadas: " + OpenStreetMaps");
```

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar\_dimon/jwebcrawlercnes/principal/ConversorHtmlParaJava.java

```
923     coordLocalizadasGMM++;
924 
925     } else {
926         resultado = getGeocodGoogleMaps(enderecoGM);
927         if (resultado[0] != null) {
928             System.out.println("Origem das coordenadas: " + GoogleMaps);
929             coordLocalizadasGMM++;
930             System.out.println("Coordenadas: " + resultado);
931         } else {
932             System.out.println("Origem das coordenadas: " + resultado);
933         }
934     }
935 }
936 }
```

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar\_dimon/jwebcrawlercnes/principal/EstabelecimentoDAO.java

```
1 package br.com.jucemar_dimon.jwebcrawlercnes.entity.Estabelecimento;
2
3 import br.com.jucemar_dimon.jwebcrawlercnes.entity.Estabelecimento;
4 import java.sql.Connection;
5 import java.sql.DriverManager;
6 import java.sql.ResultSet;
7 import java.sql.SQLException;
8 import java.sql.Statement;
9 import java.util.ArrayList;
10 import java.util.logging.Level;
11 import java.util.logging.Logger;
12 /**
13  * @author Jucemar
14  */
15 public class EstabelecimentoDAO {
16     private Connection c = null;
17     private Statement stmt = null;
18
19     public EstabelecimentoDAO {
20         try {
21             Class.forName("org.sqlite.JDBC");
22             c = DriverManager.getConnection("jdbc:sqlite:estabelecimentos.db");
23             System.out.println("Conexão com o banco de dados estabelecida com sucesso");
24         } catch (Exception e) {
25             e.printStackTrace();
26             String tabelaEstabelecimento = "CREATE TABLE IF NOT EXISTS estabelecimento "
27                     + "(cnes INTEGER NOT NULL,"
28                     + " razao_social TEXT,"
29                     + " nome_fantasia TEXT,"
30                     + " telefone TEXT,"
```

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar\_dimon/jwebcrawlercnes/principal/EstabelecimentoDAO.java

```
32          + " nome_logradouro TEXT NOT NULL,"  
33          + " numero TEXT,"  
34          + " complemento TEXT,"  
35          + " bairro TEXT,"  
36          + " municipio TEXT NOT NULL,"  
37          + " codigo_municipio TEXT NOT NULL,"  
38          + " estado TEXT NOT NULL,"  
39          + " cep TEXT NOT NULL,"  
40          + " latitude REAL,"  
41          + " longitude REAL,"  
42          + " tipo_estabelecimento TEXT NOT NULL,"  
43          + " PRIMARY KEY(cnes));";  
44 String index = "CREATE INDEX IF NOT EXISTS estabelecimento_PKIndex ON estabelecimento (cnes);";  
45 stmt = c.createStatement();  
46 stmt.executeUpdate(tabelaEstabelecimento);  
47 stmt.executeUpdate(index);  
48 stmt.close();  
49 c.close();  
50 } catch (Exception e) {  
51     System.err.println(e.getClass().getName() + ":" + e.getMessage());  
52 }  
53 System.out.println("Tabela de estabelecimentos criada com sucesso");  
54 } catch (ClassNotFoundException ex) {  
55     Logger.getLogger(EstabelecimentoDAO.class.getName()).log(Level.SEVERE, null, ex);  
56 } catch (SQLException ex) {  
57     Logger.getLogger(EstabelecimentoDAO.class.getName()).log(Level.SEVERE, null, ex);  
58 }  
59 }  
60 public boolean inserirEstabelecimento(Estabelecimento estabelecimento) {  
61     boolean novoEstabelecimento = isEstabelecimentoExiste(estabelecimento.getNome());  
62 }
```

### E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar\_dimon/jwebcrawlercnes/principal/EstabelecimentoDAO.java

```
63     if (novoEstabelecimento) {
64         try {
65             c = DriverManager.getConnection("jdbc:sqlite:estabelecimentos.db");
66             this.stmt = this.c.createStatement();
67             try {
68                 String sql = "INSERT INTO estabelecimento("
69                             + "cnes, razao_social, nome_fantasia, nome_logradouro, numero, complemento, bairro,
municipio, codigo_municipio, estado, telefone, cep, latitude, longitude, tipo_estabelecimento)"
70                             + "VALUES("
71                             + estabelecimento.getNomeO
72                             + estabelecimento.getRazaoSocialO.trimO + "•",
73                             + estabelecimento.getNomeFantasiaO.trimO + "•",
74                             + estabelecimento.getLogradouroO.trimO + "•",
75                             + estabelecimento.getNumeroO.trimO + "•",
76                             + estabelecimento.getComplementoO.trimO + "•",
77                             + estabelecimento.getBairroO.trimO + "•",
78                             + estabelecimento.getMunicipioO.trimO + "•",
79                             + estabelecimento.getCodigoMunicipioO.trimO + "•",
80                             + estabelecimento.getEstadoO.trimO + "•",
81                             + estabelecimento.getTelefoneO.trimO + "•",
82                             + estabelecimento.getCepO.trimO + "•",
83                             + estabelecimento.getLatitudeO + "•",
84                             + estabelecimento.getLongitudeO + "•",
85                             + estabelecimento.getTipoEstabelecimentoO.trimO
86                             + ");";
87             System.out.println(sql);
88             this.stmt.executeUpdate(sql);
89         } catch (SQLException e) {
90             e.printStackTrace();
91         }
92     }
```

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar\_dimon/jwebcrawlercnes/principal/EstabelecimentoDAO.java

```
93     stmt.close();
94     c.close();
95   } catch (SQLException ex) {
96     Logger.getLogger(EstabelecimentoDAO.class.getName()).log(Level.SEVERE, null, ex);
97   }
98   return novoEstabelecimento;
99 }
100 }

101 private boolean isEstabelecimentoExiste(int cnes) {
102   boolean novoEstabelecimento = true;
103   ResultSet rs = null;
104   try {
105     c = DriverManager.getConnection("jdbc:sqlite:estabelecimentos.db");
106     this.stmt = this.c.createStatement();
107     rs = this.stmt.executeQuery("SELECT cnas FROM estabelecimento where cnas=" + cnas + " ;");
108     if (rs.next()) {
109       novoEstabelecimento = false;
110     }
111     rs.close();
112     stmt.close();
113     c.close();
114   } catch (SQLException e) {
115   }
116   return novoEstabelecimento;
117 }
118
119 public int atualizarCoordenadasGeograficas(Double latitude, Double longitude, int cnas) {
120   int resultado = -1;
121   try {
122     c = DriverManager.getConnection("jdbc:sqlite:estabelecimentos.db");
123   }
```

### E:/Projetos Netbeans/JWebCrawlerCNES/src/com/jucemar\_dimon/jwebcrawlercnes/principal/EstabelecimentoDAO.java

```
124      this$stmt = this.c.createStatement();
125      resultado = this$stmt.executeUpdate("UPDATE estabelecimento set latitude=" + latitude + ", longitude=" +
longitude + " where cnes=" + cnes + ";" );
126      stmt.close();
127      c.close();
128  } catch (SQLException e) {
129      System.err.println("Não foi possível atualizar as coordenadas do estabelecimento: " + cnes + " " +
e.getMessage());
130  }
131      return resultado;
132  }
133
134  public static final Estabelecimento extrairDoCursor(ResultSet rs) throws SQLException {
135      int cnes = rs.getInt(br.com.jucemar_dimon.jwebcrawlercnes.principal.TabelaEstabelecimento.Colunas.CNES);
136      String razaoSocial =
rs.getString(br.com.jucemar_dimon.jwebcrawlercnes.principal.TabelaEstabelecimento.Colunas.RAZAO_SOCIAL);
137      String nomeFantasia =
rs.getString(br.com.jucemar_dimon.jwebcrawlercnes.principal.TabelaEstabelecimento.Colunas.NOME_FANTASIA);
138      String telefone =
rs.getString(br.com.jucemar_dimon.jwebcrawlercnes.principal.TabelaEstabelecimento.Colunas.TELEFONE);
139      String nomeLogradouro =
rs.getString(br.com.jucemar_dimon.jwebcrawlercnes.principal.TabelaEstabelecimento.Colunas.NOME_LOGRADOURO);
140      String numero =
rs.getString(br.com.jucemar_dimon.jwebcrawlercnes.principal.TabelaEstabelecimento.Colunas.NUMERO);
141      String complemento =
rs.getString(br.com.jucemar_dimon.jwebcrawlercnes.principal.TabelaEstabelecimento.Colunas.COMPLEMENTO);
142      String bairro =
rs.getString(br.com.jucemar_dimon.jwebcrawlercnes.principal.TabelaEstabelecimento.Colunas.BAIRRO);
143      String municipio =
rs.getString(br.com.jucemar_dimon.jwebcrawlercnes.principal.TabelaEstabelecimento.Colunas.MUNICIPIO);
144      String codigoMunicipio =
```

### E:\Projetos Netbeans\JWebCrawlerCNES/src/com/jucemar\_dimon/jwebcrawlerses/principal/EstabelecimentoDAO.java

```
rs.getString(br.com.jucemar_dimon.jwebcrawlerses.principal.TabelaEstabelecimento.CoLunas.CODIGO_MUNICIPIO);

145 String estado =
146 String cep = rs.getString(br.com.jucemar_dimon.jwebcrawlerses.principal.TabelaEstabelecimento.ESTADO);
147 Double latitude =
148 Double longitude =
149 String tipoEstabelecimento =
150 rs.getString(br.com.jucemar_dimon.jwebcrawlerses.principal.TabelaEstabelecimento.LATITUDE);
151 }

152 public ArrayList<Estabelecimento> listaTodosEstabelecimentos() {
153     ArrayList<Estabelecimento> resultado = new ArrayList<Estabelecimento>();
154     ResultSet rs = null;
155     try {
156         c = DriverManager.getConnection("jdbc:sqlite:estabelecimentos.db");
157         this.stmt = this.c.createStatement();
158         rs = this.stmt.executeQuery("SELECT * FROM estabelecimento;");
159         do {
160             Estabelecimento e = extrairDoCursor(rs);
161             resultado.add(e);
162         } while (rs.next());
163         rs.close();
164         stmt.close();
165         c.close();
166     } catch (SQLException e) {
167         System.out.println("Não foi possível realizar a consulta");
168     }
169 }
```

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar\_dimon/jwebcrawlers/principal/EstabelecimentoDAO.java

```
170     return resultado;
171 }
172
173 }
```

## E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar\_dimon/jwebcrawlercnes/principal/JWebCrawlerCNES.java

```
1 package br.com.jucemar_dimon.jwebcrawlercnes.principal;  
2  
3 import br.com.jucemar_dimon.jwebcrawlercnes.entidades.AtendimentoPrestado;  
4 import br.com.jucemar_dimon.jwebcrawlercnes.entidades.EstabElecimento;  
5 import br.com.jucemar_dimon.jwebcrawlercnes.entidades.EstabElecimentoResumido;  
6 import br.com.jucemar_dimon.jwebcrawlercnes.entidades.Municipio;  
7 import br.com.jucemar_dimon.jwebcrawlercnes.entidades.Profissional;  
8 import br.com.jucemar_dimon.jwebcrawlercnes.entidades.ServicoClassificacao;  
9 import br.com.jucemar_dimon.jwebcrawlercnes.sigtap.SigTap;  
10 import br.com.jucemar_dimon.jwebcrawlercnes.utilitarios.ManipuladorDeArquivos;  
11 import java.io.FileNotFoundException;  
12 import java.io.IOException;  
13 import java.util.ArrayList;  
14  
15 /**  
16 * @author Jucemar  
17 */  
18 public class JWebCrawlerCNES {  
19  
20     public static void main(String[] args) throws IOException {  
21         ManipuladorDeArquivos bkp = new ManipuladorDeArquivos();  
22         ConversorHtmlParaJava conversor = new ConversorHtmlParaJava();  
23         try {  
24             System.out.println("Iniciando Etapa de captura de dados do SigTap");  
25             SigTap.iniciarO();  
26             catch (FileNotFoundException eFNF) {  
27                 System.out.println(eFNF.getMessage());  
28                 System.out.println("Um dos arquivos de texto que contém uma das tabelas SIGTAP não foi localizado.");  
29                 System.out.println("Os arquivos são titulados de: tb_servico.txt e tb_servico_classificacao.txt");  
30                 System.exit(0);  
31         }  
32     }  
33 }
```

### E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar\_dimon/jwebcrawlercnes/principal/JWebCrawlerCNES.java

```
32     } catch (IOException e10) {
33         System.out.println(e10.getMessage());
34         System.out.println("Houve um problema durante a manipulação dos arquivos de texto que contém as tabelas
35         SIGTAP");
36         System.out.println("O programa será fechado");
37         System.exit(0);
38     }
39     System.out.println("Iniciando etapa de captura de dados dos Estabelecimentos");
40     ArrayList<Municipio> resultado = bkp.LeiaDadosDEArquivoTemporario();
41     if (resultado == null) {
42         resultado = conversor.getMunicipios();
43     }
44     EstabelecimentoDAO eDao = new EstabelecimentoDAO();
45     ServicoClassificacaoEstabelecimentoDAO sDao = new ServicoClassificacaoEstabelecimentoDAO();
46     ProfissionalDAO pDao = new ProfissionalDAO();
47     VinculoProfissionalEstabelecimentoDAO vDao = new VinculoProfissionalEstabelecimentoDAO();
48     AtendimentoPrestadoDAO aDao = new AtendimentoPrestadoDAO();
49     for (Municipio m : resultado) {
50         if (m.getEstabelecimentos() == null) {
51             System.out.println("Os estabelecimentos do município de " + m.getNome() + " ainda não foram salvos");
52             ArrayList<EstabelecimentoResumido> estabResumidos = conversor.getEstabelecimentoDoMunicipio(m);
53             conversor.getEstabelecimentoDoMunicipioDetalhados(estabResumidos);
54             m.setEstabelecimentos(estabResumidos);
55             bkp.salveDadosEmArquivoTemporario(resultado);
56             System.out.println("Os estabelecimentos do município de " + m.getNome() + " já foram salvos e serão
verificados sua existência na base de dados");
57             bkp.salveDadosEmArquivoTemporario(resultado);
58             for (Estabelecimento e : m.getEstabelecimentos()) {
59                 boolean novoEstabelecimento = eDao.inserirEstabelecimento(e);
```

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar\_dimon/jwebcrawlercnes/principal/JWebCrawlerCNES.java

```
60     if (novoEstabelecimento) {
61         System.out.println("0 estabelecimento " + e.getNomeFantasiaO + " será inserido na base de
dados");
62
63         for (Professional p : e.getProfissionaisVinculadosO) {
64             int id = pDao.inserirProfissional(p);
65             System.out.println("Id do profisional inserido: " + id);
66             vDao.inserirVinculoDoProfissional(id, e.getCnesO);
67
68         for (ServiçoClassificação s : e.getServicosOferecidosO) {
69             sDao.inserirServiçoClassificado(e.getCnesO, s);
70
71         for (AtendimentoPrestado a : e.getAtendimentosPrestadosO) {
72             aDao.inserirAtendimento(e.getCnesO, a);
73
74         } else {
75             System.out.println("0 estabelecimento " + e.getNomeFantasiaO + " já foi inserido na base de
dados");
76
77         }
78     }
79
80 }
```

```

E:/Projetos Netbeans/JWebCrawlerCNES/src(br/com/jucemar_dimon_jwebcrawlers/principal/ProfissionalDAO.java
1 package br.com.jucemar_dimon_jwebcrawlers.entidades.Profissional;
2
3 import br.com.jucemar_dimon_jwebcrawlers.entidades.Profissional;
4 import java.sql.Connection;
5 import java.sql.DriverManager;
6 import java.sql.ResultSet;
7 import java.sql.SQLException;
8 import java.sql.Statement;
9 import java.util.logging.Level;
10 import java.util.logging.Logger;
11
12 /**
13 * @author Jucemar
14 */
15 public class ProfissionalDAO {
16
17     private Connection c = null;
18     private Statement stmt = null;
19
20     public ProfissionalDAO() {
21         try {
22             Class.forName("org.sqlite.JDBC");
23             c = DriverManager.getConnection("jdbc:sqlite:estabeleciementos.db");
24             try {
25                 String tabelaProfissional = "CREATE TABLE IF NOT EXISTS profissional "
26                             + "(id INTEGER PRIMARY KEY AUTOINCREMENT,"
27                             + " nome TEXT NOT NULL,"
28                             + " cbo TEXT      NOT NULL,"
29                             + " cbo_descricao TEXT NOT NULL);";
30                 String index = "CREATE INDEX IF NOT EXISTS profissional_PKIndex ON profissional (id);";
31                 stmt = c.createStatement();
32             }
33         } catch (SQLException ex) {
34             Logger.getLogger(ProfissionalDAO.class.getName()).log(Level.SEVERE, null, ex);
35         }
36     }
37
38     public void inserir(Profissional p) {
39         try {
40             stmt.executeUpdate("INSERT INTO profissional (nome, cbo, cbo_descricao) VALUES ('" + p.getNome() + "','" + p.getCbo() + "','" + p.getCbo_descricao() + "');");
41         } catch (SQLException ex) {
42             Logger.getLogger(ProfissionalDAO.class.getName()).log(Level.SEVERE, null, ex);
43         }
44     }
45
46     public void alterar(Profissional p) {
47         try {
48             stmt.executeUpdate("UPDATE profissional SET nome = '" + p.getNome() + "', cbo = '" + p.getCbo() + "', cbo_descricao = '" + p.getCbo_descricao() + "' WHERE id = " + p.getId());
49         } catch (SQLException ex) {
50             Logger.getLogger(ProfissionalDAO.class.getName()).log(Level.SEVERE, null, ex);
51         }
52     }
53
54     public void excluir(Profissional p) {
55         try {
56             stmt.executeUpdate("DELETE FROM profissional WHERE id = " + p.getId());
57         } catch (SQLException ex) {
58             Logger.getLogger(ProfissionalDAO.class.getName()).log(Level.SEVERE, null, ex);
59         }
60     }
61
62     public Profissional buscarPorId(int id) {
63         Profissional p = null;
64         try {
65             ResultSet rs = stmt.executeQuery("SELECT * FROM profissional WHERE id = " + id);
66             if (rs.next()) {
67                 p = new Profissional(rs.getInt("id"), rs.getString("nome"), rs.getString("cbo"),
68                                     rs.getString("cbo_descricao"));
69             }
70         } catch (SQLException ex) {
71             Logger.getLogger(ProfissionalDAO.class.getName()).log(Level.SEVERE, null, ex);
72         }
73         return p;
74     }
75
76     public List<Profissional> buscarTodos() {
77         List<Profissional> lista = new ArrayList<Profissional>();
78         try {
79             ResultSet rs = stmt.executeQuery("SELECT * FROM profissional");
80             while (rs.next()) {
81                 lista.add(new Profissional(rs.getInt("id"), rs.getString("nome"), rs.getString("cbo"),
82                                         rs.getString("cbo_descricao")));
83             }
84         } catch (SQLException ex) {
85             Logger.getLogger(ProfissionalDAO.class.getName()).log(Level.SEVERE, null, ex);
86         }
87         return lista;
88     }
89
90     public void fecharConexao() {
91         try {
92             c.close();
93         } catch (SQLException ex) {
94             Logger.getLogger(ProfissionalDAO.class.getName()).log(Level.SEVERE, null, ex);
95         }
96     }
97 }

```

```

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar_dimon/jwebcrawlercnes/principal/ProfissionalDAO.java

32     stmt.executeUpdate(tabelaProfissional);
33
34     stmt.executeUpdate(index);
35     stmt.close();
36     c.close();
37 } catch (Exception e) {
38     System.err.println(e.getClass().getName() + ":" + e.getMessage());
39     System.exit(0);
40 } catch (ClassNotFoundException ex) {
41     Logger.getLogger(EstabelecimentoDAO.class.getName()).log(Level.SEVERE, null, ex);
42 } catch (SQLException ex) {
43     Logger.getLogger(EstabelecimentoDAO.class.getName()).log(Level.SEVERE, null, ex);
44 }
45
46 public int inserirProfissional(Profissional profissional) {
47     int ultimaId = 0;
48     try {
49         ResultSet rs = null;
50         ultimaId = isProfissionalExiste(profissional);
51         c = DriverManager.getConnection("jdbc:sqlite:estabelecimentos.db");
52         this.stmt = this.c.createStatement();
53         if (ultimaId == 0) {
54             System.out.println("Profissional não localizado na base de dados: " + profissional.getNome() + " será
55             atribuída uma id para ele");
56             String lastId = "select last_insert_rowid();";
57             String sql = "INSERT INTO profissional(
58                 + "nome, cbo, cbo_desricao"
59                 + "
60                 VALUES("
61

```

E:/Projetos Netbeans/JWebCrawlerCNES/src/com/jucemar\_dimon/jwebcrawlers/principal/ProfissionalDAO.java

```
62           + profissional.getChodescricaoO.trimO + "'); ";
63           System.out.println(sql);
64           this.stmt.executeUpdate(sql);
65           rs = this.stmt.executeQuery(lastId);
66           ultimaid = rs.getInt(1);
67           rs.close();
68       }
69       stmt.close();
70       c.close();
71       } catch (SQLException ex) {
72           Logger.getLogger(ProfissionalDAO.class.getName()).log(Level.SEVERE, null, ex);
73       }
74       return ultimaid;
75   }

76   private int isProfissionalExiste(Profissional profissional) {
77       int idprofissional = 0;
78       ResultSet rs = null;
79       try {
80           c = DriverManager.getConnection("jdbc:sqlite:estabelecimentos.db");
81           this.stmt = this.c.createStatement();
82           String buscaProfissional = "SELECT id FROM profissional WHERE nome='"
83           + profissional.getNomeO + "' AND ";
84           cbo = " + profissional.getChodescricaoO + " + ";
85           rs = this.stmt.executeQuery(buscaProfissional);
86           if (rs.nextO) {
87               idprofissional = rs.getInt("id");
88               System.out.println("Profissional encontrado na base de dados: "
89               + idprofissional);
90           }
91       } rs.close();
```

```
91     stmt.close();
92     c.close();
93   }
94   catch (SQLException ex)
95   {
96     Logger.getLogger(ProfissionalDAO.class.getName()).log(Level.SEVERE, null, ex);
97   }
98 }
99 }
```

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar\_dimon/jwebcrawlercnes/principal/ServicoClassificacaoEstabeleci

```
1 package br.com.jucemar_dimon.jwebcrawlercnes.principal;
```

```
2 
```

```
3 import br.com.jucemar_dimon.jwebcrawlercnes.entidades.ServicoClassificacao;
```

```
4 import java.sql.Connection;
```

```
5 import java.sql.DriverManager;
```

```
6 import java.sql.ResultSet;
```

```
7 import java.sql.SQLException;
```

```
8 import java.sql.Statement;
```

```
9 import java.util.logging.Level;
```

```
10 import java.util.logging.Logger;
```

```
11 
```

```
12 /**
13 * @author Jucemar
14 */
```

```
15 public class ServicoClassificacaoEstabelecimentoDAO {
```

```
16     private Connection c = null;
17     private Statement stmt = null;
18 
```

```
19     public ServicoClassificacaoEstabelecimentoDAO {
```

```
20         try {
21             Class.forName("org.sqlite.JDBC");
22             c = DriverManager.getConnection("jdbc:sqlite:estabelecimentos.db");
23         } catch {
24             String deleta = "DROP TABLE IF EXISTS servico_classificacao_estabelecimento";
25             String tabelaServicoClasificacao = "CREATE TABLE IF NOT EXISTS servico_classificacao_estabelecimento
26             "
27             + "(estabelecimento_cnes INTEGER NOT NULL,"
28             + " classificacao_id INTEGER NOT NULL,"
29             + " PRIMARY KEY(estabelecimento_cnes,classificacao_id),"
30             + " FOREIGN KEY(classificacao_id)"
```

## E:/Projetos Netbeans/JWebCrawlerCNES/src/com/jucemar\_dimon/jwebcrawlerses/principal/ServicoClassificacaoEstabeleci

```
31      + " REFERENCES classificacao(id),"
32      + " FOREIGN KEY(estabelecimento_cnes)"
33      + " REFERENCES estabelecimento(cnes));";
34
35  String index1 = "CREATE INDEX IF NOT EXISTS servico_classificacao_estabelecimento_FKIndex1 ON
servico_classificacao_estabelecimento (classificacao_id);";
36
37  String index2 = "CREATE INDEX IF NOT EXISTS servico_classificacao_estabelecimento_FKIndex2 ON
servico_classificacao_estabelecimento (estabelecimento_cnes);";
38
39  String index3 = "CREATE INDEX IF NOT EXISTS servico_classificacao_estabelecimento_PKIndex ON
servico_classificacao_estabelecimento (classificacao_id, estabelecimento_id, estabelecimento_cnes);";
40
41
42
43  stmt = c.createStatement();
44
45  stmt.executeUpdate(tabelaServicoClassificacao);
46
47  stmt.executeUpdate(index1);
48
49  stmt.executeUpdate(index2);
50
51  stmt.executeUpdate(index3);
52
53  stmt.close();
54
55  c.close();
56
57  } catch (Exception e) {
58
59      System.out.println(e.getClass().getName() + ":" + e.getMessage());
60
61  }
62
63  } catch (ClassNotFoundException ex) {
64
65      Logger.getLogger(EstabelecimentoDAO.class.getName()).log(Level.SEVERE, null, ex);
66
67  }
68
69  catch (SQLException ex) {
70
71      Logger.getLogger(EstabelecimentoDAO.class.getName()).log(Level.SEVERE, null, ex);
72
73  }
74
75  public void inserirServicoClassificacao(int cnesEstabelecimento, ServicoClassificacao servClass) {
76
77      int idClassificacao = getIdClassificacao(servClass);
78
79      if (idClassificacao != 0) {
80
81          try {
82
83              c = DriverManager.getConnection("jdbc:sqlite:estabelecimentos.db");
84
85          }
86
87      }
88
89  }
```

## E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar\_dimon/jwebcrawlercnes/principal/ServicoClassificacaoEstabeleci

```
59     this.smt = this.c.createStatement();
60     String sql = "INSERT INTO servico_classificacao_estabelecimento("
61         + "estabelecimento_cnes, classificacao_id"
62         + "VALUES('"
63         + cnesEstabelecimento + "','"
64         + idClassificacao + "')"; ;
65     System.out.println(sql);
66     this.smt.executeUpdate(sql);
67     stmt.close();
68     c.close();
69     } catch (SQLException ex) {
70     Logger.getLogger(EstabelecimentoDAO.class.getName()).log(Level.SEVERE, null, ex);
71     }
72     } else {
73     System.out.println("O serviço: " + servClass.getServico() + " classificação: " +
74 servClass.getClassificacao() + " foi descontinuado pelo MS");
75     }
76     private int getIdClassificacao(ServicoClassificacao servClass) {
77     System.out.println("Tentando localizar o Serviço: " + servClass.getServico() + " Classificação: " +
78 servClass.getClassificacao());
79     int resultado = 0;
80     ResultSet rs = null;
81     int idServiço = getIdServiço(servClass.getServico());
82     try {
83     c = DriverManager.getConnection("jdbc:sqlite:estabelecimentos.db");
84     this.smt = this.c.createStatement();
85     String idClassificacao = "SELECT id FROM classificacao WHERE servico_id=" + idServiço + " and código=" +
86     servClass.getClassificacao() + ";";
87     rs = this.smt.executeQuery(idClassificacao);
```

## E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar\_dimon/jwebcrawlercnes/principal/ServicoClassificacaoEstabeleci

```
87     if (rs.next()) {
88         resultado = rs.getInt("id");
89     } rs.close();
90     stmt.close();
91     c.close();
92     } catch (SQLException ex) {
93         Logger.getLogger(EstabelecimentoDAO.class.getName()).log(Level.SEVERE,
94                                         null, ex);
95     }
96     return resultado;
97 }
98 private int getIdServico(int servico) {
99     int resultado = 0;
100    ResultSet rs = null;
101    try {
102        c = DriverManager.getConnection("jdbc:sqlite:estabelecimentos.db");
103        this.stmt = this.c.createStatement();
104        String idServico = "SELECT id FROM servico WHERE codigo=" + servico + ";";
105        rs = this.stmt.executeQuery(idServico);
106        if (rs.next()) {
107            resultado = rs.getInt("id");
108        }
109        System.out.println("Codigo do servico: " + servico + " codigo do id: " + resultado);
110        rs.close();
111        stmt.close();
112        c.close();
113    } catch (SQLException ex) {
114        Logger.getLogger(EstabelecimentoDAO.class.getName()).log(Level.SEVERE, null, ex);
115    }
116 }
117 }
```

```
118      }
119
120 }
```

```

1 package br.com.jucemar.dimon.jwebcrawlerces.principal;
2
3 /**
4 * @author Jucemar
5 */
6 public final class TabelaEstabelecimento {
7
8     public static final String NOME_TABELA = "estabelecimento";
9
10    public static final class Colunas {
11
12        public static final String CNES = "cnes";
13        public static final String RAZAO_SOCIAL = "razao_social";
14        public static final String NOME_FANTASIA = "nome_fantasia";
15        public static final String TELEFONE = "telefone";
16        public static final String NOME_LOGRADOURO = "nome_logradouro";
17        public static final String NUMERO = "numero";
18        public static final String COMPLEMENTO = "complemento";
19        public static final String BAIRRO = "bairro";
20        public static final String MUNICIPIO = "municipio";
21        public static final String CODIGO_MUNICIPIO = "codigo_municipio";
22        public static final String ESTADO = "estado";
23        public static final String CEP = "cep";
24        public static final String LATITUDE = "latitude";
25        public static final String LONGITUDE = "longitude";
26        public static final String TIPO_ESTABELECIMENTO = "tipo_estabelecimento";
27
28    }
}

```

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar\_dimon/jwebcrawlercnes/principal/VinculoProfissionalEstabelecimento

```
1 package br.com.jucemar_dimon.jwebcrawlercnes.principal;
2
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5 import java.sql.ResultSet;
6 import java.sql.SQLException;
7 import java.sql.Statement;
8 import java.util.logging.Level;
9 import java.util.logging.Logger;
10
11 /**
12 * @author Jucemar
13 */
14 public class VinculoProfissionalEstabelecimentoDAO {
15
16     private Connection c = null;
17     private Statement stmt = null;
18
19     public VinculoProfissionalEstabelecimentoDAO() {
20
21         try {
22             Class.forName("org.sqlite.JDBC");
23             c = DriverManager.getConnection("jdbc:sqlite:estabelecimentos.db");
24
25             String tabelaProfissional = "CREATE TABLE IF NOT EXISTS vinculo_profissional_estabelecimento "
26                     + "(profissional_id INTEGER NOT NULL,"
27                     + " estabelecimento_cnes INTEGER NOT NULL,"
28                     + " PRIMARY KEY(profissional_id,estabelecimento_cnes),"
29                     + " FOREIGN KEY(profissional_id)"
30                     + " REFERENCES profissional(id),"
31                     + " FOREIGN KEY(estabelecimento_cnes)"
32                     + " REFERENCES estabelecimento(cnes));";
33
34         } catch (Exception e) {
35             e.printStackTrace();
36         }
37     }
38
39     public void inserir(VinculoProfissionalEstabelecimento vpe) {
40
41         try {
42             Connection con = DriverManager.getConnection("jdbc:sqlite:estabelecimentos.db");
43             Statement stmt = con.createStatement();
44
45             String sql = "INSERT INTO vinculo_profissional_estabelecimento (profissional_id, estabelecimento_cnes) VALUES (" +
46                     vpe.getProfissional().getId(), vpe.getEstabelecimento().getCodigo());
47
48             stmt.executeUpdate(sql);
49         } catch (SQLException ex) {
50             Logger.getLogger(VinculoProfissionalEstabelecimentoDAO.class.getName()).log(Level.SEVERE, null, ex);
51         }
52     }
53
54     public void alterar(VinculoProfissionalEstabelecimento vpe) {
55
56         try {
57             Connection con = DriverManager.getConnection("jdbc:sqlite:estabelecimentos.db");
58             Statement stmt = con.createStatement();
59
60             String sql = "UPDATE vinculo_profissional_estabelecimento SET estabelecimento_cnes = " +
61                     vpe.getEstabelecimento().getCodigo() + " WHERE id = " + vpe.getId();
62
63             stmt.executeUpdate(sql);
64         } catch (SQLException ex) {
65             Logger.getLogger(VinculoProfissionalEstabelecimentoDAO.class.getName()).log(Level.SEVERE, null, ex);
66         }
67     }
68
69     public void excluir(VinculoProfissionalEstabelecimento vpe) {
70
71         try {
72             Connection con = DriverManager.getConnection("jdbc:sqlite:estabelecimentos.db");
73             Statement stmt = con.createStatement();
74
75             String sql = "DELETE FROM vinculo_profissional_estabelecimento WHERE id = " + vpe.getId();
76
77             stmt.executeUpdate(sql);
78         } catch (SQLException ex) {
79             Logger.getLogger(VinculoProfissionalEstabelecimentoDAO.class.getName()).log(Level.SEVERE, null, ex);
80         }
81     }
82
83     public List<VinculoProfissionalEstabelecimento> listar() {
84
85         try {
86             Connection con = DriverManager.getConnection("jdbc:sqlite:estabelecimentos.db");
87             Statement stmt = con.createStatement();
88
89             String sql = "SELECT * FROM vinculo_profissional_estabelecimento";
90
91             ResultSet rs = stmt.executeQuery(sql);
92
93             List<VinculoProfissionalEstabelecimento> lista = new ArrayList<>();
94
95             while (rs.next()) {
96                 VinculoProfissionalEstabelecimento vpe = new VinculoProfissionalEstabelecimento();
97
98                 vpe.setId(rs.getInt("id"));
99                 vpe.setProfissional(new ProfissionalDAO().listar(rs.getInt("profissional_id")));
100                vpe.setEstabelecimento(new EstabelecimentoDAO().listar(rs.getInt("estabelecimento_cnes")));
101
102                lista.add(vpe);
103            }
104
105            return lista;
106        } catch (SQLException ex) {
107            Logger.getLogger(VinculoProfissionalEstabelecimentoDAO.class.getName()).log(Level.SEVERE, null, ex);
108        }
109
110        return null;
111    }
112}
```

```

E:/Projetos Netbeans/JWebCrawlerCNES/src/com/jucemar_dimon/jwebcrawlercnes/principal/VinculoProfissionalEstabelecimento

32     String index1 = "CREATE INDEX IF NOT EXISTS vinculo_profissional_estabelecimento_FKIndex1 ON
vinculo_profissional_estabelecimento (estabelecimento_cnes);";
33     String index2 = "CREATE INDEX IF NOT EXISTS vinculo_profissional_estabelecimento_FKIndex2 ON
vinculo_profissional_estabelecimento (profissional_id);";
34     String index3 = "CREATE INDEX IF NOT EXISTS vinculo_profissional_estabelecimento_PKIndex ON
vinculo_profissional_estabelecimento (profissional_id, estabelecimento_cnes);";
35
36     stmt = c.createStatement();
37     stmt.executeUpdate(tabellaProfissional);
38     stmt.executeUpdate(index1);
39     stmt.executeUpdate(index2);
40     stmt.executeUpdate(index3);
41     stmt.close();
42     c.close();
43     } catch (Exception e) {
44         System.err.println(e.getClassName() + ":" + e.getMessage());
45         System.exit(0);
46     } catch (ClassNotFoundException ex) {
47         Logger.getLogger(EstabelecimentoDAO.class.getName()).log(Level.SEVERE, null, ex);
48     } catch (SQLException ex) {
49         Logger.getLogger(EstabelecimentoDAO.class.getName()).log(Level.SEVERE, null, ex);
50     }
51
52     public void inserirVinculoDoProfissional(int profissional, int cnes) {
53         boolean inserir = isProfissionalVinculado(profissional, cnes);
54         if (inserir) {
55             try {
56                 c = DriverManager.getConnection("jdbc:sqlite:estabelecimentos.db");
57                 this.stmt = this.c.createStatement();
58                 String sql = "INSERT INTO vinculo_profissional_estabelecimento("
59

```

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar\_dimon/jwebcrawlercnes/principal/VinculoProfissionalEstabelecim

```
60      + "professional_id, estabeleimento_cnes)"  
61      + "VALUES(" +  
62          + professional + "• , •"  
63          + cnes + "•);";  
64      System.out.println(sql);  
65      this.stmt.executeUpdate(sql);  
66      stmt.close();  
67      c.close();  
68      } catch (SQLException ex) {  
69          Logger.getLogger(EstabelecimientoDAO.class.getName()).log(Level.SEVERE, null, ex);  
70      }  
71  }  
72 }  
73 }  
74 private boolean isProfessionalVinculado(int professional, int cnes) {  
75     boolean novoVinculo = true;  
76     ResultSet rs = null;  
77     try {  
78         c = DriverManager.getConnection("jdbc:sqlite:estabeleimentos.db");  
79         this.stmt = this.c.createStatement();  
80         String buscaVinculoProfessional = "SELECT * FROM vinculo_profissional_estabeleciamento WHERE  
81         professional_id=" +  
82             + professional +  
83             + " and estabeleimento_cnes=" +  
84             + cnes + "•";  
85         System.out.println(buscaVinculoProfessional);  
86         rs = this.stmt.executeQuery(buscaVinculoProfessional);  
87         if (rs.next()) {  
88             novoVinculo = false;  
89         }
```

E:/Projetos Netbeans/JWebCrawlerCNES/src(br/com/jucemar\_dimon/jwebcrawlercnes/principal/VinculoProfissionalEstabelecim

```
90         System.out.println("Professional já possui vinculo com este cnes + cbo");
91     }
92     rs.close();
93     stmt.close();
94     c.close();
95 } catch (SQLException ex) {
96     Logger.getLogger(ProfessionalDAO.class.getName()).log(Level.SEVERE, null, ex);
97 }
98 return novoVinculo;
99 }
100 }
```

## E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar\_dimon/webcrawlers/sigtap/Classificacao.java

```
1 package br.com.jucemar_dimon.webcrawlers.sigtap;
2
3 import java.io.Serializable;
4
5 /**
6 * @author Jucemar
7 */
8 public class Classificacao implements Serializable {
9
10    private int codigoServiço;
11    private int códigoClassificacao;
12    private String descrição;
13
14    public Classificacao(int códigoServiço, int códigoClassificacao, String descrição) {
15        this.codigoServiço = códigoServiço;
16        this.códigoClassificacao = códigoClassificacao;
17        this.descrição = descrição;
18    }
19
20    public int getCódigoServiço() {
21        return códigoServiço;
22    }
23
24    public int getCódigoClassificacao() {
25        return códigoClassificacao;
26    }
27
28    public String getDescrição() {
29        return descrição;
30    }
31}
```



## E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar\_dimon/jwebcrawlers/sigtap/ClassificacaoDAO.java

```
1 package br.com.jucemar_dimon.jwebcrawlers.sigtap;
2
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5 import java.sql.ResultSet;
6 import java.sql.SQLException;
7 import java.sql.Statement;
8 import java.util.logging.Level;
9 import java.util.logging.Logger;
10
11 /**
12 * @author Jucemar
13 */
14 public class ClassificacaoDAO {
15
16     private Connection c = null;
17     private Statement stmt = null;
18
19     public ClassificacaoDAO() {
20         try {
21             Class.forName("org.sqlite.JDBC");
22             c = DriverManager.getConnection("jdbc:sqlite:estabelecimentos.db");
23             System.out.println("Opened database successfully");
24
25             String deleta = "DROP TABLE IF EXISTS classificacao";
26             String tabelaClassificacao = "CREATE TABLE IF NOT EXISTS classificacao "
27                     + "(id INTEGER PRIMARY KEY AUTOINCREMENT,"
28                     + " servico_id INTEGER NOT NULL,"
29                     + " codigo INTEGER NOT NULL,"
30                     + " descricao TEXT NOT NULL,"
31                     + " FOREIGN KEY(servico_id)"
```

## E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar\_dimon/jwebcrawlercnes/sigtap/ClassificacaoDAO.java

```
32          + " REFERENCES servico(id)"  
33          + ");";  
34      String index = "CREATE INDEX IF NOT EXISTS classificacao_FKIndex ON classificacao (servico_id);";  
35      stmt = c.createStatement();  
36      stmt.executeUpdate(delete);  
37      stmt.executeUpdate(tabelaClassificacao);  
38      stmt.executeUpdate(index);  
39      stmt.close();  
40      c.close();  
41  } catch (Exception e) {  
42      System.out.println(e.getClassName() + ":" + e.getMessage());  
43      System.exit(0);  
44  }  
45  System.out.println("Tabela de classificações criada com sucesso");  
46  } catch (ClassNotFoundException ex) {  
47      Logger.getLogger(ClassificacaoDAO.class.getName()).log(Level.SEVERE, null, ex);  
48  } catch (SQLException ex) {  
49      Logger.getLogger(ClassificacaoDAO.class.getName()).log(Level.SEVERE, null, ex);  
50  }  
51  
52  public void inserirClassificacao(Classificacao classificacao) {  
53      ResultSet rs = null;  
54      try {  
55          c = DriverManager.getConnection("jdbc:sqlite:estabelecimentos.db");  
56          this.stmt = this.c.createStatement();  
57  
58          String idServico = "SELECT id FROM servico WHERE codigo=" + classificacao.get_codigoServico() + ";;"  
59          rs = this.stmt.executeQuery(idServico);  
60          String sql = "INSERT INTO classificacao("br/>61          + "codigo, servico_id, descricao)"  
62          + "VALUES (" + rs.getString("id") + ", " + classificacao.servico_id + ", " + classificacao.descricao + ")";  
63          this.stmt.executeUpdate(sql);  
64      } catch (SQLException ex) {  
65          Logger.getLogger(ClassificacaoDAO.class.getName()).log(Level.SEVERE, null, ex);  
66      }  
67  }
```

## E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar\_dimon/jwebcrawlercnes/sigtap/ClassificacaoDAO.java

```
63      + "VALUES("
64      + classificacao.getCodigoClassificacao() + ", "
65      + rs.getInt("id") + " ,
66      + classificacao.getDescricao() + "•"
67      + "); ";
68  System.out.println(sql);
69  this.stmt.executeUpdate(sql);
70  rs.close();
71  stmt.close();
72  c.close();
73  } catch (SQLException ex) {
74      Logger.getLogger(ClassificacaoDAO.class.getName()).log(Level.SEVERE, null, ex);
75  }
76
77
78  public void listarClassificacoes() {
79
80      ResultSet rs;
81      try {
82          c = DriverManager.getConnection("jdbc:sqlite:estabelecimentos.db");
83          this.stmt = this.c.createStatement();
84          rs = this.stmt.executeQuery("SELECT * FROM classificacao");
85
86          while (rs.next()) {
87              System.out.print("serviço: " + rs.getString("id") + " ");
88              System.out.print("Classificação: " + rs.getString("servico_id") + " ");
89              System.out.print("Descrição: " + rs.getString("descricao"));
90              System.out.println(" ");
91          }
92      }
93  }
```

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar\_dimon/jwebcrawlers/sigtap/ClassificacaoDAO.java

```
94     stmt.close();
95     c.close();
96   } catch (SQLException e) {
97     e.printStackTrace();
98   }
99 }
100 }
101 }
102 }
```

## E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar\_dimon/webcrawlercnes/sigtap/LeitorDeArquivos.java

```
1 package br.com.jucemar_dimon.webcrawlercnes.sigtap;
2
3 import java.io.BufferedReader;
4 import java.io.FileInputStream;
5 import java.io.FileNotFoundException;
6 import java.io.IOException;
7 import java.io.InputStreamReader;
8 import java.io.UnsupportedEncodingException;
9 import java.util.ArrayList;
10 import java.util.Collections;
11 import java.util.Comparator;
12
13 /**
14 * @author Jucemar
15 */
16 public class LeitorDeArquivos {
17
18     public ArrayList<Servico> carregarArquivoTxtServicos() throws FileNotFoundException, IOException {
19         ArrayList<Servico> resultado = new ArrayList<Servico>();
20         String nomeArquivo = "tb_servico.txt";
21         BufferedReader lerArq = new InputStreamReader(new FileInputStream(nomeArquivo),
22                                         "ISO-8859-1"));
23         String linha = "";
24         while ((linha = lerArq.readLine()) != null) {
25             String res = Limpar(linha);
26             String codigo = res.substring(0, 3);
27             String descricao = res.substring(3, res.length());
28             Servico s = new Servico(Integer.parseInt(codigo), descricao);
29             resultado.add(s);
30         }
31         lerArq.close();
32     }
33 }
```

### E:/Projetos Netbeans/JWebCrawlerCNES/src/com/jucemar\_dimon/jwebcrawlercnes/sigtap/LeitorDeArquivos.java

```
31     lerArq.close();
32     return resultado;
33 }
34
35 public ArrayList<Classificacao> carregarArquivoTxtClassificacoes() throws FileNotFoundException,
36     UnsupportedEncodingException, IOException {
37     ArrayList<Classificacao> resultado = new ArrayList<Classificacao>();
38     String nomeArquivo = "tb_servico_classificacao.txt";
39     BufferedReader lerArq = new BufferedReader(new InputStreamReader(new FileInputStream(nomeArquivo),
40         "ISO-8859-1"));
41     String linha = "";
42     while ((linha = lerArq.readLine()) != null) {
43         String res = limpar(linha);
44         String servico = res.substring(0, 3); classificacao
45         String = res.substring(3, 6); descricao =
46         String res.substring(6, res.length());
47         Classificacao c = new Classificacao(Integer.parseInt(servico), Integer.parseInt(classificacao),
48         descricao);
49         resultado.add(c);
50     }
51     lerArq.close();
52     lerArq.close();
53     Collections.sort(resultado, new Comparator() {
54         public int compare(Object o1, Object o2) {
55             Classificacao p1 = (Classificacao) o1;
56             Classificacao p2 = (Classificacao) o2;
57             return p1.getCodigoServico() < p2.getCodigoServico() ? -1 : (p1.getCodigoServico() >
58             p2.getCodigoServico() ? +1 : 0);
59         }
60     });
61     return resultado;
62 }
```

```
58      }
59
60  private String limpar(String txt) {
61      String resultado = null;
62      int i = 0;
63      boolean fim = false;
64      while (fim == false) {
65          int n = txt.indexOf(" ");
66          if (n > -1) {
67              fim = true;
68              resultado = txt.substring(0, n);
69          }
70          i++;
71      }
72      return resultado;
73  }
74
75 }
```

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar\_dimon/jwebcrawlers/sigtap/Servico.java

```
1 package br.com.jucemar_dimon.jwebcrawlers.sigtap;
2
3 import java.io.Serializable;
4
5 /**
6 * @author Jucemar
7 */
8 public class Servico implements Serializable {
9
10    private int codigo;
11    private String descricao;
12
13    public Servico(int codigo, String descricao) {
14        this.codigo = codigo;
15        this.descricao = descricao;
16    }
17
18    public int getCodigo() {
19        return codigo;
20    }
21
22    public String getDescricao() {
23        return descricao;
24    }
25}
```

## E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar\_dimon/webcrawlerses/sigtap/ServicoDAO.java

```
1 package br.com.jucemar_dimon.webcrawlerses.sigtap;
2
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5 import java.sql.ResultSet;
6 import java.sql.SQLException;
7 import java.sql.Statement;
8 import java.util.logging.Level;
9 import java.util.logging.Logger;
10
11 /**
12 * @author Jucemar
13 */
14 public class ServicoDAO {
15
16     private Connection c = null;
17     private Statement stmt = null;
18
19     public ServicoDAO() {
20         try {
21             Class.forName("org.sqlite.JDBC");
22             c = DriverManager.getConnection("jdbc:sqlite:estabelecimentos.db");
23             System.out.println("Opened database successfully");
24             try {
25                 String deleta = "DROP TABLE IF EXISTS servico";
26                 String tabelaServico = "CREATE TABLE IF NOT EXISTS servico "
27                     + "(id INTEGER PRIMARY KEY AUTOINCREMENT,"
28                     + " codigo INTEGER NOT NULL,"
29                     + " descricao TEXT NOT NULL"
30                     + ")";
31
32                 String index = "CREATE INDEX IF NOT EXISTS servico_PKIndex ON servico (id);";
33             }
34         } catch (Exception e) {
35             e.printStackTrace();
36         }
37     }
38
39     public void inserir(Servico servico) {
40         try {
41             stmt.executeUpdate("INSERT INTO servico (codigo, descricao) VALUES (" +
42                 servico.getCodigo() + ", " + servico.getDescricao());
43         } catch (SQLException ex) {
44             ex.printStackTrace();
45         }
46     }
47
48     public Servico buscarPorCodigo(int codigo) {
49         try {
50             ResultSet rs = stmt.executeQuery("SELECT * FROM servico WHERE codigo = " +
51                 codigo);
52             if (rs.next()) {
53                 Servico servico = new Servico();
54                 servico.setId(rs.getInt("id"));
55                 servico.setCodigo(rs.getInt("codigo"));
56                 servico.setDescricao(rs.getString("descricao"));
57                 return servico;
58             }
59         } catch (SQLException ex) {
60             ex.printStackTrace();
61         }
62         return null;
63     }
64
65     public void alterar(Servico servico) {
66         try {
67             stmt.executeUpdate("UPDATE servico SET descricao = " +
68                 servico.getDescricao() + " WHERE id = " + servico.getId());
69         } catch (SQLException ex) {
70             ex.printStackTrace();
71         }
72     }
73
74     public void excluir(int id) {
75         try {
76             stmt.executeUpdate("DELETE FROM servico WHERE id = " + id);
77         } catch (SQLException ex) {
78             ex.printStackTrace();
79         }
80     }
81 }
```

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar\_dimon/jwebcrawlercnes/sigtap/ServicoDAO.java

```
32     stmt = c.createStatement();
33     stmt.executeUpdate(delete);
34     stmt.executeUpdate(tableaServico);
35     stmt.executeUpdate(index);
36     stmt.close();
37     c.close();
38   } catch (Exception e) {
39     System.err.println(e.getClass().getName() + ": " + e.getMessage());
40     System.exit(0);
41   }
42   System.out.println("Tabela de serviços criada com sucesso");
43 } catch (ClassNotFoundException ex) {
44   Logger.getLogger(ClassificacaoDAO.class.getName()).log(Level.SEVERE, null, ex);
45 } catch (SQLException ex) {
46   Logger.getLogger(ClassificacaoDAO.class.getName()).log(Level.SEVERE, null, ex);
47 }
48
49 public void inserirServico(Servico servico) {
50   try {
51     c = DriverManager.getConnection("jdbc:sqlite:estabelecimentos.db");
52     this.stmt = this.c.createStatement();
53     try {
54       String sql = "INSERT INTO servico("
55         + "codigo, descricao"
56         + "VALUES("
57         + servico.getCodigo() + ", "
58         + servico.getDescricao() + "''"
59         + ");";
60       System.out.println(sql);
61     this.stmt.executeUpdate(sql);
62   }
63 }
```

### E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar\_dimon/jwebcrawlercnes/sigtap/ServicoDAO.java

```
63         } catch (SQLException e) {
64             e.printStackTrace();
65         }
66         stmt.close();
67         c.close();
68     } catch (SQLException ex) {
69         Logger.getLogger(ClassificacaoDAO.class.getName()).log(Level.SEVERE, null, ex);
70     }
71 }
72 public void listarServicos() {
73     ResultSet rs;
74     try {
75         c = DriverManager.getConnection("jdbc:sqlite:estabelecimentos.db");
76         this.stmt = this.c.createStatement();
77         rs = this.stmt.executeQuery("SELECT * FROM servico");
78         while (rs.next()) {
79             System.out.print("serviço: " + rs.getString("id") + " ");
80             System.out.print("Descrição: " + rs.getString("descricao"));
81             System.out.println(" ");
82         }
83         rs.close();
84         stmt.close();
85         c.close();
86     } catch (SQLException e) {
87         e.printStackTrace();
88     }
89 }
90 }
```

## E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar\_dimon/webcrawlercnes/sigtap/SigTap.java

```
1 package br.com.jucemar_dimon.webcrawlercnes.sigtap;
2
3 import java.io.FileNotFoundException;
4 import java.io.IOException;
5 import java.util.ArrayList;
6
7 /**
8 * @author Jucemar
9 */
10 public class SigTap {
11
12     public static void iniciar() throws FileNotFoundException, IOException {
13         LeitorDeArquivos leitor = new LeitorDeArquivos();
14         ArrayList<Servico> s = leitor.carregarArquivoTxtServicos();
15         ArrayList<Classificacao> c = leitor.carregarArquivoTxtClassificacoes();
16         ServicoDAO sd = new ServicoDAO();
17
18         for (Servico ser : s) {
19             sd.inserirServico(ser);
20         }
21         ClassificacaoDAO cd = new ClassificacaoDAO();
22         for (Classificacao cla : c) {
23             cd.inserirClassificacao(cla);
24             sd.listarServicos();
25         }
26
27     }
28 }
```

## E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar\_dimon/jwebcrawlercnes/utilitarios/ManipuladorDeArquivos.java

```
1 package br.com.jucemar_dimon.jwebcrawlercnes.utilitarios;
```

```
2 
```

```
3 import br.com.jucemar_dimon.jwebcrawlercnes.entidades.Municipio;
```

```
4 import java.io.BufferedInputStream;
```

```
5 import java.io.BufferedOutputStream;
```

```
6 import java.io.File;
```

```
7 import java.io.FileInputStream;
```

```
8 import java.io.FileOutputStream;
```

```
9 import java.io.IOException;
```

```
10 import java.io.ObjectInputStream;
```

```
11 import java.io.ObjectOutputStream;
```

```
12 import java.util.ArrayList;
```

```
13 import java.util.logging.Level;
```

```
14 import java.util.logging.Logger;
```

```
15 
```

```
16 /**
17 * @author Jucemar
18 */
19 public class ManipuladorDeArquivos {
```

```
20     private File arquivo;
```

```
21 
```

```
22     public ManipuladorDeArquivos() {
23         arquivo = new File("estabelecimentos_temp");
24         if (arquivo.exists() == false) {
25             try {
26                 arquivo.createNewFile();
27                 System.out.println("Um arquivo temporários de estabelecimentos foi criado com sucesso");
28             } catch (IOException ex) {
29                 Logger.getLogger(ManipuladorDeArquivos.class.getName()).log(Level.SEVERE, null, ex);
30             }
31         }
32     }
```

## E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar\_dimon/jwebcrawlercnes/utilitarios/ManipuladorDeArquivos.java

```
32     }
33 }
34
35 public void salveDadosEmArquivoTemporario<ArrayList<Municipio> dados) throws IOException {
36     ObjectOutputStream outputStream = null;
37     try {
38         outputStream = new ObjectOutputStream(new BufferedOutputStream(new FileOutputStream(arquivo)));
39         outputStream.writeObject(dados);
40     } catch (IOException ex) {
41         Logger.getLogger(ManipuladorDeArquivos.class.getName()).log(Level.SEVERE, null, ex);
42     } finally {
43         if (outputStream != null) {
44             outputStream.close();
45         }
46     }
47 }
48
49 public ArrayList<Municipio> leiaDadosDEArquivoTemporario() throws IOException {
50     ArrayList<Municipio> dados = null;
51     ObjectInputStream inputStream = null;
52     try {
53         inputStream = new ObjectInputStream(new BufferedInputStream(
54             new FileInputStream(arquivo)));
55         dados = (ArrayList<Municipio>) inputStream.readObject();
56         System.out.println("Há registros de um processamento anterior, a captura será retomado do ponto onde
parou");
57     } catch (IOException ex) {
58         System.out.println("Não há registros de processamentos anteriores, a captura será feita desde o início");
59     } catch (ClassNotFoundException ex) {
60         Logger.getLogger(ManipuladorDeArquivos.class.getName()).log(Level.SEVERE, null, ex);
61     } finally {
```

```
62         if (inputStream != null) {
63             inputStream.close();
64         }
65     }
66     return dados;
67 }
68 }
69 }
```

## File - E:\Android\Projetos\CadEsaude\app\src\main\AndroidManifest.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3   package="br.com.jucemar_dimon.cadesaude"
4   android:versionCode="1"
5   android:versionName="0.01">
6
7   <uses-feature
8     android:name="android.hardware.screen.portrait"
9     android:required="true" />
10  <uses-feature
11    android:name="android.hardware.telephony"
12    android:required="true" />
13  <uses-feature
14    android:name="android.hardware.location.gps"
15    android:required="true" />
16  <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"
17    />
18
19  <application
20    android:allowBackup="true"
21    android:icon="@mipmap/ic_launcher"
22    android:label="@string/app_name"
23    android:screenOrientation="portrait"
24    android:supportRtl="true"
25    android:theme="@style/MeuTemaMaterial">
26    <activity
27      android:label="@string/app_name"
28      android:launchMode="singleTop"
29      android:screenOrientation="portrait"
30      android:windowSoftInputMode="adjustPan|adjustNothing">
31      <intent-filter>
32        <action android:name="android.intent.action.VIEW" />
33      </intent-filter>
34      <intent-filter>
```

File - E:\Android\Projetos\CadEsaude\app\src\main\AndroidManifest.xml

```
36      <action android:name="android.intent.action.SEARCH" />
37  </intent-filter>
38  <meta-data
39    android:name="android.app.searchable"
40    android:resource="@xml/searchable" />
41  </activity>
42  <activity
43    android:name=".PreferenciasActivity"
44    android:label="@string/titulo_config"
45    android:launchMode="singleTop"
46    android:screenOrientation="portrait"
47    android:theme="@style/MeuTemaMaterial.Prefencias" />
48  <activity android:name=".AjudaActivity"
49    android:label="@string/titulo_ajuda"
50    android:launchMode="singleTop"
51    android:screenOrientation="portrait" />
52  <activity android:name=".SobreActivity"
53    android:label="@string/titulo_sobre"
54    android:launchMode="singleTop"
55    android:screenOrientation="portrait" />
56    <activity android:name=".BuscaGPSActivity"
57      android:label="@string/titulo_busca_gps_activity"
58      android:launchMode="singleTop"
59      android:screenOrientation="portrait"
60
61
62    android:windowSoftInputMode="adjustPan|adjustNothing">
63  <intent-filter>
64    <action android:name="android.intent.action.SEARCH" />
65  </intent-filter>
66
67  <meta-data
68    android:name="android.app.searchable"
69    android:resource="@xml/searchable" />
```

```
71 </activity>
72 <activity android:name=".EstabelecimientoActivity"
73     android:label="@string/title_activity_estabelecimento"
74     android:launchMode="singleTop"
75         android:screenOrientation="portrait" />
76 <meta-data android:name="com.google.android.gms.version"
77     android:value="@integer/google_play_services_version" />
78 <activity
79     android:name=".BuscaGeralActivity"
80     android:label="@string/title_activity_busca_geral"
81     android:launchMode="singleTop"
82         android:screenOrientation="portrait" />
83 <activity
84     android:name=".PrimeiroPassosActivity"
85     android:label="@string/title_activity_primeiros_passos"
86     android:launchMode="singleTop"
87         android:screenOrientation="portrait"
88         android:theme="@style/FullscreenTheme" />
89 <activity android:name=".PrimeiraConfiguracaoActivity"
90     android:label="@string/titulo_config"
91     android:launchMode="singleTop"
92         android:screenOrientation="portrait"
93         android:theme="@style/MeuTemaMaterial.Prefencias" />
94 <activity android:name=".AjudaTiposEstabelecimentosActivity"
95     android:label="@string/titulo_actvt_ajuda_estabelecimentos"
96     android:launchMode="singleTop"
97         android:screenOrientation="portrait" />
98 <activity android:name=".AjudaTiposDeServicosActivity"
99     android:label="@string/titulo_actvt_ajuda_servicos"
100        android:launchMode="singleTop"
101
102
103
104
```

File - E:\Android\Projetos\CadEsaude\app\src\main\AndroidManifest.xml

```
106     android:screenOrientation="portrait" />
107
108     <activity
109         android:name=".TelInicialActivity"
110         android:launchMode="singleTop"
111         android:screenOrientation="portrait">
112         <intent-filter>
113             <action android:name="android.intent.action.MAIN" />
114             <category android:name="android.intent.category.LAUNCHER" />
115         </intent-filter>
116     </activity>
117     </application>
118 </manifest>
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\AjudaActivity.java

```
1 package br.com.jucemar_dimon.cadesaude;
2 /**
3 * Created by Jucemar on 10/03/2016.
4 */
5
6 import android.content.Intent;
7 import android.support.v7.app.AppCompatActivity;
8 import android.os.Bundle;
9 import android.support.v7.widget.Toolbar;
10 import android.view.MenuItem;
11 import android.view.View;
12 import android.widget.Button;
13
14 public class AjudaActivity extends AppCompatActivity implements View.OnClickListener {
15
16     private Toolbar toolbar;
17     private Button buttonTiposEstabelecimentos;
18     private Button buttonTiposServicos;
19
20     @Override
21     protected void onCreate(Bundle savedInstanceState) {
22         super.onCreate(savedInstanceState);
23         setContentView(R.layout.activity_ajuda);
24         toolbar = (Toolbar) findViewById(R.id.toolbar_actvt_ajuda);
25         setSupportActionBar(toolbar);
26         getSupportActionBar().setDisplayHomeAsUpEnabled(true);
27         toolbar.setTitle("Ajuda");
28         buttonTiposEstabelecimentos = (Button) findViewById(R.id.btn_tipos_estabelecimentos);
29         buttonTiposServicos = (Button) findViewById(R.id.btn_tipos_servicos);
30         buttonTiposServicos.setOnClickListener(this);
31     }
32
33     @Override
34     public boolean onOptionsItemSelected(MenuItem item) {
```

File - E:\Android\Projetos\CadeSaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\AjudaActivity.java

```
36     int id = item.getItemId();
37     switch (id) {
38         case android.R.id.home:
39             onBackPressed();
40             break;
41     }
42     return super.onOptionsItemSelected(item);
43 }
44
45 @Override
46 public void onClick(View v) {
47     Intent i = null;
48     switch (v.getId()) {
49         case R.id.btn_tipos_estabelecimentos:
50             i = new Intent(this, AjudaTiposEstabelecimentosActivity.class);
51             break;
52         case R.id.btn_tipos_servicos:
53             i = new Intent(this, AjudaTiposDeServicosActivity.class);
54             break;
55     }
56     startActivity(i);
57 }
58 }
59 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\AjudaTiposDeServicosActivity.java

```
1 package br.com.jucemar_dimon.cadesaude;
2 /**
3 * Created by Jucemar on 10/03/2016.
4 */
5
6 import android.os.Bundle;
7 import android.support.v7.app.AppCompatActivity;
8 import android.support.v7.widget.Toolbar;
9 import android.view.MenuItem;
10
11 public class AjudaTiposDeServicosActivity extends AppCompatActivity {
12     private Toolbar toolbar;
13
14     @Override
15     protected void onCreate(Bundle savedInstanceState) {
16         super.onCreate(savedInstanceState);
17         setContentView(R.layout.activity_ajuda_tipos_de_servicos);
18         toolbar = (Toolbar) findViewById(R.id.toolbar_ajuda_tipos_servicos);
19         setSupportActionBar(toolbar);
20         getSupportActionBar().setDisplayHomeAsUpEnabled(true);
21         toolbar.setTitle("Tipos de serviços");
22     }
23
24     @Override
25     public boolean onOptionsItemSelected(MenuItem item) {
26         int id = item.getItemId();
27         switch (id) {
28             case android.R.id.home:
29                 onBackPressed();
30                 break;
31         }
32         return super.onOptionsItemSelected(item);
33     }
34 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\AjudaTiposEstabelecimentosActivity.java

```
1 package br.com.jucemar_dimon.cadesaude;
2 /**
3 * Created by Jucemar on 10/03/2016.
4 */
5
6 import android.os.Bundle;
7 import android.support.v7.app.AppCompatActivity;
8 import android.support.v7.widget.Toolbar;
9 import android.view.MenuItem;
10
11 public class AjudaTiposEstabelecimentosActivity extends AppCompatActivity {
12     private Toolbar toolbar;
13
14     @Override
15     protected void onCreate(Bundle savedInstanceState) {
16         super.onCreate(savedInstanceState);
17         setContentView(R.layout.activity_ajuda_tipos_estabelecimentos);
18         toolbar = (Toolbar) findViewById(R.id.toolbar_actvt_ajuda_tipos_estabelecimentos);
19         setSupportActionBar(toolbar);
20         getSupportActionBar().setDisplayHomeAsUpEnabled(true);
21         toolbar.setTitle("Tipos de estabelecimentos");
22     }
23
24     @Override
25     public boolean onOptionsItemSelected(MenuItem item) {
26         int id = item.getItemId();
27         switch (id) {
28             case android.R.id.home:
29                 onBackPressed();
30                 break;
31         }
32         return super.onOptionsItemSelected(item);
33     }
34 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\Atendimento.java

```
1 package br.com.jucemar_dimon.cadesaude;
2
3 import android.os.Parcel;
4 import android.os.Parcelable;
5
6 /**
7 * @author Jucemar
8 */
9 public class Atendimento implements Parcelable {
10     private String tipoAtendimento;
11     private String convenio;
12     private String cnes;
13
14     public Atendimento(String cnes, String tipoAtendimento, String convenio) {
15         this.tipoAtendimento = tipoAtendimento;
16         this.convenio = convenio;
17         this.cnes = cnes;
18     }
19
20     protected Atendimento(Parcel in) {
21         tipoAtendimento = in.readString();
22         convenio = in.readString();
23         cnes = in.readString();
24     }
25
26     public static final Creator<Atendimento> CREATOR = new Creator<Atendimento>() {
27         @Override
28         public Atendimento createFromParcel(Parcel in) {
29             return new Atendimento(in);
30         }
31     }
32     @Override
33     public Atendimento[] newArray(int size) {
34         return new Atendimento[size];
35     }
36 }
```

File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\Atendimento.java

```
36      };
37
38
39     @Override
40     public int describeContents() {
41         return 0;
42     }
43
44     @Override
45     public void writeToParcel(Parcel dest, int flags) {
46         dest.writeString(tipoAtendimento);
47         dest.writeString(convenio);
48         dest.writeString(cnnes);
49     }
50
51     public String getTipoAtendimento() {
52         return tipoAtendimento;
53     }
54
55     public void setTipoAtendimento(String tipoAtendimento) {
56         this.tipoAtendimento = tipoAtendimento;
57     }
58
59     public String getConvenio() {
60         return convenio;
61     }
62
63     public void setConvenio(String convenio) {
64         this.convenio = convenio;
65     }
66
67     public String getCnnes() {
68         return cnnes;
69     }
--
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\Atendimento.java

```
71  public void setCnes(String cnes) {  
72      this.cnes = cnes;  
73  }  
74  
75 }  
76
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\AtendimentoAdapter.java

```
1 package br.com.jucemar_dimon.cadesaude;
2
3 import android.content.Context;
4 import android.view.LayoutInflater;
5 import android.view.View;
6 import android.view.ViewGroup;
7 import android.widget.ArrayAdapter;
8 import android.widget.TextView;
9
10 import java.util.ArrayList;
11
12 /**
13 * Created by Jucemar on 19/03/2016.
14 */
15 public class AtendimentoAdapter extends ArrayAdapter<Atendimento> {
16
17     public AtendimentoAdapter(Context context, ArrayList<Atendimento> objects) {
18         super(context, R.layout.fragment_listas, objects);
19     }
20
21     @Override
22     public View getView(int position, View convertView, ViewGroup parent) {
23         Atendimento a = getItem(position);
24         if (convertView == null) {
25             convertView = LayoutInflater.from(getContext()).inflate(R.layout.item_listas, parent, false);
26         }
27         TextView tvAtendimento = (TextView) convertView.findViewById(R.id.item_linha_1);
28         TextView tvConvenio = (TextView) convertView.findViewById(R.id.item_linha_2);
29         tvAtendimento.setText(a.getTituloAtendimento());
30         tvConvenio.setText(a.getConvenio());
31
32         return convertView;
33     }
34 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\AtendimentoDAO.java

```
1 package br.com.jucemar_dimon.cadesaude;
2
3 import android.content.Context;
4 import android.database.Cursor;
5 import android.database.sqlite.SQLiteDatabase;
6 import android.util.Log;
7
8 import java.util.ArrayList;
9 import java.util.Set;
10 /**
11 * Created by Jucemar on 19/03/2016.
12 */
13
14 public class AtendimentoDAO {
15
16     private static AtendimentoDAO instancia;
17     private SQLiteDatabase bd;
18     private String[] colunas;
19
20     public static AtendimentoDAO getInstance(Context contexto) {
21         if (instancia == null) {
22             instancia = new AtendimentoDAO(contexto);
23         }
24         return instancia;
25     }
26
27     private AtendimentoDAO(Context contexto) {
28         BancoDados bancoDados = BancoDados.getInstance(contexto);
29         bd = bancoDados.getWritableDatabase();
30         colunas = new String[] {
31             TabelaAtendimento.Columnas.ATENDIMENTO,
32             TabelaAtendimento.Columnas.CONVENIO,
33             TabelaAtendimento.Columnas.ESTABELECIMENTO_CNES,
34             TabelaAtendimento.Columnas.ID
35         };
36     }
37
38     public void inserir(Atendimento atendimento) {
39         ContentValues valores = new ContentValues();
40         valores.put(TabelaAtendimento.Columnas.ATENDIMENTO, atendimento.getAtendimento());
41         valores.put(TabelaAtendimento.Columnas.CONVENIO, atendimento.getConvenio());
42         valores.put(TabelaAtendimento.Columnas.ESTABELECIMENTO_CNES, atendimento.getEstabelecimentoCnes());
43         valores.put(TabelaAtendimento.Columnas.ID, atendimento.getId());
44
45         bd.insert(TabelaAtendimento.TABELA_ATENDIMENTO, null, valores);
46     }
47
48     public void alterar(Atendimento atendimento) {
49         ContentValues valores = new ContentValues();
50         valores.put(TabelaAtendimento.Columnas.ATENDIMENTO, atendimento.getAtendimento());
51         valores.put(TabelaAtendimento.Columnas.CONVENIO, atendimento.getConvenio());
52         valores.put(TabelaAtendimento.Columnas.ESTABELECIMENTO_CNES, atendimento.getEstabelecimentoCnes());
53         valores.put(TabelaAtendimento.Columnas.ID, atendimento.getId());
54
55         bd.update(TabelaAtendimento.TABELA_ATENDIMENTO, valores, TabelaAtendimento.Columnas.ID + " = ?",
56                 new String[]{String.valueOf(atendimento.getId())});
57     }
58
59     public void excluir(Atendimento atendimento) {
60         bd.delete(TabelaAtendimento.TABELA_ATENDIMENTO, TabelaAtendimento.Columnas.ID + " = ?",
61                 new String[]{String.valueOf(atendimento.getId())});
62     }
63
64     public Atendimento buscarPorId(int id) {
65         Cursor cursor = bd.query(TabelaAtendimento.TABELA_ATENDIMENTO, colunas, TabelaAtendimento.Columnas.ID + " = ?",
66                 new String[]{String.valueOf(id)}, null, null, null);
67
68         if (cursor != null) {
69             cursor.moveToFirst();
70             Atendimento atendimento = new Atendimento();
71             atendimento.setId(cursor.getInt(cursor.getColumnIndex(TabelaAtendimento.Columnas.ID)));
72             atendimento.setAtendimento(cursor.getString(cursor.getColumnIndex(TabelaAtendimento.Columnas.ATENDIMENTO)));
73             atendimento.setConvenio(cursor.getString(cursor.getColumnIndex(TabelaAtendimento.Columnas.CONVENIO)));
74             atendimento.setEstabelecimentoCnes(cursor.getString(cursor.getColumnIndex(TabelaAtendimento.Columnas.ESTABELECIMENTO_CNES)));
75
76             return atendimento;
77         }
78
79         return null;
80     }
81
82     public Set buscarTodos() {
83         Set todos = new HashSet();
84
85         Cursor cursor = bd.query(TabelaAtendimento.TABELA_ATENDIMENTO, colunas, null, null, null, null, null);
86
87         if (cursor != null) {
88             cursor.moveToFirst();
89             while (!cursor.isLast()) {
90                 Atendimento atendimento = new Atendimento();
91                 atendimento.setId(cursor.getInt(cursor.getColumnIndex(TabelaAtendimento.Columnas.ID)));
92                 atendimento.setAtendimento(cursor.getString(cursor.getColumnIndex(TabelaAtendimento.Columnas.ATENDIMENTO)));
93                 atendimento.setConvenio(cursor.getString(cursor.getColumnIndex(TabelaAtendimento.Columnas.CONVENIO)));
94                 atendimento.setEstabelecimentoCnes(cursor.getString(cursor.getColumnIndex(TabelaAtendimento.Columnas.ESTABELECIMENTO_CNES)));
95
96                 todos.add(atendimento);
97             }
98         }
99
100        return todos;
101    }
102}
```

File - E:\Android\Projetos\CadESsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\AtendimentoDAO.java

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\AtendimentoDAO.java

```
69     if (i < municipios.length - 1) {
70         filtro.append("OR ");
71     }
72 }
73 String query = "SELECT DISTINCT atendimento_estabelecimento.atendimento FROM estabelecimento JOIN
74     atendimento_estabelecimento ON estabelecimento.cnes=atendimento.estabelecimento.estabelecimento_cnes WHERE " + filtro
75     + " ORDER BY atendimento_estabelecimento.atendimento";
76 Cursor c = bd.rawQuery(query, municipios);
77 resultado = new CharSequence[c.getCount()];
78 int i = 0;
79 if (c.moveToFirst()) {
80     do {
81         resultado[i] = c.getString(c.getColumnIndex(TabelaAtendimento.Colunas.ATENDIMENTO));
82         i++;
83     } while (c.moveToNext());
84 }
85 return resultado;
86 }
87 public CharSequence[] listarTiposConvenios(Set<String> setMunicipios)
88     CharSequence[] resultado = null;
89     if (setMunicipios.size() > 0) {
90         StringBuffer filtro = new StringBuffer();
91         String[] municipios = setMunicipios.toArray(new String[setMunicipios.size()]);
92         for (int i = 0; i < municipios.length; i++) {
93             filtro.append(TabelaEstabelecimento.NOME_TABELA + " ." + TabelaEstabelecimento.Colunas.CODIGO_MUNICIPIO
94             + "=? ");
95             if (i < municipios.length - 1) {
96                 filtro.append("OR ");
97             }
98         }
99     String query = "SELECT DISTINCT atendimento_estabelecimento.convenio FROM estabelecimento JOIN
atendimento_estabelecimento ON estabelecimento.estabelecimento_cnes WHERE " + filtro
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\AtendimentoDAO.java

```
99 + " ORDER BY atendimento_estabelecimento.convenio";
100 Cursor c = bd.rawQuery(query, municipios);
101 resultado = new CharSequence[c.getCount()];
102 int i = 0;
103 if (c.moveToFirst()) {
104     do {
105         resultado[i] = c.getString(c.getColumnIndex(TabelaAtendimento.Colunas.CONVENIO));
106         i++;
107     } while (c.moveToNext());
108     c.close();
109 }
110 return resultado;
111 }
112 }

113 public ArrayList<Estabelecimento> buscaStringEmConvenios(Set<String> setMunicipios, String sentenca) {
114     StringBuffer filtroMunicipios = new StringBuffer();
115     String[] municipios = setMunicipios.toArray(new String[setMunicipios.size()]);
116     for (int i = 0; i < municipios.length; i++) {
117         filtroMunicipios.append("•" + municipios[i] + "•");
118         if (i < municipios.length - 1) {
119             filtroMunicipios.append(",");
120         }
121     }
122     String query = "SELECT * FROM estabelecimento WHERE cnes IN (" + +
123     "SELECT DISTINCT atendimento_estabelecimento.estabelecimento_cnes FROM atendimento_estabelecimento
124     JOIN " +
125     "estabelecimento ON estabelecimento.cnes=atendimento_estabelecimento.estabelecimento_cnes WHERE
126     atendimento_estabelecimento.convenio " +
127     "LIKE '%" + sentenca + "%' AND estabelecimento.codigo_municipio IN (" + filtroMunicipios + ")";
128     Log.e("Query", query);
129     ArrayList<Estabelecimento> resultado = new ArrayList<Estabelecimento>();
130     Cursor c = bd.rawQuery(query, null);
131     if (c.moveToFirst()) {
132         do {
```

File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\AtendimentoDAO.java

```
132     Estabelecimento e = EstabelecimentoDAO.extrairDoCursor(c);
133     resultado.add(e);
134   } while (c.moveToNext());
135   c.close();
136   return resultado;
137 }
138 }

139 public ArrayList<Estabelecimento> buscaStringEmAtendimentos(Set<String> setMunicipios, String sentenca) {
140     StringBuffer filtroMunicipios = new StringBuffer();
141     String[] municipios = setMunicipios.toArray(new String[setMunicipios.size()]);
142     for (int i = 0; i < municipios.length; i++) {
143         filtroMunicipios.append("'" + municipios[i] + "'");
144         if (i < municipios.length - 1) {
145             filtroMunicipios.append(",");
146         }
147     }
148     String query = "SELECT * FROM estabelecimento WHERE cnes IN (" + 
149     "SELECT DISTINCT atendimento_estabelecimento.estabelecimento_cnes FROM atendimento_estabelecimento
150 JOIN " +
151     "estabelecimento ON estabelecimento.cnes=atendimento_estabelecimento.estabelecimento_cnes WHERE
152     atendimento_estabelecimento.atendimento " +
153     "LIKE '%" + sentenca + "%' AND estabelecimento.codigo_municipio IN (" + filtroMunicipios + ")";
154     Log.e("Query", query);
155     ArrayList<Estabelecimento> resultado = new ArrayList<Estabelecimento>();
156     Cursor c = bd.rawQuery(query, null);
157     if (c.moveToFirst()) {
158         do {
159             Estabelecimento e = EstabelecimentoDAO.extrairDoCursor(c);
160             resultado.add(e);
161         } while (c.moveToNext());
162     }
163     c.close();
164     return resultado;
165 }
```

File - E:\Android\Projetos\CadeSaude\app\src\main\java\br\com\br\cadesaude\AtendimentoDAO.java

```
165  
166 }  
167  
168  
169
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\AtendimentoFragment.java

```
1 package br.com.jucemar_dimon.cadesaude;
2
3 /**
4 * Created by Jucemar on 18/03/2016.
5 */
6
7 import android.os.Bundle;
8 import android.support.v4.app.Fragment;
9 import android.view.LayoutInflater;
10 import android.view.View;
11 import android.view.ViewGroup;
12 import android.widget.LinearLayout;
13 import android.widget.ListView;
14 import android.widget.TextView;
15
16 import java.util.ArrayList;
17
18 public class AtendimentoFragment extends Fragment {
19
20     public AtendimentoFragment() {
21
22     }
23
24     @Override
25     public void onCreate(Bundle savedInstanceState) {
26         super.onCreate(savedInstanceState);
27     }
28     @Override
29     public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
30         View v = inflater.inflate(R.layout.fragment_listas, container, false);
31         ArrayList<Atendimento> listaDeAtendimentos = getArguments().getParcelableArrayList("ListaDeAtendimentos");
32         if (listaDeAtendimentos.size() > 0) {
33             ListView lv = (ListView) v.findViewById(R.id.listView);
34             lv.setAdapter(new AtendimentoAdapter(aa = new AtendimentoAdapter(getActivity(), listaDeAtendimentos));
35         }
36     }
37 }
```

```
36     lv.setAdapter(aa);
37 } else {
38     LinearLayout layout = (LinearLayout) v.findViewById(R.id.view_vazio);
39     LinearLayout.LayoutParams params = new LinearLayout.LayoutParams(LinearLayout.LayoutParams.MATCH_PARENT,
40     LinearLayout.LayoutParams.WRAP_CONTENT);
41     params.bottomMargin = 16;
42     TextView textView = new TextView(getActivity());
43     textView.setLayoutParams(params);
44     textView.setText("Não há informações cadastradas");
45     layout.addView(textView);
46 }
47 }
48 }
49 }
50 }
51 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\BancoDeDados.java

```
1 package br.com.jucemar_dimon.cadesaude;
2
3 import android.content.Context;
4 import android.database.sqlite.SQLiteDatabase;
5 import android.database.sqlite.SQLiteOpenHelper;
6 import android.util.Log;
7
8 import java.io.File;
9 import java.io.FileOutputStream;
10 import java.io.IOException;
11 import java.io.InputStream;
12 import java.io.OutputStream;
13
14 /**
15 * Created by Jucemar on 10/03/2016.
16 */
17 public class BancoDeDados extends SQLiteOpenHelper {
18     public static String BD_CAMINHO = "/data/data;br.com.jucemar_dimon.cadesaude/databases/";
19     public static String BD_NOME = "estabelecimentos.db";
20     public static final int BD_VERSAO = 201610;
21     private static BancoDeDados instancia;
22     private Context contexto;
23
24     public static BancoDeDados getInstancia(Context contexto) {
25         if (instancia == null) {
26             instancia = new BancoDeDados(contexto);
27             String myPath = BD_CAMINHO + BD_NOME;
28             File banco = new File(myPath);
29             if (banco.exists() && !banco.isDirectory()) {
30                 banco.delete();
31             }
32             String caminho = "/data/data;br.com.jucemar_dimon.cadesaude/databases/";
33             File pasta = new File(caminho);
34             pasta.mkdirs();
35             File bancoPronto = new File(pasta, BD_NOME);
36         }
37     }
38 }
```

```
36     try {
37         bancoPronto.createNewFile();
38         Log.e("BancoDeDados", "arquivo de banco de dados não existe, ele será copiado");
39         instancia.copiarBancoDeDadosO;
40     } catch (IOException e) {
41         Log.e("BancoDeDados", "Houve algum erro durante a criação ou cópia do banco de dados");
42     }
43     return instancia;
44 }
45 }

46 private BancoDeDados(Context contexto) {
47     super(contexto, BD_NOME, null, BD_VERSAO);
48     this.contexto = contexto;
49 }
50

51 @Override
52 public void onCreate(SQLiteDatabase db) {
53     //Não implementado
54 }
55

56 public void copiarBancoDeDadosO {
57     try {
58         InputStream fluxoEntrada = contexto.getAssets().open(BD_NOME);
59         String arquivoFinal = BD_CMINHO + BD_NOME;
60         OutputStream fluxoSaida = new FileOutputStream(arquivoFinal);
61         byte[] buffer = new byte[1024];
62         int tamanho;
63         while ((tamanho = fluxoEntrada.read(buffer)) > 0) {
64             fluxoSaida.write(buffer, 0, tamanho);
65         }
66         fluxoSaida.flushO;
67         fluxoEntrada.closeO;
68         fluxoSaida.closeO;
69         Log.e("BancoDeDados", "O Arquivo de banco de dados foi copiado com sucesso ");
70     }
71 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\BancoDeDados.java

```
71     } catch (Exception e) {
72         Log.e("BancoDeDados", "Erro " + e.getMessage());
73     }
74 }
75
76 @Override
77 public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
78     //Não implementado
79 }
80
81 }
82
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\BuscaGeralActivity.java

```
1 package br.com.jucemar_dimon.cadesaude;
2 /**
3 * Created by Jucemar on 10/03/2016.
4 */
5
6 import android.content.Context;
7 import android.content.SharedPreferences;
8 import android.os.AsyncTask;
9 import android.os.Bundle;
10 import android.preference.PreferenceManager;
11 import android.support.design.widget.CoordinatorLayout;
12 import android.support.design.widget.Snackbar;
13 import android.support.v7.app.AppCompatActivity;
14 import android.support.v7.widget.DefaultItemAnimator;
15 import android.support.v7.widget.LinearLayoutManager;
16 import android.support.v7.widget.RecyclerView;
17 import android.support.v7.widget.Toolbar;
18 import android.util.Log;
19 import android.view.MenuItem;
20 import android.view.View;
21 import android.widget.ImageView;
22 import android.widget.ProgressBar;
23 import android.widget.TextView;
24 import com.github.clans.fab.FloatingActionButton;
25
26 import java.util.ArrayList;
27 import java.util.Arrays;
28 import java.util.Collections;
29 import java.util.HashSet;
30 import java.util.Set;
31
32 public class BuscaGeralActivity extends AppCompatActivity implements InterfaceBuscaGeralPorString, View.OnClickListener
33 {
34 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\BuscaGeralActivity.java

```
35 private ArrayList<Estabelecimento> estabeleimentos;
36 private Set<String> listaDeMunicipios;
37 private EstabelecimentoDAO estabDAO;
38 private ProgressBar barraDeProgresso;
39 private RecyclerView meuRecyclerView;
40 private EstabelecimentoAdapter meuAdapter;
41 private RecyclerView.LayoutManager meuLayoutManager;
42 private CarregadorDeDados carregadorDeDados;
43 private int posicaoNaLista; Set<String>
44 private ListaDeFavoritos; SharedPreferences config;
45 private ArrayList<Profissional> listaDeProfissionais;
46 private ArrayList<Atendimento> listaDeAtendimento;
47 private ArrayList<ServiçoClassificação> listaDeServiços;
48 private ImageView imagemBackground;
49 private TextView txtBackground;
50 private TextView txtBackgroundMunicípios;
51 private FloatingActionButton fab;
52 private SharedPreferences preferences;
53 private ArrayList<String> codMunicípio;
54 private ArrayList<String> nomeMunicípio;
55 ...
56
57 @Override
58 protected void onCreate(Bundle savedInstanceState) {
59     super.onCreate(savedInstanceState);
60     setContentView(R.layout.activity_busca_geral);
61     barraDeProgresso = (ProgressBar) findViewById(R.id.barra_progresso_busca_geral);
62     toolbar = (Toolbar) findViewById(R.id.toolbar_actvt_toolbar_actvt_busca_geral);
63     setSupportActionBar(toolbar);
64     getSupportActionBar().setDisplayHomeAsUpEnabled(true);
65     imagemBackground = (ImageView) findViewById(R.id.background_erro_busca_geral);
66     txtBackground = (TextView) findViewById(R.id.label_erro_busca);
67     txtBackground.setText(R.string.titulo_dialog_buscar_geral);
68     txtBackgroundMunicípios = (TextView) findViewById(R.id.label_municípios_nara_busca);
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\BuscaGeralActivity.java

```
70     fab = (FloatingActionButton) findViewById(R.id.fab_activity_busca_geral);
71     fab.setOnClickListener(this);
72     estabelecimentos = new ArrayList<Estabelecimento>();
73     meuRecyclerView = (RecyclerView) findViewById(R.id.lista_estabelecimentos);
74     meuRecyclerView.setVisibility(View.GONE); meuLayoutManager =
75     new LinearLayoutManager(this);
76     meuRecyclerView.setLayoutManager(meuLayoutManager);
77     meuRecyclerView.setItemAnimator(new DefaultItemAnimator());
78     meuRecyclerView.setHasFixedSize(true);
79     mostrarMunicipiosEscondidos();
80 }
81
82 @Override
83 public boolean onOptionsItemSelected(MenuItem item) {
84     int id = item.getItemId();
85     switch (id) {
86         case android.R.id.home:
87             onBackPressed();
88             break;
89     }
90     return super.onOptionsItemSelected(item);
91 }
92
93 private void mostrarMunicipiosEscondidos() {
94     SharedPreferences preferences = PreferenceManager.getDefaultSharedPreferences(this);
95     String[] arrayNomes = getResources().getStringArray(R.array.nomes_municipios_view);
96     nomeMunicipio = new ArrayList<String>(Arrays.asList(arrayNomes));
97     String[] arrayCodigos = getResources().getStringArray(R.array.codigos_municipios);
98     codMunicipio = new ArrayList<String>(Arrays.asList(arrayCodigos));
99     ...
100    Set<String> temp = preferencias.getStringSet(getResources().getString(R.string.pref_municipios_preferidos_key)
101    , new HashSet<String>());
102    StringBuffer sb = new StringBuffer();
103    ArrayList<String> tempList = new ArrayList(temp);
104    for (int i = 0; i < tempList.size(); i++) {
105        int j = codMunicipio.indexOf(tempList.get(i));
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\BuscaGeralActivity.java

```
104     sb.append(nomeMunicipio.get(j));
105     if (i < tempList.size() - 1) {
106         sb.append(" > ");
107     }
108     sb.append(" - ");
109     txtBackgroundMunicipios.setText("A busca será realizada apenas no(s) município(s): " + sb.toString());
110
111 }
112
113 @Override
114 public void informarTermoParaConsulta(int tipoBusca, String termo) {
115     barraDeProgresso.setVisibility(View.VISIBLE);
116     txtBackground.setBackground(View.GONE);
117     txtBackgroundMunicipios.setVisibility(View.GONE);
118     meuRecyclerView.setVisibility(View.GONE);
119     imagemBackground.setVisibility(View.GONE);
120     carregadorDeDados = new CarregadorDeDados(this, tipoBusca, termo);
121     carregadorDeDados.execute();
122 }
123
124 @Override
125 public void onClick(View v) {
126     BuscaGeralDialog b = new BuscaGeralDialog();
127     b.setListaDeMunicipios(listaDeMunicipios);
128     b.show(this.getFragmentManager(), "DialogBuscaGeral");
129 }
130
131 private class CarregadorDeDados extends AsyncTask<Void, Void, Void> {
132     private Context contexto;
133     private String termoBuscado;
134     private int tipoBusca;
135
136     public CarregadorDeDados(Context contexto, int tipoBusca, String termoBuscado) {
137         this.contexto = contexto;
138     }
139 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\BuscaGeralActivity.java

```
139     this.termoBuscado = termoBuscado;
140     this.tipoBusca = tipoBusca;
141     Log.e("tipoDeBusca", "Tipo de busca= " + tipoBusca + " Termo: " + termoBuscado);
142 }
143
144 @Override
145     protected void doInBackground(Void... params) {
146         config = getSharedPreferences("favoritos", MODE_PRIVATE);
147         listaDeFavoritos = config.getStringSet("ListaDeFavoritos", new HashSet<String>());
148         listaDeMunicipios = PreferenceManager.getDefaultSharedPreferences(contexto).getStringSet(getString(R.string.pref_municipios_preferidos_key), new HashSet<String>());
149         estabelecimentos.clear();
150
151         switch (tipoBusca) {
152             case BuscaGeralDialog.SPINNER_CONVENTOS_ACEITOS:
153                 AtendimentoDAO atendimentoDAOConv = AtendimentoDAO.getInstancia(getApplicationContext());
154                 estabelecimentos = atendimentoDAOConv.buscaStringEmConvenios(ListaDeMunicipios, termoBuscado);
155                 break;
156             case BuscaGeralDialog.SPINNER_FORMAS_DE_ATENDIMENTO:
157                 AtendimentoDAO atendimentoDAOAten = AtendimentoDAO.getInstancia(getApplicationContext());
158                 estabelecimentos = atendimentoDAOAten.buscaStringEmAtendimentos(ListaDeMunicipios, termoBuscado);
159                 break;
160             case BuscaGeralDialog.SPINNER_MEDICOS:
161                 termoBuscado = StringUtil.removeAcentos(termoBuscado);
162                 termoBuscado = termoBuscado.trim();
163                 ProfissionaisDAO profissionaisDAOMed = ProfissionaisDAO.getInstancia(getApplicationContext());
164                 estabelecimentos = profissionaisDAOMed.buscaStringEmNomeProfissionais(ListaDeMunicipios,
165                 termoBuscado);
166                 break;
167             case BuscaGeralDialog.SPINNER_ESPECIALIDADES:
168                 ProfissionaisDAO profissionaisDAOEsp = ProfissionaisDAO.getInstancia(getApplicationContext());
169                 estabelecimentos = profissionaisDAOEsp.buscaStringEmNomeEspecialidade(ListaDeMunicipios,
170                 termoBuscado);
171                 break;
172             case BuscaGeralDialog.SPINNER_NOME_DO_ESTABELECIMENTO:
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\BuscaGeralActivity.java

```
171     termoBuscado = StringUtils.removerAcentos(termoBuscado);
172     termoBuscado = termoBuscado.trim();
173     termoBuscado = termoBuscado.toUpperCase();
174     estabDAO = EstabelecimentoDAO.getInstance(getApplicationContext());
175     estabeleimentos = estabDAO.buscaStringEmEstabeleimentos(listaDeMunicipios, termoBuscado);
176     break;
177   case BuscaGeralDialog.SPINNER_SERVICOS_OFERECIDOS:
178     ServicoClassificacaoDAO servicoClassificacaoDAO = ServicoClassificacaoDAO.getInstance(
179     getApplicationContext());
180     estabeleimentos = servicoClassificacaoDAO.buscaStringEmServicos(listaDeMunicipios, termoBuscado);
181     break;
182   }
183   Collections.sort(estabeleimentos);
184   return null;
185 }
186 @Override
187 protected void onPostExecute(Void aVoid) {
188   super.onPostExecute(aVoid);
189   menuAdapter = new EstabelecimentoAdapter(contexto, estabeleimentos, listaDeFavoritos);
190   meuRecyclerView.setAdapter(menuAdapter);
191   if (estabeleimentos.size() == 0) {
192     txtBackground.setText(R.string.txt_erro_busca_geral);
193     txtBackground.setVisibility(View.VISIBLE);
194     imagemBackground.setVisibility(View.VISIBLE);
195     barraDeProgresso.setVisibility(View.GONE);
196     txtBackgroundMunicipios.setVisibility(View.GONE);
197     meuRecyclerView.setVisibility(View.GONE);
198   } else {
199     txtBackground.setVisibility(View.GONE);
200     imagemBackground.setVisibility(View.GONE);
201     barraDeProgresso.setVisibility(View.GONE);
202     meuRecyclerView.setVisibility(View.VISIBLE);
203   }
204 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\BuscaGeralActivity.java

```
205     LENGTH_SHORT).showO;
206     Snackbar.make(c1, "Foram localizados " + estabelecimentos.sizeO + " estabelecimentos", Snackbar.
207     LENGTH_SHORT).showO;
208 }
209 }
210 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\BuscaGeralDialog.java

```
1 package br.com.jucemar_dimon.cadesaude;
2
3 import android.app.Activity;
4 import android.app.AlertDialog;
5 import android.app.Dialog;
6 import android.app.DialogFragment;
7 import android.app.ProgressDialog;
8 import android.content.Context;
9 import android.content.DialogInterface;
10 import android.os.AsyncTask;
11 import android.os.Bundle;
12 import android.preference.PreferenceManager;
13 import android.text.Editable;
14 import android.text.TextWatcher;
15 import android.view.LayoutInflater;
16 import android.view.View;
17 import android.widget.AdapterView;
18 import android.widget.ArrayAdapter;
19 import android.widget.Button;
20 import android.widget.EditText;
21 import android.widget.Spinner;
22 import java.util.HashSet;
23 import java.util.Set;
24
25 /**
26 * Created by Jucemar on 13/04/2016.
27 */
28 public class BuscaGeralDialog extends DialogFragment implements DialogInterface.OnClickListener, TextWatcher,
~ AdapterView.OnItemSelectedListener {
30     private InterfaceBuscaGeralPorString comunicador;
31     private EditText campoConsulta;
32     private Button buttonConfirmar;
33     private Button buttonCancelar;
34     private ArrayAdapter<CharSequence> tiposDeBuscaAdapter;
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\BuscaGeralDialog.java

```
35     private Spinner comboBoxTipoBusca;
36     private Spinner comboBoxBusca;
37     private int tipoDeBusca;
38     private Set<String> listaDeMunicipios;
39     private String busca;
40     private BuscaGeralSpinnerAdapter valoresDeBuscaAdapter;
41     public static final int SPINNER_SELECAO_UMA_OPCAO = 0;
42     public static final int SPINNER_CONVENTIOS_ACEITOS = 1;
43     public static final int SPINNER_FORMAS_DE_ATENDIMENTO = 2;
44     public static final int SPINNER_MEDICOS = 3;
45     public static final int SPINNER_ESPECIALIDADES = 4;
46     public static final int SPINNER_NOME_DO_ESTABELECIMENTO = 5;
47     public static final int SPINNER_SERVICOS_OFERECIDOS = 6;
48
49     public void setListaDeMunicipios(Set<String> listaDeMunicipios) {
50         this.listaDeMunicipios = listaDeMunicipios;
51     }
52
53     public BuscaGeralDialogO {
54         this.tipoDeBusca = -1;
55     }
56
57     @Override
58     public void onAttach(Activity activity) {
59         super.onAttach(activity);
60         if (!(activity instanceof InterfaceBuscaGeralPorString)) {
61             throw new RuntimeException("A activity precisa implementar a interface InterfaceBuscaPorTermo");
62         }
63         comunicador = (InterfaceBuscaGeralPorString) activity;
64     }
65
66     @Override
67     public Dialog onCreateDialog(Bundle savedInstanceState) {
68         valoresDeBuscaAdapter = new BuscaGeralSpinnerAdapter(getActivity());
69     }
70 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\BuscaGeralDialog.java

```
70     LayoutInflater l = (LayoutInflater) getSystemService(Context.LAYOUT_INFLATER_SERVICE);
71     View v = l.inflate(R.layout.view_dialog_busca_geral, null);
72     AlertDialog.Builder builder = new AlertDialog.Builder(getActivity());
73     builder.setTitle(R.string.titulo_dialog_buscar_geral)
74         .setPositiveButton("Confirmar", this)
75         .setNegativeButton("Cancelar", this)
76         .setView(v);
77     campoConsulta = (EditText) v.findViewById(R.id.campo_busca_geral_dialog);
78     campoConsulta.addTextChangedListener(this);
79     tipsDeBuscaAdapter = new ArrayAdapter<CharSequence>(getActivity(), android.R.layout.simple_spinner_item);
80     comboBoxTipoBusca = (Spinner) v.findViewById(R.id.comboBox_tipo_busca_geral_dialog);
81     comboBoxBusca = (Spinner) v.findViewById(R.id.comboBox_busca_geral_dialog);
82     comboBoxTipoBusca.setOnItemSelectedListener(this);
83     comboBoxBusca.setOnItemSelectedListener(new EscutadorComboBoxCampoBusca());
84     String[] itens = getResources().getStringArray(R.array.tipos_buscas);
85     for (String s : itens) {
86         tiposDeBuscaAdapter.add(s);
87     }
88     tipsDeBuscaAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
89     comboBoxTipoBusca.setAdapter(tiposDeBuscaAdapter);
90     return builder.create();
91 }
92 @Override
93 public void onStart() {
94     super.onStart();
95     AlertDialog dialog = (AlertDialog) getDialog();
96     dialog.setCancelable(false);
97     dialog.setCanceledOnTouchOutside(false);
98     this.buttonConfirmar = ((AlertDialog) dialog).getButton(DialogInterface.BUTTON_POSITIVE);
99     buttonConfirmar.setEnabled(false);
100    this.buttonCancelar = ((AlertDialog) dialog).getButton(DialogInterface.BUTTON_NEGATIVE);
101 }
102 ...
103 @Override
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\BuscaGeralDialog.java

```
105  public void onClick(DialogInterface dialog, int which) {
106      switch (which) {
107          case DialogInterface.BUTTON_NEGATIVE:
108              dialog.cancel();
109          break;
110          case DialogInterface.BUTTON_POSITIVE:
111              comunicador.informarTermoParaConsulta(tipoDeBusca, busca);
112          break;
113      }
114  }
115
116  @Override
117  public void beforeTextChanged(CharSequence s, int start, int count, int after) {
118
119  }
120
121  @Override
122  public void onTextChanged(CharSequence s, int start, int before, int count) {
123      if (campoConsulta.getVisibility() == View.VISIBLE) {
124          if (s.length() < 3) {
125              buttonConfirmar.setEnabled(false);
126          } else {
127              buttonConfirmar.setEnabled(true);
128          }
129          busca = s.toString();
130      }
131  }
132  @Override
133  public void afterTextChanged(Editable s) {
134  }
135
136  @Override
137  public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {
138      tipoDeBusca = position;
...}
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\BuscaGeralDialog.java

```
140     switch (position) {
141         case SPINNER_MEDICOS:
142             campoConsulta.setVisibility(View.VISIBLE);
143             comboBoxBusca.setVisibility(View.GONE);
144             break;
145         case SPINNER_NOME_DO_ESTABELECIMENTO:
146             campoConsulta.setVisibility(View.VISIBLE);
147             comboBoxBusca.setVisibility(View.GONE);
148             break;
149         case SPINNER_SELECAO_NE_UMA_OPCAO:
150             campoConsulta.setText("");
151             campoConsulta.setVisibility(View.GONE);
152             comboBoxBusca.setVisibility(View.GONE);
153             buttonConfirmar.setEnabled(false);
154             break;
155         default:
156             campoConsulta.setVisibility(View.GONE);
157             valoresDeBuscaAdapter.clear();
158             CarregadorDeDados carregadorDeDados = new CarregadorDeDados(getActivity(), position);
159             carregadorDeDados.execute();
160             ;
161             ;
162             ;
163             ;
164     }
165     @Override
166     public void onNothingSelected(AdapterView<?> parent) {
167         //Não há necessidade de implementar, o android chama esse método somente se a dialog dismiss ou se o adapter
168         tem tamanho==0
169     }
170     private class CarregadorDeDados extends AsyncTask<Void, Void, Void> {
171         private int tipoBusca;
172         private CharSequence[] dadosSpinner02;
173         private ProgressDialog pd;
```

```
174     private Context contexto;
175
176     public CarregadorDeDados(Context contexto, int tipoBusca) {
177         this.tipoBusca = tipoBusca;
178         this.contexto = contexto;
179     }
180
181     @Override
182     protected void onPreExecute() {
183         super.onPreExecute();
184         pd = ProgressDialog.show(contexto, "Carregando", "Aguarde", true, true);
185         pd.setOnCancelListener(new DialogInterface.OnCancelListener() {
186
187             @Override
188             public void onCancel(DialogInterface dialog) {
189                 try {
190                     this.finalize();
191                 } catch (Throwable throwable) {
192                     throwable.printStackTrace();
193                 }
194             }
195         });
196     }
197
198     @Override
199     protected void doInBackground(Void... params) {
200         ListaDeMunicipios = PreferenceManager.getDefaultSharedPreferences().getActivity();
201         -getStringSet(getString(R.string.pref_municipios_key), new HashSet<String>());
202         switch (tipoDeBusca) {
203             case SPINNER_CONVENIOS_ACEITOS:
204                 AtendimentoDAO atendimentoDAOConv = AtendimentoDAO
205                     .getInstancia(contexto.getApplicationContext());
206                 dadosSpinner02 = atendimentoDAOConv.listarTiposConvenios(listaDeMunicipios);
207                 break;
208             case SPINNER_ESPECIALIDADES:
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\BuscaGeneralDialog.java

```
209     ProfissionaisDAO profissionaisDAO = ProfissionaisDAO
210     ->getInstancia(contexto.getApplicationContext());
211     dadosSpinner02 = profissionaisDAO.listarTiposProfissionais(listaDeMunicipios);
212     break;
213   case SPINNER_FORMAS_DE_ATENDIMENTO:
214     AtendimentoDAO atendimentoDAOAtend = AtendimentoDAO
215     ->getInstancia(contexto.getApplicationContext());
216     dadosSpinner02 = atendimentoDAOAtend.ListarTiposAtendimento(listaDeMunicipios);
217     break;
218   case SPINNER_SERVICOS_OFERECIDOS:
219     ServicoClassificacaoDAO servicoClassificacaoDAO = ServicoClassificacaoDAO
220     ->getInstancia(contexto.getApplicationContext());
221     dadosSpinner02 = servicoClassificacaoDAO.listarTiposServicos(listaDeMunicipios);
222     break;
223   }
224   return null;
225 }
226 @Override
227 protected void onPostExecute(Void aVoid) {
228   super.onPostExecute(aVoid);
229   if (pd.isShowing())
230     pd.dismiss();
231   valoresDeBuscaAdapter.add("Selecione um item");
232   valoresDeBuscaAdapter.notifyDataSetChanged();
233   valoresDeBuscaAdapter.setDropDownViewResource(R.layout.item_spinner);
234   comboBoxBusca.setAdapter(valoresDeBuscaAdapter);
235   comboBoxBusca.setVisibility(View.VISIBLE);
236 }
237 }
238
239 private class EscutadorComboBoxCampoBusca implements AdapterView.OnItemSelectedListener {
240   @Override
241   public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {
242     ...
243   }
244 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\BuscaGeralDialog.java

```
244     if (position > 0) {
245         buttonConfirmar.setEnabled(true);
246     } else {
247         buttonConfirmar.setEnabled(false);
248     }
249     busca = (String) parent.getItemAtPosition(position);
250 }
251
252 @Override
253 public void onNothingSelected(AdapterView<?> parent) {
254 }
255 }
256 }
257 }
258 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\BuscaGeralSpinnerAdapter.java

```
1 package br.com.jucemar_dimon.cadesaude;
2
3 import android.content.Context;
4 import android.view.LayoutInflater;
5 import android.view.View;
6 import android.view.ViewGroup;
7 import android.widget.ArrayAdapter;
8 import android.widget.TextView;
9
10 import java.util.ArrayList;
11
12 /**
13 * Created by Jucemar on 07/05/2016.
14 */
15 public class BuscaGeralSpinnerAdapter extends ArrayAdapter<CharSequence> {
16
17     private Context contexto;
18     private ArrayList<String> spinnerValues;
19
20     public BuscaGeralSpinnerAdapter(Context context, ArrayList<String> spinnerValues) {
21         super(context, R.layout.item_spinner);
22         this.contexto = context;
23         this.spinnerValues = spinnerValues;
24     }
25
26     public BuscaGeralSpinnerAdapter(Context context) {
27         super(context, R.layout.view_dialog_busca_geral);
28         this.contexto = context;
29     }
30
31     @Override
32     public View getView(int pos, View cnvView, ViewGroup prnt) {
33         return getCustomView(pos, cnvView, prnt);
34     }
35 }
```

File - E:\Android\Projetos\CadeSaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\BuscaGPEASpinningAdapter.java

```
36 package br.com.jucemardimondesousa;
37 import android.view.View;
38 import android.widget.ArrayAdapter;
39 import android.widget.TextView;
40 import android.widget.Spinner;
41 import android.widget.AdapterView;
42 import android.app.Activity;
43 import android.view.ViewGroup;
44 import android.view.LayoutInflater;
45 import android.view.View.OnClickListener;
46 import android.widget.AdapterView.OnItemSelectedListener;
47 import android.widget.TextView;
48 import android.app.AlertDialog;
49 import android.app.SearchManager;
50 import android.content.Context;
51 import android.content.Intent;
52 import android.content.IntentSender;
53 import android.content.SharedPreferences;
54 import android.content.pm.PackageManager;
55 import android.graphics.Color;
56 import android.location.Location;
57 import android.os.AsyncTask;
58 import android.os.Bundle;
59 import android.preference.PreferenceManager;
60 import android.support.design.widget.CoordinatorLayout;
61 import android.support.design.widget.Snackbar;
62 import android.support.v4.app.ActivityCompat;
63 import android.support.v4.view.MenuItemCompat;
64 import android.support.v7.app.AppCompatActivity;
65 import android.support.v7.widget.DefaultItemAnimator;
66 import android.support.v7.widget.LinearLayoutManager;
67 import android.support.v7.widget.RecyclerView;
68 import android.support.v7.widget.SearchView;
69 import android.support.v7.widget.Toolbar;
70 import android.util.Log;
71 import android.view.Menu;
72 import android.view.MenuInflater;
73 import android.view.MenuItem;
74 import android.view.View;
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\BuscaGPSActivity.java

```
36 import android.widget.ProgressBar;
37 import android.widget.TextView;
38
39 import com.github.clans.fab.FloatingActionButton;
40 import com.github.clans.fab.FloatingActionButton;
41 import com.google.android.gms.common.ConnectionResult;
42 import com.google.android.gms.common.GoogleApiAvailability;
43 import com.google.android.gms.common.api.GoogleApiClient;
44 import com.google.android.gms.common.api.GoogleApiClient.ConnectionCallbacks;
45 import com.google.android.gms.common.api.PendingResult;
46 import com.google.android.gms.common.api.ResultCallback;
47 import com.google.android.gms.common.api.Status;
48 import com.google.android.gms.location.LocationListener;
49 import com.google.android.location.LocationRequest;
50 import com.google.android.location.LocationServices;
51 import com.google.android.gms.location.LocationSettingsRequest;
52 import com.google.android.location.LocationSettingsResult;
53 import com.google.android.location.LocationSettingsStates;
54 import com.google.android.gms.location.LocationSettingsStatuses;
55 import java.util.ArrayList;
56 import java.util.Arrays;
57 import java.util.HashSet;
58 import java.util.Set;
59 import java.util.Set;
60
61 public class BuscaGPSActivity extends AppCompatActivity implements
62 CaixaDeDialogoGoogleAPI.ErrorGooglePlayService,
63 ConnectionCallbacks,
64 GoogleApiClient.OnConnectionFailedListener,
65 LocationListener, ResultCallback<LocationSettingsResult>,
66 View.OnClickListener,
67 InterfaceFiltroDeEstabelecimentos {
68
69 public static final int REQUEST_CODE_GOOGLE_PLAY_SERVICES = 1982;
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\BuscaGPSActivity.java

```
71  private static final int REQUEST_CODE_LOCATION = 2;
72  private Toolbar toolbar;
73  private String distanciaMaxima;
74  private Status status;
75  private String dadosDeBuscaNaTeLa;
76  private Location Localizacao;
77  private Double[] latLon;
78  private LocationRequest request;
79  private GoogleApiClient apiClient;
80  private EstabelecimentoDAO estabDAO;
81  private ArrayList<Estabelecimento> estabeleimentos;
82  private ArrayList<Estabelecimento> estabeleimentosFiltrados;
83  private RecyclerView meuRecyclerView;
84  private EstabelecimentoAdapter meuAdapter;
85  private RecycleView.LayoutManager meuLayoutManager;
86  private Set<String> listaDeFavoritos;
87  private Set<String> listaDeMunicipios;
88  private SharedPreferences config;
89  private FloatingActionButton fabFiltroTipoEstab;
90  private FloatingActionButton fabFiltroTipoAtend;
91  private FloatingActionButton fabFiltroTipoConv;
92  private FloatingActionButton fabFiltroTipoServ;
93  private FloatingActionButton fabFiltroTipoProf;
94  private FloatingActionMenu fabFiltroMenu;
95  private FloatingActionButton fabFiltroImparTodos;
96  private ProgressBar barraDeProgresso;
97  private TextView txtErroGps; ImageView imgErroGps;
98  private CarregadorDeDadosBGPSA carregadorDeDados;
99  private int posicaoNaLista;
100 private MenuItem botaoGps;
101 private MenuItem botaoBusca;
102 private ArrayList<Estabelecimento> bkpEstabeleimentos;
103 private ArrayList<String> tiposEstabeleimentosSelecionados;
104 private ArrayList<String> tiposAtendimentosSelecionados;
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\BuscaGPSActivity.java

```
106 private ArrayList<String> tiposConveniosSelecionados;
107 private ArrayList<String> tiposProfissionaisSelecionados;
108 private ArrayList<String> tiposServicosSelecionados;
109 private ArrayList<Integer> filtrosAtivos;
110 private CharSequence[] labelsTiposProfissionais;
111 private CharSequence[] labelsTiposEstabelecimentos;
112 private CharSequence[] labelsTiposAtendimentos;
113 private CharSequence[] labelsTiposConvenios;
114 private CharSequence[] labelsTiposServicos;
115 private boolean coordenadasAlteradas;
116 private CoordinatorLayout layout;
117 //Método que verifica se há na intent uma string armazenada cuja chave é a action de searchview
118 private void executarBuscaPorPalavra(Intent intent) {
119     if (Intent.ACTION_SEARCH.equals(intent.getAction())) {
120         dadosDeBuscaNaTela = intent.getStringExtra(SearchManager.QUERY);
121     }
122 }
123
124 @Override
125 protected void onCreate(Bundle savedInstanceState) {
126     super.onCreate(savedInstanceState);
127     setContentView(R.layout.activity_busca_gps);
128
129     //Instancia os objetos do layout
130     barraDeProgresso = (ProgressBar) findViewById(R.id.barra_progresso_gps_activity);
131     toolbar = (Toolbar) findViewById(R.id.toolbar_actvt_busca_gps);
132     txtErroGps = (TextView) findViewById(R.id.label_erro_gps);
133     imgErroGps = (ImageView) findViewById(R.id.background_erro_gps);
134     fabFiltroTipoEstab = (FloatingActionButton) findViewById(R.id.fab_filtro_tipo_estab);
135     fabFiltroTipoAtend = (FloatingActionButton) findViewById(R.id.fab_filtro_tipo_atend);
136     fabFiltroTipoConv = (FloatingActionButton) findViewById(R.id.fab_filtro_tipo_conven);
137     fabFiltroTipoServ = (FloatingActionButton) findViewById(R.id.fab_filtro_tipo_serv);
138     fabFiltroTipoProf = (FloatingActionButton) findViewById(R.id.fab_filtro_tipo_prof);
139     fabFiltroMenu = (FloatingActionButton) findViewById(R.id.fab_filtro_menu);
```

```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\BuscaGPSActivity.java

141 fabFiltroLimparTodos = (FloatingActionButton) findViewById(R.id.fab_filtro_limpar_todos);
142
143 //configura objetos do layout
144 barraDeProgresso.setVisibility(View.VISIBLE);
145 toolbar.setTitle("Buscar com GPS");
146 setSupportActionBar(toolbar);
147 getSupportActionBar().setDisplayHomeAsUpEnabled(true);
148 txtErrorGps.setVisibility(View.GONE);
149 imgErrorGps.setVisibility(View.GONE);
150 fabFiltroTipoConv.setOnClickListener(this);
151 fabFiltroTipoServ.setOnClickListener(this);
152 fabFiltroTipoProf.setOnClickListener(this);
153 fabFiltroTipoAtend.setOnClickListener(this);
154 fabFiltroTipoEstab.setOnClickListener(this);
155 fabFiltroLimparTodos.setOnClickListener(this);
156 fabFiltroMenu.setOnClosedOnTouchOutside(true);
157 layout = (CoordinatorLayout) findViewById(R.id.activity_main);
158
159 //instancia outros objetos
160 coordenadasAlteradas = false;
161 filtrosAtivos = new ArrayList<Integer>();
162 latLon = new Double[2];
163 dadosDeBuscaNaTela = "";
164 posicaoLista = -1;
165 this.estabelecimentosFiltrados = new ArrayList<Estabelecimento>();
166
167 //restaura os dados, caso hajam, dos filtros que foram salvos anteriormente caso tenha havido alguma ação no
168 dispositivo que tenha posto outra activity sobre a atual
169 if (savedInstanceState != null) {
170     posicaoLista = savedInstanceState.getInt("posicaoNaLista");
171     if (savedInstanceState.getParcelableArrayList("estabelecimentos") != null) {
172         estabelecimentos = savedInstanceState.getParcelableArrayList("estabelecimentos");
173     }
174     if (savedInstanceState.getStringArrayList("tiposEstabelecimentosSelecionados") != null) {
175         tiposEstabelecimentosSelecionados = savedInstanceState.getStringArrayList("tiposEstabelecimentosSelecionados");
176     }
177 }

```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\BuscaGPSActivity.java

```
174 tiposEstabelecimentosSelecionados");  
175 } else {  
176     this.tiposEstabelecimentosSelecionados = new ArrayList<String>();  
177 }  
178 if (savedInstanceState.getIntegerArrayList("filtrosAtivos") != null) {  
179     this.filtrosAtivos = savedInstanceState.getIntegerArrayList("filtrosAtivos");  
180 } else {  
181     this.filtrosAtivos = new ArrayList<Integer>();  
182 }  
183 if (savedInstanceState.getStringArrayList("tiposAtendimentosSelecionados") != null) {  
184     tiposAtendimentosSelecionados = savedInstanceState.getStringArrayList("tiposAtendimentosSelecionados")  
185 ;  
186 } else {  
187     this.tiposAtendimentosSelecionados = new ArrayList<String>();  
188 }  
189 if (savedInstanceState.getStringArrayList("tiposConveniosSelecionados") != null) {  
190     tiposConveniosSelecionados = savedInstanceState.getStringArrayList("tiposConveniosSelecionados");  
191 } else {  
192     this.tiposConveniosSelecionados = new ArrayList<String>();  
193 }  
194 if (savedInstanceState.getStringArrayList("tiposProfissionaisSelecionados") != null) {  
195     tiposProfissionaisSelecionados = savedInstanceState.getStringArrayList("tiposProfissionaisSelecionados"  
196 );  
197 } else {  
198     this.tiposProfissionaisSelecionados = new ArrayList<String>();  
199 }  
200 if (savedInstanceState.getStringArrayList("tiposServicosSelecionados") != null) {  
201     tiposServicosSelecionados = savedInstanceState.getStringArrayList("tiposServicosSelecionados");  
202 } else {  
203     this.tiposServicosSelecionados = new ArrayList<String>();  
204     this.tiposProfissionaisSelecionados = new ArrayList<String>();  
205     this.tiposConveniosSelecionados = new ArrayList<String>();  
206 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\BuscaGPSActivity.java

```
207     this.tiposAtendimentosSelecionados = new ArrayList<String>();
208     this.tiposEstabelecimentosSelecionados = new ArrayList<String>();
209 }
210
211 //metodo chamado para realizar verificações para saber se o dispositivo possui o google play services
212 instalado
213 verificarGooglePlayServiceO;
214
215 //Instancia o objeto da api do google e apartir de dai realiza checagens quanto ao google play services
216 apiClient = new GoogleApiClient.Builder(this)
217     .addConnectionCallbacks(this)
218     .addOnConnectionFailedListener(this)
219     .addApi(LocationServices.API)
220     .build();
221
222 private void mostraQtdeResultadosO {
223     Snackbar.make(layout, "Foram Localizados " + meuAdapter.getItemCount() + " estabelecimentos", Snackbar.LENGTH_SHORT).showO;
224 }
225
226 @Override
227 public boolean onOptionsItemSelected(MenuItem item) {
228     int id = item.getItemIdO;
229     switch (id) {
230         case android.R.id.home:
231             onBackPressedO;
232             break;
233         case R.id.action_gps: fabFiltronMenu.close(false);
234             coordenadasAIteradas =
235                 meuRecyclerView.setVisibility(View.GONE);
236                 fabFiltronMenu.setVisibility(View.GONE);
237                 barraDeProgresso.setVisibility(View.VISIBLE);
238                 botaoBusca.setVisibility(false);
239     }
}
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\BuscaGPSActivity.java

```
240     botaoGps.setVisibility(false);
241
242     }
243
244     return super.onOptionsItemSelected(item);
245
246     //metodo que cria e mostra os dados na lista caso todas as etapas de verificação do correto funcionamento da api
247     de localização do google tenham sido aprovadas
248     private void abrirRecyclerView() {
249
250         //barraDeProgresso.setVisibility(View.VISIBLE);
251         fabFiltroMenu.setVisibility(View.GONE);
252         meuRecyclerView = (RecyclerView) findViewById(R.id.lista_estabelecimentos);
253         meuRecyclerView.setVisibility(View.GONE);    meuLayoutManager = new
254         LinearLayoutManager(this);
255         meuRecyclerView.setLayoutManager(meuLayoutManager);
256         meuRecyclerView.setItemAnimator(new
257         DefaultItemAnimator());
258         meuRecyclerView.setHasFixedSize(true);
259         meuRecyclerView.addOnScrollListener(new RecyclerView.OnScrollListener() {
260
261             @Override
262             public void onScrolled(RecyclerView recyclerView, int newState) {
263                 super.onScrolled(recyclerView, newState);
264                 Log.e("Scrol", String.valueOf(newState));
265             }
266
267             @Override
268             public void onScrolled(RecyclerView recyclerView, int dx, int dy) {
269                 super.onScrolled(recyclerView, dx, dy);
270                 if (dy > 0) {
271                     fabFiltroMenu.hideMenuButton(true);
272                 } else {
273                     fabFiltroMenu.showMenuButton(true);
274                 }
275             }
276         });
277     }
278 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\BuscaGPSActivity.java

```
274     }
275   );
276   executarBuscaPorPallavra(getIntent());
277   carregarDados();
278   fabFiltroMenu.showMenuItemOn(true);
279 }

280 private void carregarDados() {
281   carregadorDeDados = new CarregadorDeDadosBGPSA(this);
282   carregadorDeDados.execute();
283 }

284 @Override
285 protected void onSaveInstanceState(Bundle outState) {
286   super.onSaveInstanceState(outState);
287   outState.putParcelableArrayList("estabelecimentos", estabelecimentos);
288   outState.putStringArrayList("tiposEstabelecimentosSelecionados", tiposEstabelecimentosSelecionados);
289   outState.putStringArrayList("tiposAtendimentosSelecionados", tiposAtendimentosSelecionados);
290   outState.putStringArrayList("tiposConveniosSelecionados", tiposConveniosSelecionados);
291   outState.putStringArrayList("tiposProfissionaisSelecionados", tiposProfissionaisSelecionados);
292   outState.putStringArrayList("tiposServicosSelecionados", tiposServicosSelecionados);
293   outState.putIntArrayList("filtradosAtivos", filtrosAtivos);
294   outState.putInt("posicaoNaLista", posicaoNaLista);
295 }

296 private void verificarGooglePlayService() {
297   GoogleApiClientability api = GoogleApiClientability.getInstance();
298   int code = api.isGooglePlayServicesAvailable(this);
299   if (code == ConnectionResult.SUCCESS) {
300     //da sequencia nos procedimento normalmente
301   } else {
302     if (code == ConnectionResult.SERVICE_MISSING || code == ConnectionResult.SERVICE_VERSION_UPDATE_REQUIRED || code == ConnectionResult.SERVICE_DISABLED) {
303       Dialog dialog = api.getErrorDialog(this, code, REQUEST_CODE_GOOGLE_PLAY_SERVICES);
304     }
305   }
306 }
307 ...
308 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\BuscaGPSActivity.java

```
309     dialog.show();
310 
311     } else {
312         CaixaDeDialogoGoogLeAPI cxDialog = new CaixaDeDialogoGoogLeAPI();
313         cxDialog.setErroGooglePlayService(this);
314         cxDialog.show(getFragmentManager(), "DialogError");
315     }
316 }
317 
318 @Override
319 protected void onNewIntent(Intent intent) {
320     super.onNewIntent(intent);
321     executarBuscaPorPalavra(intent);
322     carregarDados();
323 }
324 
325 @Override
326 public boolean onCreateOptionsMenu(Menu menu) {
327     getMenuInflater = getMenuInflater();
328     getMenuInflater.inflate(R.menu.menu_consulta_gps,
329     menu);
330     MenuItem searchItem = menu.findItem(R.id.action_search);
331     botaoBusca = searchItem;
332     SearchManager searchManager = (SearchManager) BuscaGPSActivity.this.getSystemService(Context.SEARCH_SERVICE);
333     SearchView searchView = null;
334     if (searchItem != null) {
335         searchView = (SearchView) searchItem.getActionView();
336     }
337     if (searchView != null) {
338         searchView.setSearchableInfo(SearchManager.getSearchableInfo(BuscaGPSActivity.this.getComponentName()));
339     }
340     MenuItemCompat.OnActionExpandListener expandListener = new MenuItemCompat.OnActionExpandListener() {
341         @Override
342         public boolean onMenuItemActionCollapse(MenuItem item) {
343             botaoGps.setVisible(true);
344         }
345     };
346 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\BuscaGPSActivity.java

```
344     dadosDeBuscaNaTela = "...";
345     fabFiltromenu.close(true);
346     carregarDadosO;
347     return true;
348 }
349
350     @Override
351     public boolean onMenuItemActionExpand(MenuItem item) {
352         botaogps.setVisible(false);
353         fabFiltromenu.close(true);
354         return true;
355     }
356     MenuItemCompat.setOnActionExpandListener(searchItem, expandListener);
357     return super.onCreateOptionsMenu(menu);
358 }
359
360
361     @Override
362     protected void onStart() {
363         super.onStart();
364         apiClient.connectO;
365     }
366
367     @Override
368     public void onBackPressed()
369         barraDeProgresso.setVisibility(View.GONE);
370         if (fabFiltromenu.isOpenedO {
371             fabFiltromenu.toggle(true);
372         } else {
373             if (carregadorDeDados != null) {
374                 carregadorDeDados.cancel(true);
375             }
376             super.onBackPressed();
377         }
378     }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\BuscaGPSActivity.java

```
379 }
380
381     @Override
382     protected void onStop() {
383         super.onStop();
384         apiClient.disconnect();
385     }
386
387     @Override
388     protected void onPause() {
389         super.onPause();
390         if (meuRecyclerView != null) {
391             LinearLayoutManager lm = (LinearLayoutManager) meuRecyclerView.getLayoutManager();
392             posicaoNaLista = lm.findFirstVisibleItemPosition();
393         }
394     }
395
396     @Override
397     public void onConnected(Bundle bundle) {
398         LocationRequest.create();
399         request.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY);
400         request.setInterval(500);
401         LocationSettingsRequest.Builder builder = new LocationSettingsRequest.Builder();
402         builder.setAlwaysShow(true);
403         PendingResult<LocationSettingsResult> result = LocationServices.SettingsApi.checkLocationSettings(apiClient,
404             builder.build());
405         result.setResultCallback(this);
406     }
407
408     public void onRequestPermissionsResult(int requestCode, String[] permissions, int[] grantResults) {
409         if (requestCode == REQUEST_CODE_LOCATION) {
410             if (grantResults.length == 1 && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
411                 adquirirLocalizacao();
412             } else {
413             }
414         }
415     }
416 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\BuscaGPSActivity.java

```
413     barraDeProgresso.setAlpha(0f);
414     imgErroGps.setVisibility(View.VISIBLE);
415     txtErroGps.setVisibility(View.VISIBLE);
416     Snackbar.make(fabFiltroMenu, R.string.tentar_novamente_label, Snackbar.LENGTH_INDEFINITE)
417         .setAction(R.string.tentar_novamente_action, new View.OnClickListener() {
418             @Override
419             public void onClick(View v) {
420                 adquirirLocalizacao();
421             }
422         })
423         .setActionTextColor(Color.YELLOW)
424         .show();
425     }
426 }
427 }
428 private void adquirirLocalizacao() {
429     if (ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.
430         PERMISSION_GRANTED) {
431         if (ActivityCompat.shouldShowRequestPermissionRationale(this, Manifest.permission.ACCESS_FINE_LOCATION))
432             ActivityCompat.requestPermissions(this, new String[]{Manifest.permission.ACCESS_FINE_LOCATION},
433             REQUEST_CODE_LOCATION);
434     }
435     } else {
436         LocationServices.FusedLocationApi.requestLocationUpdates(apiClient, request, this);
437     }
438 }
439 @Override
440 public void onConnectionSuspended(int i) {
441     //Não implementado
442 }
443 }
444 @Override
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\BuscaGPSActivity.java

```
446  public void onConnectionFailed(ConnectivityManager connectionResult) {
447      //Não implementado
448  }
449
450     @Override
451     public void onLocationChanged(Location location) {
452         this.localizacao = location;
453         latLon[0] = location.getLatitude();
454         latLon[1] = location.getLongitude();
455         Log.e("Coordenadas", latLon[0] + " " + latLon[1]);
456         LocationServices.FusedLocationApi.removeLocationUpdates(apiClient, this);
457         abrirRecyclerView();
458     }
459
460     @Override
461     protected void onActivityResult(int requestCode, Intent data) {
462         final LocationSettingsStates states;
463         if (data != null) {
464             states = LocationServices.SettingsStates.fromIntent(data);
465         }
466         switch (requestCode) {
467             case REQUEST_CODE_GOOGLPLAY_SERVICES:
468                 switch (resultCode) {
469                     case Activity.RESULT_OK:
470                         barraDeProgresso.setAlpha(1f);
471                         barraDeProgresso.setVisibility(View.VISIBLE);
472                         imgErroGps.setVisibility(View.GONE);
473                         txtErroGps.setVisibility(View.GONE);
474                         adquirirLocalizacao();
475                     break;
476                     case Activity.RESULT_CANCELED:
477                         barraDeProgresso.setAlpha(0f);
478                         imgErroGps.setVisibility(View.VISIBLE);
479                         txtErroGps.setVisibility(View.GONE);
480                         Snackbar.make(fabFiltroMenu, R.string.tentar_novamente_label, Snackbar.LENGTH_INDEFINITE)
481                         .show();
482                 }
483             }
484         }
485     }
486 }
```

```
File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\BuscaGPSActivity.java
481     .setAction(R.string.tentar_novamente_action, new View.OnClickListener() {
482
483         @Override
484         public void onClick(View v) {
485             try {
486                 REQUEST_CODE_GOOGLE_PLAY_SERVICES);
487             } catch (IntentSender.SendIntentException e) {
488                 status.startResolutionForResult(BuscaGPSActivity.this,
489
490             }
491             .setActionBarColor(Color.YELLOW)
492             .showO;
493         }
494         break;
495     }
496 }
497     @Override
498     public void onActivityResult(LocationSettingsResult locationSettingsResult) {
499         status = LocationSettingsResult.getStatusO;
500         switch (status.getStatusCodeO) {
501             case LocationSettingsStatusCodes.SUCCESS:
502                 adquirirLocalizacaoO;
503                 break;
504             case LocationSettingsStatusCodes.RESOLUTION_REQUIRED:
505                 try {
506                     status.startResolutionForResult(BuscaGPSActivity.this, REQUEST_CODE_GOOGLE_PLAY_SERVICES);
507                 } catch (IntentSender.SendIntentException e) {
508                     Snackbar.make(findViewById(android.R.id.content), "Algun motivo desconhecido impediu a aquisição
509                     de sua localização", Snackbar.LENGTH_LONG).setActionTextColor(Color.RED).showO;
510
511                 break;
512             case
513                 Snackbar.make(findViewById(android.R.id.content), "Algun motivo desconhecido impediu a aquisição de
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\BuscaGPSActivity.java

```
513 sua localização”, Snackbar.LENGTH_LONG).setActionTextColor(Color.RED).show();
514
515     }
516
517     @Override
518     public void onClick(View v) {
519         definirLabelsParaFiltros();
520         int id = v.getId();
521         switch (id) {
522             case R.id.fab_filtro_todos:
523                 fabFiltroMenu.toggle(true);
524                 limparTodosFiltros();
525                 break;
526             case R.id.fab_filtro_tipo_atend:
527                 fabFiltroMenu.toggle(true);
528                 FiltroGeneralDialog dialogFiltroAtendimento = new FiltroGeneralDialog();
529                 dialogFiltroAtendimento.setArgumentos(tiposAtendimentosSelecionados,
530                     InterfaceFiltroDeEstabelecimentos.FILTRO_TIPO_ATENDIMENTO, labelsTiposAtendimentos);
531                 dialogFiltroAtendimento.show(getFragmentManager(), "DialogFiltroTipoAtendimento");
532                 break;
533             case R.id.fab_filtro_tipo_conven:
534                 fabFiltroMenu.toggle(true);
535                 FiltroGeneralDialog dialogFiltroConvenio = new FiltroGeneralDialog();
536                 dialogFiltroConvenio.setArgumentos(tiposConveniosSelecionados, InterfaceFiltroDeEstabelecimentos.
537                     FILTRO_TIPO_CONVENIO, labelsTiposConvenios);
538                 dialogFiltroConvenio.show(getFragmentManager(), "DialogFiltroTipoConvenio");
539                 break;
540             case R.id.fab_filtro_tipo_prof:
541                 fabFiltroMenu.toggle(true);
542                 FiltroGeneralDialog dialogFiltroProfissional = new FiltroGeneralDialog();
543                 dialogFiltroProfissional.setArgumentos(tiposProfissionaisSelecionados,
544                     InterfaceFiltroDeEstabelecimentos.FILTRO_TIPO_PROFESSIONAL, labelsTiposProfissionais);
545                 dialogFiltroProfissional.show(getFragmentManager(), "DialogFiltroTipoProfissional");
546             
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\BuscaGPSActivity.java

```
545     case R.id.fab_filtro_tipo_serv:
546         fabFiltroMenu.toggle(true);
547         FiltroGeneralDialog dialogFiltroTipoServiço = new FiltroGeneralDialog();
548         dialogFiltroTipoServiço.setArguments(tiposServicosSelecionados, InterfaceFiltroDeEstabelecimentos.*;
549             FILTRO_TIPO_SERVICO, labelsTiposServicos);
550         dialogFiltroTipoServiço.show(getFragmentManager(), "DialogFiltroTipoServiço");
551         break;
552     case R.id.fab_filtro_tipo_estab:
553         fabFiltroMenu.toggle(true);
554         FiltroGeneralDialog dialogFiltroEstabelecimento = new FiltroGeneralDialog();
555         InterfaceFiltroDeEstabelecimentos dialogFiltroEstabelecimento.setArguments(tiposEstabelecimentosSelecionados,
556             dialogFiltroEstabelecimento.FILTRO_TIPO_ESTABELECIMENTO, labelsTiposEstabelecimentos);
557         dialogFiltroEstabelecimento.show(getFragmentManager(), "DialogFiltroEstabelecimento");
558         break;
559     }
560     @Override
561     public void filtrarPorTipoDeEstabelecimento(ArrayList<String> tiposEstabelecimentosSelecionados) {
562         this.tiposEstabelecimentosSelecionados =
563             estabelecimentosTemp =
564                 new ArrayList<Estabelecimento>();
565         if (filtrosAtivos.size() > 0) {
566             estabelecimentosTemp = this.estabelecimentosFiltrados;
567         } else {
568             estabelecimentosTemp = this.estabelecimentos;
569         }
570         if (this.tiposEstabelecimentosSelecionados.size() > 0) {
571             fabFiltroMenu.setMenuItemColorPressedResId(R.color.accent);
572             fabFiltroMenu.setMenuItemColorNormalResId(R.color.primary);
573             fabFiltroEstab.setColorNormalResId(R.color.primary);
574             fabFiltroEstab.setColorPressedResId(R.color.accent);
575         }
576         for (Estabelecimento e : estabelecimentosTemp) {
577             if (this.tiposEstabelecimentosSelecionados.contains(e.getTipoEstabelecimento())) {
```

```
578         estabeleimentosFiltradosTemp.add(e);
579     }
580     meuAdapter = new EstabelecimentoAdapter(this, estabeleimentosFiltradosTemp, listaDeFavoritos,
581     distanciaMaxima);
582     meuRecyclerView.invalidate();
583     meuRecyclerView.setAdapter(meuAdapter);
584     if (!filtradosAtivos.contains(InterfaceFiltroDeEstabelecimentos.FILTRO_TIPO_ESTABELECIMENTO)) {
585         filtradosAtivos.add(InterfaceFiltroDeEstabelecimentos.FILTRO_TIPO_ESTABELECIMENTO);
586         botaoBusca.setVisible(false);
587     }
588     this.estabeleimentosFiltrados = estabeleimentosFiltradosTemp;
589     definirLabelsParaFiltrosQ;
590     mostraQtdeResultadosO;
591 }
592 }
593 private void restaurarFiltros(int filtro) {
594     switch (filtro) {
595         case InterfaceFiltroDeEstabelecimentos.FILTRO_TIPO_ATENDIMENTO:
596             filtrarPorTipoDeEstabelecimento(this.tiposEstabelecimentosSelecionados);
597             break;
598         case InterfaceFiltroDeEstabelecimentos.FILTRO_TIPO_SERVICO:
599             filtrarPorTipoDeServiço(this.tiposServicosSelecionados);
600             break;
601         case InterfaceFiltroDeEstabelecimentos.FILTRO_TIPO_ESTABELECIMENTO:
602             filtrarPorTipoDeEstabelecimento(this.tiposEstabelecimentosSelecionados);
603             break;
604         case InterfaceFiltroDeEstabelecimento(this.tiposProfissionaisSelecionados);
605             filtrarPorTipoDeProfissional(this.tiposProfissionaisSelecionados);
606             break;
607         case InterfaceFiltroDeEstabelecimento(this.tiposConveniosSelecionados);
608             filtrarPorTipoDeConvenio(this.tiposConveniosSelecionados);
609             break;
610     }
611 }
```

```
612
613
614     @Override
615     public void filtrarPorTipoDeAtendimento(ArrayList<String> tiposAtendimentosSelecionados) {
616         ArrayList<Estabelecimento> estabelecimentosTemp = new ArrayList<Estabelecimento>();
617         ArrayList<Estabelecimento> estabelecimentosFiltradosTemp = new ArrayList<Estabelecimento>();
618         int compatibilidade = 0;
619         this.tiposAtendimentosSelecionados = tiposAtendimentosSelecionados;
620         if (estabelecimentosFiltrados.size() > 0) {
621             estabelecimentosTemp = this.estabelecimentosFiltrados;
622         } else {
623             estabelecimentosTemp = this.estabelecimentos;
624         }
625         if (this.tiposAtendimentosSelecionados.size() > 0) {
626             for (Estabelecimento e : estabelecimentosTemp) {
627                 for (Atendimento a : e.getAtendimento()) {
628                     if (this.tiposAtendimentosSelecionados.contains(a.getTipoAtendimento())) {
629                         compatibilidade++;
630                     }
631                 }
632                 if (compatibilidade > 0) {
633                     fabFiltroMenu.setMenuItemColorPressedResId(R.color.accent);
634                     fabFiltroMenu.setMenuItemNormalResId(R.color.primary);
635                     fabFiltrotipoAtend.setColorNormalResId(R.color.primary);
636                     fabFiltrotipoAtend.setColorPressedResId(R.color.accent);
637                     estabelecimentosFiltradosTemp.add(e);
638                 }
639             }
640             meuAdapter = new EstabelecimentoAdapter(this, estabelecimentosFiltradosTemp, listaDeFavoritos,
641             distanciaMaxima);
642             meuRecyclerView.invalidate();
643             meuRecyclerView.setAdapter(meuAdapter);
644             if (!filtrosAtivos.contains(FILTRO_TIPO_ATENDIMENTO)) {
645                 filtrosAtivos.add(FILTRO_TIPO_ATENDIMENTO);
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\BuscaGPSActivity.java

```
646         botaoBusca.setVISIBLE(false);
647     }
648     this.estabelecimentosFiltrados = estabelecimentosFiltradosTemp;
649     definirLabelsParaFiltrosO;
650     mostraQtdeResultadosO;
651 }
652 }

653 private void definirLabelsParaFiltrosO {
654     ArrayList<Estabelecimento> estabelecimentosFiltradosTemp = new ArrayList<Estabelecimento>();
655     if (estabelecimentosFiltrados.size() > 0) {
656         estabelecimentosFiltradosTemp = this.estabelecimentosFiltrados;
657     } else {
658         estabelecimentosFiltradosTemp = this.estabelecimentos;
659     }
660     ArrayList<CharSequence> tempLabelsTiposAtendimentos = new ArrayList<CharSequence>();
661     ArrayList<CharSequence> tempLabelsTiposProfissionais = new ArrayList<CharSequence>();
662     ArrayList<CharSequence> tempLabelsTiposConvenios = new ArrayList<CharSequence>();
663     ArrayList<CharSequence> tempLabelsTiposServicos = new ArrayList<CharSequence>();
664     ArrayList<CharSequence> tempLabelsTiposEstabelecimentos = new ArrayList<CharSequence>();
665     for (Estabelecimento e : estabelecimentosFiltradosTemp) {
666         if (!tempLabelsTiposEstabelecimentos.contains(e.getTipoEstabelecimento())) {
667             tempLabelsTiposEstabelecimentos.add(e.getTipoEstabelecimento());
668         }
669         for (Atendimento a : e.getAtendimentoO) {
670             if (!tempLabelsTiposAtendimentos.contains(a.getTipoAtendimentoO)) {
671                 tempLabelsTiposAtendimentos.add(a.getTipoAtendimentoO);
672             }
673             if (!tempLabelsTiposConvenios.contains(a.getConvenioO)) {
674                 tempLabelsTiposConvenios.add(a.getConvenioO);
675             }
676         }
677         for (ServiçoClassificacao sc : e.getServicosO) {
678             if (!tempLabelsTiposServicos.contains(sc.getDescricaoServiçoO)) {
679                 tempLabelsTiposServicos.add(sc.getDescricaoServiçoO);
680             }
681         }
682     }
683 }
```

```
681
682    }
683    for (Profissional p : e.getProfissionaisO) {
684        if (!tempLabelsTiposProfissionais.contains(p.getChodescricaoO)) {
685            tempLabelsTiposProfissionais.add(p.getChodescricaoO);
686        }
687    }
688    this.labelsTiposConvenios = tempLabelsTiposConvenios.toArray(new CharSequence[0]);
689    this.labelsTiposProfissionais = tempLabelsTiposProfissionais.toArray(new CharSequence[0]);
690    this.labelsTiposServicos = tempLabelsTiposServicos.toArray(new CharSequence[0]);
691    this.labelsTiposEstabelecimentos = tempLabelsTiposEstabelecimentos.toArray(new CharSequence[0]);
692    this.labelsTiposAtendimentos = tempLabelsTiposAtendimentos.toArray(new CharSequence[0]);
693    if (labelsTiposAtendimentos.length < 2) {
694        fabFiltroAtend.setEnabled(false);
695    } else {
696        fabFiltroAtend.setEnabled(true);
697    }
698    if (labelsTiposEstabelecimentos.length < 2) {
699        fabFiltroTipoEstab.setEnabled(false);
700    } else {
701        fabFiltroTipoEstab.setEnabled(true);
702    }
703    if (labelsTiposProfissionais.length < 2) {
704        fabFiltroTipoProf.setEnabled(false);
705    } else {
706        fabFiltroTipoProf.setEnabled(true);
707    }
708    if (labelsTiposServicos.length < 2) {
709        fabFiltroTipoServ.setEnabled(false);
710    } else {
711        fabFiltroTipoServ.setEnabled(true);
712    }
713    if (labelsTiposConvenios.length < 2) {
714        fabFiltroTipoConv.setEnabled(false);
715    }
716}
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\BuscaGPSActivity.java

```
716     } else {
717         fabFiltroTipoConv.setEnabled(true);
718     }
719     if (estabelecimentosFiltrados.size() == 1) {
720         fabFiltroTipoConv.setEnabled(false);
721         fabFiltroTipoServ.setEnabled(false);
722         fabFiltroTipoProf.setEnabled(false);
723         fabFiltroTipoEstab.setEnabled(false);
724         fabFiltroTipoAtend.setEnabled(false);
725     }
726     Arrays.sort(labelstiposEstabelecimentos);
727     Arrays.sort(labelstiposAtendimentos);
728     Arrays.sort(labelstiposConvenios);
729     Arrays.sort(labelstiposServicos);
730     Arrays.sort(labelstiposProfissionais);
731 }
732
733 @Override
734 public void limparTodosFiltros()
735 {
736     this.tiposServicosSelecionados.clear();
737     this.tiposProfissionaisSelecionados.clear();
738     this.tiposEstabelecimentosSelecionados.clear();
739     this.tiposAtendimentosSelecionados.clear();
740     this.estabelecimentosFiltrados.clear();
741     meuRecyclerView.setAdapter(meuAdapter);
742     filtrosAtivos.clear();
743     definirLabelsParaFiltros();
744     mostraQtdResultados();
745     botaoBusca.setVisibility(true);
746     fabFiltroTipoEstab.setNormalResId(R.color.fab_normal);
747     fabFiltroTipoAtend.setNormalResId(R.color.fab_normal);
748     fabFiltroTipoConv.setNormalResId(R.color.fab_normal);
749     fabFiltroTipoServ.setNormalResId(R.color.fab_normal);
--
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\BuscaGPSActivity.java

```
751     fabFilTroTipoprof.setColorNormalResId(R.color.fab_normal);
752     fabFilTroMenu.setMenuItemColorNormalResId(R.color.fab_normal);
753     fabFilTroTipoestab.setColorPressedResId(R.color.fab_pressed);
754     fabFilTroTipoAtend.setColorPressedResId(R.color.fab_pressed);
755     fabFilTroTipoConv.setColorPressedResId(R.color.fab_pressed);
756     fabFilTroTipoServ.setColorPressedResId(R.color.fab_pressed);
757     fabFilTroTipoprof.setColorPressedResId(R.color.fab_pressed);
758     fabFilTroMenu.setMenuItemColorPressedResId(R.color.fab_pressed);
759 }
760
761 @Override
762 public void filtrarPorTipoDeConvenio(ArrayList<String> tiposConveniosSeleccionados) {
763     this.tiposConveniosSeleccionados = tiposConveniosSeleccionados;
764     int compatibilidade = 0;
765     ArrayList<Estabelecimento> estabelecimentosTemp = null;
766     ArrayList<Estabelecimento> estabelecimentosTempFiltrados = new ArrayList<Estabelecimento>;
767     if (filtrosAtivos.size() > 0) {
768         estabelecimentosTemp = estabelecimentosFiltrados;
769     } else {
770         estabelecimentosTemp = estabelecimentos;
771     }
772     if (this.tiposConveniosSeleccionados.size() > 0) {
773         for (Estabelecimento e : estabelecimentosTemp) {
774             for (Atendimento a : e.getAtendimentoO) {
775                 if (this.tiposConveniosSeleccionados.contains(a.getConvenioO)) {
776                     compatibilidade++;
777                 }
778             }
779         }
780         if (compatibilidade > 0) {
781             fabFilTroMenu.setMenuItemColorPressedResId(R.color.accent);
782             fabFilTroMenu.setMenuItemColorNormalResId(R.color.primary);
783             fabFilTroTipoConv.setColorNormalResId(R.color.primary);
784             fabFilTroTipoConv.setColorPressedResId(R.color.accent);
785             estabelecimentosTempFiltrados.add(e);
786             compatibilidade = 0;
787         }
788     }
789 }
```

File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\BuscaGPSActivity.java

```
786
787     }
788     meuAdapter = new EstabelecimentoAdapter(this, estabelecimentosTempFiltrados, listaDeFavoritos,
789     distanciaMaxima);
790     meuRecyclerView.invalidate();
791     meuRecyclerView.setAdapter(meuAdapter);
792     if (!filtrosAtivos.contains(InterfaceFiltroDeEstabelecimentos.FILTRO_TIPO_CONVENIO)) {
793         filtrosAtivos.add(InterfaceFiltroDeEstabelecimentos.FILTRO_TIPO_CONVENIO);
794         botaоЁbusca.setVisible(false);
795     }
796     this.estabelecimentosFiltrados = estabelecimentosTempFiltrados;
797     definirLabelsParaFiltros();
798 }
799
800 @Override
801 public void filtrarPorTipoDeProfissional(ArrayList<String> tiposProfissionaisSelecionados) {
802     this.tiposProfissionaisSelecionados = tiposProfissionaisSelecionados;
803     int compatibilidade = 0;
804     ArrayList<Estabelecimento> estabelecimentosTemp = null;
805     ArrayList<Estabelecimento> estabelecimentosTempFiltrados = new ArrayList<Estabelecimento>();
806     if (filtrosAtivos.size() > 0) {
807         estabelecimentosTemp = estabelecimentosFiltrados;
808     } else {
809         estabelecimentosTemp = estabelecimentos;
810     }
811     if (this.tiposProfissionaisSelecionados.size() > 0) {
812         for (Estabelecimento e : estabelecimentosTemp) {
813             for (Profissional p : e.getProfissionais()) {
814                 if (this.tiposProfissionaisSelecionados.contains(p.getChqDescricaoO)) {
815                     compatibilidade++;
816                 }
817             }
818         }
819     }
820 }
```

```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\BuscaGPSActivity.java

820     fabFilroMenu.setMenuItemColorPressedResId(R.color.accent);
821     fabFilroMenu.setMenuItemColorNormalResId(R.color.primary);
822     fabFilroTipoProf.setColorNormalResId(R.color.primary);
823     fabFilroTipoProf.setColorPressedResId(R.color.accent);
824     estabelecimentosTempFiltrados.add(e);
825     compatibilidade = 0;
826 }
827 meuAdapter = new EstabelecimentoAdapter(this, estabelecimentosTempFiltrados, listaDeFavoritos,
828 distanciaMaxima);
829 meuRecyclerView.invalidate();
830 meuRecyclerView.setAdapter(meuAdapter);
831 if (!filtrosAtivos.contains(InterfaceFilroDeEstabelecimentos.FILTRO_TIPO_PROFISSIONAL)) {
832     filtrosAtivos.add(InterfaceFilroDeEstabelecimentos.FILTRO_TIPO_PROFISSIONAL);
833     botaoBusca.setVisible(false);
834 }
835 this.estabelecimentosFiltrados = estabelecimentosTempFiltrados;
836 definirLabelsParaFilros();
837 mostraQtdeResultados();
838 }
839 }
840 @Override
841 public void filtrarPorTipoDeServiço(ArrayList<String> tiposServicosSelecionados) {
842     this.tiposServicosSelecionados = tiposServicosSelecionados;
843     int compatibilidade = 0;
844     ArrayList<Estabelecimento> estabelecimentosTemp = null;
845     ArrayList<Estabelecimento> estabelecimentosTempFiltrados = new ArrayList<Estabelecimento>();
846     if (filtrosAtivos.size() > 0) {
847         estabelecimentosTemp = estabelecimentosFiltrados;
848     } else {
849         estabelecimentosTemp = estabelecimentos;
850     }
851     if (this.tiposServicosSelecionados.size() > 0) {
852         for (Estabelecimento e : estabelecimentosTemp) {
853             if (e.getTipoDeServiço().equals(this.tiposServicosSelecionados.get(0))) {
854                 estabelecimentosTemp.add(e);
855             }
856         }
857     }
858 }

```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\BuscaGPSActivity.java

```
854     for (ServiçoClassificacao sc : e.getServicosO) {
855         if (this.tiposServicosSelecionados.contains(sc.getDescricaoServicoO)) {
856             compatibilidade++;
857         }
858     }
859     if (compatibilidade > 0) {
860         fabFiltroMenu.setMenuItemColorPressedResId(R.color.accent);
861         fabFiltroMenu.setMenuItemColorNormalResId(R.color.primary);
862         fabFiltroTipoServ.setMenuItemColorNormalResId(R.color.primary);
863         fabFiltroTipoServ.setMenuItemColorPressedResId(R.color.accent);
864         estabelecimentosTempFiltrados.add(e);
865         compatibilidade = 0;
866     }
867 }
868 meuAdapter = new EstabelecimentoAdapter(this, estabelecimentosTempFiltrados, listaDeFavoritos,
869 distanciaMaxima);
870 meuRecyclerView.invalidate();
871 meuRecyclerView.setAdapter(meuAdapter);
872 if (!filtrosAtivos.contains(InterfaceFiltrosEstabelecimentos.FILTRO_TIPO_SERVICO)) {
873     filtrosAtivos.add(InterfaceFiltrosEstabelecimentos.FILTRO_TIPO_SERVICO);
874     botaоЁbusca.setVisible(false);
875 }
876 this.estabelecimentosFiltrados = estabelecimentosTempFiltrados;
877 definirLabelsParaFiltrosO;
878 mostraQtdeResultadosO;
879 }
880 }
881 @Override
882 protected void onResume() {
883     super.onResume();
884     if (meuRecyclerView != null) {
885         meuRecyclerView.setVisibility(View.GONE);
886     }
887 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\BuscaGPSActivity.java

```
888     fabFilterMenu.setVisibility(View.GONE);
889     barraDeProgresso.setVisibility(View.VISIBLE);
890 }
891
892     @Override
893     public void googlePlayServiceIndisponivelO {
894         onBackPressedO;
895     }
896
897     private class CarregadorDeDadosBGPSA extends AsyncTask<Void, Void, Void> {
898         private Context contexto; ArrayList<Profissional>
899         listaDeProfissionais;
900         ArrayList<ServicoClassificacao> listaDeServicos;
901         ArrayList<Atendimento> listaDeAtendimento;
902         ProfissionaisDAO profissionaisDAO;
903         ServicoClassificacaoDAO servicoClassificacaoDAO;
904         AtendimentoDAO atendimentoDAO;
905
906         public CarregadorDeDadosBGPSA(Context contexto) {
907             this.contexto = contexto;
908         }
909
910         @Override
911         protected void doInBackground(Void... params) {
912             Log.e("Estabelecimentos", "Carregando dados em background");
913             config = getSharedPreferences("favoritos", MODE_PRIVATE);
914             listaDeFavoritos = config.getStringSet("ListaDefavoritos", new HashSet<String>());
915             distanciaMaxima = String.valueOf(PreferenceManager.getDefaultSharedPreferences(contexto).getInt(getString(
916                 R.string.pref_distancia_estabelecimentos_key), 1));
917             if (QbkEstabelecimentos == null || coordenadasAlteradas == true) && isCancelled() == false {
918                 Log.e("Canceleamento", "1");
919                 estabDAO = EstabelecimientoDAO.getInstance(getApplicationContext());
920                 if (isCancelled() == false) {
921                     Log.e("Canceleamento", "2");
922                     estabelecimentos = estabDAO.buscaEstabelecimentosPorGPS(localizacao, distanciaMaxima);
```

```

922
923     }
924     if (isCancelledO == false) {
925         Log.e("Cancelamento", "3");
926         Log.e("Estabelecimentos", "Estabelecimentos resultantes: " + String.valueOf(estabeleimentos.size(
927             )));
928         profissionaisDAO = ProfissionaisDAO.getInstancia(getApplicationContext());
929         listaDeProfissionais = new ArrayList<Profissional>();
930         servicoClassificacaoDAO = ServicoClassificacaoDAO.getInstancia(getApplicationContext());
931         listaDeServicos = new ArrayList<ServiçoClassificado>();
932         atendimentoDAO = AtendimentoDAO.getInstancia(getApplicationContext());
933         listaDeAtendimento = new ArrayList<Atendimento>();
934         if (isCancelledO == false) {
935             Log.e("Cancelamento", "4");
936             for (Estabelecimiento e : estabeleimentos) {
937                 if (isCancelledO) {
938                     Log.e("Cancelamento", "Cancelado no for de estabeleimentos");
939                     break;
940                 } else {
941                     listaDeProfissionais = profissionaisDAO.listaProfissionais(String.valueOf(e.getChesO));
942                     listaDeServicos = servicoClassificacaoDAO.listarServicosEspecializados(String.valueOf(e.
943                         getChesO));
944                     listaDeAtendimento = atendimentoDAO.ListarAtendimentos(String.valueOf(e.getChesO));
945                     e.setProfissionais(listaDeProfissionais);
946                     e.setServicos(listaDeServicos);
947                     e.setAtendimento(listaDeAtendimento);
948                 }
949             }
950             bkpEstabelecimentos = estabelecimentos;
951             coordenadasAlteradas = false;
952             if (dadosDeBuscaNaTela.trim().length() > 0 && isCancelledO == false) {
953                 Log.e("Cancelamento", "5");
954                 dadosDeBuscaNaTela = StringUtils.removerAcentos(dadosDeBuscaNaTela);

```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\BuscaGPSActivity.java

```
955     dadosDeBuscaNaTela = dadosDeBuscaNaTela.trim();
956     ArrayList<Estabelecimento> estabelecimentosTemp = new ArrayList<Estabelecimento>;
957     for (Estabelecimento e : bkpEstabelecimentos) {
958         if (e.getNomeFantasiaO.toLowerCase().contains(dadosDeBuscaNaTela.toLowerCase()) || e.
959             getRazaoSocialO.toLowerCase().contains(dadosDeBuscaNaTela)) {
960                 estabelecimentosTemp.add(e);
961             }
962         estabelecimentos = estabelecimentosTemp;
963         definirLabelsParaFiltersO();
964     } else {
965         estabelecimentos = bkpEstabelecimentos;
966     }
967     return null;
968 }
969 @Override
970 protected void onCancelled(Void aVoid) {
971     super.onCancelled(aVoid);
972     if (estabelecimentos != null) {
973         Log.e("Cancelamento", String.valueOf(estabelecimentos.size()));
974         estabelecimentos.clear();
975     }
976     if (estabelecimentosFiltrados != null)
977         estabelecimentosFiltrados.clear();
978     if (bkpEstabelecimentos != null)
979         bkpEstabelecimentos.clear();
980     barraDeProgresso.setVisibility(View.GONE);
981     if (estabelecimentos != null)
982         Log.e("Cancelamento", String.valueOf(estabelecimentos.size()));
983     Log.e("Cancelamento", "onCancelled");
984 }
985 @Override
986 protected void onPostExecute(Void aVoid) {
987 ...
988 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\BuscaGPSActivity.java

```
989     super.onPostExecute(aVoid);
990     Log.e("Cancelamento", "onPostExecuteChamado");
991     if (filtrosAtivos.size() > 0) {
992         botaoBusca.setVisible(false);
993         Log.e("Estabelecimentos", String.valueOf(estabelecimentos.size()));
994         for (Integer i : filtrosAtivos) {
995             restaurarFiltros(i);
996         }
997     } else {
998         botaoBusca.setVisibility(true);
999     }
1000     meuAdapter = new EstabelecimentoAdapter(contexto, estabelecimentos, listaDeFavoritos, distanciaMaxima);
1001     meuRecyclerView.setAdapter(meuAdapter);
1002     if (posicaoNaLista > -1) {
1003         meuRecyclerView.scrollToPosition(posicaoNaLista);
1004     }
1005     barraDeProgresso.setVisibility(View.GONE);
1006     fabFiltroMenu.setVisibility(View.VISIBLE);
1007     meuRecyclerView.setVisibility(View.VISIBLE);
1008     toolbar.setSubtitle("Raio de " + distanciaMaxima + " Km");
1009     fabFiltroMenu.showMenuButton(true);
1010     botaoGps.setVisibility(true);
1011     definirLabelsParaFiltros();
1012     mostraQtdeResultados();
1013     }
1014 }
1015 }
1016 }
1017 }
```

File - E:\Android\Projetos\CadeSaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\CaixaDeDialogoGoogleAPI.java

```
1 package br.com.jucemar_dimon.cadesaude;
2
3 import android.app.AlertDialog;
4 import android.app.Dialog;
5 import android.app.DialogFragment;
6 import android.content.DialogInterface;
7 import android.os.Bundle;
8
9 /**
10 * Created by Jucemar on 08/05/2016.
11 */
12 public class CaixaDeDialogoGoogleAPI extends DialogFragment implements DialogInterface.OnClickListener {
13
14     private ErroGooglePlayService erroGooglePlayService;
15
16     public void setErroGooglePlayService(ErroGooglePlayService erroGooglePlayService) {
17         this.erroGooglePlayService = erroGooglePlayService;
18     }
19
20     public CaixaDeDialogoGoogleAPI() {
21     }
22
23     @Override
24     public Dialog onCreateDialog(Bundle savedInstanceState) {
25         AlertDialog.Builder builder = new AlertDialog.Builder(getActivity());
26         builder.setTitle("Erró Google Play Service")
27             .setPositiveButton("Sair", this)
28             .setMessage("Erró ao tentar adquirir suas coordenadas geográficas. Não é possível continuar.");
29
30     }
31
32     @Override
33     public void onClick(DialogInterface dialog, int which) {
34         erroGooglePlayService.googlePlayServiceIndisponivelO;
35     }
36 }
```

File - E:\Android\Projetos\CadeSaude\app\src\main\java\br\com\br\com\jucemar\_dimon\cadesaude\CaixaDeDialogoGoogleAPI.java

```
36
37     public interface ErroGooglePlayService {
38         public void googlePlayServiceIndisponivel();
39     }
40 }
41
```

```
File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\Estabelecimento.java

1 package br.com.jucemar_dimon.cadesaude;
2
3 import android.os.Parcel;
4 import android.os.Parcelable;
5
6 import java.util.ArrayList;
7 import java.util.Comparator;
8
9 /**
10 * Created by Jucemar on 10/03/2016.
11 */
12 public class Estabelecimento implements Parcelable, Comparator<Float>, Comparable<Estabelecimento> {
13
14     private int enes;
15     private String razaoSocial;
16     private String nomeFantasia;
17     private String logradouro;
18     private String numero;
19     private String complemento;
20     private String bairro;
21     private String cep;
22     private String municipio;
23     private String estado;
24     private String telefone;
25     private String tipoEstabelecimento;
26     private String latitude;
27     private Double longitude;
28     private Double distancia;
29     private ArrayList<Profissionais> profissionais;
30     private ArrayList<ServiçoClassificado> serviços;
31     private ArrayList<Atendimento> atendimento;
32     public static final String CNES = "cnes";
33     public static final String RAZAO_SOCIAL = "razaoSocial";
34     public static final String NOME_FANTASIA = "nomeFantasia";
35 }
```

File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\Estabelecimento.java

```
36     public static final String TELEFONE = "telefone";
37     public static final String LOGRADOURO = "Logradouro";
38     public static final String NUMERO = "numero";
39     public static final String COMPLEMENTO = "complemento";
40     public static final String BAIRRO = "bairro";
41     public static final String MUNICIPIO = "municipio";
42     public static final String CODIGO_MUNICIPIO = "codigoMunicipio";
43     public static final String ESTADO = "estado";
44     public static final String CEP = "cep";
45     public static final String LATITUDE = "Latitude";
46     public static final String LONGITUDE = "Longitude";
47     public static final String TIPO_ESTABELECIMENTO = "tipoEstabelecimento";
48
49     public Estabelecimento {
50
51         public Estabelecimento(int cnes, String razaoSocial, String nomeFantasia, String logradouro, String numero, String
52             complemento, String bairro, String cep, String municipio, String codigoMunicipio, String estado, String telefone,
53             String tipoEstabelecimento, Double latitude, Double longitude) {
54             this.cnes = cnes; this.razaoSocial
55             = razaoSocial; this.nomeFantasia =
56             nomeFantasia; this.logradouro =
57             logradouro; this.numero = numero;
58             this.complemento = complemento;
59             this.bairro = bairro;
60             this.cep = cep; this.municipio =
61             municipio; this.codigoMunicipio =
62             codigoMunicipio; this.estado =
63             estado; this.telefone =
64             telefone;
65             this.tipoEstabelecimento =
66             tipoEstabelecimento;
67             this.latitude = latitude;
68             this.longitude = longitude;
69         }
70     }
71 }
```

```
69
70
71     protected Estabelecimento(Parcel in) {
72
73         cnes = in.readInt(); razaoSocial =
74             in.readString(); nomeFantasia =
75                 in.readString(); logradouro =
76                     in.readString(); numero =
77                         in.readString(); complemento =
78                             in.readString(); bairro =
79                                 in.readString();
80         cep = in.readString(); municipio =
81             in.readString(); codigoMunicipio =
82                 in.readString(); estado =
83                     in.readString(); telefone =
84                         in.readString();
85         tipoEstabelecimento = in.readString();
86         latitude = in.readDouble();
87         longitude = in.readDouble();
88         distancia = in.readFloat();
89
90     public static final Creator<Estabelecimento> CREATOR = new Creator<Estabelecimento>() {
91
92         @Override
93         public Estabelecimento createFromParcel(Parcel in) {
94             return new Estabelecimento(in);
95         }
96
97         @Override
98         public Estabelecimento[] newArray(int size) {
99             return new Estabelecimento[size];
100
101    }
102
103    @Override
104    public int describeContents() {
```

```
104     return 0;
105 }
106
107     @Override
108     public void writeToParcel(Parcel dest, int flags) {
109         dest.writeInt(cnes);
110         dest.writeString(razaoSocial);
111         dest.writeString(nomeFantasia);
112         dest.writeString(logradouro);
113         dest.writeString(numero);
114         dest.writeString(complemento);
115         dest.writeString(bairro);
116         dest.writeString(cep);
117         dest.writeString(municipio);
118         dest.writeString(codigoMunicipio);
119         dest.writeString(estado);
120         dest.writeString(telefone);
121         dest.writeString(tipoEstabelecimento);
122         dest.writeDouble(latitude);
123         dest.writeDouble(longitude);
124         dest.writeFloat(distancia);
125     }
126
127     public int getCnes() {
128         return cnes;
129     }
130
131     public void setCnes(int cnes) {
132         this.cnes = cnes;
133     }
134
135     public String getRazaoSocial() {
136         return razaoSocial;
137     }
138 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\Estabelecimento.java

```
139     public void setRazaoSocial(String razaoSocial) {  
140         this.razaoSocial = razaoSocial;  
141     }  
142  
143     public String getNomeFantasia() {  
144         return nomeFantasia;  
145     }  
146  
147     public void setNomeFantasia(String nomeFantasia) {  
148         this.nomeFantasia = nomeFantasia;  
149     }  
150  
151     public String getLogradouro() {  
152         return logradouro;  
153     }  
154  
155     public void setLogradouro(String logradouro) {  
156         this.logradouro = logradouro;  
157     }  
158  
159     public String getNumero() {  
160         return numero;  
161     }  
162  
163     public void setNumero(String numero) {  
164         this.numero = numero;  
165     }  
166  
167     public String getComplemento() {  
168         return complemento;  
169     }  
170  
171     public void setComplemento(String complemento) {  
172         this.complemento = complemento;  
173     }  
174 }
```

File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\Estabelecimento.java

```
174  
175     public String getBairro() {  
176         return bairro;  
177     }  
178  
179     public void setBairro(String bairro) {  
180         this.bairro = bairro;  
181     }  
182  
183     public String getCep() {  
184         return cep;  
185     }  
186  
187     public void setCep(String cep) {  
188         this.cep = cep;  
189     }  
190  
191     public String getMunicipio() {  
192         return municipio;  
193     }  
194  
195     public void setMunicipio(String municipio) {  
196         this.municipio = municipio;  
197     }  
198  
199     public String getCodigoMunicipio() {  
200         return codigoMunicipio;  
201     }  
202  
203     public void setCodigoMunicipio(String codigoMunicipio) {  
204         this.codigoMunicipio = codigoMunicipio;  
205     }  
206  
207     public String getEstado() {  
208         return estado;  
209     }
```

```
File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\Estabelecimento.java

209 }
210
211     public void setEstado(String estado) {
212         this.estado = estado;
213     }
214
215     public String getTelefone() {
216         return telefone;
217     }
218
219     public void setTelefone(String telefone) {
220         this.telefone = telefone;
221     }
222
223     public String getTipoEstabelecimento() {
224         return tipoEstabelecimento;
225     }
226
227     public void setTipoEstabelecimento(String tipoEstabelecimento) {
228         this.tipoEstabelecimento = tipoEstabelecimento;
229     }
230
231     public Double getLatitude() {
232         return latitude;
233     }
234
235     public void setLatitude(Double latitude) {
236         this.latitude = latitude;
237     }
238
239     public Double getLongitude() {
240         return longitude;
241     }
242
243     public void setLongitude(Double longitude) {
```

```
244     this.longitude = longitude;
245 }
246 public float getDistancia() {
247     return distancia;
248 }
250 public void setDistancia(float distancia) {
251     this.distancia = distancia;
252 }
254 public ArrayList<Profissional> getProfissionais() {
255     return profissionais;
256 }
257
258 public void setProfissionais(ArrayList<Profissional> profissionais) {
259     this.profissionais = profissionais;
260 }
261
262 public ArrayList<ServicoClassificacao> getServicos() {
263     return servicos;
264 }
265
266 public void setServicos(ArrayList<ServicoClassificacao> servicos) {
267     this.servicos = servicos;
268 }
269
270 public ArrayList<Atendimento> getAtendimento() {
271     return atendimento;
272 }
273
274 public void setAtendimento(ArrayList<Atendimento> atendimento) {
275     this.atendimento = atendimento;
276 }
277
--
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\br\com\jucemar\_dimon\cadesaude\Estabelecimento.java

```
279     @Override
280     public String toString() {
281         return String.valueOf(distancia);
282     }
283
284     @Override
285     public int compareTo(Estabelecimento another) {
286         return compare(this.distancia, another.distancia);
287     }
288
289     @Override
290     public int compare(Float lhs, Float rhs) {
291         return Float.compare(lhs, rhs);
292     }
293 }
294 }
295 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\EstabelecimientoActivity.java

```
1 package br.com.jucemar_dimon.cadesaude;
2 /**
3 * Created by Jucemar on 10/03/2016.
4 */
5
6 import android.os.AsyncTask;
7 import android.os.Bundle;
8 import android.support.design.widget.TabLayout;
9 import android.support.v4.app.Fragment;
10 import android.support.v4.app.FragmentManager;
11 import android.support.v4.app.FragmentStatePagerAdapter;
12 import android.support.v4.view.ViewPager;
13 import android.support.v7.app.AppCompatActivity;
14 import android.support.v7.widget.Toolbar;
15 import android.view.MenuItem;
16 import android.view.View;
17 import android.widget.ProgressBar;
18 import android.widget.TextView;
19 import java.util.ArrayList;
20 import java.util.List;
21
22
23 public class EstabelecimientoActivity extends AppCompatActivity {
24
25     private Estabelecimiento estabelecimiento;
26     private Toolbar toolbar;
27     private TabLayout tabLayout;
28     private ViewPager viewPager;
29     private int cnes;
30     private ArrayList<Profissionais> listaDeProfissionais;
31     private ArrayList<Atendimento> listaDeAtendimento;
32     private ArrayList<ServiçoClassificação> listaDeServiços;
33     private ProgressBar barraDeProgresso;
34     private TextView txtTituloAppBar;
35 }
```

```
36
37     @Override
38     protected void onCreate(Bundle savedInstanceState) {
39         super.onCreate(savedInstanceState);
40         setContentView(R.layout.activity_estabelecimento);
41         viewPager = (ViewPager) findViewById(R.id.viewpager);
42         tabLayout = (TabLayout) findViewById(R.id.tabs);
43         barraDeProgresso = (ProgressBar) findViewById(R.id.barra_progresso_estabelecimento_activity);
44         estabelecimento = getIntent().getParcelableExtra("estabelecimento");
45         txvTituloAppBar = (TextView) findViewById(R.id.titulo_estab_activity_estab);
46         cnes = estabelecimento.getcnes();
47
48         txvTituloAppBar.setText(estabelecimento.getNameFantasia());
49         toolbar = (Toolbar) findViewById(R.id.toolbar_estabelecimento);
50         setSupportActionBar(toolbar);
51         getSupportActionBar().setDisplayHomeAsUpEnabled(true);
52         getSupportActionBar().setSupportActionBarEnabled(true);
53         getSupportActionBar().setSupportActionBar(carregadorDeDados);
54         getSupportActionBar().execute();
55
56     private void configVisualizadorDePaginas(ViewPager viewPager) {
57         VisualizadorDePaginasAdapter adapter = new VisualizadorDePaginasAdapter(getSupportFragmentManager());
58         EstabelecimentoFragment ef = new EstabelecimentoFragment();
59         ef.setArguments(getIntent().getExtras());
60         ProfissionaisFragment pf = new ProfissionaisFragment(); Bundle bpf = new
61         pf.putParcelableArrayList("ListaDeProfissionais",
62         ListaDeProfissionais); pf.setArguments(bpf);
63         ServicosFragment sf = new ServicosFragment();
64         Bundle bsf = new ServicosFragment();
65         bsf.putParcelableArrayList("ListaDeServicos", ListaDeServicos);
66         sf.setArguments(bsf);
67         AtendimentoFragment af = new AtendimentoFragment();
68         Bundle baf = new Bundle();
69     }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\EstabelecimentoActivity.java

```
71     baf.putParcelableArrayList("ListaDeAtendimentos", ListaDeAtendimento);
72     af.setArguments(baf);
73     adapter.addFragment(pf, "ESTABELECIMENTO");
74     adapter.addFragment(pf, "PROFISSIONAIS");
75     adapter.addFragment(sf, "SERVIÇOS");
76     adapter.addFragment(pf, "ATENDIMENTO");
77     viewPager.setAdapter(adapter);
78 }
79
80 @Override
81 public boolean onOptionsItemSelected(MenuItem item) {
82     int id = item.getItemId();
83     switch (id) {
84         case android.R.id.home:
85             onBackPressed();
86             break;
87     }
88     return super.onOptionsItemSelected(item);
89 }
90
91
92 private class VisualizadorDePaginasAdapter extends FragmentStatePagerAdapter {
93     private final List<Fragment> mFragmentList = new ArrayList<>();
94     private final List<String> mFragmentTitleList = new ArrayList<>();
95
96     public VisualizadorDePaginasAdapter(FragmentManager manager) {
97         super(manager);
98     }
99
100    @Override
101    public Fragment getItem(int position) {
102        return mFragmentList.get(position);
103    }
104}
```

File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\EstabelecimentoActivity.java

```
106     @Override  
107     public int getCount() {  
108         return mFragmentList.size();  
109     }  
110  
111     public void addFragment(Fragment fragment, String title) {  
112         mFragmentList.add(fragment);  
113         mFragmentTitleList.add(title);  
114     }  
115  
116     @Override  
117     public CharSequence getPageTitle(int position) {  
118         return mFragmentTitleList.get(position);  
119     }  
120  
121     private class CarregadorDeDados extends AsyncTask<Void, Void, Void> {  
122  
123         @Override  
124         protected void onPostExecute() {  
125             super.onPostExecute();  
126             barraDeProgresso.setVisibility(View.VISIBLE);  
127             viewPager.setVisibility(View.GONE);  
128         }  
129  
130         @Override  
131         protected void doInBackground(Void... params) {  
132             ProfissionaisDAO profissionaisDAO = ProfissionaisDAO.getInstancia(getApplicationContext());  
133             listaDeProfissionais = profissionaisDAO.listaProfissionais(String.valueOf(cnes));  
134             ServicoClassificacaoDAO servicoClassificacaoDAO = ServicoClassificacaoDAO.getInstancia(  
135             .~getApplicationContext());  
136             listaDeServicos = servicoClassificacaoDAO.listarServicosEspecializados(String.valueOf(cnes));  
137             AtendimentoDAO atendimentoDAO = AtendimentoDAO.getInstancia(getApplicationContext());  
138             listaDeAtendimento = atendimentoDAO.listarAtendimentos(String.valueOf(cnes));  
139             return null;  
140         }  
141     }  
142 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\EstabelecimentoActivity.java

```
140
141
142     @Override
143     protected void onPostExecute(Void aVoid) {
144         configVisualizadorDePaginas(viewPager);
145         tabLayout.setupWithViewPager(viewPager);
146         barraDeProgresso.setVisibility(View.GONE);
147         viewPager.setVisibility(View.VISIBLE);
148
149     }
150 }
151 }
152 }
153
154 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\EstabelecimentoAdapter.java

```
1 package br.com.jucemar_dimon.cadesaude;
2
3 import android.content.Context;
4 import android.content.SharedPreferences;
5 import android.support.design.widget.Snackbar;
6 import android.support.v7.widget.RecyclerView;
7 import android.util.Log;
8 import android.view.LayoutInflater;
9 import android.view.View;
10 import android.view.ViewGroup;
11
12 import java.text.NumberFormat;
13 import java.util.ArrayList;
14 import java.util.Locale;
15 import java.util.Set;
16
17 /**
18 * Created by Jucemar on 08/03/2016.
19 */
20 public class EstabelecimentoAdapter extends RecyclerView.Adapter<EstabelecimentoViewHolder> {
21     private Context contexto;
22     private ArrayList<Estabelecimento> listaDeEstabelecimentos;
23     private Set<String> listaDeFavoritos;
24     private String distanciaMaxima;
25     private SharedPreferences preferencias;
26     private NumberFormat numberFormat;
27
28     public EstabelecimentoAdapter(Context contexto, ArrayList<Estabelecimento> listaDeEstabelecimentos, Set<String>
29     ~ listaDeFavoritos) {
30         this.contexto = contexto;
31         this.listaDeEstabelecimentos = listaDeEstabelecimentos;
32         this.listaDeFavoritos = listaDeFavoritos;
33         this.numberFormat = NumberFormat.getInstance(new Locale("pt", "BR")); //para números
34         numberFormat.setMaximumFractionDigits(2);
```

```
35     }
36
37     // Construtor de Adapter para montar a Recycler View da BuscaGPSActivity
38     public EstabelecimentoAdapter(Context contexto, ArrayList<Estabelecimento> estabelecimentos, Set<String>
39     listaDeFavoritos, String distanciaMaxima) {
40         this.listaDeEstabelecimentos = estabelecimentos;
41         this.listaDeFavoritos = listaDeFavoritos;
42         this.distanciaMaxima = distanciaMaxima;
43         this.contexto = contexto;
44         this.numberFormat = NumberFormat.getNumberInstance(new Locale("pt", "BR")); //para números
45         numberFormat.setMaximumFractionDigits(2);
46
47     @Override
48     public EstabelecimentoViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
49         View v = LayoutInflater.from(contexto).inflate(R.layout.item_lista_estabelecimentos, parent, false);
50         EstabelecimentoViewHolder vh = new EstabelecimentoViewHolder(contexto, v, listaDeEstabelecimentos, this);
51         return vh;
52     }
53
54     @Override
55     public void onBindViewHolder(EstabelecimentoViewHolder holder, int position) {
56         Estabelecimento e = listaDeEstabelecimentos.get(position);
57         holder.nomeFantasia.setText(e.getNomeFantasia());
58         holder.bairro.setText(e.getBairro());
59         holder.municipio.setText(e.getMunicipio());
60         if (e.getDistanciaO > 0) {
61             holder.distancia.setText(String.format("%d Km", e.getDistanciaO));
62         }
63         holder.tipoEstabelecimento.setText(e.getTipoEstabelecimento());
64         if (e.getTelefoneO.length() > 1) {
65             holder.iconeTelefone.setImageResource(R.drawable.ic_phone_grey600_24dp);
66         } else {
67             holder.iconeTelefone.setAlpha(0.30f);
68     }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\EstabelecimentoAdapter.java

```
69     holder.iconTelefone.setEnabled(false);
70 }
71 Log.i("favoritos", listaDeFavoritos.toString());
72 if (listaDeFavoritos.contains(String.valueOf(e.getChesO))) {
73     holder.iconFavoritos.setImageResource(R.drawable.ic_star_grey600_24dp);
74 } else {
75     holder.iconFavoritos.setImageResource(R.drawable.ic_star_outline_grey600_24dp);
76 }
77 }

78
79 @Override
80 public int getItemCount() {
81     return listaDeEstabelecimentos.size();
82 }
83 }

84
85 public void avisarRemocaoEmFavoritos(Estabelecimento estabelecimento) {
86     if (contexto instanceof MainActivity) {
87         listaDeEstabelecimentos.remove(estabelecimento);
88         notifyDataSetChanged();
89         ((MainActivity) contexto).checkboxFavoritosO;
90     }
91     View v = null;
92     if (contexto instanceof MainActivity) {
93         v = ((MainActivity) contexto).findViewById(R.id.coordinator_layout_main_activity);
94     } else if (contexto instanceof BuscaGPSActivity) {
95         v = ((BuscaGPSActivity) contexto).findViewById(R.id.activity_busca_gps_layout);
96     } else {
97         v = ((BuscaGeralActivity) contexto).findViewById(R.id.activity_busca_geral_layout);
98     }
99     Snackbar.make(v, estabelecimento.getNomeFantasia() + " SAIU DOS FAVORITOS", Snackbar.LENGTH_SHORT).show();
100 }
101 ...
102 public void avisarAdicaoDeFavoritos(Estabelecimento estabelecimento) {
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\EstabelecimentoAdapter.java

```
104     View v = null;
105     if (contexto instanceof MainActivity) {
106         v = ((MainActivity) contexto).findViewById(R.id.coordinator_layout_main_activity);
107     } else if (contexto instanceof BuscaGPSActivity) {
108         v = ((BuscaGPSActivity) contexto).findViewById(R.id.activity_busca_gps_layout);
109     } else {
110         v = ((BuscaGeralActivity) contexto).findViewById(R.id.activity_busca_geral_layout);
111     }
112     Snackbar.make(v, estabelecimento.getNomeFantasia() + " AGORA É FAVORITO", Snackbar.LENGTH_SHORT).show();
113 }
114
115 }
116 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\EstabelecimentoAdapter.java

```
1 package br.com.jucemar_dimon.cadesaude;
2
3 import android.content.Context;
4 import android.content.SharedPreferences;
5 import android.support.design.widget.Snackbar;
6 import android.support.v7.widget.RecyclerView;
7 import android.util.Log;
8 import android.view.LayoutInflater;
9 import android.view.View;
10 import android.view.ViewGroup;
11
12 import java.text.NumberFormat;
13 import java.util.ArrayList;
14 import java.util.Locale;
15 import java.util.Set;
16
17 /**
18 * Created by Jucemar on 08/03/2016.
19 */
20 public class EstabelecimentoAdapter extends RecyclerView.Adapter<EstabelecimentoViewHolder> {
21     private Context contexto;
22     private ArrayList<Estabelecimento> listaDeEstabelecimentos;
23     private Set<String> listaDeFavoritos;
24     private String distanciaMaxima;
25     private SharedPreferences preferencias;
26     private NumberFormat numberFormat;
27
28     public EstabelecimentoAdapter(Context contexto, ArrayList<Estabelecimento> listaDeEstabelecimentos, Set<String>
29     ~ listaDeFavoritos) {
30         this.contexto = contexto;
31         this.listaDeEstabelecimentos = listaDeEstabelecimentos;
32         this.listaDeFavoritos = listaDeFavoritos;
33         this.numberFormat = NumberFormat.getInstance(new Locale("pt", "BR")); //para números
34         numberFormat.setMaximumFractionDigits(2);
```

```
35
36
37 // Construtor de Adapter para montar a Recycler View da BuscaGPSActivity
38 public EstabelecimentoAdapter(Context contexto, ArrayList<Estabelecimento> estabelecimentos, Set<String>
39 listaDeFavoritos, String distanciaMaxima) {
40     this.listaDeEstabelecimentos = estabelecimentos;
41     this.listaDeFavoritos =
42         this.distanciaMaxima =
43             this.contexto = contexto;
44     this.numberFormat = NumberFormat.getNumberInstance(new Locale("pt", "BR")); //para números
45     numberFormat.setMaximumFractionDigits(2);
46
47 @Override
48 public EstabelecimentoViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
49     View v = LayoutInflater.from(contexto).inflate(R.layout.item_lista_estabelecimentos, parent, false);
50     EstabelecimentoViewHolder vh = new EstabelecimentoViewHolder(contexto, v, listaDeEstabelecimentos, this);
51     return vh;
52 }
53
54 @Override
55 public void onBindViewHolder(EstabelecimentoViewHolder holder, int position) {
56     Estabelecimento e =
57         listaDeEstabelecimentos.get(position);
58     holder.nomeFantasia.setText(e.getNomeFantasia());
59     holder.bairro.setText(e.getBairro());
60     holder.municipio.setText(e.getMunicipio());
61     if (e.getDistanciaO > 0) {
62         holder.distancia.setText(String.format(e.getDistanciaO / 1000.0f) + " Km");
63         holder.tipoEstabelecimento.setText(e.getTipoEstabelecimento());
64         if (e.getTelefoneO.length() > 1) {
65             holder.iconeTelefone.setImageResource(R.drawable.ic_phone_grey600_24dp);
66         } else {
67             holder.iconeTelefone.setAlpha(0.30f);
68         }
69     }
70 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\EstabelecimentoAdapter.java

```
69     holder.iconTelefone.setEnabled(false);
70 }
71 Log.i("favoritos", listaDeFavoritos.toString());
72 if (listaDeFavoritos.contains(String.valueOf(e.getChesO))) {
73     holder.iconFavoritos.setImageResource(R.drawable.ic_star_grey600_24dp);
74 } else {
75     holder.iconFavoritos.setImageResource(R.drawable.ic_star_outline_grey600_24dp);
76 }
77 }

78
79 @Override
80 public int getItemCount() {
81     return listaDeEstabelecimentos.size();
82 }
83 }

84
85 public void avisarRemocaoEmFavoritos(Estabelecimento estabelecimento) {
86     if (contexto instanceof MainActivity) {
87         listaDeEstabelecimentos.remove(estabelecimento);
88         notifyDataSetChanged();
89         ((MainActivity) contexto).checkboxFavoritosO;
90     }
91     View v = null;
92     if (contexto instanceof MainActivity) {
93         v = ((MainActivity) contexto).findViewById(R.id.coordinator_layout_main_activity);
94     } else if (contexto instanceof BuscaGPSActivity) {
95         v = ((BuscaGPSActivity) contexto).findViewById(R.id.activity_busca_gps_layout);
96     } else {
97         v = ((BuscaGeralActivity) contexto).findViewById(R.id.activity_busca_geral_layout);
98     }
99     Snackbar.make(v, estabelecimento.getNomeFantasia() + " SAIU DOS FAVORITOS", Snackbar.LENGTH_SHORT).show();
100 }
101 ...
102 public void avisarAdicaoDeFavoritos(Estabelecimento estabelecimento) {
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\EstabelecimentoAdapter.java

```
104     View v = null;
105     if (contexto instanceof MainActivity) {
106         v = ((MainActivity) contexto).findViewById(R.id.coordinator_layout_main_activity);
107     } else if (contexto instanceof BuscaGPSActivity) {
108         v = ((BuscaGPSActivity) contexto).findViewById(R.id.activity_busca_gps_layout);
109     } else {
110         v = ((BuscaGeralActivity) contexto).findViewById(R.id.activity_busca_geral_layout);
111     }
112     Snackbar.make(v, estabelecimento.getNomeFantasia() + " AGORA É FAVORITO", Snackbar.LENGTH_SHORT).show();
113 }
114
115 }
116
```

```
1 package br.com.jucemar_dimon.cadesaude;
2
3 import android.content.Context;
4 import android.database.Cursor;
5 import android.database.sqlite.SQLiteDatabase;
6 import android.location.Location;
7 import android.util.Log;
8
9 import java.util.ArrayList;
10 import java.util.Collections;
11 import java.util.Set;
12
13 /**
14 * Created by Jucemar on 10/03/2016.
15 */
16 public class EstabelecimientoDAO {
17
18     private static EstabelecimientoDAO instancia;
19     private SQLiteDatabase bd; String[] colunas = {
20         TabelaEstabelecimiento.Columnas.CNES,
21         TabelaEstabelecimiento.Columnas.RAZAO_SOCIAL,
22         TabelaEstabelecimiento.Columnas.NOME_FANTASIA,
23         TabelaEstabelecimiento.Columnas.TELEFONE,
24         TabelaEstabelecimiento.Columnas.NOME_LOGRADOURO,
25         TabelaEstabelecimiento.Columnas.NUMERO,
26         TabelaEstabelecimiento.Columnas.COMPLEMENTO,
27         TabelaEstabelecimiento.Columnas.BAIRRO,
28         TabelaEstabelecimiento.Columnas.MUNICIPIO,
29         TabelaEstabelecimiento.Columnas.CODIGO_MUNICIPIO,
30         TabelaEstabelecimiento.Columnas.ESTADO,
31         TabelaEstabelecimiento.Columnas.CEP,
32         TabelaEstabelecimiento.Columnas.LATITUDE,
33         TabelaEstabelecimiento.Columnas.LONGITUDE,
34         TabelaEstabelecimiento.Columnas.TIPO_ESTABELECIMENTO };
```

```
File - E:\Android\Projetos\CadESEsaudo\app\src\main\java\br\com\jucemar_dimon\cadeseade\EstabelecimentoDAO.java

36
37
38 public static EstabelecimentoDAO getInstancia(Context context) {
39     if (instancia == null) {
40         instancia = new EstabelecimentoDAO(context);
41     }
42     return instancia;
43 }
44
45 private EstabelecimentoDAO(Context contexto) {
46     BancoDeDados bancoDeDados = BancoDeDados.getInstancia(contexto);
47     bd = bancoDeDados.getReadableDatabase();
48 }
49
50 public ArrayList<Estabelecimento> listaTodosEstabelecimentos() {
51     ArrayList<Estabelecimento> resultado = new ArrayList<Estabelecimento>();
52     Cursor c = bd.query(TabelaEstabelecimento.NOME_TABELA, colunas, TabelaEstabelecimento.Colunas.L
53 and " + TabelaEstabelecimento.Colunas.LONGITUDE + "<0 ", null, null, TabelaEstabelecimento.Coluna
54     if (c.moveToFirst()) {
55         do {
56             Estabelecimento e = extrairDoCursor(c);
57             resultado.add(e);
58         } while (c.moveToNext());
59     }
59     c.close();
60     return resultado;
61 }
62
63 public ArrayList<Estabelecimento> listaEstabelecimentos(Set<String> setMunicipios) {
64     StringBuffer filtro = new StringBuffer();
65     String[] municipios = setMunicipios.toArray(new String[setMunicipios.size()]);
66     for (int i = 0; i < municipios.length; i++) {
67         filtro.append("estabelecimento." + TabelaEstabelecimento.COlunas.CODIGO_MUNICIPIO + "=?"
68     }
69 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\EstabelecimentoDAO.java

```
69     if (i < municipios.length - 1) {
70         filtro.append("OR ");
71     }
72 }
73
74 ArrayList<Estabelecimento> resultado = new ArrayList<Estabelecimento>;
75 Cursor c = bd.query(TabelaEstabelecimento.NOME_TABELA, colunas, filtro.toString(), municipios, null, null,
76 TabelaEstabelecimento.Colunas.NOME_FANTASIA);
77 if (c.moveToFirst()) {
78     do {
79         Estabelecimento e = extrairDoCursor(c);
80         resultado.add(e);
81     } while (c.moveToNext());
82 }
83 c.close();
84 return resultado;
85 }
86 public CharSequence[] listarTiposEstabelecimentos(Set<String> setMunicipios) {
87     CharSequence[] resultado = null;
88     if (setMunicipios.size() > 0) {
89         StringBuffer filtro = new StringBuffer();
90         String[] municipios = setMunicipios.toArray(new String[setMunicipios.size()]);
91
92         for (int i = 0; i < municipios.length; i++) {
93             filtro.append(TabelaEstabelecimento.Colunas.CODIGO_MUNICIPIO + "=? ");
94             if (i < municipios.length - 1) {
95                 filtro.append("OR ");
96             }
97         }
98     }
99     String query = "SELECT DISTINCT tipo_estabelecimento FROM estabelecimento WHERE " + filtro + " ORDER BY
100 tipo_estabelecimento";
101 Cursor c = bd.rawQuery(query, municipios);
102 resultado = new CharSequence[c.getCount()];
```

File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\EstabelecimientoDAO.java

```
102     int i = 0;
103     if (c.moveToFirst()) {
104         do {
105             resultado[i] = c.getString(c.getColumnIndex(TabelaEstabelecimiento.CoLunas.TIPO_ESTABELECIMENTO));
106             i++;
107         } while (c.moveToNext());
108         c.close();
109     }
110     return resultado;
111 }
112
113
114 public ArrayList<Estabelecimiento> listaEstabelecimentosFavoritos(Set<String> listaDeFavoritos) {
115     if (listaDeFavoritos.size() > 0) {
116         ArrayList<Estabelecimiento> resultado = new ArrayList<Estabelecimiento>();
117         if (listaDeFavoritos != null) {
118             StringBuffer filtro = new StringBuffer();
119             String[] favoritos = listaDeFavoritos.toArray(new String[listaDeFavoritos.size()]);
120             for (int i = 0; i < favoritos.length; i++) {
121                 filtro.append(TabelaEstabelecimiento.CoLunas.CNES + " = " + favoritos[i] + " OR ");
122                 if (i < favoritos.length - 1) {
123                     filtro.append("OR ");
124                 }
125             }
126             Log.e("Filtro", filtro.toString());
127             Cursor c = bd.query(TabelaEstabelecimiento.NOME_TABELA, colunas, filtro.toString(), null, null,
128                     TabelaEstabelecimiento.CoLunas.NOME_FANTASIA);
129             if (c.moveToFirst()) {
130                 do {
131                     Estabelecimiento e = extrairDoCursor(c);
132                     resultado.add(e);
133                 } while (c.moveToNext());
134             }
135         }
136     }
137 }
```

```

136
137     return resultado;
138 } else {
139     return new ArrayList<Estabelecimento>();
140 }
141 }

142 public static final Estabelecimento extrairDoCursor(Cursor c) {
143     int cnes = c.getInt(c.getColumnIndex(TabelaEstabelecimento.Colunas.CNES));
144     String razaoSocial = c.getString(c.getColumnIndex(TabelaEstabelecimento.Colunas.RAZAO_SOCIAL));
145     String nomeFantasia = c.getString(c.getColumnIndex(TabelaEstabelecimento.Colunas.NOME_FANTASIA));
146     String telefone = c.getString(c.getColumnIndex(TabelaEstabelecimento.Colunas.TELEFONE));
147     String nomeLogradouro = c.getString(c.getColumnIndex(TabelaEstabelecimento.Colunas.NOME_LOGRADOURO));
148     String numero = c.getString(c.getColumnIndex(TabelaEstabelecimento.Colunas.NUMERO));
149     String complemento = c.getString(c.getColumnIndex(TabelaEstabelecimento.Colunas.COMPLEMENTO));
150     String bairro = c.getString(c.getColumnIndex(TabelaEstabelecimento.Colunas.BAIRRO));
151     String municipio = c.getString(c.getColumnIndex(TabelaEstabelecimento.Colunas.MUNICIPIO));
152     String c.getmunicipio = c.getString(c.getColumnIndex(TabelaEstabelecimento.Colunas.CODIGO_MUNICIPIO));
153     String estado = c.getString(c.getColumnIndex(TabelaEstabelecimento.Colunas.ESTADO));
154     String cep = c.getString(c.getColumnIndex(TabelaEstabelecimento.Colunas.CEP));
155     Double latitude = c.getDouble(c.getColumnIndex(TabelaEstabelecimento.Colunas.LATITUDE));
156     Double longitude = c.getDouble(c.getColumnIndex(TabelaEstabelecimento.Colunas.LONGITUDE));
157     String tipoEstabelecimento = c.getString(c.getColumnIndex(TabelaEstabelecimento.Colunas.TIPO_ESTABELECIMENTO));
158 ;
159     return new Estabelecimento(cnes, razaoSocial, nomeFantasia, numero, complemento, bairro, cep,
160     municipio, codigomunicipio, estado, telefone, tipoEstabelecimento, latitude, longitude);
161 }

162 public Estabelecimento getEstabelecimento(int cnes) {
163     Cursor c = bd.query(TabelaEstabelecimento.NOME_TABELA, colunas, TabelaEstabelecimento.Colunas.CNES + "=" +
164     cnes, null, null, TabelaEstabelecimento.Colunas.NOME_FANTASIA);
165     Estabelecimento estabelecimento = extrairDoCursor(c);
166     c.close();
167     return estabelecimento;

```

```

168
169     public ArrayList<Estabelecimento> buscaStringEmEstabelecimentos(Set<String> setMunicipios, String sentenca) {
170         StringBuffer filtroDeMunicipios = new StringBuffer();
171         String[] municipios = setMunicipios.toArray(new String[setMunicipios.size()]);
172         for (int i = 0; i < municipios.length; i++) {
173             filtroDeMunicipios.append("'" + municipios[i] + "'");
174             if (i < municipios.length - 1) {
175                 filtroDeMunicipios.append(",");
176             }
177         }
178         String query = "SELECT * FROM estabelecimento WHERE nome_fantasia LIKE '%" + sentenca + "%' OR razao_social
179         LIKE '%" + sentenca + "%' AND codigo_municipio IN (" + filtroDeMunicipios.toString() + ")";
180         ArrayList<Estabelecimento> resultado = new ArrayList<Estabelecimento>();
181         Cursor c = bd.rawQuery(query, null);
182         if (c.moveToFirst()) {
183             do {
184                 Estabelecimento e = extrairDoCursor(c);
185                 resultado.add(e);
186             } while (c.moveToNext());
187         }
188         c.close();
189         return resultado;
190     }
191     public ArrayList<Estabelecimento> buscaEstabelecimentosPorGPS(Location localizacao, String distanciaMaxima) {
192         if (distanciaMaxima.length() == 0) {
193             distanciaMaxima = "0";
194         }
195         ArrayList<Estabelecimento> resultadoTemporario = new ArrayList<Estabelecimento>;
196         ArrayList<Estabelecimento> resultadoFinal = new ArrayList<Estabelecimento>;
197         resultadoTemporario = this.listaTodosEstabelecimentos();
198         for (Estabelecimento e : resultadoTemporario) {
199             Double lat = e.getLatitude();
200             Double lon = e.getLongitude();
201

```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\EstabelecimentoDAO.java

```
202     float[] distancia = new float[1];
203     Location.distanceBetween(localizacao.getLatitude(), localizacao.getLongitude(), lat, lon, distancia);
204     if (distancia[0] <= Float.parseFloat(distanciaMaxima) * 1000 && lat != 0.0 && lon != 0.0) {
205         e.setDistancia(distancia[0]);
206         resultadoFinal.add(e);
207     }
208     Collections.sort(resultadoFinal);
209     for (Estabelecimento e : resultadoFinal) {
210         Log.e("Distâncias", e.getNomeFantasia() + " " + e.getDistancia());
211     }
212     return resultadoFinal;
213 }
214
215 public String buscaCodigoMunicipioAtual(Location localizacao) {
216     ArrayList<Estabelecimento> resultadoFinal = new ArrayList<Estabelecimento>();
217     resultadoFinal = this.listaTodosEstabelecimentos();
218     for (Estabelecimento e : resultadoFinal) {
219         Double lat = e.getLatitude();
220         Double lon = e.getLongitude();
221         float[] distancia = new float[1];
222         Location.distanceBetween(localizacao.getLatitude(), localizacao.getLongitude(), lat, lon, distancia);
223         e.setDistancia(distancia[0]);
224     }
225     Collections.sort(resultadoFinal);
226     return resultadoFinal.get(0).getCodigoMunicipio();
227 }
228 }
229 }
230 }
231 }
232 }
233 }
234 }
235 }
```



File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\EstabelecimientoDAO.java

```
1 package br.com.jucemar_dimon.cadesaude;
2
3 import android.content.Context;
4 import android.database.Cursor;
5 import android.database.sqlite.SQLiteDatabase;
6 import android.location.Location;
7 import android.util.Log;
8
9 import java.util.ArrayList;
10 import java.util.Collections;
11 import java.util.Set;
12
13 /**
14 * Created by Jucemar on 10/03/2016.
15 */
16 public class EstabelecimientoDAO {
17
18     private static EstabelecimientoDAO instancia;
19     private SQLiteDatabase bd; String[] colunas = {
20         TabelaEstabelecimiento.Columnas.CNES,
21         TabelaEstabelecimiento.Columnas.RAZAO_SOCIAL,
22         TabelaEstabelecimiento.Columnas.NOME_FANTASIA,
23         TabelaEstabelecimiento.Columnas.TELEFONE,
24         TabelaEstabelecimiento.Columnas.NOME_LOGRADOURO,
25         TabelaEstabelecimiento.Columnas.NUMERO,
26         TabelaEstabelecimiento.Columnas.COMPLEMENTO,
27         TabelaEstabelecimiento.Columnas.BAIRRO,
28         TabelaEstabelecimiento.Columnas.MUNICIPIO,
29         TabelaEstabelecimiento.Columnas.CODIGO_MUNICIPIO,
30         TabelaEstabelecimiento.Columnas.ESTADO,
31         TabelaEstabelecimiento.Columnas.CEP,
32         TabelaEstabelecimiento.Columnas.LATITUDE,
33         TabelaEstabelecimiento.Columnas.LONGITUDE,
34         TabelaEstabelecimiento.Columnas.TIPO_ESTABELECIMENTO };
```

```
File - E:\Android\Projetos\CadESEsauda\app\src\main\java\br\com\jucemar_dimon\cadesaude\EstabelecimentoDAO.java

36
37
38 public static EstabelecimentoDAO getInstancia(Context context) {
39     if (instancia == null) {
40         instancia = new EstabelecimentoDAO(context);
41     }
42     return instancia;
43 }
44
45 private EstabelecimentoDAO(Context contexto) {
46     BancoDeDados bancoDeDados = BancoDeDados.getInstancia(contexto);
47     bd = bancoDeDados.getReadableDatabase();
48 }
49
50 public ArrayList<Estabelecimento> listaTodosEstabelecimentos() {
51     ArrayList<Estabelecimento> resultado = new ArrayList<Estabelecimento>();
52     Cursor c = bd.query(TabelaEstabelecimento.NOME_TABELA, colunas, TabelaEstabelecimento.Colunas.L
53 and " + TabelaEstabelecimento.Colunas.LONGITUDE + "<0 ", null, null, TabelaEstabelecimento.Coluna
54     if (c.moveToFirst()) {
55         do {
56             Estabelecimento e = extrairDoCursor(c);
57             resultado.add(e);
58         } while (c.moveToNext());
59     }
59     c.close();
60     return resultado;
61 }
62
63 public ArrayList<Estabelecimento> listaEstabelecimentos(Set<String> setMunicipios) {
64     StringBuffer filtro = new StringBuffer();
65     String[] municipios = setMunicipios.toArray(new String[setMunicipios.size()]);
66     for (int i = 0; i < municipios.length; i++) {
67         filtro.append("estabelecimento." + TabelaEstabelecimento.Coliunas.CODIGO_MUNICIPIO + "=?"
68     }
69 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\EstabelecimentoDAO.java

```
69     if (i < municipios.length - 1) {
70         filtro.append("OR ");
71     }
72 }
73
74 ArrayList<Estabelecimento> resultado = new ArrayList<Estabelecimento>;
75 Cursor c = bd.query(TabelaEstabelecimento.NOME_TABELA, colunas, filtro.toString(), municipios, null, null,
76 TabelaEstabelecimento.Colunas.NOME_FANTASIA);
77 if (c.moveToFirst()) {
78     do {
79         Estabelecimento e = extrairDoCursor(c);
80         resultado.add(e);
81     } while (c.moveToNext());
82 }
83 c.close();
84 return resultado;
85 }
86 public CharSequence[] listarTiposEstabelecimentos(Set<String> setMunicipios) {
87     CharSequence[] resultado = null;
88     if (setMunicipios.size() > 0) {
89         StringBuffer filtro = new StringBuffer();
90         String[] municipios = setMunicipios.toArray(new String[setMunicipios.size()]);
91
92         for (int i = 0; i < municipios.length; i++) {
93             filtro.append(TabelaEstabelecimento.Colunas.CODIGO_MUNICIPIO + "=? ");
94             if (i < municipios.length - 1) {
95                 filtro.append("OR ");
96             }
97         }
98     }
99     String query = "SELECT DISTINCT tipo_estabelecimento FROM estabelecimento WHERE " + filtro + " ORDER BY
100 tipo_estabelecimento";
101 Cursor c = bd.rawQuery(query, municipios);
102 resultado = new CharSequence[c.getCount()];
```

File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\EstabelecimientoDAO.java

```
102     int i = 0;
103     if (c.moveToFirst()) {
104         do {
105             resultado[i] = c.getString(c.getColumnIndex(TabelaEstabelecimiento.CoLunas.TIPO_ESTABELECIMENTO));
106             i++;
107         } while (c.moveToNext());
108         c.close();
109     }
110     return resultado;
111 }
112
113
114 public ArrayList<Estabelecimiento> listaEstabelecimentosFavoritos(Set<String> listaDeFavoritos) {
115     if (listaDeFavoritos.size() > 0) {
116         ArrayList<Estabelecimiento> resultado = new ArrayList<Estabelecimiento>();
117         if (listaDeFavoritos != null) {
118             StringBuffer filtro = new StringBuffer();
119             String[] favoritos = listaDeFavoritos.toArray(new String[listaDeFavoritos.size()]);
120             for (int i = 0; i < favoritos.length; i++) {
121                 filtro.append(TabelaEstabelecimiento.CoLunas.CNES + " = " + favoritos[i] + " OR ");
122                 if (i < favoritos.length - 1) {
123                     filtro.append("OR ");
124                 }
125             }
126             Log.e("Filtro", filtro.toString());
127             Cursor c = bd.query(TabelaEstabelecimiento.NOME_TABELA, colunas, filtro.toString(), null, null,
128                     TabelaEstabelecimiento.CoLunas.NOME_FANTASIA);
129             if (c.moveToFirst()) {
130                 do {
131                     Estabelecimiento e = extrairDoCursor(c);
132                     resultado.add(e);
133                 } while (c.moveToNext());
134             }
135         }
136     }
137 }
```

```

136
137     return resultado;
138 } else {
139     return new ArrayList<Estabelecimento>();
140 }
141 }

142 public static final Estabelecimento extrairDoCursor(Cursor c) {
143     int cnes = c.getInt(c.getColumnIndex(TabelaEstabelecimento.Colunas.CNES));
144     String razaoSocial = c.getString(c.getColumnIndex(TabelaEstabelecimento.Colunas.RAZAO_SOCIAL));
145     String nomeFantasia = c.getString(c.getColumnIndex(TabelaEstabelecimento.Colunas.NOME_FANTASIA));
146     String telefone = c.getString(c.getColumnIndex(TabelaEstabelecimento.Colunas.TELEFONE));
147     String nomeLogradouro = c.getString(c.getColumnIndex(TabelaEstabelecimento.Colunas.NOME_LOGRADOURO));
148     String numero = c.getString(c.getColumnIndex(TabelaEstabelecimento.Colunas.NUMERO));
149     String complemento = c.getString(c.getColumnIndex(TabelaEstabelecimento.Colunas.COMPLEMENTO));
150     String bairro = c.getString(c.getColumnIndex(TabelaEstabelecimento.Colunas.BAIRRO));
151     String municipio = c.getString(c.getColumnIndex(TabelaEstabelecimento.Colunas.MUNICIPIO));
152     String c.getmunicipio = c.getString(c.getColumnIndex(TabelaEstabelecimento.Colunas.CODIGO_MUNICIPIO));
153     String estado = c.getString(c.getColumnIndex(TabelaEstabelecimento.Colunas.ESTADO));
154     String cep = c.getString(c.getColumnIndex(TabelaEstabelecimento.Colunas.CEP));
155     Double latitude = c.getDouble(c.getColumnIndex(TabelaEstabelecimento.Colunas.LATITUDE));
156     Double longitude = c.getDouble(c.getColumnIndex(TabelaEstabelecimento.Colunas.LONGITUDE));
157     String tipoEstabelecimento = c.getString(c.getColumnIndex(TabelaEstabelecimento.Colunas.TIPO_ESTABELECIMENTO));
158 ;
159     return new Estabelecimento(cnes, razaoSocial, nomeFantasia, numero, complemento, bairro, cep,
160     municipio, codigomunicipio, estado, telefone, tipoEstabelecimento, latitude, longitude);
161 }

162 public Estabelecimento getEstabelecimento(int cnes) {
163     Cursor c = bd.query(TabelaEstabelecimento.NOME_TABELA, colunas, TabelaEstabelecimento.Colunas.CNES + "=" +
164     cnes, null, null, TabelaEstabelecimento.Colunas.NOME_FANTASIA);
165     Estabelecimento estabelecimento = extrairDoCursor(c);
166     c.close();
167     return estabelecimento;

```

```

168
169     public ArrayList<Estabelecimento> buscaStringEmEstabelecimentos(Set<String> setMunicipios, String sentenca) {
170         StringBuffer filtroDeMunicipios = new StringBuffer();
171         String[] municipios = setMunicipios.toArray(new String[setMunicipios.size()]);
172         for (int i = 0; i < municipios.length; i++) {
173             filtroDeMunicipios.append("'" + municipios[i] + "'");
174             if (i < municipios.length - 1) {
175                 filtroDeMunicipios.append(", ");
176             }
177         }
178         String query = "SELECT * FROM estabelecimento WHERE nome_fantasia LIKE '%" + sentenca + "%' OR razao_social
179         LIKE '%" + sentenca + "%' AND codigo_municipio IN (" + filtroDeMunicipios.toString() + ")";
180         ArrayList<Estabelecimento> resultado = new ArrayList<Estabelecimento>();
181         Cursor c = bd.rawQuery(query, null);
182         if (c.moveToFirst()) {
183             do {
184                 Estabelecimento e = extrairDoCursor(c);
185                 resultado.add(e);
186             } while (c.moveToNext());
187         }
188         c.close();
189         return resultado;
190     }
191     public ArrayList<Estabelecimento> buscaEstabelecimentosPorGPS(Location localizacao, String distanciaMaxima) {
192         if (distanciaMaxima.length() == 0) {
193             distanciaMaxima = "0";
194         }
195         ArrayList<Estabelecimento> resultadoTemporario = new ArrayList<Estabelecimento>;
196         ArrayList<Estabelecimento> resultadoFinal = new ArrayList<Estabelecimento>;
197         resultadoTemporario = this.listaTodosEstabelecimentos();
198         for (Estabelecimento e : resultadoTemporario) {
199             Double lat = e.getLatitude();
200             Double lon = e.getLongitude();
201

```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\EstabelecimentoDAO.java

```
202     float[] distancia = new float[1];
203     Location.distanceBetween(localizacao.getLatitude(), localizacao.getLongitude(), lat, lon, distancia);
204     if (distancia[0] <= Float.parseFloat(distanciaMaxima) * 1000 && lat != 0.0 && lon != 0.0) {
205         e.setDistancia(distancia[0]);
206         resultadoFinal.add(e);
207     }
208     Collections.sort(resultadoFinal);
209     for (Estabelecimento e : resultadoFinal) {
210         Log.e("Distâncias", e.getNomeFantasia() + " " + e.getDistancia());
211     }
212     return resultadoFinal;
213 }
214
215 public String buscaCodigoMunicipioAtual(Location localizacao) {
216     ArrayList<Estabelecimento> resultadoFinal = new ArrayList<Estabelecimento>();
217     resultadoFinal = this.listaTodosEstabelecimentos();
218     for (Estabelecimento e : resultadoFinal) {
219         Double lat = e.getLatitude();
220         Double lon = e.getLongitude();
221         float[] distancia = new float[1];
222         Location.distanceBetween(localizacao.getLatitude(), localizacao.getLongitude(), lat, lon, distancia);
223         e.setDistancia(distancia[0]);
224     }
225     Collections.sort(resultadoFinal);
226     return resultadoFinal.get(0).getCodigoMunicipio();
227 }
228 }
229 }
230 }
231 }
232 }
233 }
234 }
235 }
```



File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\EstabelecimientoFragment.java

```
1 package br.com.jucemar_dimon.cadesaude;
2 /**
3 * Created by Jucemar on 18/03/2016.
4 */
5
6 import android.content.Intent;
7 import android.content.SharedPreferences;
8 import android.net.Uri;
9 import android.os.Bundle;
10 import android.support.v4.app.Fragment;
11 import android.util.Log;
12 import android.view.LayoutInflater;
13 import android.view.View;
14 import android.view.ViewGroup;
15 import android.widget.ImageButton;
16 import android.widget.TextView;
17 import com.github.clans.fab.FloatingActionButton;
18 import java.util.HashSet;
19 import java.util.Set;
20 import java.util.SortedSet;
21
22 public class EstabelecimientoFragment extends Fragment implements View.OnClickListener {
23
24     private Estabelecimiento estabelecimiento;
25     private TextView txvRazaoSocial;
26     private TextView txvNomeFantasia;
27     private TextView txvLogradouro;
28     private TextView txvNumero;
29     private TextView txvComplemento;
30     private TextView txvBairro;
31     private TextView txvCep;
32     private TextView txvMunicipio;
33     private TextView txvEstado;
34     private TextView txvTelefone;
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\EstabelecimientoFragment.java

```
36     private TextView txvTipoEstabelecimiento;
37     private ImageButton iconeEndereco; ImageButton
38     private ImageButton iconeTelefone; ImageButton
39     private config; SharedPreferences.Editor
40     private editorConfig; Set<String>
41     private listaDefavoritos; FloatingActionButton
42     private fabFavorito;
43
44     public EstabelecimientoFragment() {
45         // Required empty public constructor
46     }
47
48     @Override
49     public void onCreate(Bundle savedInstanceState) {
50         super.onCreate(savedInstanceState);
51         listaDefavoritos = new HashSet<String>();
52         Log.e("ExecutandoEmBackground", "Executando carregamento em background");
53         config = getActivity().getSharedPreferences("favoritos", MODE_PRIVATE);
54         editorConfig = config.edit();
55         listaDefavoritos = config.getStringSet("listadeFavoritos", new HashSet<String>());
56
57         configurarFABs();
58         if (listaDefavoritos.contains(String.valueOf(estabelecimiento.getCNes())))
59             fabFavorito.setImageResource(R.drawable.ic_star_white_24dp);
60         else {
61             fabFavorito.setImageResource(R.drawable.ic_star_outline_white_24dp);
62         }
63     }
64
65     @Override
66     public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
67         View v = inflater.inflate(R.layout.fragment_estabelecimiento, container, false);
68         getArguments().getParcelable("estabelecimiento");
69         fabFavorito = (FloatingActionButton) v.findViewById(R.id.fab_add_favorito);
70     }
71 }
```

```

71     fabFavorito.setOnClickListener(this);
72     iconeEndereco = (ImageButton) v.findViewById(R.id.icone_endereco);
73     iconeEndereco.setOnClickListener(this);
74     iconeTelefone = (ImageButton) v.findViewById(R.id.icone_telefone);
75     iconeTelefone.setOnClickListener(this);
76     if (estabelecimento.getTelefone().length() < 4) {
77         iconeTelefone.setEnabled(false);
78         iconeTelefone.setAlpha(0.30f);
79     } else {
80         iconeTelefone.setEnabled(true);
81         iconeTelefone.setAlpha(1.00f);
82     }
83     txvRazaoSocial = (TextView) v.findViewById(R.id.label_razao_social_abrir_estabelecimento);
84     txvNomeFantasia = (TextView) v.findViewById(R.id.label_nome_fantasia_abrir_estabelecimento);
85     txvLogradouro = (TextView) v.findViewById(R.id.label_logradouro_abrir_estabelecimento);
86     txvNumero = (TextView) v.findViewById(R.id.label_numero_abrir_estabelecimento);
87     txvComplemento = (TextView) v.findViewById(R.id.label_complemento_abrir_estabelecimento);
88     txvBairro = (TextView) v.findViewById(R.id.label_bairro_abrir_estabelecimento);
89     txvCep = (TextView) v.findViewById(R.id.label_cep_abrir_estabelecimento);
90     txvMunicipio = (TextView) v.findViewById(R.id.label_municipio_abrir_estabelecimento);
91     txvEstado = (TextView) v.findViewById(R.id.label_estado_abrir_estabelecimento);
92     txvTelefone = (TextView) v.findViewById(R.id.label_telefone_abrir_estabelecimento);
93     txvTipoEstabelecimento = (TextView) v.findViewById(R.id.label_tipo_estabelecimento_abrir_estabelecimento);
94     txvBairro.setText(estabelecimento.getBairro());
95     txvRazaoSocial.setText(estabelecimento.getRazaoSocial());
96     txvNomeFantasia.setText(estabelecimento.getNomeFantasia());
97     txvLogradouro.setText(estabelecimento.getLogradouro());
98     txvNumero.setText(estabelecimento.getNumero());
99     txvComplemento.setText(estabelecimento.getComplemento());
100    txvCep.setText(estabelecimento.getCep());
101    txvMunicipio.setText(estabelecimento.getMunicipio());
102    txvEstado.setText(estabelecimento.getEstado());
103    txvTelefone.setText(estabelecimento.getTelefone());
104    txvTipoEstabelecimento.setText(estabelecimento.getTipoEstabelecimento());

```

```
106     return v;
107 }
108
109     @Override
110     public void onClick(View v) {
111         int id = v.getId();
112         switch (id) {
113             case R.id.fab_add_favorito:
114                 atualizaFavoritosO;
115                 break;
116             case R.id.icone_endereco:
117                 abrirMapaO;
118                 break;
119             case R.id.icone_telefone:
120                 fazerLigacaoO;
121                 break;
122         }
123         Log.e("ListaDeFavoritosEst", ListaDeFavoritos.toStringO);
124     }
125
126     private void fazerLigacaoO {
127         Intent intent = new Intent(Intent.ACTION_DIAL);
128         String telefone = StringUtils.formatarTelefones(estabelecimento.getTelefoneO);
129         intent.setData(Uri.parse("tel:" + telefone));
130         getActivityO.startActivity(intent);
131     }
132
133     private void abrirMapaO {
134         Double lat = estabelecimento.getLatitudeO;
135         Double lon = estabelecimento.getLongitudeO;
136         String nomeFantasia = estabelecimento.getNomeFantasiaO;
137         String rua = estabelecimento.getLogradouroO;
138         String numero = estabelecimento.getNumeroO;
139         String cidade = estabelecimento.getMunicipioO;
140         String estado = estabelecimento.getEstadoO;
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\EstabelecimentoFragment.java

```
141 Uri uri = null;
142 if (lat != 0.0 && lon != 0.0) {
143     String uriInicio = "geo:" + lat + "," + lon;
144     String query = lat + "," + lon + "(" + nomeFantasia + ")";
145     String encodedQuery = Uri.encode(query);
146     String uriString = uriInicio + "?q=" + encodedQuery + "&z=16";
147     uri = Uri.parse(uriString);
148 } else {
149     String uriInicio = "geo:" + 0 + "," + 0;
150     String uriString = uriInicio + "?q=" + rua + "%2C" + numero + "%20" + cidade + "%2d" + estado;
151     uri = Uri.parse(uriString);
152 }
153 Log.e("endereço", uri.toString());
154 Intent intent = new Intent(android.content.Intent.ACTION_VIEW, uri);
155 startActivity(intent);
156 }
157 public void atualizaFavoritos() {
158     if (listaDefavoritos.contains(String.valueOf(estabelecimento.getCNes()))) {
159         listaDefavoritos.remove(String.valueOf(estabelecimento.getCNes()));
160         fabFavorito.setImageResource(R.drawable.ic_star_outline_white_24dp);
161     } else {
162         listaDefavoritos.add(String.valueOf(estabelecimento.getCNes()));
163         fabFavorito.setImageResource(R.drawable.ic_star_white_24dp);
164     }
165     editorConfig.putStringSet("listaDefavoritos", listaDefavoritos);
166     editorConfig.apply();
167     Log.e("Favoritos", "Resultado: " + config.getStringSet("listaDefavoritos", null).toString());
168 }
169 . . .
170 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\EstabelecimientoViewHolder.java

```
1 package br.com.jucemar_dimon.cadesaude;
2
3 import android.content.Context;
4 import android.content.Intent;
5 import android.content.SharedPreferences;
6 import android.net.Uri;
7 import android.support.v7.widget.RecyclerView;
8 import android.util.Log;
9 import android.view.View;
10 import android.widget.ImageButton;
11 import android.widget.TextView;
12
13 import java.util.ArrayList;
14 import java.util.HashSet;
15 import java.util.Set;
16
17 /**
18 * Created by Jucemar on 08/03/2016.
19 */
20 public class EstabelecimientoViewHolder extends RecyclerView.ViewHolder implements View.OnClickListener {
21
22     private Context context;
23     public TextView nomeFantasia;
24     public TextView tipoEstabelecimento;
25     public TextView bairro;
26     public TextView municipio;
27     public ArrayList<Estabelecimento> listaEstabelecimentos;
28     public ImageButton iconeFavoritos;
29     public ImageButton iconeTelefone;
30     public ImageButton iconeGPS;
31     public TextView distancia;
32     private SharedPreferences config;
33     private Set<String> editorConfig;
34     private EstabelecimentoAdapter estabelecimentosAdapter;
35 }
```

```

36   public EstabelecimentoViewHolder(Context context, View itemView, ArrayList<Estabelecimento> listaEstabelecimentos,
37   EstabelecimentoAdapter estabelecimentosAdapter) {
38     super(itemView);
39     this.estabelecimentosAdapter = estabelecimentosAdapter;
40     this.listaEstabelecimentos = listaEstabelecimentos;
41     itemView.setOnClickListener(this);
42     this.context = context;
43     this.nomeFantasia = (TextView) itemView.findViewById(R.id.nome_fantasia_lista_busca);
44     this.tipoEstabelecimento = (TextView) itemView.findViewById(R.id.tipo_estab_lista_busca);
45     this.bairro = (TextView) itemView.findViewById(R.id.bairro_lista_busca);
46     this.municipio = (TextView) itemView.findViewById(R.id.municipio_lista_busca);
47     this.distancia = (TextView) itemView.findViewById(R.id.distancia_lista_busca);
48     this.iconeFavoritos = (ImageButton) itemView.findViewById(R.id.flag_favorito);
49     this.iconeTelefone = (ImageButton) itemView.findViewById(R.id.flag_fone);
50     this.iconeGPS = (ImageButton) itemView.findViewById(R.id.flag_gps);
51     this.iconeFavoritos.setOnClickListener(this);
52     this.iconeTelefone.setOnClickListener(this);
53     this.iconeGPS.setOnClickListener(this);
54     config = context.getSharedPreferences("favoritos", context.MODE_PRIVATE);
55     editorConfig = config.edit();
56     listaDefavoritos = config.getStringSet("listadeFavoritos", new HashSet<String>());
57
58
59   @Override
60   public void onClick(View v) {
61     Log.e("Click", String.valueOf(listaEstabelecimentos.get(getAdapterPosition()).getRazaoSocial()));
62     Estabelecimento e = listaEstabelecimentos.get(getAdapterPosition());
63     Intent i = new Intent(context, EstabelecimentoActivity.class);
64     i.putExtra("estabelecimento", e);
65     i.putExtra("ListaPosicao", getAdapterPosition());
66     switch (v.getId()) {
67       case R.id.flag_favorito:
68         toggleFavorito(e);
69     }
70   }
71 }

```

```
70         break;
71     case R.id.flag_fone:
72         ligarParaEstabelecimento(e);
73         break;
74     case R.id.flag_gps:
75         abrirMapa(e);
76         break;
77     default:
78         context.startActivity(i);
79     }
80 }
81 private void toggleFavorito(Estabelecimento estabelecimento) {
82     if (listadefavoritos.contains(String.valueOf(estabelecimento.getChesO))) {
83         listadefavoritos.remove(String.valueOf(estabelecimento.getChesO));
84         this.iconeFavoritos.setImageResource(R.drawable.ic_star_outline_grey600_24dp);
85         estabelecimentosAdapter.avisaRemocaoEmFavoritos(estabelecimento);
86     } else {
87         listadefavoritos.add(String.valueOf(estabelecimento.getChesO));
88         this.iconeFavoritos.setImageResource(R.drawable.ic_star_grey600_24dp);
89         estabelecimentosAdapter.avisaAdicaoDeFavoritos(estabelecimento);
90     }
91     Log.e("favoritos", String.valueOf(listadefavoritos));
92     editorConfig.putStringSet("listadefavoritos", listadefavoritos);
93     editorConfig.apply();
94 }
95
96 private void ligarParaEstabelecimento(Estabelecimento estabelecimento) {
97     Intent intent = new Intent(Intent.ACTION_DIAL);
98     String telefone = StringUtil.formatarTelefones(estabelecimento.getTelefone());
99     intent.setData(Uri.parse("tel:" + telefone));
100    context.startActivity(intent);
101 }
102
103 private void abrirMapa(Estabelecimento estabelecimento) {
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\EstabelecimentoViewHolder.java

```
105     Double lat = estabelecimento.getLatitude();
106     Double lon = estabelecimento.getLongitude();
107     String nomeFantasia = estabelecimento.getNomeFantasia();
108     String rua = estabelecimento.getLogradouro();
109     String numero = estabelecimento.getNumero();
110     String cidade = estabelecimento.getMunicipio();
111     String estado = estabelecimento.getEstado();
112     Uri uri = null;
113     if (lat != 0.0 && lon != 0.0) {
114         String uriInicio = "geo:" + lat + "," + lon;
115         String query = lat + "," + lon + "(" + nomeFantasia + ")";
116         String encodedQuery = Uri.encode(query);
117         String uriString = uriInicio + "?q=" + encodedQuery + "&z=16";
118         uri = Uri.parse(uriString);
119     } else {
120         String uriInicio = "geo:" + 0 + "," + 0;
121         String uriString = uriInicio + "?q=" + rua + "%2C" + numero + "%20" + cidade + "%2d" + estado;
122         uri = Uri.parse(uriString);
123     }
124     Log.e("endereço", uri.toString());
125     Intent intent = new Intent(android.content.Intent.ACTION_VIEW, uri);
126     context.startActivity(intent);
127
128
129
130 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\FiltroDeEstabelecimentos.java

```
1 package br.com.jucemar_dimon.cadesaude;
2
3 import java.util.ArrayList;
4
5 /**
6 * Created by Jucemar on 04/04/2016.
7 */
8 public class FiltroDeEstabelecimentos {
9
10    public void filtrarPorTipoDeEstabelecimento(ArrayList<String> tiposEstabelecimentosSelecionados) {
11        ...
12    }
13    public void filtrarPorTipoDeAtendimento(ArrayList<String> tiposAtendimentosSelecionados) {
14        ...
15    }
16    public void filtrarPorTipoDeConvenio(ArrayList<String> tiposConveniosSelecionados) {
17        ...
18    }
19    public void filtrarPorTipoDeProfissional(ArrayList<String> tiposProfissionaisSelecionados) {
20        ...
21    }
22    public void filtrarPorTipoDeServiço(ArrayList<String> tiposServicosSelecionados) {
23        ...
24    }
25}
```

```
1 package br.com.jucemar_dimon.cadesaude;
2
3 import android.app.Activity;
4 import android.app.Dialog;
5 import android.app.DialogFragment;
6 import android.content.DialogInterface;
7 import android.os.Bundle;
8 import android.support.v7.app.AlertDialog;
9 import android.util.Log;
10 import android.widget.Button;
11
12 import java.util.ArrayList;
13
14 /**
15 * Created by Jucemar on 26/03/2016.
16 */
17 public class FiltroGeralDialog extends DialogFragment implements DialogInterface.OnClickListener {
18
19     private ArrayList<String> tiposSelecionados;
20     private ArrayList<String> bkpTiposSelecionados;
21     private InterfaceFiltroEstabelecimentos filtro;
22     private boolean[] itensChecados;
23     private CharSequence[] labelTipos;
24     private Button buttonConfirmar;
25     private Button buttonCancelar;
26     private int tipoDialog;
27     private AlertDialog dialog;
28     private int icone;
29     private ArrayList<String> tiposEstabelecimentosSelecionados;
30     private ArrayList<String> tiposAtendimentosSelecionados;
31     private ArrayList<String> tiposConveniosSelecionados;
32     private ArrayList<String> tiposProfissionaisSelecionados;
33     private ArrayList<String> tiposServicosSelecionados;
34
```

```
35     public FiltroGeralDialog() {
36
37 }
38
39     public void setArgumentos(ArrayList<String> tiposSelecionados, int tipoDeDialog, CharSequence[] listaLabelTipos) {
40         this.labelTipos = listaLabelTipos;
41         switch (tipoDeDialog) {
42             case InterfaceFiltroDeEstabelecimentos.FILTRO_TIPO_ATENDIMENTO:
43                 tiposAtendimentosSelecionados = tiposSelecionados;
44                 break;
45             case InterfaceFiltroDeEstabelecimentos.FILTRO_TIPO_CONVENIO:
46                 tiposConveniosSelecionados = tiposSelecionados;
47                 break;
48             case InterfaceFiltroDeEstabelecimentos.FILTRO_TIPO_ESTABELECIMENTO:
49                 tiposEstabelecimentosSelecionados = tiposSelecionados;
50                 break;
51             case InterfaceFiltroDeEstabelecimentos.FILTRO_TIPO_PROFISSIONAL:
52                 tiposProfissionaisSelecionados = tiposSelecionados;
53                 break;
54             case InterfaceFiltroDeEstabelecimentos.FILTRO_TIPO_SERVICO:
55                 tiposServicosSelecionados = tiposSelecionados;
56         }
57         this.tipoDeDialog = tipoDeDialog;
58         if (tiposSelecionados != null) {
59             this.tiposSelecionados = tiposSelecionados;
60         } else {
61             this.tiposSelecionados = new ArrayList<String>();
62         }
63
64     @Override
65     public Dialog onCreateDialog(Bundle savedInstanceState) {
66         itensChecados = new boolean[1labelTipos.length];
67         for (int i = 0; i < itensChecados.length; i++) {
68             itensChecados[i] = this.tiposSelecionados.contains(labelTipos[i]);
69     }
```

```
70
71     AlertDialog.Builder builder = new AlertDialog.Builder(getActivity());
72     builder.setTitle(getString(R.string.ttulo_dialog_filtro))
73         .setPositiveButton("Confirmar", this)
74         .setNegativeButton("Cancelar", this)
75         .setMultiChoiceItems(labelTipos, itensChecados, this);
76     this.dialog = builder.create();
77     return dialog;
78 }
79
80 @Override
81 public void onAttach(Activity activity) {
82     super.onAttach(activity);
83     if (!(activity instanceof InterfaceFiltroDeEstabelecimentos)) {
84         throw new RuntimeException("A activity precisa implementar a interface DialogFiltroEstabelecimento.");
85     }
86     filtro = (InterfaceFiltroDeEstabelecimentos) activity;
87 }
88
89 private void addOnCLick(int which, boolean isChecked) {
90     switch (tipodeDialog) {
91         case FILTRO_ATENDIMENTO:
92             if (isChecked) {
93                 tiposAtendimentosSelecionados.add((String) labelTipos[which]);
94             } else {
95                 tiposAtendimentosSelecionados.remove((String) labelTipos[which]);
96             }
97             if (tiposAtendimentosSelecionados.size() > 0) {
98                 buttonConfirmar.setEnabled(true);
99             } else {
100                 buttonConfirmar.setEnabled(false);
101             }
102         break;
103     case InterfaceFiltroDeEstabelecimentos.FILTRO_TIPO_CONVENIO:
```

```
104     if (isChecked) {
105         tiposConveniosSelecionados.add((String) labelTipos[which]);
106     } else {
107         tiposConveniosSelecionados.remove((String) labelTipos[which]);
108     }
109     if (tiposConveniosSelecionados.size() > 0) {
110         buttonConfirmar.setEnabled(true);
111     } else {
112         buttonConfirmar.setEnabled(false);
113     }
114     break;
115 case InterfaceFiltroDeEstabelecimentos.FILTRO_TIPO_ESTABELECIMENTO:
116     if (isChecked) {
117         tiposEstabelecimentosSelecionados.add((String) labelTipos[which]);
118     } else {
119         tiposEstabelecimentosSelecionados.remove((String) labelTipos[which]);
120     }
121     if (tiposEstabelecimentosSelecionados.size() > 0) {
122         buttonConfirmar.setEnabled(true);
123     } else {
124         buttonConfirmar.setEnabled(false);
125     }
126     break;
127 case InterfaceFiltroDeEstabelecimentos.FILTRO_TIPO_PROFISSIONAL:
128     if (isChecked) {
129         tiposProfissionaisSelecionados.add((String) labelTipos[which]);
130     } else {
131         tiposProfissionaisSelecionados.remove((String) labelTipos[which]);
132     }
133     if (tiposProfissionaisSelecionados.size() > 0) {
134         buttonConfirmar.setEnabled(true);
135     } else {
136         buttonConfirmar.setEnabled(false);
137     }
138 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\FiltroGeralDialog.java

```
139     case InterfaceFiltroDeEstabelecimentos.FILTRO_TIPO_SERVICO:
140         if (isChecked) {
141             tiposServicosSelecionados.add((String) labelTipos[which]);
142         } else {
143             tiposServicosSelecionados.remove((String) labelTipos[which]);
144         }
145         if (tiposServicosSelecionados.size() > 0) {
146             buttonConfirmar.setEnabled(true);
147         } else {
148             buttonConfirmar.setEnabled(false);
149         }
150     }
151 }
152 @Override
153 public void onClick(DialogInterface dialog, int which, boolean isChecked) {
154     addOnClick(which, isChecked);
155 }
156 }
157 @Override
158 public void onClick(DialogInterface dialog, int which) {
159     switch (which) {
160         case DialogInterface.BUTTON_POSITIVE:
161             escolherFiltro();
162             Log.e("Dialog", "Botão positivo clicado");
163             break;
164         case DialogInterface.BUTTON_NEGATIVE:
165             Log.e("Dialog", "Botão negativo clicado");
166             break;
167     }
168 }
169 }
170 private void escolherFiltro() {
171     switch (tipodeDialog) {
172         case InterfaceFiltroDeEstabelecimentos.FILTRO_TIPO_ATENDIMENTO:
173             ...
174     }
175 }
```

```
File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\FiltroGeralDialog.java

174     filtro.filtrarPorTipoDeAtendimento(tiposAtendimentosSelecionados);
175     break;
176   case InterfaceFiltroDeEstabelecimentos.FILTRO_TIPO_CONVENIO:
177     filtro.filtrarPorTipoDeConvenio(tiposConveniosSelecionados);
178     break;
179   case InterfaceFiltroDeEstabelecimentos.FILTRO_TIPO_ESTABELECIMENTO:
180     filtro.filtrarPorTipoDeEstabelecimento(tiposEstabelecimentosSelecionados);
181     break;
182   case InterfaceFiltroDeEstabelecimentos.FILTRO_TIPO_PROFISSIONAL:
183     filtro.filtrarPorTipoDeProfissional(tiposProfissionaisSelecionados);
184     break;
185   case InterfaceFiltroDeEstabelecimentos.FILTRO_TIPO_SERVICO:
186     filtro.filtrarPorTipoDeServiço(tiposServiçosSelecionados);
187     break;
188   }
189 }
190 @Override
191 public void onStart() { super.onStart();
192   dialog.setCanceledOnTouchOutside(false);
193   dialog.setCancelable(false);
194   this.buttonConfirmar = ((AlertDialog) dialog).getButton(DialogInterface.BUTTON_POSITIVE);
195   this.buttonCancelar = ((AlertDialog) dialog).getButton(DialogInterface.BUTTON_NEGATIVE);
196   if (tiposSelecionados.size() == 0) {
197     buttonConfirmar.setEnabled(false);
198   }
199 }
200 }
201 }
202 }
203 }  
~.
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\InfoPreference.java

```
1 package br.com.jucemar_dimon.cadesaude;
2 /**
3 * Created by Jucemar on 04/05/2016.
4 */
5
6 import android.content.Context;
7 import android.preference.Preference;
8 import android.util.AttributeSet;
9 import android.view.LayoutInflater;
10 import android.view.View;
11 import android.view.ViewGroup;
12 import android.widget.ImageView;
13 import android.widget.TextView;
14
15 public class InfoPreference extends Preference {
16     private TextView mensagem;
17     private ImageView icone;
18     private View viewDaPreferencia;
19
20     public TextView getMensagem() {
21         return mensagem;
22     }
23
24     public InfoPreference(Context context) {
25         super(context);
26     }
27
28     public InfoPreference(Context context, AttributeSet attrs) {
29         super(context, attrs);
30     }
31
32     public InfoPreference(Context context, AttributeSet attrs, int defStyle) {
33         super(context, attrs, defStyle);
34     }
35 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\InfoPreference.java

```
36     @Override  
37     protected View onCreateView(ViewGroup parent) {  
38         super.onCreateView(parent);  
39         LayoutInflater inflater = LayoutInflater.from(getContext());  
40         View view = inflater.inflate(R.layout.preference_info, parent, false);  
41         return view;  
42     }  
43  
44 }  
45  
46  
47
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\dimon\cadesaude\InterfaceBuscaGeralPorString.java

```
1 package br.com.jucemar.dimon.cadesaude;
2
3 /**
4 * Created by Jucemar on 15/04/2016.
5 */
6 public interface InterfaceBuscaGeralPorString {
7     public void informarTermoParaConsulta(int tipoBusca, String termo);
8 }
```

```
1 package br.com.jucemar_dimon.cadesaude;
2
3 import java.util.ArrayList;
4
5 /**
6 * Created by Jucemar on 04/04/2016.
7 */
8 public interface InterfaceFiltroDeEstabelecimentos {
9     public static final int FILTRO_TIPO_ESTABELECIMENTO = 1;
10    public static final int FILTRO_TIPO_ATENDIMENTO = 2;
11    public static final int FILTRO_TIPO_CONVENIO = 3;
12    public static final int FILTRO_TIPO_SERVICO = 4;
13    public static final int FILTRO_TIPO_PROFISSIONAL = 5;
14
15    public void filtrarPorTipoDeEstabelecimento(ArrayList<String> tiposEstabelecimentosSelecionados);
16
17    public void filtrarPorTipoDeAtendimento(ArrayList<String> tiposAtendimentosSelecionados);
18
19    public void filtrarPorTipoDeConvenio(ArrayList<String> tiposConveniosSelecionados);
20
21    public void filtrarPorTipoDeProfissional(ArrayList<String> tiposProfissionaisSelecionados);
22
23    public void filtrarPorTipoDeServiço(ArrayList<String> tiposServicosSelecionados);
24
25    public void LimparTodosFiltros();
26}
```

```
1 package br.com.jucemar_dimon.cadesaude;
2 /**
3 * Created by Jucemar on 04/04/2016.
4 */
5
6 import android.app.SearchManager;
7 import android.content.Context;
8 import android.content.Intent;
9 import android.content.SharedPreferences;
10 import android.os.AsyncTask;
11 import android.os.Bundle;
12 import android.preference.PreferenceManager;
13 import android.support.design.widget.CoordinatorLayout;
14 import android.support.design.widget.Snackbar;
15 import android.support.v4.view.MenuItemCompat;
16 import android.support.v7.app.AppCompatActivity;
17 import android.support.v7.widget.DefaultItemAnimator;
18 import android.support.v7.widget.LinearLayoutManager;
19 import android.support.v7.widget.RecyclerView;
20 import android.support.v7.widget.SearchView;
21 import android.support.v7.widget.Toolbar;
22 import android.util.Log;
23 import android.view.Menu;
24 import android.view.MenuInflater;
25 import android.view.MenuItem;
26 import android.view.View;
27 import android.widget.ImageView;
28 import android.widget.ProgressBar;
29 ...
30 import com.github.clans.fab.FloatingActionButton;
31
32 import java.util.ArrayList;
33 import java.util.HashSet;
34 import java.util.Set;
```

```
File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\MMainActivity.java

36
37 public class MainActivity extends AppCompatActivity implements View.OnClickListener {
38     private Sharedpreferences config;
39     private Sharedpreferences.Editor editorConfig;
40     private Set<String> listaDeMunicipios;
41     private String dadosDeBuscaNaTela;
42     private RecyclerView meuRecyclerView;
43     private LinearLayoutManager meuLayoutManager;
44     private LinearLayoutManager mLayoutManager;
45     private Set<String> listaDeFavoritos;
46     private FloatingActionButton fabBuscaGPS;
47     private FloatingActionButton fabBuscaGeral;
48     private EstabelecimentoDAO estabDAO;
49     private ArrayList<Estabelecimento> estabelecimentos;
50     private ArrayList<Estabelecimento> bkpEstabelecimentos;
51     private ProgressBar barraDeProgresso;
52     private ImageView background;
53     private TextView labelSemFavoritos;
54     private int posicaoNaLista;
55     private CoordinatorLayout layout;
56
57     private void executarBuscaPorPalavra(Intent intent) {
58         if (Intent.ACTION_SEARCH.equals(intent.getAction())) {
59             dadosDeBuscaNaTela = intent.getStringExtra(SearchManager.QUERY);
60         }
61
62     @Override
63     protected void onCreate(Bundle savedInstanceState) {
64         super.onCreate(savedInstanceState);
65         estabelecimentos = new ArrayList<>();
66         dadosDeBuscaNaTela = "";
67         posicaoNaLista = -1;
68         setContentView(R.layout.activity_main);
69         if (savedInstanceState != null) {
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\MainActivity.java

```
71     posicaoNaLista = savedInstanceState.getInt("posicaoNaLista");
72 }
73 config = getSharedPreferences("favoritos", MODE_PRIVATE);
74 layout = (CoordinatorLayout) findViewById(R.id.coordinator_layout_main_activity);
75 barraDeProgresso = (ProgressBar) findViewById(R.id.barra_progresso_main_activity);
76 background = (ImageView) findViewById(R.id.background_lista_vazia);
77 labelSemFavoritos = (TextView) findViewById(R.id.label_lista_sem_favoritos);
78 Log.e("BarraDeProgresso", (barraDeProgresso == null ? "não é null" : "não é null").toString());
79 Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar_actvt_main);
80 setSupportActionBar(toolbar);
81 meuRecyclerView = (RecyclerView) findViewById(R.id.lista_estabelecimentos);
82 meuRecyclerView.setHasFixedSize(true); meuLayoutManager = new
83 LinearLayoutManager(this);
84 meuRecyclerView.setLayoutManager(meuLayoutManager);
85 meuRecyclerView.setItemAnimator(new DefaultItemAnimator());
86 meuRecyclerView.scrollToPosition(posicaoNaLista);
87 meuRecyclerView.addOnScrollListener(new RecyclerView.OnScrollListener() {
88 @Override
89 public void onScrolled(RecyclerView recyclerView, int newState) {
90     super.onScrolled(recyclerView, newState);
91 }
92 @Override
93 public void onScrolled(RecyclerView recyclerView, int dx, int dy) {
94     super.onScrolled(recyclerView, dx, dy);
95     if (dy > 0) {
96         fabBuscaGPS.hide(true);
97         fabBuscaGeral.hide(true);
98     } else {
99         fabBuscaGPS.show(true);
100        fabBuscaGeral.show(true);
101    }
102 }
103 });
104 fabBuscaGPS = (FloatingActionButton) findViewById(R.id.fab_busca_gps);
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\MainActivity.java

```
106     fabBuscaGPS.setOnClickListerner(this);
107     fabBuscaGeral = (FloatingActionButton) findViewById(R.id.fab_busca_parametros);
108     fabBuscaGeral.setOnClickListerner(this);
109     executarBuscaPorPalavra(getIntent());
110 }
111
112 @Override
113 protected void onSaveInstanceState(Bundle outState) {
114     super.onSaveInstanceState(outState);
115     outState.putInt("posicaoNaLista", posicaoNaLista);
116 }
117
118 public void checarFavoritos() {
119     if (estabelecimentos.size() == 0)
120         labelSemFavoritos.setVisibility(View.VISIBLE);
121     background.setVisibility(View.VISIBLE);
122 }
123
124 private void carregaDados() {
125     CarregadorDeDados carregadorDeDados = new CarregadorDeDados(this);
126     carregadorDeDados.execute();
127 }
128
129 @Override
130 protected void onResume() {
131     super.onResume();
132     carregaDados();
133 }
134
135
136 @Override
137 protected void onNewIntent(Intent intent) {
138     super.onNewIntent(intent);
139     executarBuscaPorPalavra(intent);
140 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\MainActivity.java

```
141 }
142
143     @Override
144     public boolean onCreateOptionsMenu(Menu menu) {
145         MenuInflater inflater = getMenuInflater();
146         inflater.inflate(R.menu.main, menu);
147         MenuItem searchItem = menu.findItem(R.id.action_search);
148         SearchManager searchManager = (SearchManager) MainActivity.this
149             .getSystemService(Context.SEARCH_SERVICE);
150         SearchView searchView = null;
151         if (searchItem != null) {
152             searchView = (SearchView) searchItem.getActionView();
153         }
154         if (searchView != null) {
155             searchView.setSearchableInfo(searchManager.getSearchableInfo(MainActivity.this
156                 .getComponentName()));
157         }
158         MenuItemCompat.OnActionExpandListener expandListener = new MenuItemCompat
159             .OnActionExpandListener() {
160             @Override
161             public boolean onMenuItemActionCollapse(MenuItem item) {
162                 carregaDados();
163                 return true;
164             }
165             @Override
166             public boolean onMenuItemActionExpand(MenuItem item) {
167                 // Fazer algo quando o menu é expandido
168                 return true;
169             }
170         };
171         MenuItemCompat.setOnActionExpandListener(searchItem, expandListener);
172         return super.onCreateOptionsMenu(menu);
173     }
174 }
```

```
File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\MMainActivity.java

176     @Override
177     public boolean onOptionsItemSelected(MenuItem item) {
178         int id = item.getItemId();
179         switch (id) {
180             case R.id.action_configuracoes:
181                 Intent c = new Intent(this, PreferenciasActivity.class);
182                 startActivity(c);
183                 return true;
184             case R.id.action_ajuda:
185                 Intent a = new Intent(this, AjudaActivity.class);
186                 startActivity(a);
187                 return true;
188             case R.id.action_sobre:
189                 Intent s = new Intent(this, SobreActivity.class);
190                 startActivity(s);
191                 return true;
192             }
193             return super.onOptionsItemSelected(item);
194         }
195     @Override
196     public void onClick(View v) {
197         int id = v.getId();
198         switch (id) {
199             case R.id.fab_busca_gps:
200                 Intent bgps = new Intent(this, BuscaGPSActivity.class);
201                 bgps.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TASK);
202                 startActivity(bgps);
203                 break;
204             case R.id.fab_busca_parametros:
205                 Intent bg = new Intent(this, BuscaGeralActivity.class);
206                 startActivity(bg);
207                 break;
208         }
209     }
```

```
211
212     @Override
213     protected void onPause() {
214         super.onPause();
215         if (meuRecyclerView != null) {
216             LinearLayoutManager lm = (LinearLayoutManager) meuRecyclerView.getLayoutManager();
217             posicaoNaLista = lm.findFirstVisibleItemPosition();
218         }
219     }
220
221     private class CarregadorDeDados extends AsyncTask<Void, Void, Void> {
222         private Context contexto;
223
224         public CarregadorDeDados(Context contexto) {
225             this.contexto = contexto;
226         }
227
228         @Override
229         protected void doInBackground(Void... params) {
230             listaDeFavoritos = config.getStringSet("ListaDeFavoritos", new HashSet<String>());
231             listaDeMunicipios = PreferenceManager.getDefaultSharedPreferences(contexto)
232                 .getStringSet(getString(R.string.pref_municipios_preferidos_key), new HashSet<String>());
233             estabDAO = EstabelecimentoDAO.getInstance(getApplicationContext());
234             estabeleimentos = estabDAO.listaEstabelecimentosFavoritos(listaDeFavoritos);
235             if (dadosDeBuscaNaTela.trim().length() > 0) {
236                 dadosDeBuscaNaTela =StringUtil.removeAcentos(dadosDeBuscaNaTela);
237                 bkpEstabelecimentos = dadosDeBuscaNaTela.trim();
238                 bkpEstabelecimentos = new ArrayList<O>;
239                 ArrayList<Estabelecimento> estabelecimentosTemp = new ArrayList<Estabelecimento>;
240                 for (Estabelecimento e : estabelecimentos) {
241                     if (e.getNameFantasia().toLowerCase().contains(dadosDeBuscaNaTela.toLowerCase()) || e.
242                         getRazaoSocial().toLowerCase()
243                         .contains(dadosDeBuscaNaTela)) {
244                         estabelecimentosTemp.add(e);
245                 }
246             }
247         }
248         @Override
249         protected void onPostExecute(Void result) {
250             if (result != null) {
251                 resultado.setText(result.toString());
252             }
253         }
254     }
255 }
```

```
245
246
247     }
248
249     estabelecimentos = estabelecimentosTemp;
250
251     }
252
253     @Override
254     protected void onPostExecute(Void aVoid) {
255         super.onPostExecute(aVoid);
256         mAdapter = new EstabelecimentoAdapter(contexto, estabelecimentos, ListaDeFavoritos);
257         meuRecyclerView.setAdapter(mAdapter);
258         if (estabelecimentos.size() > 0) {
259             meuRecyclerView.setVisibility(View.VISIBLE);
260             barraDeProgresso.setVisibility(View.GONE);
261             background.setVisibility(View.GONE);
262             labelSemFavoritos.setVisibility(View.GONE);
263         } else {
264             meuRecyclerView.setVisibility(View.GONE);
265             barraDeProgresso.setVisibility(View.GONE);
266             background.setVisibility(View.VISIBLE);
267             labelSemFavoritos.setVisibility(View.VISIBLE);
268         }
269         if (posicaoNaLista > -1) {
270             meuRecyclerView.scrollToPosition(posicaoNaLista);
271         }
272         Log.e("Inicio", "Lista de municipio " + listaDeMunicipios.size());
273         if (listaDeMunicipios.size() == 0) {
274             Intent i = new Intent(contexto, PrimeirosPassosActivity.class);
275             startActivity(i);
276         }
277         if (dadosDeBuscaNaTela.length() > 0) {
278             Snackbar.make(layout, "Foram localizados " + mAdapter.getItemCount() + " estabelecimentos", Snackbar.LENGTH_SHORT).show();
279         }
280     }
281 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\MainActivity.java

```
279         dadosDeBuscaNaTela = "";
280     }
281
282     @Override
283     protected void onPostExecute()
284     {
285         barraDeProgresso.setVisibility(View.VISIBLE);
286         meuRecyclerView.setVisibility(View.GONE);
287     }
288
289 }
290
291
292
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\PreferenciasActivity.java

```
1 package br.com.jucemar_dimon.cadesaude;
2
3 import android.os.Bundle;
4 import android.support.v7.app.AppCompatActivity;
5 import android.view.MenuItem;
6
7 /**
8 * Created by Jucemar on 04/04/2016.
9 */
10 public class PreferenciasActivity extends AppCompatActivity {
11
12     private PreferenciasFragment preferenciasFragment;
13
14     @Override
15     protected void onCreate(Bundle savedInstanceState) {
16         super.onCreate(savedInstanceState);
17         preferenciasFragment = new PreferenciasFragment();
18         getSupportFragmentManager().setDisplayHomeAsUpEnabled(true);
19         if (savedInstanceState == null) {
20             getSupportFragmentManager().beginTransaction()
21                 .replace(android.R.id.content, preferenciasFragment, "preferenciasFragment")
22                 .commit();
23         }
24     }
25
26     @Override
27     public boolean onOptionsItemSelected(MenuItem item) {
28         int id = item.getItemId();
29         switch (id) {
30             case android.R.id.home:
31                 onBackPressed();
32         }
33         return super.onOptionsItemSelected(item);
34     }
35 }
```

```
36 }  
37  
38  
39  
40  
41  
42  
43
```

File - E:\Android\Projetos\CadeSaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\PreferenceasFragment.java

```
1 package br.com.jucemar_dimon.cadesaude;
2
3 import android.content.SharedPreferences;
4 import android.os.Bundle;
5 import android.preference.MultiSelectListPreference;
6 import android.preference.PreferenceFragment;
7 import android.preference.PreferenceManager;
8 import android.support.design.widget.Snackbar;
9 import android.util.Log;
10 import android.view.View;
11
12 import java.util.ArrayList;
13 import java.util.Arrays;
14 import java.util.HashSet;
15 import java.util.Set;
16
17 /**
18 * Created by Jucemar on 14/02/2016.
19 */
20 public class PreferenceasFragment extends PreferenceFragment implements SharedPreferences.OnSharedPreferenceChangeListener {
21
22     private MultiSelectListPreference listaDePreferencias;
23     private Set<String> selecoesNaLista;
24     private SharedPreferences preferencias;
25     private ArrayList<String> codMunicipio;
26     private ArrayList<String> nomeMunicipio;
27     private String municipioPadrao;
28     private SharedPreferences.Editor editorConfig;
29     private SeekBarPreference mSeekBar2;
30     private InfoPreference info;
31
32     @Override
33     public void onCreate(Bundle savedInstanceState) {
34         super.onCreate(savedInstanceState);
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\PreferenciasFragment.java

```
35     addPreferencesFromResource(R.xml.preferencias);
36     ListaDePreferencias = (MultiSelectPreference) findPreference(getString(R.string.
37     pref_municipios_preferidos_key));
38     preferencias = PreferenceManager.getDefaultSharedPreferences(getActivity());
39     info = (InfoPreference) findPreference(getString(R.string.pref_info_key));
40     getPreferenceScreenO.removePreference(info);
41     if (getArgumentsO != null) {
42         municipioPadrao = getArgumentsO.getString("municipioAtual");
43         Log.e("municipioAtual", municipioPadrao);
44         selecoesNaLista.add(municipioPadrao);
45         ListadePreferencias.setValues(selecoesNaLista);
46         editorConfig = listaDePreferencias.getEditor();
47         editorConfig.putStringSet("municipios_preferidos", selecoesNaLista);
48         editorConfig.apply();
49         getPreferenceScreenO.addPreference(info);
50     }
51     mSeekBar2 = (SeekBarPreference) findPreference(getString(R.string.pref_distancia_estabelecimentos_key));
52     String[] arrayNomes = getResourcesO.getStringArray(R.array.nomes_municipios_view);
53     nomeMunicipio = new ArrayList<String>(Arrays.asList(arrayNomes));
54     String[] arrayCodigos = getResourcesO.getStringArray(R.array.codigos_municipios);
55     codMunicipio = new ArrayList<String>(Arrays.asList(arrayCodigos));
56     mostrarMunicipiosEscolhidos(listaDePreferencias.getValuesO());
57 }
58 @Override
59 public void onViewCreated(View view, Bundle savedInstanceState) {
60     super.onViewCreated(view, savedInstanceState);
61     if (getArgumentsO != null) {
62         Snackbar.make(view, "Você está em " + listaDePreferencias.getSummaryO(), Snackbar.LENGTH_SHORT).showO();
63     }
64 }
65 @Override
66 public void onResumeO {
67 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\PreferenciasFragment.java

```
69     super.onResume();
70     SharedPreferences      prefs      =      getPreferenceManager().getSharedPreferences();
71     onSharedPreferenceChanged(prefs,  getString(R.string.pref_di_stancia_estabelecimentos_key));
72     prefs.registerOnSharedPreferenceChangeListener(this);
73 }
74
75     @Override
76     public void onPause() {
77         super.onPause();
78
79         SharedPreferences prefs = getPreferenceManager().getSharedPreferences();
80         prefs.unregisterOnSharedPreferenceChangeListener(this);
81     }
82
83     @Override
84     public void onSharedPreferenceChanged(SharedPreferences prefs, String key) {
85
86         if (key.equals(getString(R.string.pref_municipios_preferidos_key))) {
87             StringBuffer sb = new StringBuffer();
88             Set<String> temp = prefs.getStringSet(key, new HashSet<String>());
89             if (temp.size() > 0) {
90                 getActivity(). setResult(1);
91             }
92             ArrayList<String> tempList = new ArrayList(temp);
93             for (int i = 0; i < tempList.size(); i++) {
94                 int j = codMunicipio.indexOf(tempList.get(i));
95                 sb.append(nomeMunicipio.get(j));
96                 if (i < tempList.size() - 1) {
97                     sb.append(",");
98                 }
99             }
100            sb.append(" . ");
101
102            listaDePreferencias.setSummary(sb.toString());
103        }
104    }
```

```
104     Log.e("teste", "Chamou o onSharedPreference");
105 }
106 if (key.equals(getString(R.string.pref_distancia_estabelecimentos_key))) {
107     int i = prefs.getInt(key, 1);
108     mSeek2.setSummary(i + " Km");
109 }
110 }

111 private void mostrarMunicipiosEscolhidos(Set<String> setMunicipios) {
112     StringBuffer sb = new StringBuffer(); Set<String>
113     temp = setMunicipios; ArrayList<String> tempList =
114     new ArrayList(temp);
115     for (int i = 0; i < tempList.size(); i++) {
116         int j = codMunicipio.indexOf(tempList.get(i));
117         sb.append(nomeMunicipio.get(j));
118         if (i < tempList.size() - 1) {
119             sb.append(",");
120         }
121     }
122     sb.append(" - ");
123     if (setMunicipios.size() > 0) {
124         listaDePreferencias.setSummary(sb.toString());
125     } else {
126         listaDePreferencias.setSummary("Não há municípios selecionados.");
127     }
128 }
129 }
130 }
131 }
132 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\PrimeiraConfiguracaoActivity.java

```
1 package br.com.jucemar_dimon.cadesaude;
2 /**
3 * Created by Jucemar on 14/02/2016.
4 */
5
6 import android.Manifest;
7 import android.app.Activity;
8 import android.app.Dialog;
9 import android.content.Context;
10 import android.content.Intent;
11 import android.content.IntentSender;
12 import android.content.SharedPreferences;
13 import android.content.pm.PackageManager;
14 import android.graphics.Color;
15 import android.location.Location;
16 import android.os.AsyncTask;
17 import android.os.Bundle;
18 import android.support.design.widget.Snackbar;
19 import android.support.v4.app.ActivityCompat;
20 import android.support.v7.app.AppCompatActivity;
21 import android.util.Log;
22 import android.view.Menu;
23 import android.view.MenuInflater;
24 import android.view.MenuItem;
25 import android.view.View;
26 import android.widget.FrameLayout;
27 import android.widget.ImageView;
28 import android.widget.ProgressBar;
29 ...
30 import com.google.android.gms.common.ConnectionResult;
31 import com.google.android.gms.common.GoogleApiAvailability;
32 import com.google.android.gms.common.api.GoogleApiClient;
33 import com.google.android.gms.common.api.GoogleApiClient.ConnectionCallbacks;
34 import com.google.android.gms.common.api.PendingResult;
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\PrimeiraConfiguracaoActivity.java

```
36 import com.google.android.gms.common.api.ResultCallback;
37 import com.google.android.gms.common.api.Status;
38 import com.google.android.location.LocationListener;
39 import com.google.android.location.LocationRequest;
40 import com.google.android.gms.location.LocationServices;
41 import com.google.android.gms.location.LocationSettingsRequest;
42 import com.google.android.gms.location.LocationSettingsResult;
43 import com.google.android.gms.location.LocationSettingsStates;
44 import com.google.android.gms.location.LocationSettingsStatusCodes;
45
46 public class PrimeiraConfiguracaoActivity extends AppCompatActivity implements
47     CaixaDeDialogoGooglePlayService,
48     ConnectionCallbacks,
49     GoogleApiClient.OnConnectionFailedListener,
50     LocationListener, ResultCallback<LocationSettingsResult> {
51
52     public static final int REQUEST_CODE_GOOGLE_PLAY_SERVICES = 1982;
53     private static final int REQUEST_CODE_LOCATION = 2;
54     private LocationRequest request;
55     private GoogleApiClient apiClient;
56     private SharedPreferences config;
57     private Status status;
58     private Location localizacao; CarregadorDeDados
59     private carregadorDeDados; PreferenciasFragment
60     private preferenciasFragment; ProgressBar
61     private barraDeProgresso;
62     private TextView txtErroGps;
63     private ImageView imgErroGps;
64     private FrameLayout layoutActivity;
65     private MenuItem botaoConfirmar;
66
67     @Override
68     protected void onCreate(Bundle savedInstanceState) {
69         super.onCreate(savedInstanceState);
70     }
71 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\PrimeiraConfiguracaoActivity.java

```
71     setContentView(R.layout.activity_primeira_configuracao);
72     preferenciasFragment = new PreferenciasFragment();
73     txtErrorGps = (TextView) findViewById(R.id.label_error_primeira_config_activity);
74     imgErrorGps = (ImageView) findViewById(R.id.background_error_primeira_conf_activity);
75     barraDeProgresso = (ProgressBar) findViewById(R.id.barra_progresso_primeira_conf_activity);
76     layoutActivity = (FrameLayout) findViewById(R.id.layout_primeira_config_activity);
77     //metodo chamado para realizar verificações para saber se o dispositivo possui o google play services
78     instalado
79     verificarGooglePlayService();
80     //Instancia o objeto da api do google e apartir de dai realiza checagens quanto ao google play services
81     apiClient = new GoogleApiClient.Builder(this)
82         .addConnectionCallbacks(this)
83         .addOnConnectionFailedListener(this)
84         .addApi(LocationServices.API)
85         .build();
86
87     private void carregarDados() {
88         carregadorDeDados = new CarregadorDeDados(this);
89         carregadorDeDados.execute();
90     }
91
92     @Override
93     protected void onSaveInstanceState(Bundle outState) {
94         super.onSaveInstanceState(outState);
95     }
96
97     private void verificarGooglePlayService() {
98         GoogleApiClient api = GoogleApiClientAvailability.getINSTANCE();
99         int code = api.isGooglePlayServicesAvailable(getApplicationContext());
100        if (code == ConnectionResult.SUCCESS) {
101            //da sequencia nos procedimento normalmente
102        } else {
103            if (code == ConnectionResult.SERVICE_MISSING ||
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\PrimeiraConfiguracaoActivity.java

```
105     code == ConnectionResult.T_SERVICE_DISABLED) {
106         Dialog dialog = api.getErrorDialog(this, code, REQUEST_CODE_GOOGLE_PLAY_SERVICES);
107         dialog.show();
108     } else {
109         CaixaDeDialogoGoogleAPI cxDialog = new CaixaDeDialogoGoogleAPI();
110         cxDialog.setErroGooglePlayService(this);
111         cxDialog.show(getFragmentManager(), "DialogError");
112     }
113 }
114 }
115
116     protected void onStart() {
117         super.onStart();
118         apiClient.connect();
119     }
120
121     @Override
122     public void onBackPressed() {
123         super.onBackPressed();
124     }
125
126     @Override
127     protected void onStop() {
128         super.onStop();
129         apiClient.disconnect();
130     }
131
132     @Override
133     protected void onPause() {
134         super.onPause();
135     }
136
137     @Override
138     public void onConnected(Bundle bundle) {
...}
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\PrimeiraConfiguracaoActivity.java

```
140     request = LocationRequest.createO;
141     request.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY);
142     request.setInterval(500);
143     LocationSettingsRequest.Builder builder = new LocationSettingsRequest.Builder().addLocationRequest(request);
144     builder.setAlwaysShow(true);
145     PendingResult<LocationSettingsResult> result = LocationServices.SettingsApi.checkLocationSettings(apiClient,
146         builder.buildO);
147     result.setResultCallback(this);
148
149     @Override
150     public void onRequestPermissionsResult(int requestCode, String[] permissions, int[] grantResults) {
151         if (requestCode == REQUEST_CODE_LOCATION) {
152             if (grantResults.length == 1 && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
153                 adquirirLocalizacaoO();
154             } else {
155                 Snackbar.make(getApplicationContext(), R.string.tentar_novamente_label, Snackbar.LENGTH_INDEFINITE)
156                     .setAction(R.string.tentar_novamente_action, new View.OnClickListener() {
157                         @Override
158                         public void onClick(View v) {
159                             adquirirLocalizacaoO();
160                         }
161                     })
162                     .setActionTextColor(Color.YELLOW)
163                     .showO;
164             imgErroGps.setVisibility(View.VISIBLE);
165             txtErroGps.setVisibility(View.VISIBLE);
166             barraDeProgresso.setVisibility(View.GONE);
167         }
168     }
169
170     private void adquirirLocalizacaoO {
171         imgErroGps.setVisibility(View.GONE);
172         txtErroGps.setVisibility(View.GONE);
173     }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\PrimeiraConfiguracaoActivity.java

```
174     barraDeProgresso.setVisibility(View.VISIBLE);
175     if (ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
176         if (ActivityCompat.shouldShowRequestPermissionRationale(this, Manifest.permission.ACCESS_FINE_LOCATION)) {
177             ActivityCompat.requestPermissions(this, new String[]{Manifest.permission.ACCESS_FINE_LOCATION},
178                 REQUEST_CODE_LOCATION);
179         }
180     } else {
181         LocationServices.FusedLocationApi.requestLocationUpdates(apiClient, request, this);
182     }
183 }
184
185 @Override
186 public void onConnectionSuspended(int i) {
187     //Não implementado
188 }
189
190 @Override
191 public void onConnectionFailed(URLConnectionResult connectionResult) {
192     //Não implementado
193 }
194
195 @Override
196 public boolean onCreateOptionsMenu(Menu menu) {
197     getMenuInflater().inflate(R.menu.menu_primeira_configuracao, menu);
198     menuInflator.inflate(R.menu.menu_primeira_configuracao, menu);
199     botaoConfirmar = menu.findItem(R.id.action_salvar_config);
200     botaoConfirmar.setVisible(false);
201     return super.onCreateOptionsMenu(menu);
202 }
203
204 @Override
205 public boolean onOptionsItemSelected(MenuItem item) {
206     int id = item.getItemId();
207 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\PrimeiraConfiguracaoActivity.java

```
207     switch (id) {
208         case R.id.action_salvar_config:
209             Intent i = new Intent(PrimeiraConfiguracaoActivity.this, MainActivity.class);
210             i.addFlags(Intent.FLAG_ACTIVITY_REORDER_TO_FRONT);
211             startActivity(i);
212             finish();
213         }
214         return super.onOptionsItemSelected(item);
215     }
216     @Override
217     public void onLocationChanged(Location location) { this.localizacao = location;
218         LocationServices.FusedLocationApi.removeLocationUpdates(apiClient, this);
219         carregarDados();
220     }
222
223     @Override
224     protected void onActivityResult(int requestCode, int resultCode, Intent data) {
225         final LocationSettingsStates states;
226         if (data != null) {
227             states = LocationSettingsStates.fromIntent(data);
228         }
229         switch (requestCode) {
230             case REQUEST_CODE_GOOGLE_PLAY_SERVICES:
231                 switch (resultCode) {
232                     case Activity.RESULT_OK:
233                         adquirirLocalizacao();
234                         break;
235                     case Activity.RESULT_CANCELED:
236                         Snackbar.make(layoutActivity, R.string.tentar_novamente_label, Snackbar.LENGTH_INDEFINITE)
237                             .setAction(R.string.tentar_novamente_action, new View.OnClickListener() {
238
239                             @Override
240                             public void onClick(View v) {
241                                 try {
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\PrimeiraConfiguracaoActivity.java

```
242 REQUEST_CODE_GOOGLE_PLAY_SERVICES);      } catch (IntentSender.SendIntentException e) {  
243     e.printStackTrace();  
244 }  
245 }  
246 }  
247 }  
248     .setActionTextColor(Color.YELLOW)  
249     .showO;  
250     imgErroGps.setVisibility(View.VISIBLE);  
251     txtErroGps.setVisibility(View.VISIBLE);  
252     barraDeProgresso.setVisibility(View.GONE);  
253 }  
254     break;  
255 }  
256 }  
257 @Override  
258 public void onResult(LocationSettingsResult locationSettingsResult) {  
259     status = LocationSettingsResult.getStatusO;  
260     switch (status.getStatusCodeO) {  
261         case LocationSettingsStatusCodes.SUCCESS:  
262             adquirirLocalizacaoO;  
263             break;  
264         case LocationSettingsStatusCodes.RESOLUTION_REQUIRED:  
265             try {  
266                 status.startResolutionForResult(PrimeiraConfiguracaoActivity.this,  
267                 REQUEST_CODE_GOOGLE_PLAY_SERVICES);  
268             } catch (IntentSender.SendIntentException e) {  
269                 Snackbar.makeLayoutActivity("Algum motivo desconhecido impediu a aquisição de sua localização",  
270                 Snackbar.LENGTH_LONG).setActionTextColor(Color.RED).showO;  
271                 imgErroGps.setVisibility(View.VISIBLE);  
272                 txtErroGps.setVisibility(View.VISIBLE);  
273                 barraDeProgresso.setVisibility(View.GONE);  
274             }  
275         }  
276     }  
277 }  
278 }  
279 }  
280 }  
281 }  
282 }  
283 }  
284 }  
285 }  
286 }  
287 }  
288 }  
289 }  
290 }  
291 }  
292 }  
293 }  
294 }  
295 }  
296 }  
297 }  
298 }  
299 }  
300 }  
301 }  
302 }  
303 }  
304 }  
305 }  
306 }  
307 }  
308 }  
309 }  
310 }  
311 }  
312 }  
313 }  
314 }  
315 }  
316 }  
317 }  
318 }  
319 }  
320 }  
321 }  
322 }  
323 }  
324 }  
325 }  
326 }  
327 }  
328 }  
329 }  
330 }  
331 }  
332 }  
333 }  
334 }  
335 }  
336 }  
337 }  
338 }  
339 }  
340 }  
341 }  
342 }  
343 }  
344 }  
345 }  
346 }  
347 }  
348 }  
349 }  
350 }  
351 }  
352 }  
353 }  
354 }  
355 }  
356 }  
357 }  
358 }  
359 }  
360 }  
361 }  
362 }  
363 }  
364 }  
365 }  
366 }  
367 }  
368 }  
369 }  
370 }  
371 }  
372 }  
373 }  
374 }  
375 }  
376 }  
377 }  
378 }  
379 }  
380 }  
381 }  
382 }  
383 }  
384 }  
385 }  
386 }  
387 }  
388 }  
389 }  
390 }  
391 }  
392 }  
393 }  
394 }  
395 }  
396 }  
397 }  
398 }  
399 }  
400 }  
401 }  
402 }  
403 }  
404 }  
405 }  
406 }  
407 }  
408 }  
409 }  
410 }  
411 }  
412 }  
413 }  
414 }  
415 }  
416 }  
417 }  
418 }  
419 }  
420 }  
421 }  
422 }  
423 }  
424 }  
425 }  
426 }  
427 }  
428 }  
429 }  
430 }  
431 }  
432 }  
433 }  
434 }  
435 }  
436 }  
437 }  
438 }  
439 }  
440 }  
441 }  
442 }  
443 }  
444 }  
445 }  
446 }  
447 }  
448 }  
449 }  
450 }  
451 }  
452 }  
453 }  
454 }  
455 }  
456 }  
457 }  
458 }  
459 }  
460 }  
461 }  
462 }  
463 }  
464 }  
465 }  
466 }  
467 }  
468 }  
469 }  
470 }  
471 }  
472 }  
473 }  
474 }  
475 }  
476 }  
477 }  
478 }  
479 }  
480 }  
481 }  
482 }  
483 }  
484 }  
485 }  
486 }  
487 }  
488 }  
489 }  
490 }  
491 }  
492 }  
493 }  
494 }  
495 }  
496 }  
497 }  
498 }  
499 }  
500 }  
501 }  
502 }  
503 }  
504 }  
505 }  
506 }  
507 }  
508 }  
509 }  
510 }  
511 }  
512 }  
513 }  
514 }  
515 }  
516 }  
517 }  
518 }  
519 }  
520 }  
521 }  
522 }  
523 }  
524 }  
525 }  
526 }  
527 }  
528 }  
529 }  
530 }  
531 }  
532 }  
533 }  
534 }  
535 }  
536 }  
537 }  
538 }  
539 }  
540 }  
541 }  
542 }  
543 }  
544 }  
545 }  
546 }  
547 }  
548 }  
549 }  
550 }  
551 }  
552 }  
553 }  
554 }  
555 }  
556 }  
557 }  
558 }  
559 }  
560 }  
561 }  
562 }  
563 }  
564 }  
565 }  
566 }  
567 }  
568 }  
569 }  
570 }  
571 }  
572 }  
573 }  
574 }  
575 }  
576 }  
577 }  
578 }  
579 }  
580 }  
581 }  
582 }  
583 }  
584 }  
585 }  
586 }  
587 }  
588 }  
589 }  
590 }  
591 }  
592 }  
593 }  
594 }  
595 }  
596 }  
597 }  
598 }  
599 }  
600 }  
601 }  
602 }  
603 }  
604 }  
605 }  
606 }  
607 }  
608 }  
609 }  
610 }  
611 }  
612 }  
613 }  
614 }  
615 }  
616 }  
617 }  
618 }  
619 }  
620 }  
621 }  
622 }  
623 }  
624 }  
625 }  
626 }  
627 }  
628 }  
629 }  
630 }  
631 }  
632 }  
633 }  
634 }  
635 }  
636 }  
637 }  
638 }  
639 }  
640 }  
641 }  
642 }  
643 }  
644 }  
645 }  
646 }  
647 }  
648 }  
649 }  
650 }  
651 }  
652 }  
653 }  
654 }  
655 }  
656 }  
657 }  
658 }  
659 }  
660 }  
661 }  
662 }  
663 }  
664 }  
665 }  
666 }  
667 }  
668 }  
669 }  
670 }  
671 }  
672 }  
673 }  
674 }  
675 }  
676 }  
677 }  
678 }  
679 }  
680 }  
681 }  
682 }  
683 }  
684 }  
685 }  
686 }  
687 }  
688 }  
689 }  
690 }  
691 }  
692 }  
693 }  
694 }  
695 }  
696 }  
697 }  
698 }  
699 }  
700 }  
701 }  
702 }  
703 }  
704 }  
705 }  
706 }  
707 }  
708 }  
709 }  
710 }  
711 }  
712 }  
713 }  
714 }  
715 }  
716 }  
717 }  
718 }  
719 }  
720 }  
721 }  
722 }  
723 }  
724 }  
725 }  
726 }  
727 }  
728 }  
729 }  
730 }  
731 }  
732 }  
733 }  
734 }  
735 }  
736 }  
737 }  
738 }  
739 }  
740 }  
741 }  
742 }  
743 }  
744 }  
745 }  
746 }  
747 }  
748 }  
749 }  
750 }  
751 }  
752 }  
753 }  
754 }  
755 }  
756 }  
757 }  
758 }  
759 }  
760 }  
761 }  
762 }  
763 }  
764 }  
765 }  
766 }  
767 }  
768 }  
769 }  
770 }  
771 }  
772 }  
773 }  
774 }  
775 }  
776 }  
777 }  
778 }  
779 }  
780 }  
781 }  
782 }  
783 }  
784 }  
785 }  
786 }  
787 }  
788 }  
789 }  
790 }  
791 }  
792 }  
793 }  
794 }  
795 }  
796 }  
797 }  
798 }  
799 }  
800 }  
801 }  
802 }  
803 }  
804 }  
805 }  
806 }  
807 }  
808 }  
809 }  
810 }  
811 }  
812 }  
813 }  
814 }  
815 }  
816 }  
817 }  
818 }  
819 }  
820 }  
821 }  
822 }  
823 }  
824 }  
825 }  
826 }  
827 }  
828 }  
829 }  
830 }  
831 }  
832 }  
833 }  
834 }  
835 }  
836 }  
837 }  
838 }  
839 }  
840 }  
841 }  
842 }  
843 }  
844 }  
845 }  
846 }  
847 }  
848 }  
849 }  
850 }  
851 }  
852 }  
853 }  
854 }  
855 }  
856 }  
857 }  
858 }  
859 }  
860 }  
861 }  
862 }  
863 }  
864 }  
865 }  
866 }  
867 }  
868 }  
869 }  
870 }  
871 }  
872 }  
873 }  
874 }  
875 }  
876 }  
877 }  
878 }  
879 }  
880 }  
881 }  
882 }  
883 }  
884 }  
885 }  
886 }  
887 }  
888 }  
889 }  
890 }  
891 }  
892 }  
893 }  
894 }  
895 }  
896 }  
897 }  
898 }  
899 }  
900 }  
901 }  
902 }  
903 }  
904 }  
905 }  
906 }  
907 }  
908 }  
909 }  
910 }  
911 }  
912 }  
913 }  
914 }  
915 }  
916 }  
917 }  
918 }  
919 }  
920 }  
921 }  
922 }  
923 }  
924 }  
925 }  
926 }  
927 }  
928 }  
929 }  
930 }  
931 }  
932 }  
933 }  
934 }  
935 }  
936 }  
937 }  
938 }  
939 }  
940 }  
941 }  
942 }  
943 }  
944 }  
945 }  
946 }  
947 }  
948 }  
949 }  
950 }  
951 }  
952 }  
953 }  
954 }  
955 }  
956 }  
957 }  
958 }  
959 }  
960 }  
961 }  
962 }  
963 }  
964 }  
965 }  
966 }  
967 }  
968 }  
969 }  
970 }  
971 }  
972 }  
973 }  
974 }  
975 }  
976 }  
977 }  
978 }  
979 }  
980 }  
981 }  
982 }  
983 }  
984 }  
985 }  
986 }  
987 }  
988 }  
989 }  
990 }  
991 }  
992 }  
993 }  
994 }  
995 }  
996 }  
997 }  
998 }  
999 }
```

```
274         break;
275     case LocationSettingsStatusCodes.SETTINGS_CHANGE_UNAVAILABLE:
276         Snackbar.make(layoutActivity, "Algum motivo desconhecido impediu a aquisição de sua localização",
277             Snackbar.LENGTH_LONG).setActionTextColor(Color.RED).show();
278         imgErroGps.setVisibility(View.VISIBLE);
279         txtErroGps.setVisibility(View.VISIBLE);
280         barraDeProgresso.setVisibility(View.GONE);
281         break;
282     }
283     @Override
284     protected void onResume() {
285         super.onResume();
286     }
287     private void abrirPreferencias(String resultado) {
288         Bundle b = new Bundle();
289         b.putString("municipioAtual", resultado);
290         preferenciasFragment.setArguments(b);
291         getFragmentManager().beginTransaction()
292             .replace(android.R.id.content, preferenciasFragment, "preferenciasFragment")
293             .commit();
294         botaoConfirmar.setVisibility(true);
295     }
296     @Override
297     public void googlePlayServiceIndisponivel() {
298         onBackPressed();
299         finish();
300     }
301     private class CarregadorDados extends AsyncTask<Void, Void, String> {
302         private Context contexto;
303     }
304     ...
305 }
```

```
308     private EstabelecimentoDAO estabDAO;
309     private String resultado = "";
310
311     public CarregadorDeDados(Context contexto) {
312         this.contexto = contexto;
313     }
314
315     @Override
316     protected void onPreExecute() {
317         super.onPreExecute();
318         imgErroGps.setVisibility(View.GONE);
319         txtErroGps.setVisibility(View.GONE);
320         barraDeProgresso.setVisibility(View.VISIBLE);
321     }
322
323     @Override
324     protected String doInBackground(Void... params) {
325         Log.e("Estabelecimentos", "Carregando dados em background");
326         estabDAO = EstabelecimentoDAO.getInstance(getApplicationContext());
327         String resultado = estabDAO.buscaCodigoMunicipioAtual(localizacao);
328         Log.e("resultadoMunicipio", resultado);
329         return resultado;
330     }
331
332     @Override
333     protected void onPostExecute(String resultado) {
334         super.onPostExecute(resultado);
335         imgErroGps.setVisibility(View.GONE);
336         txtErroGps.setVisibility(View.GONE);
337         barraDeProgresso.setVisibility(View.GONE);
338         abrirPreferencias(resultado);
339     }
340 }
--
```

File - E:\Android\Projetos\CadeSaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\PrimeirosPassosActivity.java

```
1 package br.com.jucemar_dimon.cadesaude;
2
3 import android.content.Intent;
4 import android.os.Bundle;
5 import android.support.v7.app.AppCompatActivity;
6 import android.view.View;
7 import android.widget.Button;
8
9 /**
10 * Created by Jucemar on 14/02/2016.
11 */
12 public class PrimeirosPassosActivity extends AppCompatActivity {
13
14     private Button botaoConfig;
15     private static final int CONFIGURACAO_MUNICIPIOS = 1;
16
17     private final View.OnClickListener mDelayHideTouchListener = new View.OnClickListener() {
18         @Override
19         public void onClick(View v) {
20             Intent i = new Intent(PrimeirosPassosActivity.this, PrimeiraConfiguracaoActivity.class);
21             startActivity(i);
22             finish();
23         }
24     };
25
26     @Override
27     protected void onCreate(Bundle savedInstanceState) {
28         super.onCreate(savedInstanceState);
29         setContentView(R.layout.activity_primeiros_passos);
30         botaoConfig = (Button) findViewById(R.id.dummy_button);
31         botaoConfig.setOnClickListener(mDelayHideTouchListener);
32     }
33
34     @Override
35     public void onBackPressed() {
36         --
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\PrimeirosPassosActivity.java

```
36     SairDialog s = new SairDialog();
37     s.show(getFragmentManager(), "Sairdialog");
38 }
39
40 }
41
```

```
File - E:\Android\Projetos\CadESEsauda\app\src\main\java\br\com\jucemar_dimon\cadeseade\ProfissionaisDAO.java

1 package br.com.jucemar_dimon.cadeseade;
2
3 import android.content.Context;
4 import android.database.Cursor;
5 import android.database.sqlite.SQLiteDatabase;
6 import android.util.Log;
7
8 import java.util.ArrayList;
9 import java.util.Set;
10
11 /**
12 * Created by Jucemar on 10/03/2016.
13 */
14 public class ProfissionaisDAO {
15
16     private static ProfissionaisDAO instancia;
17     private SQLiteDatabase bd;
18
19     public static ProfissionaisDAO getInstancia(Context contexto) {
20         if (instancia == null) {
21             instancia = new ProfissionaisDAO(contexto);
22         }
23         return instancia;
24     }
25
26     private ProfissionaisDAO(Context contexto) {
27         BancoDeDados bancoDeDados = BancoDeDados.getInstancia(contexto);
28         bd = bancoDeDados.getWritableDatabase();
29     }
30
31     public CharSequence[] listarTiposProfissionais(Set<String> setMunicipios) {
32
33         CharSequence[] resultado=null;
34
35         if(setMunicipios.size()>0) {
36
37             Cursor cursor = bd.rawQuery("SELECT tipo FROM tipos WHERE municipio IN (" + setMunicipios + ")", null);
38
39             if(cursor != null) {
40                 resultado = new CharSequence[cursor.getCount()];
41
42                 for (int i = 0; i < resultado.length; i++) {
43                     cursor.moveToNext();
44                     resultado[i] = cursor.getString(0);
45                 }
46             }
47
48             cursor.close();
49         }
50
51         return resultado;
52     }
53 }
```

```
36
37     StringBuffer filtro = new StringBuffer();
38     String[] municipios = setMunicipios.toArray(new String[setMunicipios.size()]);
39
40
41
42
43     for (int i = 0; i < municipios.length; i++) {
44         filtro.append("estabelecimento." + TabelaEstabelecimento.Colunas.CODIGO_MUNICIPIO + "=?" );
45         if (i < municipios.length - 1) {
46             filtro.append("OR ");
47         }
48     }
49 }
50
51     String query = "SELECT distinct profissional.cbo_descricao FROM estabelecimento JOIN
52     vinculo_profissional_estabelecimento ON cnes=vinculo.profissional.estabelecimento.estabelecimento_cnes" +
53     "+filtro+" "order by profissional.cbo_descricao";
54     Log.e("query",query);
55     Cursor c = bd.rawQuery(query,municipios);
56     resultado=new CharSequence[c.getCount()];
57     int i=0;
58     if (c.moveToFirst()) {
59         do {
60             resultado[i]=c.getString(c.getColumnIndex(TabelaProfissionais.Colunas.CBO_DESCRICA0));
61             i++;
62         } while (c.moveToNext());
63     }
64     c.close();
65 }
66
67
68 }
```

```
69
70
71     return resultado;
72
73 }
74
75     public ArrayList<Estabelecimento> buscaStringEmNomeProfissionais(Set<String> setMunicipios, String sentenca){
76
77         StringBuffer filtroMunicipios = new StringBuffer();
78
79         String[] municipios = setMunicipios.toArray(new String[setMunicipios.size()]);
80
81         for (int i = 0; i < municipios.length; i++) {
82             filtroMunicipios.append("%" + municipios[i] + "%");
83         }
84
85     ArrayList<Estabelecimento> resultado = new ArrayList<Estabelecimento>();
86
87     String query="SELECT * FROM estabelecimento WHERE cnes IN (SELECT DISTINCT estabelecimento.cnes FROM
88     profissional \n" +
89     "JOIN vinculo_profissional_estabelecimento ON profissional.id=vinculo_profissional_estabelecimento.
profissional_id \n" +
90     "JOIN estabelecimento ON estabelecimento.cnes=vinculo_profissional_estabelecimento.
estabelecimento_cnes WHERE profissional.nome LIKE '%" + sentenca + "%' AND estabelecimento.codigo_municipio IN ("+
91     filtroMunicipios+"));" ;
92
93
94     if (c.moveToFirst()) {
95
96         do {
97             Estabelecimento e = EstabelecimentoDAO.extrairDoCursor(c);
98             resultado.add(e);
99         } while (c.moveToNext());
99 }
```

```
100    }
101    c.close();
102
103    return resultado;
104
105  }
106
107  public ArrayList<Estabelecimento> buscaStringEmNomeEspecialidade(Set<String> setMunicipios, String sentenca){
108
109    StringBuffer filtroMunicipios = new StringBuffer();
110
111    String[] municipios = setMunicipios.toArray(new String[setMunicipios.size()]);
112
113    for (int i = 0; i < municipios.length; i++) {
114      filtroMunicipios.append("'" + municipios[i] + "'");
115    }
116
117    ArrayList<Estabelecimento> resultado = new ArrayList<Estabelecimento>();
118
119    String query = "SELECT * FROM estabelecimento WHERE cnes IN (SELECT DISTINCT estabelecimento.cnes FROM
120
121    profissional
122    +
123    "JOIN vinculo_profissional_estabelecimento ON profissional.id=vinculo_profissional_estabelecimento.
124    profissional_id
125    +
126    "JOIN estabelecimento
127    +
128    +
129    +
130    +
131    +
132    +
133    +
134    +
135    +
136    +
137    +
138    +
139    +
140    +
141    +
142    +
143    +
144    +
145    +
146    +
147    +
148    +
149    +
150    +
151    +
152    +
153    +
154    +
155    +
156    +
157    +
158    +
159    +
160    +
161    +
162    +
163    +
164    +
165    +
166    +
167    +
168    +
169    +
170    +
171    +
172    +
173    +
174    +
175    +
176    +
177    +
178    +
179    +
180    +
181    +
182    +
183    +
184    +
185    +
186    +
187    +
188    +
189    +
190    +
191    +
192    +
193    +
194    +
195    +
196    +
197    +
198    +
199    +
200    +
201    +
202    +
203    +
204    +
205    +
206    +
207    +
208    +
209    +
210    +
211    +
212    +
213    +
214    +
215    +
216    +
217    +
218    +
219    +
220    +
221    +
222    +
223    +
224    +
225    +
226    +
227    +
228    +
229    +
230    +
231    +
232    +
233    +
234    +
235    +
236    +
237    +
238    +
239    +
240    +
241    +
242    +
243    +
244    +
245    +
246    +
247    +
248    +
249    +
250    +
251    +
252    +
253    +
254    +
255    +
256    +
257    +
258    +
259    +
260    +
261    +
262    +
263    +
264    +
265    +
266    +
267    +
268    +
269    +
270    +
271    +
272    +
273    +
274    +
275    +
276    +
277    +
278    +
279    +
280    +
281    +
282    +
283    +
284    +
285    +
286    +
287    +
288    +
289    +
290    +
291    +
292    +
293    +
294    +
295    +
296    +
297    +
298    +
299    +
299    +
300    +
301    +
302    +
303    +
304    +
305    +
306    +
307    +
308    +
309    +
310    +
311    +
312    +
313    +
314    +
315    +
316    +
317    +
318    +
319    +
320    +
321    +
322    +
323    +
324    +
325    +
326    +
327    +
328    +
329    +
330    +
331    +
332    +
333    +
334    +
335    +
336    +
337    +
338    +
339    +
339    +
340    +
341    +
342    +
343    +
344    +
345    +
346    +
347    +
348    +
349    +
349    +
350    +
351    +
352    +
353    +
354    +
355    +
356    +
357    +
358    +
359    +
359    +
360    +
361    +
362    +
363    +
364    +
365    +
366    +
367    +
368    +
369    +
369    +
370    +
371    +
372    +
373    +
374    +
375    +
376    +
377    +
378    +
379    +
379    +
380    +
381    +
382    +
383    +
384    +
385    +
386    +
387    +
388    +
389    +
389    +
390    +
391    +
392    +
393    +
394    +
395    +
396    +
397    +
398    +
399    +
399    +
400    +
401    +
402    +
403    +
404    +
405    +
406    +
407    +
408    +
409    +
409    +
410    +
411    +
412    +
413    +
414    +
415    +
416    +
417    +
418    +
419    +
419    +
420    +
421    +
422    +
423    +
424    +
425    +
426    +
427    +
428    +
429    +
429    +
430    +
431    +
432    +
433    +
434    +
435    +
436    +
437    +
438    +
439    +
439    +
440    +
441    +
442    +
443    +
444    +
445    +
446    +
447    +
448    +
449    +
449    +
450    +
451    +
452    +
453    +
454    +
455    +
456    +
457    +
458    +
459    +
459    +
460    +
461    +
462    +
463    +
464    +
465    +
466    +
467    +
468    +
469    +
469    +
470    +
471    +
472    +
473    +
474    +
475    +
476    +
477    +
478    +
479    +
479    +
480    +
481    +
482    +
483    +
484    +
485    +
486    +
487    +
488    +
489    +
489    +
490    +
491    +
492    +
493    +
494    +
495    +
496    +
497    +
498    +
499    +
499    +
500    +
501    +
502    +
503    +
504    +
505    +
506    +
507    +
508    +
509    +
509    +
510    +
511    +
512    +
513    +
514    +
515    +
516    +
517    +
518    +
519    +
519    +
520    +
521    +
522    +
523    +
524    +
525    +
526    +
527    +
528    +
529    +
529    +
530    +
531    +
532    +
533    +
534    +
535    +
536    +
537    +
538    +
539    +
539    +
540    +
541    +
542    +
543    +
544    +
545    +
546    +
547    +
548    +
549    +
549    +
550    +
551    +
552    +
553    +
554    +
555    +
556    +
557    +
558    +
559    +
559    +
560    +
561    +
562    +
563    +
564    +
565    +
566    +
567    +
568    +
569    +
569    +
570    +
571    +
572    +
573    +
574    +
575    +
576    +
577    +
578    +
579    +
579    +
580    +
581    +
582    +
583    +
584    +
585    +
586    +
587    +
588    +
589    +
589    +
590    +
591    +
592    +
593    +
594    +
595    +
596    +
597    +
598    +
599    +
599    +
600    +
601    +
602    +
603    +
604    +
605    +
606    +
607    +
608    +
609    +
609    +
610    +
611    +
612    +
613    +
614    +
615    +
616    +
617    +
618    +
619    +
619    +
620    +
621    +
622    +
623    +
624    +
625    +
626    +
627    +
628    +
629    +
629    +
630    +
631    +
632    +
633    +
634    +
635    +
636    +
637    +
638    +
639    +
639    +
640    +
641    +
642    +
643    +
644    +
645    +
646    +
647    +
648    +
649    +
649    +
650    +
651    +
652    +
653    +
654    +
655    +
656    +
657    +
658    +
659    +
659    +
660    +
661    +
662    +
663    +
664    +
665    +
666    +
667    +
668    +
669    +
669    +
670    +
671    +
672    +
673    +
674    +
675    +
676    +
677    +
678    +
679    +
679    +
680    +
681    +
682    +
683    +
684    +
685    +
686    +
687    +
688    +
689    +
689    +
690    +
691    +
692    +
693    +
694    +
695    +
696    +
697    +
698    +
699    +
699    +
700    +
701    +
702    +
703    +
704    +
705    +
706    +
707    +
708    +
709    +
709    +
710    +
711    +
712    +
713    +
714    +
715    +
716    +
717    +
718    +
719    +
719    +
720    +
721    +
722    +
723    +
724    +
725    +
726    +
727    +
728    +
729    +
729    +
730    +
731    +
732    +
733    +
734    +
735    +
736    +
737    +
738    +
739    +
739    +
740    +
741    +
742    +
743    +
744    +
745    +
746    +
747    +
748    +
749    +
749    +
750    +
751    +
752    +
753    +
754    +
755    +
756    +
757    +
758    +
759    +
759    +
760    +
761    +
762    +
763    +
764    +
765    +
766    +
767    +
768    +
769    +
769    +
770    +
771    +
772    +
773    +
774    +
775    +
776    +
777    +
778    +
779    +
779    +
780    +
781    +
782    +
783    +
784    +
785    +
786    +
787    +
788    +
789    +
789    +
790    +
791    +
792    +
793    +
794    +
795    +
796    +
797    +
798    +
799    +
799    +
800    +
801    +
802    +
803    +
804    +
805    +
806    +
807    +
808    +
809    +
809    +
810    +
811    +
812    +
813    +
814    +
815    +
816    +
817    +
818    +
819    +
819    +
820    +
821    +
822    +
823    +
824    +
825    +
826    +
827    +
828    +
829    +
829    +
830    +
831    +
832    +
833    +
834    +
835    +
836    +
837    +
838    +
839    +
839    +
840    +
841    +
842    +
843    +
844    +
845    +
846    +
847    +
848    +
849    +
849    +
850    +
851    +
852    +
853    +
854    +
855    +
856    +
857    +
858    +
859    +
859    +
860    +
861    +
862    +
863    +
864    +
865    +
866    +
867    +
868    +
869    +
869    +
870    +
871    +
872    +
873    +
874    +
875    +
876    +
877    +
878    +
878    +
879    +
880    +
881    +
882    +
883    +
884    +
885    +
886    +
887    +
888    +
889    +
889    +
890    +
891    +
892    +
893    +
894    +
895    +
896    +
897    +
898    +
899    +
899    +
900    +
901    +
902    +
903    +
904    +
905    +
906    +
907    +
908    +
909    +
909    +
910    +
911    +
912    +
913    +
914    +
915    +
916    +
917    +
918    +
919    +
919    +
920    +
921    +
922    +
923    +
924    +
925    +
926    +
927    +
928    +
929    +
929    +
930    +
931    +
932    +
933    +
934    +
935    +
936    +
937    +
938    +
939    +
939    +
940    +
941    +
942    +
943    +
944    +
945    +
946    +
947    +
948    +
949    +
949    +
950    +
951    +
952    +
953    +
954    +
955    +
956    +
957    +
958    +
959    +
959    +
960    +
961    +
962    +
963    +
964    +
965    +
966    +
967    +
968    +
969    +
969    +
970    +
971    +
972    +
973    +
974    +
975    +
976    +
977    +
978    +
978    +
979    +
980    +
981    +
982    +
983    +
984    +
985    +
986    +
987    +
988    +
989    +
989    +
990    +
991    +
992    +
993    +
994    +
995    +
996    +
997    +
998    +
999    +
999    +
1000    +
1001    +
1002    +
1003    +
1004    +
1005    +
1006    +
1007    +
1008    +
1009    +
1009    +
1010    +
1011    +
1012    +
1013    +
1014    +
1015    +
1016    +
1017    +
1018    +
1019    +
1019    +
1020    +
1021    +
1022    +
1023    +
1024    +
1025    +
1026    +
1027    +
1028    +
1029    +
1029    +
1030    +
1031    +
1032    +
1033    +
1034    +
1035    +
1036    +
1037    +
1038    +
1039    +
1039    +
1040    +
1041    +
1042    +
1043    +
1044    +
1045    +
1046    +
1047    +
1048    +
1049    +
1049    +
1050    +
1051    +
1052    +
1053    +
1054    +
1055    +
1056    +
1057    +
1058    +
1059    +
1059    +
1060    +
1061    +
1062    +
1063    +
1064    +
1065    +
1066    +
1067    +
1068    +
1069    +
1069    +
1070    +
1071    +
1072    +
1073    +
1074    +
1075    +
1076    +
1077    +
1078    +
1079    +
1079    +
1080    +
1081    +
1082    +
1083    +
1084    +
1085    +
1086    +
1087    +
1088    +
1089    +
1089    +
1090    +
1091    +
1092    +
1093    +
1094    +
1095    +
1096    +
1097    +
1098    +
1099    +
1099    +
1100    +
1101    +
1102    +
1103    +
1104    +
1105    +
1106    +
1107    +
1108    +
1109    +
1109    +
1110    +
1111    +
1112    +
1113    +
1114    +
1115    +
1116    +
1117    +
1118    +
1119    +
1119    +
1120    +
1121    +
1122    +
1123    +
1124    +
1125    +
1126    +
1127    +
1128    +
1129    +
1129    +
1130    +
1131    +
1132    +
1133    +
1134    +
1135    +
1136    +
1137    +
1138    +
1139    +
1139    +
1140    +
1141    +
1142    +
1143    +
1144    +
1145    +
1146    +
1147    +
1148    +
1149    +
1149    +
1150    +
1151    +
1152    +
1153    +
1154    +
1155    +
1156    +
1157    +
1158    +
1159    +
1159    +
1160    +
1161    +
1162    +
1163    +
1164    +
1165    +
1166    +
1167    +
1168    +
1169    +
1169    +
1170    +
1171    +
1172    +
1173    +
1174    +
1175    +
1176    +
1177    +
1178    +
1179    +
1179    +
1180    +
1181    +
1182    +
1183    +
1184    +
1185    +
1186    +
1187    +
1188    +
1189    +
1189    +
1190    +
1191    +
1192    +
1193    +
1194    +
1195    +
1196    +
1197    +
1198    +
1199    +
1199    +
1200    +
1201    +
1202    +
1203    +
1204    +
1205    +
1206    +
1207    +
1208    +
1209    +
1209    +
1210    +
1211    +
1212    +
1213    +
1214    +
1215    +
1216    +
1217    +
1218    +
1219    +
1219    +
1220    +
1221    +
1222    +
1223    +
1224    +
1225    +
1226    +
1227    +
1228    +
1229    +
1229    +
1230    +
1231    +
1232    +
1233    +
1234    +
1235    +
1236    +
1237    +
1238    +
1239    +
1239    +
1240    +
1241    +
1242    +
1243    +
1244    +
1245    +
1246    +
1247    +
1248    +
1249    +
1249    +
1250    +
1251    +
1252    +
1253    +
1254    +
1255    +
1256    +
1257    +
1258    +
1259    +
1259    +
1260    +
1261    +
1262    +
1263    +
1264    +
1265    +
1266    +
1267    +
1268    +
1269    +
1269    +
1270    +
1271    +
1272    +
1273    +
1274    +
1275    +
1276    +
1277    +
1278    +
1279    +
1279    +
1280    +
1281    +
1282    +
1283    +
1284    +
1285    +
1286    +
1287    +
1288    +
1289    +
1289    +
1290    +
1291    +
1292    +
1293    +
1294    +
1295    +
1296    +
1297    +
1298    +
1299    +
1299    +
1300    +
1301    +
1302    +
1303    +
1304    +
1305    +
1306    +
1307    +
1308    +
1309    +
1309    +
1310    +
1311    +
1312    +
1313    +
1314    +
1315    +
1316    +
1317    +
1318    +
1319    +
1319    +
1320    +
1321    +
1322    +
1323    +
1324    +
1325    +
1326    +
1327    +
1328    +
1329    +
1329    +
1330    +
1331    +
1332    +
1333    +
1334    +
1335    +
1336    +
1337    +
1338    +
1339    +
1339    +
1340    +
1341    +
1342    +
1343    +
1344    +
1345    +
1346    +
1347    +
1348    +
1349    +
1349    +
1350    +
1351    +
1352    +
1353    +
1354    +
1355    +
1356    +
1357    +
1358    +
1359    +
1359    +
1360    +
1361    +
1362    +
1363    +
1364    +
1365    +
1366    +
1367    +
1368    +
1369    +
1369    +
1370    +
1371    +
1372    +
1373    +
1374    +
1375    +
1376    +
1377    +
1378    +
1379    +
1379    +
1380    +
1381    +
1382    +
1383    +
1384    +
1385    +
1386    +
1387    +
1388    +
1389    +
1389    +
1390    +
1391    +
1392    +
1393    +
1394    +
1395    +
1396    +
1397    +
1398    +
1399    +
1399    +
1400    +
1401    +
1402    +
1403    +
1404    +
1405    +
1406    +
1407    +
1408    +
1409    +
1409    +
1410    +
1411    +
1412    +
1413    +
1414    +
1415    +
1416    +
1417    +
1418    +
1419    +
1419    +
1420    +
1421    +
1422    +
1423    +
1424    +
1425    +
1426    +
1427    +
1428    +
1429    +
1429    +
1430    +
1431    +
1432    +
1433    +
1434    +
1435    +
1436    +
1437    +
1438    +
1439    +
1439    +
1440    +
1441    +
1442    +
1443    +
1444    +
1445    +
1446    +
1447    +
1448    +
1449    +
1449    +
1450    +
1451    +
1452    +
1453    +
1454    +
1455    +
1456    +
1457    +
1458    +
1459    +
1459    +
1460    +
1461    +
1462    +
1463    +
1464    +
1465    +
1466    +
1467    +
1468    +
1469    +
1469    +
1470    +
1471    +
1472    +
1473    +
1474    +
1475    +
1476    +
1477    +
1478    +
1479    +
1479    +
1480    +
1481    +
1482    +
1483    +
1484    +
1485    +
1486    +
1487    +
1488    +
1489    +
1489    +
1490    +
1491    +
1492    +
1493    +
1494    +
1495    +
1496    +
1497    +
1498    +
1499    +
1499    +
1500    +
1501    +
1502    +
1503    +
1504    +
1505    +
1506    +
1507    +
1508    +
1509    +
1509    +
1510    +
1511    +
1512    +
1513    +
1514    +
1515    +
1516    +
1517    +
1518    +
1519    +
1519    +
1520    +
1521    +
1522    +
1523    +
1524    +
1525    +
1526    +
1527    +
1528    +
1529    +
1529    +
1530    +
1531    +
1532    +
1533    +
1534    +
1535    +
1536    +
1537    +
1538    +
1539    +
1539    +
1540    +
1541    +
1542    +
1543    +
1544    +
1545    +
1546    +
1547    +
1548    +
1549    +
1549    +
1550    +
1551    +
1552    +
1553    +
1554    +
1555    +
1556    +
1557    +
1558    +
1559    +
1559    +
1560    +
1561    +
1562    +
1563    +
1564    +
1565    +
1566    +
1567    +
1568    +
1569    +
1569    +
1570    +
1571    +
1572    +
1573    +
1574    +
1575    +
1576    +
1577    +
1578    +
1579    +
1579    +
1580    +
1581    +
1582    +
1583    +
1584    +
1585    +
1586    +
1587    +
1588    +
1589    +
1589    +
1590    +
1591    +
1592    +
1593    +
1594    +
1595    +
1596    +
1597    +
1598    +
1599    +
1599    +
1600    +
1601    +
1602    +
1603    +
1604    +
1605    +
1606    +
1607    +
1608    +
1609    +
1609    +
1610    +
1611    +
1612    +
1613    +
1614    +
1615    +
1616    +
1617    +
1618    +
1619    +
1619    +
1620    +
1621    +
1622    +
1623    +
1624    +
1625    +
1626    +
1627    +
1628    +
1629    +
1629    +
1630    +
1631    +
1632    +
1633    +
1634    +
1635    +
1636    +
1637    +
1638    +
1639    +
1639    +
1640    +
1641    +
1642    +
1643    +
1644    +
1645    +
1646    +
1647    +
1648    +
1649    +
1649    +
1650    +
1651    +
1652    +
1653    +
1654    +
1655    +
1656    +
1657    +
1658    +
1659    +
1659    +
1660    +
1661    +
1662    +
1663    +
1664    +
1665    +
1666    +
1667    +
1668    +
1669    +
1669    +
1670    +
1671    +
1672    +
1673    +
1674    +
1675    +
1676    +
1677    +
1678    +
1679    +
1679    +
1680    +
1681    +
1682    +
1683    +
1684    +
1685    +
1686    +
1687    +
1688    +
1689    +
1689    +
1690    +
1691    +
1692    +
1693    +
1694    +
1695    +
1696    +
1697    +
1698    +
1699    +
1699    +
1700    +
1701    +
1702    +
1703    +
1704    +
1705    +
1706    +
1707    +
1708    +
1709    +
1709    +
1710    +
1711    +
1712    +
1713    +
1714    +
1715    +
1716    +
1717    +
1718    +
1719    +
1719    +
1720    +
1721    +
1722    +
1723    +
1724    +
1725    +
1726    +
1727    +
1728    +
1729    +
1729    +
1730    +
1731    +
1732    +
1733    +
1734    +
1735    +
1736    +
1737    +
1738    +
1739    +
1739    +
1740    +
1741    +
1742    +
1743    +
1744    +
1745    +
1746    +
1747    +
1748    +
1749    +
1749    +
1750    +
1751    +
1752    +
1753    +
1754    +
1755    +
1756    +
1757    +
1758    +
1759    +
1759    +
1760    +
1761    +
1762    +
1763    +
1764    +
1765    +
1766    +
1767    +
1768    +
1769    +
1769    +
1770    +
1771    +
1772    +
1773    +
1774    +
1775    +
1776    +
1777    +
1778    +
1779    +
1779    +
1780    +
1781    +
1782    +
1783    +
1784    +
1785    +
1786    +
1787    +
1788    +
1789    +
1789    +
1790    +
1791    +
1792    +
1793    +
1794    +
1795    +
1796    +
1797    +
1798    +
1799    +
1799    +
1800    +
1801    +
1802    +
1803    +
1804    +
1805    +
1806    +
1807    +
1808    +
1809    +
1809    +
1810    +
1811    +
1812    +
1813    +
1814    +
1815    +
1816    +
1817    +
1818    +
1819    +
1819    +
1820    +
1821    +
1822    +
1823    +
1824    +
1825    +
1826    +
1827    +
1828    +
1829    +
1829    +
1830    +
1831    +
1832    +
1833    +
1834    +
1835    +
1836    +
1837    +
1838    +
1839    +
1839    +
1840    +
1841    +
1842    +
1843    +
1844    +
1845    +
1846    +
1847    +
1848    +
1849    +
1849    +
1850    +
1851    +
1852    +
1853    +
1854    +
1855    +
1856    +
1857    +
1858    +
1859    +
1859    +
1860    +
1861    +
1862    +
1863    +
1864    +
1865    +
1866    +
1867    +
1868    +
1869    +
1869    +
1870    +
1871    +
1872    +
1873    +
1874    +
1875    +
1876    +
1877    +
1878    +
1879    +
1879    +
1880    +
1881    +
1882    +
1883    +
1884    +
1885    +
1886    +
1887    +
1888    +
1889    +
1889    +
1890    +
1891    +
1892    +
1893    +
1894    +
1895    +
1896    +
1897    +
1898    +
1899    +
1899    +
1900    +
1901    +
1902    +
1903    +
1904    +
1905    +
1906    +
1907    +
1908    +
1909    +
1909    +
1910    +
1911    +
1912    +
1913    +
1914    +
1915    +
1916    +
1917    +
1918    +
1919    +
1919    +
192
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\ProfissionaisDAO.java

```
131     } while (c.moveToNext());
132 }
133 c.close();
134 return resultado;
135 }
136 }
137 }

138 public ArrayList<Profissional> listaProfissionais(String cnes) {
139     String[] argumentos = {cnes};

140     String query = "SELECT * FROM vinculo_profissional_estabelecimento JOIN profissional ON
141 vinculo_profissional_estabelecimento.profissional_id=profissional.id WHERE vinculo_profissional_estabelecimento.
142 estabelecimento_cnes=?";
143 Cursor c = bd.rawQuery(query, argumentos);

144
145 Log.i("Cursor", String.valueOf(c.getCount()));
146 ArrayList<Profissional> resultado = new ArrayList<Profissional>;
147 //Cursor c = bd.query(TabelaProfissionais.NOME_TABLEA, null, TabelaProfissionais.Columnas.ID + "=?", 
148 idsProfissionais, null, null, TabelaProfissionais.Columnas.NOME);
149 if (c.moveToFirst()) {
150     do {
151         Profissional P = extrairDoCursor(c);
152         resultado.add(P);
153     } while (c.moveToNext());
154 }
155 c.close();
156 Log.e("ResultadoBuscaProf", String.valueOf(resultado.size()));
157 return resultado;
158 }
159 }

160 private static Profissional extrairDoCursor(Cursor c) {
161     String nome = c.getString(c.getColumnIndex(TabelaProfissionais.Columnas.NOME));
162 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\ProfissionaisDAO.java

```
163     String cbo = c.getString(c.getColumnIndex(Colunas.CBO));
164     String cboDescricao = c.getString(c.getColumnIndex(Colunas.CBO_DESCRICAO));
165
166     return new Profissional(nome, cbo, cboDescricao);
167   }
168 }
169 }
```

```
1 package br.com.jucemar_dimon.cadesaude;
2
3 /**
4 * Created by Jucemar on 18/03/2016.
5 */
6
7 import android.os.Bundle;
8 import android.support.v4.app.Fragment;
9 import android.view.LayoutInflater;
10 import android.view.View;
11 import android.view.ViewGroup;
12 import android.widget.ListView;
13
14 import java.util.ArrayList;
15
16 public class ProfissionaisFragment extends Fragment {
17
18     public ProfissionaisFragment() {
19
20
21         @Override
22         public void onCreate(Bundle savedInstanceState) {
23             super.onCreate(savedInstanceState);
24
25             @Override
26             public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
27                 View v = inflater.inflate(R.layout.fragment_listas, container, false);
28                 ListView lv = (ListView) v.findViewById(R.id.list_view);
29                 lv.setDivider(null);
30
31                 ArrayAdapter<Profissional> listaDeProfissionais = getArguments().getParcelableArrayList("listadeProfissionais");
32                 ProfissionalAdapter pa = new ProfissionalAdapter(getActivity(), listaDeProfissionais);
33                 lv.setAdapter(pa);
34             }  
        }  
    }  
}
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\ProfissionaisFragment.java

```
36     ArrayList<Profissional> erro = new ArrayList<Profissional>();
37     erro.add(new Profissional("Não há informações cadastradas", "Desculpe o transtorno"));
38     ProfessionalAdapter sa = new ProfessionalAdapter(getActivity(), erro);
39     lv.setAdapter(sa);
40 }
41 return v;
42 }
43 }
44 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\Profissional.java

```
1 package br.com.jucemar_dimon.cadesaude;
2
3 import android.os.Parcel;
4 import android.os.Parcelable;
5
6 /**
7 * Created by Jucemar on 14/02/2016.
8 */
9 public class Profissional implements Parcelable {
10
11     private String nome;
12     private String cbo;
13     private String cboDescricao;
14
15     public Profissional() {
16     }
17
18     public Profissional(String nome, String cbo, String cboDescricao) {
19         this.nome = nome;
20         this.cbo = cbo;
21         this.cboDescricao = cboDescricao;
22     }
23
24     protected Profissional(Parcel in) {
25         nome = in.readString(); cbo =
26         in.readString(); cboDescricao =
27         in.readString();
28     }
29
30     public static final Creator<Profissional> CREATOR = new Creator<Profissional>() {
31         @Override
32         public Profissional createFromParcel(Parcel in) {
33             return new Profissional(in);
34         }
35     }
36 }
```

File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\Profissional.java

```
36     @Override
37     public Profissional[] newArray(int size) {
38         return new Profissional[size];
39     }
40 }
41
42     public Profissional(String s, String s1) {
43         this.nome = s;
44         this.cboDescricao = s1;
45     }
46
47     public String getNome() {
48         return nome;
49     }
50
51     public void setNome(String nome) {
52         this.nome = nome;
53     }
54
55     public String getCbo() {
56         return cbo;
57     }
58
59     public void setCbo(String cbo) {
60         this.cbo = cbo;
61     }
62
63     public String getCboDescricao() {
64         return cboDescricao;
65     }
66
67     public void setCboDescricao(String cboDescricao) {
68         this.cboDescricao = cboDescricao;
69     }
--
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\Profissional.java

```
71     @Override  
72     public int describeContents() {  
73         return 0;  
74     }  
75  
76     @Override  
77     public void writeToParcel(Parcel dest, int flags) {  
78         dest.writeString(nome);  
79         dest.writeString(cbo);  
80         dest.writeString(cboDescricao);  
81     }  
82 }  
83 }  
84 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\ProfissionalAdapter.java

```
1 package br.com.jucemar_dimon.cadesaude;
2
3 import android.content.Context;
4 import android.view.LayoutInflater;
5 import android.view.View;
6 import android.view.ViewGroup;
7 import android.widget.ArrayAdapter;
8 import android.widget.TextView;
9
10 import java.util.ArrayList;
11
12 /**
13 * Created by Jucemar on 19/03/2016.
14 */
15 public class ProfissionalAdapter extends ArrayAdapter<Profissional> {
16
17     public ProfissionalAdapter(Context context, ArrayList<Profissional> objects) {
18         super(context, R.layout.fragment_listas, objects);
19     }
20
21     @Override
22     public View getView(int position, View convertView, ViewGroup parent) {
23         Profissional p = getItem(position);
24         if (convertView == null) {
25             convertView = LayoutInflater.from(getContext()).inflate(R.layout.item_lists, parent, false);
26         }
27         TextView tvName = (TextView) convertView.findViewById(R.id.item_Linha_1);
28         TextView tvCargo = (TextView) convertView.findViewById(R.id.item_Linha_2);
29         tvName.setText(p.getNome());
30         tvCargo.setText(p.getDescricao());
31         return convertView;
32     }
33 }
34 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\SairDialog.java

```
1 package br.com.jucemar_dimon.cadesaude;
2
3 import android.app.Activity;
4 import android.app.AlertDialog;
5 import android.app.Dialog;
6 import android.app.DialogFragment;
7 import android.content.DialogInterface;
8 import android.os.Bundle;
9
10 /**
11 * Created by Jucemar on 18/08/2016.
12 */
13 public class SairDialog extends DialogFragment implements DialogInterface.OnClickListener {
14     private PrimeirosPassosActivity contexto;
15
16     @Override
17     public void onAttach(Activity activity) {
18         super.onAttach(activity);
19         this.contexto=(PrimeirosPassosActivity)activity;
20     }
21
22     public SairDialog() {
23
24     }
25
26     @Override
27     public void onClick(DialogInterface dialog, int which) {
28         switch (which) {
29             case DialogInterface.BUTTON_POSITIVE:
30                 contexto.finishAffinity();
31                 System.exit(0);
32             break;
33             case DialogInterface.BUTTON_NEGATIVE:
34                 dismiss();
35         }
36     }
37 }
```

File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\SairDialog.java

```
36
37     }
38 }
39
40 @Override
41 public Dialog onCreateDialog(Bundle savedInstanceState) { AlertDialog.Builder
42         builder = new AlertDialog.Builder(getActivity());
43         builder.setMessage("Confirma a saída do aplicativo antes de configurar?")
44         .setTitle("Confirmação de saída")
45         .setPositiveButton("Sair", this)
46         .setNegativeButton("Cancelar", this);
47
48     return builder.create();
49 }
50 }
51
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\SeekBarPreference.java

```
1 package br.com.jucemar_dimon.cadesaude;
2 /**
3 * Created by Jucemar on 04/05/2016.
4 */
5
6 import android.content.Context;
7 import android.preference.Preference;
8 import android.util.AttributeSet;
9 import android.view.LayoutInflater;
10 import android.view.View;
11 import android.view.ViewGroup;
12 import android.widget.SeekBar;
13 import android.widget.SeekBar.OnSeekBarChangeListener;
14
15 public class SeekBarPreference extends Preference implements OnSeekBarChangeListener {
16     private SeekBar mSeekBar;
17     private int mProgress;
18
19     public SeekBarPreference(Context context) {
20         super(context);
21     }
22
23     public SeekBarPreference(Context context, AttributeSet attrs) {
24         super(context, attrs);
25     }
26
27     public SeekBarPreference(Context context, AttributeSet attrs, int defStyle) {
28         super(context, attrs, defStyle);
29     }
30
31     @Override
32     protected View onCreateView(ViewGroup parent) {
33         super.onCreateView(parent);
34         LayoutInflator inflater = LayoutInflater.from(getContext().getSystemService(Context.LAYOUT_INFLATER_SERVICE));
35         View view = inflater.inflate(R.layout.preference_seekbar, parent, false);
36     }
37 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\SeekBarPreference.java

```
36     mSeekBar = (SeekBar) view.findViewById(R.id.seekbar);
37     mSeekBar.setProgress(mProgress);
38     mSeekBar.setOnSeekBarChangeListener(this);
39     mSeekBar.setMax(25);
40     return view;
41 }
42
43 @Override
44 public void onProgressChanged(SeekBar seekBar, int progress, boolean fromUser) {
45     if (!fromUser)
46         return;
47     if (progress == 0) {
48         progress = progress + 1;
49     }
50     setValue(progress);
51 }
52
53 @Override
54 public void onStartTrackingTouch(SeekBar seekBar) {
55 }
56
57 @Override
58 public void onStopTrackingTouch(SeekBar seekBar) {
59 }
60
61 @Override
62 protected void onSetInitialValue(boolean restoreValue, Object defaultValue) {
63     setValue(restoreValue ? getPersistedInt(mProgress) : (Integer) defaultValue);
64 }
65
66 public void setValue(int value) {
67     if (shouldPersist()) {
68         persistInt(value);
69     }
70     if (value != mProgress) {
71         mProgress = value;
72         if (mOnSeekBarChangeListener != null)
73             mOnSeekBarChangeListener.onProgressChanged(seekBar, value, false);
74     }
75 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\SeekBarPreference.java

```
71     mProgress = value;
72     notifyChanged();
73 }
74 }
75 }
76 }
77
78
79
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\ServicoClassificacao.java

```
1 package br.com.jucemar_dimon.cadesaude;
2
3 import android.os.Parcel;
4 import android.os.Parcelable;
5
6 /**
7 * Created by Jucemar on 14/02/2016.
8 */
9 public class ServicoClassificacao implements Parcelable {
10     private String codigoServiço;
11     private String descriçãoServiço;
12     private String descriçãoClassificacao;
13     private String códigoClassificacao;
14
15     public ServicoClassificacao(String códigoServiço, String descriçãoServiço, String códigoClassificacao, String
16 descriçãoClassificacao) {
17         this.descriçãoServiço = descriçãoServiço;
18         this.codigoClassificacao = códigoClassificacao;
19         this.codigoServiço = códigoServiço;
20     }
21
22     protected ServicoClassificacao(Parcel in) {
23         códigoServiço = in.readString();
24         descriçãoServiço = in.readString();
25         descriçãoClassificacao = in.readString();
26         códigoClassificacao = in.readString();
27     }
28
29     public static final Creator<ServicoClassificacao> CREATOR = new Creator<ServicoClassificacao>() {
30         @Override
31         public ServicoClassificacao createFromParcel(Parcel in) {
32             return new ServicoClassificacao(in);
33         }
34     }
}
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\ServicoClassificacao.java

```
35     @Override
36     public ServicoClassificacao[] newArray(int size) {
37         return new ServicoClassificacao[size];
38     }
39 }
40
41     public ServicoClassificacao(String descricaoServico, String descricaoClassificacao) {
42         this.descricaoClassificacao = descricaoClassificacao;
43         this.descricaoServico = descricaoServico;
44     }
45
46     @Override
47     public int describeContents() {
48         return 0;
49     }
50
51     @Override
52     public void writeToParcel(Parcel dest, int flags) {
53         dest.writeString(codigoServico);
54         dest.writeString(descricaoServico);
55         dest.writeString(descricaoClassificacao);
56         dest.writeString(codigoClassificacao);
57     }
58
59     public String getCodigoServico() {
60         return codigoServico;
61     }
62
63     public void setCodigoServico(String codigoServico) {
64         this.codigoServico = codigoServico;
65     }
66
67     public String getDescricaoServico() {
68         return descricaoServico;
69     }
70 }
```

File - E:\Android\Projetos\CadeSaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\ServicoClassificacao.java

```
70
71     public void setDescricaoServico(String descricaoServico) {
72         this.descricaoServico = descricaoServico;
73     }
74
75     public String getDescricaoClassificacao() {
76         return descricaoClassificacao;
77     }
78
79     public void setDescricaoClassificacao(String descricaoClassificacao) {
80         this.descricaoClassificacao = descricaoClassificacao;
81     }
82
83     public String getCodigoClassificacao() {
84         return codigoClassificacao;
85     }
86
87     public void setCodigoClassificacao(String codigoClassificacao) {
88         this.codigoClassificacao = codigoClassificacao;
89     }
90
91 }
```

File - E:\Android\Projetos\CadeSaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\ServicoClassificacaoDAO.java

```
1 package br.com.jucemar_dimon.cadesaude;
2
3 import android.content.Context;
4 import android.database.Cursor;
5 import android.database.sqlite.SQLiteDatabase;
6 import android.util.Log;
7
8 import java.util.ArrayList;
9 import java.util.Set;
10 /**
11 * **/ 
12 * Created by Jucemar on 19/03/2016.
13 */
14 public class ServicoClassificacaoDAO {
15     private static ServicoClassificacaoDAO instancia;
16     private SQLiteDatabase bd;
17
18     public static ServicoClassificacaoDAO getInstancia(Context contexto) {
19         if (instancia == null) {
20             instancia = new ServicoClassificacaoDAO(contexto);
21         }
22         return instancia;
23     }
24
25     private ServicoClassificacaoDAO(Context contexto) {
26         BancoDeDados bancoDeDados = BancoDeDados.getInstancia(contexto);
27         bd = bancoDeDados.getWritableDatabase();
28     }
29
30     public ArrayList<ServicoClassificacao> listarServicosEspecializados(String cnes) {
31         String[] argumentos = {cnes};
32         String query = "SELECT s.descricao as servico, c.descricao as classificacao FROM servico s JOIN classificacao c ON s.id=c.servico_id JOIN servico_classificacao_estabelecimento ON servico_classificacao_estabelecimento.servico_id=c.id WHERE servico_classificacao_estabelecimento.estabelecimento_cnes=?";
33     }
34 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\ServicoClassificacaoDAO.java

```
34     Cursor c = bd.rawQuery(query, argumentos);
35     Log.i("Cursor", String.valueOf(c.getCount()));
36     ArrayList<ServicoClassificacao> resultado = new ArrayList<ServicoClassificacao>;
37     if (c.moveToFirst()) {
38         do {
39             ServicoClassificacao sc = extrairDoCursor(c);
40             resultado.add(sc);
41             while (c.moveToNext());
42         }
43         c.close();
44         Log.e("ResultadoBuscaProf", String.valueOf(resultado.size()));
45         return resultado;
46     }
47     private static ServicoClassificacao extrairDoCursor(Cursor c) {
48         String servico = c.getString(0);
49         String classificacao = c.getString(1);
50         return new ServicoClassificacao(servico.toUpperCase(), classificacao.toUpperCase());
51     }
52     public CharSequence[] listarTiposServicos(Set<String> setMunicipios) {
53         CharSequence[] resultado = null;
54         if (setMunicipios.size() > 0) {
55             StringBuffer filtroMunicipios = new StringBuffer();
56             String[] municipios = setMunicipios.toArray(new String[setMunicipios.size()]);
57             for (int i = 0; i < municipios.length; i++) {
58                 filtroMunicipios.append("•" + municipios[i] + "•");
59                 if (i < municipios.length - 1) {
60                     filtroMunicipios.append(",");
61                 }
62             }
63         }
64         String query = "SELECT DISTINCT servico.descricao FROM estabelecimento " +
65         "JOIN servico_classificacao_estabelecimento ON estabelecimento.cnes=" +
66         "servico_classificacao_estabelecimento.estabelecimento_cnes " +
67         "JOIN servico_classificacao ON servico_classificacao.estabelecimento.classificacao_id " +
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\ServicoClassificacaoDAO.java

```
68     + filtroMunicipios + ") ORDER BY servico.descricao;" ;
69     Cursor c = bd.rawQuery(query, null);
70     resultado = new CharSequence[c.getCount()];
71     int i = 0;
72     if (c.moveToFirst()) {
73         do {
74             resultado[i] = c.getString(c.getColumnIndex(TabelaServico.Columnas.DESCRICAO));
75             i++;
76         } while (c.moveToNext());
77         c.close();
78     }
79     return resultado;
80 }
81
82 public ArrayList<Estabelecimento> buscaStringEmServicos(Set<String> setMunicipios, String sentenca) {
83     StringBuffer filtroMunicipios = new StringBuffer();
84     String[] municipios = setMunicipios.toArray(new String[setMunicipios.size()]);
85     for (int i = 0; i < municipios.length; i++) {
86         filtroMunicipios.append("''' + municipios[i] + "''");
87         if (i < municipios.length - 1) {
88             filtroMunicipios.append(",");
89         }
90     }
91     String query = "SELECT * FROM estabelecimento WHERE enes IN (SELECT DISTINCT estabelecimento FROM
92     servico_classificacao_estacao_estabelecimento ON servico_classificacao_estabelecimento.estabelecimento_-
93     enes\n" +
94     "JOIN classificacao ON servico_classificacao.servico_id=estabelecimento.classificacao.servico_id\n" +
95     "JOIN servico ON servico.id=classificacao.servico_id\n" +
96     "WHERE (servico.descricao LIKE '%' + sentenca + '%' OR classificacao.descricao LIKE '%' + sentenca +
97     '%' ) AND estabelecimento.codigo_municipio IN (" + filtroMunicipios + "));";
98     ArrayList<Estabelecimento> resultado = new ArrayList<Estabelecimento>();
99     Cursor c = bd.rawQuery(query, null);
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\ServicoClassificacaoDAO.java

```
99     if (c.moveToFirst()) {
100         do {
101             Estabelecimento e = EstabelecimentoDAO.extrairDoCursor(c);
102             resultado.add(e);
103         } while (c.moveToNext());
104     }
105     c.close();
106     return resultado;
107 }
108 }
109 }
110
111
112
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\ServicosAdapter.java

```
1 package br.com.jucemar_dimon.cadesaude;
2
3 import android.content.Context;
4 import android.view.LayoutInflater;
5 import android.view.View;
6 import android.view.ViewGroup;
7 import android.widget.ArrayAdapter;
8 import android.widget.TextView;
9
10 import java.util.ArrayList;
11
12 /**
13 * Created by Jucemar on 19/03/2016.
14 */
15 public class ServicosAdapter extends ArrayAdapter<ServicoClassificacao> {
16
17     public ServicosAdapter(Context context, ArrayList<ServicoClassificacao> objects) {
18         super(context, R.layout.fragment_listas, objects);
19     }
20
21     @Override
22     public View getView(int position, View convertView, ViewGroup parent) {
23         ServicoClassificacao sc = getItem(position);
24         if (convertView == null) {
25             convertView = LayoutInflater.from(getContext()).inflate(R.layout.item_listas, parent, false);
26         }
27         TextView tvName = (TextView) convertView.findViewById(R.id.item_linha_1);
28         TextView tvCargo = (TextView) convertView.findViewById(R.id.item_linha_2);
29         tvName.setText(sc.getDescricaoServicoO);
30         tvCargo.setText(sc.getDescricaoClassificacaoO);
31         return convertView;
32     }
33
34 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\ServicosFragment.java

```
1 package br.com.jucemar_dimon.cadesaude;
2
3 import android.os.Bundle;
4 import android.support.v4.app.Fragment;
5 import android.view.LayoutInflater;
6 import android.view.View;
7 import android.view.ViewGroup;
8 import android.widget.ListView;
9
10 import java.util.ArrayList;
11
12 /**
13 * Created by Jucemar on 18/03/2016.
14 */
15 public class ServicosFragment extends Fragment {
16
17     @Override
18     public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
19         View v = inflater.inflate(R.layout.fragment_listas, container, false);
20         ArrayList<ServicoClassificacao> listaDeServicos = getArguments().getParcelableArrayList("listaDeServicos");
21         ListView lv = (ListView) v.findViewById(R.id.list_view);
22         lv.setDivider(null);
23         if (listaDeServicos.size() > 0) {
24             ServicosAdapter sa = new ServicosAdapter(getActivity(), listaDeServicos);
25             lv.setAdapter(sa);
26         } else {
27             ArrayList<ServicoClassificacao> erro = new ArrayList<ServicoClassificacao>;
28             erro.add(new ServicoClassificacao("Não há informações cadastradas", "Desculpe o transtorno"));
29             ServicosAdapter sa = new ServicosAdapter(getActivity(), erro);
30             lv.setAdapter(sa);
31         }
32     }
33 }
34 }
```

36  
37  
38

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\SobreActivity.java

```
1 package br.com.jucemar_dimon.cadesaude;
2
3 import android.content.pm.PackageInfo;
4 import android.content.pm.PackageManager;
5 import android.os.Bundle;
6 import android.support.v7.app.AppCompatActivity;
7 import android.support.v7.widget.Toolbar;
8 import android.view.MenuItem;
9 import android.widget.TextView;
10
11 public class SobreActivity extends AppCompatActivity {
12
13     private Toolbar toolbar;
14     private TextView txvAppVersao;
15     private TextView txvBdVersao;
16
17     @Override
18     protected void onCreate(Bundle savedInstanceState) {
19         super.onCreate(savedInstanceState);
20         setContentView(R.layout.activity_sobre);
21         toolbar = (Toolbar) findViewById(R.id.toolbar_actvt_sobre);
22         setSupportActionBar(toolbar);
23         getSupportActionBar().setDisplayHomeAsUpEnabled(true);
24         toolbar.setTitle("Sobre");
25         txvAppVersao = (TextView) findViewById(R.id.sobre_txv_app_versao);
26         txvBdVersao = (TextView) findViewById(R.id.sobre_txv_bd_versao);
27         PackageManager pInfo = getPackageManager();
28         try {
29             pInfo.getPackageInfo(getPackageName(), 0);
30         } catch (PackageManager.NameNotFoundException e) {
31             e.printStackTrace();
32         }
33         String appVersao = pInfo.getPackageName() + " " + pInfo.getVersionName();
34         txvAppVersao.setText(getString(R.string.appVersao) + " " + appVersao);
35         txvBdVersao.setText(getString(R.string.bdVersao) + " " + BancoDeDados.BD_VERSAO);
36     }
37 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\SobreActivity.java

```
36
37
38     @Override
39     public boolean onOptionsItemSelected(MenuItem item) {
40         int id = item.getItemId();
41         switch (id) {
42             case android.R.id.home:
43                 onBackPressed();
44                 break;
45         }
46         return super.onOptionsItemSelected(item);
47     }
48
49 }
50 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\StringUtils.java

```
1 package br.com.jucemar_dimon.cadesaude;
2
3 import android.util.Log;
4
5 import java.text.Normalizer;
6
7 /**
8 * Created by Jucemar on 20/04/2016.
9 */
10 public class StringUtils {
11
12     public static String removerAcentos(String str) {
13         CharSequence cs = new StringBuilder(str == null ? "" : str);
14         return Normalizer.Form.NFKD.replaceAll(cs, "\p{InCombiningDiacriticalMarks}+");
15     }
16
17     public static String formatarTelefones(String telefoneSujeito) {
18         StringBuffer telefoneFinal = new StringBuffer();
19         StringBuffer telefoneLimpo = new StringBuffer();
20         for (int i = 0; i < telefoneSujeito.length(); i++) {
21             String s = telefoneSujeito.substring(i);
22             try {
23                 Integer.parseInt(s);
24                 telefoneLimpo.append(s);
25             } catch (NumberFormatException e) {
26                 Log.e("Telefone", "Caractere inválido no telefone: " + s);
27             }
28             Log.e("Telefone", "telefone: " + telefoneLimpo.toString());
29             if (telefoneLimpo.length() == 10) {
30                 String fim = (String) telefoneLimpo.subSequence(2, 10);
31                 String inicio = (String) telefoneLimpo.subSequence(0, 2);
32                 if (telefoneLimpo.charAt(2) == '8' || telefoneLimpo.charAt(2) == '9') {
33                     telefoneFinal.append(inicio);
34                     telefoneFinal.append("9");
35                 }
36             }
37         }
38     }
39 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\StringUtils.java

```
36     telefoneFinal.append(fim);
37     Log.e("Telefone", " inicio.toString());
38     Log.e("Telefone", " fim.toString());
39     Log.e("Telefone", " telefoneFinal.toString());
40 } else {
41     telefoneFinal.append("0");
42     telefoneFinal.append(inicio);
43     telefoneFinal.append(fim);
44 } else {
45     telefoneFinal = telefoneLimpo;
46 }
47 return telefoneFinal.toString();
48 }
49 }
50 }
51 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\TabelaAtendimento.java

```
1 package br.com.jucemar_dimon.cadesaude;
2
3 /**
4 * Created by Jucemar on 10/03/2016.
5 */
6 public final class TabelaAtendimento {
7     public static final String NOME_TABELA = "atendimento_estabelecimento";
8
9     public static final class Colunas {
10         public static final String ID = "id";
11         public static final String ATENDIMENTO = "atendimento";
12         public static final String CONVENIO = "convenio";
13         public static final String ESTABELECIMENTO_CNES = "estabelecimento_cnes";
14     }
15 }
16 }
17 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\TabelaClassificacao.java

```
1 package br.com.jucemar_dimon.cadesaude;
2
3 /**
4 * Created by Jucemar on 10/03/2016.
5 */
6 public final class TabelaClassificacao {
7     public static final String NOME_TABELA = "classificacao";
8
9     public static final class Colunas {
10         public static final String ID = "id";
11         public static final String SERVICO_ID = "servico_id";
12         public static final String CODIGO = "codigo";
13         public static final String DESCRICAO = "descricao";
14     }
15 }
16 }
17 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\TabelaEstabelecimento.java

```
1 package br.com.jucemar_dimon.cadesaude;
2
3 /**
4 * Created by Jucemar on 10/03/2016.
5 */
6 public final class TabelaEstabelecimento {
7     public static final String NOME_TABELA = "estabelecimento";
8
9     public static final class Colunas {
10         public static final String CNES = "cnes";
11         public static final String RAZAO_SOCIAL = "razao_social";
12         public static final String NOME_FANTASIA = "nome_fantasia";
13         public static final String TELEFONE = "telefone";
14         public static final String NOME_LOGRADOURO = "nome_logradouro";
15         public static final String NUMERO = "numero";
16         public static final String COMPLEMENTO = "complemento";
17         public static final String BAIRRO = "bairro";
18         public static final String MUNICIPIO = "municipio";
19         public static final String CODIGO_MUNICIPIO = "codigo_municipio";
20         public static final String ESTADO = "estado";
21         public static final String CEP = "cep";
22         public static final String LATITUDE = "Latitude";
23         public static final String LONGITUDE = "Longitude";
24     }
25
26 }
27 ~o
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\TabelaProfissionais.java

```
1 package br.com.jucemar_dimon.cadesaude;
2
3 /**
4 * Created by Jucemar on 10/03/2016.
5 */
6 public final class TabelaProfissionais {
7     public static final String NOME_TABELA="profissional";
8
9     public static final class Colunas{
10         public static final String ID="id";
11         public static final String NOME="nome";
12         public static final String CBO_DESCRICAO="cbo_descricao";
13         public static final String
14
15     }
16
17
18
19
20 }
21
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\TabelaServico.java

```
1 package br.com.jucemar_dimon.cadesaude;
2
3 /**
4 * Created by Jucemar on 10/03/2016.
5 */
6 public final class TabelaServico {
7     public static final String NOME_TABELA = "servico";
8
9     public static final class Colunas {
10         public static final String ID = "id";
11         public static final String CODIGO = "codigo";
12         public static final String DESCRICAO = "descricao";
13     }
14
15 }
16
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\TabelaServiçoClassificacao.java

```
1 package br.com.jucemar_dimon.cadesaude;
2
3 /**
4 * Created by Jucemar on 10/03/2016.
5 */
6 public final class TabelaServiçoClassificacao {
7     public static final String NOME_TABELA = "servico_classificacao_estabelecimento";
8
9     public static final class Colunas {
10         public static final String ESTABELECIMENTO_CNES = "estabelecimento_cnes";
11         public static final String CLASSIFICACAO_ID = "classificacao_id";
12     }
13
14 }
15
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\TabelaVinculosProfissional.java

```
1 package br.com.jucemar_dimon.cadesaude;
2
3 /**
4 * Created by Jucemar on 10/03/2016.
5 */
6 public final class TabelaVinculosProfissional {
7     public static final String NOME_TABELA = "vinculo_profissional_estabelecimento";
8
9     public static final class Colunas {
10         public static final String ID = "id";
11         public static final String ESTABELECIMENTO_CNES = "estabelecimento_cnes";
12     }
13
14 }
15
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\TelainicialActivity.java

```
1 package br.com.jucemar_dimon.cadesaude;
2 /**
3 * Created by Jucemar on 10/03/2016.
4 */
5
6 import android.content.Intent;
7 import android.content.pm.PackageInfo;
8 import android.content.pm.PackageManager;
9 import android.os.Bundle;
10 import android.os.Handler;
11 import android.preference.PreferenceManager;
12 import android.support.v7.app.AppCompatActivity;
13 import android.widget.TextView;
14 import java.util.HashSet;
15 import java.util.Set;
16 import java.util.Set;
17
18 public class TelainicialActivity extends AppCompatActivity {
19
20     private static int SPLASH_TIME_OUT = 2000;
21     private Set<String> listaDeMunicipios;
22     private TextView txv_versao;
23
24     @Override
25     protected void onCreate(Bundle savedInstanceState) {
26         super.onCreate(savedInstanceState);
27         setContentView(R.layout.activity_tela_inicial);
28         txv_versao = (TextView) findViewById(R.id.txv_versao_app);
29         PackageInfo pInfo = null;
30
31         try {
32             pInfo = getPackageManager().getPackageInfo(getPackageName(), 0);
33         } catch (PackageManager.NameNotFoundException e) {
34             e.printStackTrace();
35         }
36         String appVersao = pInfo.versionName;
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\_dimon\cadesaude\TelainicialActivity.java

```
36     txv_versao.setText("Versão " + appVersao);
37     new Handler().postDelayed(new Runnable() {
38         /*
39          * Exibindo a tela inicial.
40         */
41         @Override
42         public void run() {
43             int qtd = PreferenceManager.getDefaultSharedPreferences(getApplicationContext())
44                 .getStringSet(getString(R.string.pref_municipios_key), new HashSet<String>()).size();
45             /*
46                // Esse método será executado sempre que o tempo acabar
47                // E inicia a activity principal
48            if (qtd > 0) {
49                Intent i = new Intent(TelaInicialActivity.this, MainActivity.class);
50                startActivity(i);
51            } else {
52                Intent i = new Intent(TelaInicialActivity.this, PrimeirosPassosActivity.class);
53                startActivity(i);
54            }
55            /*
56            // Fecha a tela de apresentação
57            finish();
58        }
59        @Override
60        public void onBackPressed() {
61            System.exit(0);
62        }
63    }
64}
65``
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity\_ajuda.xml

```
1 <android.support.design.widget.CoordinatorLayout xmlns:android="http://schemas.android.com/apk/res/android"
2     xmlns:app="http://schemas.android.com/apk/res-auto"
3         android:layout_width="match_parent"
4             android:layout_height="match_parent">
5
6     <android.support.design.widget.AppBarLayout
7         android:layout_width="match_parent"
8             android:layout_height="wrap_content"
9                 android:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar">
10
11         <android.support.v7.widget.Toolbar
12             android:id="@+id/toolbar_actvtaAjuda"
13                 android:layout_width="match_parent"
14                     android:layout_height="?attr/actionBarSize"
15                     android:background="?attr/colorPrimary"
16                     app:layout_scrollFlags="scrollEnterAlways"
17                     app:popupTheme="@style/ThemeOverlay.AppCompat.Light" />
18     </android.support.design.widget.AppBarLayout>
19
20     <android.support.v4.widget.NestedScrollView
21             android:id="@+id/nestedScrollView"
22                 android:layout_width="match_parent"
23                     android:layout_height="match_parent"
24                     android:scrollbars="vertical"
25                     android:layout_behavior="@string/appbar_scrolling_view_behavior">
26
27         <LinearLayout
28             android:layout_width="match_parent"
29                 android:layout_height="match_parent"
30                     android:layout_margin="16dp"
31                     android:orientation="vertical">
32
33             <ImageView
34                 android:layout_width="wrap_content"
35                     android:layout_height="wrap_content">
36             </LinearLayout>
37     </NestedScrollView>
38 
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity\_ajuda.xml

```
36     android:layout_gravity="center"
37     android:layout_margin="16dp"
38     android:src="@drawable/ic_help_circle_outline_grey600_48dp" />
39
40     <TextView
41         android:layout_width="wrap_content"
42         android:layout_height="wrap_content"
43         android:layout_margin="16dp"
44         android:text="@string/cabecalho_ajuda" />
45
46     <Button
47         android:id="@+id/btn_tipos_estabelecimentos"
48         android:layout_width="match_parent"
49         android:layout_height="wrap_content"
50         android:layout_margin="16dp"
51         android:text="@string/label_btn_tipos_estabelecimentos" />
52
53     <Button
54         android:id="@+id/btn_tipos_servicos"
55         android:layout_width="match_parent"
56         android:layout_height="wrap_content"
57         android:layout_margin="16dp"
58         android:text="@string/label_btn_tipos_servicos" />
59     </LinearLayout>
60 </android.support.v4.widget.NestedScrollView>
61 </android.support.design.widget.CoordinatorLayout>
62
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity\_ajuda\_tipos\_de\_servicos.xml

```
1 <android.support.design.widget.CoordinatorLayout xmlns:android="http://schemas.android.com/apk/res/android"
2     xmlns:app="http://schemas.android.com/apk/res-auto"
3         android:layout_width="match_parent"
4             android:layout_height="match_parent">
5
6     <android.support.design.widget.AppBarLayout
7         android:layout_width="match_parent"
8             android:layout_height="wrap_content"
9                 android:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar">
10
11         <android.support.v7.widget.Toolbar
12             android:id="@+id/toolbar_actvt_ajuda_tipos_servicos"
13                 android:layout_width="match_parent"
14                     android:layout_height="?attr/actionBarSize"
15                     android:background="?attr/colorPrimary"
16                     app:layout_scrollFlags="scrollEnterAlways"
17                     app:popupTheme="@style/ThemeOverlay.AppCompat.Light" />
18     </android.support.design.widget.AppBarLayout>
19
20     <android.support.v4.widget.NestedScrollView
21             android:id="@+id/nestedScrollView"
22                 android:layout_width="match_parent"
23                     android:layout_height="match_parent"
24                     android:scrollbars="vertical"
25                     android:layout_behavior="@string/appbar_scrolling_view_behavior">
26
27         <TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
28             android:layout_width="match_parent"
29                 android:layout_height="match_parent"
30                     android:layout_margin="16dp"
31                     android:paddingBottom="32dp"
32                     android:shrinkColumns="1">
33
34             <TableRow
```

```
File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity_ajuda_tipos_de_servicos.xml

36   android:layout_width="match_parent"
37   android:layout_height="wrap_content"
38   android:background="@color/primary_light"
39   android:gravity="center">
40
41   <TextView
42     android:layout_margin="8dp"
43     android:layout_weight="2"
44     android:width="0dp"
45     android:singleLine="false"
46     android:text="Serviço especializado"
47     android:textStyle="bold" />
48
49   <TextView
50     android:layout_margin="8dp"
51     android:layout_weight="3"
52     android:width="0dp"
53     android:text="Detalhes do serviço"
54     android:textStyle="bold" />
55
56   <TableRow
57     android:layout_width="match_parent"
58     android:layout_height="wrap_content"
59     android:background="#e0e0e0"
60     android:gravity="center">
61
62   <TextView
63     android:layout_margin="8dp"
64     android:layout_weight="2"
65     android:width="0dp"
66     android:singleLine="false"
67     android:text="@string/101cl" />
68
69   <TextView
```

```
File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity_ajuda_tipos_de_servicos.xml

71
72    android:layout_margin="8dp"
73    android:layout_weight="3"
74    android:width="0dp"
75    android:text="@string/101c2" />
76
77    <TableRow
78        android:layout_width="match_parent"
79        android:layout_height="wrap_content"
80        android:background="@color/primary_light"
81        android:gravity="center">
82
83        <TextView
84            android:layout_margin="8dp"
85            android:layout_weight="2"
86            android:width="0dp"
87            android:text="@string/102c1" />
88
89        <TextView
90            android:layout_margin="8dp"
91            android:layout_weight="3"
92            android:width="0dp"
93            android:text="@string/102c2" />
94
95        <TableRow
96            android:layout_width="match_parent"
97            android:layout_height="wrap_content"
98            android:background="#e0e0e0"
99            android:gravity="center">
100
101        <TextView
102            android:layout_margin="8dp"
103            android:layout_weight="2"
104            android:width="0dp"
```

```
File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity_ajuda_tipos_de_servicos.xml

106     android:text="@string/103c1" />
107
108     <TextView
109         android:layout_margin="8dp"
110         android:layout_weight="3"
111         android:width="0dp"
112         android:text="@string/103c2" />
113
114     <TableRow
115         android:layout_width="match_parent"
116         android:layout_height="wrap_content"
117         android:background="@color/primary_light"
118         android:gravity="center">
119
120         <TextView
121             android:layout_margin="8dp"
122             android:layout_weight="2"
123             android:width="0dp"
124             android:text="@string/104c1" />
125
126         <TextView
127             android:layout_margin="8dp"
128             android:layout_weight="3"
129             android:width="0dp"
130             android:text="@string/104c2" />
131
132     </TableRow>
133
134     <TableRow
135         android:layout_width="match_parent"
136         android:layout_height="wrap_content"
137         android:background="#e0e0e0"
138         android:gravity="center">
139
140     </TableRow>
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity\_ajuda\_tipos\_de\_servicos.xml

```
141     android:layout_margin="8dp"
142     android:layout_weight="2"
143     android:width="0dp"
144     android:text="@string/105c1" />
145
146     <TextView
147         android:layout_margin="8dp"
148         android:layout_weight="3"
149         android:width="0dp"
150         android:text="@string/105c2" />
151     </TableRow>
152
153     <TableRow
154         android:layout_width="match_parent"
155         android:layout_height="wrap_content"
156         android:background="@color/primary_light"
157         android:gravity="center">
158
159         <TextView
160             android:layout_margin="8dp"
161             android:layout_weight="2"
162             android:width="0dp"
163             android:text="@string/106c1" />
164
165         <TextView
166             android:layout_margin="8dp"
167             android:layout_weight="3"
168             android:width="0dp"
169             android:text="@string/106c2" />
170     </TableRow>
171
172     <TableRow
173         android:layout_width="match_parent"
174         android:layout_height="wrap_content"
175         android:background="#e0e0e0" />
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity\_ajuda\_tipos\_de\_servicos.xml

```
176      android:gravity="center">
177
178      <TextView
179          android:layout_margin="8dp"
180          android:layout_weight="2"
181          android:width="0dp"
182          android:text="@string/I07c1" />
183
184      <TextView
185          android:layout_margin="8dp"
186          android:layout_weight="3"
187          android:width="0dp"
188          android:text="@string/I07c2" />
189      </TableRow>
190
191      <TableRow
192          android:layout_width="match_parent"
193          android:layout_height="wrap_content"
194          android:background="@color/primary_light"
195          android:gravity="center">
196
197          <TextView
198              android:layout_margin="8dp"
199              android:layout_weight="2"
200              android:width="0dp"
201              android:text="@string/I08c1" />
202
203          <TextView
204              android:layout_margin="8dp"
205              android:layout_weight="3"
206              android:width="0dp"
207              android:text="@string/I08c2" />
208      </TableRow>
209
... 
```

```
File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity_ajuda_tipos_de_servicos.xml

211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
```

```
File - E:\Android\Projetos\CadESaudé\app\src\main\res\layout\activity_ajuda_tipos_de_servicos.xml

246      </TableRow>
247
248      <TableRow
249          android:layout_width="match_parent"
250          android:layout_height="wrap_content"
251          android:background="#e0e0e0"
252          android:gravity="center">
253
254          <TextView
255              android:layout_margin="8dp"
256              android:layout_weight="2"
257              android:width="0dp"
258              android:text="@string/l11c1" />
259
260          <TextView
261              android:layout_margin="8dp"
262              android:layout_weight="3"
263              android:width="0dp"
264              android:text="@string/l11c2" />
265
266      </TableRow
267
268      <TableRow
269          android:layout_width="match_parent"
270          android:layout_height="wrap_content"
271          android:background="@color/primary_light"
272          android:gravity="center">
273
274          <TextView
275              android:layout_margin="8dp"
276              android:layout_weight="2"
277              android:width="0dp"
278              android:text="@string/l12c1" />
279
280      <TextView
281          android:layout_margin="8dp"
282          android:layout_weight="3"
283          android:width="0dp"
284          android:text="@string/l12c2" />
285
286  </Table>
```

```
File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity_ajuda_tipos_de_servicos.xml

281      android:layout_weight="3"
282      android:width="0dp"
283      android:text="@string/l12c2" />
284
285    <TableRow
286      android:layout_width="match_parent"
287      android:layout_height="wrap_content"
288      android:background="#e0e0e0"
289      android:gravity="center">
290
291      <TextView
292        android:layout_margin="8dp"
293        android:layout_weight="2"
294        android:width="0dp"
295        android:text="@string/l13c1" />
296
297      <TextView
298        android:layout_margin="8dp"
299        android:layout_weight="3"
300        android:width="0dp"
301        android:text="@string/l13c2" />
302
303    </TableRow>
304
305    <TableRow
306      android:layout_width="match_parent"
307      android:layout_height="wrap_content"
308      android:background="@color/primary_light"
309      android:gravity="center">
310
311      <TextView
312        android:layout_margin="8dp"
313        android:layout_weight="2"
314        android:width="0dp"
315        android:text="@string/l14c1" />
316
```

File - E:\Android\Projetos\CadESaude\app\src\main\res\layout\activity\_ajuda\_tipos\_de\_servicos.xml

```
316
317   <TextView
318     android:layout_margin="8dp"
319     android:layout_weight="3"
320     android:width="0dp"
321     android:text="@string/l14c2" />
322
323   <TableRow
324     android:layout_width="match_parent"
325     android:layout_height="wrap_content"
326     android:background="#e0e0e0"
327     android:gravity="center">
328
329   <TextView
330     android:layout_margin="8dp"
331     android:layout_weight="2"
332     android:width="0dp"
333     android:text="@string/l15c1" />
334
335   <TextView
336     android:layout_margin="8dp"
337     android:layout_weight="3"
338     android:width="0dp"
339     android:text="@string/l15c2" />
340
341   <TableRow
342     android:layout_width="match_parent"
343     android:layout_height="wrap_content"
344     android:background="@color/primary_light"
345     android:gravity="center">
346
347   <TextView
348     android:layout_margin="8dp"
349     android:layout_weight="3"
350     android:width="0dp"
351     android:text="@string/l15c3" />
```

```
File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity_ajuda_tipos_de_servicos.xml

351      android:layout_weight="2"
352          android:width="0dp"
353          android:text="@string/l16c1" />
354
355      <TextView
356          android:layout_margin="8dp"
357          android:layout_weight="3"
358          android:width="0dp"
359          android:text="@string/l16c2" />
360
361      <TableRow
362          android:layout_width="match_parent"
363          android:layout_height="wrap_content"
364          android:background="#e0e0e0"
365          android:gravity="center">
366
367          <TextView
368              android:layout_margin="8dp"
369              android:layout_weight="2"
370              android:width="0dp"
371              android:text="@string/l17c1" />
372
373          <TextView
374              android:layout_margin="8dp"
375              android:layout_weight="3"
376              android:width="0dp"
377              android:text="@string/l17c2" />
378
379      <TableRow
380          android:layout_width="match_parent"
381          android:layout_height="wrap_content"
382          android:background="@color/primary_light"
383          android:gravity="center">
384
385
```

```
386
387    <TextView
388        android:layout_margin="8dp"
389        android:layout_weight="2"
390        android:width="0dp"
391        android:text="@string/118c1" />
392
393    <TextView
394        android:layout_margin="8dp"
395        android:layout_weight="3"
396        android:width="0dp"
397        android:text="@string/118c2" />
398
399    <TableRow
400        android:layout_width="match_parent"
401        android:layout_height="wrap_content"
402        android:background="#e0e0e0"
403        android:gravity="center">
404
405        <TextView
406            android:layout_margin="8dp"
407            android:layout_weight="2"
408            android:width="0dp"
409            android:text="@string/119c1" />
410
411        <TextView
412            android:layout_margin="8dp"
413            android:layout_weight="3"
414            android:width="0dp"
415            android:text="@string/119c2" />
416
417    <TableRow
418        android:layout_width="match_parent"
419        ...
```

```
File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity_ajuda_tipos_de_servicos.xml

421           android:layout_height="wrap_content"
422           android:background="@color/primary_light"
423           android:gravity="center">
424
425             <TextView
426               android:layout_margin="8dp"
427               android:layout_weight="2"
428               android:width="0dp"
429               android:text="@string/l120c1" />
430
431             <TextView
432               android:layout_margin="8dp"
433               android:layout_weight="3"
434               android:width="0dp"
435               android:text="@string/l120c2" />
436           </TableRow>
437
438           <TableRow
439             android:layout_width="match_parent"
440             android:layout_height="wrap_content"
441             android:background="#e0e0e0"
442             android:gravity="center">
443
444             <TextView
445               android:layout_margin="8dp"
446               android:layout_weight="2"
447               android:width="0dp"
448               android:text="@string/l121c1" />
449
450             <TextView
451               android:layout_margin="8dp"
452               android:layout_weight="3"
453               android:width="0dp"
454               android:text="@string/l121c2" />
455           </TableRow>
```

```
File - E:\Android\Projetos\CadESAude\app\src\main\res\layout\activity_ajuda_tipos_de_servicos.xml

456 <TableRow
457     android:layout_width="match_parent"
458     android:layout_height="wrap_content"
459     android:background="@color/primary_light"
460     android:gravity="center">
461
462     <TextView
463         android:layout_margin="8dp"
464         android:layout_weight="2"
465         android:width="0dp"
466         android:text="@string/122c1" />
467
468     <TextView
469         android:layout_margin="8dp"
470         android:layout_weight="3"
471         android:width="0dp"
472         android:text="@string/122c2" />
473     </TableRow>
474
475     <TableRow
476         android:layout_width="match_parent"
477         android:layout_height="wrap_content"
478         android:background="#e0e0e0"
479         android:gravity="center">
480
481     <TextView
482         android:layout_margin="8dp"
483         android:layout_weight="2"
484         android:width="0dp"
485         android:text="@string/123c1" />
486
487     <TextView
488         android:layout_margin="8dp"
489         android:layout_weight="3"
490         android:width="0dp"
491         android:text="@string/123c2" />
492
493     <TextView
494         android:layout_margin="8dp"
495         android:layout_weight="2"
496         android:width="0dp"
497         android:text="@string/123c3" />
498
499     <TextView
500         android:layout_margin="8dp"
501         android:layout_weight="3"
502         android:width="0dp"
503         android:text="@string/123c4" />
504
505     <TextView
506         android:layout_margin="8dp"
507         android:layout_weight="2"
508         android:width="0dp"
509         android:text="@string/123c5" />
510
511     <TextView
512         android:layout_margin="8dp"
513         android:layout_weight="3"
514         android:width="0dp"
515         android:text="@string/123c6" />
516
517     <TextView
518         android:layout_margin="8dp"
519         android:layout_weight="2"
520         android:width="0dp"
521         android:text="@string/123c7" />
522
523     <TextView
524         android:layout_margin="8dp"
525         android:layout_weight="3"
526         android:width="0dp"
527         android:text="@string/123c8" />
528
529     <TextView
530         android:layout_margin="8dp"
531         android:layout_weight="2"
532         android:width="0dp"
533         android:text="@string/123c9" />
534
535     <TextView
536         android:layout_margin="8dp"
537         android:layout_weight="3"
538         android:width="0dp"
539         android:text="@string/123c10" />
540
541     <TextView
542         android:layout_margin="8dp"
543         android:layout_weight="2"
544         android:width="0dp"
545         android:text="@string/123c11" />
546
547     <TextView
548         android:layout_margin="8dp"
549         android:layout_weight="3"
550         android:width="0dp"
551         android:text="@string/123c12" />
552
553     <TextView
554         android:layout_margin="8dp"
555         android:layout_weight="2"
556         android:width="0dp"
557         android:text="@string/123c13" />
558
559     <TextView
560         android:layout_margin="8dp"
561         android:layout_weight="3"
562         android:width="0dp"
563         android:text="@string/123c14" />
564
565     <TextView
566         android:layout_margin="8dp"
567         android:layout_weight="2"
568         android:width="0dp"
569         android:text="@string/123c15" />
570
571     <TextView
572         android:layout_margin="8dp"
573         android:layout_weight="3"
574         android:width="0dp"
575         android:text="@string/123c16" />
576
577     <TextView
578         android:layout_margin="8dp"
579         android:layout_weight="2"
580         android:width="0dp"
581         android:text="@string/123c17" />
582
583     <TextView
584         android:layout_margin="8dp"
585         android:layout_weight="3"
586         android:width="0dp"
587         android:text="@string/123c18" />
588
589     <TextView
590         android:layout_margin="8dp"
591         android:layout_weight="2"
592         android:width="0dp"
593         android:text="@string/123c19" />
594
595     <TextView
596         android:layout_margin="8dp"
597         android:layout_weight="3"
598         android:width="0dp"
599         android:text="@string/123c20" />
599 ...
```

```
File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity_ajuda_tipos_de_servicos.xml

491      android:width="0dp"
492      android:text="@string/123c2" />
493  </TableRow>
494
495  <TableRow
496      android:layout_width="match_parent"
497      android:layout_height="wrap_content"
498      android:background="@color/primary_light"
499      android:gravity="center">
500
501      <TextView
502          android:layout_margin="8dp"
503          android:layout_weight="2"
504          android:width="0dp"
505          android:text="@string/124c1" />
506
507      <TextView
508          android:layout_margin="8dp"
509          android:layout_weight="3"
510          android:width="0dp"
511          android:text="@string/124c2" />
512  </TableRow>
513
514  <TableRow
515      android:layout_width="match_parent"
516      android:layout_height="wrap_content"
517      android:background="#e0e0e0"
518      android:gravity="center">
519
520      <TextView
521          android:layout_margin="8dp"
522          android:layout_weight="2"
523          android:width="0dp"
524          android:text="@string/125c1" />
525
```

File - E:\Android\Projetos\CadESaude\app\src\main\res\layout\activity\_ajuda\_tipos\_de\_servicos.xml

```
526      <TextView  
527          android:layout_margin="8dp"  
528          android:layout_weight="3"  
529          android:width="0dp"  
530          android:text="@string/125c2" />  
531      </TableRow>  
532  
533      <TableRow  
534          android:layout_width="match_parent"  
535          android:layout_height="wrap_content"  
536          android:background="@color/primary_light"  
537          android:gravity="center">  
538  
539          <TextView  
540              android:layout_margin="8dp"  
541              android:layout_weight="2"  
542              android:width="0dp"  
543              android:text="@string/126c1" />  
544  
545          <TextView  
546              android:layout_margin="8dp"  
547              android:layout_weight="3"  
548              android:width="0dp"  
549              android:text="@string/126c2" />  
550      </TableRow>  
551  
552      <TableRow  
553          android:layout_width="match_parent"  
554          android:layout_height="wrap_content"  
555          android:background="#e0e0e0"  
556          android:gravity="center">  
557  
558          <TextView  
559              android:layout_margin="8dp"  
560              android:layout_weight="2"  
561          </TextView>
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity\_ajuda\_tipos\_de\_servicos.xml

```
561     android:width="0dp"
562     android:singleLine="false"
563     android:text="@string/l127c1" />
564
565     <TextView
566         android:layout_margin="8dp"
567         android:layout_weight="3"
568         android:width="0dp"
569         android:singleLine="false"
570         android:text="@string/l127c2" />
571     </TableRow>
572
573     <TableRow
574         android:layout_width="match_parent"
575         android:layout_height="wrap_content"
576         android:background="@color/primary_light"
577         android:gravity="center">
578
579         <TextView
580             android:layout_margin="8dp"
581             android:layout_weight="2"
582             android:width="0dp"
583             android:text="@string/l128c1" />
584
585         <TextView
586             android:layout_margin="8dp"
587             android:layout_weight="3"
588             android:width="0dp"
589             android:text="@string/l128c2" />
590     </TableRow>
591
592     <TableRow
593         android:layout_width="match_parent"
594         android:layout_height="wrap_content"
595         android:background="#e0e0e0" />
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity\_ajuda\_tipos\_de\_servicos.xml

```
596         android:gravity="center">
597
598     <TextView
599         android:layout_margin="8dp"
600         android:layout_weight="2"
601         android:width="0dp"
602         android:text="@string/l129c1" />
603
604     <TextView
605         android:layout_margin="8dp"
606         android:layout_weight="3"
607         android:width="0dp"
608         android:text="@string/l129c2" />
609     </TableRow>
610
611     <TableRow
612         android:layout_width="match_parent"
613         android:layout_height="wrap_content"
614         android:background="@color/primary_light"
615         android:gravity="center">
616
617         <TextView
618             android:layout_margin="8dp"
619             android:layout_weight="2"
620             android:width="0dp"
621             android:text="@string/l130c1" />
622
623         <TextView
624             android:layout_margin="8dp"
625             android:layout_weight="3"
626             android:width="0dp"
627             android:text="@string/l130c2" />
628     </TableRow>
629
630     <TableRow
```

```
File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity_ajuda_tipos_de_servicos.xml

631   android:layout_width="match_parent"
632   android:layout_height="wrap_content"
633   android:background="#e0e0e0"
634   android:gravity="center">
635
636   <TextView
637     android:layout_margin="8dp"
638     android:layout_weight="2"
639     android:width="0dp"
640     android:text="@string/l31c1" />
641
642   <TextView
643     android:layout_margin="8dp"
644     android:layout_weight="3"
645     android:width="0dp"
646     android:text="@string/l31c2" />
647   </TableRow>
648
649   <TableRow
650     android:layout_width="match_parent"
651     android:layout_height="wrap_content"
652     android:background="@color/primary_light"
653     android:gravity="center">
654
655   <TextView
656     android:layout_margin="8dp"
657     android:layout_weight="2"
658     android:width="0dp"
659     android:text="@string/l32c1" />
660
661   <TextView
662     android:layout_margin="8dp"
663     android:layout_weight="3"
664     android:width="0dp"
665     android:text="@string/l32c2" />
666
```

```
File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity_ajuda_tipos_de_servicos.xml

666
667 </TableRow>
668
669 <TableRow
670     android:layout_width="match_parent"
671     android:layout_height="wrap_content"
672     android:background="#e0e0e0"
673     android:gravity="center">
674
675     <TextView
676         android:layout_margin="8dp"
677         android:layout_weight="2"
678         android:width="0dp"
679         android:text="@string/133c1" />
680
681     <TextView
682         android:layout_margin="8dp"
683         android:layout_weight="3"
684         android:width="0dp"
685         android:text="@string/133c2" />
686 </TableRow>
687
688 <TableRow
689     android:layout_width="match_parent"
690     android:layout_height="wrap_content"
691     android:background="@color/primary_Light"
692     android:gravity="center">
693
694     <TextView
695         android:layout_margin="8dp"
696         android:layout_weight="2"
697         android:width="0dp"
698         android:text="@string/134c1" />
699
700     <TextView
701         android:layout_margin="8dp"
702         android:layout_weight="3"
703         android:width="0dp"
704         android:text="@string/134c2" />
705 </TableRow>
```

```
File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity_ajuda_tipos_de_servicos.xml

701      android:layout_weight="3"
702      android:width="0dp"
703      android:text="@string/134c2" />
704
705    <TableRow
706      android:layout_width="match_parent"
707      android:layout_height="wrap_content"
708      android:background="#e0e0e0"
709      android:gravity="center">
710
711      <TextView
712        android:layout_margin="8dp"
713        android:layout_weight="2"
714        android:width="0dp"
715        android:text="@string/135c1" />
716
717      <TextView
718        android:layout_margin="8dp"
719        android:layout_weight="3"
720        android:width="0dp"
721        android:text="@string/135c2" />
722
723    </TableRow>
724
725    <TableRow
726      android:layout_width="match_parent"
727      android:layout_height="wrap_content"
728      android:background="@color/primary_light"
729      android:gravity="center">
730
731      <TextView
732        android:layout_margin="8dp"
733        android:layout_weight="2"
734        android:width="0dp"
735        android:text="@string/136c1" />
736
```

```
736      <TextView  
737          android:layout_margin="8dp"  
738          android:layout_weight="3"  
739          android:width="0dp"  
740          android:text="@string/136c2" />  
741      </TableRow>  
742  
743      <TableRow  
744          android:layout_width="match_parent"  
745          android:layout_height="wrap_content"  
746          android:background="#e0e0e0"  
747          android:gravity="center">  
748  
749          <TextView  
750              android:layout_margin="8dp"  
751              android:layout_weight="2"  
752              android:width="0dp"  
753              android:text="@string/137c1" />  
754  
755          <TextView  
756              android:layout_margin="8dp"  
757              android:layout_weight="3"  
758              android:width="0dp"  
759              android:text="@string/137c2" />  
760      </TableRow>  
761  
762      <TableRow  
763          android:layout_width="match_parent"  
764          android:layout_height="wrap_content"  
765          android:background="@color/primary_light"  
766          android:gravity="center">  
767  
768          <TextView  
769              android:layout_margin="8dp"  
770          </TextView>
```

```
File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity_ajuda_tipos_de_servicos.xml

771
772    android:layout_weight="2"
773    android:width="0dp"
774    android:text="@string/138c1" />
775
776    <TextView
777        android:layout_margin="8dp"
778        android:layout_weight="3"
779        android:width="0dp"
780        android:text="@string/138c2" />
781
782    <TableRow
783        android:layout_width="match_parent"
784        android:layout_height="wrap_content"
785        android:background="#e0e0e0"
786        android:gravity="center">
787
788        <TextView
789            android:layout_margin="8dp"
790            android:layout_weight="2"
791            android:width="0dp"
792            android:text="@string/139c1" />
793
794        <TextView
795            android:layout_margin="8dp"
796            android:layout_weight="3"
797            android:width="0dp"
798            android:text="@string/139c2" />
799
800    <TableRow
801        android:layout_width="match_parent"
802        android:layout_height="wrap_content"
803        android:background="@color/primary_light"
804        android:gravity="center">
     ...
```

```
File - E:\Android\Projetos\CadESaude\app\src\main\res\layout\activity_ajuda_tipos_de_servicos.xml

806
807 <TextView
808     android:layout_margin="8dp"
809     android:layout_weight="2"
810     android:width="0dp"
811     android:text="@string/140c1" />
812
813 <TextView
814     android:layout_margin="8dp"
815     android:layout_weight="3"
816     android:width="0dp"
817     android:text="@string/140c2" />
818 </TableRow>
819
820 <TableRow
821     android:layout_width="match_parent"
822     android:layout_height="wrap_content"
823     android:background="#e0e0e0"
824     android:gravity="center">
825
826 <TextView
827     android:layout_margin="8dp"
828     android:layout_weight="2"
829     android:width="0dp"
830     android:text="@string/141c1" />
831
832 <TextView
833     android:layout_margin="8dp"
834     android:layout_weight="3"
835     android:width="0dp"
836     android:text="@string/141c2" />
837 </TableRow>
838
839     android:layout_width="match_parent"
```

```
File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity_ajuda_tipos_de_servicos.xml

841     android:layout_height="wrap_content"
842         android:background="@color/primary_light"
843         android:gravity="center">
844
845     <TextView
846         android:layout_margin="8dp"
847         android:layout_weight="2"
848         android:width="0dp"
849         android:text="@string/l42c1" />
850
851     <TextView
852         android:layout_margin="8dp"
853         android:layout_weight="3"
854         android:width="0dp"
855         android:text="@string/l42c2" />
856     </TableRow>
857
858     <TableRow
859         android:layout_width="match_parent"
860         android:layout_height="wrap_content"
861         android:background="#e0e0e0"
862         android:gravity="center">
863
864     <TextView
865         android:layout_margin="8dp"
866         android:layout_weight="2"
867         android:width="0dp"
868         android:text="@string/l43c1" />
869
870     <TextView
871         android:layout_margin="8dp"
872         android:layout_weight="3"
873         android:width="0dp"
874         android:text="@string/l43c2" />
875     </TableRow>
--
```

File - E:\Android\Projetos\CadESaude\app\src\main\res\layout\activity\_ajuda\_tipos\_de\_servicos.xml

```
876
877    <TableRow
878        android:layout_width="match_parent"
879        android:layout_height="wrap_content"
880        android:background="@color/primary_light"
881        android:gravity="center">
882
883        <TextView
884            android:layout_margin="8dp"
885            android:layout_weight="2"
886            android:width="0dp"
887            android:text="@string/l44c1" />
888
889        <TextView
890            android:layout_margin="8dp"
891            android:layout_weight="3"
892            android:width="0dp"
893            android:text="@string/l44c2" />
894    </TableRow>
895
896    <TableRow
897        android:layout_width="match_parent"
898        android:layout_height="wrap_content"
899        android:background="#e0e0e0"
900        android:gravity="center">
901
902        <TextView
903            android:layout_margin="8dp"
904            android:layout_weight="2"
905            android:width="0dp"
906            android:text="@string/l45c1" />
907
908        <TextView
909            android:layout_margin="8dp"
910            android:layout_weight="3"
911            android:width="0dp"
912            android:text="@string/l45c2" />
913
914    </TableRow>
915
916
```

```
File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity_ajuda_tipos_de_servicos.xml

911     android:width="0dp"
912     android:text="@string/l45c2" />
913 </TableRow>
914
915     <TableRow
916         android:layout_width="match_parent"
917         android:layout_height="wrap_content"
918         android:background="@color/primary_light"
919         android:gravity="center">
920
921         <TextView
922             android:layout_margin="8dp"
923             android:layout_weight="2"
924             android:width="0dp"
925             android:text="@string/l46c1" />
926
927         <TextView
928             android:layout_margin="8dp"
929             android:layout_weight="3"
930             android:width="0dp"
931             android:text="@string/l46c2" />
932     </TableRow>
933
934     <TableRow
935         android:layout_width="match_parent"
936         android:layout_height="wrap_content"
937         android:background="#e0e0e0"
938         android:gravity="center">
939
940         <TextView
941             android:layout_margin="8dp"
942             android:layout_weight="2"
943             android:width="0dp"
944             android:text="@string/l47c1" />
945
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity\_ajuda\_tipos\_de\_servicos.xml

```
946      <TextView  
947          android:layout_margin="8dp"  
948          android:layout_weight="3"  
949          android:width="0dp"  
950          android:text="@string/l47c2" />  
951      </TableRow>  
952      <TableRow  
953          android:layout_width="match_parent"  
954          android:layout_height="wrap_content"  
955          android:background="@color/primary_light"  
956          android:gravity="center">  
957          <TextView  
958              android:layout_margin="8dp"  
959              android:layout_weight="2"  
960              android:width="0dp"  
961              android:text="@string/l48c1" />  
962          <TextView  
963              android:layout_margin="8dp"  
964              android:layout_weight="3"  
965              android:width="0dp"  
966              android:text="@string/l48c2" />  
967          </TableRow>  
968      <TableRow  
969          android:layout_width="match_parent"  
970          android:layout_height="wrap_content"  
971          android:background="#e0e0e0"  
972          android:gravity="center">  
973          <TextView  
974              android:layout_margin="8dp"  
975              android:layout_weight="2"  
976          </TextView  
977          <TextView  
978              android:layout_margin="8dp"  
979              android:layout_weight="2"  
980          </TextView>
```

```
File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity_ajuda_tipos_de_servicos.xml

981     android:width="0dp"
982     android:text="@string/149c1" />
983
984     <TextView
985         android:layout_margin="8dp"
986         android:layout_weight="3"
987         android:width="0dp"
988         android:text="@string/149c2" />
989     </TableRow>
990
991     <TableRow
992         android:layout_width="match_parent"
993         android:layout_height="wrap_content"
994         android:background="@color/primary_light"
995         android:gravity="center">
996
997         <TextView
998             android:layout_margin="8dp"
999             android:layout_weight="2"
1000            android:width="0dp"
1001            android:text="@string/150c1" />
1002
1003         <TextView
1004             android:layout_margin="8dp"
1005             android:layout_weight="3"
1006             android:width="0dp"
1007             android:text="@string/150c2" />
1008     </TableRow>
1009
1010     <TableRow
1011         android:layout_width="match_parent"
1012         android:layout_height="wrap_content"
1013         android:background="#e0e0e0"
1014         android:gravity="center">
1015
```

File - E:\Android\Projetos\CadESaude\app\src\main\res\layout\activity\_ajuda\_tipos\_de\_servicos.xml

```
1016 <TextView  
1017     android:layout_margin="8dp"  
1018     android:layout_weight="2"  
1019     android:width="0dp"  
1020     android:text="@string/151c1" />  
1021  
1022     <TextView  
1023         android:layout_margin="8dp"  
1024         android:layout_weight="3"  
1025         android:width="0dp"  
1026         android:text="@string/151c2" />  
1027 </TableRow>  
1028  
1029     <TableRow  
1030         android:layout_width="match_parent"  
1031         android:layout_height="wrap_content"  
1032         android:background="@color/primary_light"  
1033         android:gravity="center">  
1034         <TextView  
1035             android:layout_margin="8dp"  
1036             android:layout_weight="2"  
1037             android:width="0dp"  
1038             android:text="@string/152c1" />  
1039 </TableRow>  
1040  
1041     <TableRow  
1042         android:layout_width="match_parent"  
1043         android:layout_height="wrap_content"  
1044         android:text="@string/152c2" />  
1045 </TableRow>  
1046  
1047     <TableRow  
1048         android:layout_width="match_parent"  
1049         android:layout_height="wrap_content"  
1050         ...
```

```
File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity_ajuda_tipos_de_servicos.xml

1051
1052     android:background="#e0e0e0"
1053     android:gravity="center">
1054
1055         <TextView
1056             android:layout_margin="8dp"
1057             android:layout_weight="2"
1058             android:width="0dp"
1059             android:text="@string/153c1" />
1060
1061         <TextView
1062             android:layout_margin="8dp"
1063             android:layout_weight="3"
1064             android:width="0dp"
1065             android:text="@string/153c2" />
1066
1067         <TableRow
1068             android:layout_width="match_parent"
1069             android:layout_height="wrap_content"
1070             android:background="@color/primary_light"
1071             android:gravity="center">
1072
1073         <TextView
1074             android:layout_margin="8dp"
1075             android:layout_weight="2"
1076             android:width="0dp"
1077             android:text="@string/154c1" />
1078
1079         <TextView
1080             android:layout_margin="8dp"
1081             android:layout_weight="3"
1082             android:width="0dp"
1083             android:text="@string/154c2" />
1084
1085     </Table>
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity\_ajuda\_tipos\_de\_servicos.xml

```
1086 <TableRow
1087     android:layout_width="match_parent"
1088     android:layout_height="wrap_content"
1089     android:background="#e0e0e0"
1090     android:gravity="center">
1091
1092     <TextView
1093         android:layout_margin="8dp"
1094         android:layout_weight="2"
1095         android:width="0dp"
1096         android:text="@string/155c1" />
1097
1098     <TextView
1099         android:layout_margin="8dp"
1100         android:layout_weight="3"
1101         android:width="0dp"
1102         android:text="@string/155c2" />
1103 </TableRow>
1104
1105     <TableRow
1106         android:layout_width="match_parent"
1107         android:layout_height="wrap_content"
1108         android:background="@color/primary_light"
1109         android:gravity="center">
1110
1111     <TextView
1112         android:layout_margin="8dp"
1113         android:layout_weight="2"
1114         android:width="0dp"
1115         android:text="@string/156c1" />
1116
1117     <TextView
1118         android:layout_margin="8dp"
1119         android:layout_weight="3"
1120         android:width="0dp"
1121 ...
```

```
File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity_ajuda_tipos_de_servicos.xml

1121     android:text="@string/156c2" />
1122 
1123 </TableRow>
1124 
1125     android:layout_width="match_parent"
1126     android:layout_height="wrap_content"
1127     android:background="#e0e0e0"
1128     android:gravity="center">
1129 
1130 <TextView
1131     android:layout_margin="8dp"
1132     android:layout_weight="2"
1133     android:width="0dp"
1134     android:text="@string/157c1" />
1135 
1136 <TextView
1137     android:layout_margin="8dp"
1138     android:layout_weight="3"
1139     android:width="0dp"
1140     android:text="@string/157c2" />
1141 
1142 </TableRow>
1143 
1144     android:layout_width="match_parent"
1145     android:layout_height="wrap_content"
1146     android:background="@color/primary_light"
1147     android:gravity="center">
1148 
1149 <TextView
1150     android:layout_margin="8dp"
1151     android:layout_weight="2"
1152     android:width="0dp"
1153     android:text="@string/158c1" />
1154 
1155 ...
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity\_ajuda\_tipos\_de\_servicos.xml

```
1156     android:layout_margin="8dp"
1157     android:layout_weight="3"
1158     android:width="0dp"
1159     android:text="@string/158c2" />
1160 </TableRow>
1161
1162 <TableRow
1163     android:layout_width="match_parent"
1164     android:layout_height="wrap_content"
1165     android:background="#e0e0e0"
1166     android:gravity="center">
1167
1168     <TextView
1169         android:layout_margin="8dp"
1170         android:layout_weight="2"
1171         android:width="0dp"
1172         android:text="@string/159c1" />
1173
1174     <TextView
1175         android:layout_margin="8dp"
1176         android:layout_weight="3"
1177         android:width="0dp"
1178         android:text="@string/159c2" />
1179 </TableRow>
1180
1181 <TableRow
1182     android:layout_width="match_parent"
1183     android:layout_height="wrap_content"
1184     android:background="@color/primary_light"
1185     android:gravity="center">
1186
1187     <TextView
1188         android:layout_margin="8dp"
1189         android:layout_weight="2"
1190         android:width="0dp"
1191 ...
```

```
File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity_ajuda_tipos_de_servicos.xml

1191
1192    android:text="@+id/texto"
1193
1194    <TextView
1195        android:layout_margin="8dp"
1196            android:layout_weight="3"
1197            android:width="0dp"
1198            android:text="@string/160c1" />
1199
1200    <TableRow
1201        android:layout_width="match_parent"
1202            android:layout_height="wrap_content"
1203            android:background="#e0e0e0"
1204            android:gravity="center">
1205
1206    <TextView
1207        android:layout_margin="8dp"
1208            android:layout_weight="2"
1209            android:width="0dp"
1210            android:text="@string/161c1" />
1211
1212    <TextView
1213        android:layout_margin="8dp"
1214            android:layout_weight="3"
1215            android:width="0dp"
1216            android:text="@string/161c2" />
1217
1218    <TableRow
1219        android:layout_width="match_parent"
1220            android:layout_height="wrap_content"
1221            android:background="@color/primary_light"
1222            android:gravity="center">
1223
1224    ...
```

```
File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity_ajuda_tipos_de_servicos.xml

1226
1227     android:layout_margin="8dp"
1228     android:layout_weight="2"
1229     android:width="0dp"
1230     android:text="@string/162c1" />
1231
1232     <TextView
1233         android:layout_margin="8dp"
1234         android:layout_weight="3"
1235         android:width="0dp"
1236         android:text="@string/162c2" />
1237     </TableRow>
1238
1239     <TableRow
1240         android:layout_width="match_parent"
1241         android:layout_height="wrap_content"
1242         android:background="#e0e0e0"
1243         android:gravity="center">
1244
1245         <TextView
1246             android:layout_margin="8dp"
1247             android:layout_weight="2"
1248             android:width="0dp"
1249             android:text="@string/163c1" />
1250
1251     <TextView
1252         android:layout_margin="8dp"
1253         android:layout_weight="3"
1254         android:width="0dp"
1255     </TableRow>
1256
1257     <TableRow
1258         android:layout_width="match_parent"
1259         android:layout_height="wrap_content"
1260         android:background="@color/primary_light"
1261     </TableRow>
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity\_ajuda\_tipos\_de\_servicos.xml

```
1261     android:gravity="center">
1262
1263     <TextView
1264         android:layout_margin="8dp"
1265         android:layout_weight="2"
1266         android:width="0dp"
1267         android:text="@string/164c1" />
1268
1269     <TextView
1270         android:layout_margin="8dp"
1271         android:layout_weight="3"
1272         android:width="0dp"
1273         android:text="@string/164c2" />
1274
1275     </TableRow>
1276
1277     <TableRow
1278         android:layout_width="match_parent"
1279         android:layout_height="wrap_content"
1280         android:background="#e0e0e0"
1281         android:gravity="center">
1282
1283     <TextView
1284         android:layout_margin="8dp"
1285         android:layout_weight="2"
1286         android:width="0dp"
1287         android:text="@string/165c1" />
1288
1289     <TextView
1290         android:layout_margin="8dp"
1291         android:layout_weight="3"
1292         android:width="0dp"
1293         android:text="@string/165c2" />
1294
1295     </TableRow>
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity\_ajuda\_tipos\_de\_servicos.xml

```
1296     <TableRow
1297         android:layout_width="match_parent"
1298         android:layout_height="wrap_content"
1299         android:background="@color/primary_light"
1300         android:gravity="center">
1301
1302         <TextView
1303             android:layout_margin="8dp"
1304             android:layout_weight="2"
1305             android:width="0dp"
1306             android:text="@string/166c1" />
1307
1308         <TextView
1309             android:layout_margin="8dp"
1310             android:layout_weight="3"
1311             android:width="0dp"
1312             android:text="@string/166c2" />
1313     </TableRow>
1314
1315     <TableRow
1316         android:layout_width="match_parent"
1317         android:layout_height="wrap_content"
1318         android:background="#e0e0e0"
1319         android:gravity="center">
1320
1321         <TextView
1322             android:layout_margin="8dp"
1323             android:layout_weight="2"
1324             android:width="0dp"
1325             android:text="@string/167c1" />
1326
1327         <TextView
1328             android:layout_margin="8dp"
1329             android:layout_weight="3"
1330             android:width="0dp"
1331     </TableRow>
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity\_ajuda\_tipos\_de\_servicos.xml

```
1331      android:text="@string/167c2" />
1332    </TableRow>
1333
1334  </TableLayout>
1335
1336  </android.support.v4.widget.NestedScrollView>
1337
1338  </android.support.design.widget.CoordinatorLayout>
1339
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity\_ajuda\_tipos\_estabelecimentos.xml

```
1 <android.support.design.widget.CoordinatorLayout xmlns:android="http://schemas.android.com/apk/res/android"
2     xmlns:app="http://schemas.android.com/apk/res-auto"
3         android:layout_width="match_parent"
4             android:layout_height="match_parent">
5
6     <android.support.design.widget.AppBarLayout
7         android:layout_width="match_parent"
8             android:layout_height="wrap_content"
9                 android:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar">
10
11         <android.support.v7.widget.Toolbar
12             android:id="@+id/toolbar_actvt_ajuda_tipos_estabelecimentos"
13                 android:layout_width="match_parent"
14                     android:layout_height="?attr/actionBarSize"
15                     android:background="?attr/colorPrimary"
16                     app:layout_scrollFlags="scrollEnterAlways"
17                     app:popupTheme="@style/ThemeOverlay.AppCompat.Light" />
18
19         <android.support.design.widget.AppBarLayout>
20
21             <android.support.v4.widget.NestedScrollView
22                 android:id="@+id/nestedScrollView"
23                     android:layout_width="match_parent"
24                     android:layout_height="match_parent"
25                     android:scrollbars="vertical"
26                     android:layout_behavior="@string/appbar_scrolling_view_behavior">
27
28                 <LinearLayout
29                     android:layout_width="match_parent"
30                     android:layout_height="match_parent"
31                         android:layout_margin="16dp"
32                         android:orientation="vertical"
33                         android:paddingBottom="16dp">
34
35                     <TextView
36                         android:layout_width="match_parent"
37                         android:layout_height="wrap_content"
38                         android:text="Aqui vai o conteúdo da tela de ajuda de tipos de estabelecimentos."/>
39
40                 </LinearLayout>
41             </NestedScrollView>
42         </AppBarLayout>
43     </CoordinatorLayout>
```

```
File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity_ajuda_tipos_estabelecimentos.xml
36      android:layout_height="wrap_content"
37      android:text="@string/centro_hemato_titulo"
38      android:textAppearance="@android:style/TextAppearance.Medium" />
39
40      <TextView
41          android:layout_width="match_parent"
42          android:layout_height="wrap_content"
43          android:layout_marginBottom="16dp"
44          android:text="@string/centro_hemato" />
45
46      <TextView
47          android:layout_width="match_parent"
48          android:layout_height="wrap_content"
49          android:text="@string/caps_titulo"
50          android:textAppearance="@android:style/TextAppearance.Medium" />
51
52      <TextView
53          android:layout_width="match_parent"
54          android:layout_height="wrap_content"
55          android:layout_marginBottom="16dp"
56          android:text="@string/caps" />
57
58      <TextView
59          android:layout_width="match_parent"
60          android:layout_height="wrap_content"
61          android:text="@string/parto_isolado_titulo"
62          android:textAppearance="@android:style/TextAppearance.Medium" />
63
64      <TextView
65          android:layout_width="match_parent"
66          android:layout_height="wrap_content"
67          android:layout_marginBottom="16dp"
68          android:text="@string/parto_isolado" />
69      <TextView
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity\_ajuda\_tipos\_estabelecimentos.xml

```
71     android:layout_width="match_parent"
72     android:layout_height="wrap_content"
73     android:text="@string/centro_saude_titulo"
74     android:textAppearance="@android:style/TextAppearance.Medium" />
75
76     <TextView
77         android:layout_width="match_parent"
78         android:layout_height="wrap_content"
79         android:layout_marginBottom="16dp"
80         android:text="@string/centro_saude" />
81
82     <TextView
83         android:layout_width="match_parent"
84         android:layout_height="wrap_content"
85         android:text="@string/clinica_titulo"
86         android:textAppearance="@android:style/TextAppearance.Medium" />
87
88     <TextView
89         android:layout_width="match_parent"
90         android:layout_height="wrap_content"
91         android:text="@string/consultorio_isolado_titulo"
92         android:textAppearance="@string/clinica" />
93
94     <TextView
95         android:layout_width="match_parent"
96         android:layout_height="wrap_content"
97         android:text="@string/consultorio_isolado"
98         android:textAppearance="@android:style/TextAppearance.Medium" />
99
100    <TextView
101        android:layout_width="match_parent"
102        android:layout_height="wrap_content"
103        android:layout_marginBottom="16dp"
104        android:text="@string/consultorio_isolado" />
105
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity\_ajuda\_tipos\_estabelecimentos.xml

```
106 <TextView  
107     android:layout_width="match_parent"  
108     android:layout_height="wrap_content"  
109     android:text="@string/hospital_especializado_titulo"  
110     android:textAppearance="@android:style/TextAppearance.Medium" />  
111  
112     <TextView  
113         android:layout_width="match_parent"  
114         android:layout_height="wrap_content"  
115         android:layout_marginBottom="16dp"  
116         android:text="@string/hospital_especializado" />  
117  
118     <TextView  
119         android:layout_width="match_parent"  
120         android:layout_height="wrap_content"  
121         android:text="@string/hospital_geral_titulo"  
122         android:textAppearance="@android:style/TextAppearance.Medium" />  
123  
124     <TextView  
125         android:layout_width="match_parent"  
126         android:layout_height="wrap_content"  
127         android:layout_marginBottom="16dp"  
128         android:text="@string/hospital_geral" />  
129  
130     <TextView  
131         android:layout_width="match_parent"  
132         android:layout_height="wrap_content"  
133         android:text="@string/hospital_dia_titulo"  
134         android:textAppearance="@android:style/TextAppearance.Medium" />  
135  
136     <TextView  
137         android:layout_width="match_parent"  
138         android:layout_height="wrap_content"  
139         android:layout_marginBottom="16dp"  
140         android:text="@string/hospital_dia" />
```

```
141
142    <TextView
143        android:layout_width="match_parent"
144        android:layout_height="wrap_content"
145        android:text="@string/labo_saude_publica_titulo"
146        android:textAppearance="@android:style/TextAppearance.Medium" />
147
148    <TextView
149        android:layout_width="match_parent"
150        android:layout_height="wrap_content"
151        android:layout_marginBottom="16dp"
152        android:text="@string/labo_saude_publica" />
153
154    <TextView
155        android:layout_width="match_parent"
156        android:layout_height="wrap_content"
157        android:text="@string/policlinica_titulo"
158        android:textAppearance="@android:style/TextAppearance.Medium" />
159
160    <TextView
161        android:layout_width="match_parent"
162        android:layout_height="wrap_content"
163        android:layout_marginBottom="16dp"
164        android:text="@string/policlinica" />
165
166    <TextView
167        android:layout_width="match_parent"
168        android:layout_height="wrap_content"
169        android:text="@string/posto_de_saude_titulo"
170        android:textAppearance="@android:style/TextAppearance.Medium" />
171
172    <TextView
173        android:layout_width="match_parent"
174        android:layout_height="wrap_content"
175        android:layout_marginBottom="16dp"
176        android:text="--"
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity\_ajuda\_tipos\_estabelecimentos.xml

```
176     android:text="@string/posto_de_saude" />
177
178     <TextView
179         android:layout_width="match_parent"
180         android:layout_height="wrap_content"
181         android:text="@string/upa_titulo"
182         android:textAppearance="@android:style/TextAppearance.Medium" />
183
184     <TextView
185         android:layout_width="match_parent"
186         android:layout_height="wrap_content"
187         android:layout_marginBottom="16dp"
188         android:text="@string/upa" />
189
190     <TextView
191         android:layout_width="match_parent"
192         android:layout_height="wrap_content"
193         android:text="@string/pronto_socorro_especializado_titulo"
194         android:textAppearance="@android:style/TextAppearance.Medium" />
195
196     <TextView
197         android:layout_width="match_parent"
198         android:layout_height="wrap_content"
199         android:text="@string/pronto_socorro_especializado"
200         android:textAppearance="@android:style/TextAppearance.Medium" />
201
202     <TextView
203         android:layout_width="match_parent"
204         android:layout_height="wrap_content"
205         android:text="@string/pronto_socorro_geral_titulo"
206         android:textAppearance="@android:style/TextAppearance.Medium" />
207
208     <TextView
209         android:layout_width="match_parent"
210         android:layout_height="wrap_content"
211         ...
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity\_ajuda\_tipos\_estabelecimentos.xml

```
211     android:layout_marginBottom="16dp"
212     android:text="@string/pronto_socorro_geral" />
213
214     <TextView
215         android:layout_width="match_parent"
216         android:layout_height="wrap_content"
217         android:text="@string/sadt_titulo"
218         android:textAppearance="@android:style/TextAppearance.Medium" />
219
220     <TextView
221         android:layout_width="match_parent"
222         android:layout_height="wrap_content"
223         android:layout_marginBottom="16dp"
224         android:text="@string/sadt" />
225
226     <TextView
227         android:layout_width="match_parent"
228         android:layout_height="wrap_content"
229         android:text="@string/unidade_mista_titulo"
230         android:textAppearance="@android:style/TextAppearance.Medium" />
231
232     <TextView
233         android:layout_width="match_parent"
234         android:layout_height="wrap_content"
235         android:layout_marginBottom="16dp"
236         android:text="@string/unidade_mista" />
237
238     </LinearLayout>
239     </android.support.v4.widget.NestedScrollView>
240
241 </android.support.design.widget.CoordinatorLayout>
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity\_busca\_geral.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <android.support.design.widget.CoordinatorLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:fab="http://schemas.android.com/apk/res-auto"
5     xmlns:tools="http://schemas.android.com/tools"
6     android:id="@+id/activity_busca_geral_layout"
7     android:layout_width="match_parent"
8     android:layout_height="match_parent"
9     android:fitsSystemWindows="true"
10    tools:context="br.com.iucemar.dimon.cadesaude.BuscaGeralActivity">
11
12    <android.support.design.widget.AppBarLayout
13        android:layout_width="match_parent"
14        android:layout_height="wrap_content"
15        android:theme="@style/MeuTemaMaterial.AppBarOverlay">
16
17        <android.support.v7.widget.Toolbar
18            android:id="@+id/toolbar_actvt_busca_geral"
19            android:layout_width="match_parent"
20            android:layout_height="?attr/actionBarSize"
21            android:background="?attr/colorPrimary"
22            app:popupTheme="@style/MeuTemaMaterial.PopupOverlay" />
23
24    </android.support.design.widget.AppBarLayout>
25
26    <include layout="@layout/content_busca_geral" />
27
28    <com.github.clans.fab.FloatingActionButton
29        android:id="@+id/fab_activity_busca_geral"
30        android:layout_width="wrap_content"
31        android:layout_height="wrap_content"
32        android:layout_gravity="end|bottom"
33        android:layout_margin="16dp"
34        android:src="@drawable/ic_magnify_white_24dp"
35        fab:fab_label="Busca por tipos"
36    <!--
```

File - E:\Android\Projetos\CadESaude\app\src\main\res\layout\activity\_busca\_geral.xml

```
36     fab:fab_size="normal" />
37
38 </android.support.design.widget.CoordinatorLayout>
39
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity\_busca\_gps.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <android.support.design.widget.CoordinatorLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:fab="http://schemas.android.com/apk/res-auto"
5     xmlns:tools="http://schemas.android.com/tools"
6         android:id="@+id/activity_busca_gps_layout"
7         android:layout_width="match_parent"
8         android:layout_height="match_parent"
9         android:fitsSystemWindows="true"
10        tools:context="br.com.iucemar.dimon.cadesaude.BuscaGPSActivity">
11
12    <android.support.design.widget.AppBarLayout
13        android:layout_width="match_parent"
14        android:layout_height="wrap_content"
15        android:theme="@style/MeuTemaMaterial.AppBarOverlay">
16
17        <android.support.v7.widget.Toolbar
18            android:id="@+id/toolbar_actvt_busca_gps"
19            android:layout_width="match_parent"
20            android:layout_height="?attr/actionBarSize"
21            app:popupTheme="@style/MeuTemaMaterial.PopupOverlay" />
22
23    </android.support.design.widget.AppBarLayout>
24
25    <include layout="@layout/content_busca_gps" />
26
27    <com.github.clans.fab.FloatingActionButton
28        android:id="@+id/fab_filtrar_menu"
29        android:layout_width="match_parent"
30        android:layout_height="match_parent"
31        android:layout_gravity="end|bottom"
32        android:elevation="2dp"
33        android:padding="@dimen/text_margin"
34        android:visibility="gone"
35        fab:menu_animationDelayPerItem="30"
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity\_busca\_gps.xml

```
36     fab:menu_backgroundColor="#AFFFFFFF"
37     fab:menu_buttonSpacing="0dp"
38     fab:menu_colorNormal="#E53935"
39     fab:menu_colorPressed="#FF5252"
40     fab:menu_colorRipple="#99FFFFFF"
41     fab:menu_fab_hide_animation="@anim/fab_scale_down"
42     fab:menu_fab_label="Fechar"
43     fab:menu_fab_show_animation="@anim/fab_scale_up"
44     fab:menu_fab_size="normal"
45     fab:menu_icon="@drawable/ic_filter_white_24dp"
46     fab:menu_labels_colorNormal="#333333"
47     fab:menu_labels_colorPressed="#444444"
48     fab:menu_labels_colorRipple="#66FFFFFF"
49     fab:menu_labels_cornerRadius="3dp"
50     fab:menu_labels_ellipsize="none"
51     fab:menu_labels_margin="0dp"
52     fab:menu_labels_maxLines="-1"
53     fab:menu_labels_padding="8dp"
54     fab:menu_labels_paddingBottom="4dp"
55     fab:menu_labels_paddingLeft="8dp"
56     fab:menu_labels_paddingRight="8dp"
57     fab:menu_labels_paddingTop="4dp"
58     fab:menu_labels_position="left"
59     fab:menu_labels_showShadow="true"
60     fab:menu_labels_singleLine="false"
61     fab:menu_labels_textColor="#FFFFFF"
62     fab:menu_labels_textSize="14sp"
63     fab:menu_openDirection="up"
64     fab:menu_shadowColor="#66000000"
65     fab:menu_shadowRadius="2dp"
66     fab:menu_shadowXOffset="1dp"
67     fab:menu_shadowYOffset="2dp"
68
69 <com.github.clans.floatingActionButton
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity\_busca\_gps.xml

```
71    android:id="@+id/fab_filtro_limpar_todos"
72    android:layout_width="wrap_content"
73    android:layout_height="wrap_content"
74    android:src="@drawable/ic_filter_remove_white_24dp"
75    fab:fab_label="@string/fab_filtro_limpar_todos"
76    fab:size="mini" />
77
78 <com.github.clans.fab.FloatingActionButton
79     android:id="@+id/fab_filtro_tipo_convenio"
80     android:layout_width="wrap_content"
81     android:layout_height="wrap_content"
82     android:src="@drawable/ic_credit_card_white_24dp"
83     fab:fab_label="@string/fab_filtro_tipo_convenio"
84     fab:size="mini" />
85
86 <com.github.clans.fab.FloatingActionButton
87     android:id="@+id/fab_filtro_tipo_atend"
88     android:layout_width="wrap_content"
89     android:layout_height="wrap_content"
90     android:src="@drawable/ic_healing_white_24dp"
91     fab:fab_label="@string/fab_filtro_tipo_atendimento"
92     fab:size="mini" />
93
94 <com.github.clans.fab.FloatingActionButton
95     android:id="@+id/fab_filtro_tipo_prof"
96     android:layout_width="wrap_content"
97     android:layout_height="wrap_content"
98     android:src="@drawable/ic_school_white_24dp"
99     fab:fab_label="@string/fab_filtro_tipo_profissional"
100    fab:size="mini" />
101
102 <com.github.clans.fab.FloatingActionButton
103     android:id="@+id/fab_filtro_tipo_serv"
104     android:layout_width="wrap_content"
105     android:layout_height="wrap_content"
106    ...
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity\_busca\_gps.xml

```
106     android:src="@drawable/ic_local_hospital_white_24dp"
107     fab:fab_label="@string/fab_filtro_tipo_servico"
108     fab:fab_size="mini" />
109
110     <com.github.clans.fab.FloatingActionButton
111         android:id="@+id/fab_filtro_tipo_estab"
112         android:layout_width="wrap_content"
113         android:layout_height="wrap_content"
114         android:src="@drawable/ic_stethoscope_white_24dp"
115         fab:fab_label="@string/fab_filtro_tipo_estabelecimento"
116         fab:fab_size="mini" />
117     </com.github.clans.fab.FloatingActionButton>
118
119     </android.support.design.widget.CoordinatorLayout>
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity\_estabelecimento.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <android.support.design.widget.CoordinatorLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:fab="http://schemas.android.com/apk/res-auto"
5     xmlns:tools="http://schemas.android.com/tools"
6         android:layout_width="match_parent"
7         android:layout_height="match_parent"
8         android:fitsSystemWindows="true"
9         tools:context="br.com.jucemar.dimon.cadesaude.EstabLecimientoActivity">
10
11     <android.support.design.widget.AppBarLayout
12         android:layout_width="match_parent"
13         android:layout_height="wrap_content"
14         android:theme="@style/MeuTemaMaterial.AppBarOverlay">
15
16         <android.support.v7.widget.Toolbar
17             android:id="@+id/toolbar_actvt_estabelecimento"
18             android:layout_width="match_parent"
19             android:layout_height="wrap_content"
20             android:background="?attr/colorPrimary"
21             app:layout_scrollFlags="scrollEnterAlways"
22             app:popupTheme="@style/MeuTemaMaterial.PopupOverlay">
23
24             <TextView
25                 android:id="@+id/titulo_estab_activity_estab"
26                 android:layout_width="match_parent"
27                 android:layout_height="match_parent"
28                 android:layout_marginRight="8dp"
29                 android:gravity="center_vertical"
30                 android:textAppearance="@style/TextAppearance.Medium" />
31
32         <android.support.design.widget.TabLayout
33             android:id="@+id/tabs"
34             android:layout_width="match_parent"
35         <!--
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity\_estabelecimento.xml

```
36     android:layout_height="wrap_content"
37     app:tabGravity="fill"
38     app:tabIndicatorColor="@color/icons"
39     app:tabMode="scrollable"
40     app:tabSelectedTextColor="@color/icons"
41     app:tabTextColor="@color/primary_light" />
42   </android.support.design.widget.AppBarLayout>
43
44   <ProgressBar
45     android:id="@+id/barra_progresso_estabelecimento_activity"
46     android:layout_width="wrap_content"
47     android:layout_height="wrap_content"
48     android:layout_gravity="center"
49     android:visibility="visible" />
50
51   <android.support.v4.view.ViewPager
52     android:id="@+id/viewpager"
53     android:layout_width="match_parent"
54     android:layout_height="match_parent"
55     android:background="@color/background_material_light"
56     android:visibility="gone"
57     app:layout_behavior="@string/appbar_scrolling_view_behavior" />
58   </android.support.design.widget.CoordinatorLayout>
59
60
```

File - E:\Android\Projetos\CadEStaude\app\src\main\res\layout\activity\_main.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <android.support.design.widget.CoordinatorLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:fab="http://schemas.android.com/apk/res-auto"
5     xmlns:tools="http://schemas.android.com/tools"
6     android:id="@+id/coordinator_layout_main_activity"
7     android:layout_width="match_parent"
8     android:layout_height="match_parent"
9     android:fitsSystemWindows="true"
10    tools:context="br.com.iucemar.dimon.cadesaude.MainActivity">
11
12    <android.support.design.widget.AppBarLayout
13        android:id="@+id/appbar"
14        android:layout_width="match_parent"
15        android:layout_height="wrap_content"
16        android:theme="@style/MeuTemaMaterial.AppBarOverlay">
17
18        <android.support.v7.widget.Toolbar
19            android:id="@+id/toolbar_main"
20            android:layout_width="match_parent"
21            android:layout_height="?attr/actionBarSize"
22            android:background="?attr/colorPrimary"
23            app:popupTheme="@style/MeuTemaMaterial.PopupOverlay" />
24
25        <LinearLayout
26            android:layout_width="match_parent"
27            android:layout_height="wrap_content"
28            android:orientation="horizontal"
29            android:padding="16dp">
30
31            <ImageView
32                android:layout_width="wrap_content"
33                android:layout_height="wrap_content"
34                android:layout_marginRight="16dp"
35                android:src="@drawable/ic_star_white_24dp" />
```

```
36      <TextView  
37          android:layout_width="match_parent"  
38          android:layout_height="wrap_content"  
39              android:layout_gravity="center_vertical"  
40              android:text="Meus favoritos"  
41              android:textAppearance="@android:style/TextAppearance.Medium" />  
42      </LinearLayout>  
43  </android.support.design.widget.AppBarLayout>  
44  
45  <include layout="@layout/content_main" />  
46  
47  <com.github.clans.fab.FloatingActionButton  
48      android:id="@+id/fab_busca_parametros"  
49          android:layout_width="wrap_content"  
50              android:layout_height="wrap_content"  
51              android:layout_gravity="end|bottom"  
52          android:layout_marginBottom="16dp"  
53          android:layout_marginRight="16dp"  
54          android:padding="@dimen/text_margin"  
55          android:src="@drawable/ic_magnify_white_24dp"  
56          app:colorNormal="#DA4336"  
57          app:colorPressed="#B75043"  
58          app:colorRipple="#99FFFFFF"  
59          app:hideAnimation="@anim/fab_scale_down"  
60          app:label="Busca por parâmetros"  
61          app:shadowColor="#66000000"  
62          app:shadowRadius="4dp"  
63          app:shadowXOffset="1dp"  
64          app:shadowYOffset="3dp"  
65          app:showAnimation="@anim/fab_scale_up"  
66          app:showShadow="true"  
67          android:elevation="10dp" />  
68  
69  <com.github.clans.fab.FloatingActionButton
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity\_main.xml

```
71     android:id="@+id/fab_busca_gps"
72     android:layout_width="wrap_content"
73     android:layout_height="wrap_content"
74     android:layout_marginBottom="88dp"
75     android:layout_marginRight="16dp"
76     android:padding="@dimen/text_margin"
77     android:src="@drawable/ic_crosshairs_gps_white_24dp"
78     app:fab_colorNormal="#DA4336"
79     app:fab_colorPressed="#E75043"
80     app:fab_colorRipple="#99FFFFFF"
81     app:fab_hideAnimation="@anim/fab_scale_down"
82     app:fab_label="Busca"      por       Localização"
83     app:fab_shadowColor="#66000000"
84     app:fab_shadowRadius="4dp"
85     app:fab_shadowXOffset="1dp"
86     app:fab_shadowYOffset="3dp"
87     app:fab_showAnimation="@anim/fab_scale_up"
88     app:fab_showShadow="true"
89     app:fab_size="normal"
90     app:layout_anchor="@+id/fab_busca_parametros"
91     app:layout_anchorGravity="center" />
92 </android.support.design.widget.CoordinatorLayout>
93
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity\_primeira\_configuracao.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2
3 <FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
4   xmlns:app="http://schemas.android.com/apk/res-auto"
5   xmlns:tools="http://schemas.android.com/tools"
6   android:id="@+id/layout_primeira_config_activity"
7   android:layout_width="match_parent"
8   android:layout_height="match_parent"
9   tools:context="br.com.jucemar.dimon.cadesaude.PrimeiraConfiguracaoActivity">
10
11   <LinearLayout
12     android:layout_width="wrap_content"
13     android:layout_height="wrap_content"
14     android:layout_gravity="center"
15     android:orientation="vertical">
16
17     <ProgressBar
18       android:id="@+id/barra_progresso_primeira_conf_activity"
19       android:layout_width="wrap_content"
20       android:layout_height="wrap_content"
21       android:layout_gravity="center"
22       android:visibility="visible" />
23
24     <ImageView
25       android:id="@+id/background_erro_primeira_conf_activity"
26       android:layout_width="wrap_content"
27       android:layout_height="wrap_content"
28       android:layout_gravity="center_horizontal"
29       android:src="@drawable/crosshairs"
30       android:visibility="gone" />
31
32     <TextView
33       android:id="@+id/label_erro_primeira_config_activity"
34       android:layout_width="wrap_content"
35       android:layout_height="wrap_content"
36       android:text="ERRO: Acesso negado ao sistema de cadastro de saude."/>
37
38   </LinearLayout>
39
40 </FrameLayout>
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity\_primeira\_configuracao.xml

```
36     android:layout_gravity="center_horizontal"
37     android:layout_marginTop="30dp"
38     android:text="@string/txt_erro_gps"
39     android:textAppearance="@android:style/TextAppearance_Large"
40     android:textColor="#BDBDBD"
41     android:visibility="gone" />
42 </LinearLayout>
43 </FrameLayout>
44 </FrameLayout>
```

```
File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity_primeiros_passos.xml

1 <FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
2   xmlns:tools="http://schemas.android.com/tools"
3     android:layout_width="match_parent"
4     android:layout_height="match_parent"
5     tools:context="br.com.jucemar_dimon.cadesaude.PrimeirosPassosActivity">
6
7   <LinearLayout
8     android:layout_width="match_parent"
9     android:layout_height="match_parent"
10    android:background="@color/primary_light"
11    android:orientation="vertical">
12
13     <ImageView
14       android:layout_width="wrap_content"
15       android:layout_height="0dp"
16       android:layout_marginBottom="16dp"
17       android:layout_marginTop="72dp"
18       android:layout_weight="2"
19       android:src="@drawable/ic_laucher_web" />
20
21     <TextView
22       android:id="@+id/fullscreen_content"
23       android:layout_width="wrap_content"
24       android:layout_height="0dp"
25       android:layout_margin="16dp"
26       android:layout_marginBottom="16dp"
27       android:layout_weight="2"
28       android:gravity="center_horizontal"
29       android:keepScreenOn="true"
30       android:text="@string/dummy_content"
31       android:textColor="@color/primary_text"
32       android:textSize="14sp" />
33
34 </LinearLayout>
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity\_primeiros\_passos.xml

```
36 <FrameLayout
37     android:layout_width="match_parent"
38     android:layout_height="match_parent"
39     android:layout_gravity="left|center_vertical"
40     android:fitsSystemWindows="true">
41
42     <LinearLayout
43         android:id="@+id/fullscreen_content_controls"
44         style="?metaButtonBarStyle"
45         android:layout_width="match_parent"
46         android:layout_height="wrap_content"
47         android:layout_gravity="bottom|center_horizontal"
48         android:background="@color/primary"
49         android:orientation="horizontal"
50         tools:ignore="UselessParent">
51
52         <Button
53             android:id="@+id/dummy_button"
54             style="?metaButtonBarButtonStyle"
55             android:layout_width="0dp"
56             android:layout_height="wrap_content"
57             android:layout_weight="1"
58             android:enabled="true"
59             android:text="@string/dummy_button"
60             android:textAppearance="@android:style/TextAppearance_Medium"
61             android:textColor="@color/primary_light" />
62
63     </LinearLayout>
64
65 </FrameLayout>
66 --
```

File - E:\Android\Projetos\CadeSaude\app\src\main\res\layout\activity\_sobre.xml

```
1 <android.support.design.widget.CoordinatorLayout xmlns:android="http://schemas.android.com/apk/res/android"
2   xmlns:app="http://schemas.android.com/apk/res-auto"
3     android:layout_width="match_parent"
4     android:layout_height="match_parent">
5
6   <android.support.design.widget.AppBarLayout
7     android:layout_width="match_parent"
8     android:layout_height="wrap_content"
9     android:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar">
10
11     <android.support.v7.widget.Toolbar
12       android:id="@+id/toolbar_actvt_sobre"
13       android:layout_width="match_parent"
14       android:layout_height="?attr/actionBarSize"
15       android:background="?attr/colorPrimary"
16       app:layout_scrollFlags="scrollEnterAlways"
17       app:popupTheme="@style/ThemeOverlay.AppCompat.Light" />
18
19   </android.support.design.widget.AppBarLayout>
20
21   <android.support.v4.widget.NestedScrollView
22     android:id="@+id/nestedScrollView"
23     android:layout_width="match_parent"
24     android:layout_height="match_parent"
25     android:scrollbars="vertical"
26     android:layout_behavior="@string/appbar_scrolling_view_behavior">
27
28     <LinearLayout
29       android:layout_width="match_parent"
30       android:layout_height="wrap_content"
31       android:layout_margin="16dp"
32       android:orientation="vertical">
33
34       android:layout_width="match_parent"
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity\_sobre.xml

```
36     android:layout_height="match_parent"
37     android:layout_marginBottom="16dp"
38     android:layout_marginTop="16dp"
39     android:gravity="center_horizontal"
40     android:text="@string/appNome" />
41
42     <ImageView
43         android:layout_width="wrap_content"
44         android:layout_height="wrap_content"
45         android:gravity="center_horizontal"
46         android:src="@mipmap/ic_launcher" />
47
48     <TextView
49         android:id="@+id/sobre_txv_app_versao"
50         android:layout_width="match_parent"
51         android:layout_height="match_parent"
52         android:layout_marginTop="16dp"
53         android:gravity="center_horizontal"
54         android:text="@string/appVersao" />
55
56     <TextView
57         android:id="@+id/sobre_txv_bd_versao"
58         android:layout_width="match_parent"
59         android:layout_height="match_parent"
60         android:layout_marginTop="16dp"
61         android:gravity="center_horizontal"
62         android:text="@string/bdVersao" />
63
64     <TextView
65         android:layout_width="match_parent"
66         android:layout_height="match_parent"
67         android:layout_gravity="fill_horizontal"
68         android:layout_marginTop="16dp"
69         android:gravity="center_horizontal"
70         android:paddingBottom="32dp"
71     </ViewGroup>
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity\_sobre.xml

```
71   android:text="@string/appDesc" />
72
73 </LinearLayout>
74 </android.support.v4.widget.NestedScrollView>
75
76 </android.support.design.widget.CoordinatorLayout>
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity\_tela\_inicial.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2
3 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     android:background="@drawable/city"
8     android:orientation="vertical"
9     android:paddingBottom="@dimen/activity_vertical_margin"
10    android:paddingLeft="@dimen/activity_horizontal_margin"
11    android:paddingRight="@dimen/activity_horizontal_margin"
12    android:paddingTop="@dimen/activity_vertical_margin"
13    tools:context="br.com.iucemar.dimon.cadesaude.TelaInicialActivity">
14
15 <ImageView
16     android:id="@+id/imageView"
17     android:layout_width="wrap_content"
18     android:layout_height="0dp"
19     android:layout_gravity="center_vertical"
20     android:layout_marginTop="64dp"
21     android:layout_weight="2"
22     android:src="@drawable/ic_launcher_web" />
23
24 <TextView
25     android:layout_width="wrap_content"
26     android:layout_height="0dp"
27     android:layout_gravity="center_horizontal"
28     android:layout_weight="1"
29     android:gravity="center"
30     android:text="CadEsaude"
31     android:textAppearance="@style/TextAppearance.Large" />
32
33 <TextView
34     android:layout_width="wrap_content"
35     android:layout_height="0dp"
36     android:layout_weight="1"
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity\_tela\_inicial.xml

```
36     android:layout_gravity="fill_horizontal"
37     android:layout_marginBottom="0dp"
38     android:layout_weight="1"
39     android:gravity="center"
40     android:text="Cadastro dos Estabelecimentos de Saúde de Santa Catarina"
41     android:textAppearance="@android:style/TextAppearance.Medium" />
42
43     <TextView
44         android:id="@+id/txtv_versao_app"
45         android:layout_width="wrap_content"
46         android:layout_height="0dp"
47         android:layout_gravity="right"
48         android:layout_weight="1"
49         android:gravity="bottom"
50         android:textAppearance="@android:style/TextAppearance.Small" />
51
52     </LinearLayout>
53
54
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\content\_busca\_geral.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5         android:layout_width="match_parent"
6         android:layout_height="match_parent"
7         app:layout_behavior="@string/appbar_scrolling_view_behavior"
8         tools:context="br.com.jucemar_dimon.cadesaude.MainActivity">
9
10    <android.support.v7.widget.RecyclerView
11        android:id="@+id/lista_estabelecimentos"
12        android:layout_width="match_parent"
13        android:layout_height="match_parent"
14        android:backgroundTint="@color/divider"
15        android:scrollbars="vertical"
16        android:visibility="gone" />
17
18    <ProgressBar
19        android:id="@+id/barra_progresso_busca_geral_activity"
20        android:layout_width="wrap_content"
21        android:layout_height="wrap_content"
22        android:layout_gravity="center"
23        android:visibility="visible" />
24
25    <LinearLayout
26        android:layout_width="wrap_content"
27        android:layout_height="wrap_content"
28        android:layout_gravity="center"
29        android:orientation="vertical">
30
31        <ImageView
32            android:id="@+id/background_erro_busca_geral"
33            android:layout_width="wrap_content"
34            android:layout_height="wrap_content"
35            android:layout_gravity="center_horizontal"
36        </>
37    </LinearLayout>
38
39</FrameLayout>
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\content\_busca\_geral.xml

```
36     android:src="@drawable/magnify_background"
37     android:visibility="visible" />
38
39     <TextView
40         android:id="@+id/label_erro_busca"
41         android:layout_width="wrap_content"
42         android:layout_height="wrap_content"
43         android:layout_gravity="center_horizontal"
44         android:text="@string/txt_erro_busca_geral"
45         android:textAppearance="@android:style/TextAppearance_Large"
46         android:textColor="#BDBDBD"
47         android:visibility="visible" />
48
49     <TextView
50         android:id="@+id/label_municipios_para_busca"
51         android:layout_width="wrap_content"
52         android:layout_height="wrap_content"
53         android:layout_gravity="center_horizontal"
54         android:layout_margin="16dp"
55         android:gravity="center"
56         android:textAppearance="@android:style/TextAppearance_Small"
57         android:textColor="#BDBDBD"
58         android:visibility="visible" />
59     </LinearLayout>
60
61     </FrameLayout>
62
63     <!--
```

File - E:\Android\Projetos\CadESEsaudé\app\src\main\res\layout\content\_busca\_gps.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
3   xmlns:app="http://schemas.android.com/apk/res-auto"
4   xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     app:layout_behavior="@string/appbar_scrolling_view_behavior"
8     tools:context="br.com.jucemar_dimon.cadesaude.BuscaGPSActivity">
9
10   <android.support.v7.widget.RecyclerView
11     android:id="@+id/lista_estabelecimentos"
12     android:layout_width="match_parent"
13     android:layout_height="match_parent"
14     android:backgroundTint="@color/divider"
15     android:scrollbars="vertical"
16     android:visibility="gone" />
17
18   <LinearLayout
19     android:layout_width="wrap_content"
20     android:layout_height="wrap_content"
21     android:layout_gravity="center"
22     android:orientation="vertical">
23
24     <ProgressBar
25       android:id="@+id/barra_progresso_gps_activity"
26       android:layout_width="wrap_content"
27       android:layout_height="wrap_content"
28       android:layout_gravity="center"
29       android:visibility="visible" />
30
31   <LinearLayout
32     android:layout_width="wrap_content"
33     android:layout_height="wrap_content"
34     android:layout_gravity="center"
35     android:orientation="vertical">
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\content\_busca\_gps.xml

```
36     android:layout_height="wrap_content"
37     android:layout_gravity="center"
38     android:orientation="vertical">>
39
40     <ImageView
41         android:id="@+id/background_error_gps"
42         android:layout_width="wrap_content"
43         android:layout_height="wrap_content"
44         android:layout_gravity="center_horizontal"
45         android:src="@drawable/crosshairs"
46         android:visibility="gone" />
47
48     <TextView
49         android:id="@+id/label_error_gps"
50         android:layout_width="wrap_content"
51         android:layout_height="wrap_content"
52         android:layout_gravity="center_horizontal"
53         android:layout_marginTop="30dp"
54         android:text="@string/txt_error_gps"
55         android:textAppearance="@android:style/TextAppearance.Large"
56         android:textColor="#BDBDBD"
57         android:visibility="gone" />
58
59     </LinearLayout>
60
61
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\content\_fragment\_estabelecimento.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
3     android:id="@+id/scrol1_estabelecimento"
4     android:layout_width="match_parent"
5     android:layout_height="match_parent">
6
7     <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
8         xmlns:tools="http://schemas.android.com/tools"
9             android:layout_width="match_parent"
10            android:layout_height="match_parent"
11            android:padding="@dimen/text_margin">
12
13         <ImageButton
14             android:id="@+id/icon_rzao_social"
15             android:layout_width="wrap_content"
16             android:layout_height="wrap_content"
17             android:layout_marginBottom="16dp"
18             android:layout_marginRight="16dp"
19             android:background="@android:color/transparent"
20             android:src="@drawable/ic_briefcase_grey600_24dp" />
21
22         <TextView
23             android:id="@+id/label_rzao_social_abrir_estabelecimento"
24             android:layout_width="wrap_content"
25             android:layout_height="wrap_content"
26             android:layout_alignParentEnd="true"
27             android:layout_alignParentRight="true"
28             android:layout_alignTop="@+id/icon_rzao_social"
29             android:layout_marginBottom="4dp"
30             android:layout_toEndOf="@+id/icon_rzao_social"
31             android:layout_toRightOf="@+id/icon_rzao_social"
32             android:layout="@+id/icon_rzao_social"
33             android:lines="2"           android:text="Razão"           Social:
34             android:textAppearance="?android:attr/textAppearanceMedium"      />
```

```
36
37 <TextView
38     android:id="@+id/label_nome_fantasia_abrir_estabelecimento"
39     android:layout_width="wrap_content"
40     android:layout_height="wrap_content"
41     android:layout_alignLeft="@+id/label_razao_social_abrir_estabelecimento"
42     android:layout_alignRight="true"
43     android:layout_alignParentStart="@+id/label_razao_social_abrir_estabelecimento"
44     android:layout_below="@+id/label_razao_social_abrir_estabelecimento"
45     android:layout_marginBottom="16dp"
46     android:layout_marginTop="16dp"
47     android:text="Nome"
48     android:textAppearance="?android:attr/textAppearanceMedium" />
49
50 <ImageButton
51     android:id="@+id/icone_tipo_estabelecimento"
52     android:layout_width="wrap_content"
53     android:layout_height="wrap_content"
54     android:layout_alignLeft="@+id/icone_razao_social"
55     android:layout_alignParentLeft="true"
56     android:layout_alignParentStart="@+id/icone_razao_social"
57     android:layout_below="@+id/label_nome_fantasia_abrir_estabelecimento"
58     android:layout_marginRight="16dp"
59     android:background="@android:color/transparent"
60     android:src="@drawable/ic_stethoscope_grey600_24dp" />
61
62 <TextView
63     android:id="@+id/label_tipo_estabelecimento_abrir_estabelecimento"
64     android:layout_width="wrap_content"
65     android:layout_height="wrap_content"
66     android:layout_alignRight="true"
67     android:layout_below="@+id/label_nome_fantasia_abrir_estabelecimento"
68     android:layout_marginBottom="16dp"
69     android:layout_toRightOf="@+id/icone_tipo_estabelecimento"
-~
```

```
File - E:\Android\Projetos\CadeSaude\app\src\main\res\layout\content_fragment_estabelecimento.xml
71    android:background="@+id/color/transparent"
72    android:text="Tipo
73    android:textAppearance="?android:attr/textAppearanceMedium" />
74
75    <ImageButton
76        android:id="@+id/icon_endereco"
77        android:layout_width="wrap_content"
78        android:layout_height="wrap_content"
79        android:layout_alignParentLeft="true"
80        android:layout_below="@+id/label_tipo_estabelecimento_abrir_estabelecimento"
81        android:layout_marginRight="16dp"
82        android:background="@+id/color/transparent"
83        android:src="@drawable/ic_map_marker_grey600_24dp" />
84
85    <TextView
86        android:id="@+id/label_logradouro_abrir_estabelecimento"
87        android:layout_width="wrap_content"
88        android:layout_height="wrap_content"
89        android:layout_alignLeft="@+id/label_complemento_abrir_estabelecimento"
90        android:layout_alignParentRight="true"
91        android:layout_alignStart="@+id/label_complemento_abrir_estabelecimento"
92        android:layout_alignTop="@+id/icon_endereco"
93        android:layout_marginBottom="4dp"
94        android:text="Logradouro:"/>
95        android:textAppearance="?android:attr/textAppearanceMedium" />
96
97    <TextView
98        android:id="@+id/label_numero_abrir_estabelecimento"
99        android:layout_width="wrap_content"
100        android:layout_height="wrap_content"
101        android:layout_alignLeft="@+id/label_logradouro_abrir_estabelecimento"
102        android:layout_alignParentRight="true"
103        android:layout_alignStart="@+id/label_logradouro_abrir_estabelecimento"
104        android:layout_below="@+id/label_logradouro_abrir_estabelecimento"
105        android:layout_marginBottom="4dp"
```

```
File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\content_fragment_estabelecimento.xml

106      android:text="Número:"  
107      android:textAppearance="?android:attr/textAppearanceMedium" />  
  
108      <TextView  
109          android:id="@+id/label1_complemento_abrir_estabelecimento"  
110          android:layout_width="wrap_content"  
111          android:layout_height="wrap_content"  
112          android:layout_alignLeft="@+id/label1_municipio_abrir_estabelecimento"  
113          android:layout_alignParentRight="true"  
114          android:layout_alignStart="@+id/label1_municipio_abrir_estabelecimento"  
115          android:layout_below="@+id/label1_numero_abrir_estabelecimento"  
116          android:layout_marginBottom="4dp"  
117          android:text="Complemento:"  
118          android:textAppearance="?android:attr/textAppearanceMedium" />  
  
119      <TextView  
120          android:id="@+id/label1_bairro_abrir_estabelecimento"  
121          android:layout_width="wrap_content"  
122          android:layout_height="wrap_content"  
123          android:layout_alignLeft="@+id/label1_complemento_abrir_estabelecimento"  
124          android:layout_alignParentRight="true"  
125          android:layout_alignStart="@+id/label1_complemento_abrir_estabelecimento"  
126          android:layout_below="@+id/label1_complemento_abrir_estabelecimento"  
127          android:layout_marginBottom="4dp"  
128          android:text="Bairro:"  
129          android:textAppearance="?android:attr/textAppearanceMedium" />  
  
130      <TextView  
131          android:id="@+id/label1_cep_abrir_estabelecimento"  
132          android:layout_width="wrap_content"  
133          android:layout_height="wrap_content"  
134          android:layout_alignLeft="@+id/label1_bairro_abrir_estabelecimento"  
135          android:layout_alignParentRight="true"  
136          android:layout_alignStart="@+id/label1_bairro_abrir_estabelecimento"  
137          android:layout_below="@+id/label1_bairro_abrir_estabelecimento"  
138          android:text="Bairro:"  
139          android:textAppearance="?android:attr/textAppearanceMedium" />
```

```
File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\content_fragment_estabelecimento.xml

141      android:layout_marginBottom="4dp"
142      android:text="CEP:"
143      android:textAppearance="?android:attr/textAppearanceMedium" />
144
145  <TextView
146      android:id="@+id/label1_municipio_abrir_estabelecimento"
147      android:layout_width="wrap_content"
148      android:layout_height="wrap_content"
149      android:layout_alignParentRight="true"
150      android:layout_below="@+id/label1_cep_abrir_estabelecimento"
151      android:layout_marginBottom="4dp"
152      android:layout_toEndOf="@+id/icon_endereco"
153      android:layout_toRightOf="@+id/icon_endereco"
154      android:text="Município:"
155      android:textAppearance="?android:attr/textAppearanceMedium" />
156
157  <TextView
158      android:id="@+id/label1_estado_abrir_estabelecimento"
159      android:layout_width="wrap_content"
160      android:layout_height="wrap_content"
161      android:layout_alignParentRight="true"
162      android:layout_below="@+id/label1_municipio_abrir_estabelecimento"
163      android:layout_marginBottom="16dp"
164      android:layout_toEndOf="@+id/icon_endereco"
165      android:layout_toRightOf="@+id/icon_endereco"
166      android:text="Estado:"
167      android:textAppearance="?android:attr/textAppearanceMedium" />
168
169  <ImageButton
170      android:id="@+id/icon_telefone"
171      android:layout_width="wrap_content"
172      android:layout_height="wrap_content"
173      android:layout_alignParentLeft="true"
174      android:layout_below="@+id/label1_estado_abrir_estabelecimento"
175      android:layout_marginRight="16dp"
176
177  <EditText
178      android:id="@+id/edit_endereco"
179      android:layout_width="wrap_content"
180      android:layout_height="wrap_content"
181      android:layout_alignParentLeft="true"
182      android:layout_alignParentTop="true"
183      android:layout_marginBottom="16dp"
184      android:layout_toEndOf="@+id/icon_endereco"
185      android:layout_toRightOf="@+id/icon_endereco"
186      android:padding="10dp"
187      android:scrollbars="vertical"
188      android:text="Endereço" />
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\content\_fragment\_estabelecimento.xml

```
176     android:background="@android:color/transparent"
177     android:paddingBottom="16dp"
178     android:src="@drawable/ic_phone_grey600_24dp" />
179
180     <TextView
181         android:id="@+id/lable1_telefone_abrir_estabelecimento"
182         android:layout_width="wrap_content"
183         android:layout_height="wrap_content"
184         android:layout_alignBottom="@id/icone_telefone"
185         android:layout_below="@id/lable1_estado_abrir_estabelecimento"
186         android:layout_toEndOf="@+id/icone_telefone"
187         android:layout_toRightOf="@+id/icone_telefone"
188         android:text="Telefone:"
189         android:textAppearance="?android:attr/textAppearanceMedium" />
190
191     </RelativeLayout>
192 </ScrollView>
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\content\_main.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5         android:layout_width="match_parent"
6         android:layout_height="match_parent"
7         app:layout_behavior="@string/appbar_scrolling_view_behavior"
8         tools:context="br.com.jucemar_dimon.cadesaude.MainActivity"
9         tools:showIn="@Layout/activity_main">
10
11     <android.support.v7.widget.RecyclerView
12         android:id="@+id/lista_estabelecimentos"
13         android:layout_width="match_parent"
14         android:layout_height="match_parent"
15         android:backgroundTint="@color/divider"
16         android:scrollbars="vertical"
17         android:visibility="gone" />
18
19     <ProgressBar
20         android:id="@+id/barra_progresso_main_activity"
21         android:layout_width="wrap_content"
22         android:layout_height="wrap_content"
23         android:layout_gravity="center"
24         android:visibility="visible" />
25
26     <LinearLayout
27         android:layout_width="wrap_content"
28         android:layout_height="wrap_content"
29         android:layout_gravity="center"
30         android:orientation="vertical">
31
32         <ImageView
33             android:id="@+id/background_Lista_vazia"
34             android:layout_width="wrap_content"
35             android:layout_height="wrap_content"
36             ...>
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\content\_main.xml

```
36     android:layout_gravity="center_horizontal"
37     android:src="@drawable/stethoscope_background"
38     android:visibility="gone" />
39
40     <TextView
41         android:id="@+id/label_lista_sem_favoritos"
42         android:layout_width="wrap_content"
43         android:layout_height="wrap_content"
44         android:layout_gravity="center_horizontal"
45         android:layout_marginTop="30dp"
46         android:text="@string/txt_lista_sem_favoritos"
47         android:textAppearance="@android:style/TextAppearance_Large"
48         android:textColor="#BDBDBD"
49         android:visibility="gone" />
50     </LinearLayout>
51 </FrameLayout>
52
53
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\fragment\_estabelecimento.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <android.support.design.widget.CoordinatorLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:fab="http://schemas.android.com/apk/res-auto"
5     xmlns:tools="http://schemas.android.com/tools"
6         android:id="@+id/asaaas"
7         android:layout_width="match_parent"
8         android:layout_height="match_parent"
9         android:fitsSystemWindows="false">
10
11     <ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
12         android:id="@+id/scrol1_estabelecimento"
13         android:layout_width="match_parent"
14         android:layout_height="match_parent"
15         android:paddingBottom="64dp">
16
17         <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
18             xmlns:tools="http://schemas.android.com/tools"
19             android:layout_width="match_parent"
20             android:layout_height="match_parent"
21             android:padding="@dimen/text_margin">
22
23             <ImageButton
24                 android:id="@+id/icone_razao_social"
25                 android:layout_width="wrap_content"
26                 android:layout_height="wrap_content"
27                 android:layout_marginBottom="16dp"
28                 android:layout_marginRight="16dp"
29                 android:background="@android:color/transparent"
30                 android:src="@drawable/ic_briefcase_grey600_24dp" />
31
32             <TextView
33                 android:id="@+id/label_razao_social_abrir_estabelecimento"
34                 android:layout_width="wrap_content"
35                 android:layout_height="wrap_content"
36             <!--
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\fragment\_estabelecimento.xml

```
36     android:layout_alignParentEnd="true"
37     android:layout_alignParentRight="true"
38     android:layout_alignTop="@+id/icone_razao_social"
39     android:layout_marginBottom="4dp"
40     android:layout_toEndOf="@+id/icone_razao_social"
41     android:layout_toRightOf="@+id/icone_razao_social"
42     android:layout="@+id/icone_razao_social"
43     android:lines="2"
44     android:text="Razão"                                Social:""
45     android:textAppearance="?android:attr/textAppearanceMedium" />
46
47 <TextView
48     android:id="@+id/label1_nome_fantasia_abrir_estabelecimento"
49     android:layout_width="wrap_content"
50     android:layout_height="wrap_content"
51     android:layout_alignLeft="@+id/label1_razao_social_abrir_estabelecimento"
52     android:layout_alignParentRight="true"
53     android:layout_alignStart="@+id/label1_razao_social_abrir_estabelecimento"
54     android:layout_below="@+id/label1_razao_social_abrir_estabelecimento"
55     android:layout_marginBottom="16dp"
56     android:layout_margin="16dp"
57     android:lines="2"
58     android:text="Nome"                                Fantasia:""
59     android:textAppearance="?android:attr/textAppearanceMedium" />
60
61 <ImageButton
62     android:id="@+id/icone_tipo_estabelecimento"
63     android:layout_width="wrap_content"
64     android:layout_height="wrap_content"
65     android:layout_alignLeft="@+id/icone_razao_social"
66     android:layout_alignParentLeft="true"
67     android:layout_alignStart="@+id/icone_razao_social"
68     android:layout_below="@+id/label1_nome_fantasia_abrir_estabelecimento"
69     android:marginRight="16dp"
70     android:background="@android:color/transparent"
```

```
File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\fragment_estabelecimento.xml

71      android:textAppearance="?android:attr/textAppearanceMedium" />
72
73  <TextView
74      android:id="@+id/label1_tipo_estabelecimento_abrir_estabelecimento"
75      android:layout_width="wrap_content"
76      android:layout_height="wrap_content"
77      android:layout_alignParentRight="true"
78      android:layout_below="@+id/label1_nome_fantasia_abrir_estabelecimento"
79      android:layout_marginBottom="16dp"
80      android:layout_toRightOf="@+id/icono_tipo_estabelecimento"
81      android:background="@android:color/transparent"
82      android:text="Tipo"
83      android:textAppearance="?android:attr/textAppearanceMedium" />
84
85  <ImageButton
86      android:id="@+id/icono_endereco"
87      android:layout_width="wrap_content"
88      android:layout_height="wrap_content"
89      android:layout_alignParentLeft="true"
90      android:layout_below="@+id/label1_tipo_estabelecimento_abrir_estabelecimento"
91      android:layout_marginRight="16dp"
92      android:background="@android:color/transparent"
93      android:src="@drawable/ic_map_marker_grey600_24dp" />
94
95  <TextView
96      android:id="@+id/label1_logradouro_abrir_estabelecimento"
97      android:layout_width="wrap_content"
98      android:layout_height="wrap_content"
99      android:layout_alignLeft="@+id/label1_complemento_abrir_estabelecimento"
100     android:layout_alignParentRight="true"
101     android:layout_alignStart="@+id/label1_complemento_abrir_estabelecimento"
102     android:layout_alignTop="@+id/icono_endereco"
103     android:layout_marginBottom="4dp"
104     android:text="Logradouro:"
```

```
106
107 <TextView
108     android:id="@+id/label_numero_abrir_estabelecimento"
109     android:layout_width="wrap_content"
110     android:layout_height="wrap_content"
111     android:layout_alignLeft="@+id/label_logradouro_abrir_estabelecimento"
112     android:layout_alignParentRight="true"
113     android:layout_alignStart="@+id/label_logradouro_abrir_estabelecimento"
114     android:layout_below="@+id/label_logradouro_abrir_estabelecimento"
115     android:layout_marginBottom="4dp"
116     android:text="Número:"
117     android:textAppearance="?android:attr/textAppearanceMedium" />
118
119 <TextView
120     android:id="@+id/label_complemento_abrir_estabelecimento"
121     android:layout_width="wrap_content"
122     android:layout_height="wrap_content"
123     android:layout_alignLeft="@+id/label_municipio_abrir_estabelecimento"
124     android:layout_alignParentRight="true"
125     android:layout_alignStart="@+id/label_municipio_abrir_estabelecimento"
126     android:layout_below="@+id/label_numero_abrir_estabelecimento"
127     android:layout_marginBottom="4dp"
128     android:text="Complemento:"
129     android:textAppearance="?android:attr/textAppearanceMedium" />
130
131 <TextView
132     android:id="@+id/label_bairro_abrir_estabelecimento"
133     android:layout_width="wrap_content"
134     android:layout_height="wrap_content"
135     android:layout_alignLeft="@+id/label_complemento_abrir_estabelecimento"
136     android:layout_alignParentRight="true"
137     android:layout_alignStart="@+id/label_complemento_abrir_estabelecimento"
138     android:layout_below="@+id/label_complemento_abrir_estabelecimento"
139     android:layout_marginBottom="4dp"
140     android:text="Bairro:"
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\fragment\_estabelecimento.xml

```
141                                         android:textAppearance="?android:attr/textAppearanceMedium" />
142
143     <TextView
144         android:id="@+id/label1_cep_abrir_estabelecimento"
145         android:layout_width="wrap_content"
146         android:layout_height="wrap_content"
147         android:layout_alignLeft="@+id/label1_bairro_abrir_estabelecimento"
148         android:layout_alignParentRight="true"
149         android:layout_alignStart="@+id/label1_bairro_abrir_estabelecimento"
150         android:layout_below="@+id/label1_bairro_abrir_estabelecimento"
151         android:layout_marginBottom="4dp"
152         android:text="CEP:"
153         android:textAppearance="?android:attr/textAppearanceMedium" />
154
155     <TextView
156         android:id="@+id/label1_municipio_abrir_estabelecimento"
157         android:layout_width="wrap_content"
158         android:layout_height="wrap_content"
159         android:layout_alignParentRight="true"
160         android:layout_below="@+id/label1_cep_abrir_estabelecimento"
161         android:layout_marginBottom="4dp"
162         android:layout_toEndOf="@+id/icon_endereco"
163         android:layout_toRightOf="@+id/icon_endereco"
164         android:text="Município:"
165         android:textAppearance="?android:attr/textAppearanceMedium" />
166
167     <TextView
168         android:id="@+id/label1_estado_abrir_estabelecimento"
169         android:layout_width="wrap_content"
170         android:layout_height="wrap_content"
171         android:layout_alignParentRight="true"
172         android:layout_below="@+id/label1_municipio_abrir_estabelecimento"
173         android:layout_marginBottom="16dp"
174         android:layout_toEndOf="@+id/icon_endereco"
175         android:layout_toRightOf="@+id/icon_endereco"
176
177     <--
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\fragment\_estabelecimento.xml

```
176     android:text="Estado:"  
177     android:textAppearance="?android:attr/textAppearanceMedium" />  
178  
179     <ImageButton  
180         android:id="@+id/icone_telefone"  
181         android:layout_width="wrap_content"  
182         android:layout_height="wrap_content"  
183         android:layout_alignParentLeft="true"  
184         android:layout_below="@+id/label1_estado_abrir_estabelecimento"  
185         android:layout_marginRight="16dp"  
186         android:background="@android:color/transparent"  
187         android:paddingBottom="16dp"  
188         android:src="@drawable/ic_phone_grey600_24dp" />  
189  
190     <TextView  
191         android:id="@+id/label1_telefone_abrir_estabelecimento"  
192         android:layout_width="wrap_content"  
193         android:layout_height="wrap_content"  
194         android:layout_alignBottom="@+id/icone_telefone"  
195         android:layout_below="@+id/label1_estado_abrir_estabelecimento"  
196         android:layout_toEndOf="@+id/icone_telefone"  
197         android:layout_toRightOf="@+id/icone_telefone"  
198         android:text="Telefone:"  
199         android:textAppearance="?android:attr/textAppearanceMedium" />  
200     </RelativeLayout>  
201 </ScrollView>  
202  
203 <com.github.clans.fab.FloatingActionButton  
204     android:id="@+id/fab_add_favorito"  
205     android:layout_width="wrap_content"  
206     android:layout_height="wrap_content"  
207     android:layout_gravity="right|bottom"  
208     android:layout_marginBottom="72dp"  
209     android:layout_marginRight="16dp"  
... -
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\fragment\_estabelecimento.xml

```
211     android:src="@drawable/ic_star_outline_white_24dp"
212     app:fab_colorNormal="#DA4336"
213     app:fab_colorPressed="#E75043"
214     app:fab_colorRipple="#99FFFFFF"
215     app:fab_hideAnimation="@anim/fab_scale_down"
216     app:fab_label="Busca" por parâmetros"
217     app:fab_shadowColor="#66000000"
218     app:fab_shadowRadius="4dp"
219     app:fab_shadowXOffset="1dp"
220     app:fab_shadowYOffset="3dp"
221     app:fab_showAnimation="@anim/fab_scale_up"
222     app:fab_showShadow="true"
223     anno:fab_size="normal" />
224
225 </android.support.design.widget.CoordinatorLayout>
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\fragment\_listas.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:id="@+id/view_vazio"
4     android:layout_width="match_parent"
5     android:layout_height="match_parent"
6     android:layout_marginBottom="50dp"
7     android:orientation="vertical">
8
9     <ListView
10        android:id="@+id/list_view"
11        android:layout_width="match_parent"
12        android:layout_height="match_parent"
13        android:layout_marginBottom="50dp"
14        android:listSelector="@android:color/transparent" />
15
16     </LinearLayout>
17
18
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\item\_lista\_estabelecimentos.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="match_parent"
4     android:layout_height="wrap_content"
5     android:layout_marginTop="2dp"
6     android:background="@drawable/selector"
7     android:orientation="vertical"
8     android:paddingBottom="8dp"
9     android:paddingLeft="16dp"
10    android:paddingRight="8dp"
11    android:paddingTop="8dp">
12
13     <TextView
14         android:id="@+id/nome_fantasia_lista_busca"
15         android:layout_width="match_parent"
16         android:layout_height="wrap_content"
17         android:layout_alignParentLeft="true"
18         android:layout_alignParentStart="true"
19         android:layout_marginBottom="5dp"
20         android:layout_toLeftOf="@+id/flag_favorito"
21         android:layout_toStartOf="@+id/flag_favorito"
22         android:gravity="center_vertical"
23         android:lines="2"
24         android:text="Nome"
25         android:textAppearance="@color/orange"
26         android:textColor="@color/medium_blue" />
27
28     <TextView
29         android:id="@+id/tipo_estab_lista_busca"
30         android:layout_width="match_parent"
31         android:layout_height="wrap_content"
32         android:layout_alignParentStart="true"
33         android:layout_below="@+id/nome_fantasia_lista_busca"
34         android:layout_marginBottom="5dp" />
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\item\_lista\_estabelecimentos.xml

```
36     android:layout_toLeftOf="@+id/Flag_fone"
37     android:layout_toStartOf="@+id/Flag_fone"
38     android:gravity="center_vertical"
39     android:lines="2"
40     android:text="Tipo de estabelecimento"
41     android:textAppearance="?android:attr/textAppearanceMedium"
42     android:textColor="@color/secondary_text" />
43
44     <TextView
45         android:id="@+id/bairro_lista_busca"
46         android:layout_width="wrap_content"
47         android:layout_height="wrap_content"
48         android:layout_alignEnd="@+id/tipo_estab_Lista_busca"
49         android:layout_alignParentLeft="true"
50         android:layout_alignParentStart="true"
51         android:layout_alignRight="@+id/tipo_estab_Lista_busca"
52         android:layout_below="@+id/tipo_estab_Lista_busca"
53         android:layout_marginBottom="5dp"
54         android:text="Bairro"
55         android:textAppearance="?android:attr/textAppearanceSmall"
56         android:textColor="@color/secondary_text" />
57
58     <TextView
59         android:id="@+id/municipio_lista_busca"
60         android:layout_width="wrap_content"
61         android:layout_height="wrap_content"
62         android:layout_alignParentLeft="true"
63         android:layout_alignParentStart="true"
64         android:layout_below="@+id/bairro_lista_busca"
65         android:text="Cidade"
66         android:textAppearance="?android:attr/textAppearanceSmall"
67         android:textColor="@color/secondary_text" />
68
69     <TextView
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\item\_lista\_estabelecimentos.xml

```
71     android:id="@+id/distancia_Lista_busca"
72     android:layout_width="wrap_content"
73     android:layout_height="wrap_content"
74     android:layout_alignBaseline="@+id/municipio_lista_busca"
75     android:layout_alignBottom="@+id/municipio_lista_busca"
76     android:layout_toLeftOf="@+id/flag_gps"
77     android:layout_toStartOf="@+id/flag_gps"
78     android:textAppearance="?android:attr/textAppearanceSmall" />
79
80     <ImageButton
81         android:id="@+id/flag_favorito"
82         android:layout_width="wrap_content"
83         android:layout_height="wrap_content"
84         android:layout_alignParentEnd="true"
85         android:layout_alignParentRight="true"
86         android:layout_alignTop="@+id/nome_fantasia_Lista_busca"
87         android:layout_centerHorizontal="true"
88         android:background="@android:color/transparent"
89         android:padding="16dp"
90         android:src="@drawable/ic_star_outline_grey600_24dp"
91         android:visibility="visible" />
92
93     <ImageButton
94         android:id="@+id/flag_fone"
95         android:layout_width="wrap_content"
96         android:layout_height="wrap_content"
97         android:layout_alignLeft="@+id/flag_favorito"
98         android:layout_alignStart="@+id/flag_favorito"
99         android:layout_centerVertical="true"
100        android:background="16dp"
101        android:src="@drawable/ic_phone_grey600_24dp"
102        android:visibility="visible" />
103
104    ...
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\item\_lista\_estabelecimentos.xml

```
106     android:id="@+id/flag_gps"
107     android:layout_width="wrap_content"
108     android:layout_height="wrap_content"
109     android:layout_alignLeft="@+id/flag_fone"
110     android:layout_alignStart="@+id/flag_fone"
111     android:layout_below="@+id/tipo_estab_lista_busca"
112     android:background="@android:color/transparent"
113     android:padding="16dp"
114     android:src="@drawable/ic_crosshairs_gps_grey600_24dp"
115     android:visibility="visible" />
116 
117 </RelativeLayout>
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\item\_listas.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="match_parent"
4     android:layout_height="wrap_content"
5     android:gravity="center_vertical"
6     android:orientation="vertical"
7     android:padding="16dp">
8
9     <TextView
10        android:id="@+id/item_linha_1"
11        android:layout_width="match_parent"
12        android:layout_height="wrap_content"
13        android:clickable="false"
14        android:text="Medium Text"
15        android:textAppearance="?attr/textAppearanceMedium" />
16
17     <TextView
18        android:id="@+id/item_linha_2"
19        android:layout_width="match_parent"
20        android:layout_height="wrap_content"
21        android:clickable="false"
22        android:text="Small Text"
23        android:textAppearance="?attr/textAppearanceSmall"
24        android:textStyle="italic" />
25 </LinearLayout>
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\item\_spinner.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2
3 <TextView xmlns:android="http://schemas.android.com/apk/res/android"
4   android:id="@+id/item_spinner"
5   android:layout_width="match_parent"
6   android:layout_height="wrap_content"
7   android:gravity="center_vertical"
8   android:paddingBottom="16dp"
9   android:paddingLeft="8dp"
10  android:paddingRight="8dp"
11  android:paddingTop="16dp"
12  android:textAppearance="?android:attr/textAppearanceSmall"
13  android:textColor="@color/primary_text" />
14
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\preference\_info.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="wrap_content"
4     android:layout_height="wrap_content"
5     android:layout_margin="16dp"
6     android:background="@color/primary_light"
7     android:gravity="center"
8     android:orientation="vertical">
9
10    <ImageView
11        android:id="@+id/icone_pref_info"
12        android:layout_width="wrap_content"
13        android:layout_height="wrap_content"
14        android:layout_gravity="center_horizontal"
15        android:layout_marginBottom="16dp"
16        android:layout_marginTop="32dp"
17        android:src="@drawable/ic_alert_grey600_48dp" />
18
19    <TextView
20        android:id="@+id/msg_pref_info"
21        android:layout_width="wrap_content"
22        android:layout_height="wrap_content"
23        android:layout_gravity="center_horizontal"
24        android:layout_marginBottom="32dp"
25        android:layout_marginLeft="16dp"
26        android:layout_marginRight="16dp"
27        android:text="@string/msg_info_preferencias"
28        android:textAlignment="center"
29        android:textAppearance="@style/TextAppearance.Small" />
30
31 </LinearLayout>
```

File - E:\Android\Projetos\CadDESaude\app\src\main\res\layout\preference\_seekbar.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="match_parent"
4     android:layout_height="match_parent"
5     android:gravity="center_vertical"
6     android:orientation="vertical">
7
8     <TextView
9         style="?android:textAppearanceMedium"
10        android:layout_width="match_parent"
11        android:layout_height="wrap_content"
12        android:layout_marginLeft="8dp"
13        android:layout_marginTop="8dp"
14        android:text="@string/distancia_estabelecimentos_titulo"
15        android:textColor="?android:attr/textColorSecondary" />
16
17     <TextView
18         style="?android:textAppearanceSmall"
19        android:layout_width="match_parent"
20        android:layout_height="wrap_content"
21        android:layout_marginBottom="8dp"
22        android:layout_marginLeft="8dp"
23        android:text="@string/distancia_title_seekbar"
24        android:textColor="?android:attr/textColorSecondary" />
25
26     <LinearLayout
27         android:layout_width="match_parent"
28         android:layout_height="wrap_content"
29         android:layout_marginBottom="16dp" />
30
31     <SeekBar
32         android:id="@+id/seekbar"
33         android:layout_width="0dp"
34         android:layout_height="wrap_content"
35         android:layout_weight="85" />
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\preference\_seekbar.xml

```
36
37     <TextView
38         android:id="@+id/id(summary"
39         android:layout_width="0dp"
40         android:layout_height="wrap_content"
41
42         android:layout_weight="15" />
43     </LinearLayout>
44 </LinearLayout>
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\view\_dialog\_busca\_geral.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="match_parent"
4     android:layout_height="match_parent"
5     android:orientation="vertical"
6     android:padding="16dp">
7
8     <TextView
9         android:layout_width="match_parent"
10        android:layout_height="wrap_content"
11        android:layout_margin="16dp"
12        android:text="@string/tipo_busca" />
13
14     <Spinner
15         android:id="@+id/comboBox_tipo_busca_geral_dialog"
16         android:layout_width="match_parent"
17         android:layout_height="wrap_content"
18         android:layout_margin="16dp"
19         android:spinnerMode="dialog" />
20
21     <Spinner
22         android:id="@+id/campo_busca_geral_dialog"
23         android:layout_width="match_parent"
24         android:layout_height="wrap_content"
25         android:layout_margin="16dp"
26         android:spinnerMode="dialog"
27         android:visibility="gone" />
28
29     <EditText
30         android:id="@+id/campo_busca_geral_dialog"
31         android:layout_width="match_parent"
32         android:layout_height="wrap_content"
33         android:layout_margin="16dp"
34         android:hint="Digite      0      termo"
35         android:visibility="gone" />
36
37
```

File - E:\Android\Projetos\CadeSaude\app\src\main\res\layout\view\_dialog\_busca\_geral.xml

36  
37 </LinearLayout>

File - E:\Android\Projetos\CadEsaude\app\src\main\res\menu\menu\_consulta\_gps.xml

```
1 <menu xmlns:android="http://schemas.android.com/apk/res/android"
2   xmlns:app="http://schemas.android.com/apk/res-auto"
3   xmlns:tools="http://schemas.android.com/tools"
4   tools:context="br.com.jucemar.dimon.cadesaude.EstablecimientoActivity">
5
6   <item
7     android:id="@+id/action_search"
8     android:icon="@drawable/ic_magnify_white_24dp"
9     android:iconifiedByDefault="true"
10    android:orderInCategory="100"
11    android:title="search"
12    android:visible="false"
13    app:actionViewClass="android.support.v7.widget.SearchView"
14    android:showAsAction="always|collapseActionView" />
15
16   <item
17     android:id="@+id/action_gps"
18     android:icon="@drawable/ic_crosshairs_gps_white_24dp"
19     android:orderInCategory="100"
20     android:title="GPS"
21     android:visible="false"
22     app:showAsAction="always" />
23
24 </menu>
```

```
File - E:\Android\Projetos\CadEsaude\app\src\main\res\menu_main.xml
1 <menu xmlns:android="http://schemas.android.com/apk/res/android"
2
3   xmlns:app="http://schemas.android.com/apk/res-auto"
4   xmlns:tools="http://schemas.android.com/tools"
5   tools:context="br.com.jucemar_dimon.cadesaude.MainActivity">
6
7   <item
8     android:id="@+id/action_configuracoes"
9     android:orderInCategory="100"
10    android:title="@string/action_configuracoes"
11    app:showAsAction="never" />
12
13   <item
14     android:id="@+id/action_ajuda"
15     android:orderInCategory="100"
16     android:title="@string/action_Ajuda"
17     app:showAsAction="never" />
18
19   <item
20     android:id="@+id/action_sobre"
21     android:orderInCategory="100"
22     android:title="@string/action_sobre"
23     app:showAsAction="never" />
24
25   <item
26     android:id="@+id/action_search"
27     android:icon="@drawable/ic_magnify_white_24dp"
28     android:iconifiedByDefault="true"
29     android:orderInCategory="100"
30     android:title="search"
31     app:actionViewClass="android.support.v7.widget.SearchView"
32     app:showAsAction="always|collapseActionView" />
33
34 </menu>
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\menu\menu\_primeira\_configuracao.xml

```
1 <menu xmlns:android="http://schemas.android.com/apk/res/android"
2   xmlns:app="http://schemas.android.com/apk/res-auto"
3   xmlns:tools="http://schemas.android.com/tools"
4   tools:context="br.com.jucemar.dimon.cadesaude.EstablecimentoActivity">
5
6   <item
7     android:id="@+id/action_salvar_config"
8     android:icon="@drawable/ic_check_white_24dp"
9     android:orderInCategory="100"
10    android:title="Salvar"
11    app:showAsAction="always" />
12 </menu>
13
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\values\colors.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <resources>
3   <color name="primary">#4CAF50</color>
4   <color name="primary_dark">#388E3C</color>
5   <color name="primary_light">#C8E6C9</color>
6   <color name="accent">#8bc34a</color>
7   <color name="primary_text">#212121</color>
8   <color name="secondary_text">#727272</color>
9   <color name="icons">#FFFFFF</color>
10  <color name="divider">#B6B6B6</color>
11  <color name="black_overlay">#66000000</color>
12  <color name="fab_normal">#E53935</color>
13  <color name="fab_pressed">#FF5252</color>
14 </resources>
15
```

```

1 <resources>
2   <string name="app_name">CadEsaude</string>
3
4   <!-- Valores para as opções de menu principal-->
5   <string name="action_configuracoes">Configurações</string>
6   <string name="action_sobre">Sobre</string>
7   <string name="action_Ajuda">Ajuda</string>
8   <string name="titulo_config">Configurações</string>
9   <string name="titulo_ajuda">Ajuda</string>
10  <string name="titulo_sobre">Sobre</string>
11
12  <!-- Valores para as preferências do aplicativo-->
13  <string name="pref_municipios_preferidos_key">municípios_preferidos</string>
14  <string name="pref_distancia_estabelecimentos_key">distância_estabelecimentos</string>
15  <string name="pref_info_key">info</string>
16  <string name="pref_estabelecimentos_favoritos_key">estabelecimentos_favoritos</string>
17  <string name="pref_estabelecimentos_favoritos_file">favoritos</string>
18  <string name="municipios_preferidos_título">Municípios
19  <string name="distancia_estabelecimentos_título">Distância
20  <string name="municipios_preferidos_subtítulo">Não há municípios selecionados
21  <string name="municipios_dialog_title">seleccione
22  <string name="distancia_title_seekbar">Selecione a distância máxima</string>
23  <string name="municipios_preferidos_default" />
24
25  <string-array name="codigos_municípios">
26    <item>420005</item>
27    <item>420010</item>
28    <item>420020</item>
29    <item>420030</item>
30    <item>420040</item>
31    <item>420050</item>
32    <item>420055</item>
33    <item>420060</item>
34    <item>420070</item>
35

```

```
36      <item>420075</item>
37      <item>420080</item>
38      <item>420090</item>
39      <item>420100</item>
40      <item>420110</item>
41      <item>420120</item>
42      <item>420125</item>
43      <item>420127</item>
44      <item>420130</item>
45      <item>420140</item>
46      <item>420150</item>
47      <item>420160</item>
48      <item>420165</item>
49      <item>420170</item>
50      <item>420180</item>
51      <item>420190</item>
52      <item>420195</item>
53      <item>420205</item>
54      <item>420200</item>
55      <item>420207</item>
56      <item>421280</item>
57      <item>422000</item>
58      <item>420208</item>
59      <item>420209</item>
60      <item>420210</item>
61      <item>420213</item>
62      <item>420215</item>
63      <item>420220</item>
64      <item>420230</item>
65      <item>420240</item>
66      <item>420243</item>
67      <item>420250</item>
68      <item>420253</item>
69      <item>420257</item>
```

```
71 <item>420245</item>
72 <item>420270</item>
73 <item>420280</item>
74 <item>420285</item>
75 <item>420287</item>
76 <item>420290</item>
77 <item>420300</item>
78 <item>420310</item>
79 <item>420315</item>
80 <item>420320</item>
81 <item>420330</item>
82 <item>420340</item>
83 <item>420350</item>
84 <item>420360</item>
85 <item>420370</item>
86 <item>420380</item>
87 <item>420325</item>
88 <item>420390</item>
89 <item>420395</item>
90 <item>420400</item>
91 <item>420410</item>
92 <item>420415</item>
93 <item>420417</item>
94 <item>420419</item>
95 <item>420420</item>
96 <item>420425</item>
97 <item>420430</item>
98 <item>420435</item>
99 <item>420440</item>
100 <item>420445</item>
101 <item>420455</item>
102 <item>420450</item>
103 <item>420460</item>
104 <item>420470</item>
```

```
106    <item>420480</item>
107    <item>420490</item>
108    <item>420500</item>
109    <item>420510</item>
110    <item>420515</item>
111    <item>420517</item>
112    <item>420519</item>
113    <item>420520</item>
114    <item>420530</item>
115    <item>420535</item>
116    <item>420540</item>
117    <item>420543</item>
118    <item>420545</item>
119    <item>420550</item>
120    <item>420555</item>
121    <item>420560</item>
122    <item>420570</item>
123    <item>420580</item>
124    <item>420590</item>
125    <item>420600</item>
126    <item>420610</item>
127    <item>420620</item>
128    <item>420630</item>
129    <item>420640</item>
130    <item>420650</item>
131    <item>420660</item>
132    <item>420665</item>
133    <item>420670</item>
134    <item>420675</item>
135    <item>420680</item>
136    <item>420690</item>
137    <item>420700</item>
138    <item>420710</item>
139    <item>420720</item>
... -----
```

```
141 <item>420740</item>
142 <item>420750</item>
143 <item>420757</item>
144 <item>420760</item>
145 <item>420765</item>
146 <item>420768</item>
147 <item>420770</item>
148 <item>420775</item>
149 <item>420780</item>
150 <item>420785</item>
151 <item>420790</item>
152 <item>420800</item>
153 <item>420810</item>
154 <item>420820</item>
155 <item>420830</item>
156 <item>420840</item>
157 <item>420845</item>
158 <item>420850</item>
159 <item>420860</item>
160 <item>420870</item>
161 <item>420880</item>
162 <item>420890</item>
163 <item>420895</item>
164 <item>420900</item>
165 <item>420910</item>
166 <item>420915</item>
167 <item>420917</item>
168 <item>420920</item>
169 <item>420930</item>
170 <item>420940</item>
171 <item>420945</item>
172 <item>420950</item>
173 <item>420960</item>
174 <item>420970</item>
```

```
176 <item>420985</item>
177 <item>420990</item>
178 <item>421000</item>
179 <item>421003</item>
180 <item>421005</item>
181 <item>421010</item>
182 <item>421020</item>
183 <item>421030</item>
184 <item>421040</item>
185 <item>421050</item>
186 <item>421055</item>
187 <item>421060</item>
188 <item>421070</item>
189 <item>421080</item>
190 <item>421085</item>
191 <item>421090</item>
192 <item>421100</item>
193 <item>421105</item>
194 <item>421110</item>
195 <item>421120</item>
196 <item>421125</item>
197 <item>421130</item>
198 <item>421140</item>
199 <item>421145</item>
200 <item>421150</item>
201 <item>421160</item>
202 <item>421165</item>
203 <item>421170</item>
204 <item>421175</item>
205 <item>421180</item>
206 <item>421185</item>
207 <item>421187</item>
208 <item>421189</item>
209 <item>421190</item>
```

```
211 <item>421205</item>
212 <item>421210</item>
213 <item>421220</item>
214 <item>421223</item>
215 <item>421225</item>
216 <item>421227</item>
217 <item>421230</item>
218 <item>421240</item>
219 <item>421250</item>
220 <item>421260</item>
221 <item>421265</item>
222 <item>421270</item>
223 <item>421290</item>
224 <item>421300</item>
225 <item>421310</item>
226 <item>421315</item>
227 <item>421320</item>
228 <item>421330</item>
229 <item>421335</item>
230 <item>421340</item>
231 <item>421350</item>
232 <item>421360</item>
233 <item>421370</item>
234 <item>421380</item>
235 <item>421390</item>
236 <item>421400</item>
237 <item>421410</item>
238 <item>421415</item>
239 <item>421420</item>
240 <item>421430</item>
241 <item>421440</item>
242 <item>421450</item>
243 <item>421460</item>
244 <item>421480</item>
```

```
246 <item>421490</item>
247 <item>421500</item>
248 <item>421505</item>
249 <item>421507</item>
250 <item>421510</item>
251 <item>421520</item>
252 <item>421530</item>
253 <item>421535</item>
254 <item>421540</item>
255 <item>421545</item>
256 <item>421550</item>
257 <item>421555</item>
258 <item>421560</item>
259 <item>421565</item>
260 <item>421567</item>
261 <item>421568</item>
262 <item>421569</item>
263 <item>421570</item>
264 <item>421580</item>
265 <item>421575</item>
266 <item>421590</item>
267 <item>421600</item>
268 <item>421605</item>
269 <item>421610</item>
270 <item>421620</item>
271 <item>421630</item>
272 <item>421635</item>
273 <item>421625</item>
274 <item>421640</item>
275 <item>421650</item>
276 <item>421660</item>
277 <item>421670</item>
278 <item>421680</item>
279 <item>421690</item>
... --
```

```
281    <item>421710</item>
282    <item>421715</item>
283    <item>421720</item>
284    <item>421725</item>
285    <item>421730</item>
286    <item>421740</item>
287    <item>421750</item>
288    <item>421755</item>
289    <item>421760</item>
290    <item>421770</item>
291    <item>421775</item>
292    <item>421780</item>
293    <item>421790</item>
294    <item>421795</item>
295    <item>421800</item>
296    <item>421810</item>
297    <item>421820</item>
298    <item>421825</item>
299    <item>421830</item>
300    <item>421835</item>
301    <item>421840</item>
302    <item>421850</item>
303    <item>421860</item>
304    <item>421870</item>
305    <item>421875</item>
306    <item>421880</item>
307    <item>421885</item>
308    <item>421890</item>
309    <item>421895</item>
310    <item>421900</item>
311    <item>421910</item>
312    <item>421915</item>
313    <item>421917</item>
314    <item>421920</item>
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\values\strings.xml

```
316   <item>421935</item>
317   <item>421940</item>
318   <item>421950</item>
319   <item>421960</item>
320   <item>421970</item>
321   <item>421985</item>
322 </string-array>
323
324 <string-array name="nomes_municípios_view">
325   <item>Abdon Batista</item>
326   <item>Abelardo Luz</item>
327   <item>Agrolândia</item>
328   <item>Agronômica</item>
329   <item>Água Doce</item>
330   <item>Águas de Chapecó</item>
331   <item>Águas Mornas</item>
332   <item>Alfredo Wagner</item>
333   <item>Alto Bela Vista</item>
334   <item>Anchieta</item>
335   <item>Angelina</item>
336   <item>Anita Garibaldi</item>
337   <item>Anitápolis</item>
338   <item>Antônio Carlos</item>
339   <item>Apiaúna</item>
340   <item>Arabutã</item>
341   <item>Araquari</item>
342   <item>Araranguá</item>
343   <item>Armazém</item>
344   <item>Arroio Trinta</item>
345   <item>Arvoredo</item>
346   <item>Ascurra</item>
347   <item>Atalanta</item>
348   <item>Aurora</item>
349   <item>Ra Inéario Arroio do Silvânia</item>
- - -
```

```
351   <item>BaInéário Barra do Sul</item>
352   <item>BaInéário Camboriú</item>
353   <item>BaInéario Gaivota</item>
354   <item>BaInéario Piçarras</item>
355   <item>BaInéario Rincão</item>
356   <item>Bandeirante</item>
357   <item>Barra Bonita</item>
358   <item>Barra Velha</item>
359   <item>Bela Vista do Toldo</item>
360   <item>Belmonte</item>
361   <item>Bento Novo</item>
362   <item>Biguaçu</item>
363   <item>Blumenau</item>
364   <item>Bocaina do Sul</item>
365   <item>Bom Jardim da Serra</item>
366   <item>Bom Jesus</item>
367   <item>Bom Jesus do Oeste</item>
368   <item>Bom Retiro</item>
369   <item>Bombinhas</item>
370   <item>Botuverá</item>
371   <item>Braço do Norte</item>
372   <item>Braço do Trombudo</item>
373   <item>Brunópolis</item>
374   <item>Brusque</item>
375   <item>Caçador</item>
376   <item>Caibi</item>
377   <item>Calmon</item>
378   <item>Camboriú</item>
379   <item>Campo Alegre</item>
380   <item>Campo Belo do Sul</item>
381   <item>Campo Erê</item>
382   <item>Campos Novos</item>
383   <item>Canelinha</item>
384   <item>Canoinhas</item>
384   <item>Capão Alto</item>
-- --
```

```
386 <item>Capinzal</item>
387 <item>Capivari de Baixo</item>
388 <item>Catanduvas</item>
389 <item>Caxambu do Sul</item>
390 <item>CeIsso Ramos</item>
391 <item>Cerro Negro</item>
392 <item>Chapadão do Lageado</item>
393 <item>Chapecó</item>
394 <item>Cocal do Sul</item>
395 <item>Concórdia</item>
396 <item>Cordilheira Alta</item>
397 <item>Coronel Freitas</item>
398 <item>Coronel Martins</item>
399 <item>Correia Pinto</item>
400 <item>Corupá</item>
401 <item>Criciúma</item>
402 <item>Cunha Porã</item>
403 <item>Cunhataí</item>
404 <item>Curitibanos</item>
405 <item>Descanso</item>
406 <item>Dionísio Cerqueira</item>
407 <item>Dona Emma</item>
408 <item>Doutor Pedrinho</item>
409 <item>Entre Rios</item>
410 <item>Ermo</item>
411 <item>Ervá Velho</item>
412 <item>Faxinal dos Guedes</item>
413 <item>Flor do Sertão</item>
414 <item>Florianópolis</item>
415 <item>Formosa do Sul</item>
416 <item>Forquilhainha</item>
417 <item>Fraiburgo</item>
418 <item>Frei Rogério</item>
419 <item>Galvão</item>
... <item>Caranaba</item>
```

```
421 <item>Garuva</item>
422 <item>Gaspar</item>
423 <item>Governador Celso Ramos</item>
424 <item>Grão Pará</item>
425 <item>Gravatal</item>
426 <item>Guabiruba</item>
427 <item>Guaraciaba</item>
428 <item>Guaramirim</item>
429 <item>Guarujá do Sul</item>
430 <item>Guatambú</item>
431 <item>Herval d'Oeste</item>
432 <item>Ibiama</item>
433 <item>Ibicaré</item>
434 <item>Ibirama</item>
435 <item>Içara</item>
436 <item>Ilhota</item>
437 <item>Imaruí</item>
438 <item>Imbituba</item>
439 <item>Imbuia</item>
440 <item>Indaiá I</item>
441 <item>Iomerê</item>
442 <item>Ipira</item>
443 <item>Iporã do Oeste</item>
444 <item>Ipuáçu</item>
445 <item>Ipumirim</item>
446 <item>Iraceminha</item>
447 <item>Irani</item>
448 <item>Irati</item>
449 <item>Irineópolis</item>
450 <item>Itá</item>
451 <item>Itaiópolis</item>
452 <item>Itajaí</item>
453 <item>Itapema</item>
454 <item>Itapiranga</item>
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\values\strings.xml

```
456   <item>Itaporanga</item>
457   <item>Jaborá</item>
458   <item>Jacinto Machado</item>
459   <item>Jaguaruna</item>
460   <item>Jaraguá do Sul</item>
461   <item>Jardinópolis</item>
462   <item>Joaçaba</item>
463   <item>Joinville</item>
464   <item>José Boiteux</item>
465   <item>Jupiá</item>
466   <item>Lacerdópolis</item>
467   <item>Lages</item>
468   <item>Laguna</item>
469   <item>Lajeado Grande</item>
470   <item>Laurentino</item>
471   <item>Lauro Müller</item>
472   <item>Lebon Régis</item>
473   <item>Leoberto Leal</item>
474   <item>Lindóia do Sul</item>
475   <item>Lontras</item>
476   <item>Luiz Alves</item>
477   <item>Luzerna</item>
478   <item>Macieira</item>
479   <item>Mafra</item>
480   <item>Major Gercino</item>
481   <item>Major Vieira</item>
482   <item>Maracajá</item>
483   <item>Maravilha</item>
484   <item>Marema</item>
485   <item>Massaranduba</item>
486   <item>Matos Costa</item>
487   <item>Meleiro</item>
488   <item>Mirim Doce</item>
489   <item>Modelo</item>
... 
```

491 <item>Monte Carlo</item>  
492 <item>Monte Castelo</item>  
493 <item>Morro da Fumaça</item>  
494 <item>Morro Grande</item>  
495 <item>Morro Navegantes</item>  
496 <item>Nova Erechim</item>  
497 <item>Nova Itaberaba</item>  
498 <item>Nova Trento</item>  
499 <item>Nova Veneza</item>  
500 <item>Novo Horizonte</item>  
501 <item>Orleans</item>  
502 <item>Otacílio Costa</item>  
503 <item>Ouro</item>  
504 <item>Ouro Verde</item>  
505 <item>Paiá</item>  
506 <item>Paineiras</item>  
507 <item>Palhoça</item>  
508 <item>Palma Sola</item>  
509 <item>Palmeira</item>  
510 <item>Palmitos</item>  
511 <item>Papanduva</item>  
512 <item>Paraíso</item>  
513 <item>Passo de Torres</item>  
514 <item>Passos Maia</item>  
515 <item>Paulo Lopes</item>  
516 <item>Pedras Grandes</item>  
517 <item>Penha</item>  
518 <item>Peritiba</item>  
519 <item>Pescaria Brava</item>  
520 <item>Petrolândia</item>  
521 <item>Pinhalzinho</item>  
522 <item>Pinheiro Preto</item>  
523 <item>Piratuba</item>  
524 <item>Planalto Alegre</item>

```
526 <item>Ponte Alta</item>
527 <item>Ponte Alta do Norte</item>
528 <item>Ponte Serrada</item>
529 <item>Porto Belo</item>
530 <item>Porto União</item>
531 <item>Pouso Redondo</item>
532 <item>Praia Grande</item>
533 <item>Presidente Castello Branco</item>
534 <item>Presidente Getúlio</item>
535 <item>Presidente Nereu</item>
536 <item>Princesa</item>
537 <item>Quilombo</item>
538 <item>Rancho Queimado</item>
539 <item>Rio das Antas</item>
540 <item>Rio do Campo</item>
541 <item>Rio do Oeste</item>
542 <item>Rio do Sul</item>
543 <item>Rio dos Cedros</item>
544 <item>Rio Fortuna</item>
545 <item>Rio Negrinho</item>
546 <item>Rio Rufino</item>
547 <item>Riqueza</item>
548 <item>Rodeio</item>
549 <item>Salete</item>
550 <item>Saltinho</item>
551 <item>Salto Veloso</item>
552 <item>Sangão</item>
553 <item>Santa Cecília</item>
554 <item>Santa Helena</item>
555 <item>Santa Rosa de Lima</item>
556 <item>Santa Rosa do Sul</item>
557 <item>Santa Terezinha</item>
558 <item>Santa Terezinha do Progresso</item>
559 <item>Santiago do Sul</item>
...
```

```
561    <item>Santo Amaro da Imperatriz</item>
562    <item>São Bento do Sul</item>
563    <item>São Bernardo</item>
564    <item>São Bonifácio</item>
565    <item>São Carlos</item>
566    <item>São Cristovão do Sul</item>
567    <item>São Domingos</item>
568    <item>São Francisco do Sul</item>
569    <item>São João Batista</item>
570    <item>São João do Itaperiú</item>
571    <item>São João do Oeste</item>
572    <item>São João do Sul</item>
573    <item>São Joaquim</item>
574    <item>São José</item>
575    <item>São José do Cedro</item>
576    <item>São José do Cerrito</item>
577    <item>São Lourenço do Oeste</item>
578    <item>São Ludgero</item>
579    <item>São Martinho</item>
580    <item>São Miguel da Boa Vista</item>
581    <item>São Miguel do Oeste</item>
582    <sup>i+em</sup><item>São Pedro de Alcântara</item>
583    <item>Saudades</item>
584    <item>Schroeder</item>
585    <item>Seara</item>
586    <item>Serra Alta</item>
587    <item>Siderópolis</item>
588    <item>Sombrio</item>
589    <item>Sul Brasil</item>
590    <item>Taió</item>
591    <item>Tangará</item>
592    <item>Tigrinhos</item>
593    <item>Tijucas</item>
594    <item>Timbé do Sul</item>
595    <item>Timbó</item>
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\values\strings.xml

```
596   <item>Timbó Grande</item>
597   <item>Três Barras</item>
598   <item>Treviso</item>
599   <item>Treze de Maio</item>
600   <item>Treze Tílias</item>
601   <item>Trombudo Central</item>
602   <item>Tubarão</item>
603   <item>Tunápolis</item>
604   <item>Turvo</item>
605   <item>União do Oeste</item>
606   <item>Urubici</item>
607   <item>Urupema</item>
608   <item>Urussanga</item>
609   <item>Vargeão</item>
610   <item>Vargem</item>
611   <item>Vargem Bonita</item>
612   <item>Vidal Ramos</item>
613   <item>Videira</item>
614   <item>Vitor Meireles</item>
615   <item>Witmarsum</item>
616   <item>Xanxerê</item>
617   <item>Xavantina</item>
618   <item>Xaxim</item>
619   <item>Zortéa</item>
620   <!-- ... -->
621   <string-array name="nomes_municípios_bd">
622   <item>ABDON BATISTA</item>
623   <item>ABELARDO LUZ</item>
624   <item>AGROLANDIA</item>
625   <item>AGRONOMICA</item>
626   <item>AGUA DOCE</item>
627   <item>AGUAS DE CHAPECO</item>
628   <item>AGUAS FRIAS</item>
629   <item>MORNAS</item>
-->
```

```
631 <item>ALFREDO WAGNER</item>
632 <item>ALTO BELA VISTA</item>
633 <item>ANCHIETA</item>
634 <item>ANGELINA</item>
635 <item>ANITA GARIBOLDI</item>
636 <item>ANITAPOLIS</item>
637 <item>ANTONIO CARLOS</item>
638 <item>APIUNA</item>
639 <item>ARABUTA</item>
640 <item>ARAQUARI</item>
641 <item>ARARANGUA</item>
642 <item>ARMAZEM</item>
643 <item>ARROIO TRINTA</item>
644 <item>ARVOREDO</item>
645 <item>ASCURRA</item>
646 <item>ATALANTA</item>
647 <item>AUTORA</item>
648 <item>BALNEARIO ARROIO DO SILVA</item>
649 <item>BALNEARIO BARRA DO SUL</item>
650 <item>BALNEARIO CAMBORIU</item>
651 <item>BALNEARIO GAIOTA</item>
652 <item>BALNEARIO PICARRAS</item>
653 <item>BALNEARIO RINCAO</item>
654 <item>BANDEIRANTE</item>
655 <item>BARRA BONITA</item>
656 <item>BARRA VELHA</item>
657 <item>BELA VISTA DO TOLDO</item>
658 <item>BELMONTE</item>
659 <item>BENEDITO NOVO</item>
660 <item>BIGUACU</item>
661 <item>BLUMENAU</item>
662 <item>BOCAINA DO SUL</item>
663 <item>BOM JARDIM DA SERRA</item>
664 <item>BOM JESUS DO OESTE</item>
665 ...
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\values\strings.xml

```
666 <item>BOM RETIRO</item>
667 <item>BOMBINHAS</item>
668 <item>BOTUVERA</item>
669 <item>BRACO DO NORTE</item>
670 <item>BRACO DO TROMBUDO</item>
671 <item>BRUNOPOLIS</item>
672 <item>BRUSQUE</item>
673 <item>CACADOR</item>
674 <item>CAIBI</item>
675 <item>CALMON</item>
676 <item>CAMBORIU</item>
677 <item>CAMPO ALEGRE</item>
678 <item>CAMPO BELO DO SUL</item>
679 <item>CAMPO ERE</item>
680 <item>CAMPOS NOVOS</item>
681 <item>CANELINHA</item>
682 <item>CANOINHAS</item>
683 <item>CAPAO ALTO</item>
684 <item>CAPINZAI</item>
685 <item>CAPIVARI DE BAIIXO</item>
686 <item>CATANDUVAS</item>
687 <item>CAXAMBU DO SUL</item>
688 <item>CELSO RAMOS</item>
689 <item>CERRO NEGRO</item>
690 <item>CHAPADAO DO LAGEADO</item>
691 <item>CHAPECO</item>
692 <item>COCAL DO SUL</item>
693 <item>CONCORDIA</item>
694 <item>CORDILHEIRA ALTA</item>
695 <item>CORONEL FREITAS</item>
696 <item>CORONEL MARTINS</item>
697 <item>CORREIA PINTO</item>
698 <item>CRICIUMA</item>
699 <item>CUNHA PORA</item>
--
```

```
701 <item>CUNHATAI</item>
702 <item>CURITIBANOS</item>
703 <item>DESCANSO</item>
704 <item>DIONISIO CERQUEIRA</item>
705 <item>DONA EMMA</item>
706 <item>DOUTOR PEDRINHO</item>
707 <item>ENTRE RIOS</item>
708 <item>ERMO</item>
709 <item>ERVAL VELHO</item>
710 <item>FAXINAL DOS GUEDES</item>
711 <item>FLOR DO SERTAO</item>
712 <item>FLORIANOPOLIS</item>
713 <item>FORMOSA DO SUL</item>
714 <item>FORQUILHINHA</item>
715 <item>FRAIBURGO</item>
716 <item>FREI ROGERIO</item>
717 <item>GALVAO</item>
718 <item>GAROPABA</item>
719 <item>GARUVA</item>
720 <item>GASPAR</item>
721 <item>GOVERNADOR CELSO RAMOS</item>
722 <item>GRAO PARA</item>
723 <item>GRAVATAL</item>
724 <item>GUABIRUBA</item>
725 <item>GUARACIABA</item>
726 <item>GUARAMIRIM</item>
727 <item>GUARUJA DO SUL</item>
728 <item>GUATAMBU</item>
729 <item>HERVAL D OESTE</item>
730 <item>IBIAM</item>
731 <item>IBICARE</item>
732 <item>IBIRAMA</item>
733 <item>ICARA</item>
734 <item>ILHOTA</item>
```

```
736 <item>IMBITUBA</item>
737 <item>IMBUIA</item>
738 <item>INDIATL</item>
739 <item>HOMERE</item>
740 <item>IPIRA</item>
741 <item>IPORA DO OESTE</item>
742 <item>IPUACU</item>
743 <item>IPUMIRIM</item>
744 <item>IRACEMINHA</item>
745 <item>IRANI</item>
746 <item>IRATI</item>
747 <item>IRINEOPOLIS</item>
748 <item>ITA</item>
749 <item>ITAIOPOLIS</item>
750 <item>ITAJAI</item>
751 <item>ITAPEMA</item>
752 <item>ITAPIRANGA</item>
753 <item>ITAPOA</item>
754 <item>ITUPORANGA</item>
755 <item>JABORA</item>
756 <item>JACINTO MACHADO</item>
757 <item>JAGUARUNA</item>
758 <item>JARAGUA DO SUL</item>
759 <item>JARDINOPOLIS</item>
760 <item>JOACABA</item>
761 <item>JOINVILLE</item>
762 <item>JOSE BOITEUX</item>
763 <item>JUPIA</item>
764 <item>LACERDOPOLIS</item>
765 <item>LAGES</item>
766 <item>LAGUNA</item>
767 <item>LAJEADO GRANDE</item>
768 <item>LAURENTINO</item>
769 <item>LAURO MULLER</item>
--
```

```
771 <item>LEOBERTO LEAL</item>
772 <item>LINDOIA DO SUL</item>
773 <item>LONTRAS</item>
774 <item>LUIZ ALVES</item>
775 <item>LUZERNA</item>
776 <item>MACIEIRA</item>
777 <item>MAFRA</item>
778 <item>MAJOR GERCINO</item>
779 <item>MAJOR VIEIRA</item>
780 <item>MARACAJA</item>
781 <item>MARAVilha</item>
782 <item>MAREMA</item>
783 <item>MASSARANDUBA</item>
784 <item>MATOS COSTA</item>
785 <item>MELEIRO</item>
786 <item>MIRIM DOCE</item>
787 <item>MODELO</item>
788 <item>MONDAY</item>
789 <item>MONTE CARLO</item>
790 <item>MONTE CASTELO</item>
791 <item>MORRO DA FUMACA</item>
792 <item>MORRO GRANDE</item>
793 <item>NAVAGANTES</item>
794 <item>NOVA ERECHIM</item>
795 <item>NOVA ITABERABA</item>
796 <item>NOVA TRENTO</item>
797 <item>NOVA VENEZA</item>
798 <item>NOVO HORIZONTE</item>
799 <item>ORLEANS</item>
800 <item>OTACILIO COSTA</item>
801 <item>OURO</item>
802 <item>OURO VERDE</item>
803 <item>PAINEL</item>
804 <item>PALHOCA</item>
--
```

```
806 <item>PALMA SOLA</item>
807 <item>PALMEIRA</item>
808 <item>PALMITOS</item>
809 <item>PAPANDUVA</item>
810 <item>PARAISO</item>
811 <item>PASSO DE TORRES</item>
812 <item>PASSOS MAIA</item>
813 <item>PAULO LOPES</item>
814 <item>PEDRAS GRANDES</item>
815 <item>PENHA</item>
816 <item>PERITIBA</item>
817 <item>PESCARIA BRAVA</item>
818 <item>PETROLANDIA</item>
819 <item>PINHAL ZINHO</item>
820 <item>PINHEIRO PRETO</item>
821 <item>PIRATUBA</item>
822 <item>PLANALTO ALEGRE</item>
823 <item>POMERODE</item>
824 <item>PONTE ALTA</item>
825 <item>PONTE ALTA DO NORTE</item>
826 <item>PONTE SERRADA</item>
827 <item>PORTO BELO</item>
828 <item>PORTO UNIAO</item>
829 <item>POUZO REDONDO</item>
830 <item>PRAIA GRANDE</item>
831 <item>PRESIDENTE CASTELLO BRANCO</item>
832 <item>PRESIDENTE GETULIO</item>
833 <item>PRESIDENTE NEREU</item>
834 <item>PRINCESA</item>
835 <item>QUILOMBO</item>
836 <item>RANCHO QUEIMADO</item>
837 <item>RIO DAS ANTAS</item>
838 <item>RIO DO CAMPO</item>
839 <item>RIO DO OESTE</item>
840 <item>RIO DO SUL</item>
... 
```

```
841 <item>RIO DOS CEDROS</item>
842 <item>RIO FORTUNA</item>
843 <item>RIO NEGRINHO</item>
844 <item>RIO RUFINO</item>
845 <item>RIQUEZA</item>
846 <item>RODEIO</item>
847 <item>ROMELANDIA</item>
848 <item>SALETE</item>
849 <item>SALTINHO</item>
850 <item>SALTO VELOSO</item>
851 <item>SANGAO</item>
852 <item>SANTA CECILIA</item>
853 <item>SANTA HELENA</item>
854 <item>SANTA ROSA DE LIMA</item>
855 <item>SANTA ROSA DO SUL</item>
856 <item>SANTA TEREZINHA</item>
857 <item>SANTA TEREZINHA DO PROGRESSO</item>
858 <item>SANTIAGO DO SUL</item>
859 <item>SANTO AMARO DA IMPERATRIZ</item>
860 <item>SAO BENTO DO SUL</item>
861 <item>SAO BERNARDINO</item>
862 <item>SAO BONIFACIO</item>
863 <item>SAO CARLOS</item>
864 <item>SAO CRISTOVAO DO SUL</item>
865 <item>SAO DOMINGOS</item>
866 <item>SAO FRANCISCO DO SUL</item>
867 <item>SAO JOAO BATTISTA</item>
868 <item>SAO JOAO DO ITAPERIU</item>
869 <item>SAO JOAO DO OESTE</item>
870 <item>SAO JOAQUIM</item>
871 <item>SAO JOSE</item>
872 <item>SAO JOSE DO CEDRO</item> JOSE
873 <item>SAO DO CERRITO</item>
874 <item>SAO LOURENCO DO OESTE</item>
- - -
```

```
876 <item>SAO JUDGERO</item>
877 <item>SAO MARTINHO</item>
878 <item>SAO MIGUEL DA BOA VISTA</item>
879 <item>SAO MIGUEL DO OESTE</item>
880 <item>SAO PEDRO DE ALCANTARA</item>
881 <item>SAUDADES</item>
882 <item>SCHROEDER</item>
883 <item>SEARA</item>
884 <item>SERRA ALTA</item>
885 <item>SIDEROPOLIS</item>
886 <item>SOMBARIO</item>
887 <item>SUL BRASIL</item>
888 <item>TAIO</item>
889 <item>TANGARA</item>
890 <item>TIGRINHOS</item>
891 <item>TIJUCAS</item>
892 <item>TIMBE DO SUL</item>
893 <item>TIMBO</item>
894 <item>TIMBO GRANDE</item>
895 <item>TRES BARRAS</item>
896 <item>TREVISO</item>
897 <item>TREZE DE MAIO</item>
898 <item>TREZE TILHAS</item>
899 <item>TROMBUDO CENTRAL</item>
900 <item>TUBARAO</item>
901 <item>TUNAPOLIS</item>
902 <item>TURVO</item>
903 <item>UNIAO DO OESTE</item>
904 <item>URUBICI</item>
905 <item>URUPEMA</item>
906 <item>URUSSANGA</item>
907 <item>VARGEO</item>
908 <item>VARGEM BONITA</item>
909 <item>VARGEM DOURADA</item>
```

File - E:\Android\Projetos\CadESaude\app\src\main\res\values\strings.xml

```
911 <item>VIDEIRA</item>
912 <item>VITOR MEIRELES</item>
913 <item>WITMARSUM</item>
914 <item>XANXERE</item>
915 <item>XAVANTINA</item>
916 <item>XAXIM</item>
917 <item>ZORTEA</item>
918 </string-array>
919
920 <string-array name="tipos_etabElecimento">
921 <item>CENTRAL DE NOTIFICACAO, CAPTACAO E DISTRIB DE ORGAOS ESTADUAIS</item>
922 <item>CENTRO DE ATENCAO HEMOTERAPIA E OU HEMATOLOGICA</item>
923 <item>CENTRO DE APOIO A SAUDE DA FAMILIA</item>
924 <item>CENTRO DE ATENCAO PSICOSSOCIAL</item>
925 <item>CENTRO DE PARTO NORMAL – ISOLADO</item>
926 <item>CENTRO DE SAUDE/UNIDADE BASICA</item>
927 <item>CLINICA/CENTRO DE ESPECIALIDADE</item>
928 <item>CONSULTORIO ISOLADO</item>
929 <item>HOSPITAL ESPECIALIZADO</item>
930 <item>HOSPITAL GERAL</item>
931 <item>HOSPITAL/DIA – ISOLADO</item>
932 <item>LABORATORIO CENTRAL DE SAUDE PUBLICA LACEN</item>
933 <item>LABORATORIO DE SAUDE PUBLICA</item>
934 <item>POLICLINICA</item>
935 <item>POSTO DE SAUDE</item>
936 <item>PRONTO SOCORRO ESPECIALIZADO</item>
937 <item>PRONTO SOCORRO GERAL</item>
938 <item>UNIDADE DE APOIO DIAGNOSE E TERAPIA (SADT ISOLADO)</item>
939 <item>UNIDADE MISTA</item>
940 </string-array>
941
942 <string-array name="tipos_atendimento">
943 <item>AMBULATORIAL</item>
944 <item>INTERNACAO</item>
... 
```

```

946   <item>SADT</item>
947   <item>URGENCIA</item>
948
949   </string-array>
950
951   <string name="title_activity_estabelecimento">EstabelecimentoActivity</string>
952   <string name="action_settings">Settings</string>
953   <string name="txt_lista_sem_favoritos">Você não possui favoritos</string>
954
955   <!-- Strings de BuscaGpsActivity -->
956   <string name="fab_filtro_tipo_estabelecimento">Tipo de estabelecimento</string>
957   <string name="fab_filtro_tipo_atendimento">Tipo de atendimento</string>
958   <string name="fab_filtro_tipo_convenio">Tipo de convênio</string>
959   <string name="fab_filtro_tipo_servico">Tipo de serviço</string>
960   <string name="fab_filtro_tipo_profissional">Tipo de profissional</string>
961   <string name="fab_filtro_limpar_todos">Limpar filtros</string>
962   <string name="txt_erro_gps">Verifique seu GPS</string>
963   <string name="titulo_busca_gps_activity">Consulta por proximidade</string>
964
965   <!-- String FiltroGeralDialog -->
966   <string name="ttulo_dialog_filtro">Selecione os itens a serem filtrados</string>
967
968   <!-- Strings de BuscaGpsActivity SnackBar-->
969   <string name="tentar_novamente_action">TENTAR NOVAMENTE</string>
970   <string name="tentar_novamente_label">Localização obrigatória</string>
971
972   <string name="tipos_convenio">
973     <item>PARTICULAR</item>
974     <item>PLANO / PROPRIO</item>
975     <item>PLANO / SEGUNDO TERCEIRO</item>
976     <item>PLANO DE SAUDE PRIVADO</item>
977     <item>PLANO DE SAUDE PUBLICO</item>
978   </string-array>
979
...

```

File - E:\Android\Projetos\CadESaude\app\src\main\res\values\strings.xml

```
981 <string name="title_activity_busca_geral">Busca geral</string>
982 <string name="txt_erro_busca_geral">Nenhum resultado localizado</string>
983
984 <!-- BuscaGeralActivity -->
985 <string name="label_tipo_busca">Selecione um tipo de busca:</string>
986 <string name="titulo_dialog_buscar_geral">Busca geral</string>
987 <string-array name="tipos_buscas">
988 <item>Selecione uma opção</item>
989 <item>Tipos de convênios</item>
990 <item>Formas de atendimento</item>
991 <item>Nome do profissional</item>
992 <item>Especialidades</item>
993 <item>Nome do estabelecimento</item>
994 <item>Serviços oferecidos</item>
995 </string-array>
996
997 <!-- SobreActivity -->
998 <string name="appNome">CadESaúde</string>
999 <string name="appDesc">Aplicativo desenvolvido por Jucemar Dímon como trabalho de conclusão do curso de Sistemas
1000 de Informações da Universidade Federal de Santa Catarina(UFSC). Este aplicativo tem como
1001 objetivo não somente a aplicação dos conhecimentos adquiridos ao longo do curso como também a criação de uma
1002 ferramenta para auxiliar o cidadão catarinense na busca por estabelecimentos de saúde na sua região.</string>
1003 <string name="appVersao">Versão do aplicativo</string>
1004 <string name="bdVersao">Versão do Banco de dados</string>
1005 <string name="title_activity_primeiros_passos">PrimeirosPassosActivity</string>
1006 <string name="dummy_button">Configurações</string>
1007 <string name="dummy_content">Seja bem vindo ao aplicativo CadESaúde.\n Neste seu primeiro acesso você deverá
1008 clicar no botão configurações e selecionar os municípios que deseja incluir nas buscas por estabelecimentos de saúde.
1009 </string>
1010 <!-- AjudaActivity - Tela inicial -->
1011 <string name="cabecalho_ajuda">Neste menu do aplicativo você poderá tirar algumas dúvidas sobre termos
1012 específicos que são utilizados no sistema de cadastros de estabelecimentos de saúde do DATASUS.</string>
1013 <string name="label_btntipos_estabelecimentos">Tipos de estabelecimentos</string>
1014 <string name="label_btntipos_servicos">Tipos de serviços existentes</string>
```

```

1011
1012 <!-- AjudaActivity - Tipos de estabelecimentos -->
1013 <string name="titulo_estabelecimentos">Tipos de estabelecimentos</string>
1014
1015 <string name="posto_de_saude_titulo">Posto de Saúde</string>
1016 <string name="posto_de_saude">Unidade destinada à prestação de assistência a uma determinada população, de forma
1017 programada ou não, por profissional de nível médio, com a presença intermitente ou não do profissional médico.</
1018 string>
1019 <string name="centro_saude_titulo">Centro de Saúde/Unidade Básica de Saúde</string>
1020 > de forma programada ou não, nas especialidades básicas, podendo oferecer assistência odontológica e de outros
1021 profissionais de nível superior. A assistência deve ser permanente e prestada por médico generalista ou especialista
1022 nestas áreas. Podendo ou não oferecer: SADT e Pronto atendimento 24 Horas.</string>
1023
1024 <string name="policlinica_titulo">Policlínica</string>
1025 <string name="policlinica">Unidade de saúde para prestação de atendimento ambulatorial em várias especialidades,
1026 incluindo ou não as especialidades básicas, podendo ainda oferecer outras especialidades não médicas. Podendo ou não
1027 oferecer: SADT e Pronto atendimento 24 Horas.</string>
1028
1029 <string name="hospital_geral_titulo">Hospital Geral</string>
1030 <string name="hospital_geral">Hospital destinado à prestação de atendimento nas especialidades básicas, por
1031 especialistas e/ou outras especialidades médicas. Pode dispor de serviço de Urgência/Emergência. Deve dispor também
1032 de SADT de média complexidade. Podendo Ter ou não SIPAC.</string>
1033
1034 <string name="hospital_especializado_titulo">Hospital Especializado</string>
1035 <string name="hospital_especializado">Hospital destinado à prestação de assistência à saúde em uma única
1036 especialidade/área. Pode dispor de serviço de Urgência/Emergência e SADT. Podendo Ter ou não SIPAC. Geralmente de
1037 referência regional, macro regional ou estadual.</string>
1038
1039 <string name="unidade_mista_titulo">Unidade Mista</string>
1040 <string name="unidade_mista">Unidade de saúde básica destinada à prestação de atendimento em atenção básica e
1041 integral à saúde, de forma programada ou não, nas especialidades básicas, podendo oferecer assistência odontológica e
1042 de outros profissionais, com unidade de internação, sob administração única. A assistência médica deve ser
1043 permanente e prestada por médico especialista ou generalista. Pode dispor de urgência/emergência e SADT básico ou de

```

```

1031 rotina.</string>
1032
1033 <string name="pronto_socorro_geral_titulo">Pronto Socorro Geral</string>
1034 <string name="pronto_socorro_geral">Unidade destinada à prestação de assistência a pacientes com ou sem risco de
1035 vida, cujos agravos necessitam de atendimento imediato. Podendo ter ou não internação.</string>
1036
1037 <string name="pronto_socorro_especializado_titulo">Pronto Socorro Geral</string>
1038 <string name="pronto_socorro_especializado">Unidade destinada à prestação de assistência em uma ou mais
1039 especialidades, a pacientes com ou sem risco de vida, cujos agravos necessitam de atendimento imediato.</string>
1040
1041 <string name="consultorio_isolado_titulo">Consultório Isolado</string>
1042 <string name="consultorio_isolado">Sala isolada destinada à prestação de assistência médica ou odontológica ou
1043 de outros profissionais de saúde de nível superior.</string>
1044
1045 <string name="sadt_titulo">Unidade de Serviço de Apoio de Diagnose e Terapia (SADT ISOLADO)</string>
1046 <string name="sadt">Unidades isoladas onde são realizadas atividades que auxiliam a determinação de diagnóstico
1047 e/ou complementam o tratamento e a reabilitação do paciente.</string>
1048
1049 <string name="parto_isolado_titulo">Centro de Parto Normal Isolado</string>
1050 <string name="parto_isolado">Unidade intra-hospitalar ou isolada, especializada no atendimento da mulher no
1051 período puerperal.</string>
1052
1053 <string name="hospital_dia_titulo">Hospital/Dia - Isolado</string>
1054 <string name="labo_saude_publica_titulo">Laboratório Central de Saúde Pública</string>
1055 <string name="labo_saude_publica">Estabelecimento de Saúde que integra o Sistema Nacional de Laboratórios de
1056 Saúde Pública - SISLAB, em conformidade com normalização vigente.</string>
1057 <string name="centro_hematologica_titulo">Centro de Atenção Hemoterapica e ou Hematologica</string>

```

```

1058 <string name="centro_hemato">Estabelecimento que realiza o ciclo do sangue, desde a captação do doador, processamento, testes sorológicos, testes imunohematológicos, distribuição e transfusão de sangue de maneira total ou parcial.</string>
1059
1060 <string name="caps_titulo">Centro De Atenção Psicossocial (CAPS)</string>
1061 <string name="caps">Entende-se por Centro de Atenção Psicossocial o tipo de estabelecimento isolado com as especificações seguintes: Estabelecimento de saúde de esfera administrativa pública com serviços de saúde mental abertos e comunitários do Sistema Único de Saúde SUS destinados a prestar atenção diária às pessoas que sofrem com transtornos mentais severos e persistentes, em sua área territorial, em regime de tratamento intensivo, semi-intensivo e não-intensivo. Devem oferecer cuidados clínicos e de reabilitação psicossocial e têm como objetivo substituir o modelo hospitalocêntrico, evitar internações, favorecer o exercício da cidadania e da inclusão social dos usuários e de suas famílias, e executar as atribuições de supervisão e regulamentação de rede de serviços de saúde mental, quando de natureza pública.</string>
1062
1063 <string name="upa_titulo">Pronto Atendimento</string>
1064 <string name="upa">Estabelecimento autônomo não-hospitalar, que integra a Rede de Atenção às Urgências e Emergências, destinado à assistência aos pacientes acometidos por quadros de urgência e emergência, realizando o atendimento inicial, estabilizando o paciente e definindo a necessidade ou não de encaminhamento a serviços de maior complexidade.</string>
1065
1066 <!-- AjudaActivity - Serviços especializados -->
1067 <string name="titulo_actvt_ajuda_servicos">Tipos de serviços</string>
1068 <string name="101c1">Atenção à Doença Renal Crônica</string>
1069 <string name="101c2">Tratamento Dialítico-Hemodialise; Confecção/Intervenção de Acessos para Diálise; Tratamento Nefrologia em Geral; Tratamento Diálitico Peritoneal; Tratamento em DRC</string>
1070
1071 <string name="102c1">Atenção Às Pessoas com Doenças Raras</string>
1072 <string name="102c2">Atenção Especializada Em Doenças Raras; Referência em Doenças Raras</string>
1073
1074 <string name="103c1">Atenção às Pessoas em Situação de Violência Sexual</string>
1075 <string name="103c2">Referência para Atenção Integral às Pessoas em Situação de Violência Sexual; Atenção Integral à Saúde de Adolescentes em Situação de Violência Sexual; Atenção Integral à Saúde de Crianças em Situação de Violência Sexual; Atenção Integral à Saúde de Homens em Situação de Violência Sexual; Atenção Integral à Saúde de Pessoas Idosas em Situação de Violência Sexual; Interrupção de Gravidez nos Casos Previstos em Lei; Atenção
1076

```

```
1076 Ambulatorial às Pessoas em Situação de Violência Sexual;Coleta de Vestígios de Violência Sexual</string>
1077
1078 <string name="104c1">Atenção Básica</string>
1079 <string name="104c2">Atenção Básica; Atenção Básica com Saúde Bucal</string>
1080
1081 <string name="105c1">Atenção Especializada no Processo Transsexualizador</string>
1082 <string name="105c2">Acompanhamento Clínico, pré e pós operatório e hormonoterapia; Cirurgia e Acompanhamento
1083 pré e pós ororatório</string>
1084 <string name="106c1">Atendimento Itinerante de Assistência e Ensino em Saúde</string>
1085 <string name="106c2">ODONTOLOGIA - PROJETO SAÚDE NA ESCOLA (PSE); ODONTOLOGIA - PROJETO BRASIL ALFABETIZADO (CPBA
1086 ); OFTALMOLOGIA - PROJETO SAÚDE NA ESCOLA (PSE); OFTALMOLOGIA - PROJETO BRASIL ALFABETIZADO (PBA)</string>
1087 <string name="107c1">Cirurgia Vascular</string>
1088 <string name="107c2">Fístula arteriovenosa sem enxerto; Fístula arteriovenosa com enxerto</string>
1089 <string name="108c1">Diagnóstico por Imagem</string>
1090 <string name="108c2">Radiologia; Ultra-sonografia; Tomografia computadorizada; Ressonância magnética; Radiologia
1091 intervencionista; Radiologia por telemedicina; Ultra-sonografia por telemedicina; Tomografia computadorizada por
1092 telemedicina;Ressonância magnética por telemedicina; Radiologia intervencionista por telemedicina; Mamografia;
1093 Mamografia por Telemedicina</string>
1094 <string name="109c1">Estratégia de Saúde da Família</string>
1095 <string name="109c2">Saúde da família; Saúde bucal MI; Saúde bucal MII</string>
1096 <string name="110c1">Estratégias de Agentes Comunitário de Saúde</string>
1097 <string name="110c2">Estratégia de Agentes Comunitários de Saúde - EACS; Estratégia de Agentes Comunitários de
1098 Saúde com Saúde Bucal - (modalidade I); Estratégia de Agentes Comunitários de Saúde com Saúde Bucal - (modalidade II)
1099 <string name="111c1">Hospital-Dia</string>
1100 <string name="111c2">Saúde mental; AIDS; Geriátrico; Fibrose cística; Cirúrgico / Diagnóstico; Acompanhamento pós
1101 -transplante de medula óssea</string>
1102 <string name="112c1">Medicina Nuclear</string>
```

```

1103 <string name="112c2">Medicina nuclear in vivo; Medicina nuclear in vitro; Medicina nuclear in vivo por
1104 telemedicina</string>
1105 <string name="113c1">Regulação Assistencial dos Serviços de Saúde</string>
1106 <string name="113c2">Central de Regulação de Internações Hospitalares; Central de Regulação de Consultas e Exames;
1107 Central de Regulação de Urgências; Unidade de autorizações de TFD interestaduais; Unidade de autorizações de TFD
1108 interessenciais; Central Estadual de Regulação da Alta Complexidade</string>
1109 <string name="114c1">Serviço de Apoio a Saúde da Família</string>
1110 <string name="114c2">NASF 1; NASF 2</string>
1111 <string name="115c1">Serviço de Atenção a Adué do Trabalhador</string>
1112 <string name="115c2">Atendimento/acompanhamento em saúde do trabalhador; Centro de referência em saúde do
1113 trabalhador CEREST estadual; VIGILÂNCIA EM SAÚDE DO TRABALHADOR (VISAT)</string>
1114 <string name="116c1">Serviço de Atenção a DST/HIV/AIDS</string>
1115 <string name="116c2">Testagem e aconselhamento; Diagnóstico e tratamento; Diagnóstico, tratamento e capacitação
1116 técnica</string>
1117 <string name="117c1">Serviço de Atenção à Obesidade</string>
1118 <string name="117c2">Tratamento clínico, cirúrgico, reparador e acompanhamento ao obeso</string>
1119 <string name="118c1">Serviço de Atenção a Saúde Auditiva</string>
1120 <string name="118c2">Serviço de Atenção à Saúde Auditiva na Média Complexidade; Serviço de Atenção à Saúde
1121 Auditiva na Alta Complexidade; Terapia fonoaudiológica; Diagnóstico em audiology/otologia; Implante Coclear; Triagem
1122 Auditiva Neonatal; Diagnóstico em Audiologia/Otologia po Telemédicina; Atenção Especializada às Pessoas com
1123 Deficiência Auditiva</string>
1124 <string name="119c1">Serviço de Atenção á Saúde das Pessoas Ostomizadas</string>
1125 <string name="119c2">Atenção as Pessoas Ostomizadas I; Atenção as Pessoas Ostomizadas II</string>
1126 <string name="120c1">Serviço de Atenção a Saúde dos Adolescentes em Conflito com a Lei, em Regime de Internação e
1127 Internação Provisória</string>
1128 <string name="120c2">Unidade sócio-educativa com até 40 adolescentes; Unidade sócio-educativa com de 41 a 100
adolescentes; Unidade sócio-educativa com de 101 a 180 adolescentes; Unidade sócio-educativa com mais de 180

```

```
1127 adolescentes</string>
1128
1129 <string name="121c1">Serviço de Atenção a Saúde no Sistema Penitenciário</string>
1130 <string name="121c2">Atendimento em presídio até 100 presos; Atendimento em manicômio até 100 presos;
1131 Atendimento em presídio acima de 100 presos; Atendimento em manicômio acima de 100 presos</string>
1132
1133 <string name="122c1">Serviço de Atenção a Saúde Reprodutiva</string>
1134 <string name="122c2">Atenção à infertilidade; Contracepção clínica; Laqueadura; Vasectomia</string>
1135
1136 <string name="123c1">Serviço de Atenção ao Paciente com Tubercolose</string>
1137 <string name="123c2">Diagnóstico e tratamento multidrogas resistentes</string>
1138
1139 <string name="124c1">Serviço de Atenção ao Pré-Natal, Parto e Nascimento</string>
1140 <string name="124c2">Acompanhamento do pré-natal de risco habitual; Acompanhamento do pré-natal de alto risco;
1141 Parto em gestação de risco habitual; Parto em gestação de alto risco; Centro de parto normal; Casa da Gestante, Bebê
1142 e Puérpera</string>
1143
1144 <string name="125c1">Serviço de Atenção Cardiovascular/Cardiologia</string>
1145 <string name="125c2">Eletrofisiologia; Cirurgia cardiovascular (Pediatrício);
1146 Cirurgia vascular; Cardiologia intervencionista (Hemodinâmica); Cardiologia Endovascular Extracardíaco; Cardiologia
1147 Clínica</string>
1148
1149 <string name="126c1">Serviço de Atenção Domiciliar</string>
1150 <string name="126c2">Assistência Domiciliar; Internação Domiciliar de Atenção Domiciliar
1151 - EMAD; Equipe Multidisciplinar de Apoio - EMAP</string>
1152
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\values\strings.xml

```
1153 <string name="129c1">Serviço de Atenção em Urologia</string>
1154 <string name="129c2">Urologia em Geral; Litotripsia</string>
1155
1156 <string name="130c1">Serviço de Atenção Integral em Hanseniae</string>
1157 <string name="130c2">Serviço de Atenção Integral em Hanseniae Tipo I, Serviço de Atenção Integral em Hanseniae
1158 Tipo II, Serviço de Atenção Integral em Hanseniae Tipo III</string>
1159 <string name="131c1">Serviço de Atenção Psicossocial</string>
1160 <string name="131c2">Residência terapêutica em saúde mental; Atendimento psicossocial; Serviço Hospitalar para
1161 Atenção a Saúde Mental; Serviço Residencial Terapêutico Tipo I - SRT tipo I; Serviço Residencial Terapêutico Tipo II
1162 - SRT tipo II; UA AdulTo; UA InfantoJuvenil; Unidade de Atenção em Regime Residencial</string>
1163 <string name="132c1">Serviço de Atendimento Móvel de Urgências</string>
1164 <string name="132c2">Ambulância de transporte; Unidade de suporte básico de vida terrestre (USB); Unidade de
1165 suporte avançado de vida terrestre (USA); Unidade de Suporte Básico de vida: Equipe Embarcação; Veículos de
1166 intervenção rápida; Outros veículos; Ambulância de resgate; Motolâncias; Unidade de Suporte Avançado de vida: Equipe
1167 Embarcação; Equipe Aeromédico</string>
1168 <string name="133c1">Serviço de Banco de Tecidos</string>
1169 <string name="133c2">Banco de valvula cardíaca humana; Banco de tecido músculo esquelético</string>
1170
1171 <string name="134c1">Serviço de Cirurgia Reparadora</string>
1172 <string name="134c2">Tratamento da Lipodistrofia do portador de HIV / AIDS; Tratamento em queimados; Tratamento
1173 da Lipopatofia facial do portador de HIV / AIDS</string>
1174 <string name="135c1">Serviço de Cirurgia Reparadora</string>
1175 <string name="135c2">Cirurgia torácica</string>
1176 <string name="136c1">Serviço de Controle de Tabagismo</string>
1177 <string name="136c2">Abordagem e tratamento do fumante</string>
1178 <string name="137c1">Serviço de Cuidados Intermédios</string>
1179 <string name="137c2">NEONATAL CONVENCIONAL; NEONATAL CANGURU; PEDIÁTRICO; ADULTO</string>
1180 <string name="138c1">Serviço de Diagnóstico por Anatomia Patológica e/ou Citopatologia</string>
```

```

1181   <string name="138c2">Exames anatomo-patológicos; Exames citopatológicos; Monitoramento Externo da Qualidade dos
1182   Exames Citopatológicos do Colo do Útero</string>
1183   <string name="139c1">Serviço de Diagnóstico por Laboratório Clínico</string>
1184   <string name="139c2">Exames Bioquímicos; Exames hematológicos e hemostasia; Exames sorológicos e imunológicos;
1185   Exames coprológicos; Exames de uroanálise; Exames hormonais; Exames de vigilância epidemiológica e ambiental;
1186   Exames toxicológicos ou de monitorização terapêutica; Exames microbiológicos; Exames em outros líquidos biológicos;
1187   Exames de genética; Exames para triagem neonatal; Exames imuno-hematológicos</string>
1188   <string name="140c1">Serviço de Diagnóstico por Métodos Gráficos/Dinâmicos</string>
1189   <string name="140c2">Teste Ergométrico; Teste de Holter; Exame Eletrocardiográfico; Exame Eletroencefalográfico;
1190   Teste Ergométrico por Telemedicina; Teste Holter por Telemedicina; Exame Eletrocardiográfico por Telemedicina; Exame
1191   Eletroencefalografia por Telemedicina; Exame Eletroneuromiografia; Exame video-eletroencefalografia; Potenciais
1192   Evocados; Potenciais Evocados por Telemedicina</string>
1193   <string name="141c1">Serviço de Dispensação de Órteses, Próteses e Materiais Especiais</string>
1194   <string name="141c2">OPM auxiliares de locomoção; OPM ortopédicas; OPM auditivas; OPM oftalmológicas; OPM em
1195   gastroenterologia; OPM em urologia; OPM em odontologia; OPM buco-maxilo-facial; Substituição/Troca em órteses/
1196   próteses; OPM em queimados; OPM em nefrologia; OPM - Projeto Olhar Brasil - Programa Saúde
1197   na Escola (PSE); OPM Projeto Olhar Brasil - Programa Brasil Alfabetizado (PBA)</string>
1198   <string name="142c1">Serviço de Endocrinologia</string>
1199   <string name="142c2">Diagnóstico/Tratamento das doenças endócrinas, metabólicas e nutricionais; Cirurgia de
1200   glândulas endócrinas</string>
1201   <string name="143c1">Serviço de Endoscopia</string>
1202   <string name="143c2">Do aparelho digestivo; Do aparelho respiratório; Do aparelho urinário; Do aparelho
1203   ginecológico</string>
1204   <string name="144c1">Serviço de Farmácia</string>
1205   <string name="144c2">Dispensação de Medicamentos do Componente Especializado da Assistência Farmacêutica;
1206   Farmácia Popular; Farmácia com manipulação homeopática; Dispensação de medicamentos estratégicos; Dispensação de
1207   medicamentos básicos; Farmácia Hospitalar</string>
1208   <string name="145c1">Serviço de Fisioterapia</string>

```

```
1202 <string name="145c2">Assistência fisioterapêutica em alterações obstétricas, neonatais e uroginecológicas;  
1203 Assistência fisioterapêutica em alterações oncológicas; Assistência fisioterapêutica em oftalmologia; Assistência  
1204 fisioterapêutica cardiovesselares e pneumo-funcionais; Assistência fisioterapêutica nas disfunções músculo-  
1205 esqueléticas (todas as origens); Assistência fisioterapêutica em queimados; Assistência fisioterapêutica nas  
1206 alterações em neurologia; Diagnóstico cinético funcional</string>  
1207  
1208 <string name="146c1">Serviço de Hemoterapia</string>  
1209 <string name="146c2">Procedimentos destinados à obtenção do sangue para fins de assistência hemoterápica;  
1210 Diagnóstico em hemoterapia; Procedimentos especiais em hemoterapia; Medicina transfusional</string>  
1211  
1212 <string name="147c1">Serviço de Laboratório de Histocompatibilidade</string>  
1213 <string name="147c2">Exames de histocompatibilidade por meio de sorologia; Exames de histocompatibilidade por  
1214 sorologia e biologia</string>  
1215  
1216 <string name="148c1">Serviço de Laboratório de Prótese Dentária</string>  
1217 <string name="148c2">Laboratório Regional de Prótese Dentária</string>  
1218  
1219 <string name="149c1">Serviço de Oftalmologia</string>  
1220 <string name="149c2">Diagnóstico em oftalmologia; Tratamento cirúrgico  
1221 do aparelho da visão; Projeto Olhar Brasil; Diagnóstico em Oftalmologia por Telemedicina; Projeto Olhar Brasil -  
Programa Saúde na Escola (PSE); Projeto Olhar Brasil - Programa Brasil Alfabetizado (PBA)</string>  
1222  
1223 <string name="150c1">Serviço de Oncologia</string>  
1224 <string name="150c2">Oncologia pediátrica; Hematologia; Oncologia clínica; Radioterapia; Oncologia cirúrgica</string>
```

```
1224      <string name="153c1">Serviço de Práticas Integrativas e Complementares</string>
1225      <string name="153c2">Acupuntura; Fitoterapia; Outras técnicas em medicina tradicional chinesa; Práticas corporais
1226      /Atividade física; Homeopatia; Terapêutica; Medicina antroposófica</string>
1227
1228      <string name="154c1">Serviço de Reabilitação visual</string>
1229      <string name="154c2">Reabilitação Visual; Reabilitação Intelectual; Reabilitação Física; Reabilitação visual/
1230      mental/múltiplas deficiências; Reabilitação Auditiva; Oficina Ortopédica Fixa; Oficina Ortopédica Intinerante
1231      Terrestre; Oficina Ortopédica; Intinerante Fluvial; Atenção Fonoaudiológica; Atenção Fisioterapeútica; Atenção à
1232      Saúde das Pessoas Ostomizadas I; Atenção à Saúde das Pessoas Ostomizadas II</string>
1233
1234      <string name="155c1">Serviço de Suporte Nutricional</string>
1235      <string name="155c2">Enteral; Enteral/Parenteral; com manipulação/fabricação</string>
1236
1237      <string name="156c1">Serviço de Telemedicina</string>
1238      <string name="156c2">Exame eletrocardiográfico por telemedicina</string>
1239
1240      <string name="157c1">Serviço de Terapia Intensiva</string>
1241      <string name="157c2">ADULTO; NEONATAL; PEDIÁTRICO; QUEIMADOS; DOENÇA CORONARIANA (UCO)</string>
1242
1243      <string name="158c1">Serviço de Traumatologia e Ortopedia</string>
1244      <string name="158c2">Serviço de Traumatologia e Ortopedia; Serviço de Traumatologia e Ortopedia Pediátrica (até
1245      21 anos de idade); Serviço de Traumatologia e Ortopedia de Urgência</string>
1246
1247      <string name="159c1">Serviço de Traumato-Ortopedia</string>
1248      <string name="159c2">Diagnóstico e tratamento em cintura escapular; Diagnóstico e tratamento em membros
1249      superiores; Diagnóstico e tratamento em coluna vertebral e caixa torácica; Diagnóstico e tratamento em cintura
1250      pélvica; Diagnóstico e tratamento em membros inferiores</string>
1251
1252      <string name="160c1">Serviço de Triagem Neonatal</string>
1253      <string name="160c2">Tratamento de recém-nascido com hipotireoidismo e fenilcetonuria; Tratamento de recém-
1254      nascido com doenças falciformes; Tratamento de recém-nascido com fibrose cística; Tratamento de recém-nascido com
1255      outras doenças congênitas; Acompanhamento a pacientes com diagnósticos de Hipoplásia Adrenal Congênita;
1256      Acompanhamento a pacientes com diagnósticos de Deficiência Biotinidase</string>
```

```

1249   <string name="161c1">Serviço de Urgência e Emergência</string>
1250   <string name="161c2">Pronto Socorro Geral; Pronto Socorro Especializado; Pronto Atendimento; Estabilização de
paciente crítico/grave; Atendimento ao paciente com Acidente Vascular Cerebral (AVC); Pronto Atendimento Clínico;
Pronto Atendimento Pediátrico; Pronto Atendimento Obstétrico; Pronto Atendimento Psiquiátrico; Pronto Atendimento
Oftalmológico; Pronto Atendimento Odontológico; Pronto Socorro Pediátrico; Pronto Socorro Obstétrico; Pronto Socorro
Cardiovascular; Pronto Socorro Neurologia/Neurocirurgia; Pronto Socorro Traumato-Ortopédico; Pronto Socorro
Odontológico; Pronto Socorro Oftalmológico; Pronto Socorro Geral/Clínico</string>
1251
1252   <string name="162c1">Serviço de Videolaparoscopia</string>
1253   <string name="162c2">Diagnóstica; Cirúrgica</string>
1254
1255   <string name="163c1">Serviço de Vigilância em Saúde</string>
1256   <string name="163c2">Vigilância sanitária; Vigilância ambiental; Núcleo de vigilância
hospitalar</string>
1257
1258   <string name="164c1">Serviço Móvel de Urgência (exceto SAMU)</string>
1259   <string name="164c2">Transporte terrestre; Transporte aéreo; Transporte marítimo</string>
1260
1261   <string name="165c1">Serviço Posto de Coleta de Materiais Biológicos</string>
1262   <string name="165c2">Coleta realizada fora das estrutura laboratorial</string>
1263
1264   <string name="166c1">Teleconsultoria Síncrona; Segunda Opinião Formativa</string>
1265   <string name="166c2">Teleconsultoria Assíncrona; Teleconsultoria
1266
1267   <string name="167c1">Transplante</string>
1268   <string name="167c2">Rim; Medula óssea; Coração; Pulmão; Córnea / Esclera; Fígado; Pâncreas; Retirada de órgãos;
Pele; Válvulas cardíacas; Osteocondroligamentos; Separação, avaliação biomicroscópica e conservação da córnea/esclera
; Contagem endotelial corneana; Acompanhamento de paciente transplantado; Ações para doação e captação de órgãos e
tecidos; Retirada de globo ocular humano para transplante</string>
1269   <string name="msg_info_preferencias">Em sua primeira visualização do aplicativo foi utilizada a sua posição
geográfica atual para definir o município onde você está localizado neste momento. É interessante que você selecione
na lista acima outros municípios de seu interesse para que apareçam nos resultados de suas buscas por
estabelecimentos.</string>
1270
1271 </resources>

```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\values\styles.xml

```
1 <resources>
2
3     <style name="MeuTemaMaterial" parent="Theme.AppCompat.Light.DarkActionBar">
4
5         <item name="windowNoTitle">true</item>
6         <item name="windowActionBar">false</item>
7
8             <item name="colorPrimary">@color/primary</item>
9             <item name="colorPrimaryDark">@color/primary_dark</item>
10            <item name="colorAccent">@color/accent</item>
11            <item name="android:textColorPrimary">@color/primary_text</item>
12            <item name="android:textColorPrimaryInverse">@color/primary_text</item>
13            <item name="android:textColorSecondary">@color/secondary_text</item>
14        </style>
15
16     <style name="MeuTemaMaterial_Prefencias" parent="MeuTemaMaterial">
17         <item name="android:actionBarTheme">@style/ThemeOverlay.AppCompat.Dark.ActionBar</item>
18         <item name="windowActionBar">true</item>
19         <item name="windowNoTitle">false</item>
20     </style>
21
22     <style name="MeuTemaMaterial_AppBarOverlay" parent="ThemeOverlay.AppCompat.Dark.ActionBar" />
23
24     <style name="MeuTemaMaterial_PopupOverlay" parent="ThemeOverlay.AppCompat.Light" />
25
26     <style name="FullscreenTheme" parent="MeuTemaMaterial">
27         <item name="android: actionBarStyle">@style/FullscreenActionBarStyle</item>
28         <item name="android: windowActionBarOverlay">true</item>
29         <item name="android: windowBackground">@null</item>
30         <item name="metaButtonBarButtonStyle">?android:attr/buttonBarStyle</item>
31         <item name="metaButtonBarButtonStyle">?android:buttonBarStyle</item>
32     </style>
33
34     <style name="FullscreenActionBarStyle" parent="Widget.AppCompat.ActionBar">
35         <item name="android: background">@color/black_overlay</item>
36     </style>
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\values\styles.xml

```
36   </style>
37
38 </resources>
39
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\xml\preferencias.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <PreferenceScreen xmlns:android="http://schemas.android.com/apk/res/android"
3   android:title="@string/titulo_config">
4
5   <PreferenceCategory android:title="Municípios">
6     <MultiSelectPreference
7       android:defaultValue="@array/municipios_preferidos_default"
8         android:dialogTitle="@string/municipios_dialog_title"
9           android:entries="@array/nomes_municipios_view"
10          android:entryValues="@array/codigos_municipios"
11            android:key="@string/pref_municipios_preferidos_key"
12              android:selectable="true"
13                android:summary="@string/municipios_preferidos_subtitulo"
14                  android:title="@string/municipios_preferidos_titulo" />
15
16   <PreferenceCategory android:title="Estabelecimentos">
17     <br.com.jucemar.dimon.cadesaude.SeekBarPreference
18       android:defaultValue="1"
19         android:key="@string/pref_distancia_estabelecimentos_key" />
20
21   <PreferenceCategory>
22     <br.com.jucemar.dimon.cadesaude.InfoPreference
23       android:enabled="false"
24         android:key="@string/pref_info_key"
25           android:selectable="false" />
26
27   </PreferenceScreen>
28
29
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\xml\searchable.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <searchable xmlns:android="http://schemas.android.com/apk/res/android"
3   android:hint="Pesquisa"
4   android:label="@string/app_name">
5 </searchable>
```

```

1 apply plugin: 'com.android.application'
2
3 android {
4     compileSdkVersion      23
5     buildToolsVersion      '23.0.1'
6
7     defaultConfig {
8         applicationId "br.com.jucemar.dimon.cadesaude"
9         minSdkVersion    14
10        targetSdkVersion 23
11        versionCode     12
12        versionName     "1.0.1"
13    }
14    buildTypes {
15        release {
16            minifyEnabled false
17            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
18        }
19    }
20}
21
22 dependencies {
23     compile fileTree(include: ['*.jar'], dir: 'libs')
24     testCompile 'junit:junit:+'
25
26     compile 'com.android.support:appcompat-v7:23.+'
27     compile 'com.android.support:design:23.+'
28     compile 'com.android.support:support-v4:23.+'
29     compile 'com.android.support:recyclerview-v7:23.+'
30     compile 'com.github.clans:fab:1.6.2'
31     compile 'com.google.android.gms:play-services-location:+'
32
33 }
34 }
```

## APÊNDICE D - ARTIGO SOBRE O TCC

# CadESaúde: Um aplicativo para a plataforma Android para localizar estabelecimentos de saúde públicos e privados no Estado de Santa Catarina

Jucemar Dimon<sup>1</sup>

<sup>1</sup>Curso de Sistemas de Informações – Instituto de Informática e Estatística – Universidade Federal de Santa Catarina (UFSC) Florianópolis – SC – Brasil

jucemar.dimon@grad.ufsc.br

**Abstract.** *There are several ways to locate health information on the internet, but they can come from many different sources and may be outdated. Differently from the search results on the Internet, the National Registry of Health Establishments (CNES) is the official data source of the Ministry of Health. The system developed in this work provided a solution to the population of the State of Santa Catarina, which assists in the search of establishments of health. The application developed for the Android platform was distributed in the Google Play Store and provides the updated data about these establishments, because it uses the same information contained in CNES in its internal database.*

**Resumo.** *Há na internet diversas formas de se localizar informações de um estabelecimento de saúde, porém podem vir de muitas fontes diferentes e podem estar desatualizadas. Diferentemente dos resultados de busca na internet, o Cadastro Nacional de Estabelecimentos de Saúde (CNES) é a fonte de dados oficial do Ministério da Saúde. O sistema desenvolvido neste trabalho forneceu uma solução à população do Estado de Santa Catarina que auxilia na busca de estabelecimentos de saúde. O aplicativo desenvolvido para a plataforma Android foi distribuído na Google Play Store e fornece os dados devidamente atualizados sobre esses estabelecimentos, pois utiliza em seu banco de dados interno as mesmas informações contidas no CNES.*

### 1. Introdução

Com a grande difusão e popularização de sites de anúncios para as empresas divulgarem seus serviços na internet, muitas pessoas podem preferir esses meios para encontrarem a informação que necessitam sobre uma empresa. Ao realizar uma busca por um tipo de estabelecimento específico, como uma clínica médica, hospital ou posto de saúde utilizando esta metodologia, recebe-se uma grande quantidade de sites com informações sobre o local desejado como: MAPLink<sup>68</sup>; GuiaMais<sup>69</sup>; Telelistas<sup>70</sup>; Foursquare<sup>71</sup> e Google Mapas<sup>72</sup>. Esses permitem que usuários terceiros cadastrem informações sobre as empresas, no entanto, posteriormente, tais

---

<sup>68</sup> <http://maplink.com.br/>

<sup>69</sup> <http://www.guiamais.com.br/florianopolis-sc>

<sup>70</sup> <http://www.telelistas.net/>

<sup>71</sup> <https://pt.foursquare.com/>

<sup>72</sup> <https://www.google.com.br/maps>

informações podem estar desatualizadas e podem prejudicar quem necessita de sua localização de forma rápida.

O avanço das plataformas mobile e a sua popularização no mercado de smartphones, que em 2014 teve um crescimento de 23% segundo artigo publicado por Exame.com (2015), vêm despertando grande interesse nos desenvolvedores de aplicações para esta plataforma. Nesse sentido, muitos estabelecimentos de saúde como a Clinica Privada de Especialidades Villa María (2015) já possuem seu aplicativo para divulgação de seus serviços e operadoras de convênios médicos, como Portal Unimed Ltda (2015), sendo disponibilizado como ferramenta para seus clientes localizarem seus profissionais credenciados. As empresas do ramo de tecnologia que desenvolvem esses aplicativos, também projetam soluções genéricas para auxiliar os usuários de smartphones na localização de uma clínica, hospital, posto de saúde, entre outros, como desenvolvido por Wintimeipc (2015) em seu aplicativo MedSquare para a plataforma Android.

Pode-se perceber que existem várias formas de se localizar um estabelecimento de saúde: através de provedores de busca na internet; através de aplicativos específicos de operadoras de convênios de saúde; através de aplicativos dos próprios estabelecimentos e, até mesmo, através de catálogos impressos em papel. São variadas fontes de informações diferentes e separadas uma das outras, o que pode causar divergências nos dados além de poder prejudicar o cidadão que necessita de um rápido resultado. Foi pensando neste contexto que se desenvolveu este trabalho, com o objetivo de criar uma solução que pudesse auxiliar o cidadão a localizar um estabelecimento de saúde, o qual atenda suas necessidades de forma eficiente. Utilizando-se de tecnologias atuais da plataforma mobile como o Android, desenvolveu-se um aplicativo off-line que utiliza como banco de dados as informações contidas no CNES (Cadastro Nacional dos Estabelecimentos de Saúde), optou-se por este por serem informações fidedignas dos estabelecimentos de saúde, sendo a única fonte de consulta utilizada pelo aplicativo. Dessa forma, torna-se possível que os usuários utilizem um único aplicativo para realizar as buscas por estabelecimentos de saúde de todos os tipos, sendo eles públicos ou privados.

## **2. Conceitos**

Nesta seção serão levantados, de forma sucinta, alguns conceitos que ajudarão na compreensão do tema focado neste artigo.

### **2.1. Webcrawler**

Um web crawler, segundo Tavares e Lóscio (2013), é um agente de software que acessa a WEB de forma automatizada através de URLs para extrair dados, sendo esse processo denominado crawling. O trabalho de Tavares, Lóscio e Oliveira (2012) complementa o conceito definindo que este componente automatizado visita as páginas HTML e navega pelo seu conteúdo capturando-o, os links contidos nas páginas são adicionados a sua fila de URLs que serão visitadas na sequência, assim, um web crawler pode varrer toda a web de forma recursiva.

### **2.2. Data Access Object**

O padrão de desenvolvimentos conhecido como DAO, do inglês *Data Access Object*, segundo Taborda (2015), é um padrão introduzido no ambiente JEE para simplificar e desacoplar a interação das aplicações Java com a API JDBC. No padrão DAO, todas as comunicações com o mecanismo de persistência são mediadas por um objeto escrito especificamente para mapear as informações dos objetos específicos de transporte para a API de persistência e vice e versa. Toda a lógica de mapeamento e execução das instruções é deixada dentro do objeto DAO, desta

forma, isolando a aplicação da API de persistência por completo. O objeto DAO é criado para operar e interagir com mecanismo de persistência em nome da aplicação.

### **2.3. Singleton**

O Singleton é uma convenção que garante que um único objeto de uma classe de interesse será instanciado e que há apenas um ponto global de acesso a ela, ou seja, em qualquer ponto da aplicação de uma forma geral caso haja a intenção de uso do objeto cuja classe foi implementada usando este padrão, chama-se o construtor do objeto e caso ele já tenha sido criado, tem-se uma referência dele. Segundo Freeman et al. (2009), este tipo de padrão de projeto é muito útil quando os objetos possuem atributos de configuração que não podem se repetir.

### **2.4. Cadastro Nacional dos Estabelecimentos de Saúde**

O CNES (Cadastro Nacional dos Estabelecimentos de Saúde) é um projeto do Ministério da Saúde que consiste na criação de um banco de dados unificado, contendo o cadastro detalhado de todos os estabelecimentos de saúde do país que prestam atendimento à população. Este projeto foi instituído com a publicação da Portaria nº 511 de Brasil (2000), a qual detalha como deve ser feito todo o processo de cadastramento, atualização e transmissão dos dados de todos os estabelecimentos do país para o banco de dados central do Ministério da Saúde, administrado pelo DATASUS. Ao longo dos anos diversas portarias, notas técnicas e manuais tem sido publicado para regular os processos dos responsáveis pela realização do cadastro e atualização dessas informações, como é o caso da Portaria nº 1.646 de Brasil (2015), que reitera a importância e obrigatoriedade do cadastramento de todos os estabelecimentos de saúde pelos responsáveis de cada município, de sua manutenção e veracidade dos dados que são alimentados no sistema, assim como da transmissão mensal dessas informações do banco de dados local para o banco de dados federal, com a possibilidade de aplicação de penalidade aos gestores dos municípios caso essas exigências não sejam atendidas.

Os responsáveis pelo cadastro dos estabelecimentos são as Secretarias Municipais ou Estaduais de Saúde, esse trata-se do preenchimento de uma série de fichas que detalham todas as características físicas, jurídicas e estruturais do local onde funciona o estabelecimento, assim como os recursos humanos, tecnológicos e serviços oferecidos. Esses dados são adicionados a um banco de dados local através de um software específico, o qual na sequência pode conectar-se ao servidor central federal ou estadual para sincronização dos dados.

### **2.5. Android**

O Android é um sistema operacional para dispositivos móveis, sendo a base do mesmo originada em 2003, na cidade de Palo Alto na Califórnia. Segundo Meyer (2016), ele foi desenvolvido por Andy Rubin, Rich Miner, Nick Sears e Chris White, tais empresários já iniciados no ramo da tecnologia, foram os fundadores da Android, Inc.. Em 2005 a Google adquiriu a empresa Android, Inc, e criou a sua nova divisão de pesquisa em tecnologia móvel, mas foi somente a partir de 2007 que um grande passo foi dado, marcando a história do Android e fazendo com que outras empresas da mesma divisão de mercado começassem a se preocupar: grandes fabricantes como Samsung, Sony, HTC, operadoras como as americanas Sprint Nextel e T-Mobile, e fabricantes de hardware como Qualcomm e a Texas Instruments, além do próprio Google, reuniram-se em um consórcio de tecnologia e fundaram a OHA<sup>73</sup>. Várias outras grandes empresas do ramo de telefonia, hardware, semicondutores e de marketing e comércio participam e contribuem para a sua evolução, sendo que todas essas empresas compartilham suas expectativas e *insights* nos congressos promovidos pela OHA(Open Handset Alliance). O

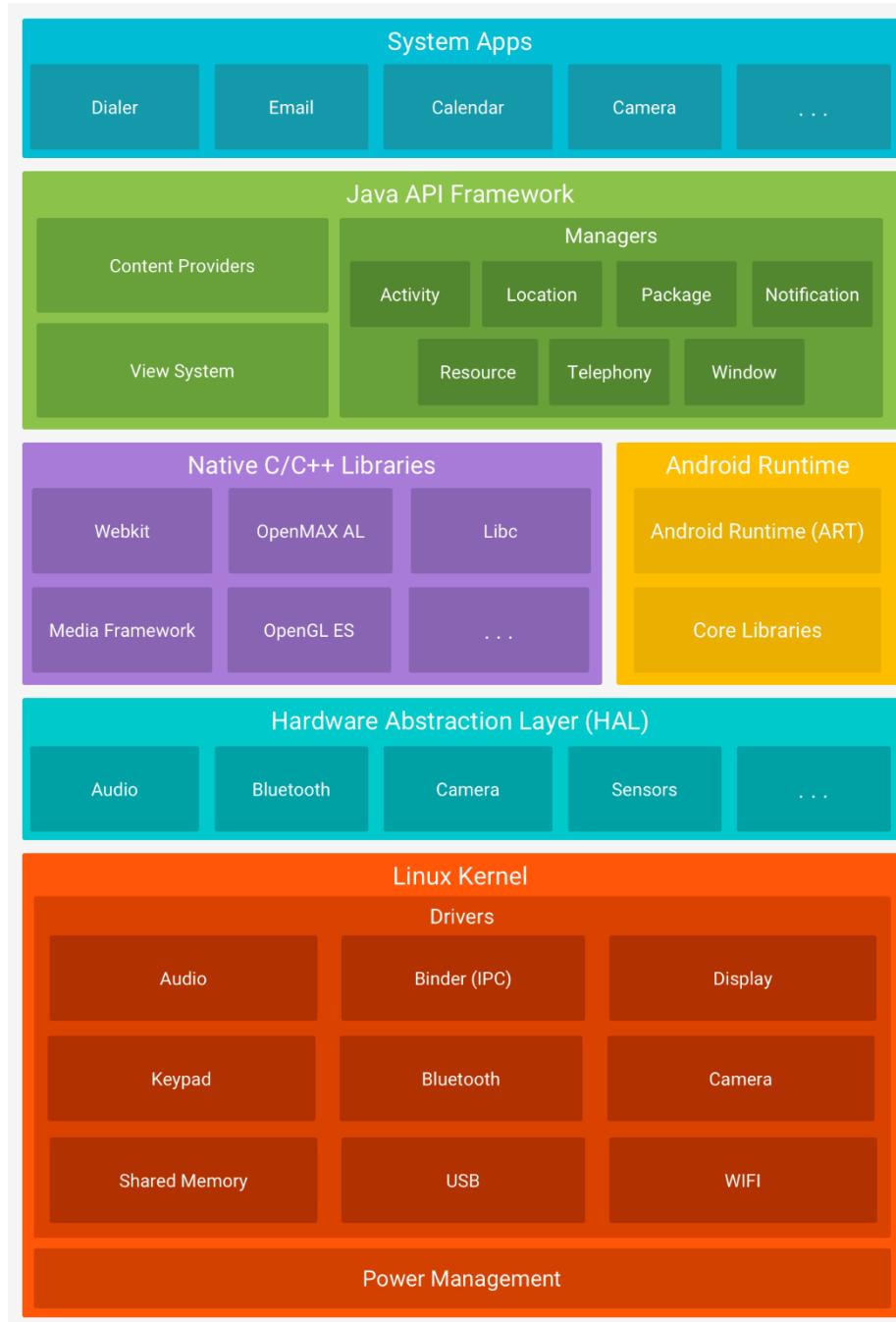
---

<sup>73</sup> Open Handset Alliance - <http://www.openhandsetalliance.com>

principal objetivo deste consórcio é a busca de um padrão de dispositivo móvel aberto e que atenda às necessidades atuais de todos.

De acordo com Alves (2016), a Google ao longo dos anos vem evoluindo seu sistema operacional de forma cada vez mais rápida. Lançado em 2009, o Android 1.5 (*Cupcake*) foi sua versão mais rudimentar, porém, ainda no mesmo ano, recebeu duas novas versões, respectivamente: Android 1.6 (*Donut*) e Android 2.0 (*Eclair*). No ano de 2010, a Google lança Android 2.2 (*Froyo*), além disso, no final do mesmo ano é liberado o Android 2.3 (*Gingerbread*), tornando-se uma das melhores versões já lançadas até aquele momento pela Google, esta liderou a popularidade por um longo tempo entre os aparelhos da época. No início de 2011, a versão menos lembrada do Android foi lançada, o Android 3.0 (*Honeycomb*), voltada especificamente para *tablets*, já no fim de 2011 o Google lançou a versão 4.0 (*Ice Cream Sandwich*) baseada no padrão de *design* do Android 3.0. Em 2012, é lançado o Android 4.1 (*Jelly Bean*) e suas atualizações que se estenderam da 4.1 até a 4.3. No ano de 2013, é lançado o Android 4.4 (*Kitkat*). Uma revolução visual foi promovida pelo lançamento do Android 5.0 (*Lollipop*) no fim de 2014, com sua linguagem visual denominada *Material Design*. No segundo semestre de 2015, o Android 6.0 (*Marshmallow*) foi apresentado oficialmente trazendo revoluções nas questões de segurança e consumo de bateria.

Em sua arquitetura, que pode ser vista na Figura 1, percebe-se que o sistema tem como base um kernel Linux. Este kernel faz toda a comunicação com a camada de hardware através de seus drivers.



**Figura 42. Arquitetura da plataforma Android**

**Fonte:** Android (2016)

Trabalhando em conjunto com o *kernel* do Linux existe a HAL<sup>74</sup>(*Hardware Abstraction Layer*). Esta camada é composta por vários módulos e interfaces que fornecem a capacidade de interação com uma linguagem de alto nível como o Java. Ao lado existe na estrutura da plataforma o *Android Runtime* que é composto pelas *Core Libraries*, juntamente com a máquina virtual do Android. As *Core Libraries* são as responsáveis por disponibilizar ao desenvolvedor tudo que é utilizável pelo Java SE. A ART (*Android Runtime*) é a máquina virtual do Android utilizada para a execução dos aplicativos. Como os aplicativos são desenvolvidos e compilados em Java eles precisam de uma máquina virtual para poderem ser

<sup>74</sup> <https://source.android.com/devices/index.html#Hardware Abstraction Layer>

interpretados. Acima dessas camadas existe o *framework* de aplicações do Android, neste estão as funcionalidades que vão permitir ao desenvolvedor criar seus aplicativos. No topo desta estrutura temos a camada de aplicações onde ficam as aplicações do Android. Por padrão o sistema operacional já possui diversos aplicativos nativo, porém como os aplicativos de terceiros também serão posicionados nessa camada, é possível substituir um aplicativo nativo por um terceiro, pois a plataforma percebe ambos da mesma forma.

### **3. Trabalhos correlatos**

No ano de 2014, os autores Sarrinho e Campos (2014) publicaram um artigo relatando o desenvolvimento do projeto SAMob (Sistema Móvel de Geolocalização e Geoprocessamento para Locais de Atendimento em Saúde). O sistema tinha como funcionalidades a possibilidade de seus usuários localizarem um estabelecimento de saúde e além disso, compartilhar, avaliar suas opiniões sobre o mesmo. O sistema era composto por duas aplicações: uma servidora que era armazenada em nuvem e utilizava a plataforma da *Google App Engine* e outra cliente através de um aplicativo para a plataforma Android. A aplicação servidora tinha como funcionalidade gerenciar os estabelecimentos de saúde e fornecer os dados as aplicações clientes que se conectavam, sendo esta última, um aplicativo que fazia a conexão com o servidor para retornar as informações sobre as buscas dos usuários, além disso, ela possibilitava aos usuários não somente a busca, mas também a qualificação dos serviços dos estabelecimentos através de comentários. Após a conclusão do desenvolvimento, os autores submeteram o sistema a um teste de usabilidade e obtiveram um bom resultado. Em suas conclusões do projeto, eles sugerem em seus trabalhos futuros que o sistema tenha seu aplicativo adaptado para outras plataformas móveis como: iPhone, Windows Phone e Firefox OS.

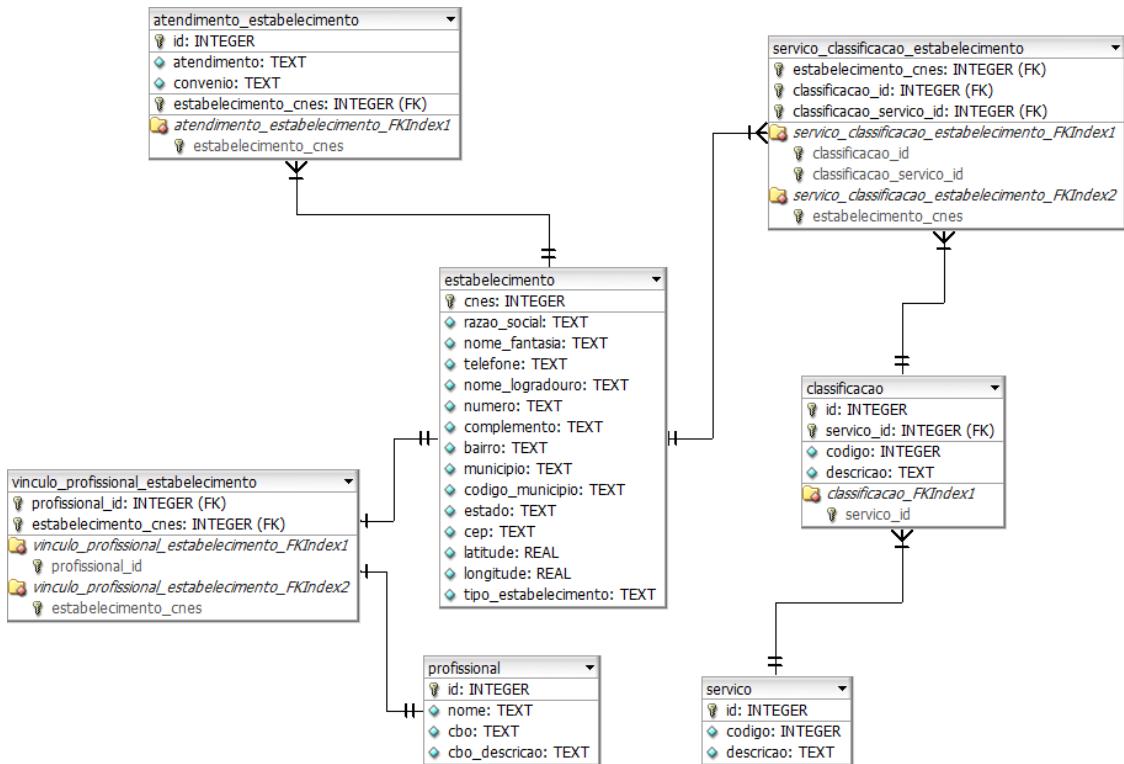
Nas lojas de aplicativos da Google e da IOS pode ser encontrado o aplicativo *MedSquare*, o qual foi desenvolvido pela empresa WINTIME IPC com o objetivo de melhorar o atendimento das pacientes nos estabelecimentos de saúde Medsquare (2015). Dentre as suas funcionalidades, os usuários podem: localizar estabelecimentos de saúde por nome, especialidade médica ou localização geográfica; comentar e qualificar sobre o atendimento naquele estabelecimento e interagir com outros usuários do aplicativo através de um chat. O aplicativo depende de uma conexão com a internet pois seus dados estão armazenados em um servidor, além disso, para que os estabelecimentos apareçam nos resultados de busca eles necessitam ser cadastrados no banco de dados, sendo que essa ação deve ser solicitada à empresa pelo responsável pelo estabelecimento. Este aplicativo já possui uma grande quantidade de usuários atualmente e a empresa investiu em atualizações de suas versões até 12 de agosto de 2015, data na qual a última atualização foi disponibilizada na loja de aplicativos da Google. Nesta última versão, aparece a informação de que a partir de agora o aplicativo possui em sua base de dados as informações do CNES. Não há nenhum tipo de informação referente a atualização desses dados, como são dados atualizados frequentemente pelo DATASUS, não há como saber se sua base de dados está atualizada.

### **4. O desenvolvimento do sistema**

Nesta seção serão apresentadas as etapas que envolveram o desenvolvimento do projeto. Na primeira seção serão tratadas as questões relacionadas a ferramenta de captura de dados, nas próximas serão descritos todos os processos envolvidos no desenvolvimento do aplicativo para a plataforma Android.

#### 4.1. A ferramenta de captura de dados

Para o desenvolvimento do aplicativo, primeiramente, foi necessário projetar quais informações sobre os estabelecimentos que seriam disponibilizadas aos usuários, sendo essas definidas como: Nome fantasia; Razão social; Endereço; Telefone; Profissionais que atuam no estabelecimento; Serviços; Categorias de atendimento; Convênios e coordenadas geográficas. Após a definição dos dados, um esquema de banco de dados foi criado para armazenar essas informações. O programa utilizado para criação do esquema foi o *DB Designer Fork*<sup>75</sup> e o esquema pode ser visto na Figura 2.



**Figura 43. Esquema de bando de dados**

**Fonte: Desenvolvido pelo autor**

A próxima etapa do desenvolvimento foi a criação de uma ferramenta que realizasse a captura dos dados do portal do CNES, essa foi desenvolvida na linguagem Java utilizando a IDE *Netbeans*<sup>76</sup> em sua versão 8. Para realizar tal ação a ferramenta deveria ter o comportamento de um *webcrawler*, varrendo as páginas HTML e capturando seu conteúdo que na sequencia seriam manipulados e extraídos os dados de interesse neste projeto. Para conseguir tal efeito na ferramenta foi utilizado a biblioteca para Java *Jsoup*<sup>77</sup> integrada ao projeto. Além dessa funcionalidade a ferramenta também necessitaria gerar e manipular um banco de dados para armazenar tais informações. O SGBD (Sistema Gerenciado de Banco de Dados) que foi utilizado nesta ferramenta foi o *SQLite*<sup>78</sup> em sua versão 3. A escolha desse SGBD foi tomada devido sua grande compatibilidade com a plataforma Android. Para que fosse possível trabalhar com este SGBD dentro da ferramenta foi utilizada a biblioteca para Java *sqlite-jdbc*<sup>79</sup>. Com o objetivo de ilustrar as tarefas e o fluxo delas nesta aplicação de captura de dados, nomeada

<sup>75</sup> <http://sourceforge.net/projects/dbdesigner-fork>

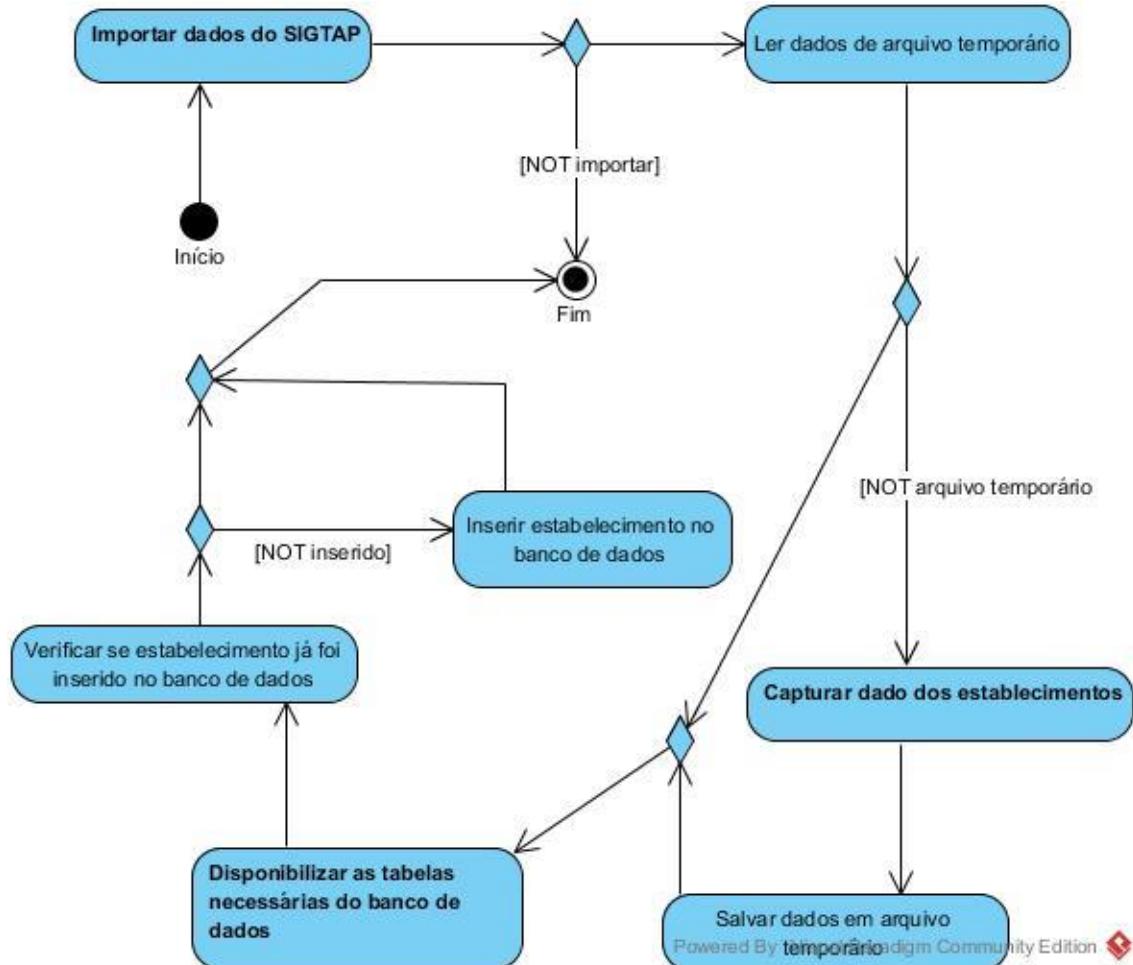
<sup>76</sup> <https://netbeans.org>

<sup>77</sup> <https://jsoup.org>

<sup>78</sup> <https://sqlite.org>

<sup>79</sup> <https://bitbucket.org/xerial/sqlite-jdbc>

como *JWebCrawlerCNES*, foi criado um diagrama de atividades utilizando UML. Este diagrama pode ser visto em detalhes na Figura 3.



**Figura 44. Diagrama de atividades UML**

**Fonte:** Desenvolvido pelo autor

A primeira etapa realizada pela ferramenta é a captura dos dados referente ao SIGTAP (Sistema de Gerenciamento da Tabela de Procedimentos, Medicamentos e OPM do SUS). Estes dados, armazenados em arquivos de texto, são retirados do portal do DATASUS e importados pela ferramenta para preencher as tabelas *serviço* e *classificação* do banco de dados, para que se tenha a informação mais atualizada sobre estas informações. As coordenadas geográficas dos estabelecimentos são adquiridas dentro da atividade de capturar dados dos estabelecimentos mencionado pelo diagrama. Ao final do processo, a ferramenta gera um arquivo de banco de dados, sendo esse utilizado no aplicativo desenvolvido.

#### 4.2. O aplicativo para Android

Após a execução da ferramenta de captura tem-se em mão um arquivo de banco de dados com informações de estabelecimentos de saúde de todos o Estado de Santa Catarina. A próxima ação foi definir a forma como esses dados seriam apresentados ao usuário através do aplicativo, então pensou-se nas funcionalidades que poderiam ser disponibilizadas aos usuários utilizando-se das possibilidades possíveis que um smartphone pode oferecer: realizar ligações telefônicas; possibilidade de trabalhar com geolocalização e a possibilidade de interagir com mapas. Dessa forma foram criadas as funcionalidades para o aplicativo. São elas:

- Realizar a busca por estabelecimentos nas redondezas;
- Filtrar os resultados das buscas por parâmetros;
- Visualizar todos os dados referentes a um estabelecimento;
- Realizar buscas por estabelecimentos dentro de um município utilizando parâmetros;
- Permitir ao usuário fazer uma ligação telefônica ao estabelecimento;
- Permitir ao usuário definir um estabelecimento como favorito;
- Permitir ao usuário visualizar a localização de um estabelecimento em um mapa.

Com as funcionalidades do aplicativo definidas, a próxima etapa foi a definição das telas que o usuário teria no aplicativo e a distribuição dessas funcionalidades entre as telas pensadas, sendo o fluxo do desenvolvimento baseado em telas. A cada definição de uma delas, o processo de programação foi iniciado, dessa forma, utilizou-se para tal a ferramenta *Android Studio*<sup>80</sup> em sua versão 2.2.2. Nas próximas seções serão apresentadas as telas desenvolvidas e suas funcionalidades.

#### 4.2.1. A tela de apresentação do aplicativo

Ao abrir o aplicativo, o usuário será levado a tela de apresentação. Esta tela pode ser visualizada na Figura 4. Não há nenhuma funcionalidade em especial nela que o usuário possa realizar.



**Figura 45. Tela de apresentação**

**Fonte: Desenvolvido pelo autor**

Nesta tela de apresentação ocorre em segundo plano uma rotina de verificação para constatar se esta é a primeira vez que o aplicativo foi aberto, assim como se ele já foi configurado. Caso negativo, o usuário será levado a tela da Figura 5, na qual ele deve fazer a

<sup>80</sup> <https://developer.android.com/studio/index.html?hl=pt-br>

configuração inicial do aplicativo. No outro caso, o usuário será levado a tela da Figura 6, nessa ele poderá visualizar todos os seus estabelecimentos definidos como favoritos.

#### **4.2.2. A tela de configuração inicial**

A tela de configuração inicial tem como objetivo exigir do usuário que defina alguns parâmetros que são importantes para o funcionamento do aplicativo, na mesma os usuários devem definir qual o raio de distância em quilômetros que os seus resultados de busca retornarão na funcionalidade de busca por GPS. Além disso, é nela que o aplicativo captura as coordenadas geográficas atuais do dispositivo para descobrir qual o município que o usuário se encontra, no entanto, o mesmo também pode definir mais municípios de interesse que serão utilizados na funcionalidade de busca por parâmetros. A Figura 5 apresenta a tela descrita.



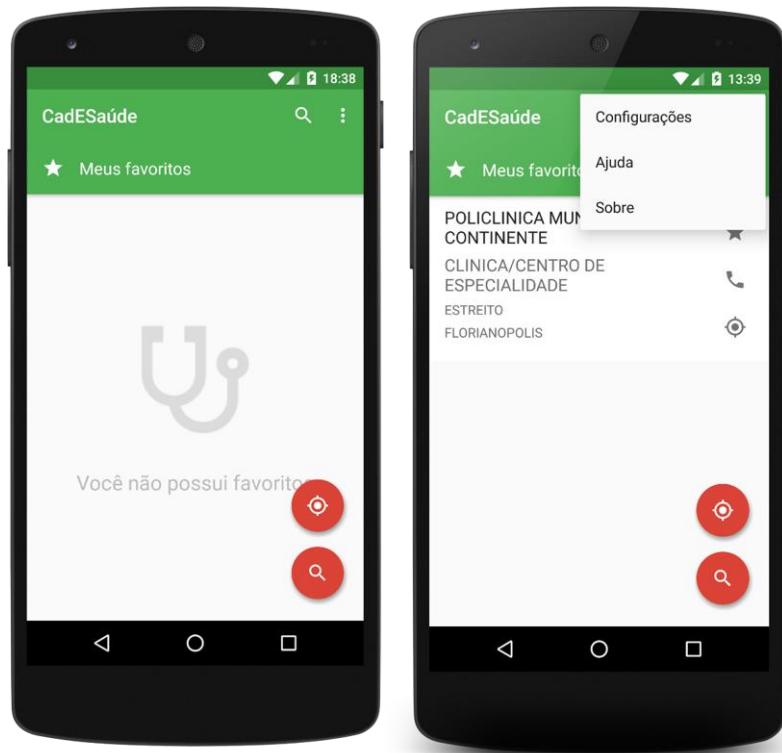
**Figura 46. Tela de configuração inicial**

**Fonte:** Desenvolvido pelo autor

Após todas as configurações serem feitas o aplicativo está preparado para fornecer todas as funcionalidades propostas. A seção seguinte mostra a tela inicial do aplicativo onde o usuário poderá visualizar todos os seus estabelecimentos favoritos.

#### **4.2.3. A tela inicial do aplicativo**

A tela inicial do aplicativo permite que o usuário acesse o menu de opções, esse fornece acesso a algumas funcionalidades como: visualizar informações sobre o aplicativo; visualiza a opção que dá acesso a tela de ajuda, que auxilia o usuário listando e explanando os vários tipos de estabelecimentos de saúde e de serviços especializados existentes, explicando alguns termos técnicos; visualizar a opção que fornece acesso a tela de configuração. A Figura 6 mostras as opções possíveis.



**Figura 47. Tela inicial**

**Fonte:** Desenvolvido pelo autor

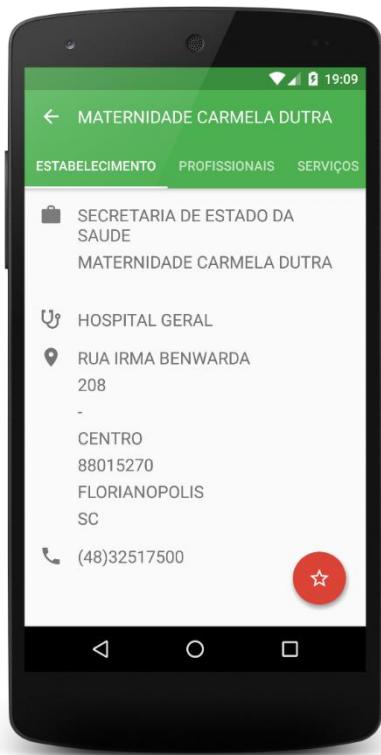
A partir desta tela o usuário poderá ter acesso as seguintes funcionalidades:

- Busca de estabelecimentos por localização geográfica. Para tal ele deverá acionar o botão de ação flutuante com símbolo de GPS;
- Busca de estabelecimentos por parâmetros. É possível acessar esta funcionalidade tocando no botão de ação flutuante com ícone de lupa.
- Esta tela apresenta uma lista de estabelecimentos contendo os estabelecimentos que foram definidos como favoritos em uma situação anterior. Pode-se interagir com esta lista e executar ações com cada um dos estabelecimentos contidos nela. É possível visualizar o cadastro completo do estabelecimento tocando nesse item na lista. Pode-se retirá-lo dos favoritos tocando no ícone de estrela. Há a possibilidade de ligar para o estabelecimento tocando no ícone do telefone ou visualizar sua posição em um mapa tocando no ícone de GPS.
- O usuário pode utilizar a busca pelo nome de um estabelecimento que esteja em sua lista de favoritos, basta o mesmo tocar no ícone de lupa na parte superior da tela, na barra de ações para que um campo de busca seja habilitado e se possa introduzir o termo a ser buscado.

Esta tela é a principal do aplicativo. Para fechá-lo, basta o pressionar o botão *back* do dispositivo.

#### **4.2.4. A tela de visualização do estabelecimento**

A tela que detalha todas as informações sobre um estabelecimento pode ser vista na Figura 7. Nela pode-se obter informações sobre sua localização e telefone de contato, assim como os profissionais vinculados, os serviços que ela possui, as formas de atendimentos (emergencial, ambulatorial, internação) e seus tipos de convênios aceitos (SUS, particular, Convênio).



**Figura 48. Tela de detalhes do estabelecimento**

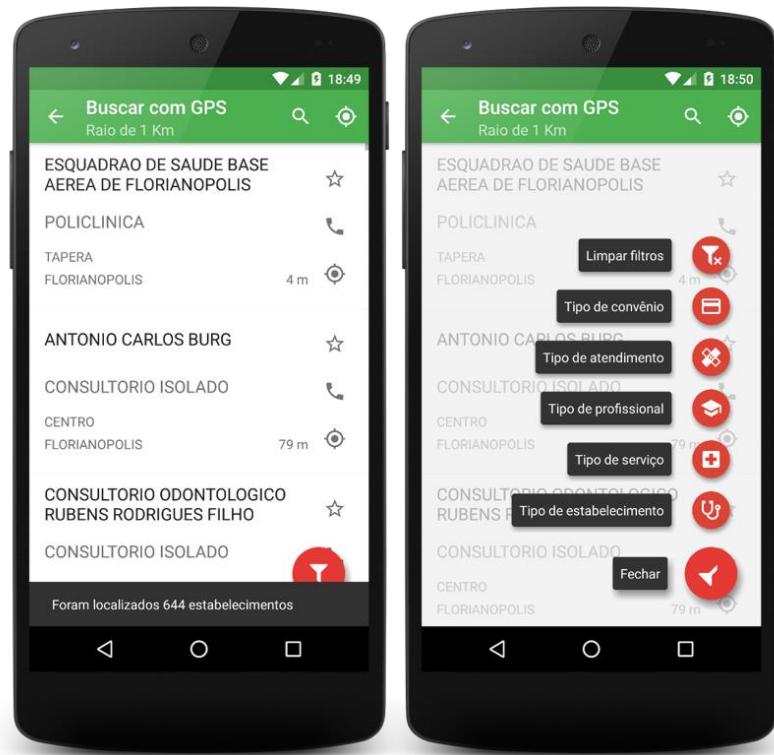
**Fonte:** Desenvolvido pelo autor

Nesta tela o usuário também tem acesso as funcionalidades já informadas anteriormente: adicionar ou excluir o estabelecimento aos seus favoritos tocando no botão com ícone de estrela, ligar para o estabelecimento tocando no ícone de telefone e ver o estabelecimento em um mapa tocando no ícone de marcação de local.

Todos os dados mostrados nesta tela são dados que estão armazenados no banco de dados interno do aplicativo. O mesmo banco gerado com a ferramenta de captura de dados explicada na seção 4.1. Todos os processamentos das consultas são feitas utilizando os recursos do dispositivo, logo, ele não necessita de uma conexão com a internet para funcionar, exceto quando a funcionalidade de visualizar a localização do estabelecimento em um mapa, pois, para esta funcionalidade, o aplicativo repassa essa solicitação a plataforma do Android e ela solicita tal ação ao aplicativo padrão de visualização de mapas do dispositivo.

#### 4.2.5. A tela de busca por GPS

A tela que possui a principal funcionalidade do aplicativo pode ser vista na Figura 8. Esta tela de busca de estabelecimentos utiliza o valor definido na tela de configurações explicado na seção 4.2.2. Esse valor referente ao raio de busca é exibido no subtítulo da tela para que o usuário seja informado sobre a distância máxima dos resultados. Ao entrar nesta tela, a qual é acessível através da tela inicial como visto na seção 4.2.3, um processo em segundo plano faz o processamento dos estabelecimentos armazenado no banco de dados que estão dentro do valor do raio de distância pré-definido. Esses estabelecimentos são apresentados na tela em uma lista de itens que possuem as mesmas funcionalidades apresentadas na seção 4.2.3.



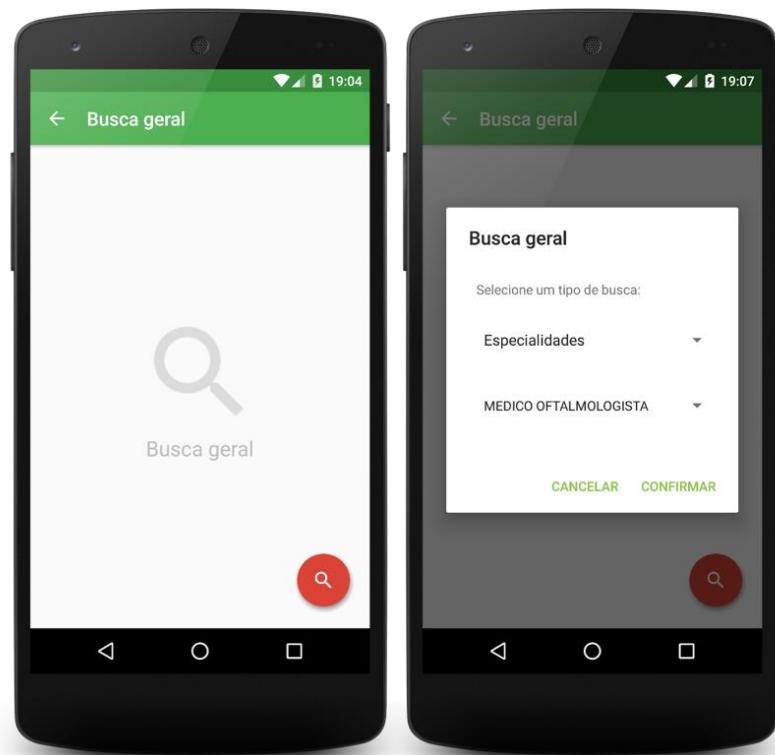
**Figura 49. Tela de busca por GPS**

**Fonte:** Desenvolvido pelo autor

A tela de buscas ainda permite que uma série de filtros sejam utilizados para ajudar ao usuário na busca pelo estabelecimento que atenda suas necessidades. Existe ainda a possibilidade de uma busca pelo nome do estabelecimento, essa é feita tocando no ícone de lupa localizado na barra de ações no topo da tela. O usuário tem a possibilidade de reprocessar as buscas e calcular as distâncias tocando no ícone de GPS localizado na barra de ferramentas, no topo da tela.

#### **4.2.6. A tela de busca por parâmetros**

Esta funcionalidade permite que o usuário faça a busca por estabelecimentos de saúde localizados dentro dos municípios que foram escolhidos na tela de configurações do aplicativo. Este processo foi explanado na seção 4.2.2, sendo que esse influencia nos resultados das buscas realizadas no banco de dados, sendo essa realizada por meio da seleção de alguma característica específica do estabelecimento como parâmetro. A Figura 9 demonstra as telas onde são feitas as configurações das buscas.



**Figura 50. Tela de busca por parâmetros**

**Fonte:** Desenvolvido pelo autor

A busca a ser realizada é configurada em uma janela de diálogo. Esta janela apresenta os possíveis valores para parâmetros de busca baseados nas características dos estabelecimentos localizados nos municípios que o usuário configurou no aplicativo, como foi explicado na seção 4.2.2. Ao montar a configuração da busca e confirma-la, a janela se fecha e uma lista é apresentada contendo os resultados das buscas, essa lista possui as mesmas características e funcionalidades da lista apresentada na seção 4.2.3. Para realizar uma nova busca o usuário deverá tocar no botão com ícone de lupa para que a janela de diálogo seja apresentada novamente.

#### **4.2.7. As atualizações do aplicativo**

O aplicativo utiliza como base para as consultas os dados referentes ao CNES, os quais são atualizados mensalmente pelas Secretarias Municipais e Estaduais de Saúde, de acordo com as legislações que regulam esse processo (BRASIL, 2015). Como a manutenção desses cadastros é feita mensalmente, pensou-se em lançar uma nova versão do aplicativo a cada mês, contendo os dados recentes sobre os estabelecimentos. Os usuários que possuem alguma versão instalada serão notificados sobre as novas versões lançadas, visto que o aplicativo se encontra disponível na loja de aplicativo da Google.

### **5. Conclusão**

Com o desenvolvimento da ferramenta de captura de dados, a qual gera o banco de dados com as informações recentemente atualizadas dos estabelecimentos de saúde que se encontram no CNES, juntamente com o aplicativo para a plataforma Android, foi possível atingir o principal objetivo deste trabalho. Após seu término, acrescenta-se na loja de aplicativos da Google mais uma ferramenta de utilidade pública para a população. Espera-se que ela faça parte da rotina do cidadão do nosso Estado e de seus visitantes em sua estadia em algum município de Santa Catarina.

Para trabalhos futuros, sugere-se uma análise da usabilidade do aplicativo, pois não foi possível cobrir esta questão durante o desenvolvimento do sistema, sendo assim, levanta-se aqui a sugestão ao leitor para o desenvolvimento dos mais atuais testes de usabilidade com usuários reais e através dos resultados de uma análise aprofundada, caso haja necessidade, providencie a reengenharia da interface gráfica do aplicativo para que a mesma se torne mais intuitiva.

Seria de grande utilidade se todos os Estados do país pudessem ter acesso a um aplicativo como este, porém a forma como foi desenvolvido, sendo um aplicativo totalmente off-line, não suportaria um banco de dados contendo todos os estabelecimentos do país, visto que o tamanho do seu arquivo poderia ultrapassar os limites definidos pela plataforma do Android, além disso, os recursos de processamento teriam que ser maiores e aumentaria o consumo do dispositivo. Sendo assim sugere-se para trabalhos futuros um estudo com maior aprofundamento, afim de promover a melhor solução para disponibilizar a toda população do país uma ferramenta que auxilie na localização de estabelecimentos de saúde.

## Referências

- BRASIL. MINISTRO DE ESTADO DA SAÚDE. Portaria nº 1.646, de 02 de outubro de 2015. Diário Oficial da União, Brasília, 2 out. 2015, v. , n. 190, p. 669-670.
- EXAME.COM. Vendas mundiais de smartphones crescem 23% em 2014. Disponível em: <<http://exame.abril.com.br/tecnologia/noticias/vendas-mundiais-de-smartphones-crescem-23-em-2014>>. Acesso em: 23 mai. 2015.
- CLINICA PRIVADA DE ESPECIALIDADES VILLA MARÍA. Clinica Especialidades. Disponível em: <<https://play.google.com/store/apps/details?id=clinicaespecialidades.myapp.name>>. Acesso em: 25 out. 2015.
- PORTAL UNIMED LTDA. Guia Médico Nacional Unimed para smartphone. Disponível em: <<https://play.google.com/store/apps/details?id=com.unimed.guiamdico>>. Acesso em: 23 mai. 2015.
- WINTIMEIPC. MedSquare. Disponível em: <[https://play.google.com/store/apps/details?id=br.com.medsquare.medsquare&hl=pt\\_BR](https://play.google.com/store/apps/details?id=br.com.medsquare.medsquare&hl=pt_BR)>. Acesso em: 24 mai. 2015.
- TAVARES, Alberto Trindade; LÓSCIO, Bernadette Farias. Processamento de consultas na Web de Dados: uma abordagem para busca de fontes de dados relevantes. In: SIMPÓSIO BRASILEIRO DE BANCO DE DADOS. 28., 2013, Recife. Short Papers... Recife: SBC, 2013. Disponível em: <<http://sbbd2013.cin.ufpe.br/Proceedings/artigos/pdfs/SBBD-Proceedings-Complete.pdf>>. Acesso em: 25 out. 2015.
- TAVARES, Alberto Trindade; LÓSCIO, Bernadette Farias; OLIVEIRA, Hélio Rodrigues de. Buscando Fontes de Dados Relevantes para Aplicações Linked Data. In: SIMPÓSIO BRASILEIRO DE SISTEMAS MULTIMÍDIA E WEB. 18., 2012, São Paulo. Anais IX Workshop de Trabalhos de Iniciação Científica... São Paulo: SBC, 2012. Disponível em: <[http://sws2012.ime.usp.br/download/WebMedia2012\\_Anais.pdf](http://sws2012.ime.usp.br/download/WebMedia2012_Anais.pdf)>. Acesso em: 25 out. 2015.
- TABORDA, Sérgio Manuel Marcos. DAO. Disponível em: <<https://sergiotaborda.wordpress.com/desenvolvimento-de-software/java/patterns/dao>>. Acesso em: 30 mai. 2015.
- FREEMAN, Eric; FREEMAN, Elisabeth; SIERRA, Kathy; et al. Use a Cabeça: Padrões e Projetos. 2 ed. Rio de Janeiro: Alta Books, 2009.

BRASIL. MS-SAS: Secretaria de Assistência a Saúde . Portaria nº 511, de 29 de dezembro de 2000. Diário Oficial da União, Brasília, 4 jan. 2001, v. , n. 3-E, p. 11-20.

BRASIL. MINISTRO DE ESTADO DA SAÚDE. Portaria nº 1.646, de 02 de outubro de 2015. Diário Oficial da União, Brasília, 2 out. 2015, v. , n. 190, p. 669-670.

MEYER, Maximiliano. A história do Android. Disponível em: <<https://www.oficinadanet.com.br/post/13939-a-historia-do-android>>. Acesso em: 31 ago. 2016.

ALVES, Paulo. A evolução do Android: do Cupcake ao Marshmallow, conheça todas as versões. Disponível em: <<http://www.techtudo.com.br/noticias/noticia/2015/10/a-evolucao-do-android-do-cupcake-ao-marshmallow-conheca-todas-as-versoes.html>>. Acesso em: 31 ago. 2016.

ANDROID. Platform Architecture. 2016. Disponível em: <<https://developer.android.com/guide/platform/index.html>>. Acesso em: 1 set. 2016.

SARINHO, Victor Travassos; CAMPOS, Luan Rios. SAMob - Sistema Móvel de Geolocalização e Geoprocessamento para Locais de Atendimento em Saúde. Journal of Health Informatics. São Paulo; v. 6, n. 3, set. 2014. p. 96-101. Disponível em: <<http://www.jhi-sbis.saude.ws/ojs-jhi/index.php/jhi-sbis/article/view/304>>. Acesso em: 18 out. 2015.

MEDSQUARE. medsquare. Disponível em: <<http://www.medsquare.com.br>>. Acesso em: 25 out. 2015.

**ANEXOS**

## **ANEXO A - TIPOS DE ESTABELECIMENTOS/UNIDADES**

01 - Posto de Saúde: Unidade destinada à prestação de assistência a uma determinada população, de forma programada ou não, por profissional de nível médio, com a presença intermitente ou não do profissional médico.

02 - Centro de Saúde/Unidade Básica de Saúde: Unidade para realização de atendimentos de atenção básica e integral a uma população, de forma programada ou não, nas especialidades básicas, podendo oferecer assistência odontológica e de outros profissionais de nível superior. A assistência deve ser permanente e prestada por médico generalista ou especialista nestas áreas. Podendo ou não oferecer: SADT e Pronto atendimento 24 Horas.

04 - Policlínica: Unidade de saúde para prestação de atendimento ambulatorial em várias especialidades, incluindo ou não as especialidades básicas, podendo ainda ofertar outras especialidades não médicas. Podendo ou não oferecer: SADT e Pronto atendimento 24 Horas.

05 - Hospital Geral: Hospital destinado à prestação de atendimento nas especialidades básicas, por especialistas e/ou outras especialidades médicas. Pode dispor de serviço de Urgência/Emergência. Deve dispor também de SADT de média complexidade. Podendo ter ou não SIPAC.

07 - Hospital Especializado: Hospital destinado à prestação de assistência à saúde em uma única especialidade/área. Pode dispor de serviço de Urgência/Emergência e SADT. Podendo ter ou não SIPAC Geralmente de referência regional, macrorregional ou estadual.

15 - Unidade Mista: Unidade de saúde básica destinada à prestação de atendimento em atenção básica e integral à saúde, de forma programada ou não, nas especialidades básicas, podendo oferecer assistência odontológica e de outros profissionais, com unidade de internação, sob administração única. A assistência médica deve ser permanente e prestada por médico especialista ou generalista. Pode

dispor de urgência/emergência e SADT básico ou de rotina. Geralmente nível hierárquico 5.

20 - Pronto Socorro Geral: Unidade destinada à prestação de assistência a pacientes com ou sem risco de vida, cujos agravos necessitam de atendimento imediato. Podendo ter ou não internação.

21 - Pronto Socorro Especializado: Unidade destinada à prestação de assistência em uma ou mais especialidades, a pacientes com ou sem risco de vida, cujos agravos necessitam de atendimento imediato. MS/SAS/DRAC/CGSI-Coordenação Geral dos Sistemas de Informação Manual Técnico do Cadastro Nacional de Estabelecimentos de Saúde - Versão 2 Atualizado em outubro de 2006.

22-Consultório Isolado: sala isolada destinada à prestação de assistência médica ou odontológica ou de outros profissionais de saúde de nível superior.

32-Unidade Móvel Fluvial: Barco/navio equipado como unidade de saúde, contendo no mínimo um consultório médico e uma sala de curativos, podendo ter consultório odontológico.

36-Clínica Especializada/Amb. Especializado: Clínica Especializada destinada à assistência ambulatorial em apenas uma especialidade/área da assistência. (Centro Psicossocial/Reabilitação, etc..).

39 - Unidade de Serviço de Apoio de Diagnose e Terapia: Unidades isoladas onde são realizadas atividades que auxiliam a determinação de diagnóstico e/ou complementam o tratamento e a reabilitação do paciente.

40 - Unidade Móvel Terrestre: Veículo automotor equipado, especificamente, para prestação de atendimento ao paciente.

42 - Unidade Móvel de Nível Pré-hospitalar na Área de Urgência e Emergência: Veículo terrestre, aéreo ou hidroviário destinado a prestar atendimento de urgência e

emergência pré-hospitalar a paciente vítima de agravos a sua saúde. (PTMS/GM 824, de 24/Jun/1999).

43 - Farmácia: Estabelecimento de saúde isolado em que é feita a dispensação de medicamentos básicos/essenciais (Programa Farmácia Popular) ou medicamentos excepcionais/alto custo previstos na Política Nacional de Assistência Farmacêutica.

50 - Unidade de Vigilância em Saúde: É o estabelecimento isolado que realiza trabalho de campo a partir de casos notificados e seus contatos, tendo como objetivos: identificar fontes e modo de transmissão; grupos expostos a maior risco; fatores determinantes; confirmar o diagnóstico e determinar as principais características epidemiológicas, orientando medidas de prevenção e controle a fim de impedir a ocorrência de novos eventos e/ou o estabelecimento de saúde isolado responsável pela execução de um conjunto de ações, capaz de eliminar, diminuir ou prevenir riscos à saúde capaz de eliminar, diminuir ou prevenir riscos à saúde e de intervir nos problemas sanitários decorrentes do meio ambiente, da produção e circulação de bens e da prestação de serviços de interesse da saúde.

60 - Cooperativa: Unidade administrativa que disponibiliza seus profissionais cooperados para prestarem atendimento em estabelecimento de saúde.

61 - Centro de Parto Normal Isolado: Unidade intra-hospitalar ou isolada, especializada no atendimento da mulher no período gravídico puerperal, conforme especificações da PT/MS 985/99.

62 - Hospital /Dia- Isolado: – Unidades especializadas no atendimento de curta duração com caráter intermediário entre a assistência ambulatorial e a internação.

64 - Central de Regulação de Serviços de Saúde: É a unidade responsável pela avaliação, processamento e agendamento das solicitações de atendimento, garantindo o acesso dos usuários do SUS, mediante um planejamento de referência e contra referência.

67 - Laboratório Central de Saúde Pública - LACEN- Estabelecimento de Saúde que integra o Sistema Nacional de Laboratórios de Saúde Pública - SISLAB, em conformidade com normalização vigente.

## ANEXO B - TIPOS DE SERVIÇOS ESPECIALIZADOS

Código	Descrição do serviço
100	Serviço de Atenção à Saúde no Sistema Penitenciário
101	Estratégia de Saúde da Família
102	Estratégias de Agentes Comunitário de Saúde
103	Serviço de Atendimento Móvel de Urgências
104	Regulação Assistencial dos Serviços de Saúde
105	Serviço de Atenção em Neurologia/ Neurocirurgia
106	Serviço de Atenção a DST/HIV/AIDS
107	Serviço de Atenção à Saúde Auditiva
108	Serviço de Atenção a Saúde do Trabalhador
109	Serviço de Atenção à Saúde dos Adolescentes em Conflito com a Lei, em Regime de Internação e Internação Provisória
110	Serviço de Atenção à Saúde Reprodutiva
111	Serviço de Atenção ao Paciente com Tuberculose
112	Serviço de Atenção ao Pré-Natal, Parto e Nascimento
113	Serviço de Atenção Domiciliar
114	Serviço de Atenção em Saúde Bucal
115	Serviço de Atenção Psicossocial
116	Serviço de Atenção Cardiovascular/Cardiologia
117	Serviço de Cirurgia Reparadora
118	Serviço de Cirurgia Torácica
119	Serviço de Controle de Tabagismo
120	Serviço de Diagnóstico por Anatomia Patológica e/ou Cito patologia
121	Diagnóstico por Imagem
122	Serviço de Diagnóstico por Métodos Gráficos/Dinâmicos
123	Serviço de Dispensação de Órteses, Próteses e Materiais Especiais
124	Serviço de Endocrinologia
125	Serviço de Farmácia
126	Serviço de Fisioterapia
127	Serviço de Atenção à Obesidade
128	Serviço de Hemoterapia
129	Serviço de Laboratório de Histocompatibilidade
130	Atenção à Doença Renal Crônica
131	Serviço de Oftalmologia
132	Serviço de Oncologia
133	Serviço de Pneumologia
134	Serviço de Práticas Integrativas e Complementares
135	Serviço de Reabilitação
136	Serviço de Suporte Nutricional
137	Serviço de Telemedicina
138	Serviço de Traumato-Ortopedia
139	Serviço de Triagem Neonatal
140	Serviço de Urgência e Emergência
141	Serviço de Vigilância em Saúde

142	Serviço de Endoscopia
143	Serviço Móvel de Urgência (exceto SAMU)
144	Serviço Posto de Coleta de Materiais Biológicos
145	Serviço de Diagnóstico por Laboratório Clínico
146	Serviço de Videolaparoscopia
147	Serviço de Apoio a Saúde da Família
148	Hospital-Dia
149	Transplante
150	Cirurgia Vascular
151	Medicina Nuclear
153	Atenção Especializada no Processo Transexualizador
154	Serviço de Banco de Tecidos
155	Serviço de Traumatologia Ortopedia
156	Serviço de Atenção à Saúde das Pessoas Ostomizadas
157	Serviço de Laboratório de Prótese Dentaria
158	Serviço de Atenção Integral em Hanseníase
159	Atenção Básica
160	Tele consultoria
162	Serviço de Terapia Intensiva
163	Serviço de Cuidados Intermediários
164	Serviço de Órteses, Próteses e Materiais Especiais em Reabilitação
165	Atenção às Pessoas em Situação de Violência Sexual
167	Atendimento Itinerante de Assistência e Ensino em Saúde
168	Atenção às Pessoas com Doenças Raras
169	Serviço de Atenção em Urologia