

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA ELÉTRICA – PATOS DE MINAS
ENGENHARIA ELETRÔNICA E DE TELECOMUNICAÇÕES

CAIO MATHEUS PEREIRA BRAGA

DESENVOLVIMENTO DE *SOFTWARE* PARA ANÁLISE
QUALITATIVA DO SINAL DE REDES MÓVEIS 4G DE
PADRÃO LTE

Patos de Minas - MG

2018

CAIO MATHEUS PEREIRA BRAGA

**DESENVOLVIMENTO DE *SOFTWARE* PARA ANÁLISE
QUALITATIVA DO SINAL DE REDES MÓVEIS 4G DE
PADRÃO LTE**

Projeto de pesquisa apresentado à banca examinadora como requisito parcial de avaliação da disciplina de TCC 2 da graduação em Engenharia Eletrônica e de Telecomunicações, da Faculdade de Engenharia Elétrica, da Universidade Federal de Uberlândia, Campus Patos de Minas.

Orientador(a): Prof. Dr. Eliana Pantaleão

Co-orientador(a): Prof. Dr. Diego de Brito Piau

Patos de Minas - MG

2018

CAIO MATHEUS PEREIRA BRAGA

**DESENVOLVIMENTO DE *SOFTWARE* PARA ANÁLISE
QUALITATIVA DO SINAL DE REDES MÓVEIS 4G DE PADRÃO LTE**

Projeto de pesquisa apresentado à banca examinadora como requisito parcial de avaliação da disciplina de TCC 2 da graduação em Engenharia Eletrônica e de Telecomunicações, da Faculdade de Engenharia Elétrica, da Universidade Federal de Uberlândia, Campus Patos de Minas.

Orientador(a): Prof.^a Dr.^a Eliana Pantaleão

Co-orientador(a): Prof. Dr. Diego de Brito Piau

Patos de Minas, 13 de dezembro de 2018.

Banca Examinadora:

Prof.^a Dr.^a Eliana Pantaleão – FACOM/UFU (Orientadora)

Prof. Dr. Laurence Rodrigues do Amaral – FACOM/UFU (Membro 1)

Prof.^a Dr.^a Karine Barbosa Carbonaro – FEELT/UFU (Membro 2)

Prof. Dr. Diego de Brito Piau – FEELT/UFU (Co-orientador)

AGRADECIMENTOS

Agradeço a todos os envolvidos na realização deste trabalho, em especial à minha orientadora Dr.^a Eliana Pantaleão e ao meu co-orientador Dr. Diego de Brito Piau, que não mediram esforços para me guiar à conclusão deste trabalho. Agradeço novamente ao Dr. Diego Brito Piau, com quem desenvolvi inúmeros trabalhos de pesquisa durante a graduação, o que certamente me trouxe experiência e sabedoria suficientes, tanto para a realização deste trabalho, quanto para as próximas etapas e desafios como um futuro engenheiro. Agradeço também a cada um dos professores que estiveram presentes na minha formação, e que com toda certeza me proporcionaram muito conhecimento ao compartilhar um pouco de sua sabedoria.

Agradeço a toda equipe da HST *Card Technology*, que me deram a oportunidade de estar lá, e fazer parte da família, desenvolvendo meu período de estágio em um ambiente que me traz total crescimento profissional e pessoal.

Agradeço imensamente a Deus, que guiou todos os meus passos durante a graduação, me mostrando que eu poderia ir muito além do que eu acreditava ser capaz, e me dando força em todos os momentos difíceis.

Agradeço infinitamente a toda a minha família, que sempre me deu o apoio necessário para que eu estivesse na universidade e desse o melhor de mim. Em especial agradeço a minha mãe, que desde minha infância sempre confiou no meu potencial e me apoiou em todos os meus passos até a graduação, e nunca mediu esforços para que eu chegasse lá, mesmo que isso lhe custasse mais do que ela poderia oferecer.

Por fim, agradeço aos inúmeros amigos feitos durante esses anos na universidade por cada momento de descontração juntos, cada dia de estudos ou a cada conselho e palavra amiga, em especial àqueles amigos da sétima turma de Engenharia Eletrônica e de Telecomunicações. Certamente me tornei mais forte com a ajuda de vocês, meus amigos!

RESUMO

Nos últimos anos, o crescimento do número de usuários de redes móveis em todo mundo tem crescido expressivamente. Este crescimento traz inúmeros desafios para os profissionais e empresas que atuam nessa área, que estão sempre buscando melhorar os serviços que prestam para atrair cada vez mais usuários. Este trabalho mostra o desenvolvimento de um *software* que possibilita a medição de variáveis importantes na análise da qualidade e desempenho do sinal proveniente de redes 4G de padrão LTE de uma forma simples e econômica, além do tratamento desses dados para a confecção de análises sucintas sobre os dados medidos. Vários parâmetros que afetam a qualidade são apresentados nesse trabalho, além das formas de obtê-los através do desenvolvimento de *software* para *smartphones*.

Palavras-chave: LTE. Android. Desenvolvimento de *software*. Redes móveis. Qualidade.

ABSTRACT

In the last years, the number of mobile networks users have increased extremely. This growth brings countless challenges for the professionals and companies that work in this area, who are always looking to improve their services to attract more users. This project shows the development of a *software* that allows allow the measurement of important variables in the analysis of quality and performance of the signal coming from 4G networks of LTE standard in a simple and cheap way. Several parameters that affect the signal quality are shown in this project and also the way to get them.

Index Terms: LTE. Android. Software Development. Mobile Networks. Quality.

LISTA DE FIGURAS

FIGURA 1: TAXA DE CRESCIMENTO DA POPULAÇÃO E DO NÚMERO DE CELULARES A NÍVEL MUNDIAL.	15
FIGURA 2: QUANTIDADE DE CELULARES NO MUNDO.	17
FIGURA 3: CRESCIMENTO DO NÚMERO DE REDES MÓVEIS NO BRASIL.	18
FIGURA 4: COMPETIÇÃO DAS PRESTADORAS DO SERVIÇO DE TELEFONIA MÓVEL.	18
FIGURA 5: ILUSTRAÇÃO DE UM SISTEMA DE TELEFONIA CELULAR BÁSICO.	21
FIGURA 6: SISTEMA DE TELEFONE SEM FIO.	22
FIGURA 7: ESQUEMA DE UM SISTEMA AMPS BÁSICO.	23
FIGURA 8: ARQUITETURA DE UMA REDE LTE.	26
FIGURA 9: ORGANIZAÇÃO DOS COMPONENTES DE UMA REDE EPC.	27
FIGURA 10: TÉCNICAS DE UTILIZAÇÃO DE VÁRIAS ANTENAS.	30
FIGURA 11: MAPEAMENTO CQI X SNR.	31
FIGURA 12: <i>SCANNER</i> TSMW.	32
FIGURA 13: <i>MODEM</i> HUAWEI 392.	33
FIGURA 14: INTERFACE GRÁFICA DO USUÁRIO COM O <i>SOFTWARE</i> ROHMES.	34
FIGURA 15: INTERFACE DO <i>SOFTWARE</i> QUALIPOC.	35
FIGURA 16: VENDA DE <i>SMARTPHONES</i> NO MUNDO.	36
FIGURA 17: <i>RADIO INTERFACE LAYER</i>	37
FIGURA 18: INTERFACE DO ANDROID STUDIO.	41
FIGURA 19: TELA INICIAL DO APLICATIVO.	43
FIGURA 20: TELA DE GRAVAÇÃO E MEDIÇÃO DOS DADOS.	44
FIGURA 21: TELA DO MAPA DE COBERTURA.	45
FIGURA 22: TELA COM GRÁFICO.	45
FIGURA 23: TELA PARA ESCOLHA DA OPÇÃO DE MODELO DE GRÁFICO.	46
FIGURA 24: GRÁFICO DE LINHAS.	47
FIGURA 25: GRÁFICO DE BARRAS.	47
FIGURA 26: TELA DE PESQUISA DETALHADA.	48
FIGURA 27: VALORES ESTATÍSTICOS SOBRE OS DADOS GRAVADOS.	49
FIGURA 28: IMPLEMENTAÇÃO DO MÉTODO <code>ON_SIGNAL_STRENGTH_CHANGED</code>	50
FIGURA 29: TRECHO DE CÓDIGO DO MÉTODO <code>ON_SIGNAL_STRENGTH_CHANGED()</code>	51
FIGURA 30: TRECHO DE CÓDIGO DO MÉTODO <code>CONFIGURAR_SERVICO()</code>	52
FIGURA 31: IMPLEMENTAÇÃO DO MÉTODO <code>ATUALIZAR()</code>	53
FIGURA 32: CLASSE <code>CREATE_DATABASE</code>	54
FIGURA 33: MÉTODO <code>INSERT_DATA()</code>	55

FIGURA 34: MÉTODO READDATA().	56
FIGURA 35: IMPLEMENTAÇÃO DO EVENTO DE CLIQUE DO BOTÃO DE GRAVAÇÃO DOS DADOS.	57
FIGURA 36: IMPLEMENTAÇÃO DA <i>THREAD</i> RESPONSÁVEL PELA GRAVAÇÃO DOS DADOS NO BANCO.	58
FIGURA 37: IMPLEMENTAÇÃO DO MÉTODO ONCREATE() DA CLASSE MAPADECOBERTURA.	59
FIGURA 38: IMPLEMENTAÇÃO DO MÉTODO ONMAPREADY() DA CLASSE MAPADECOBERTURA.	60
FIGURA 39: TRECHO DE CÓDIGO DA CLASSE GRAPHSACTIVITY.	61
FIGURA 40: TRECHO DE CÓDIGO DA CLASSE DATAILSACTIVITY.	62
FIGURA 41: IMPLEMENTAÇÃO DO MÉTODO READDATAByID.	62
FIGURA 42: IMPLEMENTAÇÃO DOS MÉTODOS DE BUSCA DE DADOS ESTATÍSTICOS NO BANCO.	63
FIGURA 43: CONFIGURAÇÕES DE LOCALIZAÇÃO DO TELEFONE.	64
FIGURA 44: CONFIGURAÇÃO DE REDE RECOMENDADA.	65
FIGURA 45: MAPA DE COBERTURA DA OPERADORA TIM NA CIDADE DE CAMPINAS, SP.	69
FIGURA 46: INTENSIDADE DO SINAL FORNECIDA PELO SISTEMA MOSAICO DA ANATEL.	69
FIGURA 47: MAPA DE COBERTURA DO BAIRRO VILA INDUSTRIAL E SUAS MEDIAÇÕES.	70
FIGURA 48: INFORMAÇÕES ESTATÍSTICAS SOBRE OS DADOS MEDIDOS	71
FIGURA 49: GRÁFICOS DE LINHA PLOTADOS PELO <i>SOFTWARE</i> DESENVOLVIDO.	71
FIGURA 50: DISTRIBUIÇÃO DE FREQUÊNCIA DE RSRP E RSRQ.	72
FIGURA 51: MAPA DE COBERTURA DA OPERADORA TIM NAS MEDIAÇÕES DO BAIRRO VILA INDUSTRIAL.	73
FIGURA 52: PONTO MÍNIMO DE RSRQ.	75
FIGURA 53: DETALHES DE MEDIÇÃO DO PONTO QUE APRESENTOU O MENOR VALOR DE RSRQ.	75
FIGURA 54: RESULTADO DE BUSCA A PONTO MÍNIMO DE RSRQ ATRAVÉS DO SISTEMA MOSAICO.	76
FIGURA 55: PONTO MÁXIMO DE RSRP NO GRÁFICO DE LINHAS.	77
FIGURA 56: RESULTADO DE BUSCA DOS DETALHES DO PONTO DE MEDIÇÃO COM VALOR MÁXIMO DE RSRP.	78
FIGURA 57: RESULTADO DE BUSCA A PONTO MÁXIMO DE RSRP ATRAVÉS DO SISTEMA MOSAICO.	79
FIGURA 58: LOCALIZAÇÃO DE ANTENAS DA TIM E DOS PONTOS DE MÁXIMO E MÍNIMO DE RSRP E RSRQ.	80

LISTA DE QUADROS E TABELAS

QUADRO 1: CLASSES DA API “ANDROID.TELEPHONY” PARA ANÁLISE DA QUALIDADE DO SINAL 4G	
LTE.	38
TABELA 2: TABELA COM ALGUNS VALORES QUE MEDIDOS PELO <i>SOFTWARE</i> DESENVOLVIDO.....	68

LISTA DE ABREVIATURAS E SIGLAS

1G	<i>1th Generation of Mobile Network</i>
2G	<i>2th Generation of Mobile Network</i>
3G	<i>3th Generation of Mobile Network</i>
4G	<i>4th Generation of Mobile Network</i>
3GPP	<i>3rd Generation Partnership Project</i>
AMPS	<i>Advanced Mobile Phone System</i>
ANATEL	<i>Agência Nacional De Telecomunicações</i>
API	<i>Application Programming Interface</i>
CDMA	<i>Code Division Multiple Access</i>
CN	<i>Core Network</i>
CP	<i>Cyclic Prefix</i>
CQI	<i>Channel Quality Indicator</i>
dBm	<i>Decibel Milliwatt</i>
EDGE	<i>Enhanced Data Rates For GSM Evolution</i>
E-UTRAN	<i>Evolved Terrestrial Radio Access Network</i>
EPC	<i>Evolved Packet Core</i>
EPS	<i>Evolved Packet System</i>
eNodeB	<i>Evolved Node B</i>
FM	<i>Frequency Modulation</i>
GPS	<i>Global Positioning System</i>
GSM	<i>Global System for Mobile Communications</i>
HSPA	<i>High Speed Packet Access</i>
ITU	<i>International Telecommunication Union</i>
IP	<i>Internet Protocol</i>

LTE	<i>Long Term Evolution</i>
MIMO	<i>Multiple-input and multiple-output</i>
MSC	<i>Mobile Switching Center</i>
MME	<i>Mobility Management Entity</i>
OFDM	<i>Orthogonal frequency-division multiplexing</i>
OFDMA	<i>Orthogonal Frequency Division Multiplexing / Multiple Access</i>
PCRF	<i>Policy and charging rules function</i>
P-GW	<i>Packet Data Network Gateway</i>
PSTN	<i>Public switched telephone network</i>
QOS	<i>Quality of Service</i>
RF	Radiofrequência
RIL	<i>Radio Interface Layer</i>
RNC	<i>Radio Network Controller</i>
RSRP	<i>Reference Signal Reference Power</i>
RSRQ	<i>Reference Signal Reference Quality</i>
RSSI	<i>Received Signal Strength Indicator</i>
RSSNR	<i>Received Signal Signal Noise Ratio</i>
S-GW	<i>Serving Gateway</i>
SINR	<i>Signal to Interference and Noise Ratio</i>
SS7	Sistema de Sinalização 7
UE	<i>User Equipment</i>
UMTS	<i>Universal Mobile Telecommunication System</i>
XML	<i>Extensible Markup Language</i>

SUMÁRIO

1	INTRODUÇÃO.....	15
1.1.	Tema do projeto	17
1.2.	Justificativas.....	17
1.3.	Objetivos	19
1.4.	Problematização	19
1.5.	Hipóteses.....	20
1.6.	Considerações finais	20
2	DESENVOLVIMENTO TEÓRICO	21
2.1	Introdução à telefonia celular.....	21
2.2	As diferentes gerações de telefonia móvel.....	22
2.2.1	Primeira geração - 1G	23
2.2.2	Segunda Geração – 2G.....	24
2.2.3	Terceira geração – 3G	24
2.2.4	Quarta geração – 4G.....	24
2.3	Redes LTE	25
2.3.1	Arquitetura da rede LTE	25
2.3.1.1	E-utran.....	26
2.3.1.2	EPC.....	27
2.4	Parâmetros de desempenho e qualidade	28
2.4.1	<i>Reference Signal Reference Power</i>	28
2.4.1.1	Resolução da Anatel quanto ao limiar mínimo de RSRP	28
2.4.2	<i>Reference Signal Reference Quality</i>	29
2.4.3	<i>Reference Signal Strength Indicator</i>	29
2.4.4	<i>Signal to Interference And Noise Ratio</i>	29
2.4.5	Técnicas de múltiplas antenas	30

2.4.6	<i>Channel Quality Indicator</i> (CQI).....	31
2.4.7	<i>Throughput</i>	32
2.5	Aparelhos e <i>softwares</i> de medição comerciais	32
2.5.1	<i>Scanner</i>	32
2.5.1.1	<i>Scanner</i> Rohde & Schwarz® TSMW.....	32
2.5.2	<i>Modem</i> Huawei® E392	33
2.5.3	<i>Software</i> Rohde & Schwarz® Rohmes 4	33
2.5.4	<i>Software</i> Rohde & Schwarz® Qualipoc android.....	34
2.6	Desenvolvimento de <i>software</i> para medição da qualidade do sinal	35
2.6.1	Arquitetura Android e acesso à camada de rádio	36
2.7	Considerações finais	39
3	MATERIAIS, TECNOLOGIAS E MÉTODOS.....	40
3.1	Materiais e tecnologias	40
3.1.1	Android Studio	40
3.1.2	<i>Smartphone</i>	41
3.1.3	Banco de dados SQLite.....	41
3.1.4	API de mapas do Google.....	42
3.1.5	API de gráficos.....	42
3.2	Métodos	42
3.2.1	Desenvolvimento do <i>software</i> de medição.....	42
3.2.1.1	Criação dos <i>Layouts</i>	43
3.2.1.1.1	Tela inicial	43
3.2.1.1.2	Tela de medição dos dados.....	44
3.2.1.1.3	Tela do mapa de cobertura.....	44
3.2.1.1.4	Telas de gráficos	45
3.2.1.1.5	Tela de pesquisa a pontos de interesse	48
3.2.1.1.6	Tela de estatísticas	48

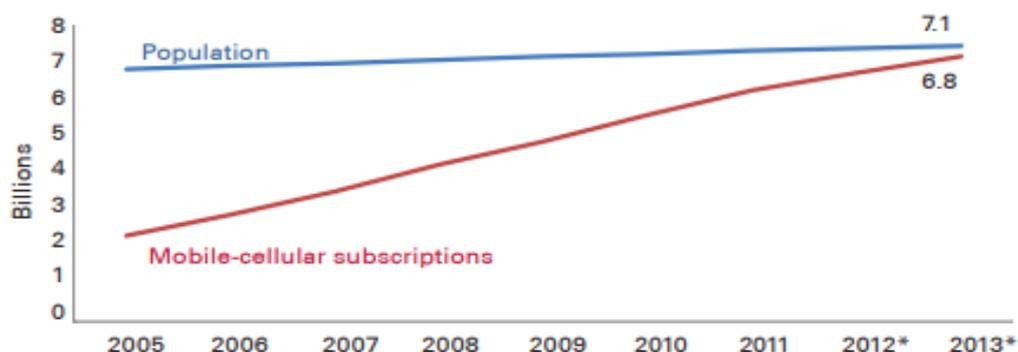
3.2.1.2	Acesso aos dados de interesse	49
3.2.1.2.1	Dados relacionados à qualidade das redes LTE	49
3.2.1.2.2	Dados sobre localização	52
3.2.1.3	Banco de dados.....	53
3.2.1.4	Gravação dos dados no banco de dados	56
3.2.1.5	API de mapas e consulta ao banco de dados	59
3.2.1.6	Geração de gráficos com os valores medidos	61
3.2.1.7	Busca de informações no banco de dados	62
3.2.1.8	Dados estatísticos	63
3.2.2	Medições	64
3.3	Considerações finais	67
4	RESULTADO E DISCUSSÕES	68
4.1	Resultado das medições no bairro Vila Industrial	70
4.1.1	Discussões sobre o resultado das medições no bairro Vila Industrial.....	73
4.2	Considerações finais	80
5	CONCLUSÃO, TRABALHOS FUTUROS E CONTRIBUIÇÕES	81
	REFERÊNCIAS	83

1 INTRODUÇÃO

É notável que nas últimas décadas o uso de telefones celulares tem crescido exponencialmente no Brasil e no mundo. Atualmente, 66% da população mundial utiliza o telefone celular, e até 2025 esse número chegará a 71% (GSMA, 2018). A utilização em massa desses sistemas se deve à capacidade que eles têm de atender um grande número de usuários em um grande espaço físico, além da possibilidade de se obter uma alta confiabilidade através de um bom projeto das redes utilizadas (RAPPAPORT, 1996). Porém, com o avanço dos anos, cada vez mais usuários utilizam essas redes móveis, e melhorias frequentes e significativas desses sistemas de comunicação são necessárias, o que se torna um desafio para as prestadoras desse serviço, que devem procurar cada vez métodos mais eficientes de avaliação da qualidade de suas redes móveis.

Junto com o crescimento do uso de *smartphones* e da internet em redes móveis, surgiu também a necessidade de maiores taxas de transmissão nas redes móveis, pois esses aparelhos atualmente são capazes de enviar imagens, vídeos e ainda permitem o acesso a diversas redes sociais que estão hospedadas na internet, o que traz um grande tráfego de dados (MENDES, 2014; ROLIN, 2014). O aumento na taxa de transmissão se tornou possível graças ao avanço das gerações de redes móveis. Devido a sua alta capacidade de transmissão de dados, as redes de quarta geração, ou 4G, são as redes que mais se adaptam as características requeridas pelo atual momento do mercado de comunicações móveis (ROLIN, 2014). Como pode ser visto na Figura 1, segundo a *International Telecommunication Union* (ITU), a taxa de crescimento do número de celulares é maior do que a taxa de crescimento da população mundial, o que poderá tornar as redes móveis cada vez mais sobrecarregadas e fazer crescer a busca pela melhoria de qualidade.

Figura 1: Taxa de crescimento da população e do número de celulares a nível mundial.



Fonte: ITU (2018).

O padrão *Long Term Evolution* (LTE) tem sido o mais utilizado dentre os disponíveis na quarta geração, pois apresenta uma maior qualidade de serviços, maior eficiência espectral, infraestruturas mais baratas, entre outras muitas características favoráveis ao LTE.

As operadoras devem seguir regras rígidas de qualidade que são definidas pela Anatel (Agência Nacional de Telecomunicações) (ANATEL, 2018a), e o não cumprimento dessas regras pode acarretar em multas altas para essas operadoras, além da perda de clientes para concorrentes que consigam prestar melhores serviços. Como exemplo, a Anatel assegura que as operadoras devem entregar aos usuários uma cobertura do serviço de telefonia móvel em pelo menos 80% das áreas urbanas dos municípios em que operam (ANATEL, 2018b). Outra regulamentação indica que sempre que uma operadora de telefonia celular prestar serviços de internet, padrões mínimos de qualidade devem ser respeitados, como a média mensal de velocidade, que não pode ser menor que 80% da ofertada ao usuário (ANATEL, 2018c). Existem outras várias regulamentações da Anatel para a fiscalização da qualidade de redes celulares, como o mínimo de potência de sinal que deverá chegar a um celular, entre outras.

Com o intuito de monitorar suas redes e garantir uma melhor qualidade e experiência para seu usuário, além de evitar penalidades perante as agências regulamentadoras de telecomunicações, as operadoras realizam diversos testes de campo para medir a qualidade do sinal em suas redes. Na maioria das vezes, essas medições são feitas através de *softwares* que geram vários tipos de resultados de análises de desempenho das conexões entre rede móvel e telefone celular. Porém o acesso aos aparelhos e aos *softwares* de medição é geralmente bastante limitado, pois eles apresentam altos custos de aquisição (ALBERO; ŠTERN; KOS, 2013).

Dois métodos muito utilizados pelas operadoras são o *drive-test* e o *walk-test*. Nesse tipo de teste de campo, medições sobre a conexão das redes celulares de várias tecnologias (GSM - *Global System for Mobile Communications*, LTE - *Long Term Evolution*, UMTS - *Universal Mobile Telecommunication System*, CDMA - *Code Division Multiple Access*, e etc.) são realizadas nas áreas de cobertura utilizando aparelhos de medição e *softwares* capazes de capturar e armazenar informações sobre essas redes, e que têm um custo de aquisição muito elevado. Estes testes permitem que as operadoras identifiquem os níveis de sinal em cada setor, interferências, alterações na rede e diversos outros parâmetros (ANATEL, 2018b), (TELECOM HALL BR, 2018). Uma forma simples e barata de se realizar um teste dessa natureza é utilizando *smartphones* conectados às redes móveis, pois através deles é

possível o desenvolvimento de aplicações para os mesmos fins que podem apresentar qualidade semelhante à de sistemas de medição profissionais.

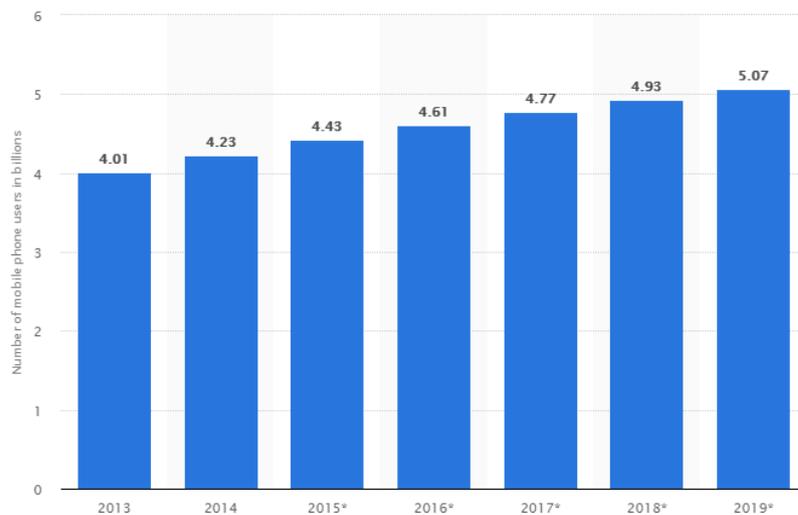
1.1. Tema do projeto

Esse trabalho visa o desenvolvimento de um *software* para avaliação da medição do desempenho e qualidade do sinal de redes móveis 4G, com ênfase em medições de redes do padrão LTE. O *software* foi desenvolvido para plataformas móveis e instalado em *smartphones* para realização de medições. O *software* desenvolvido tem funcionalidades auxiliares para a análise das informações medidas, como gráficos, cálculos estatísticos e mapas de cobertura.

1.2. Justificativas

Atualmente existem mais de 4 bilhões de telefones celulares no mundo (STATISTA, 2018), e como ilustrado na Figura 1, é esperado que esse número ultrapasse a barreira dos 5 bilhões em 2019.

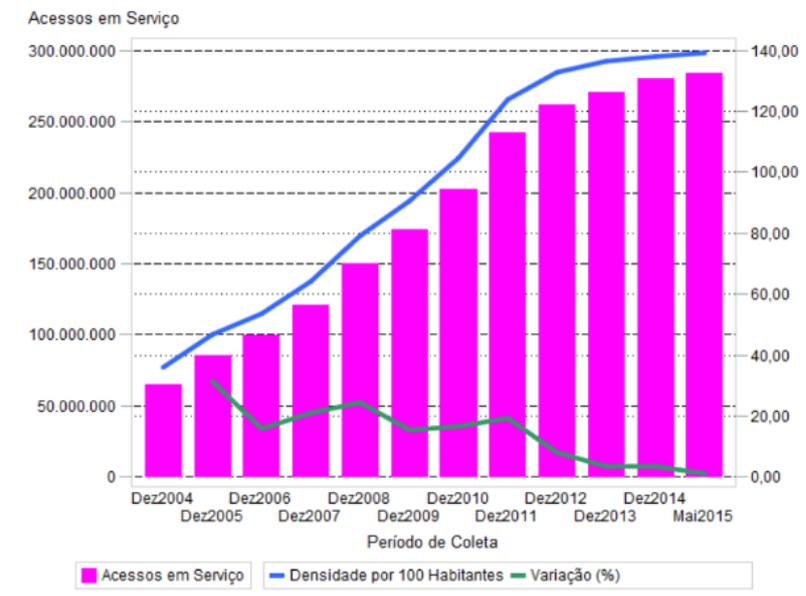
Figura 2: Quantidade de celulares no mundo.



Fonte: Statista (2018)

Só no Brasil já existem mais de 284 milhões de redes ativas (ANATEL, 2018d). Segundo um levantamento feito pela Anatel, esse número tem crescido de forma rápida, como ilustrado na Figura 3.

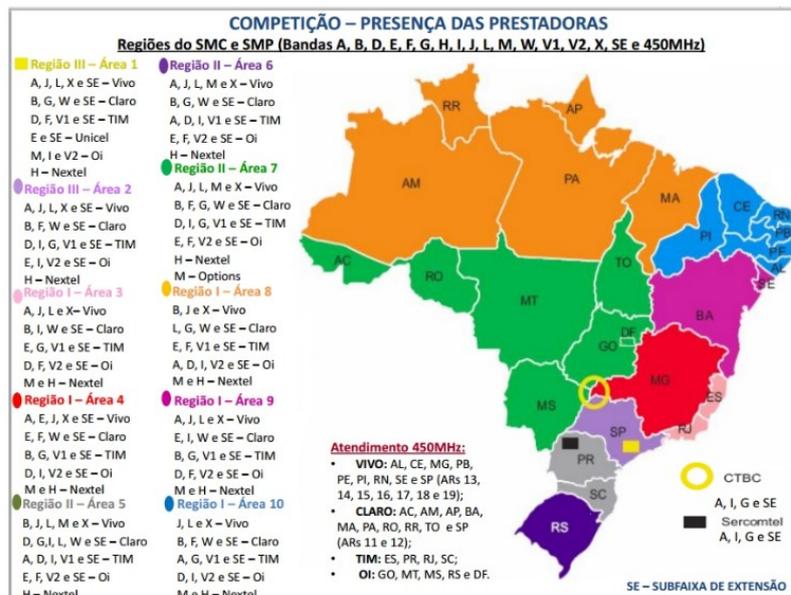
Figura 3: Crescimento do número de redes móveis no Brasil.



Fonte: Anatel (2018d).

Aliada à grande popularidade do telefone celular, a competição entre as operadoras também tem crescido, o que pode ser visto no mapa de competição das prestadoras de serviço elaborado pela Anatel, ilustrado na Figura 4. Uma análise sucinta da Figura 4 mostra que muitas operadoras prestam serviço na mesma área, o que faz com que a busca por melhorias para atrair mais usuários seja grande e constante.

Figura 4: Competição das prestadoras do serviço de telefonia móvel.



Fonte: Anatel (2018e)

A tendência no mercado da telefonia celular é de crescimento, assim como a evolução das tecnologias aplicadas, e isso eleva a competição entre as operadoras prestadoras desse serviço, que buscam sempre melhorar seus serviços para atrair mais usuários. A melhora das redes passa pela identificação de pontos negativos que possam ser solucionados, e, sendo assim, se torna justificável o desenvolvimento de *softwares* instalados em aparelhos de medição que sejam simples e baratos. Eles devem possibilitar a identificação de algumas características negativas das redes de telefonia móvel e faça uma análise qualitativa dos sinais recebidos por um telefone celular.

1.3. Objetivos

Considerando o grande avanço da tecnologia celular e a busca constante por melhorias na qualidade das redes móveis, esse trabalho tem como objetivo geral o desenvolvimento de um *software* para a análise qualitativa do sinal de comunicações móveis. A análise em redes LTE se deve a crescente taxa de usuários que utilizam serviços de internet em seu celular, e que assim, passam a utilizar as redes 4G LTE, que atualmente são as redes que contém a maior qualidade desse tipo de serviço entre as utilizadas no Brasil (MENDES, 2014).

O *software* desenvolvido deve ser capaz de capturar informações sobre a qualidade e desempenho do sinal recebido em telefones celulares. A execução dos testes é feita por meio de *drive-tests* e *walk-tests* aplicados às redes comerciais existentes nos locais de teste escolhidos. Tem-se ainda como objetivo o desenvolvimento de ferramentas extras para o *software*, como a plotagem de gráficos, cálculos de dados estatísticos e a fabricação de mapas de cobertura que facilitem a elaboração de relatórios sobre as informações capturadas.

1.4. Problematização

A busca constante por melhorias que é observada no mercado de telefonia móvel faz com que as operadoras busquem meios para avaliar e testar o sinal proveniente de suas redes. Levando em consideração a alta taxa de crescimento desse mercado, o presente trabalho busca responder ao problema de pesquisa: Existe algum meio simples e barato de se coletar dados das redes móveis que torne possível a avaliação do sinal proveniente do serviço de redes móveis prestado pelas operadoras?

1.5. Hipóteses

Hipótese 1 – A vegetação e as construções civis em torno do ponto de medição, além do relevo, são fatores extremamente limitantes à qualidade do sinal, pois são capazes de produzir atenuações severas no mesmo (RAPPAPORT, 1996).

Hipótese 2 – Utilizando somente um *smartphone* como forma de colher informações do sinal de redes móveis, além de *softwares* auxiliares, é possível analisar de forma satisfatória a qualidade do sinal em qualquer localidade que tenha cobertura da operadora avaliada.

1.6. Considerações finais

O *software* proposto foi desenvolvido para a plataforma Android devido à facilidade de acesso às ferramentas de código aberto que o Google fornece para os desenvolvedores dessa plataforma, e pelo menor custo dos equipamentos que utilizam esse sistema (ALBERO; ŠTERN; KOS, 2013). As redes avaliadas são redes 4G de padrão LTE, por serem as redes mais atuais aplicadas no Brasil.

Através do site www.developer.android.com, o Google disponibiliza milhares de APIs que auxiliam os desenvolvedores em seus projetos. Dentre elas, existe uma API (*Application Programming Interface* - Interface de Programação de Aplicativos) chamada “*android.telephony*”, que fornece informações relevantes sobre a rede de telefonia celular em que o aparelho está conectado e foi utilizada para o desenvolvimento do *software* de captura do sinal (DEVELOPERS, 2018c).

2 DESENVOLVIMENTO TEÓRICO

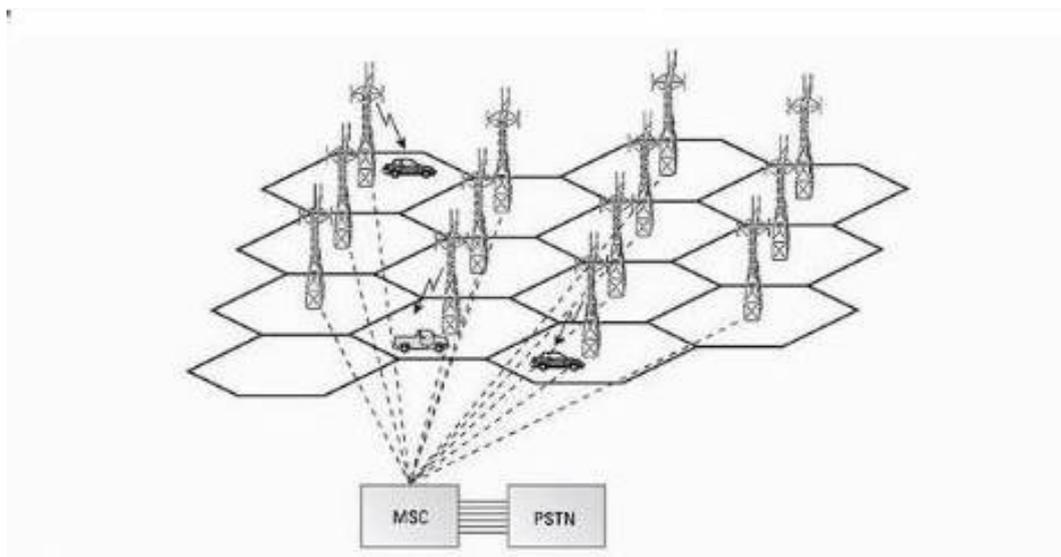
Neste capítulo serão abordados alguns tópicos sobre redes móveis, com ênfase em redes 4G padrão LTE. Outros tópicos sobre o desenvolvimento de *software* para o sistema operacional Android com foco na captura de dados que possam ser utilizados na análise da qualidade de redes móveis também serão apresentados.

2.1 Introdução à telefonia celular

Sistemas de telefonia celulares são sistemas capazes de fornecer uma ligação de telefones às PSTN (*Public switched telephone network* - rede telefônica mundial comutada por circuitos) através de uma conexão sem fio, desde que esses telefones estejam no alcance de rádio do sistema em uso (RAPPAPORT, 1996).

A principal característica que dá nome a esse tipo de sistema é a forma como a área geográfica de cobertura é dividida: divide-se a área de cobertura total em áreas menores, chamadas de células, onde cada célula deve ter um transmissor próprio. A Figura 5 faz uma ilustração desse tipo de sistema.

Figura 5: Ilustração de um sistema de telefonia celular básico.



Fonte: Rappaport (1996).

Como é possível notar na Figura 5, um sistema celular básico é constituído por estações móveis que se conectam com estações-base, que são representadas pelas antenas de cada célula. Cada estação-base está conectada à MSC (*Mobile Switching Centre* – Centro de

comutação móvel), que é o ponto responsável por fazer a ligação de todas as estações-base à PSTN (RAPPAPORT, 1996).

A divisão em pequenas células, cada uma com um transmissor, como citado anteriormente, dá a esse tipo de sistema a possibilidade de atender muitos usuários em uma área grande, o que poderia ser um problema, devido ao espectro limitado que existe em qualquer tipo de meio de telecomunicações. Essa possibilidade se deve à reutilização de frequências que esse tipo de sistema possui, pois como o alcance do transmissor de cada célula é limitado a uma pequena área geográfica, a frequência de transmissão utilizada nessa célula pode ser reaproveitada em outra célula que esteja longe o bastante para não causar interferências (RAPPAPORT, 1996).

A Figura 6 ilustra a comunicação de um ponto de vista detalhado. Como existem muitas células e o usuário pode estar se movendo constantemente usando o telefone móvel, existe a possibilidade de que ele troque várias vezes de estações-base. Isso só é possível devido ao *handoff*, uma tecnologia que consegue perceber quando um telefone celular está passando de uma célula para outra, ou quando o sinal proveniente de uma célula vizinha está mais forte do que o sinal da célula em que o telefone está conectado atualmente, e faz a comutação da conexão para a outra célula sem que o usuário perceba.

Figura 6: Sistema de telefone sem fio.



Fonte: Rappaport (1996).

Como exemplo da capacidade deste sistema, uma única MSC é capaz de tratar de 100 mil usuários e realizar 5 mil chamadas telefônicas de forma simultânea (RAPPAPORT, 1996).

2.2 As diferentes gerações de telefonia móvel

O sistema de telefonia móvel mostrado na Seção 2.1 faz uma tratativa simples a respeito do tema, porém com o passar do tempo várias melhorias foram desenvolvidas para se alcançar mais capacidade e agregar mais serviços ao aparelho celular. Atualmente, o telefone

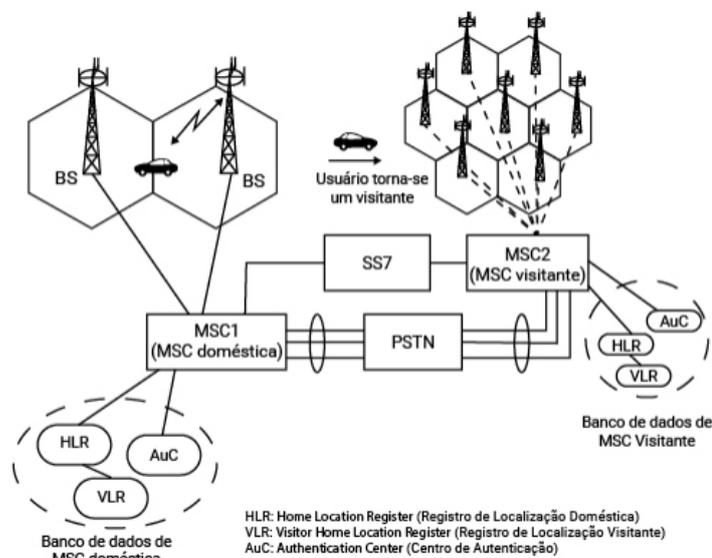
celular é capaz de executar praticamente todas as funções de um computador, e isso se deve ao rápido e gigantesco avanço da tecnologia nessa área, tanto do lado dos fabricantes de aparelhos celulares, quanto no lado das operadoras de telefonia de celular, que atualmente prestam serviços de internet de altíssima velocidade.

Foram desenvolvidas 4 gerações diferentes de tecnologia de redes móveis desde que a primeira foi desenvolvida. A quarta geração é a última desenvolvida e a mais utilizada atualmente. A terceira ainda é amplamente utilizada, sendo que as duas tem um grande destaque na evolução das redes móveis e tornaram possível a transferência de dados através da internet. Porém, nada disso seria possível se não fosse o grande avanço trazido pelas duas primeiras tecnologias. A seguir uma apresentação simplificada de cada geração de telefonia móvel será feita.

2.2.1 Primeira geração - 1G

A primeira geração de telefonia móvel é conhecida como AMPS (*Advanced Mobile Phone System* – Sistema Avançado de Telefonia Móvel). Esse tipo de sistema tinha uma capacidade limitada, pois apresentava pouco espaço para crescimento do espectro (DEBNATH; ROBERT, 2013). No AMPS, a transmissão de voz era feita de forma analógica, seguindo uma modulação do tipo FM (*Frequency modulation* – Modulação em frequência), e a transmissão de dados era bastante ineficaz (RAPPAPORT, 1996). A Figura 7 ilustra a arquitetura básica de um sistema AMPS.

Figura 7: Esquema de um sistema AMPS básico.



Fonte: Rappaport (1996).

Como se pode observar por meio da Figura 7, a MSC é responsável por todo o controle dentro do seu mercado, como controle de chamada, transferência de célula, além da troca de sinalização pelo canal SS7, entre outras muitas responsabilidades que ficam encarregadas às MSCs (RAPPAPORT, 1996).

2.2.2 Segunda Geração – 2G

A segunda geração trouxe melhoras expressivas nas redes móveis com um aumento considerável de sua capacidade. Em comparação ao AMPS, as redes 2G atendiam um número muito maior de clientes em uma mesma área geográfica e com maior qualidade, devido principalmente a técnicas de digitalização dos sinais que passou a ser utilizada (MENDES, 2014).

A codificação de voz também foi incluída nesses sistemas, e em conjunto com as técnicas de digitalização permitiu que os sinais tivessem um tamanho menor, o que diminuiu a carga computacional e trouxe um aumento na capacidade de transmissão do sistema (RAPPAPORT, 1996).

2.2.3 Terceira geração – 3G

Apoiadas no avanço da tecnologia digital, as redes de terceira geração foram desenvolvidas. Essas redes, através do acesso à internet, tornaram possíveis as transmissões com qualidade superior de voz, imagens, vídeos e etc. (IESB, 2018).

Entre as características dessas redes, o acesso à internet de banda larga foi a melhoria mais importante de todas, pois tornou possível a transmissão de dados e trouxe um grande avanço no mercado das comunicações móveis, fazendo com que as grandes empresas disputassem e explorassem cada vez mais esse mercado. Além do uso da internet, uma excelente qualidade de voz foi adicionada, além de importantes serviços online (MENDES, 2014).

2.2.4 Quarta geração – 4G

Através das redes 4G, o acesso à internet se tornou ainda mais presente no meio das comunicações móveis. Nesse tipo de tecnologia, os usuários podem ter milhares de possibilidades, podendo escolher entre diversas aplicações e ambientes. As premissas dessa tecnologia ditam que os usuários poderão acessar as informações em qualquer lugar e a qualquer hora, enviando e recebendo um volume grande de informações, dados, fotos, vídeos, entre outros (DEBNATH; ROBERT, 2013).

As redes 4G têm uma eficiência e qualidade 10 vezes maiores que a anterior 3G, sendo isso possível através do uso de um conjunto de redes que utiliza o protocolo IP (*Internet Protocol* - Protocolo de Internet) e fazem o acesso de alta qualidade à internet. A transmissão é feita por um transmissor OFDM (*Orthogonal frequency-division multiplexing* - Multiplexação por Divisão de Frequência Ortogonal) em conjunto com uma rede IP, que traz grande qualidade de comunicação sem muito atraso na escolha dos multi-caminhos, graças também às técnicas avançadas de MIMO (*Multiple-input and multiple-output* - Múltiplas-Entradas e Múltiplas-Saídas) (MENDES, 2014).

2.3 Redes LTE

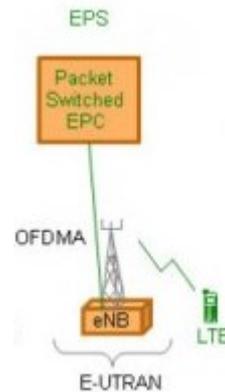
O padrão LTE foi denominado pelo grupo 3GPP através do *release 8* com o intuito de realizar melhoras nas especificações do 3G (ROLIN, 2014). Os principais requisitos levantados para o LTE foram os seguintes (3GPP, 2018a), (3GPP, 2018b):

- Necessidade de garantir a continuidade da competitividade do sistema 3G para o futuro;
- Atender a demanda do usuário por taxas de dados e qualidade de serviço mais alta;
- Busca por um melhor sistema de comutação por pacotes;
- Demanda contínua por redução de custos;
- Baixa complexidade;
- Necessidade de variadas opções de largura de banda;
- Alta eficiência espectral;
- Altas taxas de dados de pico;
- Flexibilidade na frequência e na largura de banda.

2.3.1 Arquitetura da rede LTE

LTE (*Long Term Evolution*) ou E-UTRAN (*Evolved Universal Terrestrial Access Network*) é parte do EPS (*Evolved Packet System*), que é a arquitetura resultante. O EPC (*Evolved Packet Core*) que é a parte de acesso ao núcleo das redes IP, também faz parte do EPS. Houve também a introdução do método de acesso OFDMA (*Orthogonal Frequency Division Multiplexing Access*) como interface de acesso de rede aérea. A Figura 8 mostra um esquema da arquitetura EPS do padrão LTE (ROLIN, 2014), (3GPP, 2018b).

Figura 8: Arquitetura de uma rede LTE.



Fonte: 3GPP (2018b)

Observando a figura 8, nota-se que a arquitetura do padrão LTE tem 3 componentes principais, que são o E-UTRAN (*Evolved Universal Terrestrial Radio Access Network*), o EPC e o equipamento do usuário. Nas subseções seguir, um detalhamento de cada um desses componentes será feito.

2.3.1.1 E-UTRAN

A conexão do usuário com a rede LTE começa através da E-UTRAN quando o equipamento do usuário se comunica com as eNB, ou eNodeB (*Enhanced Base Station - Estações de Base Melhoradas*). Diferentemente da geração anterior, onde as tarefas de processamento eram feitas no RNC (*Radio Network Controller*), o processamento de informações é todo feito nos eNodeB para trazer uma maior simplificação da arquitetura (PUC-RIO, 2018). Ao eNodeB são encarregadas as seguintes funções:

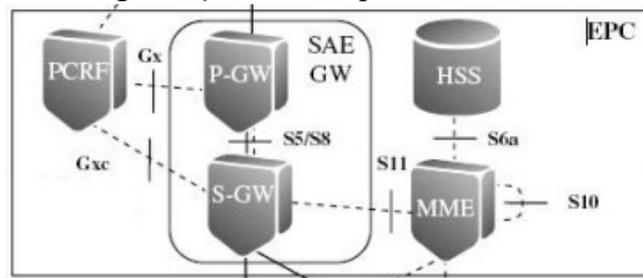
- Transmissão dos dados: Realização da interface de rádio para que os usuários se conectem as redes EPC. O eNodeB é responsável também pela modulação e demodulação dos sinais, e pela codificação e decodificação do canal (ROLIN, 2014).
- Coordenação da interferência Inter-cell: Controle da interferência entre células através da troca de informações entre os diferentes canais de rádio, associadas às diferentes células (ROLIN, 2014).
- Balanceamento da carga: Esta função garante a qualidade do serviço aos usuários. O eNodeB, considerando o tráfego momentâneo toma decisões para que, se necessário, seja feito o *handover* para a distribuição da carga entre as células, para que nenhuma célula fique congestionada (ROLIN, 2014).

- Sincronização: Cada eNodeB possui um circuito que possibilita a recepção de sinais de maneira aleatória ao método de sincronismo escolhido (ROLIN, 2014).
- Mobilidade: Quando o terminal está no modo ativo, o eNodeB gera a mobilidade.
- *Paging*: Esta função faz com que o equipamento do usuário contate a E-UTRAN quando o terminal está no modo de espera. Se ele estiver ativo, uma mensagem de aviso é endereçada (ROLIN, 2014).

2.3.1.2 EPC

O EPC é uma rede de alta taxa de transmissão baseada em IP com grande qualidade e desempenho. Ela trouxe uma enorme evolução em relação a geração antecessora, pois tornou possível o uso da comutação por pacotes e protocolo IP, enquanto na geração anterior a comutação era por circuitos. A Figura 9 mostra a organização e os componentes de uma rede EPC (ROLIN, 2014).

Figura 9: Organização dos componentes de uma rede EPC.



Fonte: Rolin (2014).

Os componentes principais de uma rede EPC e suas responsabilidades são:

- *Mobility Management Entity* (MME): Sinalização do equipamento do usuário e o núcleo da rede, processos de *paging* e autenticação de usuário.
- *Serving-Gateway* (S-GW): Responsável pela conectividade com redes de dados externas, além de encaminhar pacotes para o eNodeB e para o P-GW.
- *Packet Data Network Gateway* (P-GW): Responsável pelo transporte e direcionamento dos pacotes IP. É por meio dele que os dados entram e saem do sistema.
- *Policy and Charging Rules Function* (PCRF): responsável pela tarifação e controle de QoS (*Quality of Service* – Qualidade de Serviço) de cada usuário, pelas políticas de tomada de decisão e pelo controle das funcionalidades.

2.4 Parâmetros de desempenho e qualidade

O equipamento do usuário e os eNodeB, que representam a camada física, estão constantemente realizando medições nas redes, e transmitindo-as para as camadas superiores para que decisões importantes sejam tomadas e se mantenha a qualidade do sinal. A seguir os parâmetros importantes para esse tipo de medição serão detalhados.

2.4.1 *Reference Signal Reference Power*

O *Reference Signal Reference Power* (RSRP) é uma das mais importantes medidas do nível de qualidade do sinal para redes LTE, sendo também uma das medições mais básicas. O RSRP indica a potência média do sinal em toda a largura de banda do canal. Existe outro parâmetro chamado *Reference Signal Reference Quality* (RSRQ), que em conjunto com o RSRP é utilizado por algoritmos no cálculo para que o equipamento do usuário tome a decisão sobre qual célula tem o melhor nível de qualidade, sendo assim um importante parâmetro para o *handover*. O RSRP é medido em decibel miliwatt (dBm), e tem valores comuns entre -44 dBm e -140 dBm (ROLIN, 2014).

Os valores de RSRP são divididos em 3 faixas para que se possa realizar a análise de QoS (ROLIN, 2014). A determinação das faixas é a seguinte:

- Maior que -75 dBm, qualidade excelente;
- Entre -75 dBm e -95 dBm, qualidade aceitável;
- Menor que -95 dBm, qualidade inaceitável.

2.4.1.1 **Resolução da Anatel quanto ao limiar mínimo de RSRP**

A Anatel, através da Portaria nº 1718, de 07 de dezembro de 2017, definiu os seguintes valores como limite mínimo de RSRP que as operadoras podem entregar aos usuários da tecnologia LTE (ANATEL, 2018g):

- VP1: percentual de cobertura considerando o limiar de recepção do RSRP de -110 dBm;
- VP2: percentual de cobertura considerando o limiar de recepção do RSRP de -102 dBm.

2.4.2 Reference Signal Reference Quality

O *Reference Signal Reference Quality* (RSRQ) indica a existência de interferências na rede, e também é utilizado como parâmetro nos cálculos do *handover*, sendo um parâmetro importante para análises de qualidade (ROLIN, 2014). O RSRQ é medido em dBm, e seus valores comuns variam entre -3 dBm e -19,5 dBm (ROLIN, 2014). Assim como no RSRP, existem faixas de valores para que se possa realizar a análise de QoS. As faixas são as seguintes:

- Maior que -9 dBm, boa qualidade;
- Entre -12 dBm e -9 dBm, qualidade aceitável;
- Menor que -13 dBm, qualidade inaceitável.

Os resultados de RSRQ e RSRP podem ser utilizados em conjunto para uma análise de cobertura e interferência em um mesmo local. Se o RSRP e o RSRQ forem medidos em um mesmo ponto geográfico por um UE (*User Equipment* - Equipamento do Usuário), e com uma mudança de condições ou posição o valor de RSRP se manter constante e o de RSRQ sofrer degradação, essa situação indicara que existem ruídos e interferências na rede neste local. Se os dois valores sofrerem degradação, então isso indicará falta de cobertura ou problemas graves na rede local (ROLIN, 2014).

2.4.3 Reference Signal Strength Indicator

O *Reference Signal Strength Indicator* (RSSI) é utilizado para medir a força do sinal, e indica toda a energia dentro da banda passante do sinal. Diferentemente do RSRP que mede a potência média do sinal, o RSSI mede a potência do sinal de forma eficaz, considerando todas as subportadoras ativas existentes (ROLIN, 2014).

O RSSI é um parâmetro também de medição da potência total da banda larga, com a presença de interferências e ruídos térmicos, e têm relação direta com a transferência de dados na rede, pois, quanto maior a transferência de dados, maior será o valor de RSSI (ROLIN, 2014). Os valores de RSSI podem variar entre -35dBm e -85dBm.

2.4.4 Signal to Interference And Noise Ratio

O parâmetro *Signal To Interference and Noise Ratio* (SINR) também é uma medida de qualidade do sinal, que, porém, não é definida pelo 3GPP. O SINR é bastante utilizado pelas operadoras, pois relaciona de uma forma melhor a relação entre as condições de RF

(Radiofrequência) e o *Throughput*. Apesar de não ser um parâmetro definido pelo 3GPP, é por vezes utilizado nos *drive-tests* (AFROZ, 2015).

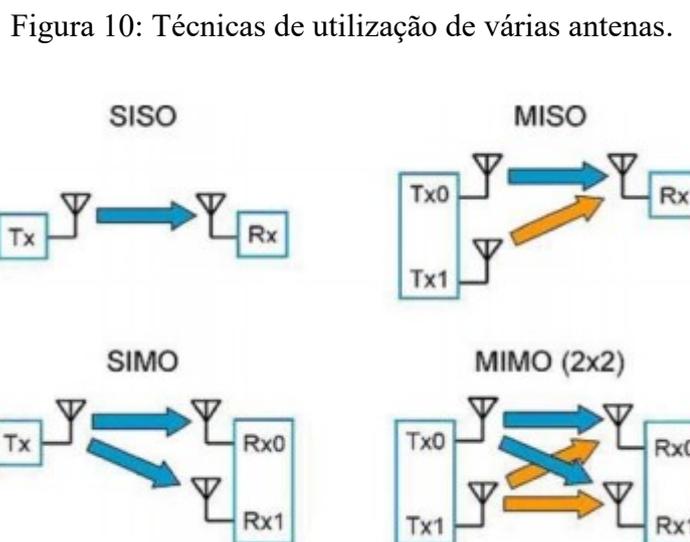
A maior aplicação do SINR está no cálculo do CQI (*Channel Quality Indicator* - indicador de qualidade de canal), onde as UE calculam esse valor e o reporta aos eNodeB. Com posse do valor de CQI, o eNodeB seleciona o esquema de código de modulação e codificação mais adequado, definindo o número de símbolos que serão utilizados na comunicação (AFROZ, 2015).

Em outras palavras, o SINR mede a potência do sinal e a soma do poder médio de interferência das outras células e do ruído de fundo, além de outras transmissões simultâneas, e pode ser chamado no 4G de *Reference Signal Signal to Noise Ratio* (RSSNR) (AFROZ, 2015).

2.4.5 Técnicas de múltiplas antenas

A técnica de múltiplas antenas é utilizada para melhorar o desempenho da rede. Com o uso dessas técnicas é possível o aumento da taxa de transmissão e da eficiência espectral. Essa prática tem como base o fenômeno dos múltiplos caminhos, onde os sinais provenientes de várias antenas transmissoras poderão percorrer caminhos diferentes e chegar a uma receptora, e através de uma combinação correta dos diferentes sinais que chegam às receptoras, é possível se obter uma melhor qualidade (MICROWAVE JOURNAL, 2018), (ROLIN, 2014).

A Figura 10 mostra as principais formas de utilização da técnica de múltiplas antenas.



Fonte: Rolin (2014).

A sigla SI indica uma entrada única, SO uma saída única, MI indica entrada múltipla e MO saída múltipla.

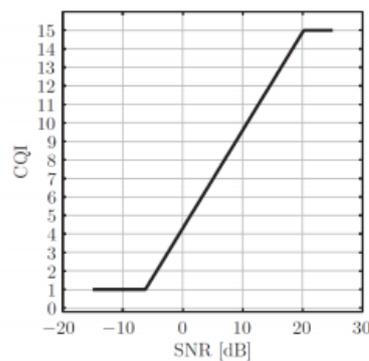
A técnica de MIMO é a mais utilizada no padrão LTE, pois através dela um aumento considerável das taxas de transmissão é obtido (ROLIN, 2014). A técnica MIMO explora as técnicas de múltiplo caminho, para transmissão e recepção, sendo que no receptor uma combinação correta dos vários sinais recebidos deve ser feita para o aumento da qualidade do sinal.

2.4.6 *Channel Quality Indicator (CQI)*

O *Channel Quality Indicator (CQI)*, como já citado na seção 2.4.3 deste trabalho, pode auxiliar na seleção do esquema de modulação e codificação em uma eNodeB, e é uma informação derivada do sinal de qualidade.

O CQI é calculado através do SINR, e como o SINR é uma medida não definida pelo 3GPP, essa medida varia em todos os modelos de UE, e é influenciada pela qualidade de cada um. Por exemplo, se um UE utilizar algoritmos avançados de processamento do sinal, provavelmente pode receber uma taxa maior de transmissão. A relação entre o CQI e a relação sinal ruído pode ser observada na Figura 11.

Figura 11: Mapeamento CQI x SNR.



Fonte: Rolin (2014).

O CQI varia entre valores de 1 a 15, como visto na Figura 11. Quanto mais próximo de 15, maior será a qualidade do sinal, e maior será o valor de SINR (*Signal Noise Ratio – Relação Sinal Ruído*), que representa a relação entre o sinal e o ruído, ou seja, a amplitude da potência do ruído será menor que a amplitude da potência do sinal, o que indica uma boa qualidade de sinal.

2.4.7 *Throughput*

O *throughput* pode ser definido como a medida de velocidade de transferência de dados em um sistema, definida pela relação entre a quantidade de dados transferidos de um lugar a outro e o tempo gasto para a transferência.

2.5 **Aparelhos e *softwares* de medição comerciais**

Atualmente, os aparelhos mais utilizados para realização de testes em níveis comerciais são *scanners* e *modems*, que trabalham sempre em conjunto com *softwares* que extraem resultados técnicos dos dados que são medidos.

2.5.1 *Scanner*

Um *scanner* é um aparelho capaz de capturar informações sobre parâmetros de qualidade da rede, e é um grande aliado na análise de QoS.

2.5.1.1 *Scanner Rohde & Schwarz*[®] TSMW

Esse aparelho é um analisador de rede da marca Rohde & Schwarz[®]. Na Figura 12 pode-se observar uma foto real do TSMW.

Figura 12: *Scanner* TSMW.



Fonte: Microwave Journal (2018).

Segundo o fabricante, sua arquitetura é definida por *software* baseada em FPGA, com alto desempenho e máxima flexibilidade, além de prontidão operacional. Por meio da folha de dados do fabricante, observa-se que os seguintes dados de parâmetro de qualidade são aferidos por ele são RSRP, RSRQ, RSSI, SINR, Doppler, CP (*Cyclic Prefix*), CN (*Core Network*) (ROHDE E SCHWARZ, 2018).

2.5.2 *Modem Huawei*[®] E392

Esse aparelho é utilizado para fazer conexão com a rede 4G nos testes de redes LTE. Segundo dados do fabricante, quando conectado à porta USB de um computador em uma área de serviço LTE, HSPA +, UMTS, EDGE, GPRS, GSM, é possível navegar na internet e enviar, receber mensagens ou e-mails, através das redes sem fio (4G LTE MALL, 2018).

Para realização de um *drive-test*, um exemplo de rotina de tarefas de um *modem*, segundo Rolin (2014), seria a seguinte:

- Início do *script*;
- Inicia no modo idle. Após 5 s entra automaticamente no modo ativo (*call*);
- Conexão de dados com 10 sessões em paralelo;
- Realizar um *download* de arquivo de 1 GB;
- Ao término do *download* termina a sessão;
- Realizar um *upload* de arquivo de 1 GB;
- Ao término do *upload*, termina a sessão;
- Retornar ao início do *script*.

A Figura 13 mostra uma foto real desse *modem*.

Figura 13: *Modem Huawei* 392.



Fonte: 4G LTE MALL (2018).

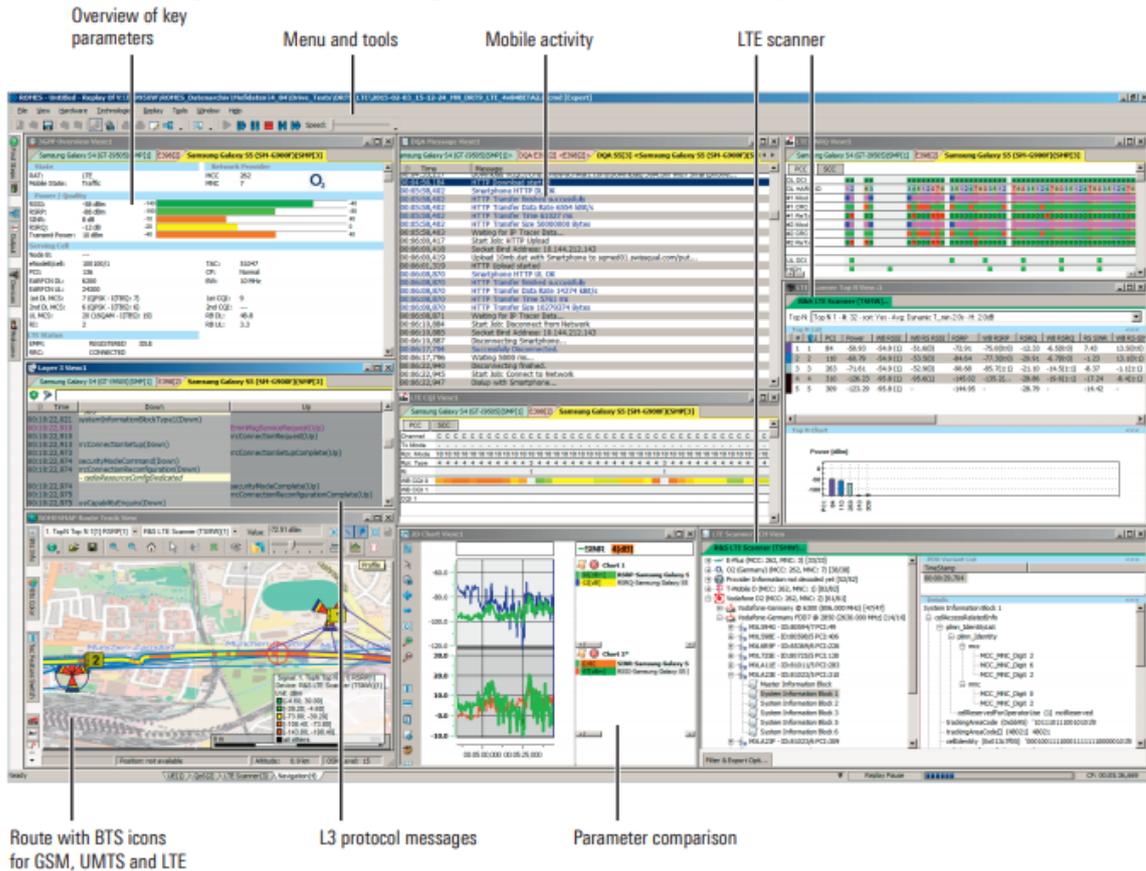
2.5.3 *Software Rohde & Schwarz*[®] Rohmes 4

Esta ferramenta é um *software* que trabalha em conjunto com aparelhos de medição, e que gera resultados a cerca das medições realizadas por *scanners* ou aparelhos móveis de forma instantânea. Fornece resultados e soluções para problemas como falta de cobertura, identificação de interferências, além de realizar teste de QoS completos. Este *software* é ainda

capaz de receber os dados e processá-los instantaneamente para calcular as estatísticas em tempo real (ROHDE E SCHWARS, 2018).

A Figura 14 mostra a interface gráfica deste *software*.

Figura 14: Interface gráfica do usuário com o *software* Rohmes.



Fonte: Rohde e Schwarz (2018).

2.5.4 Software Rohde & Schwarz® Qualipoc android

Este é um *software* desenvolvido pela Rohde e Schwarz que realiza a medição de alguns parâmetros de qualidade de rede. Segundo o fabricante, este *software* tem funções de testes de voz, dados, vídeo e mensagens, e como os testes são desenvolvidos em um *smartphone* real, acredita-se que a experiência seja a mais próxima possível da experiência de usuários reais das redes. A Rohde e Schwarz cita ainda a facilidade que este *software* traz para engenheiros que projetam redes de rádio frequência, que podem testar de forma rápida os níveis de qualidade do sinal devido à portabilidade trazida por um *smartphone*. A Figura 15 mostra a interface deste *software* com o usuário

Figura 15: Interface do *software* QualiPoc.



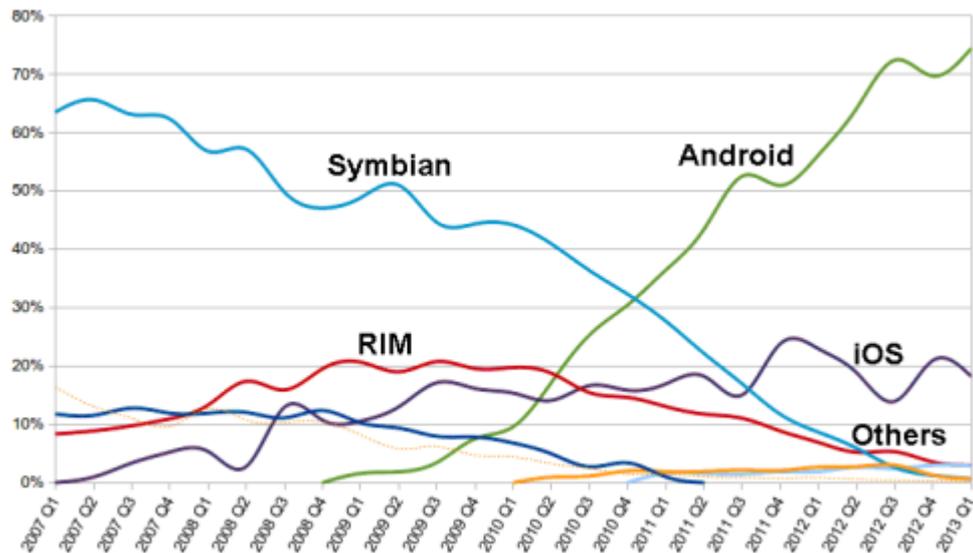
Fonte: Rohde e Schwarz (2018).

Analisando a Figura 15, nota-se que o *software* realiza uma análise através de dados citados na seção 2.4 deste trabalho, como RSRP e RSRQ, que são dois dos mais importantes parâmetros para análise da qualidade.

2.6 Desenvolvimento de *software* para medição da qualidade do sinal

O Android é um sistema operacional presente em smartphones de diversas marcas, e muito difundido atualmente. A Figura 16 mostra o crescimento de venda dos telefones com sistema Android a nível mundial.

Figura 16: Venda de *smartphones* no mundo.



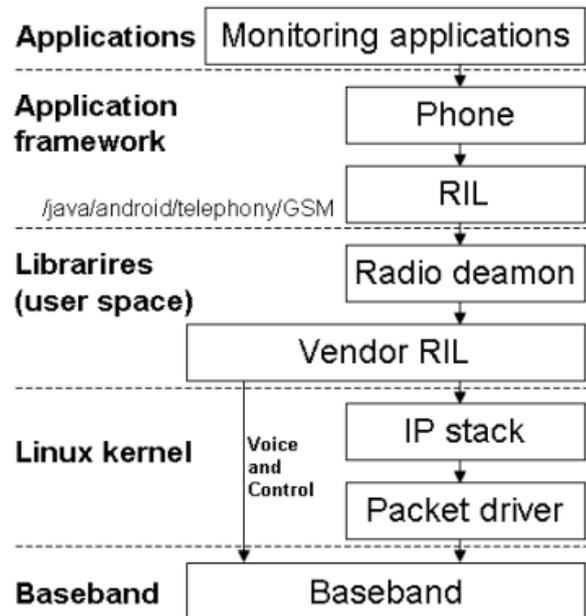
Fonte: Albero (2014).

Como se pode observar, a venda de *smartphones* com Android supera a venda de *smartphones* com qualquer outro sistema operacional. O crescimento da comunidade de desenvolvedores para o sistema Android e o seu mercado cresce em conjunto com o número de usuários (ALBERO; ŠTERN; KOS, 2014). O crescimento da comunidade de desenvolvedores traz a eles vários benefícios, como fóruns de discussão entre os desenvolvedores, várias aplicações de código aberto e etc.

2.6.1 Arquitetura Android e acesso à camada de rádio

O Android pode ser considerado uma pilha de *software* que contém várias camadas baseadas em Linux de código aberto (DEVELOPERS, 2018a). Cada camada pode ser vista como um grupo de componentes cooperativos do sistema operacional, e juntas incluem o sistema operacional, o *middleware* e os aplicativos do sistema.

Para que medições de qualidade do sistema sejam feitas, o acesso às informações da camada de rádio do sistema deve ser realizado. A camada que fornece uma interface de rádio é a chamada *Radio Interface Layer (RIL)*. A Figura 17 mostra um esquemático do RIL.

Figura 17: *Radio Interface Layer*.

Fonte: Albero (2013).

Para se acessar as informações necessárias do RIL, deve-se utilizar códigos abertos do Android e APIs que são disponibilizadas no site <https://developer.android.com/reference/>, que é a página oficial do sistema operacional Android (DEVELOPERS, 2018c). Utilizando a API “android.telephony” e suas diversas classes, o acesso às informações do *hardware* de rádio do sistema é possível.

A camada RIL mostrada na Figura 17 tem dois principais componentes, o *Radio Daemon* e o *Vendor RIL*. O *Daemon* se comunica com os serviços de telefonia e os envia até o *Vendor RIL*, que os despacha para as camadas inferiores (ALBERO; ŠTERN; KOS, 2014).

A última versão da API “android.telephony” contém várias classes que podem ser utilizadas para se capturar informações sobre a qualidade do serviço de telefonia móvel de quarta geração e padrão LTE. As principais classes que podem ser utilizadas para essa finalidade estão presentes no Quadro 1, que contém também a sua descrição segundo a página do Android.

Quadro 1: Classes da API “android.telephony” para análise da qualidade do sinal 4G LTE.

Classe	Descrição
SignalStrength	Contém informações relacionadas à força do sinal do telefone
CellInfoLte	Informações sobre uma célula em um ponto.
CellSignalStrengthLte	Informações relacionadas à força do sinal LTE
TelephonyManager	Fornecer acesso a informações sobre os serviços de telefonia no dispositivo

Fonte: Developers (2018c).

Analisando a documentação da classe CellSignalStrengthLte, que está disponível no site <https://developer.android.com/reference/android/telephony/CellSignalStrengthLte>, é possível observar que muitos parâmetros citados na Seção 2.4 deste trabalho como importantes para análise da qualidade de serviço de redes LTE podem ser obtidos através de alguns métodos da classe. Esses métodos são os seguintes:

- getDBm: obtém o valor de potência sinal momentâneo em dBm;
- getRsrq: obtém o valor de RSRQ;
- getRssnr: obtém o valor de RSSNR;
- getRsrp: obtém o valor de RSRP;

Através do uso das classes citadas no Quadro 1 e seus métodos, a análise de desempenho e qualidade de redes LTE se torna possível. Somente através de códigos desenvolvidos para Android não é possível realizar uma análise de todos os parâmetros utilizados em QoS em testes profissionais, porém um teste de grande qualidade, com os parâmetros mais relevantes para o funcionamento destas redes poderá ser feito.

2.7 Considerações finais

Existem vários parâmetros que são utilizados nas análises de redes móveis, e várias possibilidades de análise deles, porém o uso de equipamentos avançados além de somente um *smartphone* é necessário para que se possam acessar todos os parâmetros e variáveis, como *scanners* e *modems*. Todavia, o uso de um *smartphone* com sistema operacional Android como aparelho de medição já traz uma grande gama de possibilidades e parâmetros que tornam possível uma boa análise quando se faz uso das APIs disponibilizadas para o acesso às informações da interface de rádio do *smartphone*.

3 MATERIAIS, TECNOLOGIAS E MÉTODOS

3.1 Materiais e tecnologias

A seguir os recursos necessários para a execução do trabalho serão mostrados, assim como as principais tecnologias utilizadas para o desenvolvimento do *software* e como eles são aplicados no trabalho em desenvolvimento.

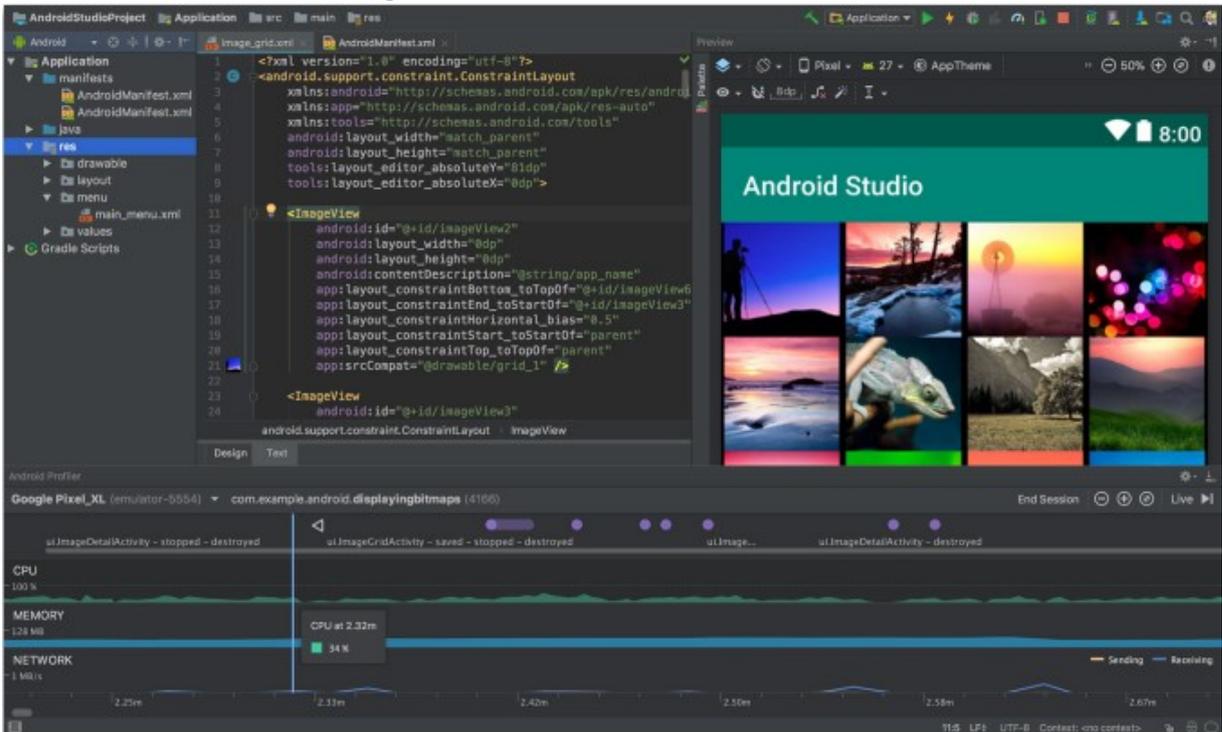
3.1.1 Android Studio

Para o desenvolvimento do *software* de medição foi utilizado o ambiente de desenvolvimento integrado chamado Android Studio. Este ambiente é gratuito, e pode ser obtido pelo link <https://developer.android.com/studio/>. Segundo Developers (2018d), o Android Studio fornece recursos como:

- Sistema de compilação flexível baseado no *Gradle*;
- Emulador rápido com inúmeros recursos;
- Ambiente unificado para desenvolvimento para todos os dispositivos Android;
- *Instant Run* para aplicar alterações a aplicativos em execução sem precisar compilar um novo APK;
- Modelos de códigos e integração com GitHub para ajudar a criar recursos comuns dos aplicativos e importar exemplos de código;
- Ferramentas de verificação de código suspeito para detectar problemas de desempenho, usabilidade, compatibilidade com versões e outros.

A Figura 18 mostra a interface do Android Studio com o usuário.

Figura 18: Interface do Android Studio.



Fonte: Developers (2018d).

3.1.2 *Smartphone*

O *software* desenvolvido pode ser instalado em *smartphones* para realização das medições. O *smartphone* é responsável por se conectar a rede 4G e capturar as variáveis de interesse, e deve atender os seguintes requisitos:

- Sistema operacional Android com versões que suportem a API utilizada;
- Compatibilidade com a tecnologia 4G LTE;

3.1.3 Banco de dados SQLite

Os dados medidos foram salvos pelo aplicativo em um banco de dados. Um banco de dados é utilizado para que se possam salvar dados em formato de tabelas, e em cada coluna de uma tabela tipos diferentes de dados podem ser salvos (ANDROIDPRO, 2018). No contexto deste trabalho podem ser citadas colunas de RSRP, RSRQ, SINR, latitude e longitude.

No caso deste trabalho os dados foram somente inseridos e lidos das tabelas, e isso é feito através dos seguintes comandos SQL:

- INSERT, usado para se inserir dados;
- SELECT, usado para recuperar dados.

O SQLite, que será utilizado neste projeto, é o banco de dados oficial do Android, e funciona como qualquer outro banco de dados com sintaxe SQL. Ele é perfeitamente compatível com o Android por requerer memória limitada para ser executada de aproximadamente 250 kBytes (ANDROIDPRO, 2018).

O SQLite tem algumas limitações quanto aos dados que podem ser salvos em suas tabelas, sendo que apenas dados INTEGER, TEXT e REAL são aceitos, entre outras limitações, porém, é um banco de dados leve e bastante rápido, ideal para dispositivos Android (ANDROIDPRO, 2018).

3.1.4 API de mapas do Google

O Google Maps oferece uma API para o Android para se utilizar mapas. A API tem suporte automaticamente ao acesso à exibição dos mapas, resposta de gestos, adição de marcadores e etc., de forma bastante simples (DEVELOPERS, 2018e).

A exibição de marcadores será muito importante neste projeto, pois ela trará a possibilidade de se construir o mapa de cobertura, com a marcação dos pontos medidos de acordo com o seu nível de sinal.

3.1.5 API de gráficos

Neste trabalho a interação dos dados medidos com a confecção de gráficos é muito importante. Os gráficos permitem que o usuário possa interpretar os dados de uma forma mais rápida e prática, além disso, podem ajudar a extrair informações relevantes para a análise de resultados.

Para o desenvolvimento do *software* a API GraphView foi utilizada. A GraphView é uma biblioteca desenvolvida por terceiros e é utilizada para se plotar gráficos de forma rápida, prática e programática utilizando a plataforma Android (GRAPHVIEW, 2018).

3.2 Métodos

A seguir serão mostrados os métodos utilizados para realização do trabalho, que envolvem o desenvolvimento do *software* proposto e a medição dos dados e suas análises.

3.2.1 Desenvolvimento do *software* de medição

As ferramentas citadas na Seção 2.6 foram de suma importância para o desenvolvimento do *software*, e trouxeram a possibilidade do acesso às informações sobre

medições de qualidade do sistema. O principal objetivo do *software* é a captura de dados pertinentes à qualidade do sinal de redes 4G para análises com um bom grau de desempenho e usabilidade, e, além disso, o *software* deverá gravar as informações em um banco de dados, pois a manipulação delas se torna mais eficaz dessa forma.

3.2.1.1 Criação dos *Layouts*

A seguir serão apresentadas as telas desenvolvidas do *software*, e o funcionamento básico de cada uma. Em todas as telas o *Relative Layout* foi utilizado.

As telas no Android podem ser desenvolvidas de duas formas. A forma mais simples é a de arrastar componentes, como botões e imagens, para a área da tela no Android Studio. Quando um componente é arrastado, um código em linguagem XML (*Extensible Markup Language*) é criado automaticamente pelo Android Studio para defini-lo. Outra forma de se criar os *layouts*, é através do desenvolvimento de códigos pela linguagem XML diretamente, de forma programática. No trabalho aqui exposto, utilizou-se o método de gerar os *layouts* através da criação direta de códigos em linguagem XML.

3.2.1.1.1 Tela inicial

A tela inicial do aplicativo pode ser vista na Figura 19, e é mostrada quando o usuário abre o aplicativo. Nesta tela existem botões que dão acesso a outras telas do aplicativo, que serão citadas nas próximas seções deste trabalho.

Figura 19: Tela inicial do aplicativo.



Fonte: O autor.

3.2.1.1.2 Tela de medição dos dados

A tela mostrada na Figura 20 é uma das mais importantes do trabalho em desenvolvimento, pois é a tela em que as ações de captura dos dados primordiais para as análises de qualidade do sinal são executadas. Nela o usuário pode visualizar o valor dos dados medidos através dos campos de texto, e iniciar ou pausar a gravação das informações no banco de dados através do botão presente na parte inferior da tela. Os dados são atualizados em tempo real na tela do *smartphone* sempre que sofrem alguma alteração em seu valor.

Figura 20: Tela de gravação e medição dos dados.



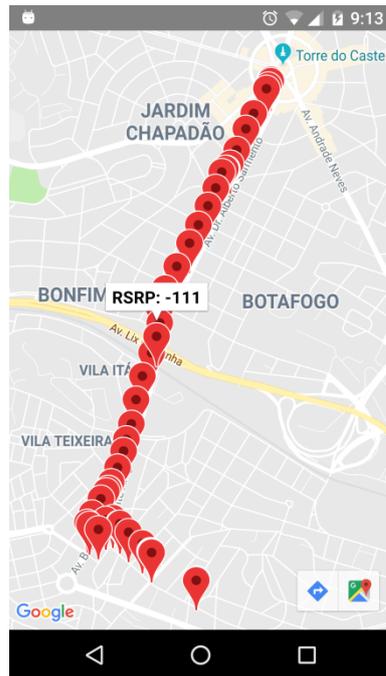
Fonte: O autor.

3.2.1.1.3 Tela do mapa de cobertura

A tela mostrada na Figura 21 mostra o mapa de cobertura que é gerado através dos dados medidos e salvos no banco de dados. Cada marcador presente no mapa representa um ponto medido e salvo no banco de dados. Os marcadores têm a sua cor definida em função dos valores de RSRP, RSRQ ou RSSNR. Existem três possibilidades de cores:

- Marcador verde: indica que a qualidade no local é excelente;
- Marcador amarelo: indica que a qualidade no local é aceitável;
- Marcador vermelho: indica que a qualidade no local é ruim;

Figura 21: Tela do mapa de cobertura.



Fonte: O autor.

3.2.1.1.4 Telas de gráficos

A Figura 22 mostra a tela onde podem ser escolhidas as opções de dados com os quais serão plotados os gráficos disponíveis. Existem três opções de gráficos, plotados com os valores de RSRP, RSRQ ou RSSNR.

Figura 22: Tela com gráfico.

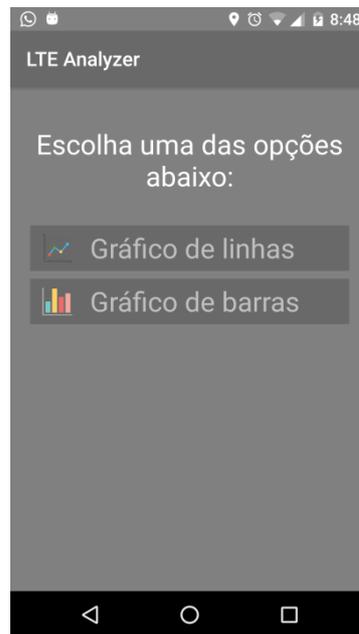


Fonte: o autor.

Ao se clicar em uma das opções da tela mostrada na Figura 22, a tela presente na Figura 23 é mostrada. A tela da Figura 23 traz ao usuário a possibilidade de escolher qual modelo de gráfico ele deseja visualizar. Duas opções estão disponíveis:

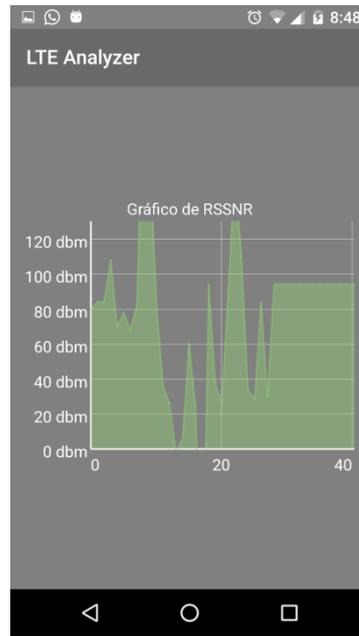
- Gráfico de linhas: esse gráfico é mostrado na Figura 24. É plotado com os valores de RSRP, RSRQ ou RSSNR no eixo Y, e com o valor de identificação de um ponto no banco de dados no eixo X, e dessa forma, quem está analisando o gráfico poderá notar alguma anomalia, e se desejar saber mais informações sobre os outros dados medidos no mesmo ponto, basta fazer uma consulta no banco de dados buscando pelo valor da identificação mostrada no eixo X e descobrir todas as informações sobre o mesmo.
- Gráfico de barras: esse gráfico é mostrado na Figura 25. É plotado utilizando faixas de valores que indicam a qualidade do serviço, e representa um gráfico de distribuição de frequência. Além disso, na tela onde esse gráfico é plotado há mais algumas informações adicionais que ajudam na análise de qualidade.

Figura 23: Tela para escolha da opção de modelo de gráfico.



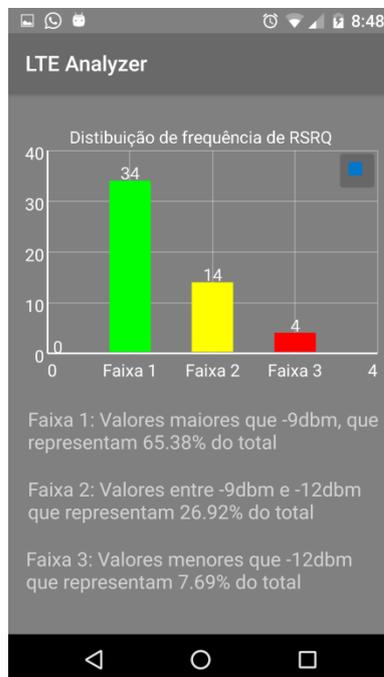
Fonte: O autor.

Figura 24: Gráfico de linhas.



Fonte: O autor.

Figura 25: Gráfico de barras.

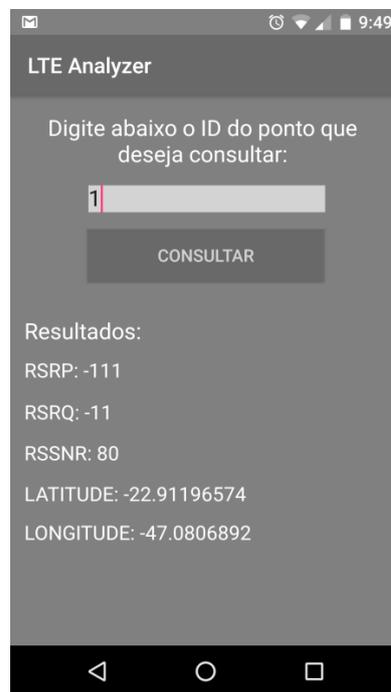


Fonte: O autor.

3.2.1.1.5 Tela de pesquisa a pontos de interesse

Na Figura 26 é mostrada a tela onde o usuário pode fazer a pesquisa detalhada de um ponto medido de seu interesse. Como pode ser visto na figura, existe um campo de texto onde um valor numérico de identificação pode ser inserido, e quando o botão “CONSULTAR” é clicado, uma busca por esse valor de identificação é feito no banco de dados, e todos os dados relacionados a ele são mostrados na tela. Essa tela pode ser usada em conjunto com a tela de gráfico de linhas mostrada na Seção 3.2.1.1.4 para análise e estudo de pontos específicos.

Figura 26: Tela de pesquisa detalhada.



Fonte: O autor.

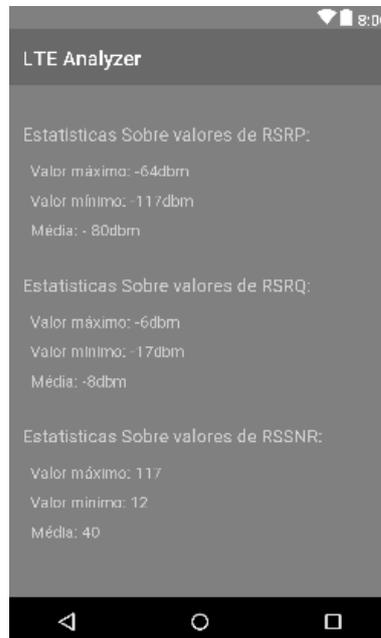
3.2.1.1.6 Tela de estatísticas

Na Figura 27 é mostrada a tela onde algumas informações sobre os valores de RSRP, RSRQ e RSSNR gravados no banco de dados são mostradas na tela. Os seguintes valores são mostrados sobre os parâmetros:

- Máximo: o valor máximo, ou seja, o melhor valor medido entre todos os valores de cada coluna da tabela;
- Mínimo: o valor mínimo, ou seja, o pior valor medido entre todos os valores de cada coluna da tabela;

- Média: o valor médio entre todos os valores de cada coluna da tabela.

Figura 27: Valores estatísticos sobre os dados gravados.



Fonte: O autor.

3.2.1.2 Acesso aos dados de interesse

A seguir será discutido o desenvolvimento das classes que são responsáveis por capturar os dados de telefonia móvel e localização que são salvos no banco de dados. Serão mostradas as ferramentas e APIs utilizadas.

3.2.1.2.1 Dados relacionados à qualidade das redes LTE

O acesso aos dados que podem ser utilizados para avaliar a qualidade das redes móveis de padrão LTE foi feito através das teorias mostradas na Seção 2.6.1 deste trabalho, utilizando principalmente a classe da API “*Android Telephony*” chamada `PhoneStateListener`. A `PhoneStateListener`, de acordo com o próprio site oficial do Android Studio, é uma classe que pode monitorar as alterações em dados de telefonia específicos do dispositivo, incluindo o estado do serviço, a força do sinal, entre outros (DEVELOPER, 2018).

Para utilizar os métodos da classe `PhoneStateListener`, foi criada no projeto uma classe chamada `SignalStrengthListener`, que é filha de `PhoneStateListener`, e, sendo assim pode utilizar todos os seus métodos.

Os métodos da classe `PhoneStateListener` fazem o acesso a informações importantes do telefone, e por isso algumas permissões devem ser requeridas para que o aplicativo possa

utilizá-la. As seguintes permissões devem ser definidas no código do arquivo AndroidManifest:

- READ_PHONE_STATE: Permissão requerida para se obter o acesso ao estado do telefone;
- ACCESS_NETWORK_STATE: Permissão requerida para se obter o acesso ao estado da rede;
- CHANGE_NETWORK_STATE: Permissão requerida para se capturar os eventos de mudanças no sinal da rede.

A classe criada chamada SignalStrengthListener tem somente um método, que é sobrescrito da sua classe mãe. Este método se chama onSignalStrengthChanged(), e tem sua implementação mostrada na Figura 28.

Figura 28: Implementação do método onSignalStrengthChanged.

```

267 public void onSignalStrengthsChanged(android.telephony.SignalStrength signalStrength)
268 {
269
270     ((TelephonyManager) getSystemService(TELEPHONY_SERVICE)).listen(signalStrengthListener,
271         SignalStrengthListener.LISTEN_SIGNAL_STRENGTHS);
272
273     tm = (TelephonyManager) getSystemService(Context.TELEPHONY_SERVICE);
274
275     String stringLte = signalStrength.toString();
276     String[] partsOfLteSignal = stringLte.split(regex: " ");
277     String lteRsrp = partsOfLteSignal[9];
278     String lteRsrq = partsOfLteSignal[10];
279     String lteRssnr = partsOfLteSignal[11];
280
281     if (tm.getNetworkType() == TelephonyManager.NETWORK_TYPE_LTE) {
282         try{
283             if (ActivityCompat.checkSelfPermission( getAppContext(), Manifest.permission.ACCESS_COARSE_LOCATION)
284                 != PackageManager.PERMISSION_GRANTED) {
285                 return;
286             }
287
288             cellInfoList = tm.getAllCellInfo();
289             for (CellInfo cellInfo : cellInfoList)
290             {
291                 if (cellInfo instanceof CellInfoLte)
292                 {
293                     cellSignalStrengthLte = ((CellInfoLte) cellInfo).getCellSignalStrength().getDbm();
294                 }
295             }
296         }

```

Fonte: O autor.

O método onSignalStrengthChanged() é executado sempre que há alguma mudança nos parâmetros do sinal de telefonia do dispositivo. Quando ele é chamado para ser executado, o próprio Android passa a ele um objeto signalStrength que contém as

informações sobre a rede de telefonia. Durante a execução do método esse objeto é manipulado para que se obtenham ao final os valores das variáveis em que se tem interesse.

As principais manipulações do objeto que o método recebe estão presentes nas linhas 275 a 279 do código mostrado na Figura 28. Os seguintes passos são executados nessas linhas:

- O objeto **signalStrength** recebido pelo método é transformado em uma String chamada **stringLte** na linha 275 do código.
- Na linha 275 há a declaração de um vetor de Strings chamado **partsOfLteSignal**. Através do método da classe String chamado `split(" ")` que é executado na linha 275, a String **stringLte** é separada em vários valores, e cada valor destes é salvo dentro de uma posição do vetor **partsOfLteSignal**.
- O vetor **partsOfLteSignal** contém todas as informações sobre a força do sinal, e as informações de interesse estão nas posições 8, 9 e 10, que indicam respectivamente os valores de RSRQ, RSRP e RSSNR.
- Desta forma, nas linhas 277, 278 e 279 respectivamente, as variáveis chamadas **lteRsrq**, **lteRsrp** e **lteRssnr** recebem o valor de RSRQ, RSRP e RSSNR;

O trecho de código mostrado na Figura 29 é executado dentro do método `onSignalStrengthChanged()`. Este trecho de código é muito importante, pois é ele que faz a atualização dos valores na tela mostrada na Seção 3.2.1.1.2 deste trabalho, que é a tela de medição de dados. Como os valores são atualizados sempre que há uma mudança no sinal, o técnico ou qualquer usuário que esteja operando o aplicativo e tenha um conhecimento prévio, consegue ver qual o atual cenário da qualidade de sua rede. Além disso, nas linhas 305, 306 e 311, respectivamente, as variáveis **rsrp**, **rsrq** e **rssnr** têm seus valores atualizados, o que é importante, pois são essas as variáveis que são salvas no banco de dados.

Figura 29: Trecho de código do método `onSignalStrengthChanged()`.

```

304         rsrpTextView.setText(String.valueOf("RSRP VALUE: " + lteRsrp));
305         rsrp = Integer.valueOf(lteRsrp);
306
307         rsrqTextView.setText(String.valueOf("RSRQ VALUE: " + lteRsrq));
308         rsrq = Integer.valueOf(lteRsrq);
309
310         rssnrTextView.setText(String.valueOf("RSSNR VALUE: " + lteRssnr));
311         rssnr = Integer.valueOf(lteRssnr);

```

Fonte: O autor.

3.2.1.2.2 Dados sobre localização

A captura dos dados de localização também é uma parte importante deste trabalho, pois através deles se torna possível fazer o mapa de cobertura das localidades onde os testes são aplicados.

Para que o aplicativo tenha acesso às informações de localização do dispositivo, as seguintes permissões devem ser definidas no código do arquivo AndroidManifest.xml:

- ACCESS_COARSE_LOCATION;
- ACCESS_FINE_LOCATION.

No código da classe em que se devem ser capturadas as informações de localização deve-se pedir permissão ao usuário para acessar os dados, e, além disso, deve-se garantir que o serviço de GPS (*Global Position System*) do dispositivo esteja ativado. Depois de feitas as configurações de todas as permissões, um método deve ser implementado para se configurar o serviço de GPS e capturar as mudanças de localização do dispositivo. O método utilizado para configurar o serviço de GPS foi denominado de configurarServiço(), e tem seu principal trecho de código mostrado na Figura 30.

Figura 30: Trecho de código do método configurarServiço().

```

190 public void configurarServiço()
191 {
192     try {
193         locationManager = (LocationManager) getSystemService(Context.LOCATION_SERVICE);
194         LocationListener locationListener = new LocationListener()
195         {
196             public void onLocationChanged(Location location) { atualizar(location); }
197             public void onStatusChanged(String provider, int status, Bundle extras) { }
198             public void onProviderEnabled(String provider) { }
199             public void onProviderDisabled(String provider) { }
200         };
201         locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER,
202             minTime: 0, minDistance: 0, locationListener);
203     }
204     catch (SecurityException ex)
205     {
206         Toast.makeText(context: this, ex.getMessage(), Toast.LENGTH_LONG).show();
207     }
208 }
209
210
211
212

```

Fonte: O autor.

Para se capturar os eventos de mudança da localização do dispositivo, é utilizada a classe LocationListener, que é nativa da API que trata de dados de GPS do Android. A classe LocationListener é utilizada para receber notificações da classe LocationManager sempre que

há uma alteração na localização do dispositivo (DEVELOPER, 2018). Na Figura 30, nota-se que um objeto da classe `LocationManager` é instanciado, assim como um objeto da classe `LocationListener`. Quando um objeto da classe `LocationListener` é instanciado, obrigatoriamente o método `onLocationChanged()` deve ser implementado. No método `onLocationChanged()` define-se o que será feito quando há uma mudança na localização. Como mostrado na linha 196 do código mostrado na Figura 30, o método `atualizar()` é executado quando há uma mudança na localização.

O método `atualizar()` têm sua implementação mostrada na Figura 31, onde nota-se que o método recebe um objeto de localização chamado **location** que contém as informações sobre a localização do dispositivo. Dois métodos do objeto são utilizados nas linhas 215 e 216 respectivamente, os métodos `getLatitude()` e `getLongitude()`, que são nativos da classe `Location` do Android, e retornam o valor de latitude e longitude para as variáveis **latPoint** e **lngPoint**. Após isso, os valores de latitude e longitude são utilizados nas linhas 221 e 222 para configurar os elementos de texto da tela de medição com os valores atuais de localização.

Figura 31: Implementação do método `atualizar()`.

```

213 public void atualizar(Location location)
214 {
215     Double latPoint = location.getLatitude();
216     Double lngPoint = location.getLongitude();
217
218     lattitude = latPoint.toString();
219     longitude = lngPoint.toString();
220
221     latitudeTextView.setText("Latitude: " + latPoint.toString());
222     longitudeTextView.setText("Longitude: " + lngPoint.toString());
223 }
224

```

Fonte: O autor.

3.2.1.3 Banco de dados

O banco de dados `SQLite`, que é um banco de dados nativo do Android, foi utilizado para gravar os dados medidos. Duas classes principais e nativas do Android foram utilizadas para que o banco de dados possa funcionar de forma estável no aplicativo, e são elas:

- `SQLiteDatabase`: classe que contém todos os métodos para utilização do banco de dados;

- SQLiteOpenHelper: classe que gerencia a criação do banco de dados.

Para se utilizar a classe SQLiteOpenHelper, foi criada uma classe chamada CreateDatabase, que herda os métodos da classe SQLiteOpenHelper. Na Figura 32, pode-se notar que a classe contém como variáveis todos os dados que devem ser salvos. Na linha 20 dos códigos da Figura 32, o construtor da classe é definido. O construtor é utilizado para criar o banco de dados sempre que a classe CreateDatabase for instanciada. Nas linhas 25 a 36, no método onCreate() é definido o comando SQL que cria a tabela onde os dados serão salvos.

Figura 32: Classe CreateDatabase.

```

7 public class CreateDatabase extends SQLiteOpenHelper
8 {
9     static final String NOME_BANCO = "pontosMedidos.db";
10    static final String TABELA = "pontosMedidos";
11    static final String ID = "id";
12    static final String RSRP = "rsrp";
13    static final String RSRQ = "rsrq";
14    static final String RSSNR = "rssnr";
15    static final String LATITUDE = "latitude";
16    static final String LONGITUDE = "longitude";
17    static final int VERSAO = 1;
18
19    public CreateDatabase(Context context)
20    {
21        super(context, NOME_BANCO, factory: null, VERSAO);
22    }
23
24    @Override
25    public void onCreate(SQLiteDatabase db)
26    {
27        String sql = "CREATE TABLE "+TABELA+" ("
28            + ID + " integer primary key autoincrement,"
29            + RSRP + " integer,"
30            + RSRQ + " integer,"
31            + RSSNR + " integer,"
32            + LATITUDE + " text,"
33            + LONGITUDE + " text"
34            + ")";
35        db.execSQL(sql);
36    }
37

```

Fonte: O autor.

A classe CreateDatabase tem dois métodos, e são eles:

- Método onCreate(): Método chamado quando a aplicação cria o banco de dados pela primeira vez, responsável por criar a tabela onde os dados serão salvos;
- Método onUpgrade(): Método responsável pela atualização do banco de dados quando alguma informação estrutural deve ser alterada.

Outra classe criada para se lidar com banco de dados foi a chamada DatabaseController, que é responsável por inserir e recuperar os dados do banco de dados. Assim, sempre que for necessário inserir ou ler dados das tabelas do banco de dados, basta utilizar os seus métodos com os parâmetros necessários. A classe DatabaseController tem basicamente dos métodos, e são eles:

- insertData(): Este método é mostrado na Figura 33, e como se pode notar, ele recebe todos os parâmetros que devem ser gravados no banco, cria um comando SQL e o executa para inserir os dados no banco;
- readData (): Este método é mostrado na Figura 34, e de acordo com sua implementação ele executa um comando SQL que recupera todos os dados do banco e os retorna em um objeto do tipo Cursor. O objeto do tipo Cursor é utilizado posteriormente para fazer uma varredura nos dados retornados e se utilizar cada um deles.

Figura 33: Método insertData().

```

18 public String insertData(int rsrp, int rsrq, int rssnr,
19                        String latitude, String longitude)
20 {
21     ContentValues valores;
22
23     long resultado;
24
25     db = banco.getWritableDatabase();
26
27     valores = new ContentValues();
28
29     valores.put(CreateDatabase.RSRP, rsrp);
30     valores.put(CreateDatabase.RSRQ, rsrq);
31     valores.put(CreateDatabase.RSSNR, rssnr);
32     valores.put(CreateDatabase.LATITUDE, latitude);
33     valores.put(CreateDatabase.LONGITUDE, longitude);
34
35     resultado = db.insert(CreateDatabase.TABELA, nullColumnHack: null, valores);
36     db.close();
37
38     if (resultado == -1)
39     {
40         return "Erro ao inserir registro";
41     }
42     else
43     {
44         return "Registro Inserido com sucesso";
45     }
46 }

```

Fonte: O autor.

Figura 34: Método readData().

```

48 public Cursor readData()
49 {
50     Cursor cursor;
51     String[] campos = {banco.ID, banco.RSRP, banco.RSRQ, banco.RSSNR,
52                       banco.LATITUDE, banco.LONGITUDE};
53
54     db = banco.getReadableDatabase();
55
56     cursor = db.query(banco.TABELA, campos, selection: null, selectionArgs: null,
57                     groupBy: null, having: null, orderBy: null, limit: null);
58
59     return cursor;
60 }

```

Fonte: O autor.

3.2.1.4 Gravação dos dados no banco de dados

A gravação dos dados é feita na *activity* da tela de medição dos dados mostrada na Seção 3.2.1.1.2 deste trabalho. Na Figura 20 pode-se notar que a tela de medição de dados contém um botão para iniciar a gravação de dados, e quando ele é clicado, um evento dispara um processo paralelo para gravar os dados no banco de dados, que é chamado de *Thread*.

Todos os elementos gráficos de aplicativos Android são executados em uma *Thread* principal, que não pode ser interferida por outros processos, pois o sistema operacional entenderá que se essa *Thread* sofrer uma pausa demasiada o aplicativo sofreu um travamento, e assim a aplicação será interrompida e fechada. Como na tela de gravação de dados há elementos gráficos que são constantemente atualizados, que são os valores atuais das variáveis medidas, entende-se que este processo não pode ser interrompido por outro. Como a gravação no banco de dados é um processo crítico que pode sofrer demoras ou travamentos, é recomendável que este processo seja feito em uma *Thread*, que se resume a um processo paralelo à execução do aplicativo.

Na Figura 35 pode-se ver a implementação do evento de clique no botão que inicializa a gravação de dados. Na linha 141 do código mostrado na Figura 35, nota-se que há a instanciação de um *Thread*, e na linha 142 há a inicialização dela.

Figura 35: Implementação do evento de clique do botão de gravação dos dados.

```

124      record.setOnClickListener((view) -> {
129
130          if (isToRecord == 0) {
131              isToRecord = 1;
132          }
133          else
134          {
135              isToRecord = 0;
136          }
137
138          if (isToRecord == 1)
139          {
140              record.setText("PAUSAR A GRAVAÇÃO");
141              recordThread = new Thread(new SaveData());
142              recordThread.start();
143          }
144          else
145          {
146              record.setText("INICIAR GRAVAÇÃO DE DADOS");
147          }
148      });

```

Fonte: O autor

Na Figura 36 verifica-se a implementação da *Thread* instanciada anteriormente. Toda *Thread* deve ter um método `run()`, que é disparado quando ela é iniciada. Nota-se que neste método existe um laço de repetição que é executado até que se aperte o botão de pausar a gravação de dados, quando o laço é interrompido e a execução da *Thread* é finalizada. Dentro deste laço de repetição, na linha 246 do código, a gravação dos dados é feita no banco através do método `insertData()`, que pertence a classe `DatabaseController` e foi explicado na Seção 3.2.1.3 deste trabalho. Ainda no código Figura 36, na linha 233 o método `sleep(milis: 8000)` é executado, e define que os dados serão gravados no banco de dados de 8 em 8 segundos.

Figura 36: Implementação da *Thread* responsável pela gravação dos dados no banco.

```

225 public class SaveData implements Runnable
226 {
227     @Override
228     public void run()
229     {
230         while (isToRecord == 1)
231         {
232             try{
233                 Thread.sleep( millis: 8000);
234             }
235             catch(InterruptedException e)
236             {
237                 e.printStackTrace();
238             }
239
240             if (latitude.equals("") && longitude.equals(""))
241             {
242                 Log.i( tag: "LATITUDE: ", msg: "LATITUDE NULA");
243             }
244             else
245             {
246                 crud.insertData(rsrp, rsrq, rssnr, latitude, longitude);
247             }
248         }
249     }
250 }
251

```

Fonte: O autor.

É importante saber como são definidas as tabelas do banco de dados para o total entendimento deste trabalho. A tabela criada para que os dados sejam salvos tem as seguintes colunas:

- ID: Este dado é definido como um dado de auto incremento, ou seja, cada vez que uma linha que representa um ponto é salva, ela recebe um ID que é único, e que recebe um incremento para que seu valor não se repita entre linhas da mesma tabela.
- RSRQ: Dado numérico que indica o valor de RSRQ no ponto que está sendo medido.
- RSRP: Dado numérico que indica o valor de RSRP no ponto que está sendo medido.
- RSSNR: Dado numérico que indica o valor de RSSNR no ponto que está sendo medido.
- LATITUDE: Dado de texto que define a latitude do ponto que está sendo medido.
- LONGITUDE: Dado de texto que define a longitude do ponto que está sendo medido.

3.2.1.5 API de mapas e consulta ao banco de dados

A API de mapas do Google *Maps* foi utilizada para fazer o mapa de cobertura através dos dados pertinentes à qualidade do sinal e de localização medidos. Na Seção 3.2.1.1.3 na Figura 21 é mostrado o *layout* da tela que contém o mapa. A classe criada para gerenciar os mapas se chama *MapaDeCobertura*.

Para utilizar a API do Google *Maps* os dois métodos explicados a seguir devem ser obrigatoriamente implementados.

- `onCreate()`: método que é executado quando a *activity* do mapa é inicializada. A implementação deste método pode ser visto na Figura 37.
- `onMapReady()`: método que é executado quando o mapa é carregado e está pronto para ser exibido. A implementação deste método pode ser visto na Figura 38.

Figura 37: Implementação do método `onCreate()` da classe *MapaDeCobertura*.

```
24      @Override
25      protected void onCreate(Bundle savedInstanceState) {
26          super.onCreate(savedInstanceState);
27          setContentView(R.layout.activity_mapa_de_cobertura);
28          // Obtain the SupportMapFragment and get notified when the map is ready to be used.
29          SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager()
30              .findFragmentById(R.id.map);
31          mapFragment.getMapAsync(new OnMapReadyCallback() {
32              @Override
33              public void onMapReady() {
34                  crud = new DatabaseController(getApplicationContext());
35              }
36          });
37      }
```

Fonte: O autor.

Figura 38: Implementação do método `onMapReady()` da classe `MapaDeCobertura`.

```

36 public void onMapReady(GoogleMap googleMap) {
37     mMap = googleMap;
38     cursor = crud.readData();
39     int columnLatIndex = cursor.getColumnIndex( S: "latitude");
40     int columnLonIndex = cursor.getColumnIndex( S: "longitude");
41     int columnRsrpIndex = cursor.getColumnIndex( S: "rsrp");
42
43     while (cursor.moveToNext()) {
44         LatLng latlng = new LatLng(Double.valueOf(cursor.getString(columnLatIndex)),
45             Double.valueOf(cursor.getString(columnLonIndex)));
46         int rsrp = Integer.valueOf(cursor.getString(columnRsrpIndex));
47
48         if (rsrp >= -75) {
49             mMap.addMarker(new MarkerOptions().position(latlng).title("RSRP: "
50                 + cursor.getString(columnRsrpIndex)).icon(BitmapDescriptorFactory.
51                 defaultMarker(BitmapDescriptorFactory.HUE_GREEN)));
52         } else if (rsrp < -75 && rsrp >= -95) {
53             mMap.addMarker(new MarkerOptions().position(latlng).title("RSRP: "
54                 + cursor.getString(columnRsrpIndex)).icon(BitmapDescriptorFactory.
55                 defaultMarker(BitmapDescriptorFactory.HUE_ORANGE)));
56         } else if (rsrp < -95) {
57             mMap.addMarker(new MarkerOptions().position(latlng).title("RSRP: "
58                 + cursor.getString(columnRsrpIndex)).icon(BitmapDescriptorFactory.
59                 defaultMarker(BitmapDescriptorFactory.HUE_RED)));
60         }
61     }
62     mMap.animateCamera(CameraUpdateFactory.newLatLngZoom(new LatLng( v: -22.9064,
63                                     v1: -47.0616),
64                                     v: 12.8f));
65     cursor.close();
66 }

```

Fonte: O autor.

No método `onCreate()` mostrado na Figura 37, há a instanciação de um objeto do banco de dados na linha 32. Este objeto é utilizado no método `onMapReady()` para que se possa recuperar os dados do banco de dados.

O método `onMapReady()` busca todos os dados gravados no banco de dados através do método `readData()` que pertence a classe `DatabaseController`. O método faz uma varredura por todos os dados do banco, e usa a latitude e longitude para posicionar marcadores no mapa. É importante ressaltar que a cor do marcador plotado no mapa varia em função do valor de RSRP mostrado no ponto medido. As cores foram definidas de acordo com as faixas de valores citadas na Seção 2.4.1 deste trabalho, e são elas:

- Maior que -75 dBm, qualidade excelente, marcador verde;
- Entre -75 dBm e -95 dBm, qualidade aceitável, marcador amarelo;
- Menor que -95 dBm, qualidade inaceitável, marcador vermelho.

3.2.1.6 Geração de gráficos com os valores medidos

Como não existe uma API do próprio Android que trate de forma simples e robusta da construção de gráficos, foi utilizada uma API de terceiros, chamado Graphview. Depois de adicionadas as dependências da biblioteca Graphview no projeto, foi criada uma classe para gerenciar gráficos chamada GraphsActivity. Essa classe é responsável por fazer uma consulta no banco de dados, criar uma série com os dados lidos e adicionar esses dados ao gráfico. A Figura 39 mostra o trecho de código que executa a criação de gráficos na classe GraphsActivity.

Figura 39: Trecho de código da classe GraphsActivity.

```

20  protected void onCreate(Bundle savedInstanceState)
21  {
22      super.onCreate(savedInstanceState);
23      setContentView(R.layout.activity_rsrpgraph);
24      GraphView graph = (GraphView) findViewById(R.id.graph);
25      crud = new DatabaseController(getApplicationContext());
26      cursor = crud.readData();
27
28      if(cursor.getCount() > 0)
29      {
30          DataPoint[] points = new DataPoint[cursor.getCount()];
31          int columnRsrpIndex = cursor.getColumnIndex("rsrp");
32
33          while ( cursor.moveToNext() )
34          {
35              int rsrp = Integer.valueOf(cursor.getString(columnRsrpIndex));
36              points[cursor.getPosition()] = new DataPoint(cursor.getPosition(), rsrp);
37          }
38
39          cursor.close();
40
41          LineGraphSeries<DataPoint> series = new LineGraphSeries<>(points);
42
43          //set series appearance parameters
44          series.setColor(Color.argb( alpha: 95, red: 147, green: 219, blue: 112));
45          series.setDrawBackground(true);
46          series.setBackground(Color.argb( alpha: 95, red: 147, green: 219, blue: 112));
47          graph.addSeries(series);
48      }
49

```

Fonte: O autor.

Na linha 24 do trecho de código mostrado na Figura 39 um objeto da biblioteca Graphview é instanciado. Na linha 30 é criado um vetor chamado **points**, onde cada ponto do gráfico será armazenado com seus valores de X e Y. Na linha 36 todos os dados recuperados do banco de dados são salvos um por um em cada posição do vetor points, sendo o eixo X definido pelo campo “id” lido no banco de dados, e o eixo Y definido pelos valores dos campos de RSRP, RSRQ ou RSSNR. Na linha 41 é criada uma série com gráficos de linha

através do vetor `points`, e na linha 48 essa série é adicionada ao objeto da biblioteca `Graphview`, e assim o gráfico é criado.

3.2.1.7 Busca de informações no banco de dados

A busca de informações extras de pontos no banco de dados é feita através da tela mostrada na Figura 26, presente na Seção 3.2.1.1.5 deste trabalho. Nessa tela o usuário pode inserir um valor de identificação, e ao clicar no botão “CONSULTAR” esse valor é recuperado pela classe `DetailsActivity`. O trecho de código mostrado na Figura 40 recupera o valor inserido e chama o método `readDataById()` da classe `DatabaseController` passando esse valor como parâmetro.

Figura 40: Trecho de código da classe `DetailsActivity`.

```
cursor = crud.readDataById(Integer.valueOf(idEnteredValue.getText().toString()));
```

Fonte: O autor.

A implementação do método `readDataById()` pode ser vista na Figura 41. No código mostrado na figura, é montada uma *query* de seleção, onde o parâmetro de seleção é a variável `id`, que é recebida pelo método como parâmetro, e desta forma somente será retornada na variável `cursor` a linha que tem esse valor de identificação recebido.

Figura 41: Implementação do método `readDataById`.

```
60 public Cursor readDataById(int id)
61 {
62     Cursor cursor;
63     String[] campos = {banco.ID, banco.RSRP, banco.RSRQ, banco.RSSNR,
64                       banco.LATITUDE, banco.LONGITUDE};
65
66     db = banco.getReadableDatabase();
67
68     // Filter results WHERE "title" = 'My Title'
69     String selection = banco.ID + "=?";
70     String[] selectionArgs = {String.valueOf(id)};
71
72     cursor = db.query(banco.TABELA, campos, selection, selectionArgs,
73                     groupBy: null, having: null, orderBy: null, limit: null);
74
75     return cursor;
76 }
```

Fonte: O autor.

3.2.1.8 Dados estatísticos

Os dados estatísticos são mostrados na Figura 27 presente na Seção 3.2.1.1.6 deste trabalho. Nesta tela o usuário pode ver as informações estatísticas através de campos de texto. Os valores de máximo, mínimo e da média são obtidos através da classe DatabaseController por comandos de consulta personalizados que são executados no banco de dados. Na Figura 42 é mostrado o código com os métodos `getMinRsrp()`, `getAvgRsrp()` e `getBiggestRsrp()`, todos da classe DatabaseController.

Figura 42: Implementação dos métodos de busca de dados estatísticos no banco.

```

78     public int getBiggestRsrp()
79     {
80         Cursor cursor;
81         db = banco.getReadableDatabase();
82         cursor = db.rawQuery( sql: "SELECT MAX(rsrp) FROM pontosMedidos", selectionArgs: null);
83         cursor.moveToFirst();
84         int maxValue = cursor.getInt( 0);
85         cursor.close();
86         return maxValue;
87     }
88     public int getMinRsrp()
89     {
90         Cursor cursor;
91         db = banco.getReadableDatabase();
92         cursor = db.rawQuery( sql: "SELECT MIN(rsrp) FROM pontosMedidos", selectionArgs: null);
93         cursor.moveToFirst();
94         int minValue = cursor.getInt( 0);
95         cursor.close();
96         return minValue;
97     }
98     public double getAvgRsrp()
99     {
100        String query= "SELECT AVG(rsrp) From pontosMedidos";
101        Cursor cursor = db.rawQuery(query, selectionArgs: null);
102        cursor.moveToFirst();
103        double AverageValue = cursor.getDouble( 0);
104        cursor.close();
105        return AverageValue;
106    }

```

Fonte: O autor.

A diferença entre os métodos mostrados na Figura 42 está nas *queries*, ou comandos que cada um executa para buscar as informações no banco de dados. Para buscar o valor máximo, o parâmetro de seleção utilizado deve ser **MAX(rsrp)**, para o valor mínimo usa-se **MIN(rsrp)** e para o valor da média usa-se **AVG(rsrp)**, onde **rsrp** é o nome da coluna da tabela que deve ser consultada. Métodos equivalentes foram implementados também para os valores de RSRQ e RSSNR, com mudança apenas no nome da coluna consultada.

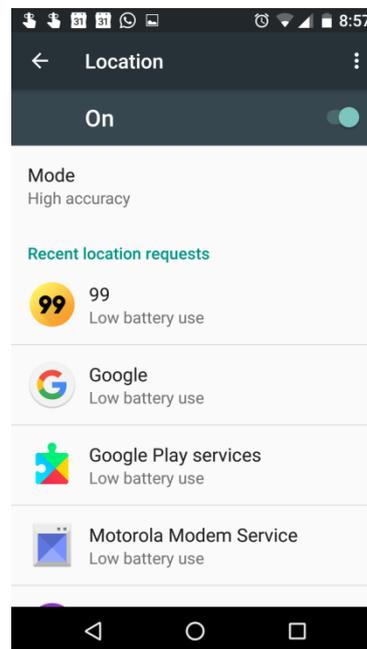
3.2.2 Medições

As medições foram feitas tomando como base testes realizados no modelo de *drive-test* e *walk-test*, que são formas comuns de se realizar avaliações em redes de telecomunicações. O equipamento utilizado para medições foi um *smartphone* ASUS ZENFONE ZE554KL, que tem as seguintes configurações:

- Sistema operacional: Android Oreo 8.0.
- Memória RAM: 4GB.
- Processador: *Octa-core* com *clock* de 2GHz.
- Memória interna: 64GB.

Devem ser feitas configurações específicas no *smartphone* para um uso correto do aplicativo. As configurações mostradas na Figura 43 e na Figura 44 foram feitas com essa finalidade, antes do início das medições.

Figura 43: Configurações de localização do telefone.

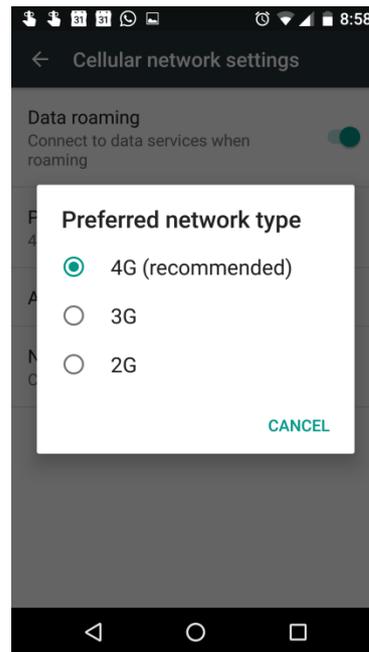


Fonte: O autor.

A Figura 43 mostra as configurações dos serviços de localização do *smartphone*, que devem estar ativas, e devem estar configuradas para fornecer a localização com alta precisão. Quando o aplicativo for instalado pela primeira vez no *smartphone*, as permissões para uso desses serviços serão requisitadas, e devem ser aceitas pelo usuário para o correto funcionamento do *software* de medição.

A Figura 44 mostra a configuração básica de serviços de rede móvel que deve ser assumida para o funcionamento correto do *software*, onde nota-se que o *smartphone* está configurado para conectar-se preferencialmente em redes de 4G. Recomenda-se o uso dessa configuração para que se atinja um melhor aproveitamento na quantidade de pontos medidos quando se está no percurso de medição, pois quando o telefone não está conectado em redes 4G de padrão LTE os pontos medidos são descartados.

Figura 44: Configuração de rede recomendada.



Fonte: O autor.

A Figura 44 mostra a configuração básica de serviços de rede móvel que deve ser assumida para o funcionamento correto do *software*, onde nota-se que o *smartphone* está configurado para conectar-se preferencialmente em redes de 4G. Recomenda-se o uso dessa configuração para que se atinja um melhor aproveitamento na quantidade de pontos medidos quando se está no percurso de medição, pois quando o telefone não está conectado em redes 4G de padrão LTE os pontos medidos são descartados.

Ao fazer um paralelo com testes profissionais de *drive-test* e *walk-test*, há uma deficiência na riqueza de dados, pois nem todos os parâmetros utilizados podem ser obtidos utilizando somente o *smartphone*. Nos testes realizados utilizando o *software* desenvolvido neste trabalho, o *smartphone* se comporta como um *scanner*, capturando dados importantes na qualidade de redes móveis. As informações que são capturadas pelo aparelho serão:

- Latitude e Longitude, que servem para identificar a localização de cada medição;

- RSRP (*Reference Signal Received Power*);
- RSRQ (*Reference Signal Received Quality*);
- RSSNR (*Reference Signal to Interference and Noise Ratio*).

As medições foram feitas com o *smartphone* em vários estados de funcionamento, dentre eles:

- Modo *idle* (ocioso);
- Aparelho conectado à rede realizando *download* de arquivos;
- Aparelho conectado à rede realizando *upload* de arquivos;
- Durante a realização de chamadas de voz.

Os testes foram realizados na cidade de Campinas, onde se buscou explorar todas as adversidades do meio, como áreas de alta e baixa densidade demográfica, áreas com vários prédios, áreas com muita vegetação, e também foram exploradas condições meteorológicas adversas, para que todas as possibilidades sejam estudadas e a rede seja bem avaliada. Os testes foram feitos em redes comerciais reais da operadora TIM.

Os percursos foram avaliados e definidos de acordo com a necessidade dos testes, buscando contemplar todas as situações possíveis citadas no parágrafo anterior. Os testes foram feitos no bairro Vila Industrial e nas suas mediações, pois nessa região existem partes com vários e poucos prédios, com relevo acidentado e plano e ainda partes com bastante vegetação, e por isso cobre toda a necessidade de variedade que os testes requerem.

Dados estatísticos, gráficos e mapas são gerados automaticamente pelo aplicativo e serão analisados nas próximas seções para se avaliar os resultados das medições utilizando o *software* desenvolvido e gerar discussões sobre a viabilidade do seu uso para se analisar redes de telefonia móveis 4G de padrão LTE.

3.3 Considerações finais

Poucos recursos de *hardware* foram utilizados no desenvolvimento deste trabalho, visto que somente um *smartphone* foi utilizado para a captura dos dados, além de um computador para desenvolvimento do *software*, o que atende um dos objetivos do trabalho. Algumas dificuldades foram encontradas durante o desenvolvimento pela falta de informações e documentação sobre algumas APIs que lidam com sinais de redes móveis no Android.

A escolha de se gravar os dados em um banco de dados se mostrou bastante eficaz, pois com seu uso o armazenamento de pontos se tornou leve e rápido. Esse era um ponto crítico do *software*, pois o número de pontos medidos é muito elevado, e outra forma de armazenar os dados certamente seria problemática. Com o uso de todas as ferramentas e através do suporte do Android para o uso do banco de dados, o aplicativo teve seu desempenho otimizado em relação a outros métodos de armazenamento.

4 RESULTADO E DISCUSSÕES

A seguir os resultados obtidos através da medição dos dados serão apresentados e posteriormente discutidos. Os dados capturados pelo *software* foram organizados na tabela de seu banco de dados interno, que é uma tabela como a Tabela 2, que contém dados reais obtidos nas medições.

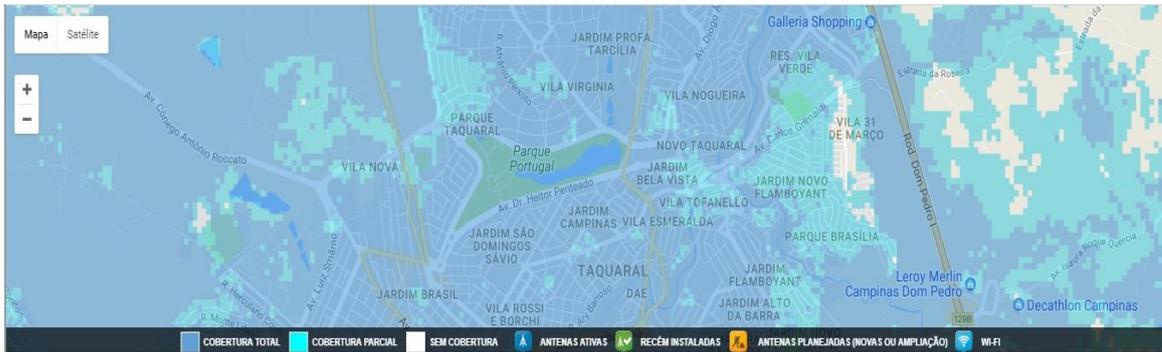
Tabela 2: Tabela com alguns valores medidos pelo *software* desenvolvido.

Ponto	RSRQ (dBm)	RSRP (dBm)	RSSNR (dBm)	Latitude	Longitude
95	-12	-85	-20	-22,90690374	-47,081774010
96	-12	-85	-20	-22,90690374	-47,081774010
97	-11	-87	66	-22,90747931	-47,080222160
98	-10	-79	140	-22,90802252	-47,080589840
99	-11	-72	166	-22,90850649	-47,080907710
100	-10	-75	164	-22,90896144	-47,081014180
101	-10	-76	144	-22,90928866	-47,081317480
102	-12	-76	48	-22,90968044	-47,081706430
103	-9	-72	242	-22,91009184	-47,082106300
104	-11	-71	132	-22,91048271	-47,082510950

Fonte: O Autor.

Os dados armazenados foram utilizados para se plotar gráficos que serão analisados nas seções seguintes. Foram gerados gráficos de distribuição de frequência e de linhas utilizando os valores de RSRP, RSRQ e SINR obtidos em cada região. Os gráficos de distribuição de frequência deverão auxiliar na detecção da quantidade de pontos que estão em faixas satisfatórias ou não. Já os gráficos de linha são responsáveis por auxiliar a detecção de um ponto que possa apresentar valores anormais nas variáveis medidas. Através dos dados colhidos foi gerado também um mapa de cobertura simplificado, que deverá ser comparado aos mapas de cobertura que as operadoras fornecem para os usuários da rede, como o mostrado na Figura 45.

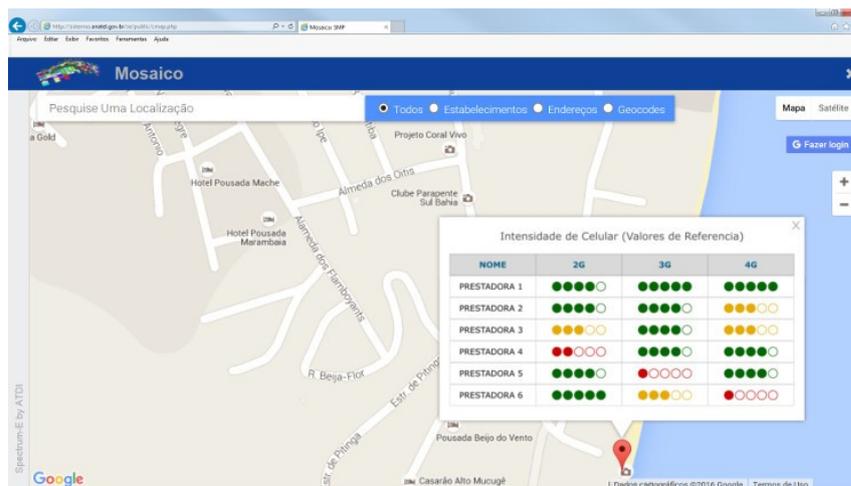
Figura 45: Mapa de cobertura da operadora TIM na cidade de Campinas, SP.



Fonte: Tim (2018a).

A Anatel, através de uma ferramenta do Sistema Mosaico disponibiliza uma ferramenta para consulta da intensidade do sinal em qualquer ponto do Brasil. A ferramenta contempla as tecnologias 2G, 3G e 4G de qualquer operadora que preste serviço no Brasil. A intensidade do sinal mostrada pela Anatel é calculada com base nas informações técnicas que as operadoras cadastram no banco de dados da Anatel (ANATEL, 2018e). A Figura 46 mostra a interface da ferramenta e a intensidade do sinal em um determinado ponto. O *software* desenvolvido é capaz de colher os dados de latitude e longitude da localização em que a medição é feita, o que traz a possibilidade da comparação de intensidade entre os valores medidos e os mostrados pela ferramenta da Anatel.

Figura 46: Intensidade do sinal fornecida pelo sistema Mosaico da Anatel.



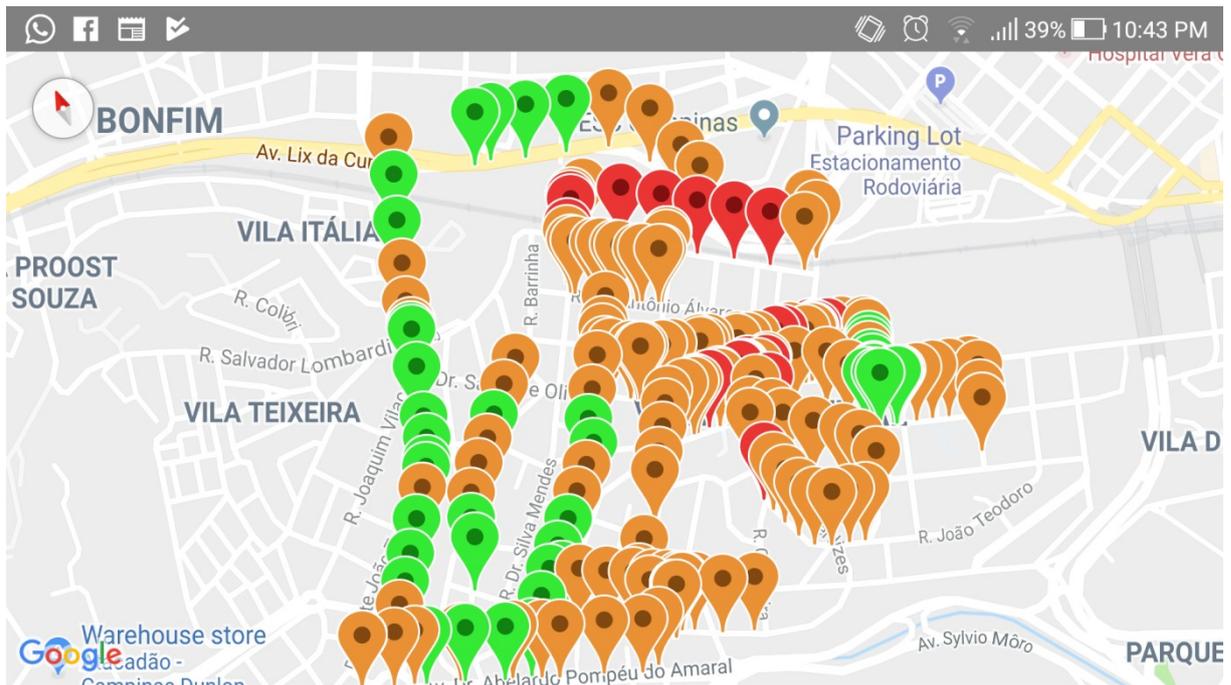
Fonte: Anatel (2018f).

Além disso, seguindo o que foi citado na Seção 2.4.2 deste trabalho, pode-se realizar a comparação entre os valores de RSRQ e RSRP e verificar se haverá ruídos ou interferências na rede.

4.1 Resultado das medições no bairro Vila Industrial

Como citado na Seção 3.2.1, foi escolhido o bairro Vila Industrial e parte dos seus bairros vizinhos para se realizar as medições devido à diversidade geográfica dessa área. A Figura 47 mostra um *print* do mapa de cobertura gerado pelo *software* através dos dados medidos. Ao todo 346 pontos foram registrados durante as medições, onde cada medição é simbolizada por um marcador no mapa.

Figura 47: Mapa de cobertura do bairro Vila Industrial e suas mediações.



Fonte: O autor.

Além do mapa de cobertura, também foram gerados gráficos de RSRP, RSRQ e RSSNR e algumas informações estatísticas sobre os dados medidos. A Figura 48 contém um *print* da tela onde algumas informações estatísticas são mostradas.

Figura 48: Informações estatísticas sobre os dados medidos



Fonte: O autor.

A Figura 49 mostra os *prints* das telas dos gráficos de linha plotados com os valores de RSRP, RSRQ e RSSNR gerados através dos dados medidos.

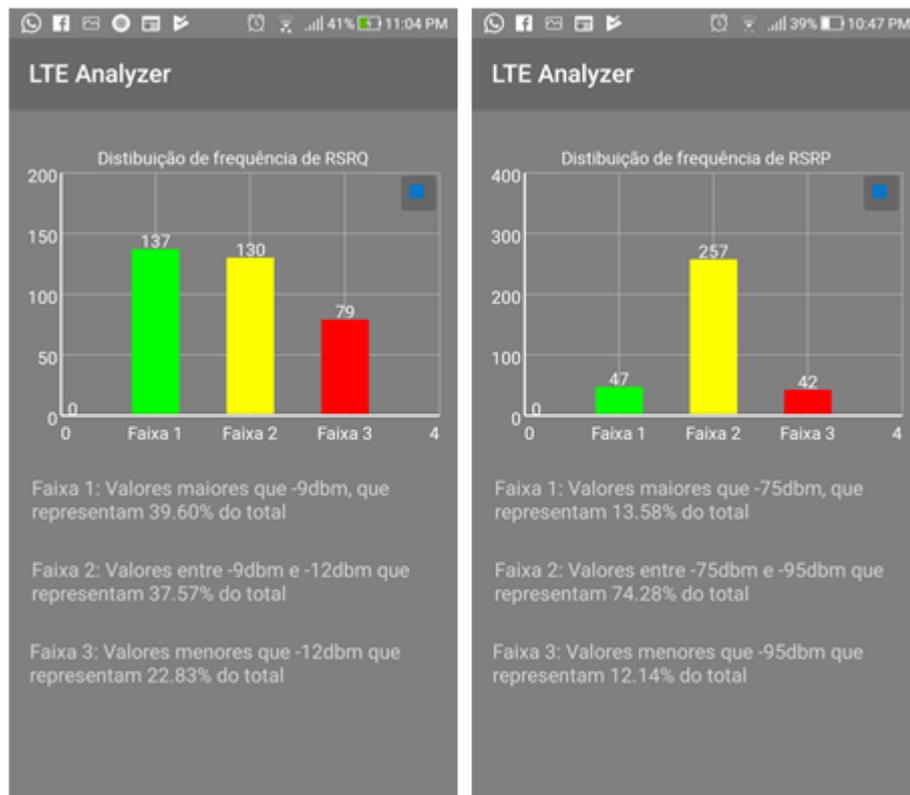
Figura 49: Gráficos de linha plotados pelo *software* desenvolvido.



Fonte: O autor.

A Figura 50 mostra um *print* da tela dos gráficos de barras plotados com os valores medidos de RSRP e RSRQ. Esse modelo de gráfico não foi gerado para a variável RSSNR, pois nenhuma referência que indicasse faixas de valores bons, aceitáveis ou ruins foi encontrada, porém, como esse parâmetro indica uma relação entre o potencial do sinal e dos ruídos e interferências pode-se considerar que quando maior e mais positivo melhor será o sinal recebido.

Figura 50: Distribuição de frequência de RSRP e RSRQ.



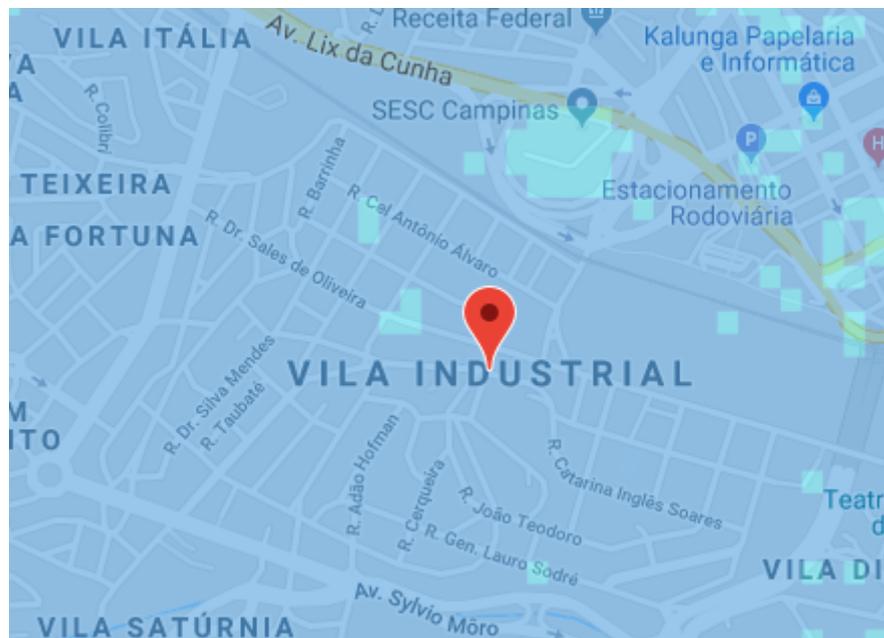
Fonte: O autor.

As barras indicam a quantidade de pontos que pertencem a uma determinada faixa de valores que está descrita logo abaixo do gráfico e foi definida de acordo com a descrição das Seções 2.4.1 e 2.4.2. A Faixa 1 tem a cor verde e indica valores bons, a Faixa 2 tem a cor amarela e indica valores aceitáveis e a Faixa 3 tem a cor vermelha e indica valores ruins. Em baixo do gráfico está a descrição de cada faixa e a porcentagem de valores presentes em relação ao total de valores medidos.

4.1.1 Discussões sobre o resultado das medições no bairro Vila Industrial

A Figura 51 mostra o mapa de cobertura fornecido pela operadora TIM em seu *website*. Existem diferenças entre o mapa gerado pelo *software* desenvolvido e o mapa fornecido pela operadora, porém, algumas conclusões podem ser tomadas ao se comparar os dois. A principal diferença entre eles é que o mapa da TIM indica somente se existe ou não cobertura para a tecnologia 4G em uma localização, não expondo o nível do sinal. O mapa de cobertura gerado pelo *software* desenvolvido tem seus marcadores plotados em função da variável RSRP, que indica o valor da potência do sinal de referência recebido pelo *smartphone*. O mapa de cobertura da TIM têm duas cores, azul escuro e azul claro, sendo o azul escuro um indicador de cobertura total, e o azul claro um indicador de cobertura parcial (TIM, 2018). Esses tipos de mapas fornecidos pelas operadoras são em sua maioria feitos em linhas muito gerais, somente indicando a presença ou ausência de cobertura, não fornecendo o nível de qualidade ou potência dos sinais, e, além disso, como são feitos através de predições podem apresentar falhas.

Figura 51: Mapa de cobertura da operadora TIM nas mediações do bairro Vila Industrial.



Fonte: Tim (2018a).

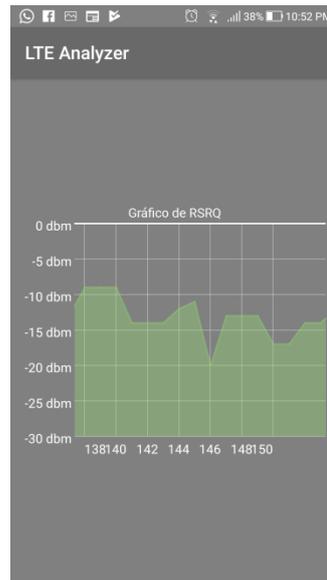
Ao se comparar os dois mapas de cobertura, da Figura 47 e da Figura 51, pode-se concluir que essa funcionalidade está correta no mapa gerado pelo *software* desenvolvido, pois todos os valores medidos de RSRP indicaram que existe cobertura 4G total na região

onde os testes foram feitos, assim como é indicado no mapa fornecido pela operadora TIM. Apesar de existirem pontos marcados em vermelho no mapa de cobertura da Figura 47, atesta-se que existe cobertura 4G na região, pois segundo os valores mostrados na Figura 48 o ponto com valor mínimo de RSRP medido obteve o resultado de -103 dBm, que está dentro do limiar de aceitação indicado pela Anatel citado na seção 2.4.1.1 deste trabalho, que é de -110 dBm. O valor médio de RSRP mostrado na Figura 48 também pode ser usado para atestar a afirmação de que o *software* se comportou como o esperado, pois a média de todos os valores está em uma faixa que indica uma cobertura aceitável.

Analisando os gráficos de distribuição de frequência dos pontos medidos que estão presentes na Figura 50, é possível constatar que a rede da operadora TIM comporta-se de forma aceitável na região onde os testes foram feitos, pois apenas 22,83% dos valores de RSRQ medidos estão em uma faixa considerada ruim e 77,17% estão na faixa de valores ótimos ou aceitáveis, e para os valores de RSRP medidos apenas 12,14% dos pontos estavam em uma faixa de valores considerados ruins e 87,84% estavam na faixa de valores ótimos ou aceitáveis. Isso pode ser usado de auxílio mais uma vez para constatar o correto funcionamento do *software* desenvolvido neste trabalho, pois através desses resultados pode-se afirmar que a rede avaliada apresenta um funcionamento estável, assim como o mapa de cobertura fornecido pela operadora sugere.

Ao se analisar os valores mostrados na Figura 48 que indicam os valores de máximo, mínimo e a média, há indícios de que a rede se comporta bem, pois os valores médios estão em faixas aceitáveis, o que é comprovado pelos gráficos de distribuição de frequência. Somente há uma ressalva a ser feita para o valor mínimo de RSRQ, que está fora do limite de valores considerados normais segundo a teoria apresentada na Seção 2.4.2 deste trabalho. Utilizando os gráficos de linha é possível observar o valor mínimo de cada variável medida e a sua identificação nesse ponto, e através da tela de consultas ao banco podem ser encontradas mais informações sobre as informações dos valores medidos no local. De acordo com a Figura 48 o valor mínimo de RSRQ é de -20 dBm. Ao se analisar o gráfico de linha de RSRQ, o ponto mínimo com identificação 146 mostrado na Figura 52 foi encontrado.

Figura 52: Ponto mínimo de RSRQ.



Fonte: O autor.

Após identificar o ponto mínimo mostrado na Figura 52 e constatar que sua identificação está em torno do valor de 146 das informações no banco de dados, foi feita a busca pelo ponto na tela de consultas, e através dessa funcionalidade os valores mostrados na Figura 53 foram encontrados.

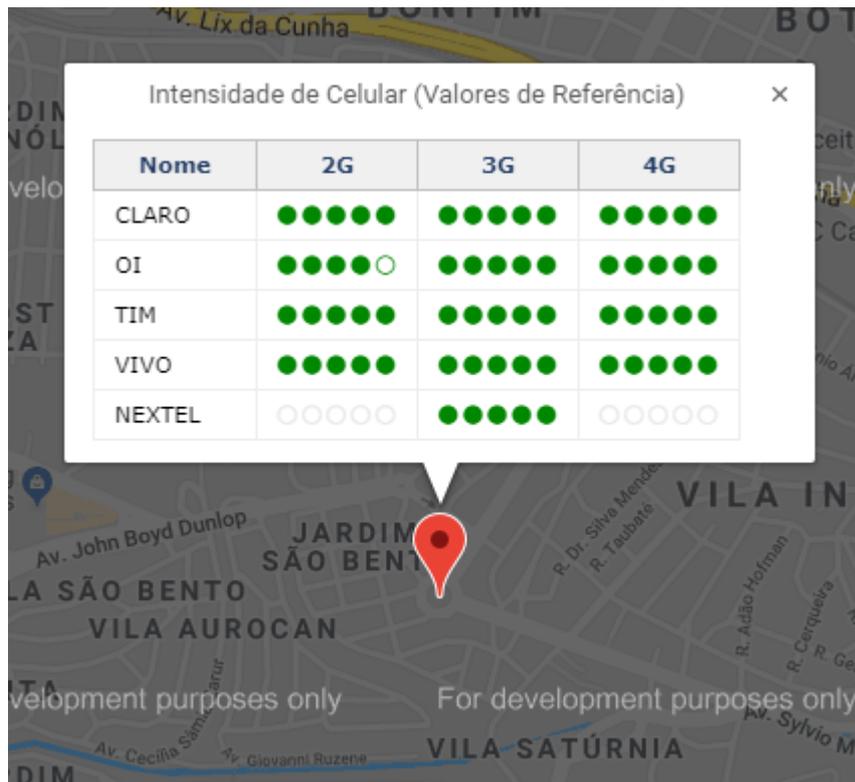
Figura 53: Detalhes de medição do ponto que apresentou o menor valor de RSRQ.

The screenshot shows the 'LTE Analyzer' app interface. At the top, there's a status bar with icons for WhatsApp, Facebook, Messenger, and Telegram, along with signal strength, Wi-Fi, and battery (38%) indicators. The app title 'LTE Analyzer' is displayed. Below it, a search prompt says 'Digite abaixo o ID do ponto que deseja consultar:'. A text input field contains the number '147'. Below the input field is a button labeled 'CONSULTAR'. Underneath, the results are listed under the heading 'Resultados:'. The results are: RSRP: -99, RSRQ: -20, RSSNR: -108, LATITUDE: -22.9105694, and LONGITUDE: -47.08690568.

Fonte: O autor.

Através do resultado da consulta no banco de dados mostrada na Figura 53 é possível observar que, além do valor de RSRQ estar anormal, os valores de RSRP e RSSNR também se encontram degradados e em faixas consideradas ruins. Segundo a teoria citada na Seção 2.4.2 deste trabalho, uma degradação conjunta dos valores de RSRQ e RSRP indica falhas de cobertura ou problemas graves na rede local. Além disso, o valor ruim de RSSNR indica que a amplitude dos ruídos e interferências é muito maior que da potência do sinal. Essas constatações podem ser usadas pela operadora para desenvolver estudos mais detalhados da rede no local. Como a consulta no banco de dados retorna também a latitude e a longitude, pode-se e comparar os valores medidos pelo *software* desenvolvido com os apresentados pelo sistema Mosaico no mesmo ponto. A Figura 54 mostra o resultado da pesquisa da intensidade do sinal no ponto em estudo através do sistema Mosaico. A busca feita através da latitude e longitude retornou a localidade de uma praça chamada Praça dos Viajantes, localizada nas mediações do bairro Vila Industrial.

Figura 54: Resultado de busca a ponto mínimo de RSRQ através do sistema Mosaico.



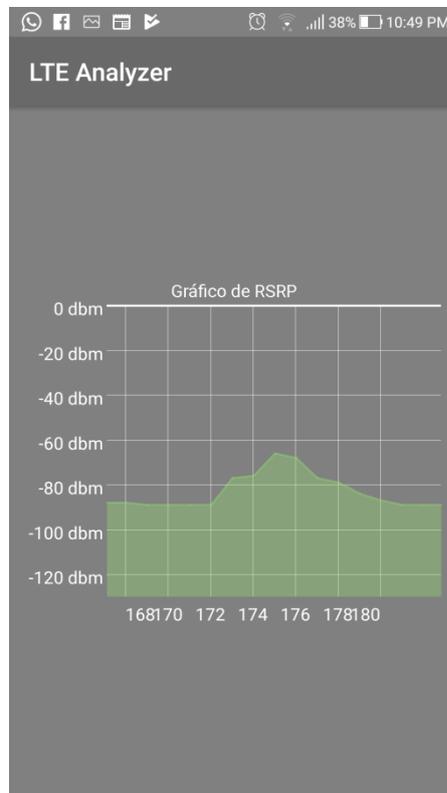
Fonte: Anatel (2018f).

Nota-se que o resultado da pesquisa feita através do sistema Mosaico mostra uma realidade diferente daquela mostrada pelo *software* desenvolvido, pois os sinais descritos por ele estão em níveis bons de qualidade. Existem algumas possibilidades que explicam essa

diferença, sendo uma delas a falha dos resultados mostrados pelo sistema Mosaico, visto que as informações mostradas por ele são obtidas através de previsões do sinal, e sendo assim, não são considerados fenômenos que podem ocorrer no dia-a-dia, como chuva, aglomeração de carros e pessoas, aumento de vegetação e etc. A diferença pode ser explicada principalmente por atenuações, como por exemplo, da chuva e umidade, visto que no cenário em que as medições foram feitas estava chovendo em grande parte do tempo, e atenuações devido à umidade e chuva podem ser severas para sistemas sem fio (BRAGA, 2015). Há também a presença de várias árvores de grande porte no local e alguns prédios nas mediações, o que pode causar áreas de sombra no sistema ou atenuações severas no sinal.

O ponto máximo RSRP também pode ser usado para comprovar o funcionamento correto do *software* desenvolvido. Analisando a Figura 48 pode-se constatar que o valor máximo de RSRP é de -66 dBm. Através da análise do gráfico de linhas de RSRP mostrado na Figura 49 encontrou-se o ponto de máximo que está detalhado na Figura 55 a seguir. Na Figura 55 nota-se que o ponto máximo está em torno do valor de identificação 176 que pode ser observado no eixo X do gráfico.

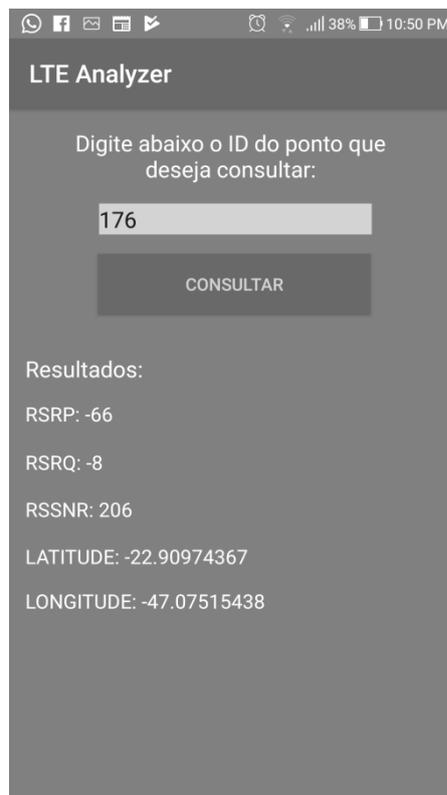
Figura 55: Ponto máximo de RSRP no gráfico de linhas.



Fonte: O autor.

Ao se fazer uma busca utilizando o *software* desenvolvido, o resultado mostrado na Figura 56 foi obtido. Na Figura 56 nota-se que os valores de RSRP e de RSRQ estão em faixas consideradas excelentes para o sinal, e que o valor de RSSNR está bastante elevado, o que significa que a amplitude do sinal é muito maior que a dos ruídos e interferências.

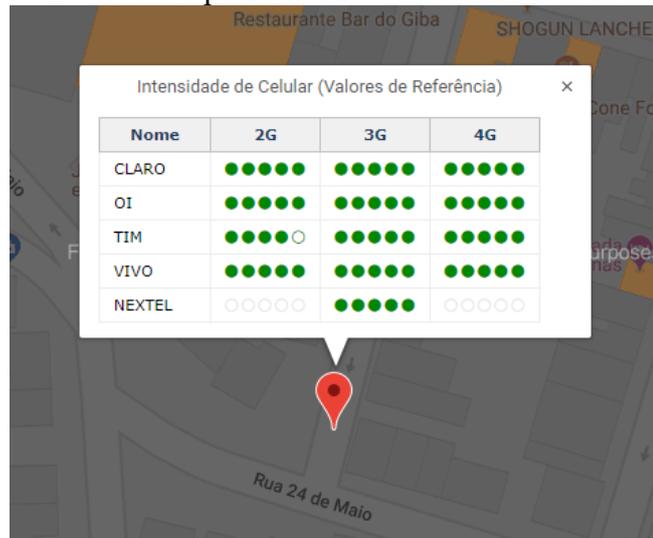
Figura 56: Resultado de busca dos detalhes do ponto de medição com valor máximo de RSRP



Fonte: O autor.

Agora, através dos valores de latitude e longitude obtidos na busca mostrada na Figura 56, pode-se fazer uma comparação dos valores medidos com os valores que a Anatel divulga através do sistema Mosaico. A Figura 57 mostra o resultado da pesquisa através da ferramenta Mosaico para a mesma localidade onde o valor de RSRP medido foi o máximo obtido.

Figura 57: Resultado de busca a ponto máximo de RSRP através do sistema Mosaico.



Fonte: Anatel (2018f).

Nota-se que o resultado da pesquisa mostrado na Figura 57 que foi feita através do sistema Mosaico mostra que o sinal no local tem um bom nível de qualidade, assim como foi medido pelo *software* desenvolvido. A principal explicação para a qualidade do sinal nesse ponto ser elevada é a proximidade do ponto com uma torre transmissora da operadora TIM. A distância de separação entre torre de transmissão e o receptor é um fator crítico, pois além de atenuações que o sinal sofre devido à propagação no espaço livre, os obstáculos que causam outros tipos de ruídos e atenuações também se tornam mais frequentes quando há um aumento da distância entre transmissor e receptor. O fator distância também pode ser usado para justificar os valores ruins obtidos no ponto da Praça dos Viajantes, que foi analisado anteriormente. Através de pesquisas no site da TIM encontrou-se a localização das torres de transmissão mais próximas dos pontos onde o valor mínimo de RSRQ e máximo de RSRP foram medidos. A Figura 58 mostra o mapa com as torres além de duas marcações que representam em verde o ponto onde o valor máximo de RSRP foi medido e em vermelho o ponto onde o valor mínimo de RSRQ foi medido. O ponto marcado em vermelho que representa o local onde valores ruins foram medidos (ponto onde o valor mínimo de RSRQ foi medido) encontra-se bem mais afastado de uma torre de transmissão em comparação com o ponto marcado em verde, onde valores bons foram medidos (ponto onde o valor máximo de RSRP foi medido).

5 CONCLUSÃO, TRABALHOS FUTUROS E CONTRIBUIÇÕES

Ao se analisar os resultados e discussões finais do trabalho presentes no Capítulo 4, afirma-se que todos os objetivos iniciais definidos na Seção 1.3 foram atendidos, pois através das teorias apresentadas no Capítulo 2, e principalmente através das bibliotecas que auxiliaram no desenvolvimento do aplicativo e foram citadas na Seção 2.6.1, o *software* desenvolvido permitiu a captura de informações utilizadas na análise de qualidade e detecção de problemas de redes móveis. Também foram desenvolvidas as funcionalidades extras propostas para o *software*, como a plotagem de gráficos, a construção de mapas e algumas análises estatísticas utilizando os dados medidos, que foram de suma importância na análise dos resultados finais, tornando-os mais práticos e fáceis de serem analisados.

Algumas comparações entre os dados medidos e os dados fornecidos pela operadora avaliada e pela Anatel foram feitas, e os resultados obtidos foram positivos, o que pode afirmar o correto funcionamento do *software*. Porém, para que o uso do aplicativo seja efetivado no futuro a nível profissional, mais testes devem ser realizados e mais parâmetros devem ser adicionados às medições. Um exemplo de teste a ser realizado para que se ateste com elevado grau de confiabilidade o funcionamento do *software* desenvolvido, é a comparação dos valores medidos pelo *software* com valores medidos por outros aplicativos já validados. Mais estudos também podem ser realizados para que se encontrem mais possibilidades de parâmetros que podem ser medidos, como por exemplo, mais informações sobre a célula à qual o smartphone está conectado, mais informações sobre *handover* e etc.

Na Seção 1.4 foi apresentada a problematização, que levantava a seguinte questão: Existe algum meio simples e barato de se coletar dados das redes móveis que torne possível a avaliação precisa do sinal proveniente desse tipo de serviço prestado pelas operadoras? A resposta para a problematização é positiva, pois os métodos utilizados neste trabalho são simples e baratos, e as medições mostraram que a precisão é aceitável, porém para garantir testes mais detalhados, algumas melhorias devem ser realizadas no *software* desenvolvido.

Na Seção 1.5 duas hipóteses foram levantadas. Depois de feitos os testes e as análises dos resultados, em resposta à Hipótese 1, percebeu-se que todas as variáveis como relevo, construções civis e vegetação podem produzir variações no sinal, e podem causar atenuações severas. Em resposta à Hipótese 2, e utilizando os resultados e discussões presentes no Capítulo 4 como base, afirma-se que através do uso de um *smartphone* é possível sim realizar

análises satisfatórias no sinal das redes móveis, porém, o uso do *software* desenvolvido atualmente permite somente análises mais sucintas, mas que podem ser muito úteis na identificação de problemas.

Algumas melhorias futuras podem ser adicionadas no *software* para que ele se torne mais preciso. Uma delas é a adição da separação de medições realizadas em condições diferentes, como por exemplo, ao se avaliar uma rede será possível identificar as condições meteorológicas e outras variáveis antes do início da medição, e ao gerar resultados o *software* irá analisar todas as condições registradas através de uma combinação dos valores medidos. Assim, acredita-se que os resultados poderão ser mais próximos do ideal.

Como trabalhos futuros, pode-se citar também a conexão deste aplicativo com a internet, e a disponibilização dele para todos os usuários da rede, e assim, eles poderão registrar suas próprias medidas e auxiliar as operadoras na coleta de dados, participando diretamente da melhoria dos serviços.

O trabalho poderá ser disponibilizado nos repositórios do GitLab, e assim, os códigos do *software* desenvolvido serão abertos a toda a comunidade de desenvolvedores, auxiliando futuros colegas de profissão e principalmente de universidade que queiram desenvolver trabalhos ligados à análise da qualidade de sinais de redes móveis. Também se espera que o *software* desenvolvido possa contribuir com operadoras na melhoria e detecção de problemas em suas redes.

REFERÊNCIAS

3GPP. **UTRA-UTRAN Long Term Evolution (LTE) and 3GPP System Architecture Evolution (SAE)**. Disponível em: <ftp://ftp.3gpp.org/Inbox/2008_web_files/LTA_Paper.pdf>. Acesso em: novembro de 2018. (2018a)

_____. **LTE**. Disponível em: <<http://www.3gpp.org/technologies/keywords-acronyms/98-lte>>. Acesso em: novembro de 2018. (2018b)

4G LTE MALL. **Product Description** . Disponível em <https://www.4gltemall.com/blog/wpcontent/uploads/2012/11/HUAWEI_E392_TDD_LTE_USB_Stick_Product_Description-Specification-and-datasheet.pdf>. Acesso em: novembro de 2018.

AFROZ, Farhana et al. **SINR, RSRP, RSSI and RSRQ measurements in long term evolution networks**. International Journal of Wireless & Mobile Networks, 2015.

ALBERO, Fernando Molina; ŠTERN, Andrej; KOS, Andrej. **Mobile network measurements using Android**. ERK-2013 International Electrotechnical and Computer Science Conference. 2013. p. 97-100.

ANDROIDPRO. **Guardando dados com SQLite**. Disponível em: <<https://www.androidpro.com.br/blog/armazenamento-de-dados/sqlite/>>. Acesso em: novembro de 2018.

ANATEL. **Portal Institucional**. Disponível em: <www.anatel.gov.br>. Acesso em: novembro de 2018. (2018a)

_____. **Portal do consumidor: cobertura e zona de sombra**. Disponível em: <<http://www.anatel.gov.br/consumidor/banda-larga/direitos/83-seus-direitos/telefoniamovel/direitos>>. Acesso em: novembro de 2018. (2018b)

_____. **Velocidade de conexão à Internet**. Disponível em <http://www.anatel.gov.br/consumidor/index.php?option=com_content&view=article&id=231&Itemid=707>. Acesso em: novembro de 2018c. (2018c)

_____. **Telefonia móvel - Acessos**. Disponível em: <<http://www.anatel.gov.br/dados/destaque-1/270-destaques-smp>>. Acesso em: novembro de 2018. (2018d)

_____. **Mapa de Competição do Serviço Móvel Pessoal Brasil.** Disponível em: <<http://www.anatel.gov.br/dados/destaque-1/254-mapa-de-competicao-do-smp-brasil-competicao-telefonica-movel>>. Acesso em: novembro de 2018. (2018e)

_____. **Anatel lança ferramenta para consulta de intensidade de sinal celular.** Disponível em: <<http://www.anatel.gov.br/institucional/ultimas-noticiass/1208-anatel-lanca-ferramenta-para-consulta-de-intensidade-sinal-celular>>. Acesso em: novembro de 2018. (2018f)

_____. **Portaria nº 1718, de 07 de dezembro de 2017.** Disponível em: <<http://www.anatel.gov.br/legislacao/procedimentos-de-fiscalizacao/972-portaria-1718>>. Acesso em: novembro de 2018. (2018g)

BRAGA, Caio et al. **ENLACE DE MICRO-ONDAS RELACIONADO A ATENUAÇÃO DEVIDO À CHUVA.** XII Conferência de Estudos em Engenharia Elétrica, Uberlândia, 2015.

DEBNATH, Arunabha, ROBERT, Abhinandan. **Review on NEXT Generation Technologies of Wireless Communication.** International Journal on Recent and Innovation Trends in Computing and Communication, v. 1, n. 4, p. 346 – 350, 2013.

DE SOUZA, Tiago Barros; GUARDIEIRO, Paulo Roberto. **QUALIDADE DE SERVIÇO EM REDES MÓVEIS DE 4º GERAÇÃO LTE.** Conferência de estudos em Engenharia Elétrica, 2012.

DEVELOPERS. **Arquitetura da plataforma.** Disponível em <<https://developer.android.com/guide/platform/?hl=pt-br>>. Acesso em: novembro de 2018. (2018a)

_____. **API Reference.** Disponível em <<https://developer.android.com/reference>>. Acesso em: novembro de 2018. (2018b)

_____. **“Android.telephony”.** Disponível em <<https://developer.android.com/reference/android/telephony/package-summary>>. Acesso em: novembro de 2018. (2018c)

_____. **Android Studio.** Disponível em <<https://developer.android.com/studio/>>. Acesso em: novembro de 2018. (2018d)

_____. **Overview.** Disponível em <<https://developers.google.com/maps/documentation/android-sdk/intro>>. Acesso em: novembro de 2018. (2018e)

GRAPHVIEW. **GraphView - open source graph plotting library for Android**. Disponível em < <http://www.android-graphview.org/>>. Acesso em: novembro de 2018.

GSMA. **The Mobile Economy 2018**. Disponível em <<https://www.gsma.com/mobileeconomy/wp-content/uploads/2018/02/The-Mobile-Economy-Global-2018.pdf>>. Acesso em: novembro de 2018.

IESB, Instituto de Educação Superior de Brasília. **Tecnologias de redes móveis**. Disponível em:<
https://iesb.blackboard.com/bbcswebdav/institution/Ead/_disciplinas/EADG456/T11/sections/pdf/TASK5369.pdf>. Acesso em: novembro de 2018.

ITU, International Telecommunications Union. **The World in 2013: ICT facts and figures**. Disponível em:<<http://www.itu.int/en/ITU-D/Statistics/Documents/facts/ICTFactsFigures2013-e.pdf>>. Acesso em: novembro de 2018.

LEG. **Usando simulação para mostrar resultados**. Disponível em <<http://www.leg.ufpr.br/~paulojus/embrapa/Rembrapa/Rembrapase28.html>>. Acesso em: novembro de 2018.

MENDES, José Renato Ribeiro. **5G: a quinta geração**. Monografia (Especialista em Teleinformática e Redes de Computadores). Universidade Tecnológica Federal do Paraná, UTFPR, 2014.

MICROWAVE JOURNAL. **Drive Test Software Offers Complete LTE MIMO Measurements**. Disponível em <<http://www.microwavejournal.com/articles/print/17147-drive-test-software-offers-complete-lte-mimo-measurements>>. Acesso em: novembro de 2018.

PUC-RIO. **Tecnologia LTE (Long Term Evolution)**. Disponível em: <
https://www.maxwell.vrac.puc-rio.br/18484/18484_3.PDF>. Acesso em: novembro de 2018.

RAPPAPORT, Theodore S. et al. **Wireless communications: principles and practice**. New Jersey: prentice hall PTR, 1996.

ROLIN, Evandro Cherubini. **Análise da qualidade e do desempenho da cobertura celular long term evolution em Curitiba, Paraná**. Dissertação (mestrado em Engenharia Elétrica). Universidade Federal do paraná, 2014.

ROHDE E SCHWARZ. **LTE Drive Test - How to benefit from using a R&S®TSMW or R&S®TSME**. Disponível em < <https://cdn.rohde->

schwarz.com/pws/dl_downloads/dl_application/application_notes/1sp17/1SP17_4e_LTE_Drive_Test_Scanner.pdf >. Acesso em: novembro de 2018.

_____. **QualiPoc Android The premier handheld troubleshooter.** Disponível em <https://cdn.rohde-schwarz.com/pws/dl_downloads/dl_common_library/dl_brochures_and_datasheets/pdf_1/service_support_30/Rohde-Schwarz_QualiPoc_Android_3607-1607-12_v0200_120dpi.pdf>. Acesso em: novembro de 2018.

_____. **R&S@ROMES4 Drive Test Software Mobile coverage and QoS measurements in mobile networks.** Disponível em <https://cdn.rohde-schwarz.com/pws/dl_downloads/dl_common_library/dl_brochures_and_datasheets/pdf_1/ROMES4_bro_en_5214-2062-12_v1200.pdf>. Acesso em: novembro de 2018.

STATISTA. **Number of mobile phone users worldwide from 2013 to 2019 (in billions).** Disponível em <<https://www.statista.com/statistics/274774/forecast-of-mobile-phone-users-worldwide/>>. Acesso em: novembro de 2018.

TELECOM HALL BR. **O que é Drive Test (ou Drive Testing) de RF?** Disponível em <<http://www.telecomhall.com/Br/o-que-e-drive-test-testing-de-rf.aspx>>. Acesso em: novembro de 2018.

TIM. **Mapa de cobertura.** Disponível em <<http://www.tim.com.br/rj/para-voce/cobertura-e-roaming/mapa-de-cobertura>>. Acesso em: novembro de 2018. (2018a)

TIM. **Antenas.** Disponível em <<https://tim-geoportal.geoportal3d.com.br/>>. Acesso em: novembro de 2018. (2018b)

USUÁRIOS DO EXCEL. **Construindo uma curva de Gauss no Excel.** Disponível em <<https://usuariosdoexcel.wordpress.com/2013/05/17/0161-construindo-uma-curva-de-gauss-no-excel/>>. Acesso em: novembro de 2018.