



Universidade Federal
do Rio de Janeiro

Escola Politécnica

BANCO DE DADOS DE DISPONIBILIDADE DE ESPECTRO PARA RÁDIOS COGNITIVOS

Marcelo Rocha Machado

Projeto de Graduação apresentado ao Curso de Computação e Informação da Escola Politécnica da Universidade Federal do Rio de Janeiro como parte dos requisitos necessários para a obtenção do grau de Engenheiro de Computação e Informação.

Orientadores: José Ferreira de Rezende
Marcel William Rocha da Silva

Rio de Janeiro
Abril de 2013

BANCO DE DADOS DE DISPONIBILIDADE DE ESPECTRO PARA RÁDIOS
COGNITIVOS

Marcelo Rocha Machado

PROJETO SUBMETIDO AO CORPO DOCENTE DO CURSO DE
COMPUTAÇÃO E INFORMAÇÃO DA ESCOLA POLITÉCNICA DA
UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE
DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE
ENGENHEIRO DE COMPUTAÇÃO E INFORMAÇÃO.

Examinadores:

Prof. José Ferreira de Rezende, Dr.

Prof. Marcel William Rocha da Silva, D.Sc.

Prof. Aloysio de Castro Pinto Pedroza, Dr.

Prof. Ricardo Guerra Marroquim, D.Sc.

RIO DE JANEIRO, RJ – BRASIL
ABRIL DE 2013

Machado, Marcelo Rocha

Banco de Dados de Disponibilidade de Espectro para Rádios Cognitivos/Marcelo Rocha Machado. – Rio de Janeiro: UFRJ/POLI – COPPE, 2013.

XII, 72 p.: il.; 29,7cm.

Orientadores: José Ferreira de Rezende

Marcel William Rocha da Silva

Projeto (graduação) – UFRJ/ Escola Politécnica/ Curso de Computação e Informação, 2013.

Referências Bibliográficas: p. 70 – 72.

1. Rádios Cognitivos. 2. Banco de Dados. 3. REST. I. de Rezende, José Ferreira *et al.* II. Universidade Federal do Rio de Janeiro, Escola Politécnica/ Curso de Computação e Informação. III. Título.

Agradecimentos

Agradeço aos meus pais que me incentivam desde meu nascimento. Agradeço a meu irmão, um incômodo necessário. Agradeço a meus amigos, os novos, os velhos, os ausentes e os presentes. Agradeço a minha namorada, que me suporta em todos os momentos. Agradeço a meus orientadores, Rezende e Marcel. Agradeço ao restante da minha família, muito grande para agradecer a todos individualmente, mas muito importantes para mim.

Resumo do Projeto de Graduação apresentado à Escola Politécnica/COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Engenheiro de Computação e Informação.

BANCO DE DADOS DE DISPONIBILIDADE DE ESPECTRO PARA RÁDIOS COGNITIVOS

Marcelo Rocha Machado

Abril/2013

Orientadores: José Ferreira de Rezende

Marcel William Rocha da Silva

Curso: Engenharia de Computação e Informação

Rádios Cognitivos utilizam canais do espectro de rádio frequência para transmitir dados, mudando a frequência em uso de acordo com sua disponibilidade. No entanto, alguns canais são licenciados para uso e, portanto, são usados preferencialmente pelos dispositivos que detêm essa licença. Outros dispositivos devem identificar os canais que não estão em uso por unidades licenciadas para escolher um canal disponível para uso. O trabalho consiste em desenvolver uma base de dados na qual dispositivos possam pesquisar os canais disponíveis para uso no Estado do Rio de Janeiro. Foi feita também uma avaliação sobre a disponibilidade de canais nessa mesma região.

Palavras-Chave: Rádios Cognitivos, Banco de Dados, REST.

Abstract of the Undergraduate Project presented to Poli/COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Computer and Information Engineer.

SPECTRUM AVAILABILITY DATA BASE FOR COGNITIVE RADIOS

Marcelo Rocha Machado

April/2013

Advisors: José Ferreira de Rezende
Marcel William Rocha da Silva

Course: Computer and Information Engineering

Cognitive Radios use radio frequency channels to transmit data, changing the used frequencies as they became available. However, some channels have its use licensed and therefore are used preferably by devices which hold said license. Other devices must identify channels that are not being used by those licensed unit in order to choose a channel for use. This works aims at developing a database which devices can query for the available channel in the State of Rio de Janeiro. An analysis was made regarding the channel availability in said region.

Keywords: Cognitive Radios, Data Base, REST.

Sumário

Lista de Figuras	ix
Lista de Tabelas	xi
Lista de Abreviaturas	xii
1 Introdução	1
1.1 Motivação	2
1.2 Objetivo	3
1.2.1 Metodologia	3
2 Base de Dados de WhiteSpace	4
2.1 Soluções existentes	4
2.1.1 Ferramentas	6
2.2 Protocol to Access White Space Database (PAWS)	6
2.2.1 Descrição Geral	7
2.2.2 Funcionamento	8
3 Base de Dados Proposta	11
3.1 Informações Gerais	11
3.1.1 Desenvolvimento	12
3.1.2 Definições	12
3.2 Coleta de dados	14
3.2.1 Implementação	15
3.2.2 Extensão	15
3.2.3 Qualidade da Informação	16
3.3 Base de Dados	17
3.3.1 Data Access Object	18
3.3.2 Implementação	19
3.3.3 Extensão	19
3.3.4 Canais de transmissão	23
3.4 Determinação do White Space	25

3.4.1	Algoritmo	25
3.4.2	Implementação	30
3.4.3	Extensão	30
3.5	Web Service Desenvolvido	31
3.5.1	REST	31
3.5.2	Implementação	33
3.5.3	Extensão	33
3.5.4	API	33
4	Análises Realizadas	51
4.1	Dataset criado	51
4.2	Disponibilidade de canais no Rio de Janeiro	52
4.2.1	Modelo Free Spaces	53
4.2.2	Modelo Two Ray Ground	55
4.2.3	Modelo Log Distance	56
4.3	Resultado da análise	64
4.3.1	Diferença entre Modelos	64
4.3.2	Conclusão	65
5	Conclusão e Trabalhos Futuros	67
5.1	Conclusão	67
5.2	Trabalhos Futuros	67
5.2.1	Parceria com a ANATEL	67
5.2.2	Cálculo de Disponibilidade	68
5.2.3	PAWS	69
5.2.4	Geolocalização	69
	Referências Bibliográficas	70

Lista de Figuras

2.1	<i>Diagrama de funcionamento das Bases de Dados de WhiteSpace.</i>	6
2.2	<i>Protocol to Access White Space Database.</i>	7
2.3	<i>Funcionamento do PAWS.</i>	8
3.1	<i>Diagrama do Sistema Desenvolvido.</i>	11
3.2	<i>Factories.</i>	12
3.3	<i>Interferência.</i>	13
3.4	<i>Antena.</i>	16
3.5	<i>Parser.</i>	17
3.6	Diagrama com o processo de Parsing.	18
3.7	Vizualização de antenas indicadas pela ANATEL.	18
3.8	Classes de Dados (DO).	20
3.9	Diagrama Entidade Relacionamento.	21
3.10	Exemplo da classe <i>AntenaMYSQLDao</i> .	21
3.11	Diagrama de classes para montar a base de dados.	22
3.12	Diagrama de classes DAO do MYSQL.	22
3.13	Modelo de propagação Free Space.	27
3.14	Modelo de propagação Two Ray Ground.	28
3.15	<i>Diagrama de classes dos modelos de propagação desenvolvidos.</i>	31
3.16	<i>Diagrama da classe DeviceRange</i>	34
3.17	Diagrama da classe <i>Channel</i> .	35
3.18	<i>Diagrama da classe ChannelAPI</i>	37
3.19	<i>Diagrama da classe AntenaAPI</i>	39
3.20	<i>Diagrama da classe WSDeviceAPI</i>	42
3.21	<i>Diagrama da classe TransmissionDeviceAPI</i>	47
4.1	Dataset Utilizado	52
4.2	Mapa de calor para o modelo Free Space	54
4.3	Histograma de disponibilidade de canais para o modelo Free Space	54
4.4	Mapa de calor para o modelo Two Ray Ground	55

4.5	Histograma de disponibilidade de canais para o modelo Two Ray Ground	56
4.6	Mapa de calor para o modelo Log Distance usando 2 como constante de decaimento	57
4.7	Histograma de disponibilidade de canais para o modelo Log Distance usando 2 como constante de decaimento	57
4.8	Mapa de calor para o modelo Log Distance usando 2,5 como constante de decaimento	58
4.9	Histograma de disponibilidade de canais para o modelo Log Distance usando 2,5 como constante de decaimento	59
4.10	Mapa de calor para o modelo Log Distance usando 3 como constante de decaimento	60
4.11	Histograma de disponibilidade de canais para o modelo Log Distance usando 3 como constante de decaimento	60
4.12	Mapa de calor para o modelo Log Distance usando 3,5 como constante de decaimento	61
4.13	Histograma de disponibilidade de canais para o modelo Log Distance usando 3,5 como constante de decaimento	62
4.14	Mapa de calor para o modelo Log Distance usando 4 como constante de decaimento	63
4.15	Histograma de disponibilidade de canais para o modelo Log Distance usando 4 como constante de decaimento	63

Lista de Tabelas

3.1	Informações sobre os canais de transmissão.	23
-----	---	----

Lista de Abreviaturas

ANATEL	Agência Nacional de Telecomunicações, p. 1
BDWS	Bases de Dados de WhiteSpace, p. 4
CEPT	European Conference of Postal and Telecommunications Administrations, p. 1
DAO	Data Access Object, p. 18
DE	Dispositivo Escravo, p. 7
DM	Dispositivo Mestre, p. 7
ECC	Electronic Communications Committee, p. 1
FCC	Federal Communications Commission, p. 1
IETF	Internet Engineering Task Force, p. 6
PAWS	Protocol to Access White Space Database, p. 6
RFC	Request For Comments, p. 7
SGBD	Sistema de Gerenciamento de Banco de Dados, p. 11
UP	Usuários Primários, p. 1
URI	Uniform Resource Identifier, p. 32
US	Usuários Secundários, p. 1
WISE	WhIte Space test Environment for broadcast frequencies, p. 5
WS	WhiteSpaces, p. 1

Capítulo 1

Introdução

As frequências de rádio transmissão são bens finitos e preciosos. O licenciamento de canais é feito por uma agência reguladora, uma organização única de cada país. No Brasil isso é feito pela Agência Nacional de Telecomunicações (ANATEL), nos Estados Unidos fica a cargo da Federal Communications Commission (FCC), que é tomada como referência para estudos nesse campo. Na Europa o licenciamento fica a cargo da Electronic Communications Committee (ECC) que faz parte do European Conference of Postal and Telecommunications Administrations (CEPT).

A entidade que comprar a licença tem o direito de utilizar um determinado canal de frequência em uma determinada região. No entanto, esses canais acabam sendo subutilizados. Podem existir locais onde a entidade que detém a licença para uma determinada frequência não a utilize. Esses espaços licenciados, porém não utilizados, são chamados de WhiteSpaces (WS). Os WSs constituem um desperdício, já que uma banda que poderia estar em uso fica reservada para o detentor da licença. Dispositivos de transmissão da entidade detentora da licença são chamados de “Usuários Primários” (UP).

Pela definição do IEEE [1], rádios cognitivos são dispositivos móveis que podem mudar a frequência de transmissão que utilizam. Esses dispositivos se aproveitam do WS podendo utilizar as frequências licenciadas que não estão em uso pelos UPs. Para isso os dispositivos não licenciados, ou “Usuários Secundários”(US) devem ser capazes de determinar as frequências disponíveis. Um rádio cognitivo deve, então, ser capaz de escolher um canal de frequência para a transmissão de dados, podendo modificar esse canal para utilizar uma frequência que não esteja em uso pelos UPs. Não é do escopo desse trabalho o modo como um rádio cognitivo muda de um canal para o outro. É assumido que os rádios cognitivos conseguem modificar sua frequência de transmissão de acordo com o canal que for selecionado.

Um rádio cognitivo deve evitar a situação de interferência. Quando um US interferir no funcionamento de um UP, ou seja, quando ele interfere com seus receptores, ele deve imediatamente interromper a utilização daquela frequência e iniciar uma

busca por uma nova. Essa situação acarreta um custo para o dispositivo, já que ele vai precisar interromper sua operação até determinar uma nova frequência a ser utilizada.

Existem dois modelos definidos para que os rádios cognitivos determinem qual frequência utilizar.

Uma maneira é pelo sensoriamento. Nela, o US deve analisar a potência recebida em diferentes frequências de forma a determinar qual está disponível. Esse modo, embora já tenha sido bastante pesquisado, possui algumas particularidades de implementação. Para realizar o sensoriamento um hardware específico é necessário, o que aumenta o seu custo de manufatura bem como seu uso de energia. Métodos para realizar o sensoriamento do espectro estão fora do escopo desse trabalho

Outra é utilizar uma base de dados que possua informações sobre a disponibilidade do espectro de frequência. Essa base tira a responsabilidade de escolher uma frequência do US, concentrando todo o poder de decisão na base que contenha essas informações. Entretanto, para que um dispositivo acesse essa base é preciso que ele já esteja conectado em uma rede, o que impõe certos limites à sua implementação.

1.1 Motivação

O modelo de base de dados representa uma forma de controle maior do WS. As agências reguladoras atualmente tem o controle sobre o licenciamento do espectro de frequência. Caso os USs comecem a utilizar o modelo de sensoriamento, as agências perdem o controle sobre o uso do espectro de frequência. Uma vez que as frequências disponíveis passam a ser determinadas por uma base de dados vinculada ao órgão regulador, essa entidade pode atualizar regras sobre a disponibilidade do espectro rapidamente, sem se preocupar com dispositivos utilizando esse recurso.

Leve em consideração o seguinte exemplo. Assumindo que os US utilizem o sensoriamento, um determinado canal C pode ter sido escolhido como disponível para uma determinada região. Caso a agência reguladora conceda essa canal C para alguma operação, ocorrerá uma interferência entre os dispositivos até que os USs escolham um novo canal. Os USs vão ter que escolher um novo canal, perdendo a conexão que haviam estabelecido na frequência anterior e sendo obrigados a realizar o sensoriamento novamente. Caso já estivesse sendo utilizado o modelo de base de dados o órgão regulador poderia mudar regras nessa base e até comunicar aos dispositivos que o canal passará a ser utilizado. Dessa forma, a condição de interferência será evitada, bem como a perda não planejada do sinal.

Devido à importância do controle de uso do WS esse modelo de base de dados recebeu grande atenção da FCC. Como será aprofundado no próximo capítulo, diversos países já possuem uma base dados em operação, e muitos outros já as estão

desenvolvendo. Além disso, existe um protocolo sendo criado que padroniza a comunicação entre dispositivos e bases. Dispositivos que operam no WS já estão sendo criados, e a criação de uma base passa a ser pré-requisito para sua operação. Fica evidente a carência brasileira por uma base que provenha esse serviço.

1.2 Objetivo

O objetivo do sistema desenvolvido foi criar um marco inicial para o estabelecimento de uma base de dados de disponibilidade de canais para rádios cognitivos no Brasil. Foi almejado, também, desenvolver uma base que possa ser utilizada para futuras pesquisas na área.

1.2.1 Metodologia

No trabalho realizado foi tentado reproduzir as capacidades de bases já bem definidas, como a Spectrum Bridge. Para isso foram buscadas informações sobre os DPs em funcionamento no Rio de Janeiro. Foi, então, modelada uma base que contenha tais informações. Foram implementados modelos de propagação para identificar possíveis WS no Estado. Finalmente um Web Service foi desenvolvido para comunicação com tal base. Concluindo, o Web Service foi utilizado para extrair informações sobre a disponibilidade de canais no Estado do Rio de Janeiro.

Capítulo 2

Base de Dados de WhiteSpace

Uma Base de Dados de WhiteSpace (BDWS) deve prover um serviço: indicar a um US as frequências do WS disponíveis para uso. Para que isso seja possível, a base precisa de informações sobre todos os UPs existentes. Dessa forma, ela consegue determinar, para uma dada localização, as frequências que não estão em uso pelos UPs.

O espectro de frequência, no entanto, é bem amplo e, portanto, o intervalo de frequências utilizadas deve ser restrito. O FCC possui definições para uso de frequência que vão desde 9kHz (Rádio Navegação) até 275 GHz (Rádio Astronomia) [2]. Será considerado apenas o espectro de frequência reservado para televisão, que vai de 54 Mhz até 806Mhz.

Assim, para poder prover esse serviço, o BDWS deve possuir informações sobre todas as antenas de televisão que operam em uma determinada região. A falta de informação sobre as antenas, ou alguma informação errada, acarreta em erros no cálculo, podendo o canal indicado como disponível estar em uso. É imprescindível que exista uma relação de cooperação entre a agência reguladora e a base de dados para garantir a qualidade das informações usadas.

No trabalho desenvolvido, a região em questão foi restrita ao Estado do Rio de Janeiro. Isso foi feito de forma a simplificar a base de informações que foi construída.

Deve ser destacado, também, o caso de microfones sem fio. Tais microfones, quando utilizados em grandes eventos, precisam operar no WS. Assumindo um universo onde o uso do WS está difundido pelas BDWS, tais microfones podem gerar uma interferência com outros DS. Para isso a FCC determinou que tais eventos devem buscar a frequência a ser utilizada pelas BDWSs em operação [3].

2.1 Soluções existentes

Como já foi mencionado, é de interesse da ANATEL, ou de qualquer outra agência reguladora, controlar o uso do WS. Tanto que, no final de 2011, a FCC se uniu a um

grupo de empresas de forma a definir regras para a utilização do WS. Esse grupo seria responsável pela criação e administração das BDWS [4]. Esse grupo recebeu grande apoio da FCC, diversos workshops foram organizados e o órgão regulador americano o auxiliava provendo todas as informações referentes às antenas de TV e outros dispositivos que utilizam o espectro de frequência em análise [5].

As seguintes empresas participam desse grupo:

- Frequency Finder, Inc.
- Google
- KB Enterprises LLC and LS Telcom
- Key Bridge Global LLC
- NeuStar, Inc.
- Spectrum Bridge, Inc.
- Telcordia Technologies, Inc.
- Airity, Inc.
- Microsoft

O objetivo desse grupo era determinar um modo de operação para as BDWS. As bases, independente do administrador, deveriam ser síncronas, disponíveis publicamente e livres de royalties. O caminho para a padronização está sendo alcançado pelo desenvolvimento de um protocolo, explicado com mais detalhes na seção seguinte.

Desse grupo podemos destacar o Spectrum Bridge que já opera um serviço de base de dados aprovado pelo FCC [6]. Essa empresa possui um portal onde se indica uma localização e a lista de canais disponíveis é retornada. Existe também um aplicativo para android que realiza o mesmo serviço [7].

Existem outros grupos que desenvolvem trabalhos similares pelo mundo. Pode ser comentado o White Space test Environment for broadcast frequencies (WISE), um consórcio entre duas universidades finlandesas e algumas empresas da região. Ele está relacionado ao CEPT, a agência reguladora europeia. Seu trabalho consiste na formulação de algoritmos que consigam calcular devidamente a disponibilidade de canais no WS. Esses algoritmos são confirmados por pesquisas realizadas em campo.

A partir desses trabalhos já desenvolvidos pode-se descrever o funcionamento geral das BDWS. Podem existir diversas bases diferentes na mesma região, todas

operando com autorização da agência reguladora competente. Essas bases devem estar conectadas na internet, de onde virão pedidos de disponibilidade de canais de diversos dispositivos diferentes. A necessidade de padronização dessa comunicação culminou no desenvolvimento de um protocolo, que será explicado na próxima seção.

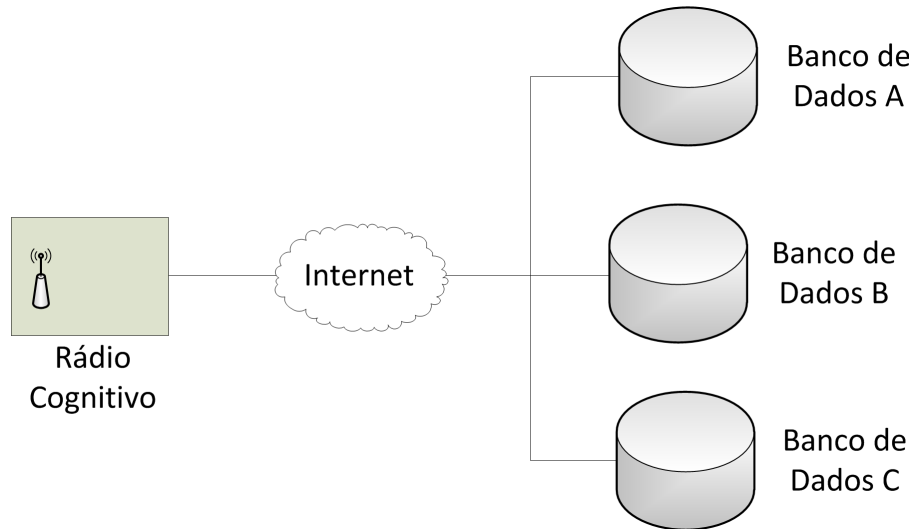


Figura 2.1: Diagrama de funcionamento das *BDWS*.

2.1.1 Ferramentas

Além dessas soluções para a determinação de disponibilidade de canais no White Space, existem outras ferramentas úteis na internet. Como será discutido mais adiante, no trabalho desenvolvido foi calculado o alcance das antenas de televisão sem levar em consideração as variações de relevo. Essas ferramentas realizam justamente esse trabalho, de calcular o alcance de uma antena dado que ela esteja em um determinado ponto do planeta.

É possível destacar duas ferramentas: a "Tower Coverage" [8] e a "Radio Mobile" [9]. O diferencial delas é que utilizam o modelo de propagação Longley-Rice [10], que leva em consideração o terreno ao redor dos dispositivos.

2.2 Protocol to Access White Space Database (PAWS)

O Internet Engineering Task Force (IETF) é a entidade responsável pela criação de padrões e protocolos que são utilizados mundialmente no campo de redes de computadores. Existe um grupo de trabalho do IETF desenvolvendo um protocolo cujas BDWS devem seguir para se comunicar com os dispositivos clientes, o Protocol to Access White Space Database (PAWS) [11].

Protocolos desenvolvidos pela IETF seguem um ciclo de vida bastante conhecido, são criados como um resumo (draft) e são, então, aprimorados até serem homologados pelo IETF. Depois de homologados os protocolos recebem a alcunha de Request For Comments (RFC) e devem ser adotados por todos os indivíduos e corporações que utilizem o serviço. No momento o PAWS ainda está em formato de draft, ou seja, ainda não pode ser considerado um padrão a ser seguido. No entanto, esse rascunho serve como uma base sobre a qual ele será desenvolvido. Podemos, então, retirar dele a ideia geral de como se portará o protocolo.

O trabalho não seguiu as orientações do IETF, mas foi construído de maneira que elas podem ser facilmente implementadas. De qualquer maneira, é válido apresentar as características básicas de protocolo proposto pelo PAWS. O sistema desenvolvido será assunto da próxima seção.

2.2.1 Descrição Geral

O PAWS busca padronizar o modo como as bases de BDWS são acessadas. O protocolo define a maneira que os dispositivos que desejam operar no WS devem buscar e utilizar as bases de dados. São definidos dois tipos de dispositivos, os Dispositivos Mestre(DM) e os Dispositivos Escravo (DE). Os DMs são os que vão se conectar às bases de dados. Os DEs realizam um pedido a um DM, que o encaminha às bases. O PAWS define a maneira como os DMs se comunicam com as bases. A interação entre as agências reguladoras e as bases, bem como entre os DMs e DEs não foi abordado, embora sua importância não seja pequena.

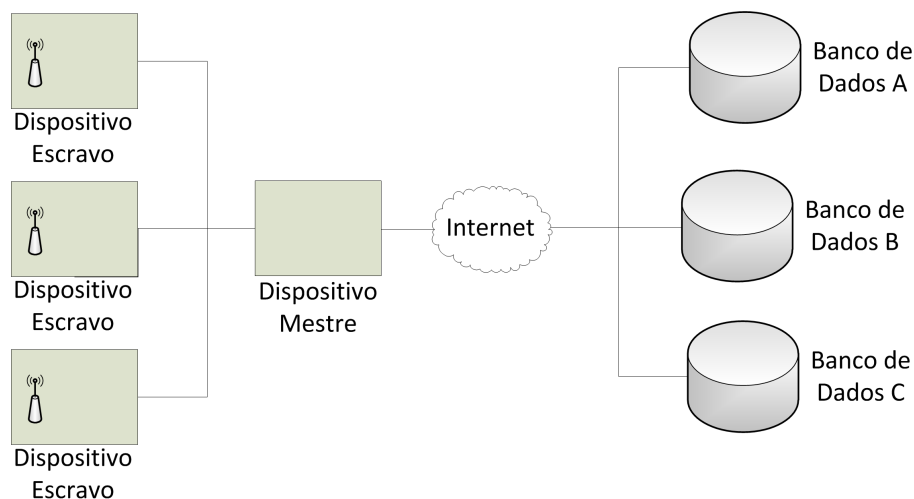


Figura 2.2: Diagrama do PAWS.

O protocolo assume que tanto os DMs quanto as BDWSs estarão conectados na internet. A comunicação entre essas entidades ocorre por um Web Service que opera em JSON-RPC [12], utilizando HTTP/TLS [13] como transporte.

O protocolo não garante que um canal de frequência indevido não seja utilizado, já que ele define apenas o canal de comunicação entre dispositivo e base de dados. Também não é definida uma autenticação dos dispositivos com as bases, usando o protocolo TLS para garantir a segurança das informações transmitidas.

É deixada em aberto a possibilidade para, futuramente, existirem micro leilões de canais do WS.

Vale ressaltar que o protocolo se encontra em rascunho, e pode ser modificado no futuro. Ele está sendo desenvolvido com diversos pontos de extensão já que cada agência reguladora pode impor algumas condições para o uso das bases. Muitos procedimentos implementados são regras impostas pela FCC.

2.2.2 Funcionamento

Foram definidos alguns casos de uso de uma BDWS e as características que um protocolo deveria ter [14]. Embora o trabalho feito não atenda a todas essas necessidades, foi construído um alicerce sobre o qual as funcionalidades faltantes podem ser construídas. Será traçado um paralelo entre o proposto pelo PAWS e o efetivamente construído.

São definidas etapas para a comunicação entre base e dispositivos:

- Descoberta
- Inicialização
- Registro
- Pedido
- Validação

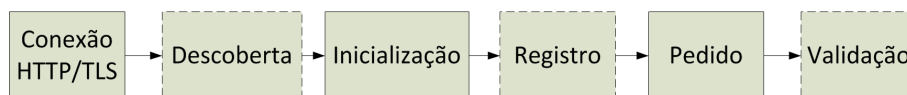


Figura 2.3: Diagrama de funcionamento do *PAWS*.

Descoberta

A comunicação entre dispositivo e base pode incluir a troca de dados sensíveis. Além disso, o dispositivo deve poder confiar na qualidade da informação que a base possui. Para prover essas garantias está sendo desenvolvido um protocolo que os dispositivos podem utilizar para adquirir um endereço de uma base de dados confiável [15].

No trabalho desenvolvido essa parte não foi implementada.

Inicialização

Depois de um dispositivo ter estabelecido uma seção TLS com a base ele deve fazer um pedido de inicialização de serviço. O DM deve informar alguns dados básicos como número de série e localização.

A base responde a esse pedido com algumas informações sobre sua utilização. Dessas informações pode ser destacada a movimentação máxima. Se o dispositivo se afastar mais do que a distância informada da sua localização ele deve realizar um novo pedido a base, já que a disponibilidade de canais pode ser diferente.

No trabalho desenvolvido essa parte não foi implementada.

Registro

A FCC exige que dispositivos que queiram utilizar a BDWS devem se registrar antes de fazer um pedido de disponibilidade de canal. Nesta fase o dispositivo deve enviar informações necessárias para o registro. São necessárias informações sobre o dono do dispositivo como: endereço, telefone e e-mail.

O registro realizado não é o equivalente a autenticação do dispositivo. As informações são obtidas apenas para determinar pessoas responsáveis pelo uso do espectro. O dispositivo pode usar a base independente das informações providas no registro.

Essa etapa é classificada como opcional, dependendo da necessidade do órgão regulador. No trabalho desenvolvido essa parte não foi implementada.

Pedido

Depois de ter passado pela etapa de inicialização o dispositivo pode realizar um pedido por canais disponíveis. O registro do dispositivo pode estar contido nessa etapa.

Basicamente o dispositivo envia para a base sua localização e recebe como resposta os canais disponíveis. O dispositivo, então, indica qual canal ele vai utilizar.

Embora não em conformidade total com o protocolo, essa parte foi implementada pelo trabalho. As principais diferenças devem ser destacadas.

No trabalho desenvolvido a localização recebida pela base é uma tupla com a latitude e longitude do dispositivo. O PAWS admite que a localização possa ser uma figura geométrica.

A resposta enviada pela base também é diferente. No protocolo proposto o canal disponível é enviado com um intervalo de tempo para o qual ele é válido. Caso esse tempo seja ultrapassado o dispositivo deve realizar um novo pedido para a base.

Validação

Um DM pode fazer pedidos para diversos DEs. A FCC exige que os DEs se registrem com a base. O DM realiza esse registro nessa etapa.

Essa etapa é classificada como opcional, dependendo da necessidade do órgão regulador. No trabalho desenvolvido essa parte não foi implementada.

Capítulo 3

Base de Dados Proposta

3.1 Informações Gerais

O sistema foi desenvolvido na linguagem de programação orientada a objeto Java, usando o Sistema de Gerenciamento de Banco de Dados (SGBD) MYSQL como banco de dados. A API foi desenvolvida no protocolo REST, utilizando a biblioteca restlet [16], e utilizando XML como formato de resposta. O desenvolvimento foi feito no ambiente Eclipse [17], e os diagramas de classe aqui exibidos foram gerados utilizando o plug-in ObjectAid.

A escolha pela linguagem de desenvolvimento Java ocorreu devido a maior familiaridade com a linguagem e as capacidades multiplataforma dela. O sistema foi desenvolvido no sistema operacional Windows 7, mas foi devidamente testado em um ambiente Linux utilizando uma máquina virtual. A ferramenta de virtualização utilizada foi a Oracle VM BOX [18].

Pode-se descrever o programa desenvolvido como sendo a etapa de pedido de disponibilidade do PAWS. Pode-se descrever o funcionamento em linhas gerais em quatro módulos. Como indicado na figura 3.1.

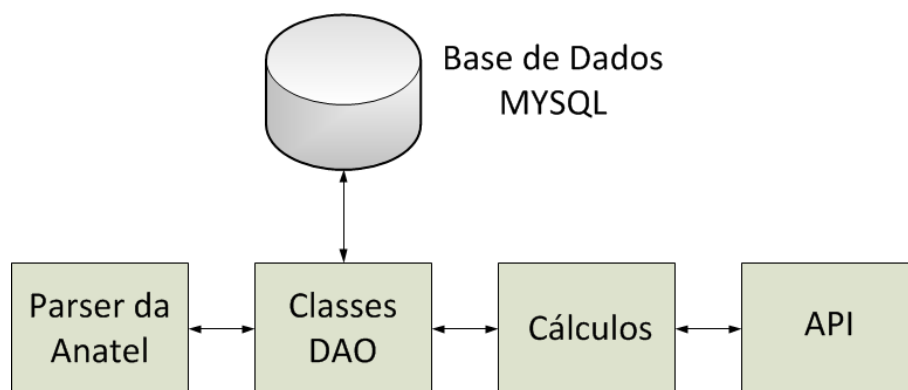


Figura 3.1: Diagrama geral do sistema desenvolvido.

Cada um desses módulos será descrito no decorrer do capítulo.

3.1.1 Desenvolvimento

Deve ser ressaltado que o objetivo do trabalho foi desenvolver um sistema que ofereça a funcionalidade básica de prover os canais disponíveis do WS mas que seja facilmente modificado para a adição de outras funções. Para facilitar o reuso do trabalho foi adotado o padrão de desenvolvimento de Fábrica [19]. Resumidamente, as interfaces que cada objeto deve estender são definidas, e existem objetos fábricas, responsáveis por criar instâncias de objetos que estendam essa interface. Dessa maneira, nenhum objeto é referenciado diretamente.

Tome como exemplo as classes onde foram implementados os modelos de propagação. Existe a interface *PropagationModel*, que define as características básicas de um modelo de propagação. Ela possui o método abstrato *maxDistance* que serve para determinar o alcance máximo de cada dispositivo de transmissão. Cada classe que implemente um modelo de propagação estende essa interface e implementa esse método. Durante o programa quando é necessário utilizar algum modelo de propagação nunca é criado um objeto diretamente das classes, mas é invocada a classe *PropagationModelFactory* que retorna um objeto que estenda a interface *PropagationModel*.

Um diagrama com as classes factories está na figura 3.2.

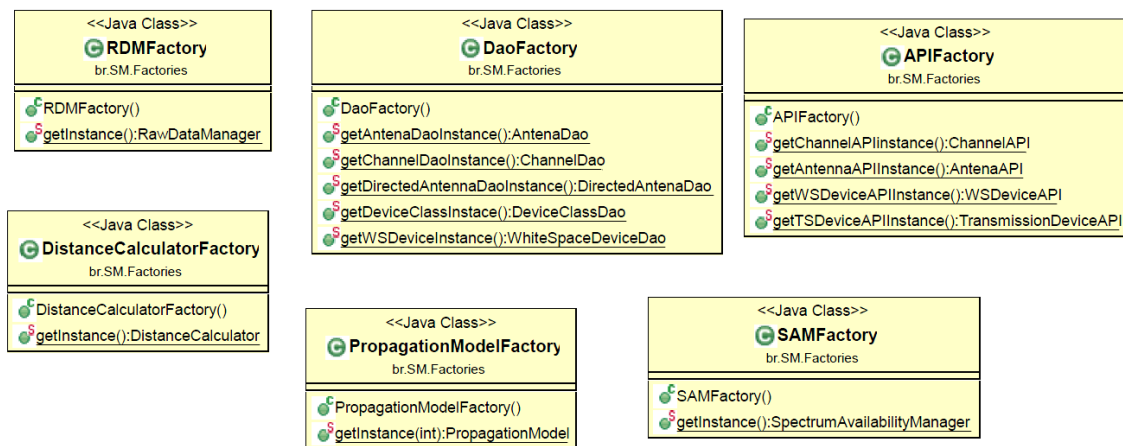


Figura 3.2: Diagrama de classe das - *Factories*

Dessa maneira o conteúdo das classes é abstraído. Um desenvolvedor deve apenas se preocupar em obedecer as interfaces. A interface define o modo como possíveis classes serão utilizadas, definindo todos os métodos acessáveis por outros objetos.

3.1.2 Definições

Para calcular a disponibilidade de canais no WS foi utilizada uma aproximação bem direta.

Com as informações sobre as antenas existentes no Estado do Rio de Janeiro é possível calcular o alcance máximo de cada antena. Por alcance máximo se entende: “A distância máxima da base da antena que um dispositivo pode estar para receber o sinal transmitido por ela com uma relação sinal/ruído mínima. Assume-se que o dispositivo não sofre nenhuma interferência além de duas vezes ruído de fundo”. Todos os dispositivos dentro do alcance da antena vão receber o sinal com qualidade suficiente. Para dispositivos fora desse alcance, a qualidade do sinal recebido será muito baixa, de modo que USs não estariam em condição de interferência com a antena.

Assumindo um US pode-se calcular seu alcance de forma similar ao da antena. A diferença é que o alcance de um US é igual à distância necessária para que sua potência seja atenuada até o valor do ruído de fundo.

A interferência nos receptores de A de duas vezes o ruído de fundo é devido ao ruído normal do ambiente somado à potência do sinal irradiado por um US que já sofreu atenuação.

Assumindo uma antena A com alcance d_1 e um US B com alcance d_2 . Sendo D a distância entre A e B. Define-se que o dispositivo B interfere os receptores da antena A se e somente se:

$$d_1 + d_2 \leq D \quad (3.1)$$

Um exemplo de interferência está na figura 3.3.

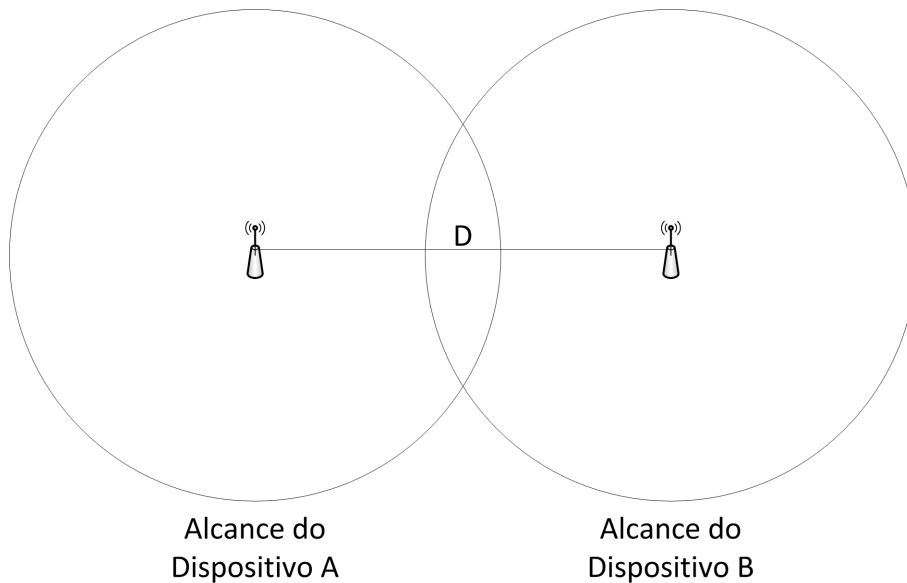


Figura 3.3: Exemplo de Interferência.

Se o dispositivo B interferir nos receptores da antena A, o canal que A utiliza

para transmissão se torna indisponível para B. Os canais que não possuam antenas em interferência com o dispositivo B estão disponíveis para uso.

O valor de D é calculado pela distância entre a antena e o dispositivo no globo terrestre. Já o cálculo de d_1 e d_2 será explicado na seção “Determinação do White Space”

3.2 Coleta de dados

Antes de entrar em operação, o sistema deve construir sua base de dados com as informações necessárias para os cálculos. Como foi indicado, é importante a existência de informações sobre todas as antenas do Rio de Janeiro. A falta de informação culmina na inviabilidade do projeto.

As informações foram acessadas no site da ANATEL [20]. Os dados foram obtidos de forma manual, não existe um Web Service, ou alguma forma de comunicação direta com a agência reguladora brasileira.

Foi baixado um arquivo com informações sobre todas as antenas de televisão do Brasil. Esse arquivo está no formato TSV (Tab Separated Values) onde cada linha contém as informações de uma antena conforme abaixo:

- Estado
- Cidade
- Dono
- Tipo
- Canal
- Complemento do Canal
- Campo Vazio
- Latitude
- Longitude
- Potência
- Informações de Antenas Direcionais

Foram utilizadas apenas as linhas onde o campo *Estado* era "RJ".

O campo *Canal* e *Complemento do Canal* foram utilizados para identificar o canal onde determinada antena opera.

Os campos *Latitude* e *Longitude* estão no formato grau-minuto-segundo, com uma letra indicando a direção da bússola. Esses valores foram convertidos para o formato decimal, que é considerado um formato mais padronizado, e por facilitar as contas que serão feitas. Esses campos identificam a posição da antena no planeta Terra.

O campo *Potência* está em kW, sendo utilizado para determinar a potência de transmissão da antena.

Para esse trabalho as antenas foram consideradas como sendo multidirecionais, ou seja, irradiam com a mesma potência em todas as direções. No entanto, existem antenas direcionais, que irradiam com uma determinada potência apenas em um cone direcionado. As informações específicas dessas antenas direcionais estão nos campos *Informações de Antenas Direcionais*. Essas informações constituem de um trio de campos para cada antena direcionada existente nessa posição. Todas as antenas direcionadas estão operando no mesmo canal, apenas com potências e direcionamentos diferentes.

As seguintes informações são indicadas para antenas direcionadas:

- Altura
- Direcionamento (Azimuth)
- Potência

Esses três campos também são utilizados pelo sistema.

3.2.1 Implementação

Existe um parser responsável por converter o arquivo extraído da ANATEL em uma lista de objetos *Antena*. As informações sobre antenas direcionadas são incluídas em uma lista de objetos *DirectedAntena*, que é associada a *Antena* que ela pertence.

O diagrama na figura 3.4 especifica esse exemplo.

O parser pode, também, buscar a altura geográfica de onde as antenas se encontram. Isso é feito usando Google Elevation API [21]. Essa informação só é utilizada pelo sistema caso não exista nenhuma informação sobre altura no arquivo da ANATEL.

O diagrama de classes para esse módulo está na figura 3.5.

3.2.2 Extensão

A obtenção de dados é abstraído para a interface *RawDataManager*. Assim sendo, caso seja determinada uma nova maneira para comunicação com a ANATEL, basta que um método herde a interface antes mencionada.

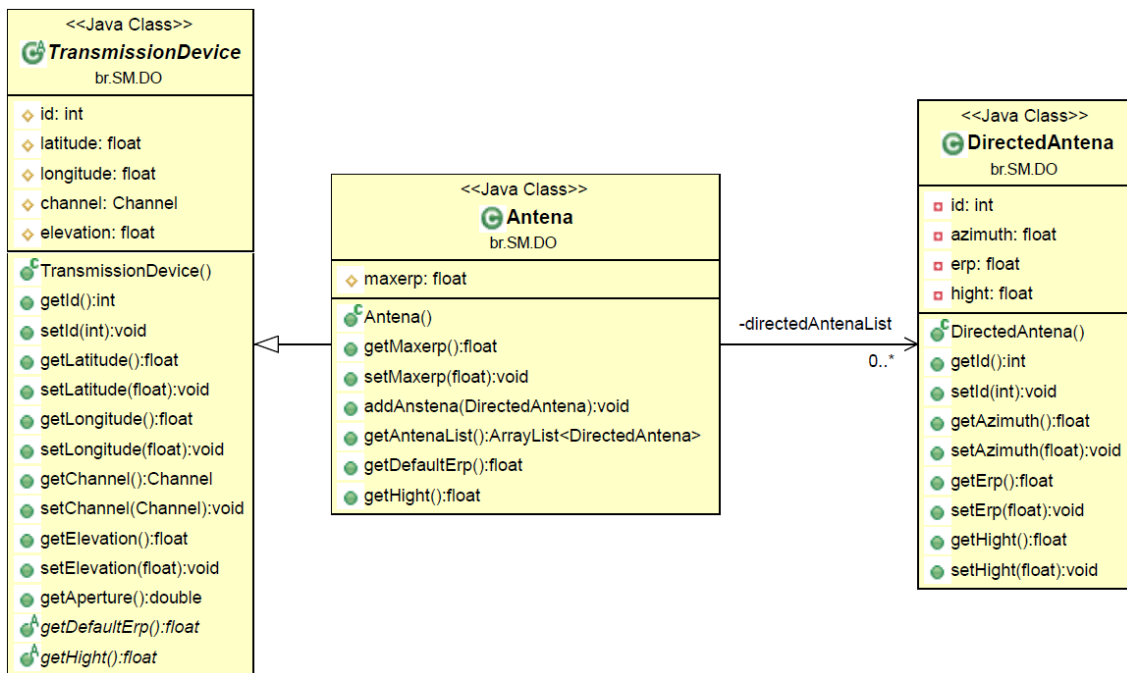


Figura 3.4: Diagrama de classes das antenas.

Na figura 3.6 está indicado como ocorre o parsing do arquivo da ANATEL sob um ponto de vista de classes utilizadas.

3.2.3 Qualidade da Informação

Como já foi indicado, garantir a qualidade da informação que é obtida da ANATEL é essencial para a BDWS criada. No entanto, foram encontradas algumas informações erradas no arquivo utilizado como fonte.

Para testar a qualidade da informação utilizada foi gerado um arquivo KML [22] com a localização das antenas. Foi utilizado o Google Earth para verificar visualmente a posição das antenas [23].

Como se pode ver na figura 3.7, algumas das antenas que estavam indicadas como localizadas no Rio de Janeiro pertenciam, na verdade, a São Paulo.

Outro ponto identificado são valores, a princípio, estranhos. Alguns valores de potência estão muito maiores que o esperado para uma antena.

É preciso firmar com a ANATEL uma relação de cooperação para aprimorar a base. Isso será discutido mais a fundo no capítulo 6 na seção de trabalhos futuros.

Vale ressaltar que o formato adotado pela ANATEL para armazenar latitude e longitude é muito impreciso. A agência só armazena o grau, minuto e segundo de cada coordenada geográfica, faltando informações sobre décimos e centésimos de segundo. Essa falta de precisão na localização das antenas deve ser corrigida a fim de prover informações o mais corretas possível.

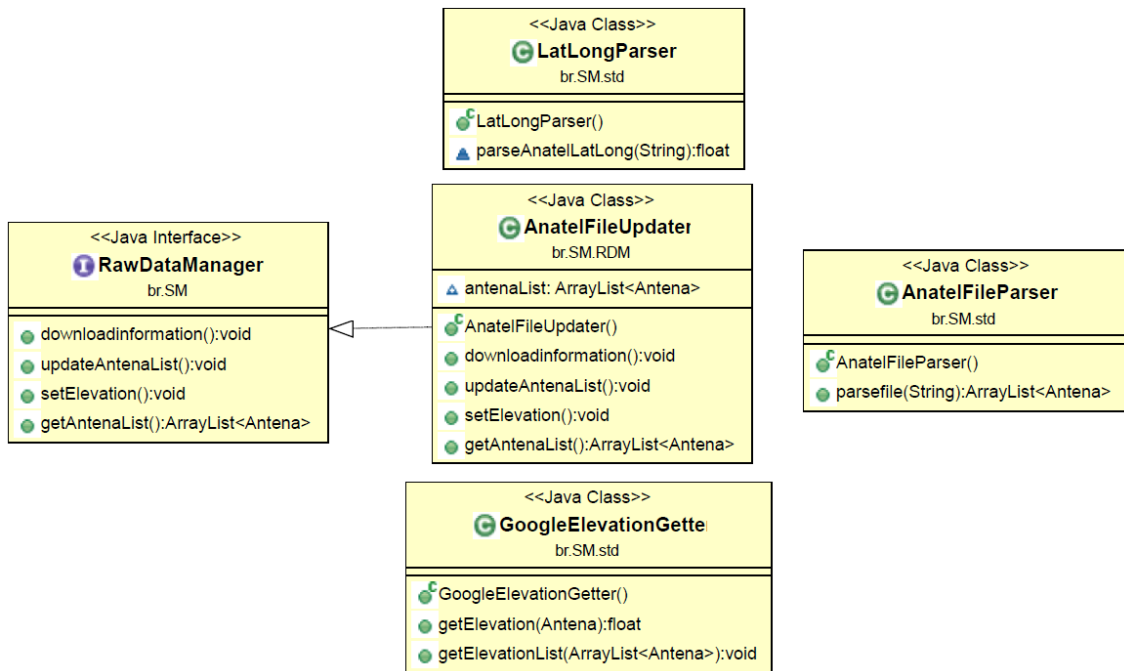


Figura 3.5: Diagrama de classes do módulo *Parser*.

3.3 Base de Dados

Além das informações obtidas da ANATEL a base de dados armazena, também, informações sobre reservas de canais feitas pelos DSs. Na modelagem de dados feita foi criada a classe *TransmissionDevice*, da qual foram estendidas as classes *Antena* e *WhiteSpaceDevice*. A diferença básica entre esses dois objetos é que a *Antena* possui uma potência de transmissão informada pela ANATEL e uma lista de *DirectedAntena* a qual está associada. Já o *WhiteSpaceDevice* está associado a uma classe de dispositivo (*DeviceClass*) que tem uma potência predeterminada.

Foram criadas três classes de dispositivos:

- "INTERNO BAIXO" com potência de 10 miliwatts
- "INTERNO ALTO" com potência de 50 miliwatts
- "EXTERNO" com potência de 100 miliwatts

As classes do tipo "INTERNO" representam dispositivos para uso interno, ou seja, dentro de alguma construção. Já a classe do tipo "EXTERNO" representa dispositivos para uso externo.

Classes criadas para representar antenas e DS, também chamadas de classes de dados, ou Data Objects estão representadas na figura 3.8.

O modelo de dados criado no SGBD está indicado no diagrama entidade relacionamento 3.9.

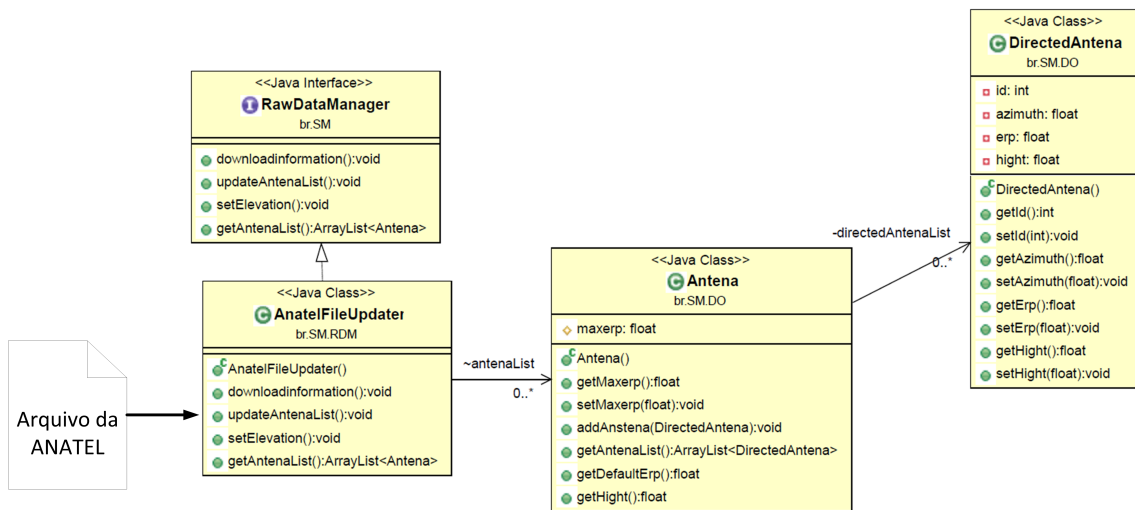


Figura 3.6: Diagrama com o processo de Parsing.

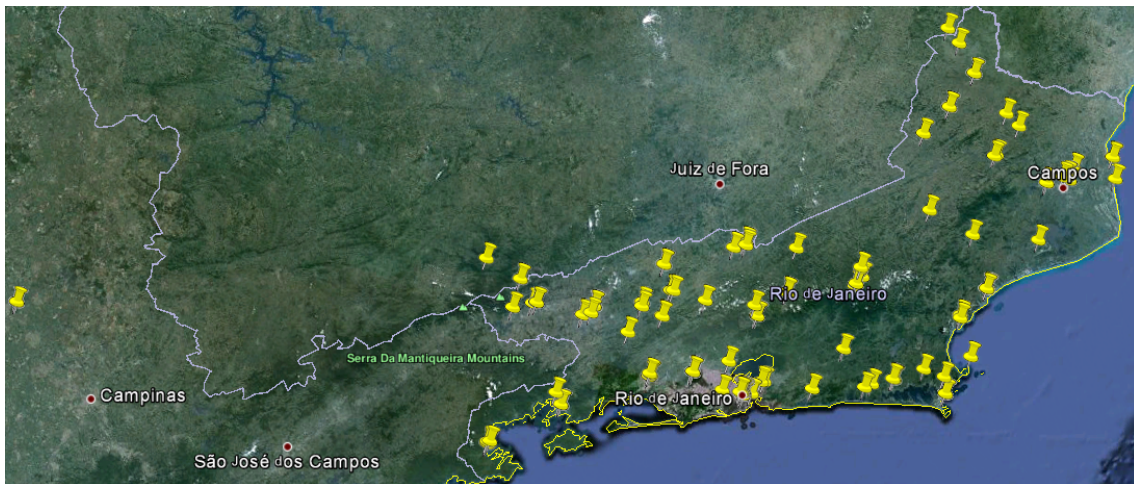


Figura 3.7: Visualização de antenas indicadas pela ANATEL.

3.3.1 Data Access Object

O acesso à base de dados foi desenvolvido usando o padrão de desenvolvimento Data Access Object(DAO). Nesse padrão todo acesso ao banco de dados é abstraído para as classes DAO. Elas possuem métodos que retornam objetos de dados (DO) de acordo com parâmetros passados.

Foram desenvolvidas interfaces DAO que possuem todos os métodos necessários para interagir com uma determinada tabela. Existe, também, a classe *BasicMySQLDao* que é responsável por gerenciar a conexão com a base de dados criada. Uma classe DAO, como a *AntenaMySQLDao*, deve herdar a interface correspondente (*AntenaDao*) e estender a *BasicMySQLDao*. Esse exemplo é retratado na figura 3.10.

3.3.2 Implementação

Além das classes DAO, foi desenvolvido uma interface cujo único objetivo é acionar a classe que herde o *RawDataManager*. A ideia é segregar funções.

As classes mencionadas na seção anterior têm apenas uma preocupação, gerar uma lista com as informações que devem ser incluídas na base. Já a interface *SpectrumAvailabilityManager* recebe essa lista e aciona as classes DAO necessárias, de modo a incluir essas informações na base de dados.

A figura 3.11 representa as classes desse módulo.

3.3.3 Extensão

Para modificar a base de dados utilizada, bem como realizar mudanças no modelo de dados, como modificar de nomes de campos ou da base, basta alterar as classes DAO do MYSQL. Devem ser construídas classes DAO para o SGBD que for utilizado. As classes criadas para o MYSQL podem ser tomadas como exemplo.

As classes DAO implementadas estão representadas na figura 3.12.

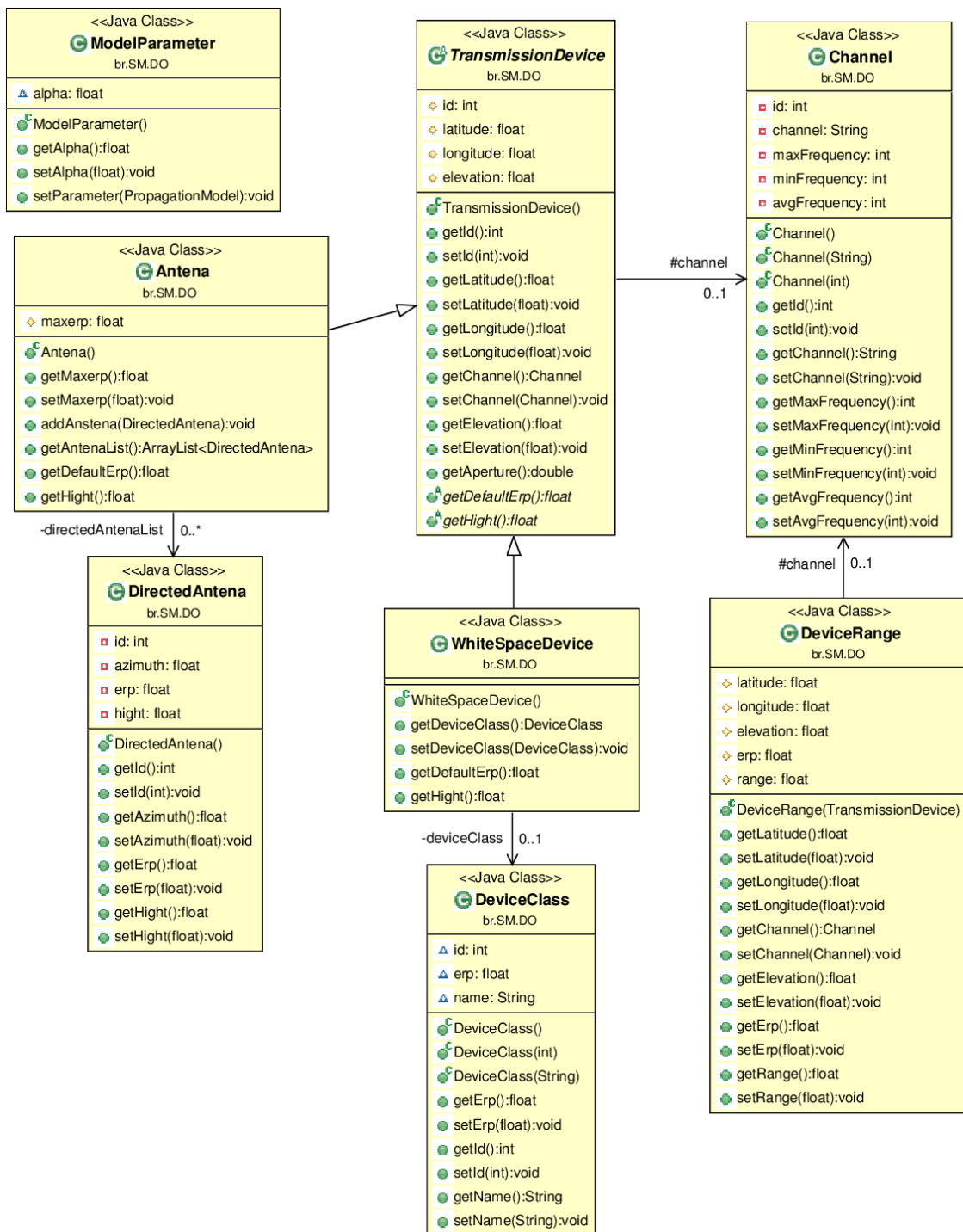


Figura 3.8: Classes de dados.

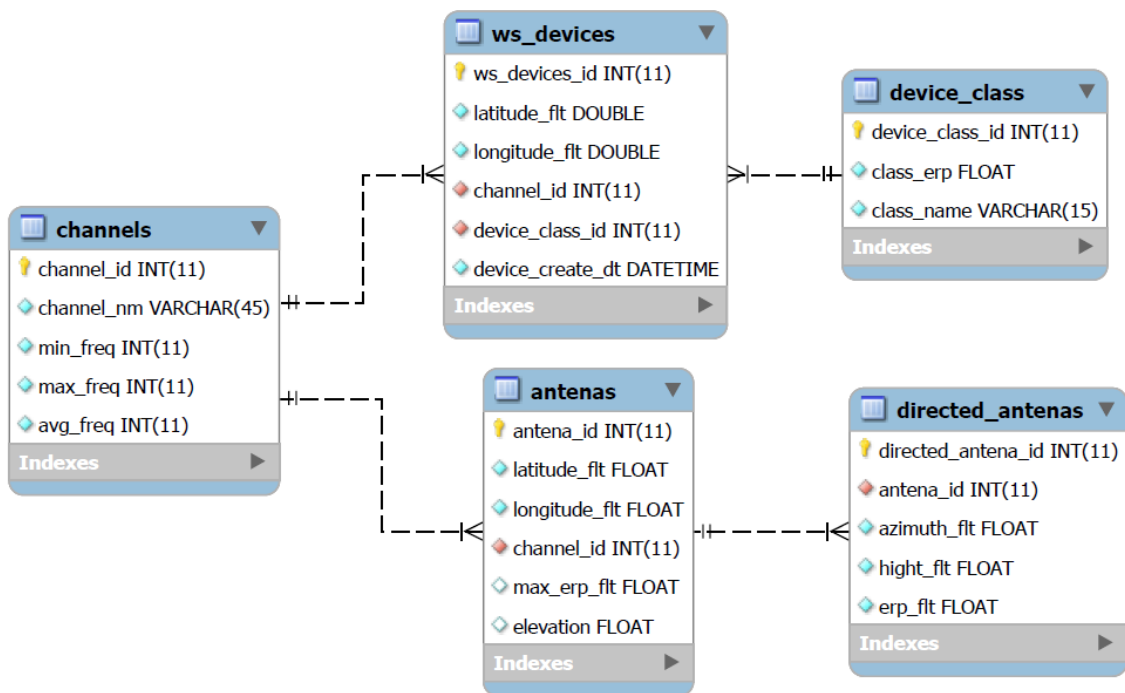


Figura 3.9: Diagrama entidade relacionamento da Base de Dados.

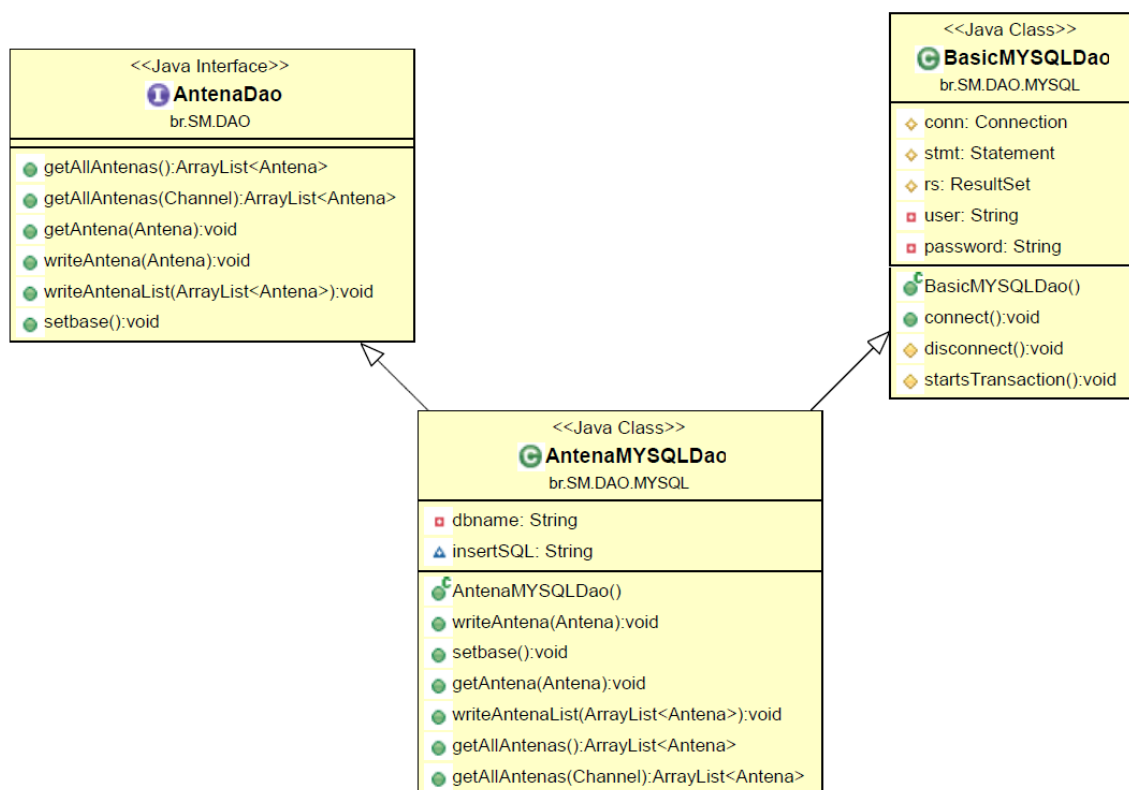


Figura 3.10: Classe AntenaMySQLDao.

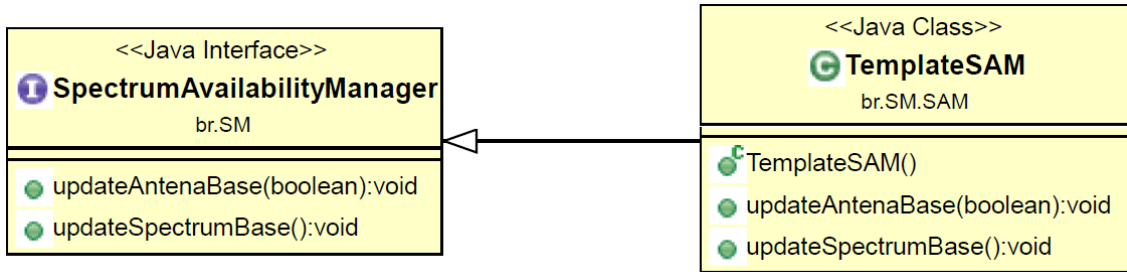


Figura 3.11: Diagramas de classe para montar a base de dados.

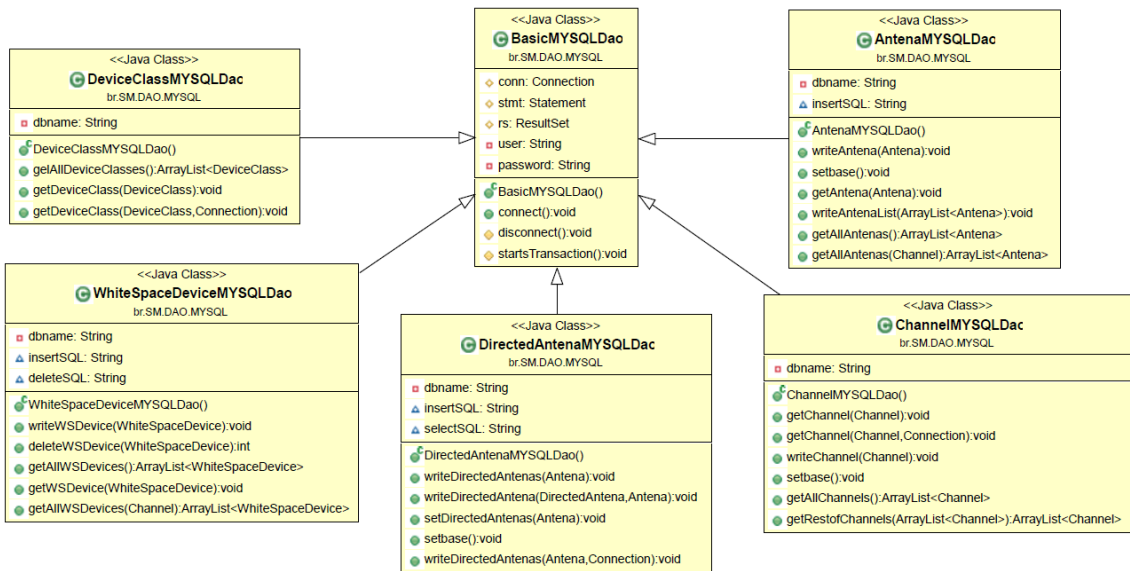


Figura 3.12: Classe DAO do MYSQL.

A classe *TemplateSAM*, a princípio, já realiza todas as funções necessárias para montar a base de dados com as informações obtidas da ANATEL e não precisa ser alterada.

3.3.4 Canais de transmissão

As informações referentes aos canais de transmissão utilizados pelas antenas também foram retiradas do site da ANATEL [24], e incluídas manualmente na base. As informações utilizadas estão na tabela 3.1.

Tabela 3.1: Tabela obtida do site da ANATEL [24]

Canal	Frequência Mínima	Frequência Média	Frequência Máxima
2	54	57	60
3	60	63	66
4	66	69	72
5	76	79	82
6	82	85	88
7	174	177	180
8	180	183	186
9	186	189	192
10	192	195	198
11	198	201	204
12	204	207	210
13	210	213	216
14	470	473	476
15	476	479	482
16	482	485	488
17	488	491	494
18	494	497	500
19	500	503	506
20	506	509	512
21	512	515	518
22	518	521	524
23	524	527	530
24	530	533	536
25	536	539	542
26	542	545	548
27	548	551	554
28	554	557	560

Tabela 3.1 – continuação

Canal	Frequência Mínima	Frequência Média	Frequência Máxima
29	560	563	566
30	566	569	572
31	572	575	578
32	578	581	584
33	584	587	590
34	590	593	596
35	596	599	602
36	602	605	608
38	614	617	620
39	620	623	626
40	626	629	632
41	632	635	638
42	638	641	644
43	644	647	650
44	650	653	656
45	656	659	662
46	662	665	668
47	668	671	674
48	674	677	680
49	680	683	686
50	686	689	692
51	692	695	698
52	698	701	704
53	704	707	710
54	710	713	716
55	716	719	722
56	722	725	728
57	728	731	734
58	734	737	740
59	740	743	746
60	746	749	752
61	752	755	758
62	758	761	764
63	764	767	770
64	770	773	776
65	776	779	782
66	782	785	788

Tabela 3.1 – continuação

Canal	Frequência Mínima	Frequência Média	Frequência Máxima
67	788	791	794
68	794	797	800
69	800	803	806

Para os cálculos descritos na seção seguinte, foi utilizado o valor de frequência média.

3.4 Determinação do White Space

A disponibilidade de canais é determinada aplicando modelos de propagação nas antenas contidas na base gerada a partir das informações da ANATEL.

3.4.1 Algoritmo

Assume-se um dispositivo x cujas latitude lat e longitude lng determinam sua localização no planeta Terra. Todas as antenas na base devem ser percorridas, de forma a determinar em quais delas o dispositivo não interfere, como definido na equação 3.1. Os seguintes modelos de propagação foram implementados, para determinar o alcance da antena e de um possível dispositivo:

- Free Space
- Two Ray Ground
- Log Distance

A definição e formulação de cada um deles, mencionadas a seguir, foi retirada de RAPPAPORT [25].

Como o sistema armazena informação de reserva de canais, além de verificar se x interfere com as antenas da base, deve ser verificado, também, a condição de interferência com outros dispositivos secundários.

Definições Gerais

O objetivo dos modelos implementados é determinar a qual distância da antena seu sinal se atenua a ponto que a relação sinal/ruído se torna igual a 10dB. Além disso, deve ser determinada a distância necessária para que a potência emitida por um possível US se atenua até chegar ao valor do ruído de fundo.

A relação sinal/ruído é definida:

$$SINR = \frac{P}{I + N} \quad (3.2)$$

Onde P é a potência do sinal, I é a interferência e N é o ruído de fundo.

Como é assumido que a única interferência no ambiente será o ruído de fundo, tem-se então que:

$$SINR = \frac{P}{N} \quad (3.3)$$

O ruído de fundo, em Watts, é definido como:

$$Noise = kTB \quad (3.4)$$

Aonde k é a constante de Boltzman, T é a temperatura do ambiente e B é a banda em hertz.

Assim, pelas equações 3.3 e 3.4 tem-se que a potência mínima do receptor é:

$$P_r = SINR \cdot kTB \quad (3.5)$$

O valor da relação sinal/ruído deve ser convertido para W de forma a facilitar os cálculos. Dessa forma:

$$10 = 10 \log_{10}(SINR_W) \quad (3.6)$$

De modo que a potência do sinal emitido pela antena em seu alcance máximo será:

$$P_a = 10kTB \quad (3.7)$$

No entanto, o dispositivo que utiliza o WS não pode interferir com os receptores da antena. Dessa forma, seu sinal deve ser atenuado até atingir o valor do ruído de fundo. Assim sendo, a relação sinal/ruído deve ser de 0dB. De modo que podemos determinar a potência no alcance máximo do dispositivo como sendo:

$$P_d = kTB \quad (3.8)$$

Free Space

O modelo de propagação Free Space admite que o transmissor e o receptor possuem um caminho em linha reta, sem obstruções entre si. Esse modelo é bem simples e sua aplicação resulta em alcances muito superiores aos que eles realmente são. Sua implementação serviu como base para os outros modelos. Ele foi desenhado na figura 3.13.

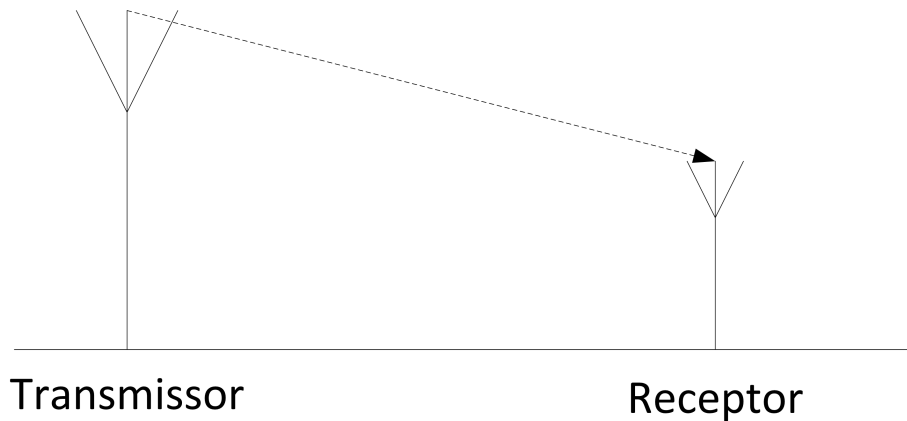


Figura 3.13: Modelo de propagação Free Space.

A potência recebida pelo receptor quando o transmissor se encontra a uma distância d é definida [25] como:

$$P_r(d) = \frac{P_t G_t G_r \lambda^2}{(4\pi)^2 d^2 L} \quad (3.9)$$

G_t e G_r são os ganhos do transmissor e receptor respectivamente. Foi assumido que a potência indicada pela ANATEL é, na verdade, $P_t G_t$, enquanto G_r foi desconsiderado.

$$G_r = 1 \quad (3.10)$$

L representa o fator de perda não relacionado ao modelo de propagação. L também foi desconsiderado.

$$L = 1 \quad (3.11)$$

λ é referente a frequência utilizada para transmissão, onde c é a velocidade da luz em m/s e f é a frequência utilizada em Hz.

$$\lambda = \frac{c}{f} \quad (3.12)$$

Assim sendo, determina-se o alcance de uma antena no modelo de Free Space pelas equações 3.9 e 3.7:

$$d_1 = \sqrt{\frac{P_t \lambda^2}{10kTBL(4\pi)^2}} \quad (3.13)$$

De maneira análoga, o alcance de um US pode ser determinado pelas equações 3.9 e 3.8:

$$d_2 = \sqrt{\frac{P_t \lambda^2}{kTBL(4\pi)^2}} \quad (3.14)$$

Two Ray Ground

O modelo Two Ray Ground, ou de Reflexão no Chão, assume a possibilidade de reflexão do sinal emitido. O sinal enviado pelo transmissor, além de alcançar o receptor por uma linha direta como no Free Space, vai refletir no chão e chegar ao receptor. Dessa forma, dois raios chegam ao receptor: o refletido e o não refletido. Como indicado na figura 3.14.

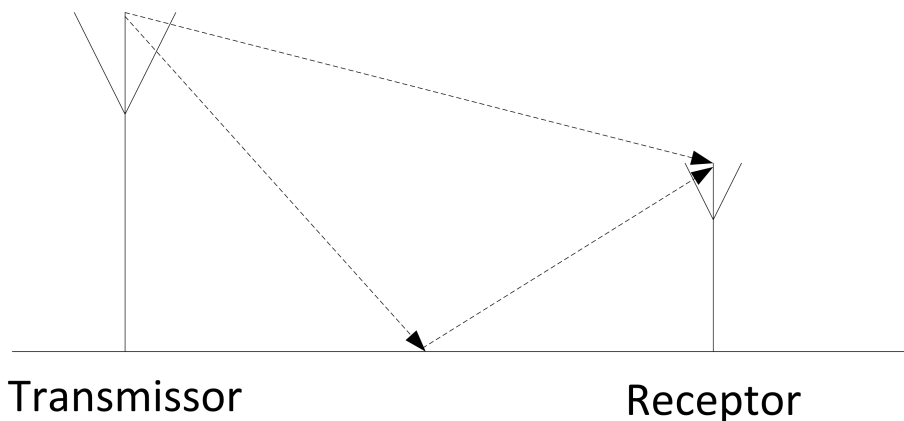


Figura 3.14: Modelo de propagação Two Ray Ground.

A potência recebida pelo receptor quando o transmissor se encontra a uma distância d é definida [25] como:

$$P_r(d) = P_t G_t G_r \frac{h_t^2 h_r^2}{d^4} \quad (3.15)$$

Onde h_t é a altura da antena transmissora e h_r é a altura da antena do receptor em metros. Foi assumida uma altura de 2 metros para casos onde não foi possível determinar a altura do dispositivo.

Assim, assumindo a equação 3.10, temos, pelas equações 3.15 e 3.7 o alcance de uma antena no modelo Two Ray Ground:

$$d_1 = \sqrt[4]{\frac{P_t h_t^2 h_r^2}{10kTB}} \quad (3.16)$$

De maneira análoga, o alcance de um US pode ser determinado pelas equações 3.9 e 3.8:

$$d_2 = \sqrt[4]{\frac{P_t h_t^2 h_r^2}{kTB}} \quad (3.17)$$

Log Distance

O modelo de propagação Log Distance assume que a potência transmitida por uma antena decai logarithmicamente em relação à distância. Dois fatores principais entram em cena, a perda ocorrida a uma distância muito pequena, e a taxa de decaimento.

L_0 determina a perda ocorrida a uma distância muito pequena. Foi considerado a distância d_0 como sendo 1 metro. Essa perda foi calculada no modelo Free Space:

$$L_0 = \left(\frac{4\pi w}{c} \right)^2 \quad (3.18)$$

Onde w é o comprimento de onda da frequência utilizada e c é a velocidade da luz em metros.

O modelo foi simplificado de forma que o alcance de uma antena é representado por:

$$d_1 = \sqrt[4]{\frac{d_0^\gamma 10^{-\frac{L_0}{10}} P_t}{10kTB}} \quad (3.19)$$

De maneira análoga, o alcance de um US é representado por:

$$d_2 = \sqrt{\frac{\gamma d_0^2 10^{\frac{-L_0}{10}} P_t}{kTB}} \quad (3.20)$$

γ representa um fator de perda que geralmente recebe valores de 2 a 6. Foi escolhido 3 como sendo valor padrão.

Constantes

As seguintes constantes foram assumidas para os cálculos descritos:

- $k = 1380 \cdot 10^{-23}$ mW
- $T = 290$ K
- $B = 6$ MHz
- $c = 299792458$ m/s

3.4.2 Implementação

Foi desenvolvida a classe abstrata *PropagationModel* que define as constantes e alguns métodos básicos para todos modelos de propagação. Foram geradas classes para cada um dos modelos de propagação descritos a cima. Eles implementam o método *maxDistance* que indica o alcance de um determinado dispositivo. A figura 3.15 representa as classes criadas.

A classe *PropagationModel* possui o método *interferes* que indica se 2 dispositivos estão em condição de interferência.

O modelo de propagação utilizado é indicado ao chamar a sua classe *Factory PropagationModelFactory*.

A classe *ModelParameter* possui uma lista com todos os parâmetros utilizados pelos modelos. O Construtor dessa classe atribui a esses parâmetros seus valores padrões. Sempre que for necessário indicar um parâmetro a um modelo isto é feito utilizando essa classe.

3.4.3 Extensão

Para criar novos modelos de propagação basta criar classes que estendam a *PropagationModel* e incluir seus possíveis parâmetros na classe *ModelParameter*.

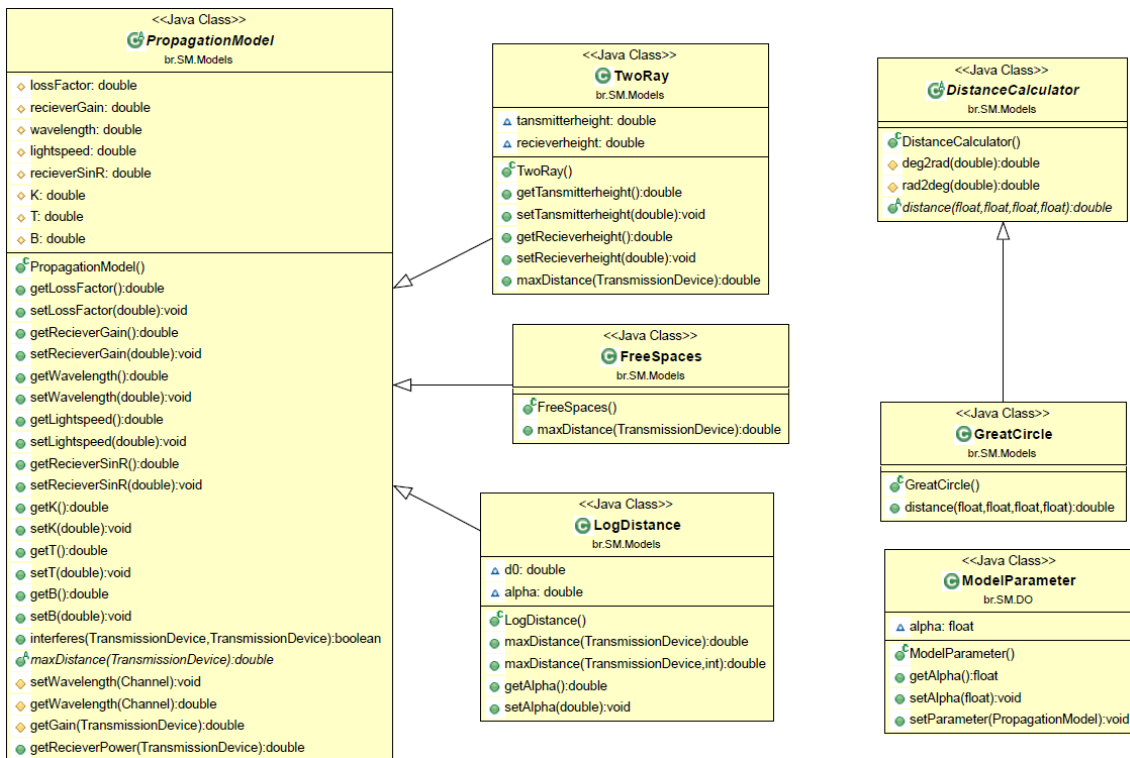


Figura 3.15: Diagrama de classes dos modelos de propagação desenvolvidos.

3.5 Web Service Desenvolvido

Web Services são uma maneira de disponibilizar um serviço cliente servidor em uma rede. Uma mensagem é enviada do cliente ao servidor, que executa o serviço e envia uma mensagem de resposta ao cliente. A interface definida pelo servidor deve ser indicada ao cliente, bem como o formato de resposta utilizado.

O Web Service desenvolvido tem dois objetivos. O primeiro é de prestar o serviço de disponibilidade de canais no WS, como descrito no PAWS. O segundo é de tornar disponível todas as informações contidas na base desenvolvida. Foi criado, então, um serviço para cada tipo de item disponível:

- Canal
- Antena
- Dispositivo Secundário

Além disso, é possível determinar o modelo de propagação que será utilizado pelo servidor para realizar os devidos cálculos.

3.5.1 REST

O Web Service foi desenvolvido no modelo REST [26]. O seu conceito básico é que a base consiste de recursos, cada um com um Uniform Resource Identifier (URI)

específico. Os URIs consistem de URLs que são alcançáveis por pedidos HTTP. Pode-se, então, interagir com toda base enviando pedidos HTTP.

Existem quatro tipos de pedido HTTP que podem ser utilizados:

- GET - utilizado para buscar um recurso
- POST - utilizado para alterar o valor de um recurso
- PUT - utilizado para alterar o valor de um recurso
- DELETE - utilizado para excluir um recurso

O POST e o PUT possuem basicamente a mesma função. A especificação do HTTP [27] define que o PUT deve ser utilizado quando um novo recurso é criado na base, enquanto o POST modifica um recurso já existente. No entanto, o PUT também pode ser usado para modificar um recurso já existente. Dessa forma, apenas o PUT foi utilizado no sistema desenvolvido.

Praticamente não existe uma diferença entre um Web Service REST e outro em SOAP, por exemplo. Os dois podem prover os mesmos serviços. No entanto, outros Web Services geralmente precisam preparar a mensagem que será enviada ao servidor. Já no REST as mensagens são simples pedidos de HTTP, com os parâmetros estando, muitas das vezes, nos próprios parâmetros da URL.

Não existe uma forma padrão para a resposta do servidor ao cliente. Entretanto, isso muitas vezes é feito via JSON ou XML, já que estes formatos apresentam uma forma mais estruturada. No servidor desenvolvido o formato de resposta escolhido foi o XML.

Pode-se resumir o funcionamento do servidor REST desenvolvido da seguinte maneira:

- O cliente faz um pedido HTTP em uma URL do servidor
- O servidor identifica qual recurso essa URL serve
- O servidor processa o pedido e retorna o recurso
- O cliente recebe a resposta do servidor no formato XML

A escolha pelo REST, em detrimento do JSON-RPC se deu pela sua maior facilidade de testes. Como o serviço REST é alcançado por um pedido HTTP, pode-se usar qualquer browser da internet para verificá-lo.

3.5.2 Implementação

Os serviços prestados pela base foram indicados nas interfaces API. As classes que herdam essa API implementam os métodos que serão utilizados remotamente. Essas classes serão descritas na seção API.

Foi utilizada a biblioteca Restlet [16] para a criação do Web Service. Aqui foram definidas outras classes que apenas invocam as classes mencionadas acima. Essa segregação foi feita para poder dissociar o serviço prestado do tipo de Web Service criado.

Dessa forma, caso seja preciso modificar o Web Service de REST para JSON-RPC, isso pode ser feito sem que seja necessário redefinir os métodos da API. Apenas as classes de interface com o cliente vão precisar ser modificadas.

A API definida pode ser separada em duas finalidades. Existe a API padrão, que realiza todos os cálculos usando o modelo de propagação Log Distance, e uma API para testes, onde o modelo a ser utilizado pode ser escolhido, bem como suas constantes alteradas. A princípio um dispositivo só precisaria utilizar a API padrão. A API de testes é utilizada apenas para realizar testes com a base, já que dois dispositivos que utilizem modelos de propagação diferentes para seus cálculos acabam recebendo informações desconexas.

3.5.3 Extensão

O tipo de Web Service utilizado pode ser facilmente alterado, basta que as classes referentes ao Web Service invoquem as classes API.

Novas funcionalidades também podem ser desenvolvidas. Elas devem ser colocadas em novas Interfaces API ou modificar as interfaces existentes.

3.5.4 API

De forma a padronizar a resposta do serviço foi criada a classe *DeviceRange*, indicada na figura 3.16. Essa classe possui todas as informações básicas de um dispositivo de transmissão:

- Latitude do dispositivo – *DeviceRange.latitude*
- Longitude do dispositivo – *DeviceRange.longitude*
- Canal utilizado pelo dispositivo – *DeviceRange.channel*
- Elevação do dispositivo – *DeviceRange.elevation*
- Potência do dispositivo – *DeviceRange.erp*

- Alcance calculado para o dispositivo – *DeviceRange.range*

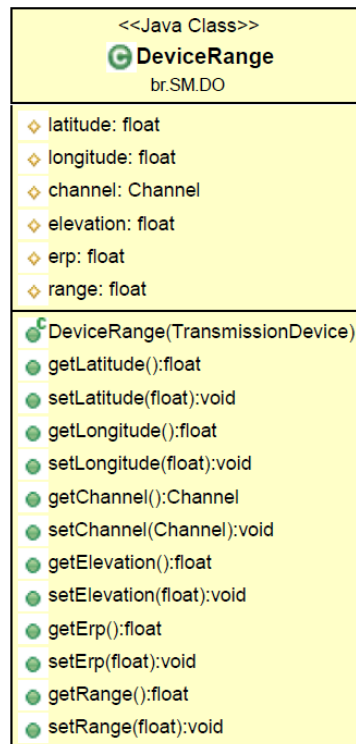


Figura 3.16: Diagrama da classe *DeviceRange*.

Outra classe utilizada nas respostas é a classe *Channel*, indicada na figura 3.17. Ela possui as informações básicas sobre um determinado canal de frequência:

- Número do canal – *Channel.channel*
- Frequência máxima do canal – *Channel.maxFrequency*
- Frequência mínima do canal – *Channel.minFrequency*
- Frequência média do canal – *Channel.avgFrequency*

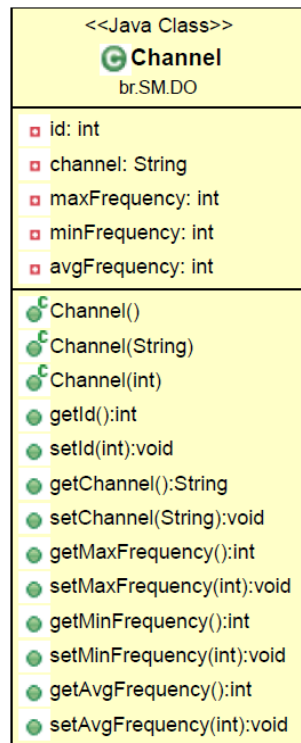


Figura 3.17: Diagrama da classe *Channel*.

Segue o formato do arquivo XML retornado

```

1 <list>
2   <br.SM.DO.DeviceRange>
3     <latitude>-23.0167</latitude>
4     <longitude>-44.2886</longitude>
5     <channel>
6       <id>24</id>
7       <channel>25</channel>
8       <maxFrequency>542</maxFrequency>
9       <minFrequency>536</minFrequency>
10      <avgFrequency>539</avgFrequency>
11    </channel>
12    <elevation>22.4531</elevation>
13    <erp>1250.0</erp>
14    <range>0.0</range>
15  </br.SM.DO.DeviceRange>
16 </list>
  
```

```

1 <list>
2   <br.SM.DO.Channel>
3     <id>1</id>
4     <channel>2</channel>
  
```

```
5 <maxFrequency>60</maxFrequency>
6 <minFrequency>54</minFrequency>
7 <avgFrequency>57</avgFrequency>
8 </br>.SM.DO.Channel>
9 </list>
```

Foram desenvolvidas quatro classes de API:

- ChannelAPI
- AntenaAPI
- WSDeviceAPI
- TransmissionDeviceAPI

Uso

Um resumo de utilização da API pode ser descrito como:

- <http://servidor/SAM/devices>
 - GET – Busca todos os dispositivos de transmissão presentes na base;
- <http://servidor/SAM/devices/location>
 - GET – Busca todos os dispositivos de transmissão presentes na base dada uma localização;
- <http://servidor/SAM/devices/channel>
 - GET – Busca todos os dispositivos de transmissão presentes na base dado um canal;
- <http://servidor/SAM/devices/antennas>
 - GET – Busca todas as antenas presentes na base;
- <http://servidor/SAM/devices/antennas/location>
 - GET – Busca todas as antenas presentes na base dada uma localização;
- <http://servidor/SAM/devices/antennas/channel>
 - GET – Busca todas as antenas presentes na base dado um canal;
- <http://servidor/SAM/devices/ws>
 - GET – Busca todos os dispositivos secundários presentes na base;

- PUT – Cadastra um dispositivo secundário na base;
- DELETE – Remove um dispositivo secundário na base;
- `http://servidor/SAM/devices/ws/location`
 - GET – Busca todos os dispositivos secundários na base dada uma localização;
- `http://servidor/SAM/devices/ws/channel`
 - GET – Busca todos os dispositivos secundários na base dado um canal;
- `http://servidor/SAM/devices/ws/classes`
 - GET – Busca todos os tipos de dispositivos secundários cadastrados na base;
- `http://servidor/SAM/channels`
 - GET – Busca todos os canais cadastrados na base;
- `http://servidor/SAM/channels/available`
 - GET – Busca os canais disponíveis para uso por um determinado dispositivo;

ChannelAPI

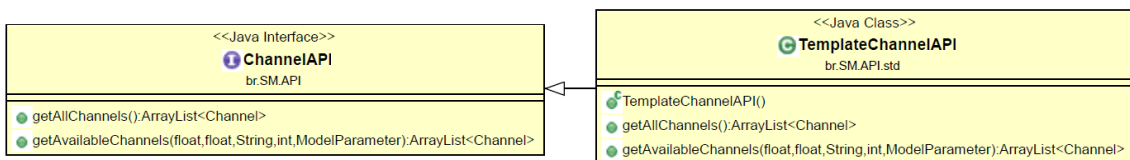


Figura 3.18: Diagrama da classe *ChannelAPI*.

A classe *ChannelAPI*, representada na figura 3.18, define dois métodos:

```
1 public ArrayList<Channel> getAllChannels ();
```

Este método retorna uma lista com todos os canais de frequência registrados na base.

Ele é alcançado pela URL `”http://servidor/SAM/channels”`, utilizando o método GET. Esse método não recebe nenhum parâmetro.

A resposta desse método é uma lista de objetos *Channel*, como indicado na seção anterior.

```
1 public ArrayList<Channel> getAvailableChannels(float lat, float  
   lng, String deviceType, int modeltype, ModelParameter par);
```

Este método retorna todos os canais disponíveis para uso em uma determinada posição. Deve ser indicado pelo cliente o tipo de dispositivo que está realizando a busca.

Ele é alcançado pela URL *"http://servidor/SAM/channels/available"*, utilizando o método GET. Pode ser utilizado, também a URL *"http://servidor/SAM/model/[modelo]/channels/available"* onde *[modelo]* indica o modelo de propagação que será utilizado para o cálculo dos alcances. Podem ser indicados três valores para modelo:

- 0 – Free Space
- 1 – Log Distance
- 2 – Two Ray Ground

Esse método recebe alguns parâmetros, que são submetidos pelo cliente como parâmetros de URL:

- *lat* – A latitude da posição pesquisada. É indicado pelo parâmetro de URL "lat".
- *lng* – A longitude da posição pesquisada. É indicado pelo parâmetro de URL "lng".
- *deviceType* – O tipo de dispositivo que deseja utilizar o WS. É indicado pelo parâmetro de URL "type".
- *modeltype* – O modelo de propagação a ser utilizado. É indicado no corpo da URL. Só utilizado pela API de testes. A API padrão utiliza automaticamente o modelo Log Distance.
- *par* – Constituído de parâmetros específicos para um determinado modelo. Só utilizado pela API de testes. Cada modelo sabe quais parâmetros deve buscar. Para o modelo Log Distance o valor de α pode ser indicado pelo parâmetro de URL "alpha".

Exemplo da URL utilizando a API padrão:

```
1 http://localhost:8182/SAM/channels/available?  
2   lat=-22.845882&lng=-43.08815&type=INTERNO+BAIXO
```

Exemplo da URL utilizando a API de teste:

```
1 http://localhost:8182/SAM/model/1/channels/available?  
2 lat=-22.845882&lng=-43.08815&type=INTERNO+BAIXO&alpha=2.5
```

A resposta desse método é uma lista de objetos *Channel*, como indicado na seção anterior.

AntenaAPI

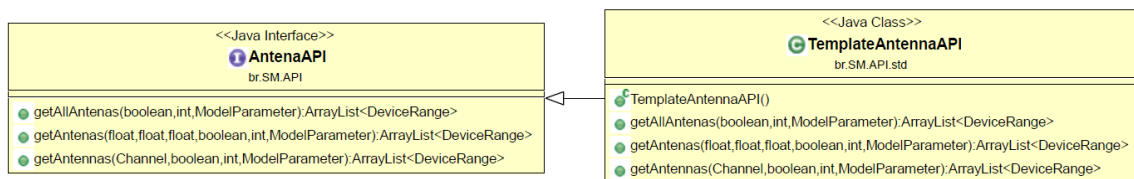


Figura 3.19: Diagrama da classe *AntenaAPI*.

A classe *AntenaAPI*, representada na figura 3.19, define três métodos:

```
1 public ArrayList<DeviceRange> getAllAntenas(boolean range, int  
2 modeltype,  
    ModelParameter par);
```

Este método retorna uma lista com todas as antenas presentes na base de dados, bem como o seu alcance.

Ele é alcançado pela URL `”http://servidor/SAM/devices/antennas”`, utilizando o método GET. Pode ser utilizado, também a URL `”http://servidor/SAM/model/[modelo]/devices/antennas”` onde `[modelo]` indica o modelo de propagação que será utilizado para o cálculo dos alcances.

Este método recebe alguns parâmetros, que são submetidos pelo cliente como parâmetros de URL:

- *range* – Um booleano que indica se a API deve calcular o alcance das antenas ou não. É indicado pelo parâmetro de URL `”range”`. Deve ser `”true”` ou `”false”`. Se nenhum valor for indicado, será utilizado o valor padrão `”false”`.
- *modeltype* – O modelo de propagação a ser utilizado. É indicado no corpo da URL. Só utilizado pela API de testes. A API padrão utiliza automaticamente o modelo Log Distance.
- *par* – Constitui de parâmetros específicos para um determinado modelo. Só utilizado pela API de testes. Cada modelo sabe quais parâmetros deve buscar.

Para o modelo Log Distance o valor de α pode ser indicado pelo parâmetro de URL "alpha".

Exemplo da URL utilizando a API padrão:

```
1 http://localhost:8182/SAM/devices/antennas?  
2   range=true
```

Exemplo da URL utilizando a API de teste:

```
1 http://localhost:8182/SAM/model/1/devices/antennas?  
2   range=true&alpha=2.5
```

A resposta desse método é uma lista de objetos *DeviceRange*, como indicado na seção anterior.

```
1   public ArrayList<DeviceRange> getAntenas(float lat, float lng,  
2     float ray, boolean range,  
       int modeltype, ModelParameter par);
```

Este método retorna todas as antenas na base que estiverem dentro de um círculo, bem como o seu alcance. O centro e raio do círculo devem ser indicados pelo cliente.

Ele é alcançado pela URL "*http://servidor/SAM/devices/antennas/location*", utilizando o método GET. Pode ser utilizado, também a URL "*http://servidor/SAM/ model/[modelo]/devices/antennas/location*" onde [*modelo*] indica o modelo de propagação que será utilizado para o cálculo dos alcances.

Esse método recebe alguns parâmetros, que são submetidos pelo cliente como parâmetros de URL:

- *range* – Um booleano que indica se a API deve calcular o alcance das antenas ou não. É indicado pelo parâmetro de URL "range". Deve ser "true" ou "false". Se nenhum valor for indicado, será utilizado o valor padrão "false".
- *lat* – A latitude do centro do círculo. É indicado pelo parâmetro de URL "lat".
- *lng* – A longitude do centro do círculo. É indicado pelo parâmetro de URL "lng".
- *ray* – O raio do círculo. É indicado pelo parâmetro de URL "ray". O valor de "ray" é em metros.

- *modeltype* – O modelo de propagação a ser utilizado. É indicado no corpo da URL. Só utilizado pela API de testes. A API padrão utiliza automaticamente o modelo LogDistance.
- *par* – Constituído de parâmetros específicos para um determinado modelo. Só utilizado pela API de testes. Cada modelo sabe quais parâmetros deve buscar. Para o modelo Log Distance o valor de α pode ser indicado pelo parâmetro de URL "alpha".

Exemplo da URL utilizando a API padrão:

```
1 http://localhost:8182/SAM/devices/antennas/location?
2   range=true&lat=-22.845882&lng=-43.08815&ray=1000
```

Exemplo da URL utilizando a API de teste:

```
1 http://localhost:8182/SAM/model/1/devices/antennas/location?
2   range=true&lat=-22.845882&lng=-43.08815&ray=1000&alpha=2.5
```

A resposta desse método é uma lista de objetos *DeviceRange*, como indicado na seção anterior.

```
1   public ArrayList<DeviceRange> getAntennas(Channel cha, boolean
2     range,
3     int modeltype, ModelParameter par)
```

Este método retorna todas as antenas que atuam em um determinado canal, bem como o seu alcance. O canal deve ser indicado pelo cliente.

Ele é alcançado pela URL "*http://servidor/SAM/devices/antennas/channel*", utilizando o método GET. Pode ser utilizado, também a URL "*http://servidor/SAM/model/[modelo]/devices/antennas/channel*" onde *[modelo]* indica o modelo de propagação que será utilizado para o cálculo dos alcances.

Esse método recebe alguns parâmetros, que são submetidos pelo cliente como parâmetros de URL:

- *cha* – O canal a ser utilizado na busca. É indicado pelo parâmetro de URL "cha". Deve ser indicado o número do canal como definido pela ANATEL na tabela 3.1.
- *modeltype* – O modelo de propagação a ser utilizado. É indicado no corpo da URL. Só utilizado pela API de testes. A API padrão utiliza automaticamente o modelo Log Distance.

- *par* – Constituído de parâmetros específicos para um determinado modelo. Só utilizado pela API de testes. Cada modelo sabe quais parâmetros deve buscar. Para o modelo Log Distance o valor de α pode ser indicado pelo parâmetro de URL "alpha".

Exemplo da URL utilizando a API padrão:

```
1 http://localhost:8182/SAM/devices/antennas/channel?
2 range=true&cha=4
```

Exemplo da URL utilizando a API de teste:

```
1 http://localhost:8182/SAM/model/1/devices/antennas/channel?
2 range=true&cha=4&alpha=2.5
```

A resposta desse método é uma lista de objetos *DeviceRange*, como indicado na seção anterior.

WSDeviceAPI

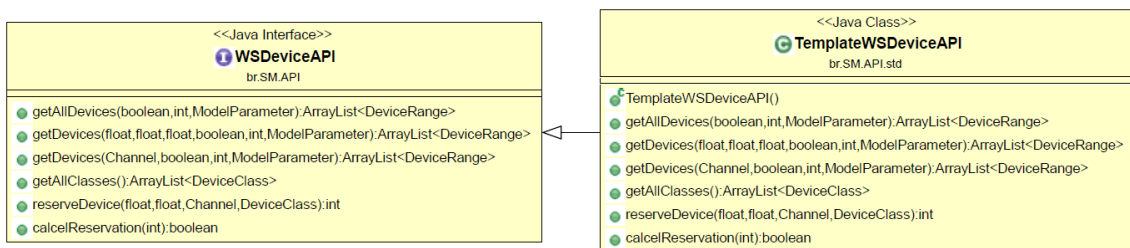


Figura 3.20: Diagrama da classe *WSDeviceAPI*.

A classe *WSDeviceAPI*, representada na figura 3.20, define seis métodos:

```
1 public ArrayList<DeviceRange> getAllDevices(boolean range, int
   modeltype, ModelParameter par);
```

Este método retorna uma lista com todos os dispositivos secundários presentes na base de dados, bem como o seu alcance.

Ele é alcançado pela URL "*http://servidor/SAM/devices/ws*", utilizando o método GET. Pode ser utilizado, também a URL "*http://servidor/SAM/model/[modelo]/devices/ws*" onde *[modelo]* indica o modelo de propagação que será utilizado para o cálculo dos alcances.

Este método recebe alguns parâmetros, que são submetidos pelo cliente como parâmetros de URL:

- *range* – Um booleano que indica se a API deve calcular o alcance dos dispositivos ou não. É indicado pelo parâmetro de URL "range". Deve ser "true" ou "false". Se nenhum valor for indicado, será utilizado o valor padrão "false".
- *modeltype* – O modelo de propagação a ser utilizado. É indicado no corpo da URL. Só utilizado pela API de testes. A API padrão utiliza automaticamente o modelo Log Distance.
- *par* – Constituído de parâmetros específicos para um determinado modelo. Só utilizado pela API de testes. Cada modelo sabe quais parâmetros deve buscar. Para o modelo Log Distance o valor de α pode ser indicado pelo parâmetro de URL "alpha".

Exemplo da URL utilizando a API padrão:

```
1 http://localhost:8182/SAM/devices/ws?
2   range=true
```

Exemplo da URL utilizando a API de teste:

```
1 http://localhost:8182/SAM/model/1/devices/ws?
2   range=true&alpha=2.5
```

A resposta desse método é uma lista de objetos *DeviceRange*, como indicado na seção anterior.

```
1   public ArrayList<DeviceRange> getDevices(float lat, float lng,
2     float ray, boolean range,
3     int modeltype, ModelParameter par);
```

Este método retorna todos dispositivos secundários da base que estiverem dentro de um círculo, bem como o seu alcance. O centro e raio do círculo devem ser indicados pelo cliente.

Ele é alcançado pela URL "*http://servidor/SAM/devices/ws/location*", utilizando o método GET. Pode ser utilizado, também a URL "*http://servidor/SAM/model/[modelo]/devices/ws/location*" onde *[modelo]* indica o modelo de propagação que será utilizado para o cálculo dos alcances.

Esse método recebe alguns parâmetros, que são submetidos pelo cliente como parâmetros de URL:

- *range* – Um booleano que indica se a API deve calcular o alcance dos dispositivos ou não. É indicado pelo parâmetro de URL "range". Deve ser "true" ou "false". Se nenhum valor for indicado, será utilizado o valor padrão "false".

- *lat* – A latitude do centro do círculo. É indicado pelo parâmetro de URL "lat".
- *lng* – A longitude do centro do círculo. É indicado pelo parâmetro de URL "lng".
- *ray* – O raio do círculo. É indicado pelo parâmetro de URL "ray". O valor de "ray" é em metros.
- *modeltype* – O modelo de propagação a ser utilizado. É indicado no corpo da URL. Só utilizado pela API de testes. A API padrão utiliza automaticamente o modelo Log Distance.
- *par* – Constituído de parâmetros específicos para um determinado modelo. Só utilizado pela API de testes. Cada modelo sabe quais parâmetros deve buscar. Para o modelo Log Distance o valor de α pode ser indicado pelo parâmetro de URL "alpha".

Exemplo da URL utilizando a API padrão:

```
1 http://localhost:8182/SAM/devices/ws/location?
2   range=true&lat=-22.845882&lng=-43.08815&ray=1000
```

Exemplo da URL utilizando a API de teste:

```
1 http://localhost:8182/SAM/model/1/devices/ws/location?
2   range=true&lat=-22.845882&lng=-43.08815&ray=1000&alpha=2.5
```

A resposta desse método é uma lista de objetos *DeviceRange*, como indicado na seção anterior.

```
1   public ArrayList<DeviceRange> getDevices(Channel cha, boolean
2     range, int modeltype, ModelParameter par);
```

Este método retorna todos os dispositivos secundários que atuam em um determinado canal, bem como o seu alcance. O canal deve ser indicado pelo cliente.

Ele é alcançado pela URL "http://servidor/SAM/devices/ws/channel", utilizando o método GET. Pode ser utilizado, também a URL "http://servidor/SAM/model/[modelo]/devices/ws/channel" onde [modelo] indica o modelo de propagação que será utilizado para o cálculo dos alcances.

Esse método recebe alguns parâmetros, que são submetidos pelo cliente como parâmetros de URL:

- *cha* – O canal a ser utilizado na busca. É indicado pelo parâmetro de URL "cha". Deve ser indicado o número do canal como definido pela ANATEL na tabela 3.1.
- *modeltype* – O modelo de propagação a ser utilizado. É indicado no corpo da URL. Só utilizado pela API de testes. A API padrão utiliza automaticamente o modelo Log Distance.
- *par* – Constitui de parâmetros específicos para um determinado modelo. Só utilizado pela API de testes. Cada modelo sabe quais parâmetros deve buscar. Para o modelo Log Distance o valor de α pode ser indicado pelo parâmetro de URL "alpha".

Exemplo da URL utilizando a API padrão:

```
1 http://localhost:8182/SAM/devices/ws/channel?
2   range=true&cha=4
```

Exemplo da URL utilizando a API de teste:

```
1 http://localhost:8182/SAM/model/1/devices/ws/channel?
2   range=true&cha=4&alpha=2.5
```

A resposta desse método é uma lista de objetos *DeviceRange*, como indicado na seção anterior.

```
1   public ArrayList<DeviceClass> getAllClasses();
```

Esse método retorna todos os tipos de dispositivos secundários que a base de dados aceita. Ele deve ser usado pelo cliente para determinar em qual tipo ele se enquadra.

Ele é alcançado pela URL "*http://servidor/SAM/devices/ws/classes*", utilizando o método GET.

Exemplo da URL:

```
1 http://localhost:8182/SAM/devices/ws/classes
```

A resposta desse método é uma lista de objetos *DeviceClass*, que tem as seguintes informações:

- Potência do dispositivo em Miliwatts – *DeviceClass.erp*

- Nome do dispositivo – *DeviceClass.name*

Exemplo de resposta em XML:

```

1 <list >
2   <br .SM.DO. DeviceClass >
3     <id >1</id >
4     <erp >10.0</erp >
5     <name >INTERNO BAIXO</name >
6   </br .SM.DO. DeviceClass >
7   <br .SM.DO. DeviceClass >
8     <id >2</id >
9     <erp >100.0</erp >
10    <name >EXTERNO</name >
11  </br .SM.DO. DeviceClass >
12  <br .SM.DO. DeviceClass >
13    <id >3</id >
14    <erp >50.0</erp >
15    <name >INTERNO ALTO</name >
16  </br .SM.DO. DeviceClass >
17 </list >

```

```

1   public int reserveDevice(float lat , float lng , Channel cha ,
      DeviceClass cls );

```

Este método é utilizado pelo cliente para cadastrar um dispositivo secundário na base. Quando o cadastro é feito, a base reserva o canal indicado para o dispositivo. Devem ser informadas a localização do dispositivo, o canal que ele vai utilizar e o tipo de dispositivo.

Ele é alcançado pela URL *"http://servidor/SAM/devices/ws"*, utilizando o método PUT.

Este método recebe alguns parâmetros, que são submetidos pelo cliente como parâmetros de URL:

- *lat* – A latitude do dispositivo. É indicado pelo parâmetro de URL "lat".
- *lng* – A longitude do dispositivo. É indicado pelo parâmetro de URL "lng".
- *cha* – O canal a ser utilizado na busca. É indicado pelo parâmetro de URL "cha". Deve ser indicado o número do canal como definido pela ANATEL na tabela 3.1.
- *cls* – O tipo de dispositivo que deseja utilizar o WS. É indicado pelo parâmetro de URL "type".

Exemplo da URL:

```
1 http://localhost:8182/SAM/devices/ws?  
2   lat=-22.845882&lng=-43.08815&cha=4&type=INTERNO+BAIXO
```

Este método retorna um inteiro para o cliente, que identifica o cadastro do dispositivo na base. Esse valor é utilizado para cancelar a reserva feita para o dispositivo. Se a resposta for 0, então a reserva não foi concluída

```
1 public boolean cancelReservation(int id);
```

Este método é responsável por cancelar uma reserva realizada por um dispositivo. O cliente deve indicar o dispositivo que foi cadastrado na base.

Ele é alcançado pela URL `http://servidor/SAM/devices/ws`, utilizando o método DELETE.

Este método recebe um parâmetro, que é submetido pelo cliente como parâmetros de URL:

- *id* – A identificação da reserva a ser cancelada. É indicado pelo parâmetro de URL "id". Essa identificação é a resposta do método *reserveDevice*

Exemplo da URL:

```
1 http://localhost:8182/SAM/devices/ws?  
2   id=2
```

TransmissionDeviceAPI

A classe *TransmissionDeviceAPI*, representada na figura 3.21, define 3 métodos:

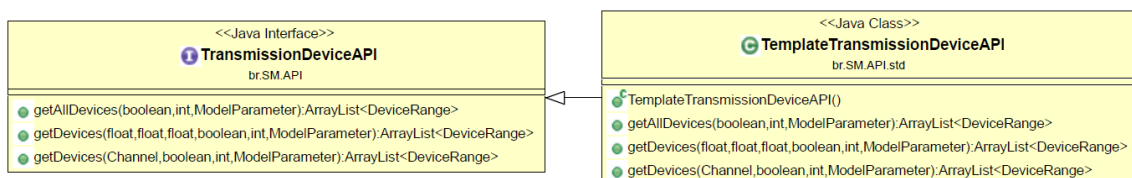


Figura 3.21: Diagrama da classe *TransmissionDeviceAPI*.

```
1 public ArrayList<DeviceRange> getAllDevices(boolean range, int  
2   modeltype,  
   ModelParameter par);
```

Este método retorna uma lista com todos os dispositivos de transmissão presentes na base de dados, bem como o seu alcance. Isso inclui todas as antenas e dispositivos secundários.

Ele é alcançado pela URL `"http://servidor/SAM/devices"`, utilizando o método GET. Pode ser utilizado, também a URL `"http://servidor/SAM/model/[modelo]/devices"` onde `[modelo]` indica o modelo de propagação que será utilizado para o cálculo dos alcances.

Este método recebe alguns parâmetros, que são submetidos pelo cliente como parâmetros de URL:

- *range* – Um booleano que indica se a API deve calcular o alcance dos dispositivos ou não. É indicado pelo parâmetro de URL "range". Deve ser "true" ou "false". Se nenhum valor for indicado, será utilizado o valor padrão "false".
- *modeltype* – O modelo de propagação a ser utilizado. É indicado no corpo da URL. Só utilizado pela API de testes. A API padrão utiliza automaticamente o modelo Log Distance.
- *par* – Constituído de parâmetros específicos para um determinado modelo. Só utilizado pela API de testes. Cada modelo sabe quais parâmetros deve buscar. Para o modelo Log Distance o valor de α pode ser indicado pelo parâmetro de URL "alpha".

Exemplo da URL utilizando a API padrão:

```
1 http://localhost:8182/SAM/devices?  
2   range=true
```

Exemplo da URL utilizando a API de teste:

```
1 http://localhost:8182/SAM/model/1/devices?  
2   range=true&alpha=2.5
```

A resposta desse método é uma lista de objetos *DeviceRange*, como indicado na seção anterior.

```
1 public ArrayList<DeviceRange> getDevices(float lat, float lng, float  
   ray, boolean range,  
2       int modeltype, ModelParameter par);
```

Este método retorna todos os dispositivos de transmissão que estiverem dentro de um círculo, bem como o seu alcance. Isso inclui as antenas e dispositivos secundários. O centro e raio do círculo devem ser indicados pelo cliente.

Ele é alcançado pela URL `"http://servidor/SAM/devices/location"`, utilizando o método GET. Pode ser utilizado, também a URL `"http://servidor/SAM/model/[modelo]/devices/location"` onde `[modelo]` indica o modelo de propagação que será utilizado para o cálculo dos alcances.

Esse método recebe alguns parâmetros que são submetidos pelo cliente como parâmetros de URL:

- *range* – Um booleano que indica se a API deve calcular o alcance dos dispositivos ou não. É indicado pelo parâmetro de URL "range". Deve ser "true" ou "false". Se nenhum valor for indicado, será utilizado o valor padrão "false".
- *lat* – A latitude do centro do círculo. É indicado pelo parâmetro de URL "lat".
- *lng* – A longitude do centro do círculo. É indicado pelo parâmetro de URL "lng".
- *ray* – O raio do círculo. É indicado pelo parâmetro de URL "ray". O valor de "ray" é em metros.
- *modeltype* – O modelo de propagação a ser utilizado. É indicado no corpo da URL. Só utilizado pela API de testes. A API padrão utiliza automaticamente o modelo Log Distance.
- *par* – Constituído de parâmetros específicos para um determinado modelo. Só utilizado pela API de testes. Cada modelo sabe quais parâmetros deve buscar. Para o modelo Log Distance o valor de α pode ser indicado pelo parâmetro de URL "alpha".

Exemplo da URL utilizando a API padrão:

```
1 http://localhost:8182/SAM/devices/location?  
2   range=true&lat=-22.845882&lng=-43.08815&ray=1000
```

Exemplo da URL utilizando a API de teste:

```
1 http://localhost:8182/SAM/model/1/devices/location?  
2   range=true&lat=-22.845882&lng=-43.08815&ray=1000&alpha=2.5
```

A resposta desse método é uma lista de objetos *DeviceRange*, como indicado na seção anterior.

```
1 public ArrayList<DeviceRange> getDevices(Channel cha, boolean range,
2     int modeltype, ModelParameter par);
```

Este método retorna todos os dispositivos de transmissão que atuam em um determinado canal, bem como o seu alcance. O canal deve ser indicado pelo cliente.

Ele é alcançado pela URL `"http://servidor/SAM/devices/channel"`, utilizando o método GET. Pode ser utilizado, também a URL `"http://servidor/SAM/model/[modelo]/devices/channel"` onde `[modelo]` indica o modelo de propagação que será utilizado para o cálculo dos alcances.

Esse método recebe alguns parâmetros, que são submetidos pelo cliente como parâmetros de URL:

- *cha* – O canal a ser utilizado na busca. É indicado pelo parâmetro de URL "cha". Deve ser indicado o número do canal como definido pela ANATEL na tabela 3.1.
- *modeltype* – O modelo de propagação a ser utilizado. É indicado no corpo da URL. Só utilizado pela API de testes. A API padrão utiliza automaticamente o modelo Log Distance.
- *par* – Constitui de parâmetros específicos para um determinado modelo. Só utilizado pela API de testes. Cada modelo sabe quais parâmetros deve buscar. Para o modelo Log Distance o valor de α pode ser indicado pelo parâmetro de URL "alpha".

Exemplo da URL utilizando a API padrão:

```
1 http://localhost:8182/SAM/devices/channel?
2   range=true&cha=4
```

Exemplo da URL utilizando a API de teste:

```
1 http://localhost:8182/SAM/model/1/devices/channel?
2   range=true&cha=4&alpha=2.5
```

A resposta desse método é uma lista de objetos *DeviceRange*, como indicado na seção anterior.

Capítulo 4

Análises Realizadas

Um dos objetivos do sistema é servir como fonte para futuras pesquisas na área. Para determinar essa possibilidade, foi feita uma análise da base desenvolvida. Essa análise pode ajudar a determinar o melhor modelo de propagação a ser utilizado. Foi avaliada a disponibilidade de canais para o Estado do Rio de Janeiro, ou seja, a quantidade de canais disponíveis para localizações predefinidas do Estado. Foi feita, também, uma análise individual sobre os canais, de forma a determinar quais são os canais que se encontram com maior disponibilidade.

Essas análises ajudam a identificar a possibilidade de implantação de rádios cognitivos no Rio de Janeiro.

4.1 Dataset criado

Primeiramente foram selecionados pontos distribuídos homogeneamente pelo Estado do Rio de Janeiro.

Para isso, foi criado um retângulo em volta do Estado conforme indicado na figura 4.1.

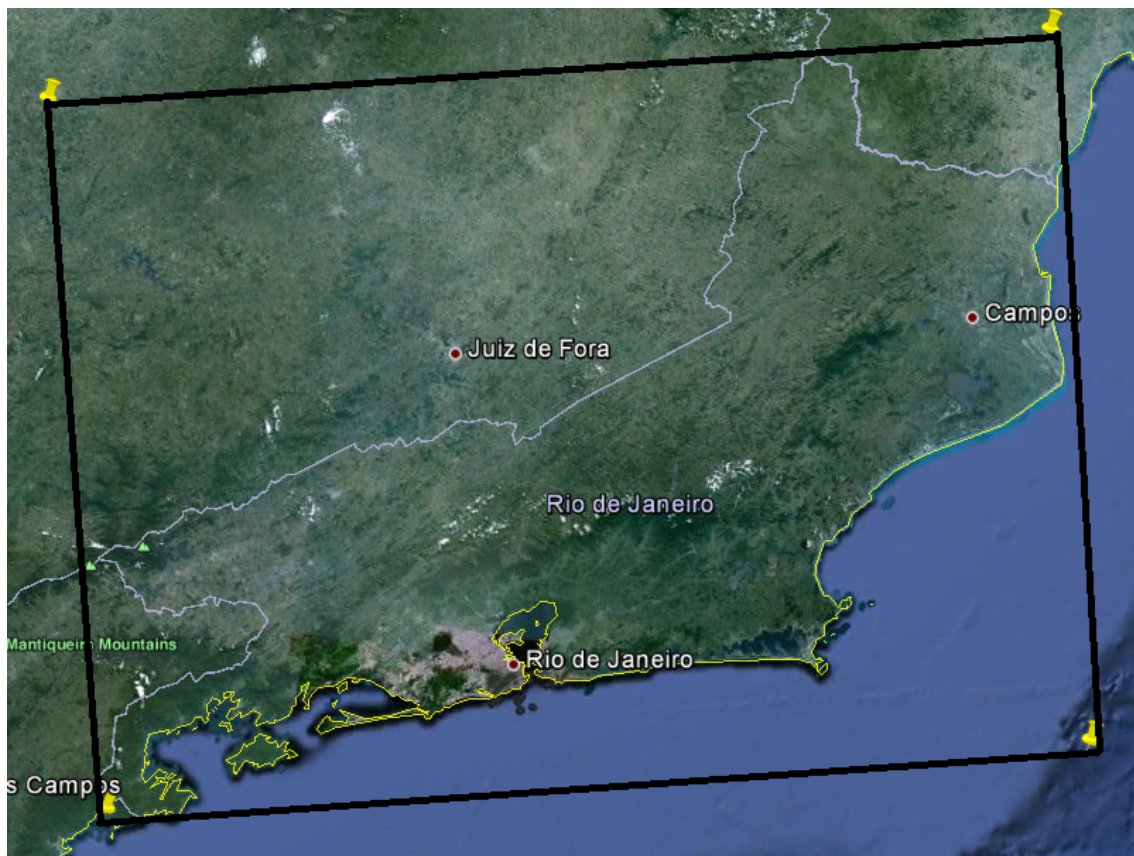


Figura 4.1: Dataset Utilizado

Esse retângulo foi percorrido em sua totalidade, variando a latitude e a longitude em 0,01. Para determinar quais desses pontos de fato faziam parte do Rio foi utilizado um serviço online aberto de geolocalização, o GeoNames [28]. Foi utilizado o serviço de geocodificação reversa, onde é informado ao sistema um par de coordenadas geográficas e um endereço é recebido como resposta. A partir desse endereço foi identificado a qual Estado o ponto pertencia.

Após se percorrer todo o retângulo foram obtidos 26776 pontos.

4.2 Disponibilidade de canais no Rio de Janeiro

O sistema desenvolvido foi utilizado para identificar, para cada um dos 26776 pontos, a disponibilidade de canais. Para cada um desses pontos o sistema foi acionado 7 vezes, sempre utilizando um modelo ou parâmetro diferente.

Para efeito de cálculo, foi considerado para cada ponto calculado um US com potência de 10 W. Esse valor, embora acima dos valores médios indicados no capítulo anterior, apresentou resultados satisfatórios. Esse valor foi escolhido para demonstrar uma situação de pior caso de disponibilidade.

Os seguintes modelos foram utilizados nas buscas:

- Modelo Free Spaces
- Modelo Two Ray Ground
- Modelo Log Distance, utilizando $\alpha = 2$
- Modelo Log Distance, utilizando $\alpha = 2, 5$
- Modelo Log Distance, utilizando $\alpha = 3$
- Modelo Log Distance, utilizando $\alpha = 3, 5$
- Modelo Log distance, utilizando $\alpha = 4$

Para cada um desses modelos foram plotados histogramas, com informações sobre os canais mais disponíveis, e foram criados mapas de calor utilizando a ferramenta MyHeatMaps [29]. Os mapas representam a quantidade de canais disponível por localização.

Os mapas de calor possuem cores que vão do roxo escuro ao amarelo. O roxo representa localizações onde o número de canais disponíveis é o mínimo identificado, enquanto o amarelo representa áreas onde o número de canais corresponde ao máximo. Cores entre o roxo e o amarelo representam regiões com valores entre o mínimo e o máximo apresentados.

Os canais 63 até 68 foram encontrados como disponíveis em todas as localizações verificadas para todos os modelos. Eles não serão demonstrados nos histogramas por escolha estética.

4.2.1 Modelo Free Spaces

A figura 4.3 representa o histograma de disponibilidade de canais utilizando o modelo Free Space. A figura 4.2 representa o mapa de calor para esse modelo.

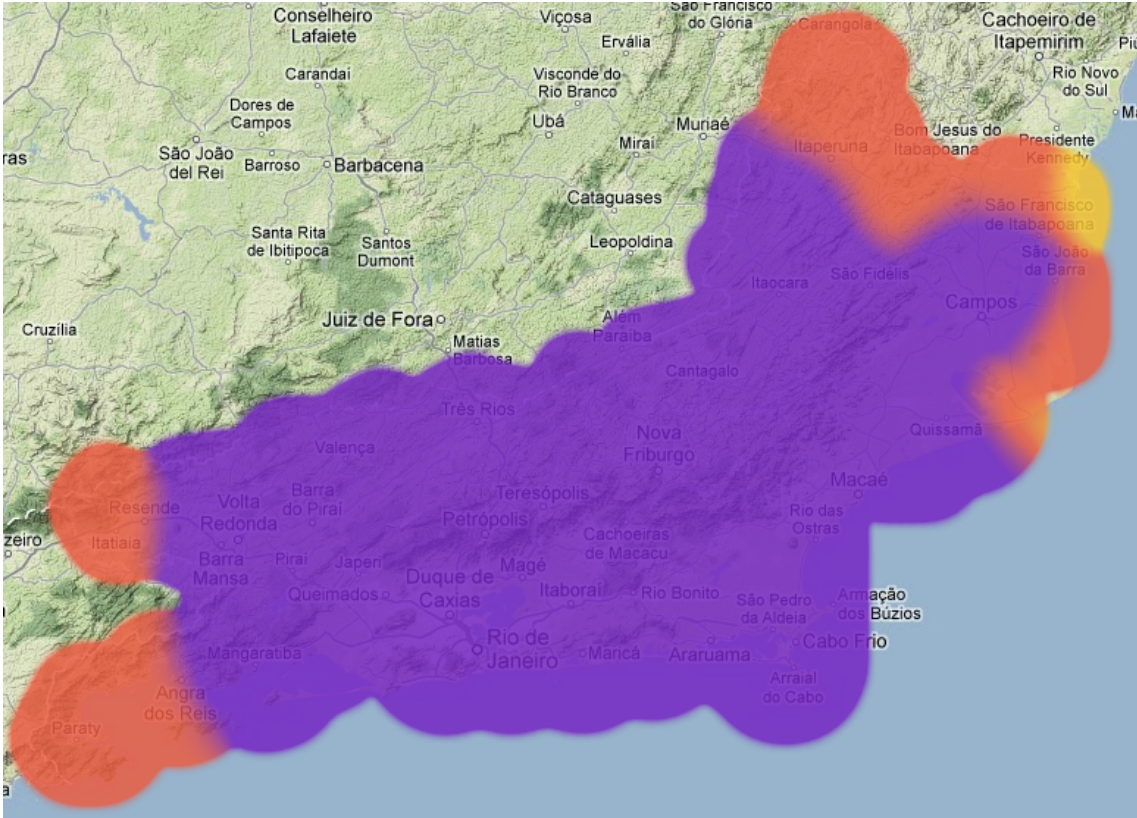


Figura 4.2: Mapa de calor para o modelo Free Space

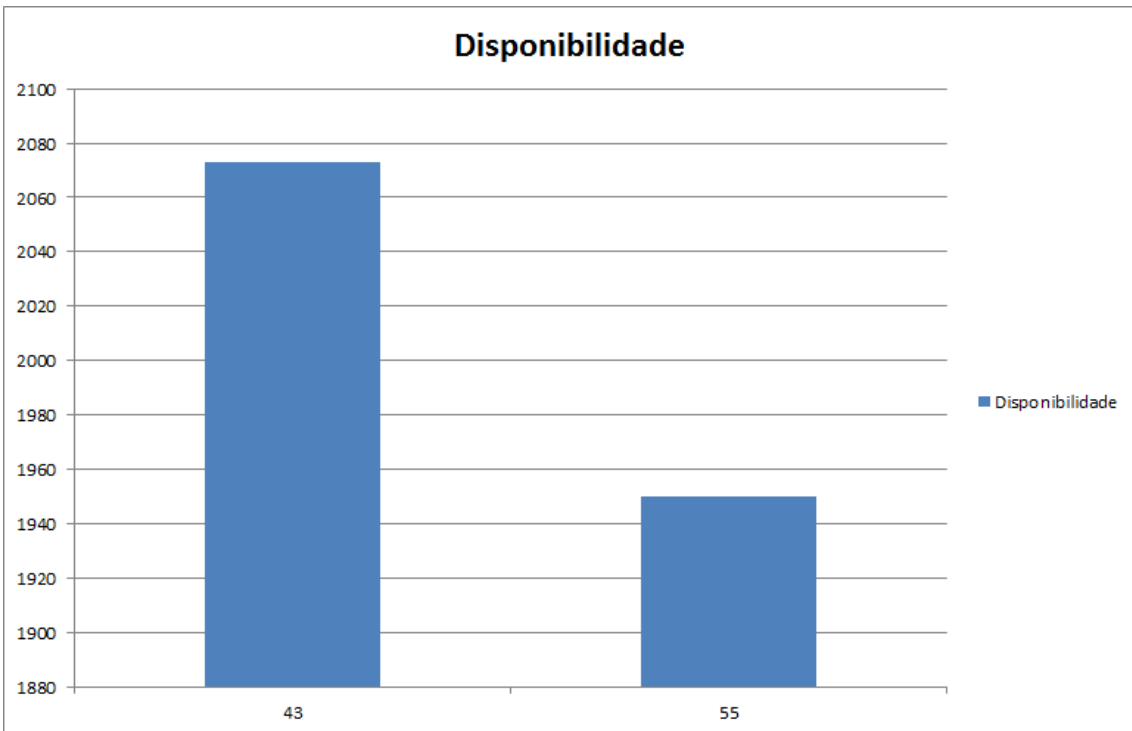


Figura 4.3: Histograma de disponibilidade de canais para o modelo Free Space

Esse modelo acusou uma média de 8,15 canais disponíveis por localização. Sendo 10 a quantidade máxima de canais disponíveis em uma localização e 8 a mínima.

4.2.2 Modelo Two Ray Ground

A figura 4.5 representa o histograma de disponibilidade de canais utilizando o modelo Two Ray Ground. A figura 4.4 representa o mapa de calor para esse modelo.

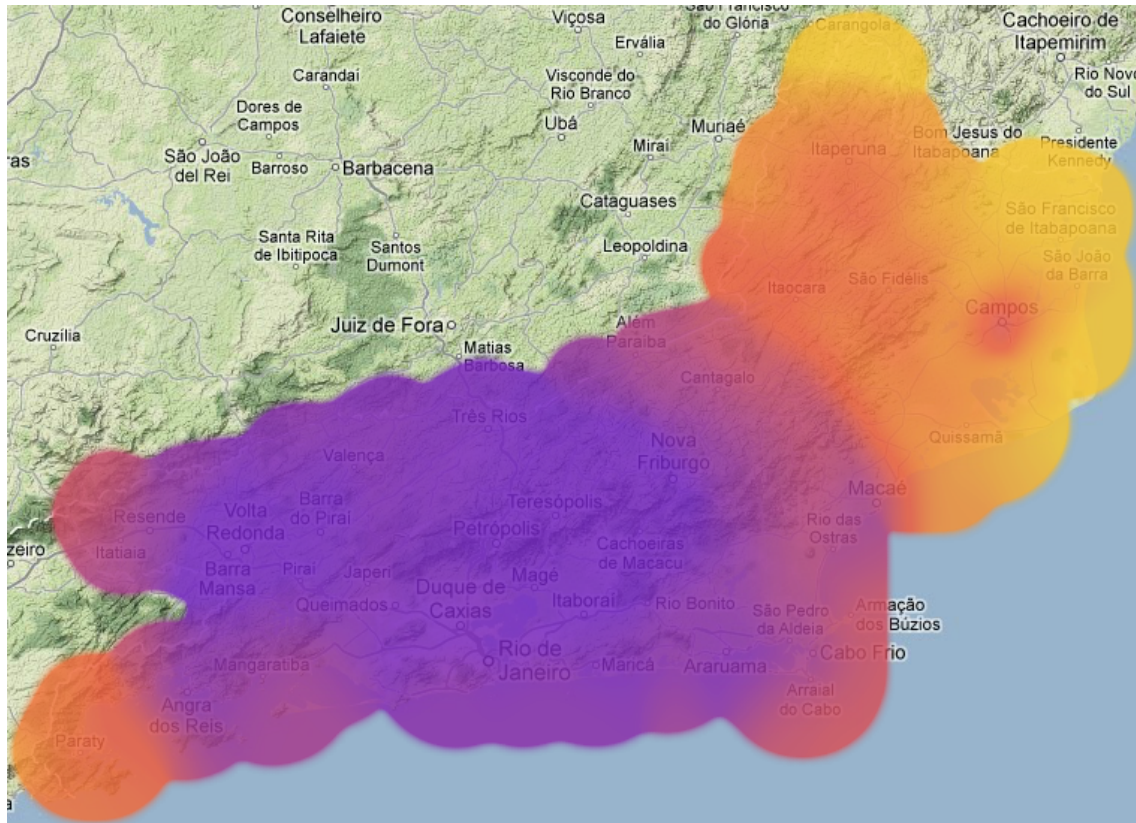


Figura 4.4: Mapa de calor para o modelo Two ray Ground

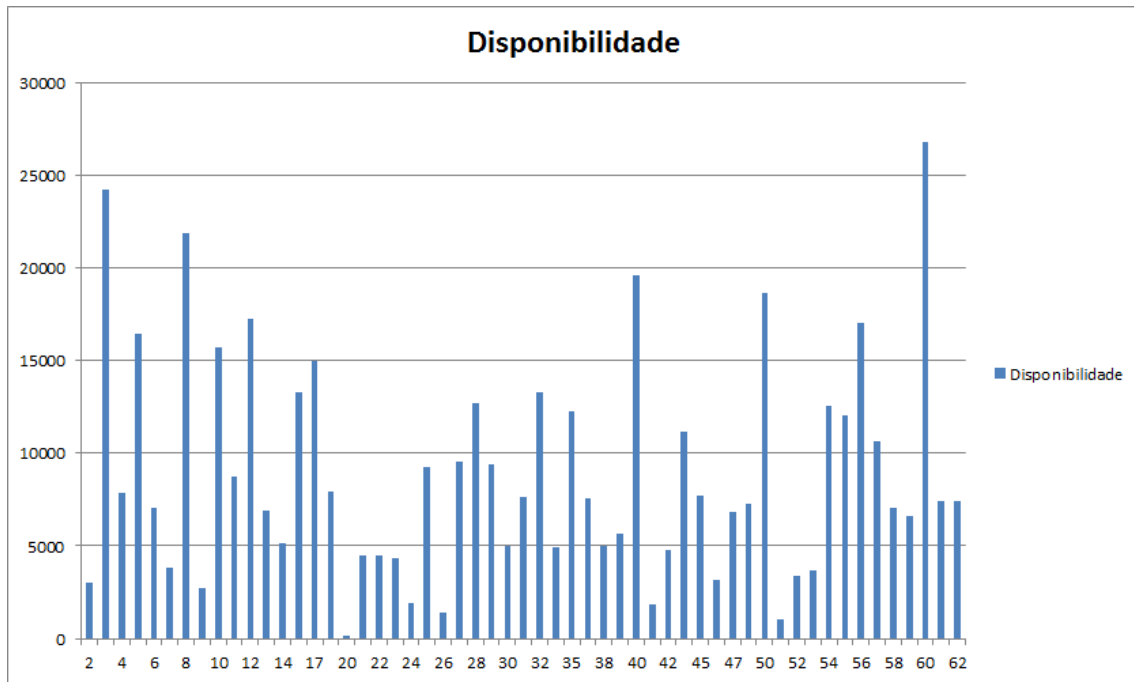


Figura 4.5: Histograma de disponibilidade de canais para o modelo Two Ray Ground

Esse modelo acusou uma média de 25,38 canais disponíveis por localização. Sendo 60 a quantidade máxima de canais disponíveis em uma localização e 10 a mínima.

4.2.3 Modelo Log Distance

Para analisar a capacidade de indicar parâmetros ao sistema, a constante de decaimento α foi alternada entre alguns valores diferentes.

Constante = 2

A figura 4.7 representa o histograma de disponibilidade de canais utilizando o modelo Log Distance usando $\alpha = 2$. A figura 4.6 representa o mapa de calor para esse modelo.

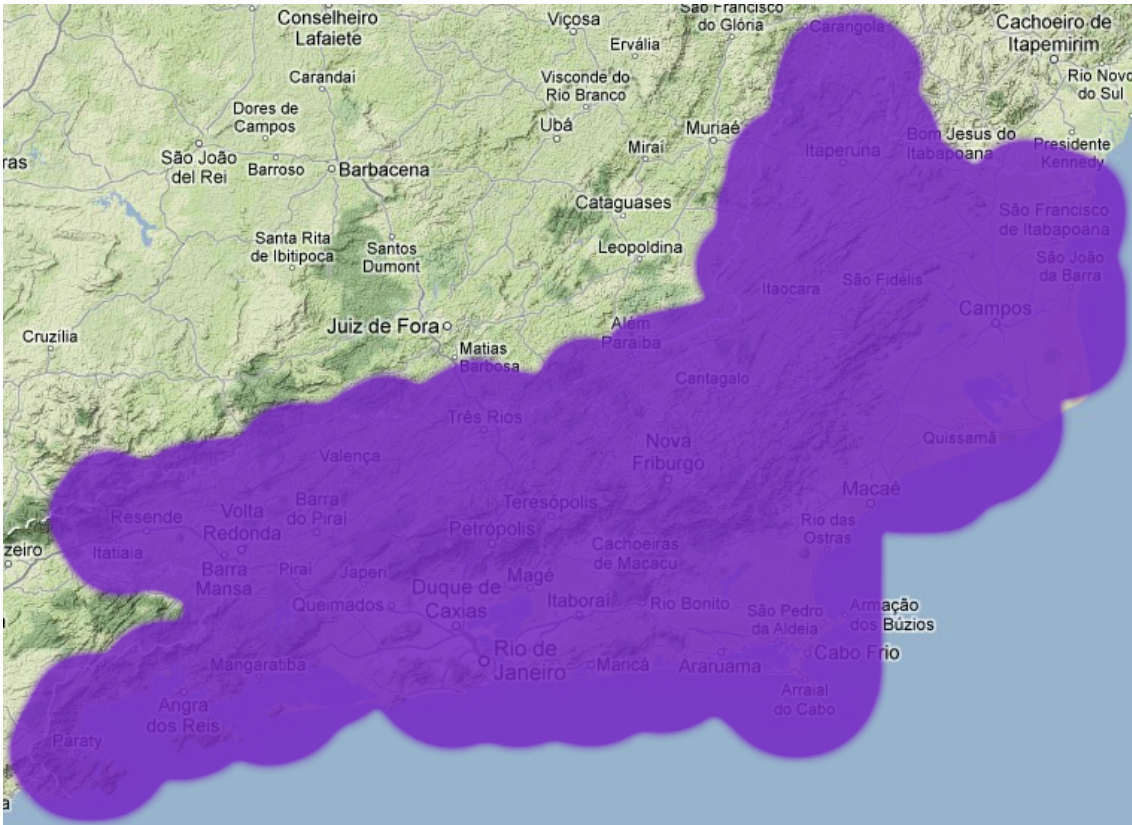


Figura 4.6: Mapa de calor para o modelo Log Distance usando 2 como constante de decaimento

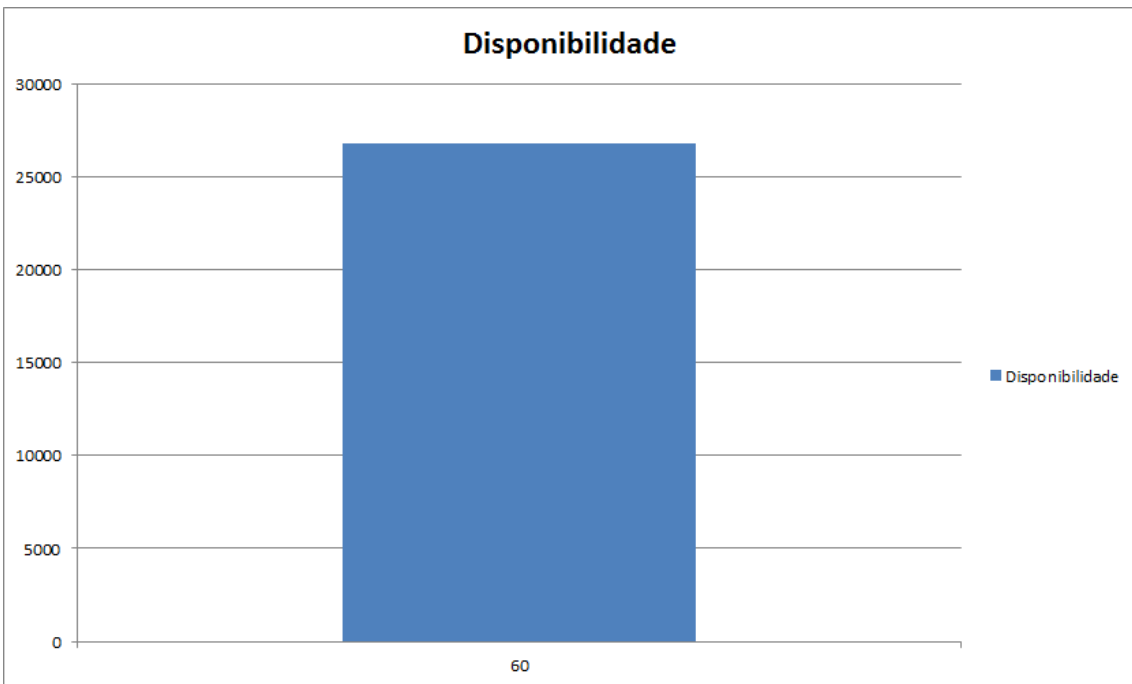


Figura 4.7: Histograma de disponibilidade de canais para o modelo Log Distance usando 2 como constante de decaimento

Esse modelo acusou uma média de 8 canais disponíveis por localização. Sendo 8 a quantidade máxima de canais disponíveis em uma localização e 8 a mínima.

Constante = 2,5

A figura 4.9 representa o histograma de disponibilidade de canais utilizando o modelo Log Distance usando $\alpha = 2,5$. A figura 4.8 representa o mapa de calor para esse modelo.

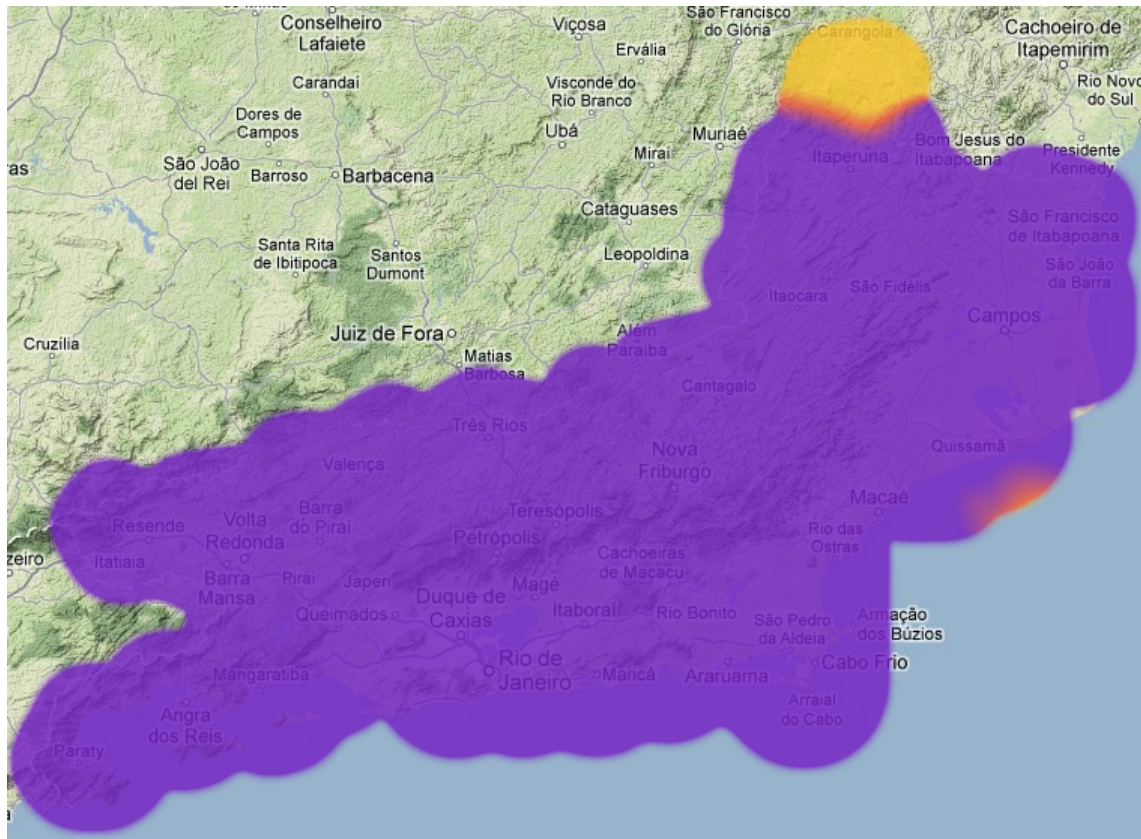


Figura 4.8: Mapa de calor para o modelo Log Distance usando 2,5 como constante de decaimento

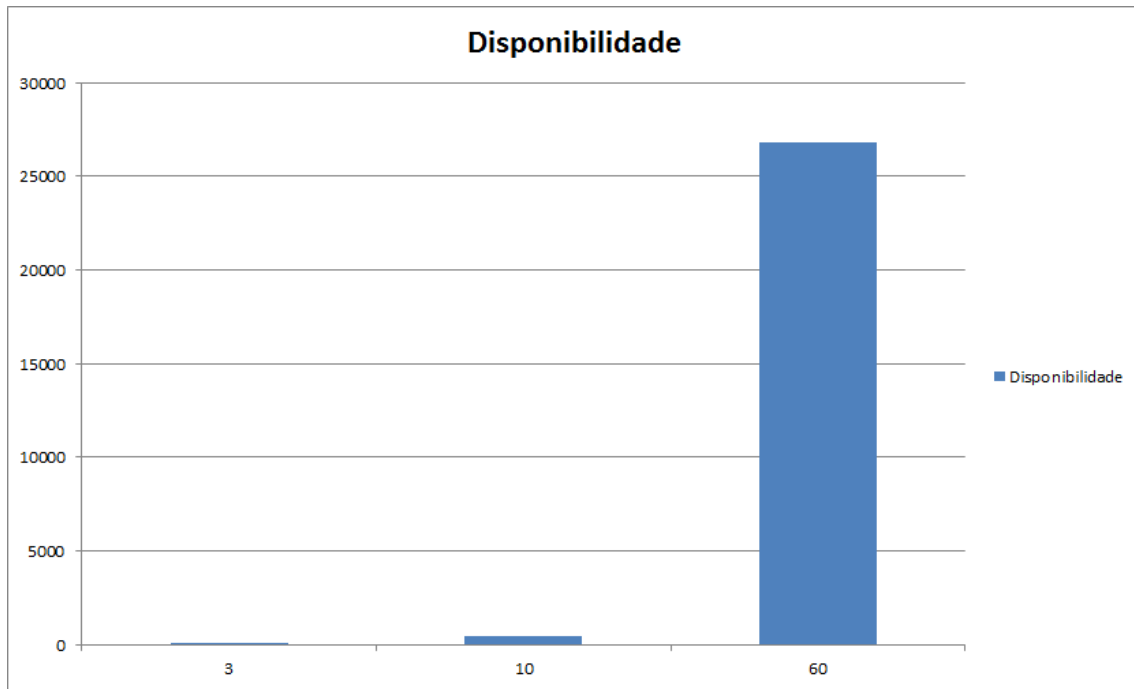


Figura 4.9: Histograma de disponibilidade de canais para o modelo Log Distance usando 2,5 como constante de decaimento

Esse modelo acusou uma média de 8,01 canais disponíveis por localização. Sendo 9 a quantidade máxima de canais disponíveis em uma localização e 8 a mínima.

Constante = 3

A figura 4.11 representa o histograma de disponibilidade de canais utilizando o modelo Log Distance usando $\alpha = 3$. A figura 4.10 representa o mapa de calor para esse modelo.

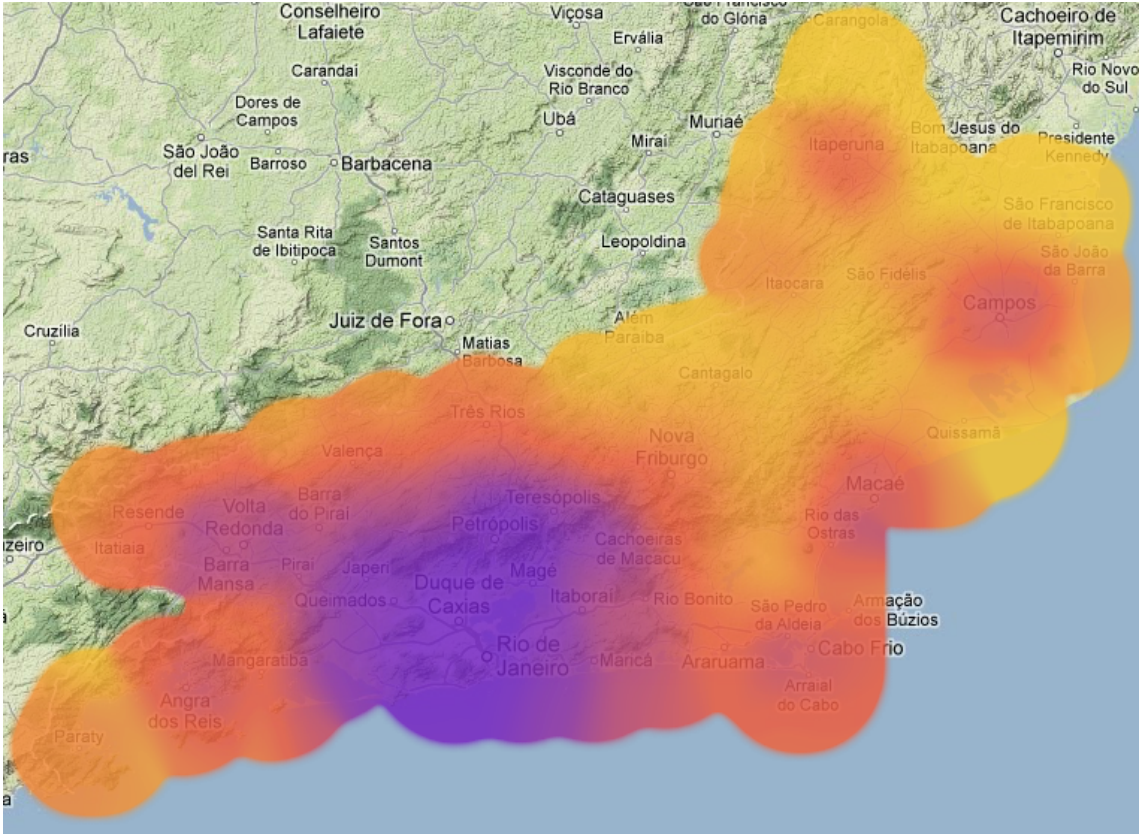


Figura 4.10: Mapa de calor para o modelo Log Distance usando 3 como constante de decaimento

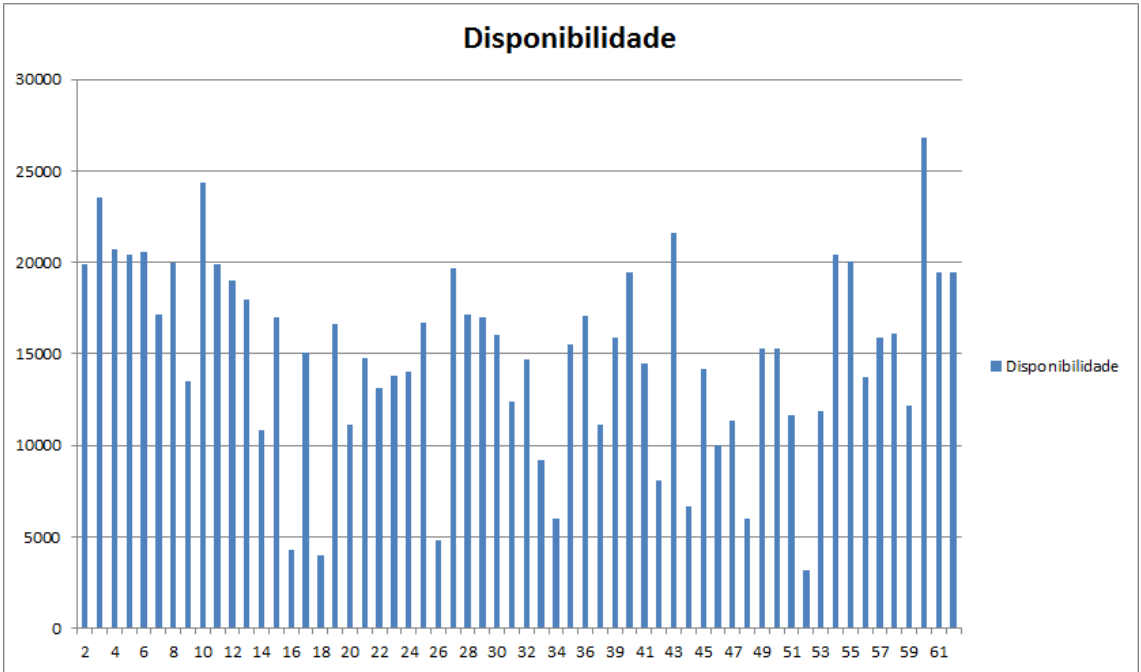


Figura 4.11: Histograma de disponibilidade de canais para o modelo Log Distance usando 3 como constante de decaimento

Esse modelo acusou uma média de 40,52 canais disponíveis por localização. Sendo 67 a quantidade máxima de canais disponíveis em uma localização e 12 a mínima.

Constante = 3,5

A figura 4.13 representa o histograma de disponibilidade de canais utilizando o modelo Log Distance usando $\alpha = 3,5$. A figura 4.12 representa o mapa de calor para esse modelo.

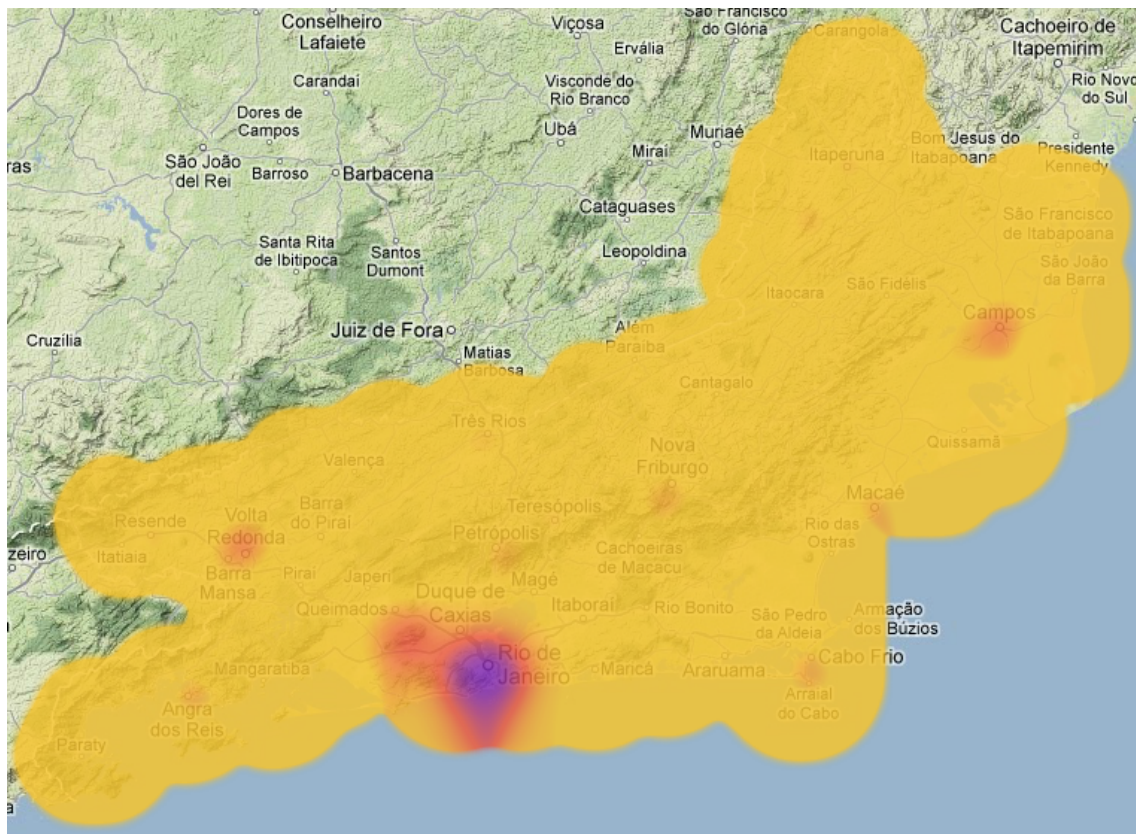


Figura 4.12: Mapa de calor para o modelo Log Distance usando 3,5 como constante de decaimento

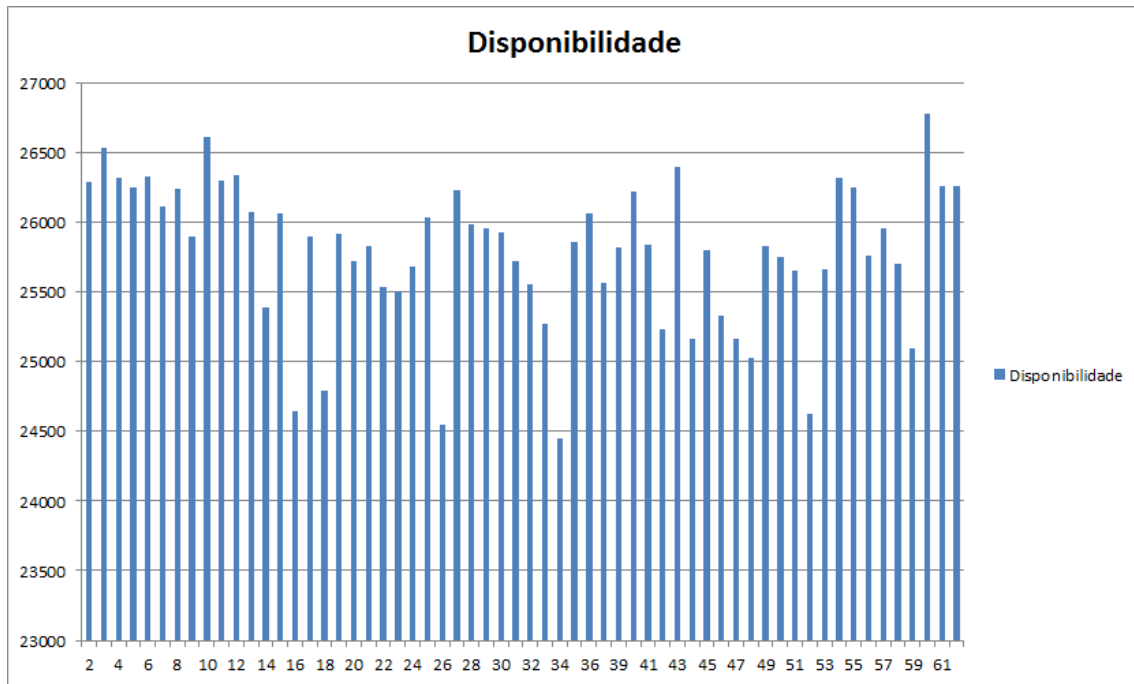


Figura 4.13: Histograma de disponibilidade de canais para o modelo Log Distance usando 3,5 como constante de decaimento

Esse modelo acusou uma média de 64,78 canais disponíveis por localização. Sendo 67 a quantidade máxima de canais disponíveis em uma localização e 22 a mínima.

Constante = 4

A figura 4.15 representa o histograma de disponibilidade de canais utilizando o modelo Log Distance usando $\alpha = 4$. A figura 4.14 representa o mapa de calor para esse modelo.

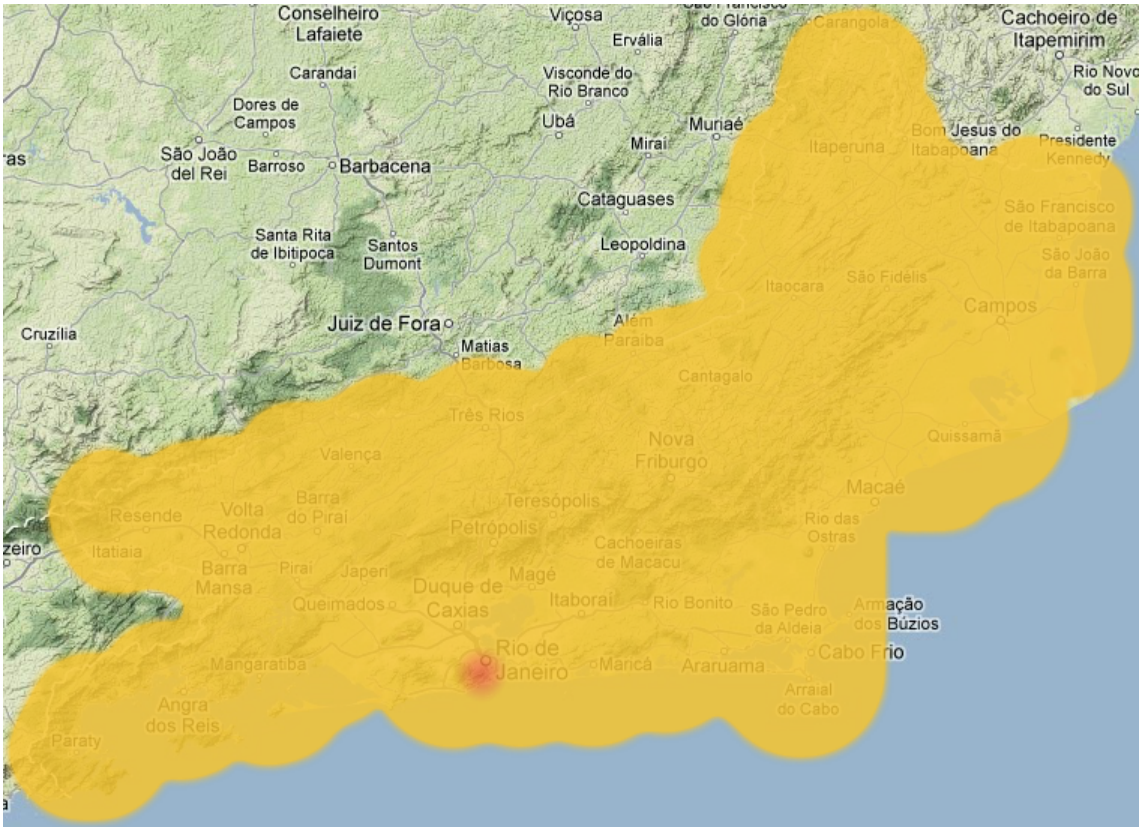


Figura 4.14: Mapa de calor para o modelo Log Distance usando 4 como constante de decaimento

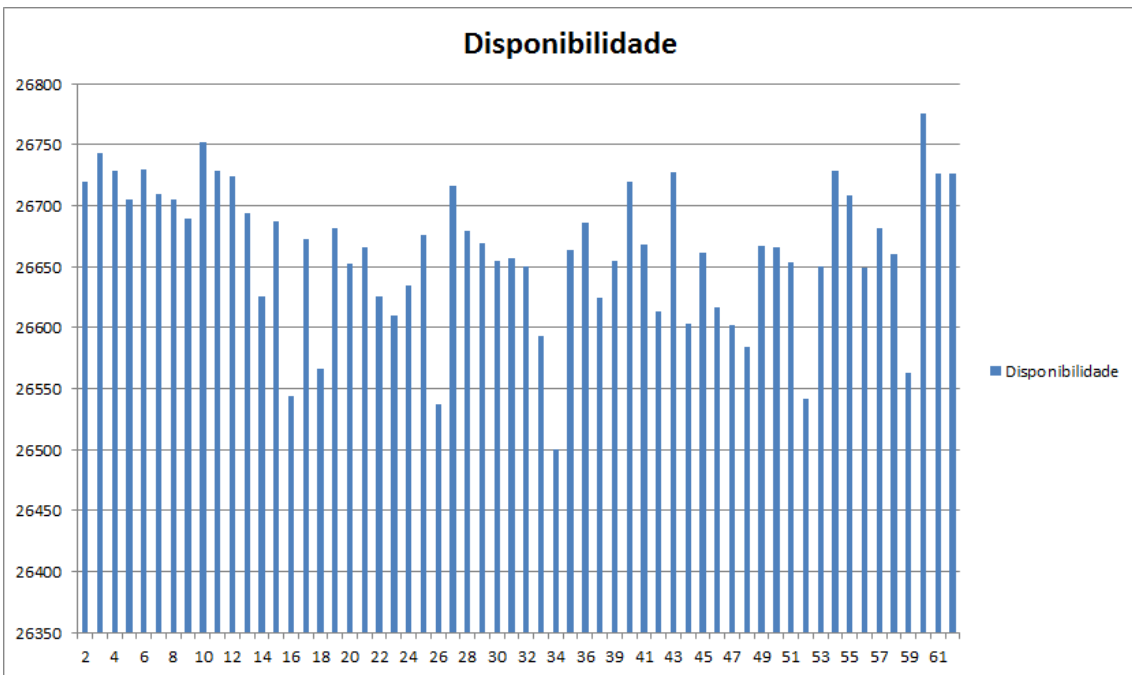


Figura 4.15: Histograma de disponibilidade de canais para o modelo Log Distance usando 4 como constante de decaimento

Esse modelo acusou uma média de 66,75 canais disponíveis por localização. Sendo 67 a quantidade máxima de canais disponíveis em uma localização e 26 a mínima.

4.3 Resultado da análise

4.3.1 Diferença entre Modelos

Como pode ser observado, a diferença de um modelo de propagação para o outro é significativa. É importante que o modelo utilizado pela base seja o mais preciso possível.

Caso o modelo seja muito sensível, ou seja, se o alcance calculado das antenas pelo modelo for muito maior que o alcance real, estará ocorrendo um desperdício. O objetivo dos rádios cognitivos, como mencionado no primeiro capítulo, é aproveitar o espectro de frequência reservado para um dispositivo primário que não está sendo utilizado. Se a base determinar um alcance maior que o real, canais que poderiam estar sendo utilizados serão considerados como ocupados.

No entanto, um modelo pouco sensível também não é bom. Caso um modelo determine alcances menores que os reais, um dispositivo secundário pode acabar escolhendo um canal que já esteja em uso por um primário. Isso acarreta uma condição de interferência entre os dispositivos. É imprescindível que essa interferência não ocorra. Um dispositivo primário tem o direito de utilizar aquele canal, que foi obtido por uma licitação, e tem prioridade total sobre os secundários.

O modelo ideal, portanto, não pode ser muito sensível, nem pouco sensível demais. É mais sensato, para a BDWS, escolher um modelo um pouco mais sensível que o ideal. Embora o desperdício possa existir, seria uma maneira de evitar a interferência entre dispositivos.

Modelo Free Space

O Free Space foi o modelo mais sensível entre os utilizados. Ao utilizá-lo, são encontrados valores muito grandes para o alcance das antenas, de forma que, analisando os mapas de calor, todo o espectro de frequência se encontra ocupado no Estado do Rio de Janeiro.

Ele não pode ser utilizado por uma BDWS, e foi implementado apenas por motivos acadêmicos.

Two Ray

O Two Ray é um pouco menos sensível que o Free Space. Os alcances das antenas calculadas por ele já são mais próximos da realidade. Os mapas de calor já apresentam um panorama mais próximo do esperado. Pode-se verificar que as regiões metropolitanas apresentam um espectro de frequência todo ocupado, enquanto que o campo apresenta regiões onde ele está todo disponível.

Esse modelo poderia ser utilizado por uma WSBD. Por prever um alcance um pouco maior que o normal é praticamente garantido que um dispositivo secundário não irá interferir com um primário. No entanto, inevitavelmente, vai desperdiçar algum canal.

Log Distance

Foram encontrados resultados bem distintos ao variar o valor da constante de decaimento α .

Ao utilizar a constante com valores 2 e 2,5, o modelo se aproxima bastante do Free Space. Assim, ele apresenta as mesmas falhas.

Já ao utilizar a constante com valor 3, o modelo chegou o mais próximo do ideal. Analisando o mapa de calor fica bem evidente a diferença na disponibilidade de canais entre regiões mais e menos povoadas. Os alcances calculados para as antenas se assemelharam a valores reais. Esse modelo foi definido como sendo padrão para a base.

No entanto, ao utilizar a constante com valores 3,5 e 4, o modelo apresentou resultados muito pouco sensíveis. Certamente, os alcances calculados são muito menores que os reais. Isso fica visível nos mapas de calor. A região com baixa disponibilidade de canais se restringe a área próximas de grupos de antenas. Se utilizado por uma BDWS esse modelo pode acabar fazendo com que os dispositivos se encontrem em posição de interferência. Por conta desses motivos, a constante não pode assumir esses valores.

4.3.2 Conclusão

Entre os modelos implementados, o Log Distance, com $\alpha = 3$ apresentou os melhores resultados. Como esperado, regiões metropolitanas estão com um grau de disponibilidade de canais muito baixo, já que grande maioria já está sendo utilizado por um dispositivo primário, enquanto que áreas campestres apresentam um grau de disponibilidade maior. Os resultados obtidos confirmam a escolha desse modelo como padrão para a BDWS desenvolvida.

Embora o modelo tenha sido considerado como o ideal entre os estudados não se deve tomá-lo como o mais completo. É ideal que seja implementado um modelo mais

completo, que utilize informações como o relevo do terreno e densidade de estruturas, de forma a sempre calcular o alcance dos dispositivos primário com maior precisão.

De maneira geral foi verificado que o Rio de Janeiro é uma área propícia para a implantação de rádios cognitivos. Embora as regiões metropolitanas já estejam densamente utilizadas, foi identificado que em áreas menos povoadas a quantidade de canais disponível é alta.

Capítulo 5

Conclusão e Trabalhos Futuros

5.1 Conclusão

O sistema desenvolvido consegue atingir o objetivo proposto. Seu formato modular facilita a modificação de alguma parte específica do software, o que facilita futuros trabalhos e aprimoramentos. Além disso, sua API permite a busca por diversas informações sobre dispositivos primários que pode ser utilizada em outras pesquisas. Como é possível implementar novos modelos de propagação, e buscar dados desses modelos aplicados a dispositivos primários reais, novas possibilidades de pesquisa são abertas.

5.2 Trabalhos Futuros

O objetivo desse trabalho foi estabelecer um marco inicial para futuros trabalhos no campo de BDWS. No entanto, para poder criar de fato uma base de dados confiável, para prover um serviço para dispositivos reais, ainda existe muito trabalho a ser feito.

A seguir serão enumeradas, em ordem de importância, maneiras de dar continuidade a esse trabalho.

5.2.1 Parceria com a ANATEL

Quem define as regras para o uso do espectro de frequências, no Brasil, é a ANATEL. Assim, a BDWS desenvolvida deve receber um aval dela para funcionar. Além disso, é de interesse da agência reguladora saber como o espectro está sendo utilizado.

Como já foi ressaltado pelos outros capítulos, a informação sobre dispositivos primários existentes deve ser confiável. Além disso, a base desenvolvida deve levar em considerações regras impostas pela agência reguladora. Suponha que a ANATEL reserve uma determinada banda para microfones sem fio, ou até mesmo bloqueie o uso de um determinado canal de frequência.

Para que a base seja devidamente alterada com as regras e reservas feitas pela agência, um canal de comunicação deve ser estabelecido. A base deve poder buscar periodicamente informações sobre os dispositivos com a ANATEL, bem como receber mensagens sobre novas regras que devem ser utilizadas no momento do cálculo de disponibilidade de canais.

Para poder atender a esses pontos é de extrema importância firmar uma parceria com a ANATEL. Essa parceria é essencial para o funcionamento de um BDWS.

É proposto que nessa parceria:

- Seja desenvolvido um Web Service para disponibilizar todas as informações disponíveis sobre os dispositivos primários. Dessa forma a carga não precisará ser manual.
- Seja realizado uma revisão dos dados já presentes no portal da agência reguladora. É importante garantir que as informações lá presentes estejam corretas. Se possível, as informações de latitude e longitude dos dispositivos primários devem ser convertidas para um formato mais preciso.
- Seja acordado, com a ANATEL, um método para que ela informe à base alterações emergenciais que devem ser feitas, bem como a criação de regras para o cálculo de disponibilidade.

Um Web Service que disponibilize as informações da ANATEL é de interesse não só da base mas também de qualquer outro grupo de pesquisa que deseje realizar trabalhos utilizando valores reais.

O método de comunicação da ANATEL com a base poderia ser outro Web Service. Ele, porém, terá de ser protegido, já que deve ser acessado apenas pela agência.

Podem existir algumas considerações e mudanças propostas pela ANATEL ao sistema. Tais mudanças devem ser implementadas.

5.2.2 Cálculo de Disponibilidade

Na base desenvolvida, os cálculos realizados para determinar o alcance das antenas, e por consequência a disponibilidade de canais, foi bastante simplificado. Foram desenvolvidos apenas três modelos de propagação, e algumas considerações foram feitas para simplificar o seu cálculo.

Uma vez que a base esteja povoada com informações confiáveis, será necessário aprimorar o método utilizado para calcular o alcance das antenas. Para isso deve ser implementado um modelo de propagação adequado. Ele deve ser mais complexo e levar em consideração, entre outros fatores, o terreno ao redor das antenas.

Além disso, como um dos objetivos da base é prover uma base para pesquisas, outros modelos de propagação também devem ser implementados de forma a aumentar as possibilidades de pesquisa.

Verificação de resultados

É interessante comparar os resultados da base com os valores reais. Para isso, seria necessário um trabalho de medição de canais disponíveis. Isso seria feito com um trabalho de campo, escolhendo alguns pontos do estado do Rio de Janeiro e comparar, os canais que realmente estão disponíveis, como indicado pela medição, e os canais indicados pela base como disponíveis.

Esse tipo de trabalho é importante já que confirma a eficácia da base ao compará-la com valores coletados em campo.

5.2.3 PAWS

Embora o PAWS ainda esteja em processo de desenvolvimento, já é possível identificar como é o seu modo de funcionamento, bem como as regras para sua implantação. É interessante que a base venha a aderir aos padrões internacionais de BDWS, que está sendo proposto nesse protocolo.

Assim sendo, deve-se, futuramente, implantar o PAWS no sistema desenvolvido. Devido à modularidade do sistema, esse protocolo seria facilmente implementado, já que ele define apenas o modo de comunicação entre base e dispositivos clientes.

5.2.4 Geolocalização

A base de dados recebe como informação de localização a latitude e longitude do dispositivo. Poderia ser utilizado, também, um serviço de geolocalização de forma que um cliente pudesse informar seu endereço ao invés das coordenadas geográficas. Isso seria possível com o uso de um serviço de geolocalização, que transforma um endereço físico em um par de coordenadas polares. Esse serviço é chamado geocodificação.

Existem alguns serviços disponíveis para o uso pela internet, como é o caso do GeoNames mencionado no capítulo anterior. Entretanto, estes serviços apresentam um limite de uso diário, que pode ser estendido na compra de um pacote. De forma a garantir um uso ininterrupto desse tipo de serviço, a geolocalização poderia ser implementada no próprio sistema.

Existem algumas soluções open source que poderiam ser utilizadas como o Gigraphy [30].

Referências Bibliográficas

- [1] INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS (IEEE). *Functional Requirements for IEEE 802.22 WRAN Standard, 802.22/05-0007r46*. Relatório técnico, set. 2005.
- [2] FEDERAL COMMUNICATIONS COMMISSION (FCC). “FCC ONLINE TABLE OF FREQUENCY ALLOCATIONS”. . <http://transition.fcc.gov/oet/spectrum/table/fcctable.pdf>, . Acessado em fevereiro de 2013.
- [3] FEDERAL COMMUNICATIONS COMMISSION (FCC). “Unlicensed Wireless Microphone Registration System Launched”. . <http://www.fcc.gov/document/unlicensed-wireless-microphone-registration-system-launched>, . Acessado em fevereiro de 2013.
- [4] FEDERAL COMMUNICATIONS COMMISSION (FCC). “TV White Spaces Databases and Database Administrators”. . <http://transition.fcc.gov/bureaus/oet/whitespace/WhiteSpacesDAGuide.pdf>, . Acessado em fevereiro de 2013.
- [5] FEDERAL COMMUNICATIONS COMMISSION (FCC). “White Space Database Administration”. . <http://www.fcc.gov/encyclopedia/white-space-database-administration>, . Acessado em fevereiro de 2013.
- [6] FEDERAL COMMUNICATIONS COMMISSION (FCC). “Chairman Announces Approval of White Spaces Database Spectrum Bridge”. . <http://www.fcc.gov/document/chairman-announces-approval-white-spaces-database-spectrum-bridge>, . Acessado em fevereiro de 2013.
- [7] “Spectrum Bridge”. . <http://whitespaces.spectrumbridge.com>. Acessado em fevereiro de 2013.

- [8] “Tower Coverage”. . <http://www.towercoverage.com/>. Acessado em março de 2013.
- [9] “Radio Mobile”. . <http://www.cplus.org/rmw/english1.htm>. Acessado em março de 2013.
- [10] HUFFORD, G., LONGLEY, A., KISSICK, W. *A Guide to the Use of the ITS Irregular Terrain Model in the Area Prediction Mode*. Relatório técnico, abr. 1982.
- [11] INTERNET ENGINEERING TASK FORCE (IETF). “Protocol to Access Spectrum Database”. . <http://tools.ietf.org/html/draft-ietf-paws-protocol-03>, fev. 2013. Rascunho em desenvolvimento.
- [12] “JSON-RPC 2.0”. . <http://www.jsonrpc.org/specification>. Acessado em fevereiro de 2013.
- [13] INTERNET ENGINEERING TASK FORCE (IETF). “HTTP Over TLS”. . <http://tools.ietf.org/html/rfc2818>, mai. 20.
- [14] INTERNET ENGINEERING TASK FORCE (IETF). “Protocol to Access White Space (PAWS) Database: Use Cases and Requirements”. . <http://tools.ietf.org/html/draft-ietf-paws-problem-stmt-usecases-rqmts-13>, fev. 2013. Rascunho em desenvolvimento.
- [15] INTERNET ENGINEERING TASK FORCE (IETF). “PAWS Database Discovery”. . <http://www.ietf.org/id/draft-wei-paws-database-discovery-00.txt>, fev. 2013. Rascunho em desenvolvimento.
- [16] “Restlet”. . <http://restlet.org/>. Acessado em fevereiro de 2013.
- [17] “Eclipse”. . <http://www.eclipse.org/>. Acessado em fevereiro de 2013.
- [18] ORACLE. “Oracle VM BOX”. . <http://www.oracle.com/technetwork/server-storage/virtualbox/downloads/index.html>. Acessado em fevereiro de 2013.
- [19] COOPER, J. W. *Java design patterns*. Addison-Wesley Professional, 2000.
- [20] AGÊNCIA NACIONAL DE TELECOMUNICAÇÕES (ANATEL). “Sistema de Informação dos Serviços de Comunicação de Massa”. . <http://sistemas.anatel.gov.br/siscom//ConsPlanoBasico/>

ListaArquivos.asp?SISQSm modulo=14611. Acessado em fevereiro de 2013.

- [21] GOOGLE. “Google Elevation API”. . <https://developers.google.com/maps/documentation/elevation/?hl=pt-br>, . Acessado em fevereiro de 2013.
- [22] GOOGLE. “Keyhole Markup Language”. . <https://developers.google.com/kml/documentation/?hl=pt-BR>, . Acessado em fevereiro de 2013.
- [23] GOOGLE. “Google Earth”. . <http://www.google.com/earth/index.html>, . Acessado em fevereiro de 2013.
- [24] ANATEL. “Consulta de Canais”. . <http://sistemas.anatel.gov.br/srd/TelaListagem.asp?PagSRD=Canal&Op=5&SISQSm modulo=34451>. Acessado em fevereiro de 2013.
- [25] RAPPAPORT, T. S. *Wireless Communications: Principles and Practice*. 2 ed. , Prentice Hall, 2001. ISBN: 9780130422323.
- [26] E SAM RUBY, L. R. *RESTful Web Services*. 1 ed. , O’Reilly, 2007. ISBN: 9780596528260.
- [27] INTERNET ENGINEERING TASK FORCE (IETF). “Hypertext Transfer Protocol – HTTP/1.1”. . <http://www.ietf.org/rfc/rfc2616.txt>, jun. 1999.
- [28] “GeoNames”. . <http://www.geonames.org/>. Acessado em fevereiro de 2013.
- [29] MYHEATMAP. “myHeatmap”. . <http://www.myheatmap.com/>. Acessado em fevereiro de 2013.
- [30] “Gisgraphy”. . <http://www.gisgraphy.com/>. Acessado em fevereiro de 2013.