

**ERIC RODRIGUES PIRES  
MATEUS NAKAJO DE MENDONÇA**

**SISTEMA WEB PARA INSTALAÇÃO DE ERBS**

São Paulo  
2018

**ERIC RODRIGUES PIRES  
MATEUS NAKAJO DE MENDONÇA**

## **SISTEMA WEB PARA INSTALAÇÃO DE ERBS**

Trabalho apresentado à Escola Politécnica  
da Universidade de São Paulo para obtenção  
do Título de Engenheiro de Computação.

São Paulo  
2018

**ERIC RODRIGUES PIRES  
MATEUS NAKAJO DE MENDONÇA**

## **SISTEMA WEB PARA INSTALAÇÃO DE ERBS**

Trabalho apresentado à Escola Politécnica  
da Universidade de São Paulo para obtenção  
do Título de Engenheiro de Computação.

Orientador:

Bruno de Carvalho Albertini

São Paulo  
2018

Dedicatória

# AGRADECIMENTOS

Thanks...

*“Epígrafe”*

-- Autor

# RESUMO

Este projeto de formatura tem como objetivo criar um sistema capaz de calcular posições para a instalação de Estações Radiobase (ERBs) de forma que a cobertura da rede de ERBs seja máxima. A partir da região dada como entrada, o sistema obterá seus dados geográficos através de um Sistema de Informações Geográficas (SIG) e utilizará programação específica para a otimização da posição de instalação. Para interface com o usuário do sistema, criaremos uma aplicação Web responsiva que permita selecionar a região na qual se pretende instalar uma ERB e mostra as posições ideais para instalação.

**Palavras-Chave** – Estações Radiobase, Otimização, Sistema de Informações Geográficas, Aplicação Web.

# ABSTRACT

This term paper intends to achieve a system capable of calculating the position to install cellular Base Stations (BS) so that we maximize the coverage network. From a given input region, the system will collect geographic data through a Geographical Information System (GIS) and utilize specific programming to optimize the placement position. For interfacing with the system user, we will develop a responsive Web application that allows the selection of a region on which we intended to place a BS, and show the ideal points for installation.

**Keywords** – Base Stations, Optimization, Geographical Information System, Web Application.



## LISTA DE FIGURAS

1	Diagrama de Gantt. . . . .	21
2	Árvore de pré-requisitos do sistema. . . . .	22
3	Mapa de ERBs sem clusterização. . . . .	33
4	Mapa de ERBs com clusterização. . . . .	34

## LISTA DE TABELAS

# LISTA DE SÍMBOLOS

**ERB:** Estação Radiobase

**SIG:** Sistema de Informações Geográficas

**TCC:** Trabalho de Conclusão de Curso

# SUMÁRIO

<b>1</b>	<b>Introdução</b>	<b>12</b>
1.1	Objetivo . . . . .	12
1.1.1	Sistema de Informação Geográfica . . . . .	13
1.1.2	Interface Web . . . . .	13
1.2	Motivação . . . . .	13
1.3	Justificativa . . . . .	14
1.4	Organização do Trabalho . . . . .	14
<b>2</b>	<b>Aspectos Conceituais</b>	<b>16</b>
2.1	Algoritmos Avaliados . . . . .	16
2.2	Método Taguchi . . . . .	16
2.2.1	Array Ortogonal . . . . .	16
2.2.2	Mapear cada nível a um valor . . . . .	17
2.2.3	Sinal ruído . . . . .	17
2.2.4	Iteração . . . . .	17
2.2.5	Critério de parada . . . . .	17
<b>3</b>	<b>Tecnologias Utilizadas</b>	<b>18</b>
3.1	Sistema Web . . . . .	18
3.2	Bases de Dados Utilizadas . . . . .	18
<b>4</b>	<b>Metodologia do Trabalho</b>	<b>20</b>
<b>5</b>	<b>Especificação de Requisitos do Sistema</b>	<b>22</b>
5.1	Atores . . . . .	23

5.2	Requisitos funcionais . . . . .	23
5.3	Requisitos não-funcionais . . . . .	24
5.4	Descrição dos casos de uso . . . . .	24
<b>6</b>	<b>Projeto e Implementação</b>	<b>33</b>
6.1	Backend . . . . .	33
6.1.1	Clusterização . . . . .	33
6.2	Métodos Numéricos . . . . .	36
6.2.1	Método Taguchi . . . . .	36
6.2.2	Método Basinhopping . . . . .	36
6.2.3	Método SQSQP . . . . .	36
6.3	Funções de Otimização . . . . .	37
6.4	Frontend . . . . .	37
6.4.1	Projeto de Telas . . . . .	37
6.4.2	Implementação com Vuetify . . . . .	37
<b>7</b>	<b>Testes e Avaliação</b>	<b>38</b>
<b>8</b>	<b>Considerações Finais</b>	<b>39</b>
8.1	Conclusões do Projeto de Formatura . . . . .	39
8.2	Contribuições . . . . .	39
8.3	Perspectivas de Continuidade . . . . .	39
	<b>Referências</b>	<b>40</b>

# 1 INTRODUÇÃO

Na revolução da informação em que vivemos hoje, em que cada vez mais pessoas estão conectadas à rede, o acesso à Internet tem se tornado cada vez mais essencial no dia-a-dia, até mesmo a populações consideradas isoladas. Empresas bem conhecidas, como Vivo e Claro, vêm se empenhando para garantir melhor acesso a mais pessoas, mas se deparam com problemas de engenharia nesta tarefa.

A extensão territorial e a densidade demográfica desigual do Brasil são dois dentre vários fatores que tornam problemas de telecomunicação mais complexos. A dimensão deste problema gera um grande potencial de mercado para empresas terceirizadas, voltadas à instalação de Estações Radiobase (ERBs) para compartilhamento ou aluguel de células telefônicas às grandes empresas de telecomunicação. Dessa forma, há demanda do mercado por ferramentas que simplifiquem e/ou automatizem a tarefa de estudo de localização de ERBs.

## 1.1 Objetivo

O objetivo deste projeto de formatura é criar um sistema que permita calcular posições para a instalação de antenas de telefonia de forma a maximizar o alcance delas. Com esse fim, levaremos em conta dados geográficos para realizarmos os cálculos.

Também é de grande importância que tal sistema tenha uma interface prática para os usuários. Portanto, uma interface web que apresente os dados requisitados é essencial para o projeto.

Outras possíveis ramificações do projeto para showcase ao público geral, que não é o público-alvo, é a localização de antenas a partir do próprio celular do usuário, e a estimativa de posição do dispositivo pelas antenas encontradas.

### 1.1.1 Sistema de Informação Geográfica

Um SIG (Sistema de Informação Geográfica) é um sistema computacional capaz de obter, gravar, gerir, analisar e visualizar dados geográficos. Seu uso permite tomar decisões, analisar estatísticas e resolver problemas de otimização a partir de dados geográficos. O SIG pode ser usado tanto em lojas de varejo para decidir onde abrir uma nova filial, como em rastrear padrões de migração, controle e o monitoramento do desmatamento, planejamento urbano, etc.

No nosso projeto, usaremos um software SIG para gravar e exibir a posição de ERBs (Estações Radiobase) atuais, o relevo e os consumidores atingidos pela rede de ERBs. Com essas informações, determinaremos as posições ótimas de ERBs de modo a maximizar a área de cobertura do sistema de telefonia. Para tanto, aplicaremos técnicas de programação linear, uma vez que estamos diante de um problema de otimização cuja função a ser otimizada é linear em relação às variáveis de entrada.

### 1.1.2 Interface Web

Para interação com o usuário, criaremos um front-end de uma aplicação Web que permita selecionar a região na qual se pretende instalar alguma ERB. Esta interface se comunicará com o back-end do SIG, para obter e calcular os dados desejados.

O design deverá ser responsivo, podendo ser utilizado em plataformas mobile ou desktop, e simples, com opções simples para apenas verificar a posição ótima de instalação de antenas em determinada área escolhida pelo usuário. Para isso, a interface deverá exibir um mapa, como por exemplo o da plataforma OpenStreetMap, com as informações do SIG, que permita ao usuário selecionar uma área desejada. Os dados serão calculados no back-end e exibidos ao usuário na tela. Para isso, será necessário desenvolver um front-end possivelmente dinâmico.

## 1.2 Motivação

Com uma análise preliminar do setor, verificamos o mercado de instalação e aluguel de torres telefônicas no Brasil para comparar as tecnologias utilizadas em softwares ou pesquisas de de ERBs. Há várias técnicas empregadas, desde programação não-linear a algoritmos evolutivos, algoritmos de polinização a programação inteira mista. Será feita uma comparação das tecnologias para verificar a que mais se adequa ao nosso caso de uso.

Também pesquisamos serviços similares da concorrência. Um dos produtos encontrados, chamado Atoll, é um software de planejamento de células e posições de ERBs, similar ao que desejamos desenvolver, porém com funcionalidades estendidas como manutenção e melhoria de locais pré-estabelecidos, e parâmetros avançados de especificação das antenas, além de módulos para outras tecnologias de telecomunicação como Wi-Fi [1]. Porém, a ferramenta parece muito voltada à instalação urbana e análise de infra-estrutura pré-existente, sem foco em uma eventual expansão. Por isso, vemos como que há necessidade do mercado por uma ferramenta voltada à ampliação de uma rede de ERBs.

### 1.3 Justificativa

Sobre potenciais clientes, verificamos a existência de empresas no Brasil para localizar antenas, alugar terrenos para a instalação de antenas ou alugar antenas para empresas de telecomunicação. A maior parte destas empresas foca em um contexto urbano, enquanto que há interesse das empresas de telecomunicação e dos governos estaduais em expansão em áreas rurais.

MyTower é um portal de locação e venda de imóveis para operadoras de telecomunicação [2]. Ele permite que o usuário cadastre seu imóvel e o anuncie para as operadoras após aprovação. O portal então faz a intermediação entre o anunciante e a operadora.

A Skysites é uma empresa que oferece soluções na área da infraestrutura de telecomunicação [3]. Ela gere um portfólio de sítios para instalação de equipamentos de telecomunicação (torres, smallcells, rooftops, etc), além de prover soluções customizadas para empresas de telecomunicação e compartilhar torres entre diferentes empresas. Outros serviços são redes para cobertura *indoor* e pequenas ERBs para melhorar a cobertura em ambiente urbano, as *small cells*.

### 1.4 Organização do Trabalho

No capítulo “Introdução” deste trabalho, definimos a motivação da realização deste sistema e o que buscamos alcançar neste projeto.

No capítulo “Aspectos Conceituais”, será realizada a contextualização dos conceitos empregados na área de aplicação e a revisão da literatura de base.

No capítulo “Tecnologias Utilizadas”, listaremos as ferramentas, algoritmos e dados necessários para o desenvolvimento do sistema deste trabalho.



No capítulo “Metodologia do Trabalho”, definiremos os processos e fases no desenvolvimento de funcionalidades deste sistema, como concepção, estudo, projeto, implementação e testes.

No capítulo “Especificação de Requisitos do Sistema”, definiremos os requisitos do nosso sistema.

## 2 ASPECTOS CONCEITUAIS

### 2.1 Algoritmos Avaliados

Em consulta à literatura pré-existente sobre o problema de otimização de instalação de ERBs, nos deparamos com várias abordagens distintas para o mesmo problema, em diferentes níveis de abstração.

A princípio, nós nos voltamos a duas alternativas: LEE et al. (2015) [9] utiliza conceitos básicos de telecomunicações, através de uma fórmula para calcular a satisfação dos usuários do sistema a partir de medidas de qualidade de banda, se baseando em um algoritmo evolutivo para otimizar a cobertura da rede. Já KARULKAR & OH (2016) [11] se baseia em uma abordagem de limites geográficos impostos no processo de projeto de antenas, utilizando programação não-linear para identificar a posição ótima.

### 2.2 Método Taguchi

O Método Taguchi é um método estatístico que tem como objetivo diminuir o número de experimentos necessários para se ajustar um conjunto de parâmetros e assim se obter um aumento de performance no sistema. No trabalho em questão, os parâmetros a serem otimizados são as coordenadas (latitude e longitude) de cada ERB a ser instalada. Como tais parâmetros são contínuos, criamos níveis associados, que correspondem (em cada parâmetro) a valores distribuídos uniformemente dentro do intervalo avaliado.

#### 2.2.1 Array Ortogonal

O primeiro passo do Método Taguchi é escolher um array ortogonal, que possua o número de colunas iguais ao número de parâmetros analisados, e possua a quantidade de níveis adequada. Array ortogonal é uma tabela, na qual existe um número  $t$  (chamado de força do array ortogonal) para o qual, qualquer seleção de  $t$  colunas na tabela, todas

as t-uplas aparecem o mesmo número de vezes.

### 2.2.2 Mapear cada nível a um valor

Em seguida, mapeamos cada nível a um valor do parâmetro. Sejam  $min_t$  e  $max_t$  os valores máximo e mínimo do intervalo em estudo de um parâmetro  $x_t$ , e  $s$  o número de níveis escolhido.

$$V_t^{(m)} = \frac{min_t + max_t}{2}$$

Definimos:

$$\beta_t^{(m)} = \frac{max_t - min_t}{s + 1}$$

Na m-ésima iteração, a função de mapeamento é a seguinte:

$$f_t^m(l) = \begin{cases} V_t^{(m)} - (s/2 - l) \cdot \beta_t^{(m)}, & 1 \leq l \leq [s/2] - 1 \\ V_t^{(m)}, & l = [s/2] \\ V_t^{(m)} + (l - s/2) \cdot \beta_t^{(m)}, & [s/2] + 1 \leq l \leq s \end{cases}$$

Onde  $s$  é a quantidade de níveis escolhida,  $l$  é o número do nível.

### 2.2.3 Sinal ruído

Depois disso, mapeamos cada valor da imagem da função objetivo para a razão sinal ruído. A fórmula é:

$$SN_i = 10 \cdot \log_{10}(y_i^2) [dB]$$

Depois, calculamos o valor médio de  $SN$  para cada parâmetro para cada nível. Para cada parâmetro  $x_t$ , o nível que tiver maior  $SN$  médio, é denotado  $V_t^{(best,m)}$

### 2.2.4 Iteração

Em seguida, verificamos o critério de parada. Se ele não estiver satisfeito, os melhores valores de cada parâmetro são usados como valores centrais para a próxima iteração. Além disso, o valor de  $\beta_t$  é reduzido de um fator  $\epsilon < 1$ .

$$\beta_t^{(m+1)} = \epsilon \beta_t^{(m)}$$

### 2.2.5 Critério de parada

## 3 TECNOLOGIAS UTILIZADAS

### 3.1 Sistema Web

No projeto, para facilidade de desenvolvimento, utilizaremos o framework Django, escrito em Python. A integração de single-page com o back-end deverá ser feita com o módulo de API REST do Django.

O Django possui funcionalidades de SIG pelo módulo GeoDjango, que utiliza como banco de dados o PostGIS (baseado em Postgres). Serão armazenados dados públicos de localização de ERBs, relevo e densidade populacional. Ele será também responsável pelos cálculos realizados para a localização de novas antenas.

Para interação com o usuário por um mapa interativo, utilizaremos a biblioteca Leaflet, escrita em JavaScript. Ela se comunicará aos dados pela API REST a ser desenvolvida, tanto para requisições quanto para exibições.

Optamos também pelo framework JavaScript conhecido por React, com a biblioteca Redux, para realizar o controle *single-page app* do nosso sistema, e a interface gráfica Django Material para elaborar as telas do nosso sistema.

### 3.2 Bases de Dados Utilizadas

Utilizamos o Mapa de ERBs Brasil presente no portal Telebrasil [4]. Essa base contém uma lista de ERBs do Brasil de novembro de 2017, com informação de operadora, endereço, e posição geográfica de cada ERB. Essas informações são essenciais para o cálculo da posição ótima da ERB para maximizar a cobertura da célula.

Utilizamos também o OpenCellID, da empresa Unwired Labs [5]. Essa base contém uma lista de ERBs do mundo inteiro, com o CGI de cada ERB. Os dados foram obtidos através da colaboração de usuários do aplicativo LocationAPI da Unwired Labs. O LocationAPI trata-se de um serviço de geolocalização que não depende de GPS. Dessa forma,

com a base da OpenCellID, podemos estimar a posição de um celular a partir das ERBs as quais ele está conectado.

Outra base de dados em estudo foi o Google Earth Engine [6], uma API específica para dados geográficos públicos do Google, como relevo e densidade populacional. Devido à extensão destes dados, e à impraticidade de armazenamento em banco próprio, será estudada a possibilidade de uma dependência desta base através de sua API.

Mais duas bases de dados que serão utilizadas no projeto são: a G-Econ [7], da Universidade de Yale, que apresenta os dados de paridade de poder de comparar geograficamente; e o SIMET-NIC [8], com o acesso e qualidade da Internet no Brasil. Acreditamos que estas duas bases de dados, em conjunto com os anteriores, permitirão uma análise aprofundada de parâmetros ótimos para a instalação de novas antenas.

## 4 METODOLOGIA DO TRABALHO

Em uma primeira etapa, definimos com o orientador a proposta e o escopo deste projeto, propondo a pesquisa a ser realizada tanto da perspectiva de implementações quanto de requisitos necessários. Levantados tais requisitos no capítulo “Especificação de Requisitos do Sistema”, a próxima etapa se baseou em projetar a realização de cada um destes requisitos de acordo com as prioridades definidas, isto é, iniciar um processo de decisões definitivas para o andamento do trabalho.

Feitas estas decisões, finalmente, realizou-se a prototipagem das nossas funcionalidades, dividida entre os dois integrantes do grupo de forma paralela para permitir o andamento de diferentes partes do sistema, com reuniões entre os integrantes e o orientador para avaliar o progresso. Após esta análise preliminar, iniciou-se o desenvolvimento definitivo do sistema na especificação final, etapa que levará mais tempo neste projeto.

Por fim, antes de realizar a apresentação do trabalho final, será realizada uma bateria de testes para verificar o funcionamento correto do sistema após o desenvolvimento, sem a elaboração de novas funcionalidades, garantindo que estamos dentro das nossas previsões de projeto.

Desta forma, nos organizamos para realizar a implementação do sistema, separando as tarefas a serem realizadas nas categorias: pesquisa, projeto, e apresentação. Utilizamos o cronograma oficial de TCC para elaborar um diagrama de Gantt na Figura 1 com a sequência de tarefas a serem divididas pelo grupo ao longo do ano para cada semana. Indicamos também o período extra-escolar em colunas azuis, aonde haverá desenvolvimento do sistema de forma mais lenta e sem cronograma oficial.

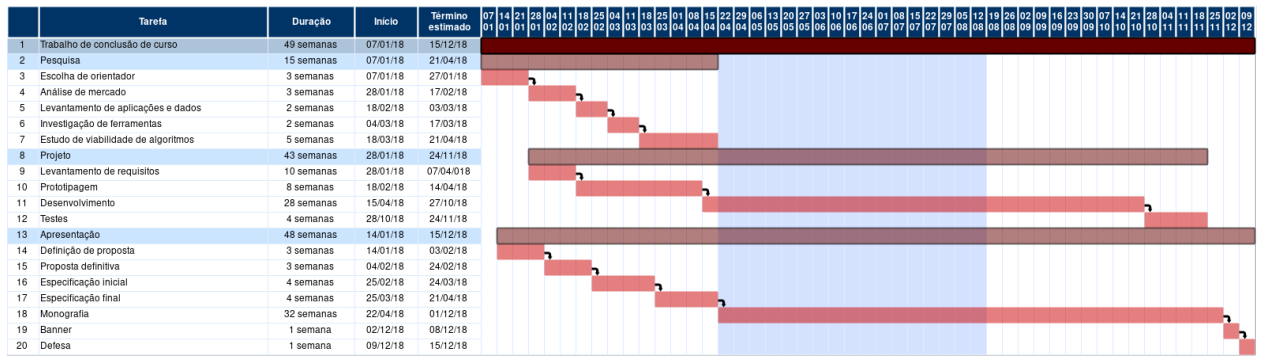


Figura 1: Diagrama de Gantt.

## 5 ESPECIFICAÇÃO DE REQUISITOS DO SISTEMA

Para definir os requisitos do nosso sistema, foi elaborada uma árvore de pré-requisitos, listando a prioridade total dada a cada componente do sistema final na Figura 2.

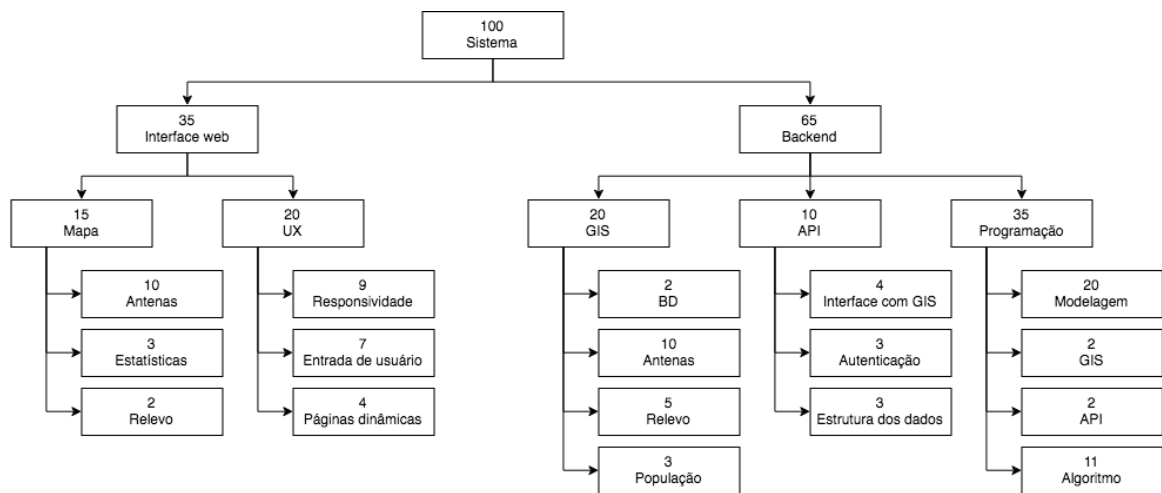


Figura 2: Árvore de pré-requisitos do sistema.

Como evidenciado pela figura, a ênfase deste projeto será no back-end, em especial na parte de modelagem e programação relacionadas ao cálculo de otimização da posição de antenas. As outras duas partes relevantes do *backend* tratam, respectivamente, do uso do banco de dados como SIG e da comunicação externa de dados via API.

Embora tenha uma ênfase menor, o front-end da interface web também será um requisito fundamental de projeto, separado na experiência do usuário e na visualização do mapa.

Definido o escopo, começamos a pensar, junto ao nosso orientador, sobre os requisitos do projeto. A estratégia utilizada foi o brainstorming, leitura de obras de referência, e pesquisa de softwares com escopo parecido. A ideia inicial era ter como ator principal do sistema o funcionário cadastrado de uma empresa de instalação de antenas, porém depois consideramos que seria complementar ao projeto considerar um usuário não-comercial,



assim como um usuário administrador. Nessa fase, perguntamo-nos como o nosso projeto poderia ajudar esses atores, qual informação ele deveria produzir. Esse processo nos levou à modelagem apresentada nas seções a seguir.

## 5.1 Atores

- Administrador
- Funcionário cadastrado de Empresa de instalação de antenas
- Usuário não-comercial

## 5.2 Requisitos funcionais

- Efetuar Login
- Efetuar Logout
- Cadastrar usuário no sistema
- Exibir mapa com ERBs
- Exibir mapa com cobertura celular estimada
- Exibir localização ideal para instalação de uma nova ERB
- Adicionar nova ERB
- Fazer login por OAuth2
- Acessar API
- Exibir ERBs às quais o celular do usuário está conectado
- Exibir mapa com qualidade do sinal por operadora
- Estimar geolocalização do usuário
- Listar usuários

## 5.3 Requisitos não-funcionais

- Funcionalidades e código bem documentados
- Interface acessível e simples
- Ser responsivo
- Ser dinâmico
- Ser rápido
- Ser seguro

## 5.4 Descrição dos casos de uso

**Caso de Uso 1:** Efetuar login no sistema.

**Descrição:** Este caso de uso descreve o processo de autenticação no sistema.

**Evento iniciador:** Usuário informa seu nome de usuário e senha.

**Atores:** Administrador, Funcionário ou Usuário não-comercial.

**Pré-condições:** Usuário entra na página de login.

**Sequência de Eventos:**

1. Usuário informa seu nome de usuário e sua senha.
2. Sistema autentica o usuário e senha.
3. Sistema exibe página inicial com opções correspondentes ao nível de privilégio do usuário.

**Pós-condições:** Usuário logado no sistema.

**Extensões:**

1. Usuário ou senha estão incorretos: sistema exibe mensagem de erro (passo 2).
2. Login por OAuth2: sistema recebe token de autenticação externa para o acesso (passo 1).

**Inclusões:** -

**Caso de Uso 2:** Efetuar logout no sistema.

**Descrição:** Este caso de uso descreve o processo de logout no sistema.

**Evento iniciador:** Usuário clica no botão de logout.

**Atores:** Administrador, Funcionário ou Usuário não-comercial.

**Pré-condições:** Usuário logado no sistema.

**Sequência de Eventos:**

1. Usuário clica no botão de logout.
2. Sistema exibe página inicial para usuários não-logados.

**Pós-condições:** Usuário não logado no sistema.

**Extensões:** -

**Inclusões:** -

**Caso de Uso 3:** Cadastrar usuário no sistema.

**Descrição:** Este caso de uso descreve o cadastro de usuários no sistema

**Evento iniciador:** Usuário clica no botão cadastrar usuário.

**Atores:** Administrador, Funcionário ou Usuário não-comercial.

**Pré-condições:** Usuário não logado no sistema.

**Sequência de Eventos:**

1. Usuário clica no botão de cadastrar usuário.
2. Sistema exibe página de cadastro.
3. Usuário digita nome, e-mail, senha e confirmação de senha.
4. Sistema valida dados e envia email de confirmação para usuário.

**Pós-condições:** Usuário cadastrado no sistema com confirmação de e-mail pendente.

**Extensões:**

1. Usuário já cadastrado: sistema exibe mensagem de erro (passo 4)
2. Senha não é igual a confirmação de senha: sistema exibe mensagem de erro (passo 4)

**Inclusões:** buscar usuário (passo 4)

**Caso de Uso 3.1:** Confirmação de email.

**Descrição:** Este caso de uso descreve o processo de confirmação de um e-mail.

**Evento iniciador:** Usuário acessa a url correspondente à confirmação de seu e-mail.

**Atores:** Administrador, Funcionário ou Usuário não-comercial.

**Pré-condições:** Usuário cadastrado no sistema com e-mail não confirmado.

**Sequência de Eventos:**

1. Usuário acessa a url correspondente à confirmação de seu e-mail.
2. Sistema confirma a validade do e-mail do usuário.
3. Sistema redireciona o usuário para página inicial.

**Extensões:** URL de confirmação inválida: sistema mostra uma mensagem de erro e redireciona o usuário para a página de login (passo 2).

**Inclusões:** -

**Caso de Uso 4:** Exibir mapa

**Descrição:** Este caso de uso descreve o processo de exibição de um mapa centrado no usuário.

**Evento iniciador:** Usuário requisita exibição de mapa.

**Pré-condições:** Usuário logado no sistema.

**Sequência de Eventos:**

1. Usuário requisita exibição de mapa.
2. Sistema pede que o usuário permita-o acessar sua geolocalização.
3. Usuário permite que sistema acesse sua geolocalização.
4. Sistema mostra um mapa centrado no usuário.

**Pós-condições:** Mapa com ERBs apresentado.

**Extensões:** Usuário não permite: sistema mostra mapa centrado numa localização padrão. (passo 4)

**Inclusões:** busca de mapa no OpenStreetMap (passo 4)

**Caso de Uso 5:** Exibir mapa com ERBs.

**Descrição:** Este caso de uso descreve a exibição das ERBs em um mapa.

**Evento iniciador:** Usuário clica na opção Mapa de ERBs.

**Atores:** Administrador, Funcionário.

**Pré-condições:** Usuário logado no sistema.

**Sequência de Eventos:**

1. Usuário clica na opção Mapa de ERBs.
2. Sistema busca ERBs na região do mapa.
3. Sistema exibe mapa com ERBs.

**Pós-condições:** Mapa com ERBs apresentado.

**Extensões:** -

**Inclusões:** Caso de uso Exibir mapa (passo 3)

**Caso de Uso 5.1:** Exibir mapa centrado em localização dada

**Descrição:** Este caso de uso descreve a exibição do mapa com ERBs centrado em uma localização dada pelo usuário.

**Evento iniciador:** Usuário insere a latitude e longitude e clica em buscar.

**Atores:** Administrador, Funcionário.

**Pré-condições:** Usuário logado no sistema e na página com mapa de ERBs.

**Sequência de Eventos:**

1. Usuário insere a latitude e longitude e clica em buscar.
2. Sistema mostra o mapa de ERBs centrado na localização dada.

**Pós-condições:** Mapa com ERBs centrado na localização dada apresentado.

**Extensões:** Latitude ou longitude inválida: sistema mostra uma mensagem de erro (passo 2)

**Inclusões:**

1. busca de mapa no OpenStreetMap (passo 2)
2. caso de uso Exibir mapa com ERBs (pré-condição)

**Caso de Uso 5.2:** Exibir ERBs no mapa por operadora.

**Descrição:** Este caso de uso descreve a exibição do mapa com ERBs para uma operadora definida pelo usuário.

**Evento iniciador:** Usuário seleciona uma operadora.

**Atores:** Administrador, Funcionário.

**Pré-condições:** Usuário logado no sistema e na página com mapa de ERBs.

**Sequência de Eventos:**

1. Usuário seleciona uma operadora.
2. Sistema mostra o mapa de ERBs da operadora dada pelo usuário.

**Pós-condições:** Mapa com ERBs da operadora dada.

**Extensões:** -

**Inclusões:** caso de uso Exibir mapa com ERBs (pré-condição)

**Caso de Uso 6:** Encontrar local de instalação de ERBs. **Descrição:** Este caso de uso descreve a exibição de posições otimizadas

para instalação de ERBs. **Evento iniciador:** Usuário clica na opção Calcular posição para instalação.

**Atores:** Administrador, Funcionário.

**Pré-condições:** Usuário logado no sistema e na página Mapa de ERBs.

**Sequência de Eventos:**

1. Usuário clica na opção Calcular posição para instalação.
2. Sistema exibe um formulário com campos: Quantidade de ERBs a serem instaladas e Parâmetros a serem considerados no cálculo.
3. Usuário preenche o formulário e clica em avançar.
4. Sistema solicita para o usuário selecionar a região de interesse.
5. Usuário seleciona a região de interesse.
6. Sistema informa que operação pode demorar alguns momentos.
7. Sistema indica os locais calculados no mapa.
8. Sistema mostra número de usuários atendidos por nova

**Pós-condições:** Mapa com posições otimizadas apresentado.

**Extensões:** Operação demora muito tempo: sistema informa que não foi possível calcular a posição (passo 7)

**Inclusões:** Caso de uso Exibir mapa com ERBs (pré-condição).

#### **Caso de Uso 7:** Adicionar nova ERB

**Descrição:** Este caso de uso descreve o processo de adição de novas ERBs na base de dados.

**Evento iniciador:** Usuário clica na opção de adicionar ERB.

**Atores:** Administrador, Funcionário.

**Pré-condições:** Usuário logado no sistema.

#### **Sequência de Eventos:**

1. Usuário clica na opção de Adicionar ERB e clica em um ponto no mapa.
2. Sistema exibe formulário com dados a serem cadastrados sobre a nova ERB.
3. Usuário preenche formulário e clica em salvar.
4. Sistema salva nova ERB.
5. Sistema exibe mapa com ERBs.

**Pós-condições:** Mapa com ERBs apresentado e nova ERB salva na base de dados.

**Extensões:** Dados inválidos: sistema exibe mensagem de erro e exibe formulário novamente (passo 4).

**Inclusões:** Caso de uso Exibir mapa com ERBs (pré-condição e passo 5).

#### **Caso de Uso 8:** Remover ERB

**Descrição:** Este caso de uso descreve o processo de remoção de ERBs da base de dados.

**Evento iniciador:** Usuário clica na opção de remover ERBs.

**Atores:** Administrador, Funcionário.

**Pré-condições:** Usuário logado no sistema.

#### **Sequência de Eventos:**

1. Usuário clica na opção de Remover ERBs e seleciona uma ERB no mapa.
2. Sistema pergunta se usuário realmente deseja remover ERB.
3. Usuário responde Sim.
4. Sistema remove ERB.

5. Sistema exibe mapa com ERBs.

**Pós-condições:** Mapa com ERBs apresentado e ERB escolhida pelo usuário removida da base de dados.

**Extensões:** -

**Inclusões:** Caso de uso Exibir mapa com ERBs (pré-condição e passo 5).

**Caso de Uso 9:** Exibir ERBs às quais o celular do usuário está conectado

**Descrição:** Este caso de uso descreve o processo de exibição de ERBs às quais o usuário está conectado

**Evento iniciador:** Usuário clica na opção Antenas conectadas.

**Atores:** Administrador, Funcionário ou Usuário não-comercial.

**Pré-condições:** Usuário logado em um celular, na página inicial do sistema.

**Sequência de Eventos:**

1. Usuário clica na opção Antenas conectadas.
2. Sistema exibe um mapa com as ERBs conectadas em destaque.

**Pós-condições:** Mapa com ERBs conectadas apresentado.

**Extensões:** Celular não está conectado a ERB nenhuma: sistema exibe mensagem de erro (passo 2).

**Inclusões:** Caso de uso Exibir mapa com ERBs (passo 2).

**Caso de Uso 10:** Exibir mapa com qualidade do sinal por operadora.

**Descrição:** Este caso de uso descreve o processo de exibição de qualidade de sinal por operadora.

**Evento iniciador:** Usuário clica na opção Qualidade de sinal

**Atores:** Administrador, Funcionário ou Usuário não-comercial.

**Pré-condições:** Usuário logado, na página inicial do sistema.

**Sequência de Eventos:**

1. Usuário clica na opção Qualidade de sinal.
2. Sistema exibe um mapa centrado na localização atual do usuário.
3. Usuário clica em um ponto do mapa.



4. Sistema exibe informações de qualidade de sinal por operadora.

**Pós-condições:** Informações de qualidade de sinal apresentadas.

**Extensões:** -

**Inclusões:** Caso de uso Exibir mapa (passo 2)

**Caso de Uso 11:** Estimar geolocalização do usuário.

**Descrição:** Este caso de uso descreve o processo de estimar a geolocalização do usuário a partir das antenas às quais ele está conectado.

**Evento iniciador:** Usuário clica na opção Estimar geolocalização.

**Atores:** Administrador, Funcionário ou Usuário não-comercial.

**Pré-condições:** Usuário logado em um celular, na página inicial do sistema.

**Sequência de Eventos:**

1. Usuário clica na opção Estimar geolocalização.
2. Sistema exibe um mapa centrado na localização estimada do usuário.
3. Sistema mostra as antenas aos quais o usuário está conectado em destaque.

**Pós-condições:** Localização estimada do usuário e antenas conectadas são mostradas na tela.

**Extensões:**

1. Número de ERBs conectadas são insuficientes para estimar posição exata: sistema exibe um mapa com a localização estimada do usuário representada por um círculo. (passo 2)
2. Usuário não está conectado a nenhuma ERB: sistema exibe mensagem de erro (passo 2)

**Inclusões:** Caso de uso Exibir mapa com ERBs (passo 2).

**Caso de Uso 12:** Listar usuários do sistema

**Descrição:** Este caso de uso descreve o processo de listar usuários do sistema.

**Evento iniciador:** Administrador clica na opção Listar Usuários.

**Atores:** Administrador

**Pré-condições:** Administrador logado no sistema, na página inicial.

**Sequência de Eventos:**

1. Administrador clica na opção Listar Usuários.
2. Sistema busca usuários e os exibe em uma lista.

**Pós-condições:** Lista de usuários apresentada.

**Extensões:** -

**Inclusões:** buscar usuários no sistema (passo 2)

**Caso de Uso 13:** Modificar papel de usuário.

**Descrição:** Este caso de uso descreve o processo de modificar papel de usuário.

**Evento iniciador:** Administrador seleciona a opção Funcionário no campo Papel do Usuário

**Atores:** Administrador

**Pré-condições:** Administrador logado no sistema, na página listar usuários.

**Sequência de Eventos:**

1. Administrador seleciona um usuário de interesse
2. Administrador seleciona a opção Funcionário no campo Papel do Usuário
3. Administrador clica em Aplicar
4. Sistema modifica usuário escolhido para ele ter papel de Funcionário

**Pós-condições:** Usuário escolhido tem papel de Funcionário

**Extensões:** -

**Inclusões:** Caso de Uso Listar usuário do sistema (pré-condição)

## 6 PROJETO E IMPLEMENTAÇÃO

### 6.1 Backend

#### 6.1.1 Clusterização

Um dos principais desafios na implementação foi realizar a exibição de múltiplas antenas como pontos em um mapa. Como muitas ERBs ficam muito próximas e, portanto, difíceis de serem visualizadas separadamente, utilizou-se um *plugin* próprio da biblioteca de mapa para a clusterização de pontos, facilitando a identificação da quantidade de antenas em uma região, com exibição variável com o zoom.

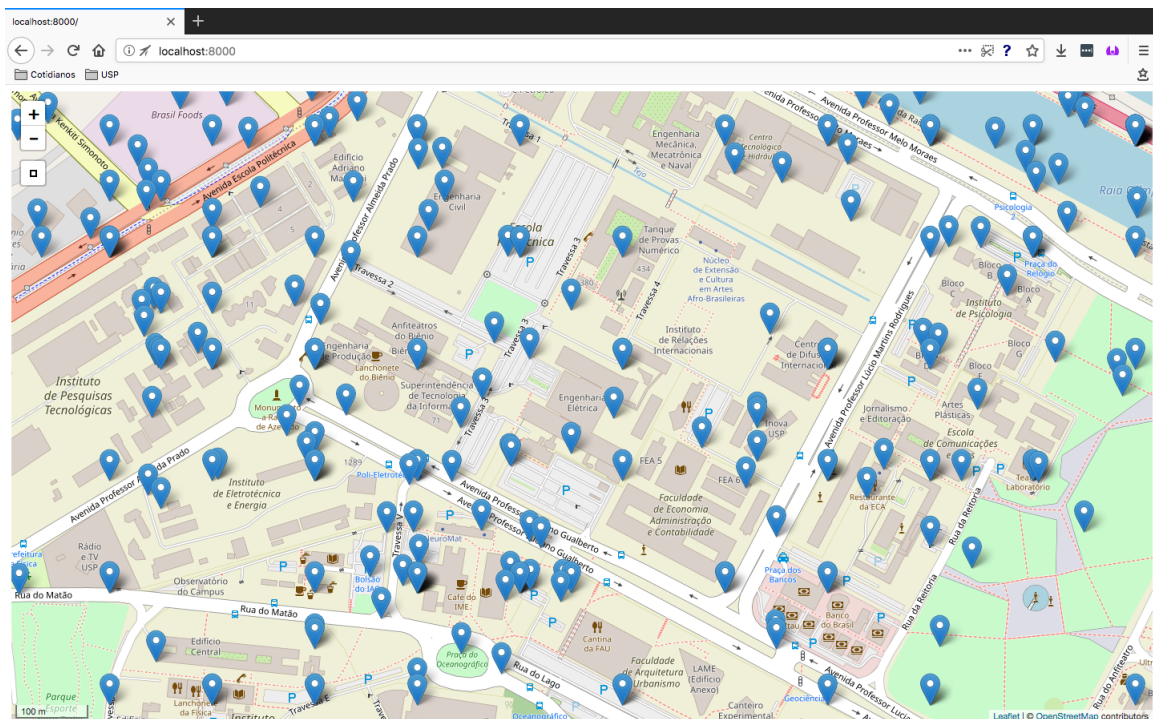


Figura 3: Mapa de ERBs sem clusterização.

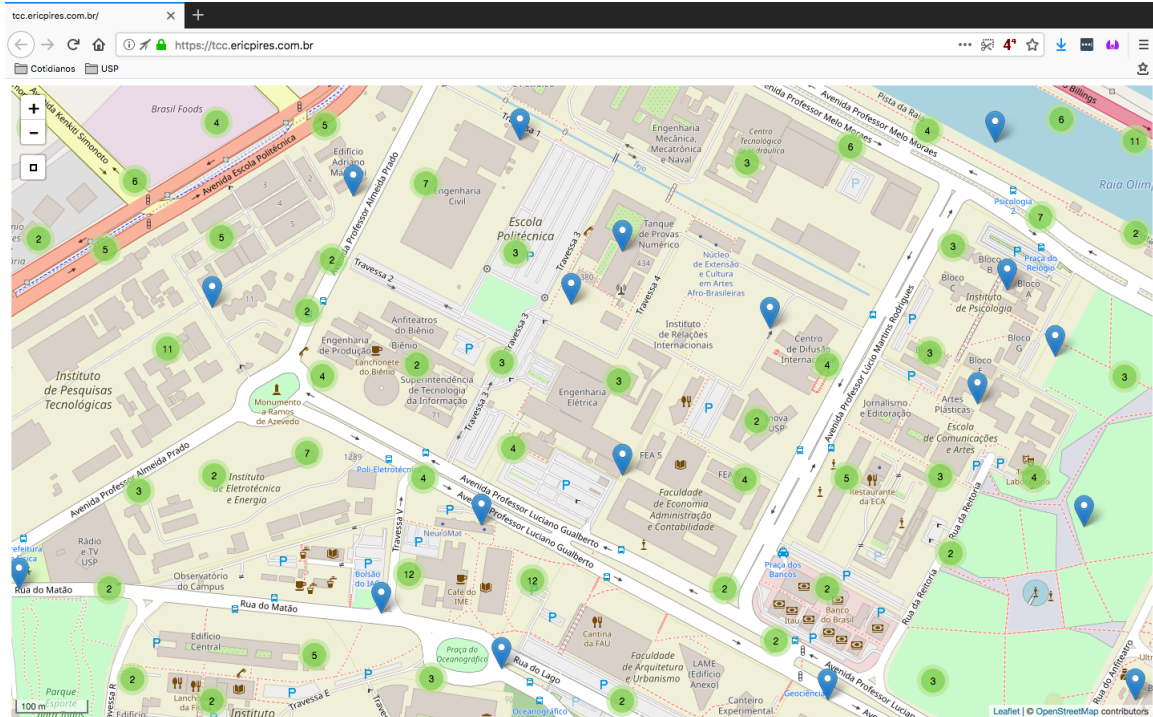


Figura 4: Mapa de ERBs com clusterização.

Com o uso desse plugin, a exibição de até cerca de 500 antenas na tela tornou-se possível. Porém, mais antenas do que isso causavam vários problemas relacionados a uma grande quantidade de dados, como demora na transmissão de dados, requisições intensivas ao servidor, e problemas de memória no navegador do cliente. Algumas medidas foram tomadas tanto no back-end quanto no front-end para tentar amenizar estes efeitos, mas sem gerar vantagens muito perceptíveis.

Dessa forma, resolveu-se utilizar algoritmos de clusterização no back-end, e então realizar a integração com o front-end. A primeira tentativa envolveu a utilização de uma biblioteca *open-source* específica para o GeoDjango, chamada “Anycluster” [12]. Embora tenha sido feita para uma configuração de servidor similar à utilizada no projeto, a instalação e documentação se mostraram confusas e incompletas. E mesmo com tentativas de ajuste manual no código, incluindo refatoração das partes não-essenciais da aplicação. Além disso, a análise do código indica que não há nenhuma forma de cache dos dados de clusterização, o que não resultaria em velocidades adequadas para uma aplicação de visualização em mapa.

Procurou-se então uma segunda alternativa de clusterização, com implementação própria para este projeto. A solução envolve utilizar o tiling de grade que o OpenStreetMap utiliza para carregar as imagens. A clusterização foi feita com fórmulas do

PostGIS específicas para clusterização de pontos. Para o cache dos clusters, utilizou o cache padrão do Django. Os resultados gerados pelo programa eram corretos, mas o uso alto de memória e o tempo para clusterizar antes de cache inviabilizam o uso no servidor. Foi realizado um aumento na capacidade da máquina do host, mas não houve ganhos consideráveis. Além disso, os dados de cluster ficam com aparência estranha, onde a separação em grids fica muito perceptível.

A terceira tentativa utilizou a mesma ideia anterior, porém para um dado nível de zoom, serão utilizados os clusters para um zoom mais próximo do mapa, portanto usando 4 vezes mais dados. Ainda houve alguns clusters estranhos na visualização em mapa, embora consideravelmente melhores no geral; e o uso de dados muito alto tanto em processamento quanto em rede torna uma utilização com muitos clientes inviável.

Em seguida, na quarta tentativa, optamos por salvar os clusters em banco de dados, onde são indexados por zoom como nas tentativas anteriores. Para acelerar o algoritmo de cálculo de clusters, utiliza-se os clusters de um zoom menor para gerar os de zooms maiores. Foram feitas algumas iterações com diferentes funções de clusterização do PostGIS para obter resultados melhores. A utilização de modelos ao invés de cache resultou em uma fácil integração com o front-end já existente, dado a facilidade do Django para lidar com modelos na API. Porém, os resultados obtidos foram inesperados, com clusters baseados na distância ponto-a-ponto ao invés de ao redor de um centro de massa. Dessa forma, lugares com muitas antenas lado-a-lado, em uma área com alta densidade de pontos como a cidade de São Paulo, aparecem como um único cluster, independente do zoom, deixando muitas áreas vazias.

A quinta tentativa se baseou nas demais, e também no conceito de geohash (anotação de cada região em um mapa com caracteres alfanuméricos, dividindo cada região sucessiva do globo em 16 partes). Foi selecionado uma precisão (i.e. um número de caracteres do geohash) diferente para cada zoom, seguindo-se como referência a biblioteca “django-geohash-cluster” [13], sendo que a precisão máxima é de 7 caracteres, que tem precisão de aproximadamente 76 metros no Equador, o que é o suficiente para determinados níveis de zoom. Além disso, o código do front-end foi modificado especificamente para fins de performance, assim obtendo clusterizações com visualização mais natural. Graças ao suporte do GeoDjango para inserir o geohash em queries, foi possível realizar a implementação em um algoritmo simples, sem escrever as queries manualmente. A principal desvantagem deste algoritmo foi a extrema demora para realizar as clusterizações em comparação com os demais algoritmos; porém, o resultado final com a integração de clusterização front-end e back-end foi melhor. Outra desvantagem foi que, como não há correspondência direta

entre níveis de zoom do mapa e precisões de geohash, determinados níveis de zoom demandam mais tempo para ler os dados. Além disso, há algumas pequenas falhas visuais perto das bordas, que deverão ser corrigidas futuramente. Por fim, foram realizadas alterações na API para tornar a transição entre clusters e ERBs opaca para o usuário final – ou seja, o script de exibição do mapa do navegador enxerga um único endpoint e não diferencia quando o servidor opta por antenas ao invés de clusters para zooms menores.

Este algoritmo foi expandido para também permitir a filtragem por operadora de celular. Esta filtragem se baseia no campo MNC do identificador de cada estação radiobase, que distingue a empresa responsável. Para filtros mais precisos, como tipo de tecnologia celular, não será utilizada clusterização, com carregamento e exibição diretamente no mapa – já que inclui um número menor de antenas a serem exibidas, e varia com o caso de uso de cada usuário, portanto sendo um processo menos repetitivo para o servidor que não exige pré-computação.

## 6.2 Métodos Numéricos

No nosso trabalho utilizamos os métodos Taguchi, Basinhopping e SLSQP. A função de otimização foi desacoplada da implementação do método numérico. Desse modo, podemos combinar quaisquer métodos numéricos implementados com quaisquer funções de otimização.

### 6.2.1 Método Taguchi

O método Taguchi foi implementado de acordo com a especificação encontrada em [10]. Os arrays ortogonais usados pelo método foram implementados como uma tabela, uma vez que seu cálculo é uma operação complexa, e os casos de uso preveem o uso de poucos parâmetros no método Taguchi. Usamos a biblioteca numpy para as operações com array, uma vez que ela provê operações eficientes nessas estruturas de dados. Criamos testes para verificar o método Taguchi, conforme será explicado no capítulo Testes e Avaliação.

### 6.2.2 Método Basinhopping

Usamos a implementação da biblioteca scipy para o método Basinhopping.

### **6.2.3 Método SQSQP**

Usamos a implementação da biblioteca `scipy` para o método SQSQP.

## **6.3 Funções de Otimização**

## **6.4 Frontend**

### **6.4.1 Projeto de Telas**

### **6.4.2 Implementação com Vuetify**

## 7 TESTES E AVALIAÇÃO



## 8 CONSIDERAÇÕES FINAIS

### 8.1 Conclusões do Projeto de Formatura

### 8.2 Contribuições

### 8.3 Perspectivas de Continuidade

## REFERÊNCIAS

- [1] Forsk. *Atoll LTE / LTE-A Planning Software — Forsk* Disponível em: <http://www.forsk.com/ltelte-pro>. Acesso em: 01º de março de 2018.
- [2] MyTower. *MyTower - Aluguel e Venda de Terrenos e Topos para Operadoras de Telecom* Disponível em: <http://www.mytower.com.br/>. Acesso em: 01º de março de 2018.
- [3] Skysites. *Skysites* Disponível em: <http://skysites.com/>. Acesso em: 01º de março de 2018.
- [4] Telebrasil. *Mapa de ERBs Brasil (antenas)*. Disponível em: <http://www.telebrasil.org.br/panorama-do-setor/mapa-de-erbs-antenas>. Acesso em: 31 de janeiro de 2018.
- [5] Unwired Labs. *OpenCellID - Largest Open Database of Cell Towers & Geolocation by Unwired Labs*. Disponível em: <https://opencellid.org/>. Acesso em: 01º de março de 2018.
- [6] Google Earth. *Google Earth Engine*. Disponível em: <https://earthengine.google.com/>. Acesso em: 16 de março de 2018.
- [7] Yale University. *Geographically based Economic data -- Brazil*. Disponível em: <https://gecon.yale.edu/brazil>. Acesso em: 24 de março de 2018.
- [8] Núcleo de Informação e Coordenação do Ponto BR. *Sistema de Medição de Tráfego Internet*. Disponível em: <http://simet.nic.br/mapas-app.html>. Acesso em: 24 de março de 2018.
- [9] LEE, S.; LEE, S.; KIM, K.; KIM, YH. *Base Station Placement Algorithm for Large-Scale LTE Heterogeneous Networks*. PLoS ONE 10(10), 2015.
- [10] Wegmann, A.; Viering I.; Klein, A. *A Joint Optimization of Antenna Parameters in a Cellular Network Using Taguchi's Method* IEEE 73rd Vehicular Technology Conference (VTC Spring), 2011.
- [11] KARULKAR, S. A.; OH, JY. *Optimal Placement of Base Station for Cellular Network Expansion*. Issues in Information Systems, volume 17, edição II, pg. 215-221, 2016.
- [12] biodiv (GitHub). *anycluster*. Disponível em: <https://github.com/biodiv/anycluster>. Acesso em: 6 de maio de 2018.
- [13] EvgeneOskin (GitHub). *django-geohash-cluster*. Disponível em: <https://github.com/EvgeneOskin/django-geohash-cluster>. Acesso em: 8 de setembro de 2018.