

UNIVERSIDADE FEDERAL DE VIÇOSA  
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS  
DEPARTAMENTO DE INFORMÁTICA

CARLOS HENRIQUE TAVARES BRUMATTI

**UM FRAMEWORK PARA DESENVOLVIMENTO DE SISTEMAS VGI COM BASE NO  
OPENSTREETMAP**

Viçosa 2021

CARLOS HENRIQUE TAVARES BRUMATTI

**UM FRAMEWORK PARA DESENVOLVIMENTO DE SISTEMAS VGI COM BASE NO  
OPENSTREETMAP**

Relatório final apresentado como parte  
do Programa Institucional de  
Voluntariado em Iniciação Científica da  
Universidade Federal de Viçosa (PIVIC/UFV)

Orientador: Jugurta Lisboa-Filho

Viçosa 2021

## **Resumo**

Este trabalho tem o objetivo de propor um framework para o desenvolvimento de Sistemas de Informação Geográfica (SIG) que utilizam Informação Geográfica Voluntária (VGI) como método de obtenção de dados. Além disso, a ferramenta deve ser capaz de exportar seus dados seguindo o formato GeoJSON, permitindo assim a leitura desses dados por outros SIG, como o QGIS por exemplo, e a associação das colaborações com os dados vetoriais existentes, isto é, pontos, linhas e polígonos, possibilitando assim o dimensionamento do fato observado. A implementação utiliza tecnologias web, destacando o uso da biblioteca gráfica Leaflet. Para idealização do framework, utilizou-se conceitos e técnicas de Engenharia de Software. Por fim, foi realizada a implementação do framework, que encontra-se disponível como software livre.

**Palavras-chave:** SIG; VGI; Framework; Exportação de dados.

### **Área de Conhecimento:**

1.03.03.03-0 Banco de Dados

1.03.03.04-9 Sistemas de Informação

### **Linha de Pesquisa:**

DPI-015 – Banco de Dados

DPI-016 – Sistemas de Informação

## Lista de Figuras

Figura 1 – Esquema geral do papel de um SGBD em um sistema computacional. Fonte: SISTEMAS DE BANCO DE DADOS: PROJETO, IMPLEMENTAÇÃO E GERENCIAMENTO .....	9
Figura 2 – Esquema geral do funcionamento da arquitetura MVC. Fonte: PADRÃO DE ARQUITETURA MVC: MODEL-VIEW-CONTROLLER .....	10
Figura 3– Interface do phpMyAdmin. Fonte: Autor .....	11
Figura 4 – Diagrama ER do banco de dados. Fonte: Autor .....	21
Figura 5 – Página inicial do framework. Fonte: Autor .....	22
Figura 6– Menu Colaborar. Fonte: Autor .....	23
Figura 7– Formulário para contato. Fonte: Autor .....	24
Figura 8– Formulário para cadastro. Fonte: Autor .....	24
Figura 9 – exibindo colaborações do sistema. Fonte: Autor .....	25
Figura 10 – Exportação das colaborações. Fonte: Autor.....	26
Figura 11 – Interface do usuário anônimo. Fonte: Autor .....	27
Figura 12 – Interface do usuário colaborador. Fonte: Autor .....	27
Figura 13– Interface do usuário moderador. Fonte: Autor.....	28
Figura 14 – Interface do usuário root (super usuário). Fonte: Autor.....	28

## Lista de Tabelas

Tabela 1 – Estabelecer conexão com o banco de dados .....	13
Tabela 2 – Login do usuário .....	13
Tabela 3 – Cadastro de usuário .....	14
Tabela 4 - Colaboração .....	14
Tabela 5 – Contato .....	15
Tabela 6 - Cadastrar categorias .....	15
Tabela 7 - Cadastrar subcategorias .....	16
Tabela 8 - Aprovar colaboração .....	16
Tabela 9 – Logout .....	16
Tabela 10 - Exportar colaborações .....	17
Tabela 11 - Configurar o sistema .....	17
Tabela 12 - Adicionar moderadores .....	17
Tabela 13 - Aprovar colaborações .....	18
Tabela 14 - Visualizar perfil do usuário .....	18
Tabela 15 - Trocar senha .....	19
Tabela 16 - Visualizar perfil do usuário .....	19
Tabela 17 - Recuperar senha .....	19
Tabela 18 - Requisitos suplementares .....	20

## Sumário

<b>1 Introdução.....</b>	<b>7</b>
1.1 Objetivo Geral .....	8
1.2 Objetivos Específicos .....	8
<b>2 Referencial Teórico .....</b>	<b>8</b>
2.1 World Wide Web .....	8
2.2 Protocolo HTTP .....	8
2.3 Servidor Web.....	9
2.4 Sistema de Gerenciamento de Banco de Dados .....	9
2.5 Arquitetura Model-View-Controller (MVC) .....	10
<b>3 Tecnologias Utilizadas.....</b>	<b>10</b>
3.1 Leaflet.....	10
3.2 MySQL .....	11
3.3 phpMyAdmin .....	11
3.4 XAMPP .....	11
3.5 PHP .....	12
3.6 GeoJSON .....	12
<b>4 Metodologia .....</b>	<b>12</b>
4.1 Requisitos funcionais e não-funcionais.....	12
4.2 Requisitos Suplementares .....	20
4.3 Modelo conceitual do Banco de Dados .....	20
<b>5 Resultados .....</b>	<b>21</b>
<b>6 Conclusões .....</b>	<b>29</b>
<b>7 Referências Bibliográficas.....</b>	<b>30</b>

## 1 Introdução

O desenvolvimento da tecnologia nos últimos anos foi impulsionado graças a popularização da internet e o acesso cada vez maior a ela. Segundo Moreira (2009), o aumento da velocidade proporcionado pelo advento da banda larga representou um marco nessa história: páginas de sites deixavam de ser estáticas e se tornavam dinâmicas, proporcionando maior interatividade com o usuário. Assim, o volume de informação consumido e a quantidade de dados gerados aumentou drasticamente, principalmente diante do surgimento das redes sociais, onde as pessoas passaram a compartilhar informações pessoais, sobretudo em relação ao local em que se encontram e o que estão fazendo. Nesse contexto, temos o surgimento do usuário-produtor ou “producer”, conforme definido por BUDHATHOKI et al (2008). Esse novo tipo de usuário, além de visualizar e interpretar a informação em tela, ele passa a colaborar com a geração de conteúdo, seja de maneira passiva, onde os sistemas coletam a informação de forma automática referentes ao acesso e uso do sistema, ou de maneira ativa, em que os usuários em si fornecem algum tipo de informação da sua realidade.

Para se trabalhar com dados geográficos, isto é, dados que apresentam coordenadas geográficas - latitude e longitude – são usados Sistemas de Informação Geográfica (SIG), possibilitando assim a manipulação, processamento e análise desses dados. Goodchild (2007) definiu uma das formas de se obter esses dados geográficos, sendo por meio de Sistemas de Informação Geográfica Voluntária (VGI - “*Volunteered Geographic Information*”), onde o usuário compartilha dados específicos de maneira voluntária, uma vez que somente ele apresenta um conhecimento maior a cerca da sua própria realidade, e disponibiliza em uma plataforma para o amplo acesso. Dessa forma, os sistemas VGI se distinguem dos sistemas tradicionais ou analógicos (Bravo e Slunter, 2015) por não necessitarem de inúmeros equipamentos para mensurar a informação geográfica ou de mesas de luz para sobreposição de mapas.

Sistemas VGI são também de grande importância principalmente para pequenos municípios que não apresentam muitos recursos financeiros e mão-de-obra especializada, já que permitem a coleta de um grande volume de dados por pessoas que não precisam passar por nenhum treinamento técnico para isso. Além disso, a presença de uma *framework* opensource, permite que os gestores municipais atuem de maneira rápida e certa frente alguma adversidade, sem grandes custos e permitindo que o processo de tomada de decisão ocorra com a presença do máximo de informações possíveis.

## 1.1 Objetivo Geral

Implementar um *framework* voltado para o desenvolvimento de Sistemas de Informação Geográfica Voluntária que possua formas de associar a colaboração do usuário com os dados vetoriais existentes e maneiras de exportar esses dados informados.

## 1.2 Objetivos Específicos

Para o objetivo geral ser alcançado, será necessário concluir as seguintes etapas:

- Definir as tecnologias a serem utilizadas.
- Definir os requisitos funcionais, não-funcionais e suplementares do *framework*.
- Formular o esquema geral para o modelo conceitual do banco de dados.
- Implementar o *framework* seguindo todas as especificações levantadas.

## 2 Referencial Teórico

Nesta seção será abordado os conceitos e técnicas mais gerais utilizadas no decorrer do desenvolvimento do *framework*.

### 2.1 World Wide Web

A Internet pode ser definida como sendo o conjunto de uma grande rede de computadores interligados entre si, sendo a *World Wide Web* (WWW ou simplesmente *Web*) um dos serviços disponíveis. A *Web* é um sistema cliente-servidor disponível na internet onde os usuários por meio do *browser* solicitam informações de servidores através do protocolo HTTP, que respondem a essas solicitações fornecendo objetos em hipertexto, como HTML, JSON e XML, por exemplo, sendo o *browser* capaz de interpretar e mostrar essa informação ao usuário que solicitou.

### 2.2 Protocolo HTTP

*HyperText Transfer Protocol* (HTTP) é o protocolo sob o qual acontece a comunicação na Internet. Por meio de solicitações (*requests*) e respostas (*responses*), as mensagens são enviadas e recebidas pelos clientes e servidores. Em suas requisições, o protocolo HTTP trabalha com um conjunto de métodos de requisição (ou verbos HTTP), sendo os mais comuns:

- GET – Responsável por solicitar um dado ao servidor.
- POST – Responsável por enviar um dado ao servidor.
- PUT – Responsável por alterar um dado presente no servidor pelo dado levado na requisição.



- PATCH – Responsável por alterar parcialmente um dado que consta no servidor
- DELETE – Responsável por excluir algum dado presente no servidor.

O protocolo HTTP apresenta as vantagens de ser simples, legível para as pessoas, não permitir duas requisições ou mais durante uma mesma conexão e permite trabalhar com a ideia de sessões de usuários, usando *cookies*.

### 2.3 Servidor Web

*Software* capaz de interpretar as requisições HTTP recebidas e fornecer uma resposta de acordo com elas. Essa resposta pode ser tanto arquivos quanto informações. Um servidor pode ser estático, quando os dados ofertados não passam por nenhum processo de alteração quando estão armazenados, ou podem ser dinâmicos (também chamados de servidores de aplicações), quando trabalham em conjunto com um banco de dados atualizando a informação presente nele.

### 2.4 Sistema de Gerenciamento de Banco de Dados

De acordo com Rob e Coronel (2010), um Sistema de Gerenciamento de Banco de Dados (SGBD) é o *software* que faz todo o gerenciamento e acesso aos dados inseridos nele. Dessa forma, atuam como um intermediário entre quem usa um sistema computacional e o banco de dados implementado no servidor. A presença de um SGBD faz com que um mesmo banco de dados possa ser acessado por diversos *softwares*, bastando para isso que os parâmetros de conexão, como nome de *host*, nome do banco, usuário e senha da base de dados sejam disponíveis.

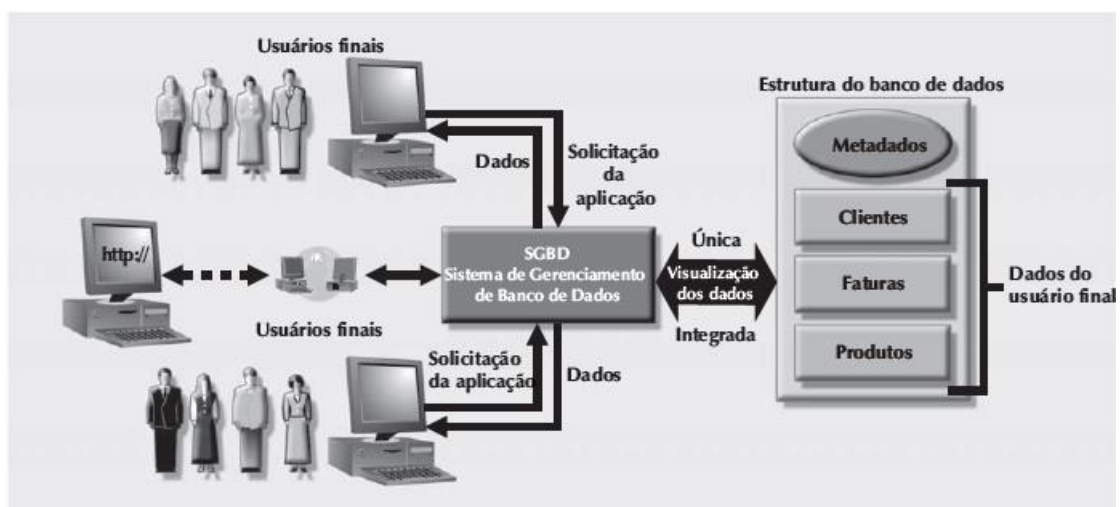


Figura 1 – Esquema geral do papel de um SGBD em um sistema computacional. Fonte: SISTEMAS DE BANCO DE DADOS: PROJETO, IMPLEMENTAÇÃO E GERENCIAMENTO

## 2.5 Arquitetura Model-View-Controller (MVC)

Comumente utilizado em aplicações web, a arquitetura Model-View-Controller (MVC) se caracteriza por ser simples, separando as camadas de apresentação (*view*), lógica do negócio (*model*) e gerenciamento do fluxo do sistema (*controller*), visando a reutilização de código e facilitando a posterior manutenção pela equipe.

Cada camada definida apresenta uma função específica dentro do fluxo de informação do sistema: as requisições são capturadas pelo *Controller* e são levadas e processadas pelo *Model*, exibindo o resultado pela *View*. O esquema de funcionamento pode ser visto na figura abaixo:

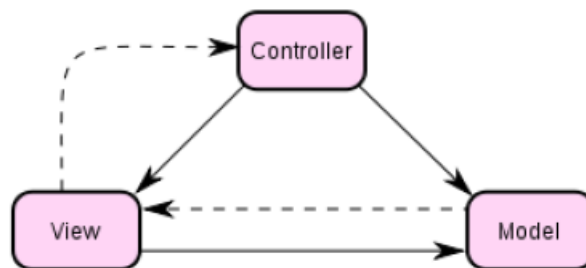


Figura 2 – Esquema geral do funcionamento da arquitetura MVC. Fonte: PADRÃO DE ARQUITETURA MVC: MODEL-VIEW-CONTROLLER

## 3 Tecnologias Utilizadas

Todas as tecnologias empregadas na construção do *framework* são *open source*, evitando custos adicionais para o desenvolvimento e manutenção dos sistemas desenvolvidos com base nessa *framework*. Foram utilizados as seguintes tecnologias:

### 3.1 Leaflet

O *Leaflet* é uma biblioteca JavaScript para mapas interativos *open-source*, desenvolvida por Vladimir Agafonkin em 13 de maio de 2011, se encontrando atualmente na versão 1.7.1, lançada em 04 de setembro de 2020. Foi desenvolvida com foco na simplicidade, performance e usabilidade, além de ser “*mobile-friendly*”, isto é, responsiva até mesmo para dispositivos móveis. Possui um conjunto grande de *plugins* disponíveis, sendo esses implementados pela comunidade, responsável também por manter e atualizar o código fonte da biblioteca.

### 3.2 MySQL

O MySQL é um Sistema de Gerenciamento de Banco de Dados Relacional (SGBDR), isto é, permite armazenar dados em tabelas que podem ou não se relacionar entre si por um sistema de chaves estrangeiras (*foreign key*). É um dos sistemas mais comuns, sendo amplamente utilizado, principalmente por empregar a *Structured Query Language* (SQL) como linguagem de consulta.

### 3.3 phpMyAdmin

phpMyAdmin é um software para administração de banco de dados bem popular atualmente. Permite a criação, operação e exclusão de tabelas no MySQL de maneira visual. Possui compatibilidade com a maioria dos sistemas operacionais, além de apresentar uma interface simples e intuitiva, com alta velocidade no processamento dos dados.

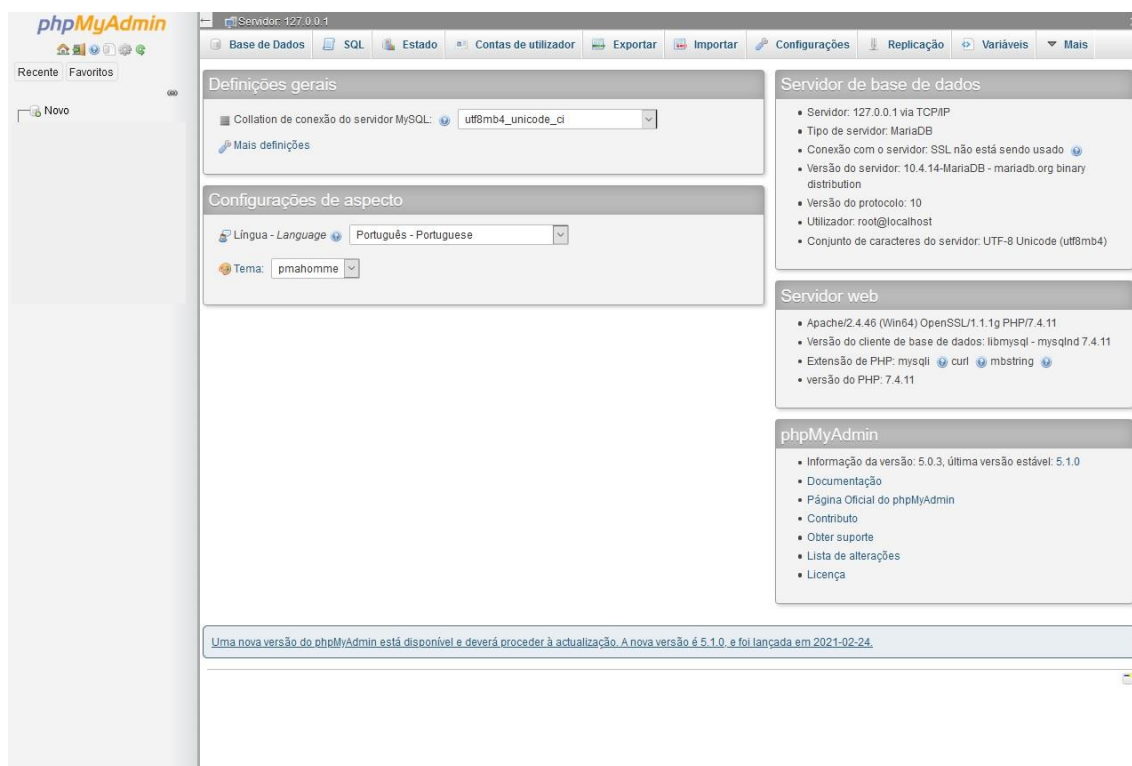


Figura 3— Interface do phpMyAdmin. Fonte: Autor

### 3.4 XAMPP

O XAMPP é um pacote *open source* de fácil instalação e manuseio que contém as principais ferramentas para iniciar um ambiente de desenvolvimento

web local, sendo elas:

- Apache (Servidor HTTP)
- MySQL (Sistema de Gerenciamento de Banco de Dados)
- FileZilla (Servidor de arquivos)
- Mercury (Servidor de *email*)
- Tomcat (Servidor web Java)
- PHP (Linguagem de programação)
- Perl (Linguagem de programação)

### 3.5 PHP

*Hypertext Preprocessor* (PHP) é uma linguagem de programação *open source* que trabalha no lado do servidor da aplicação. É bastante utilizada no desenvolvimento web porque consegue ser integrada ao código *HyperText Markup Language* (HTML), tornando-o dinâmico. Atualmente na versão 8.0.3, lançada em 04 de março de 2021, se caracteriza por ser rápida, flexível, simples e poderosa.

### 3.6 GeoJSON

Objeto JavaScript baseado no JSON e usado na comunicação entre servidores de dados e aplicações. É um formato projetado para representar dados vetoriais espaciais, como pontos, linhas e polígonos, assim como quaisquer atributos associados a eles. É padronizado pela RFC 7946, padrão internacional publicado em agosto de 2016 pela *Internet Engineering Task* (IETF).

## 4 Metodologia

Para a concepção desse projeto, primeiro foi elaborado um conjunto de requisitos funcionais, não-funcionais e suplementares, modelados de acordo com o template apresentado em sala de aula durante a disciplina de Engenharia de Software I (INF 221).

### 4.1 Requisitos funcionais e não-funcionais

RF1 Estabelecer conexão com o banco de dados	Oculto (x)
<b>Descrição:</b> O Sistema deverá se conectar ao banco de dados criado.	

Requisitos não funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF1.1 Mensagem de erro	O sistema deve retornar uma mensagem de erro caso a conexão com o banco de dados não ocorra.	Usabilidade	( )	(x)
NF1.2	A conexão com o banco de dados é automática e ocorre toda vez que o usuário selecionar alguma funcionalidade que necessita do acesso ao banco.	Implementação	( )	(x)

Tabela 1– Estabelecer conexão com o banco de dados

RF2 Login do usuário		Oculto ( )		
Descrição: O usuário deverá fazer login para entrar no sistema.				
Requisitos não funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF2.1	O usuário deverá usar o apelido ou email e a senha para fazer o login no sistema	Segurança	(x)	( )

Tabela 2 – Login do usuário

RF3 Cadastro de usuário		Oculto ( )		
Descrição: O usuário poderá realizar o seu cadastro no sistema.				
Requisitos não funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF3.1	O cadastro de usuário deverá possuir os seguintes campos: nome, apelido, faixa etária, email e senha.	Interface	(x)	( )
NF3.2	Os campos nome, apelido, email e senha deverão ser do tipo <i>input</i> , enquanto o campo faixa etária deverá ser do tipo <i>select</i> .	Interface	(x)	()

NF3.3	O campo senha deverá ser armazenado no banco de dados criptografado.	Segurança	()	(x)
-------	--	-----------	----	-----

Tabela 3 – Cadastro de usuário

RF4 Colaboração		Oculto ( )		
Descrição: O sistema irá permitir que uma colaboração seja feita de maneira voluntária.				
Requisitos não funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF4.1	Para cada colaboração realizada, será necessário associar ela a um dado vetorial como ponto, linha ou polígono.	Implementação	( )	(x)
NF4.2	Uma colaboração deverá conter os campos titulo, categoria, subcategoria, data da ocorrência, tipo de geometria e descrição.	Interface	(x)	()
NF4.3	Os campos titulo, data de ocorrência e descrição deverão ser do tipo <i>input</i> , enquanto que o campos categoria e subcategoria serão do tipo <i>select</i> . Além disso, o campo tipo de geometria deverá ser de preenchimento automático, de acordo com a geometria selecionada pelo usuário	Interface	(x)	()
NF4.4	O formulário para colaboração deverá ser exibido no <i>popup</i> da geometria.	Interface	()	(x)
NF4.5	A colaboração poderá ser realizada somente pelos usuários colaboradores, moderadores e anônimo	Segurança	()	(x)

Tabela 4 - Colaboração

RF5 Contato	Oculto ( )
<b>Descrição:</b> Qualquer pessoa deverá ser capaz de realizar um contato pelo sistema.	

Requisitos não funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF5.1	O formulário para conta deverá conter o campos nome, email, assunto e mensagem.	Interface	(x)	( )
NF5.2	Todos o campos do formulário deverão ser do tipo <i>input</i> .	Interface	(x)	( )

Tabela 5 – Contato

RF6 Cadastrar categorias		Oculto ( )		
Descrição: O sistema deverá suportar o cadastro de categorias.				
Requisitos não funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF6.1	A inserção de categorias deverá ser feita somente pelos usuários root ou moderador do sistema.	Segurança	( )	(x)
NF6.2	O campo para inserir uma nova categoria no sistema deverá ser do tipo <i>input</i> .	Interface	( )	(x)

Tabela 6 - Cadastrar categorias

RF7 Cadastrar subcategorias		Oculto ( )		
Descrição: Para cada categoria adicionada, poderá haver várias subcategories associadas.				
Requisitos não funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF7.1	A inserção de uma nova subcategoria deverá conter um campo do tipo <i>select</i> para escolha da categoria e um campo do tipo <i>input</i> para escrita da nova	Interface	( )	(x)

	subcategoria.			
--	---------------	--	--	--

Tabela 7 - Cadastrar subcategorias

RF8 Aprovar colaboração		Oculto ( )		
<b>Descrição:</b> Após uma colaboração ser realizada, ela precisa ser aprovada para assim ser exibida no sistema.				
<b>Requisitos não funcionais</b>				
Nome	Restrição	Categoria	Desejável	Permanente
NF8.1	Somente os usuários root e moderador podem aprovar colaborações.	Segurança	( )	(x)
NF8.2	Todos os dados de uma colaboração devem ser exibidos em uma mesma tabela ou um agrupamento com as informações.	Interface	( )	( )
NF8.2	O campo para autorizar a publicação deverá ser do tipo <i>select</i> .	Interface	(x)	( )

Tabela 8 - Aprovar colaboração

RF9 Logout		Oculto ( )		
Descrição: Um usuário logado no sistema deverá ser capaz encerrar a sessão com segurança.				
Requisitos não funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF9.1	Deverá haver um botão para o usuário sair do Sistema.	Interface	( )	(x)

Tabela 9 – Logout



RF10 Exportar colaborações		Oculto ( )		
Descrição: As colaborações podem ser exportadas pelo sistema.				
Requisitos não funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF10.1	Somente as colaborações aprovadas poderão ser exportadas.	Interface	( )	( )

Tabela 10 - Exportar colaborações

RF11 Configurar <i>framework</i>		Oculto (x)		
<b>Descrição:</b> A <i>framework</i> deverá ser genérica e assim capaz de ser configurada para qualquer tipo de necessidade.				
Requisitos não funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF11.1	Os campos configuráveis da <i>framework</i> deverão ser o nome do sistema, a latitude, a longitude, o zoom inicial e uma breve descrição.	Interface	()	(x)

Tabela 11 - Configurar o sistema

RF12 Adicionar moderador		Oculto ( )		
Descrição: O sistema irá permitir que novos administradores sejam escolhidos e adicionados.				
Requisitos não funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
	Para ser um moderador, o usuário deverá estar cadastrado no sistema previamente.	Usabilidade	( )	( )
	Somente o usuário root poderá definir novos moderadores para o sistema.	Segurança		

Tabela 12 - Adicionar moderadores

RF13 Aprovar colaborações		Oculto ( )		
Descrição:				
Requisitos não funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF13.1	Somente o usuário moderador poderá aprovar uma colaboração	Segurança	(x)	( )
NF13.2	Todas as colaborações devem ser exibidas a partir de uma opção do painel do usuário	Interface	( )	( )
NF13.3	Para cada colaboração realizada, deve haver uma maneira do Usuário aprovar ou não a exibição pública dela.	Interface	( )	(x)

Tabela 13 - Aprovar colaborações

RF14 Visualizar perfil do usuário		Oculto ( )		
Descrição: O Usuário será capaz de verificar seus dados cadastrados.				
Requisitos não funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF14.1	Os dados cadastrados dos usuários deverão aparecer na forma de tabela.	Interface	(x)	()

Tabela 14 - Visualizar perfil do usuário

RF15 Trocar senha		Oculto ( )		
Descrição: Será possível um usuário alterar a senha que ele se cadastrou.				
Requisitos não funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF15.1	Para altera a senha, o usuário deve informar o email cadastrado, assim como a senha antiga e a nova senha.	Segurança	(x)	()

Tabela 15 - Trocar senha

RF16 Visualizar colaborações		Oculto ( )		
Descrição: O sistema deve exibir as colaborações que forem realizadas nele.				
Requisitos não funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF16.1	Cada colaboração realizada deverá ser exibida em um <i>popup</i> assim como sua geometria no mapa.	Interface	( )	(x)
NF16.2	Somente as colaborações que forem publicadas poderão ser exibidas.	Segurança	( )	(x)

Tabela 16 - Visualizar perfil do usuário

RF17 Recuperar senha		Oculto ( )		
<b>Descrição:</b> Caso o Usuário esqueça sua senha, ele deverá ser capaz de solicitar uma nova senha.				
<b>Requisitos não funcionais</b>				
Nome	Restrição	Categoria	Desejável	Permanente
NF17.1	Após solicitar uma nova senha, o usuário deve informar o email cadastrado para que seja enviado uma nova senha para ele.	Segurança	(x)	()

Tabela 17 - Recuperar senha

## 4.2 Requisitos Suplementares

Nome	Restrição	Categoria	Desejável	Permanente
S1 Tipo de interface	A <i>framework</i> possuirá interface web	Usabilidade	( )	(x)
S2 Tipos de usuário	A <i>framework</i> implementa sistemas com 04 tipos de usuários: colaborador, moderador, anônimo e root (super usuário)	Segurança	(x)	( )
S3 Inserção dos dados nos <i>inputs</i>	A <i>framework</i> deverá verificar se todos os dados obrigatórios foram inseridos nos campos correspondentes	Segurança	( )	(x)
S4 Confirmações	O sistema deve informar para o usuário uma mensagem indicando se uma dada ação foi ou não executada corretamente	Usabilidade	(x)	(x)
S5 Exportação das colaborações	As colaborações devem ser exportadas no formato geojson.	Interface	( )	(x)

Tabela 18 - Requisitos suplementares

## 4.3 Modelo conceitual do Banco de Dados

O banco de dados da *framework* foi implementado conforme o modelo ER definido na figura 4:

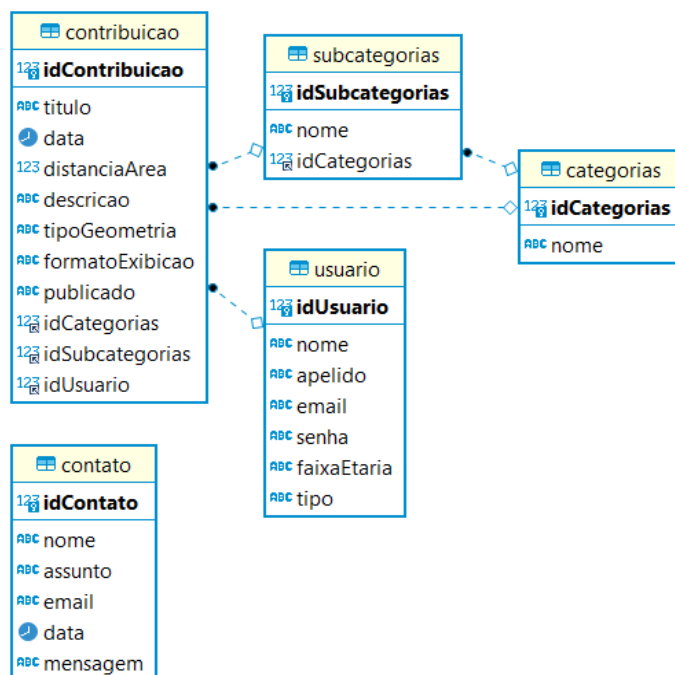


Figura 4 – Diagrama ER do banco de dados. Fonte: Autor

## 5 Resultados

Dado os requisitos funcionais, não-funcionais e suplementares levantados, implementou-se o *framework* buscando seguir as especificações levantadas. Neste capítulo, é apresentado o resultado alcançado.

A figura 5 abaixo mostra a interface inicial da *framework*. Os campos em destaque podem ser nomeados de acordo com o objetivo do sistema a ser desenvolvido. Além desses campos, a latitude, longitude e o zoom inicial do mapa que exibe as colaborações para visualização e até mesmo do mapa exibido para a realização de uma colaboração, são também configuráveis.

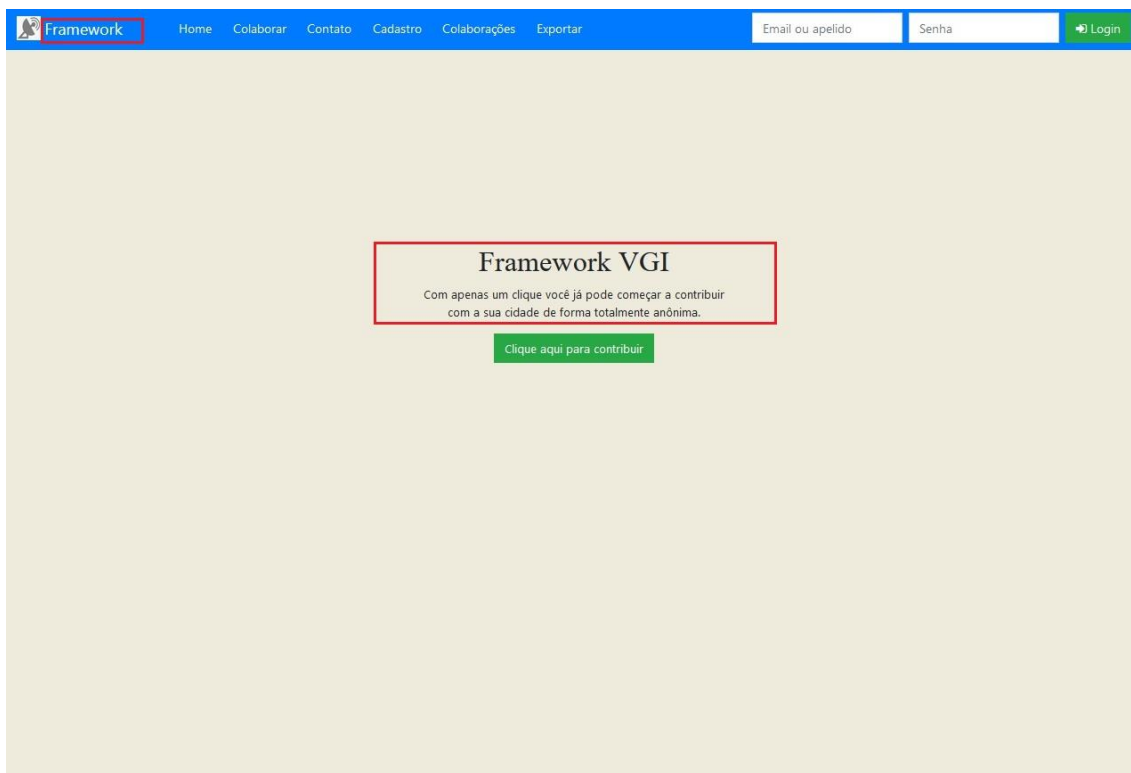


Figura 5 – Página inicial do framework. Fonte: Autor

Na opção “Colaborar”, o usuário é informado pelo sistema que primeiro é necessário fazer *login* para ser um usuário colaborador, senão é possível apenas colaborar como um usuário anônimo, conforme a figura 6 destaca:

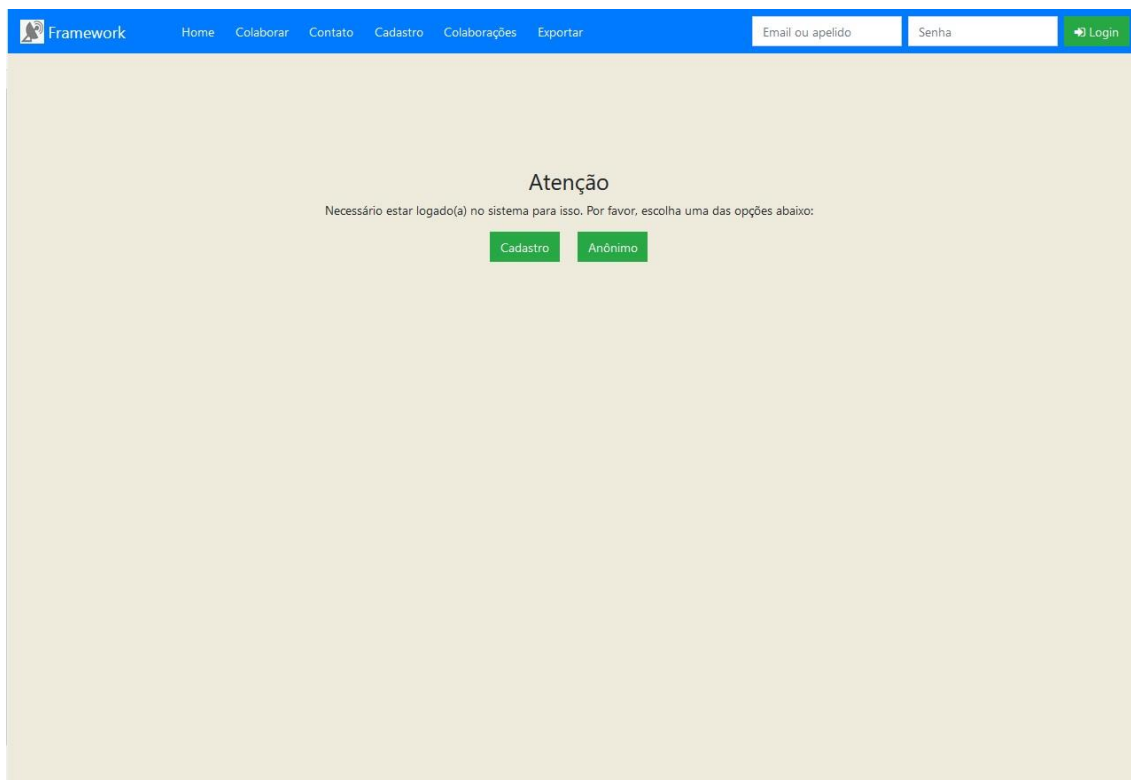


Figura 6– Menu Colaborar. Fonte: Autor

Nas opções “Contato” e “Cadastro” são exibidos os formulários, respectivamente, para mandar uma mensagem aos moderadores do sistema e para realizar o cadastro no sistema, criando um usuário colaborador, conforme as figuras 7 e 8:

Framework Home Colaborar Contato Cadastro Colaborações Exportar Email ou apelido Senha Login

## CONTATO

Nome:

Email:

Assunto:

Mensagem:

ENVIAR

Figura 7– Formulário para contato. Fonte: Autor

Framework Home Colaborar Contato Cadastro Colaborações Exportar Email ou apelido Senha Login

## CADASTRO

Nome:

Apelido:

Faixa etária:

Email:

Senha:

CADASTRAR

Figura 8– Formulário para cadastro. Fonte: Autor



Uma vez que *framework* esteja configurada e colocada em uso, as colaborações realizadas e aprovadas pelos moderadores do sistema são exibidas na opção “Colaborações”, conforme a figura 9 abaixo mostra:

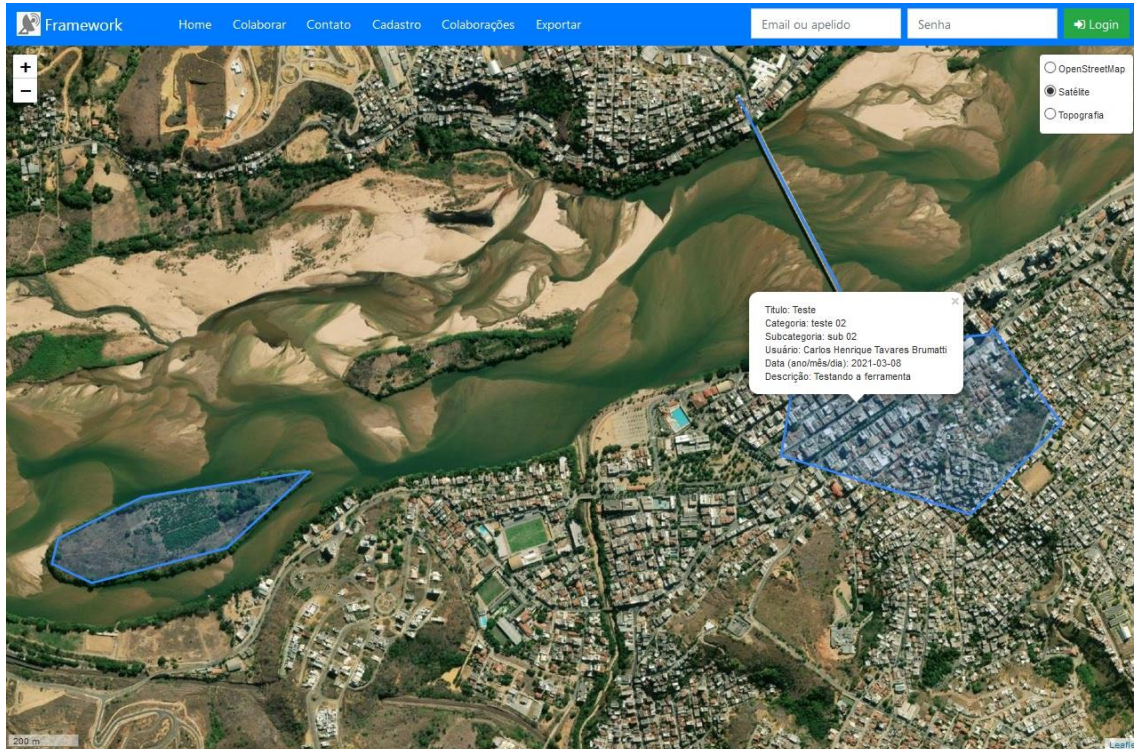


Figura 9 – exibindo colaborações do sistema. Fonte: Autor

Os dados uma vez que disponíveis para visualização se tornam também disponíveis para *download*. Para isso, basta acessar a opção “Exportar” para ter acesso às colaborações no formato geojson, conforme a figura 10 abaixo:

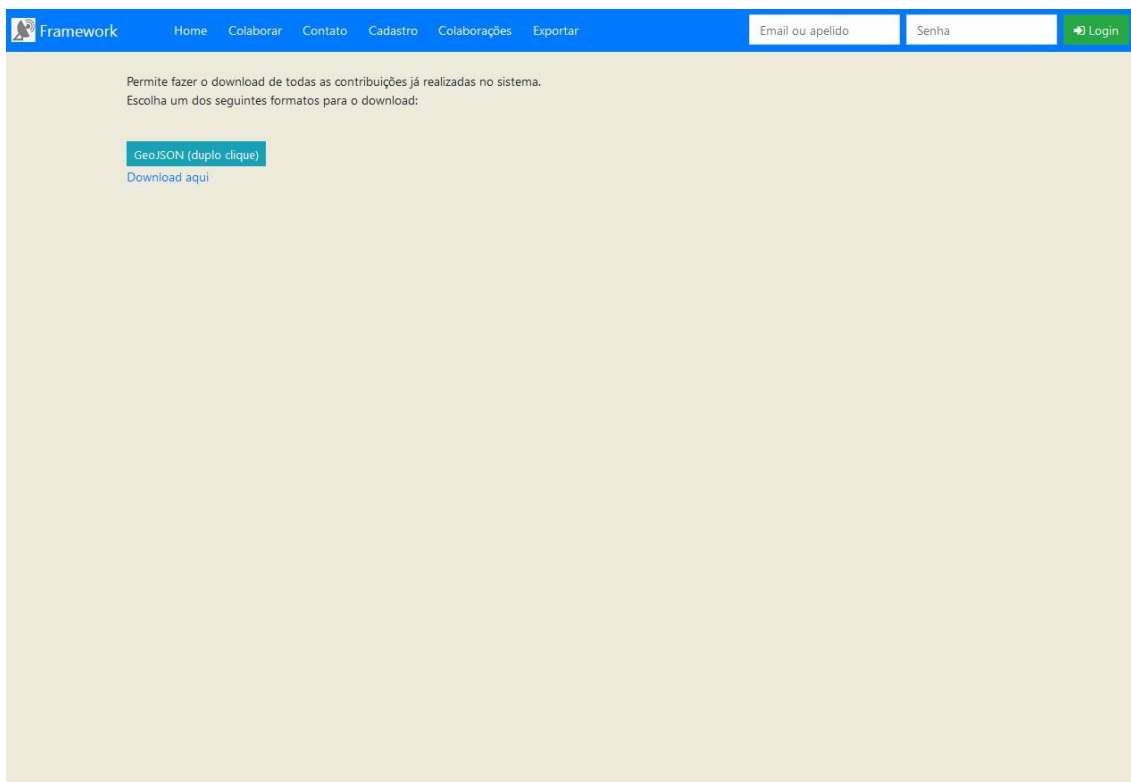


Figura 10 – Exportação das colaborações. Fonte: Autor

O sistema foi desenvolvido pensando em 04 tipos de usuários: anônimo, colaborador, moderador e root (super usuário), cada um com suas próprias funcionalidades, conforme exibido nas figuras 11, 12, 13 e 14 abaixo:

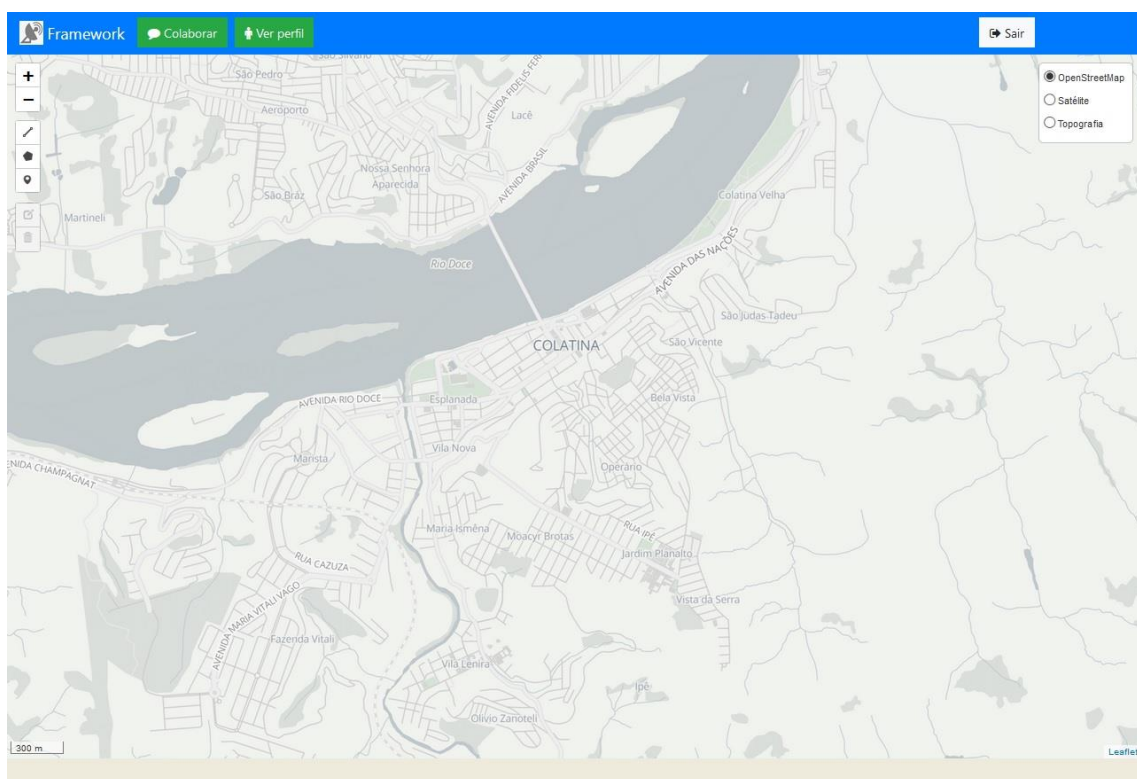


Figura 11 – Interface do usuário anônimo. Fonte: Autor

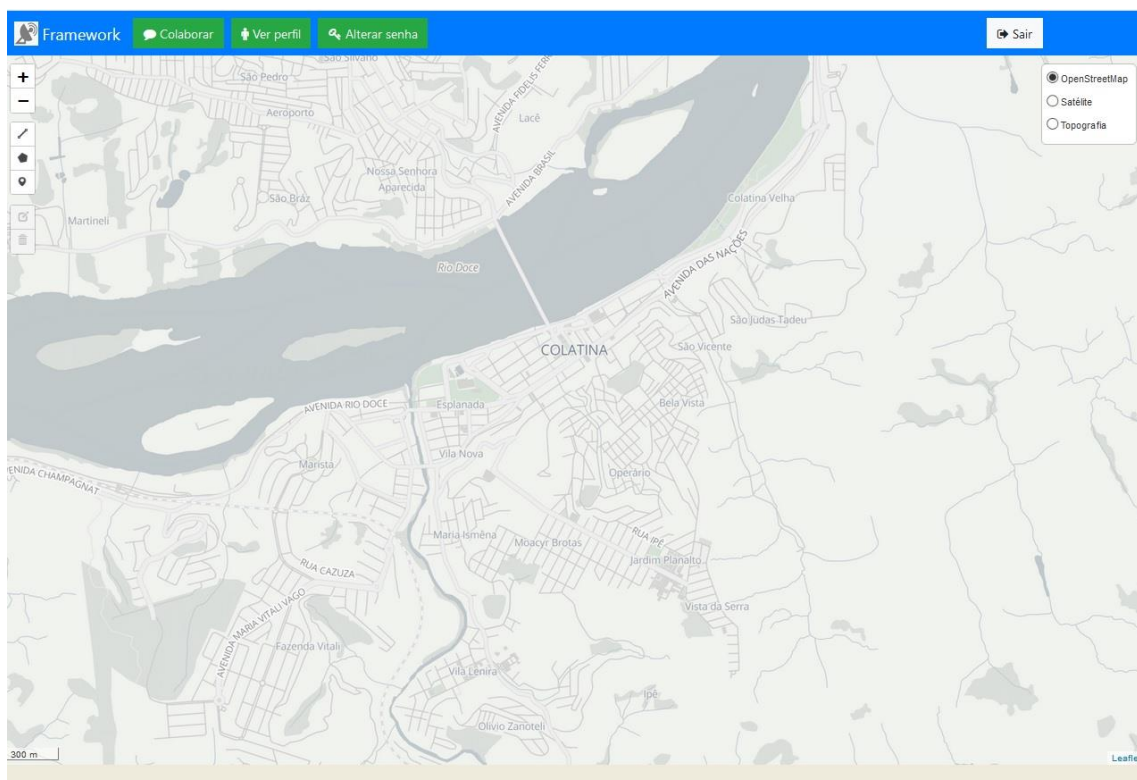


Figura 12 – Interface do usuário colaborador. Fonte: Autor

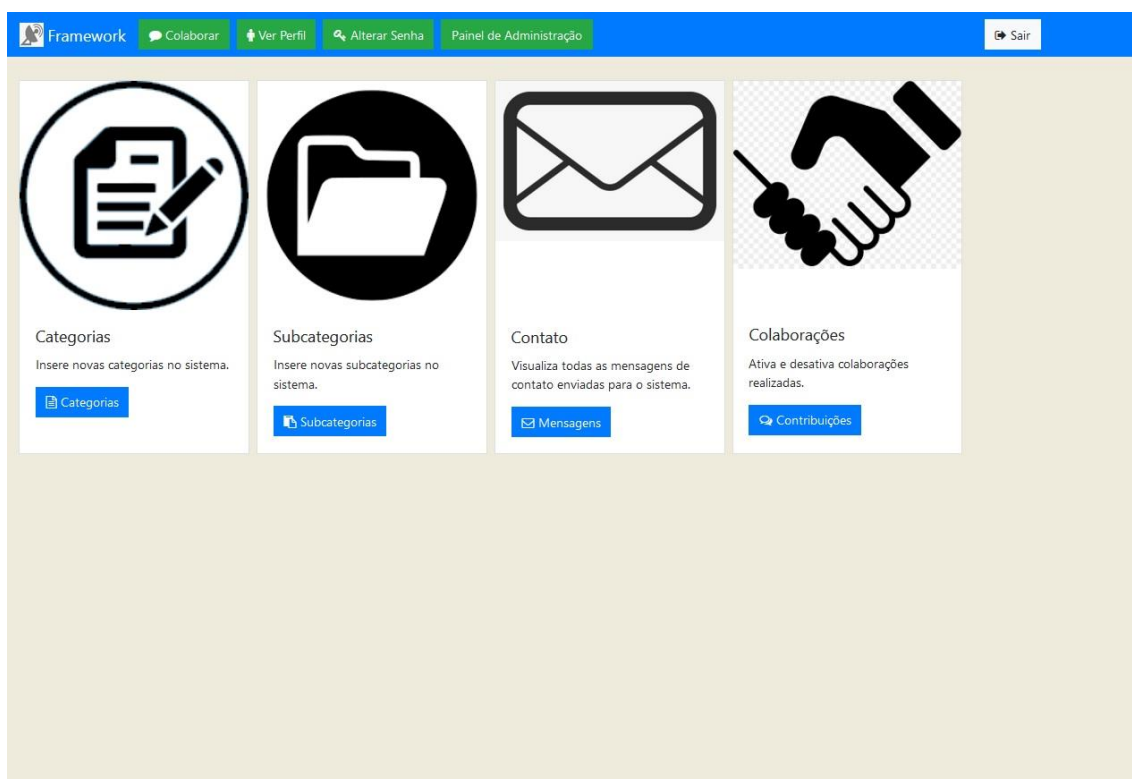


Figura 13– Interface do usuário moderador. Fonte: Autor.

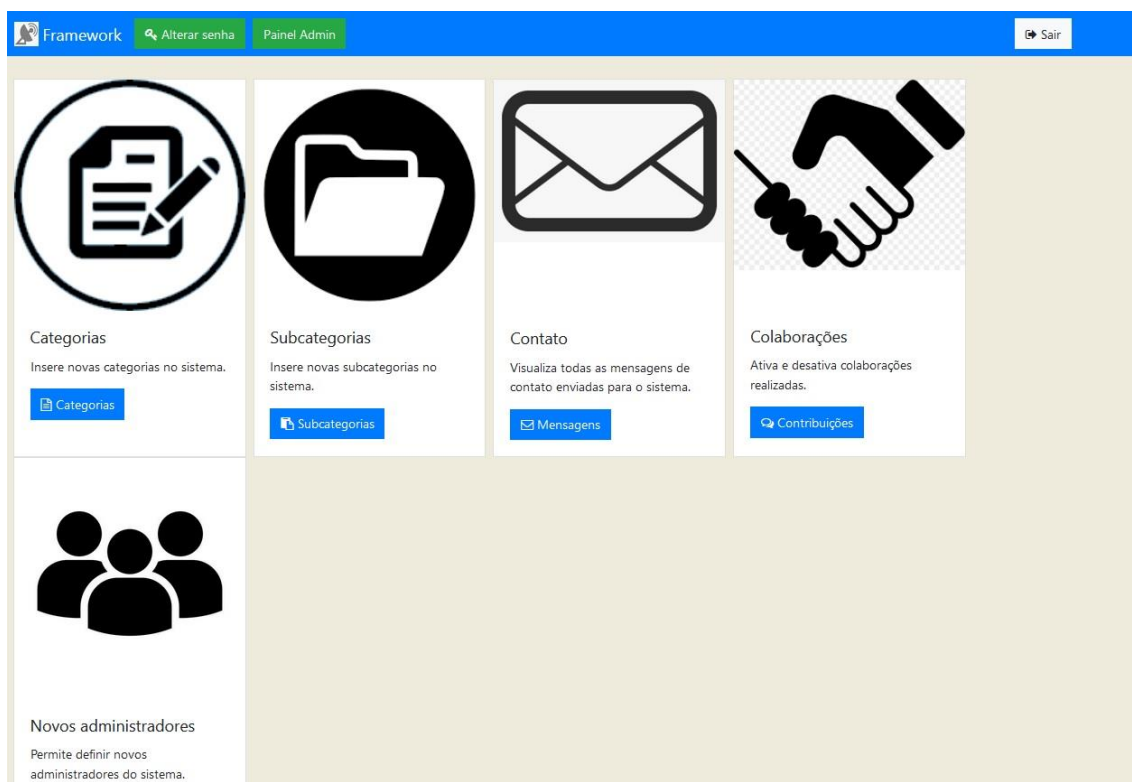


Figura 14 – Interface do usuário root (super usuário). Fonte: Autor.

## 6 Conclusões

O objetivo geral foi alcançado, uma vez que todos os objetivos específicos foram realizados, como visto anteriormente, e foi desenvolvido um produto final. Neste trabalho, desenvolveu-se um *framework* totalmente *open source* para Sistemas de Informação Geográfica que utilizam Informação Geográfica Voluntária como meio de obtenção de dados dos usuários. Sendo assim, é de extrema importância a disponibilidade de uma ferramenta desse modelo para a gestão pública, uma vez que permite a obtenção rápida de dados, enriquecendo o volume de informação disponível para o processo de tomada de decisão do gestor público. Além disso, essa ferramenta consegue transmitir uma clara visão da população em relação ao meio em que se vive, uma vez que ela é responsável pela coleta dos dados.

## 7 Referências Bibliográficas

Chapter 1 General Information. Disponível em:

<https://dev.mysql.com/doc/refman/8.0/en/introduction.html>. Acessado em: 05/03/2021

Leaflet. Disponível em: <https://leafletjs.com/>. Acessado em: 05/03/2021

GEOJSON. Disponível em: <https://geojson.org/>. Acessado em: 06/03/2021

Uma visão geral do HTTP. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Overview>. Acessado em: 06/03/2021

Métodos de requisição HTTP. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Methods>. Acessado em: 06/03/2021

INTEGRAÇÃO REST X SOAP. Disponível em: <https://www.redhat.com/pt-br/topics/integration/whats-the-difference-between-soap-rest>. Acessado em: 07/03/2021

O que é um servidor web (web server) ?. Disponível em:

[https://developer.mozilla.org/pt-](https://developer.mozilla.org/pt-BR/docs/Learn/Common_questions/What_is_a_web_server)

[BR/docs/Learn/Common\\_questions/What\\_is\\_a\\_web\\_server](https://developer.mozilla.org/pt-BR/docs/Learn/Common_questions/What_is_a_web_server). Acessado em: 07/03/2021

O que é o XAMPP ? Disponível em:

[https://www.apachefriends.org/pt\\_br/download.html](https://www.apachefriends.org/pt_br/download.html). Acessado em: 07/03/2021

PHP. Disponível em: <https://www.php.net/>. Acessado em: 07/03/2021

Rob, Peter; Coronel, Carlos. SISTEMAS DE BANCO DE DADOS: PROJETO, IMPLEMENTAÇÃO E GERENCIAMENTO. 8ª EDIÇÃO. Cengage Learning, 26 de fevereiro de 2010

LUCIANO, Josué; ALVES, Wallison Joel Barberá. Padrão de arquitetura MVC: Model-view-controller. EPeQ Fafibe, v. 1, n. 3a, p. 102-107, 2017.

COSTA, C. J. Desenvolvimento para Web. Lusocredito, Lda, Portugal, 2007

Wazlawick, R. S. ANÁLISE E PROJETO DE SISTEMAS DE INFORMAÇÃO ORIENTADOS A OBJETOS. 2ª EDIÇÃO. Elsevier Editora Ltda., 2011

BUDHATHOKI, Nama Raj et al. Reconceptualizing the role of the user of spatial data



infrastructure. GeoJournal, v. 72, n. 3-4, p. 149-160, 2008.

DOS REIS MOREIRA, Danilo. Um estudo da tecnologia Web 2.0. Disponível em: <https://files.cercomp.ufg.br/weby/up/498/o/Danilo2009.pdf>. Acesso em: 21/11/2020

LOTI, B. S.; LISBOA, M. H. M.; SILVA, P. L.; DIAS, V. E. C.; OLIVEIRA, G. D.; MIRANDA, G. H. B.; MARTINS, G. S.; MONTEIRO, C. R.; RODRIGUES, L. F.; SPERANDIO, V. G.; MENDES, V. F.; LISBOA-FILHO, J. In: CONGRESSO BRASILEIRO DE CARTOGRAFIA (CBC), 27, 2017, Rio de Janeiro. Anais... Rio de Janeiro: IEEE/GRSS, 2017. p.1-5.

J. H. S. Câmara, L. F. M. Vegi, R. O. Pereira, Z. A. Geöcze, J. Lisboa-Filho and W. D. de Souza, "ClickOnMap: A platform for development of Volunteered Geographic Information systems," *2017 12th Iberian Conference on Information Systems and Technologies (CISTI)*, Lisbon, 2017, pp. 1-6, doi: 10.23919/CISTI.2017.7975776

Souza, W. D., Vidal Filho, J. N., Ribeiro, C. A. A. S., Lisboa Filho, J., & Oliveira, D. F. (2012). Informação Geográfica Voluntária no Pantanal: um sistema Web colaborativo utilizando a API Google Maps. *Simpósio de Geotecnologias no Pantanal, Bonito, MS. Proceedings of Simpósio de Geotecnologias no Pantanal (GeoPantanal)*, 4, 763-772.

Câmara, J. H., Lisboa-Filho, J., Pereira, R. O., Souza, W. D., & Cromptvoets, J. (2019). Avaliação dos novos métodos de qualidade de VGI da plataforma ClickOnMap. *RISTI-Revista Ibérica de Sistemas e Tecnologias de Informação*, (35), 54-69.

## **Apêndice**



## Apêndice A – Requisitos não implementados

Neste apêndice será abordado a situação em que requisitos funcionais foram levantados mas não foram implementados. Sendo assim, o requisito funcional RF17 se encaixa nessa situação.

RF17 Recuperar senha		Oculto ( )		
<b>Descrição:</b> Caso o Usuário esqueça sua senha, ele deverá ser capaz de solicitar uma nova senha.				
<b>Requisitos não funcionais</b>				
Nome	Restrição	Categoria	Desejável	Permanente
NF17.1	Após solicitar uma nova senha, o usuário deve informar o email cadastrado para que seja enviado uma nova senha para ele.	Segurança	(x)	()

Tabela 17 – Recuperar senha