MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA FARROUPILHA
***CAMPUS* SANTO ÂNGELO**

QUANTITATIVOS E ESTATÍSTICAS DE RECLAMAÇÕES MEDIDAS COM AMOSTRAGEM POR GEOLOCALIZAÇÃO.

TRABALHO DE CONCLUSÃO DE CURSO

LUIMAR MOUSQUER

Santo Ângelo, RS, Brasil.
2021

QUANTITATIVOS E ESTATÍSTICAS DE RECLAMAÇÕES MEDIDAS COM AMOSTRAGEM POR GEOLOCALIZAÇÃO.

por

Luimar Mousquer

Monografia apresentada ao Instituto Federal de Educação Ciência e Tecnologia Farroupilha, como requisito parcial para obtenção do título de Tecnólogo em Sistemas para Internet.

Orientador: Prof. Juliano Gomes Weber

Santo Ângelo, RS, Brasil

2021

Ministério da Educação
Secretaria de Educação Profissional e Tecnológica
Instituto Federal de Educação Ciência e Tecnologia Farroupilha

A Comissão Examinadora, abaixo assinada,
aprova a Monografia

QUANTITATIVOS E ESTATÍSTICAS DE RECLAMAÇÕES MEDIDAS COM AMOSTRAGEM POR GEOLOCALIZAÇÃO.

elaborada por
Luimar Mousquer

como requisito parcial para obtenção do título de
Tecnólogo em Sistema para Internet

COMISÃO EXAMINADORA

_____
Juliano Gomes Weber, Me.
(Presidente/Orientador)

_____
Andreia Pereira, Me.

_____
Fábio Weber Albiero, **Me.**

Conceito Final: _____

Santo Ângelo, 25 de fevereiro de 2021.

EPÍGRAFE

A tecnologia moderna é capaz de realizar a produção sem emprego. Mas a economia moderna não consegue inventar o consumo sem salário.
(Herbert de Souza)

AGRADECIMENTOS

Em primeiro lugar, a Deus por ter me mantido com saúde durante este projeto de pesquisa me dando e forças para chegar ao objetivo. Sou grato a família pelo incentivo aos estudos e pelo apoio. Ao meu professor orientador pelas valiosas contribuições dadas durante todo o processo. A todos os professores que sempre proporcionaram um ensino de alta qualidade. Aos meus colegas do curso de Sistemas para Internet pelas trocas de ideias e ajuda mútua. Juntos conseguimos avançar e finalizar o curso, superando as dificuldades.

RESUMO

QUANTITATIVOS E ESTATÍSTICAS DE RECLAMAÇÕES MEDIDAS COM AMOSTRAGEM POR GEOLOCALIZAÇÃO.

AUTOR(A): LUIMAR MOUSQUER

ORIENTADOR(A): Juliano Gomes Weber

Data e Local da Defesa: Santo Ângelo, 25 de fevereiro de 2021.

Este trabalho apresenta um modelo estrutural para os elementos que compõem o uso da "linguagem de programação". Por meio desse modelo linguístico, procurou-se ressaltar a importância do papel das disciplinas projetuais nas atividades que caracterizam a prática de projeto de software bem como a formação humanística de programadores com base nos aspectos pedagógicos, nos tópicos de estudo e conteúdos básicos aplicáveis e profissionalizantes a todos os cursos de programadores, recomendados por comissão, do MEC, de especialistas de ensino em programação. O modelo da linguagem de engenharia – derivado de modelo para a linguagem verbal – é tomado como guia e referência para avaliar currículo pleno de curso de sistemas para internet, seja com respeito às diretrizes curriculares, às cargas horárias de disciplinas teóricas e práticas e à proporcionalidade de matérias do currículo relacionadas com as três distintas, mas interligadas, áreas do conhecimento: Ciências, Humanidades e Tecnologias. Este trabalho tem o objetivo de realizar o desenvolvimento de uma amostragem por localização completa por interface web, possibilitando assim gerenciar suas reclamações sem estar dependente de uma planilha Excel. Para isso foram utilizadas tecnologias como Java SDK, PHP MYSQL, HTML, CSS e JavaScript e frameworks como primefaces, Jasperreport, NetBeans.
Como resultado foi desenvolvido uma interface web onde é possível emitir relatórios e ter uma visão mais amigável do processo de trabalho.

Palavras-chave: Java, Java Web, desenvolvimento de sistemas, geolocalização.

ABSTRACT

This work presents a structural model for the elements that make up the use of the “programming language”. Through this linguistic model, we sought to emphasize the importance of the role of project disciplines in the activities that characterize the practice of software design as well as the humanistic training of programmers based on pedagogical aspects, study topics and basic applicable and professional content. to all programmers' courses, recommended by commission, from MEC, from specialists in teaching programming. The engineering language model - derived from the model for verbal language - is taken as a guide and reference to evaluate the full curriculum of systems course for internet, either with respect to curricular guidelines, the workloads of theoretical and practical disciplines and proportionality curriculum subjects related to the three distinct, but interconnected, areas of knowledge: Sciences, Humanities and Technologies. This work has the objective of carrying out the development of a sampling by complete location by web interface, thus making it possible to manage your complaints without being dependent on an Excel spreadsheet. For this, technologies such as Java SDK, PHP MYSQL, HTML, CSS and JavaScript and frameworks such as primefaces, Jasperreport, NetBeans were used.

As a result, a web interface was developed where it is possible to issue reports and have a more friendly view of the work process.

Keywords: Java, Java Web, systems development, geolocation.

LISTA DE TABELAS

Tabela 1 - Especificação casa de uso Cadastrar Cliente 43
Tabela 2 - Especificação caso de uso Importar Dados 44
Tabela 3 - Especificação caso de uso Levantamento Clientes e Reclamações 45
Tabela 4 - Especificação caso de uso Gerar Relatório Geral 46
Tabela 5 - Especificação caso de uso Gerar Mapa 47

LISTA DE FIGURAS

Figura 1 - Planilha Excel 14
Figura 2 - Mapa com os endereços 14
Figura 3 - Onde inserir dados planilha 15
Figura 4 - Detalhamento para receber as URL 16
Figura 5 – Resultado do mapeamento da planilha 16
Figura 6 - Onde insere dados na planilha 17
Figura 7 - Resultado após clicar em Faça Mapa 17
Figura 8 - Detalhamento para receber a URL 18
Figura 9 - Triangulação satélite GPS 20
Figura 10 - Funcionamento do AGPS 20
Figura 11 - Simbologia do JAVA 22
Figura 12 - NetBeans 23
Figura 13 - Java Server Faces (JSF) 23
Figura 14 - Layout básico do primefaces 24
Figura 15 - Glassfish 24
Figura 16 - Exemplo básico da função do Hibernate 25
Figura 17 - Código básico Javascript 26
Figura 18 - Ilustração do padrão MVC 26
Figura 19 - Ilustração do Jasperreport 27
Figura 26 - Modelo Entidade Relacionamento 32

LISTA DE ABREVIATURAS E SIGLAS

API	Interface de Programação de Aplicações
A-GPS	Sistema de Posicionamento Global Assistido
BCNF	Forma Normal de Boyce-Codd
CRUD	Creat, Read, Update, Delete
GPS	Sistema de Posicionamento Global
HTML	Hyper Text Markup Language
IDE	Ambiente de Desenvolvimento Integrado
JVM	Máquina Virtual Java
ORM	Mapeamento de Objeto Relacional
UML	Linguagem Unificada de Modelagem
XML	Extensível Linguagem de Marcação

SUMÁRIO

1 INTRODUÇÃO 13
1.1 OBJETIVOS 14
1.1.1 Objetivos geral 14
1.1.2 Objetivos específicos 14
1.2 TRABALHOS CORRELATOS 15
1.2.1 Google My Maps 15
1.2.2 Batchgeo 16
1.2.3 Easymapmaker 18
1.3 FUNDAMENTAÇÃO TEÓRICA 20
1.3.1 Localização 20
1.3.2 Normalização do banco de dados 22
1.3.3 Tecnologias usadas 23
1.3.3.1 JAVA 23
1.3.3.2 NetBeans IDE 24
1.3.3.3 Java Server Faces 24
1.3.3.4 Primefaces 25
1.3.3.5 Glassfish 25
1.3.3.6 Hibernate 26
1.3.3.7 JavaScript 26
1.3.3.8 (MVC) 27
1.3.3.9 Jasperreport 29
2 DESENVOLVIMENTO 30
2.1 API DO GOOGLE 30
2.1.1 Geocodificação 30
2.1.2 Geolocalização 30
2.2 MODELAGEM DO BANCO DE DADOS 31
2.2.1 Diagrama de Caso de Uso 31
2.2.2 Especificações de Caso de Uso 31
2.2.3 Diagrama de Sequência 31
2.3 BANCO DE DADOS 33
2.3.1 Modelo Entidade Relacionamento 33
2.3.2 Modelo lógico 33

2.3.3 Modelo físico. 34
2.4 DIAGRAMA DE CLASSES 35
2.5 PROGRAMAÇÃO DO SISTEMA 35
2.6 TELAS DO SISTEMA 36
2.6.1 Tela de apresentação do projeto. 36
3 RESULTADO E DISCUSSÕES 40
3.1 QUESTIONÁRIO DE AVALIAÇÃO E TESTES DO SISTEMA 40
3.1.1 Resultado do sistema 41
4 CONSIDERAÇÕES FINAIS 41
REFERÊNCIAS 42
APÊNDICE A - Imagem do Diagrama de Caso de Uso 43
APÊNDICE B - Caso de Uso Especificações 44
APÊNDICE C - Diagramas de Sequências 49
APÊNDICE D - Diagrama de Classes 52
APÊNDICE E - Código Fonte 53

1 INTRODUÇÃO

O mundo das tecnologias da informação é enorme trazendo consigo o desenvolvimento de sistemas, onde o profissional pode vir a escolher a melhorar o seu ambiente de trabalho complementando o que já existe, transformando uma planilha cheia de dados em algo visual e intuitivo, ou seja, transformando dados em informações.

O sistema de busca por geolocalização tem como foco principal a conversão de dados de planilha Excel para a estrutura do software onde será montado indicativo de quantidade e volume de clientes e bilhetes de reclamações de clientes, em mapa de sua região.

Assim, tem em sua função facilitar a consulta de rotas para endereços clientes específicos, onde simplificará a navegação para o usuário. Além do benefício citado acima, o software poderá emitir relatório sobre a quantidade de serviços de internet contratado por cliente em sua região de cobertura, tendo busca por nomes, endereços e designações (identificação do serviço).

O sistema de desenvolvimento é um software livre, que está implementado inteiramente com ferramentas livres, usando tecnologias atuais, que poderá ser atualizado conforme as necessidades, que um cliente específico desejar o uso do software.

As tecnologias que estarão sendo usadas, estão a linguagem UML, NetBeans, MySQL, JSF, Web Service, Java, Hibernate, que serão detalhadas no decorrer do trabalho.

1.1 OBJETIVOS

1.1.1 Objetivos geral

Criar um sistema de geolocalização, transformando uma planilha com dados estatísticos, em um mapa que faça referência a locais que necessitam mais atenção, com isso mostrando de forma visual onde teria que reforçar as ações técnicas.

1.1.2 Objetivos específicos

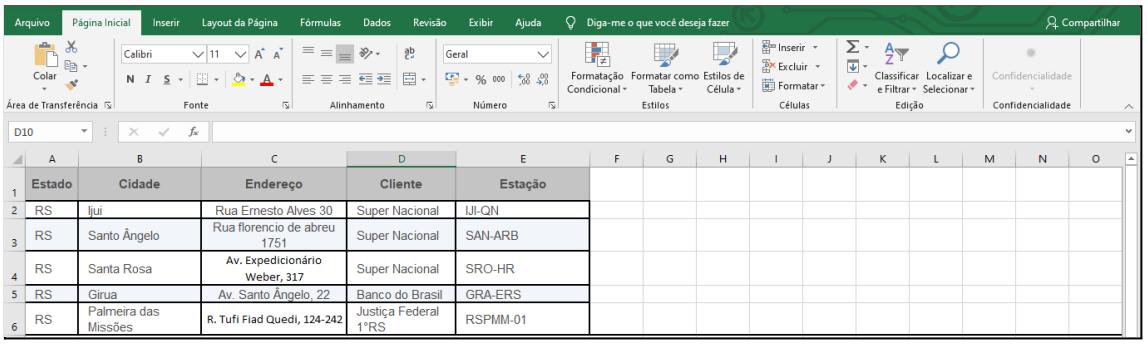
- Cadastrar clientes e seus endereços;
- Popular o mapa da região com seus clientes;
- Destacar clientes com chamados repetidos;
- Cadastrar tecnologias de enlace;
- Cadastrar serviços
- Gerar gráfico estatístico;
- Gerar relatório de cliente com chamados repetidos;
- Coletar dados de planilha Excel;
- Apresentar os dados coletados;

1.2 TRABALHOS CORRELATOS

1.2.1 Google My Maps

O sistema permite importar archivos CSV, TSV, KMZ, KML ou XLSX, com estas facilidades se monta uma planilha externa ou no google planilha com seus dados aonde com a primeira opção, pode se fazer o upload para software definindo assim seus locais no mapa com as descrições que escolher, pois todas as colunas aparecem como opção de escolha para retornar na tela os dados.

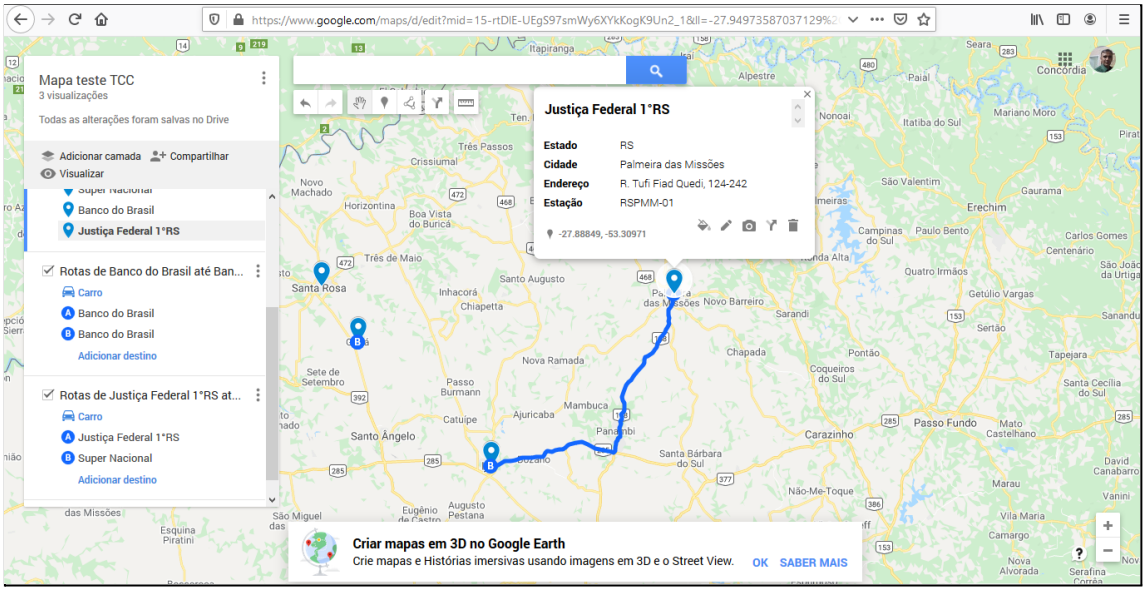
Figura 1 - Planilha Excel



Estado	Cidade	Endereço	Cliente	Estação
RS	Ijui	Rua Ernesto Alves 30	Super Nacional	IJI-QN
RS	Santo Ângelo	Rua florencio de abreu 1751	Super Nacional	SAN-ARB
RS	Santa Rosa	Av. Expedicionário Weber, 317	Super Nacional	SRO-HR
RS	Girua	Av. Santo Ângelo, 22	Banco do Brasil	GRA-ERS
RS	Palmeira das Missões	R. Tufi Fiad Quedi, 124-242	Justiça Federal 1°RS	RSPMM-01

Fonte: Compilação do autor.

Figura 2 - Mapa com os endereços



7 Fonte: Google My Maps.

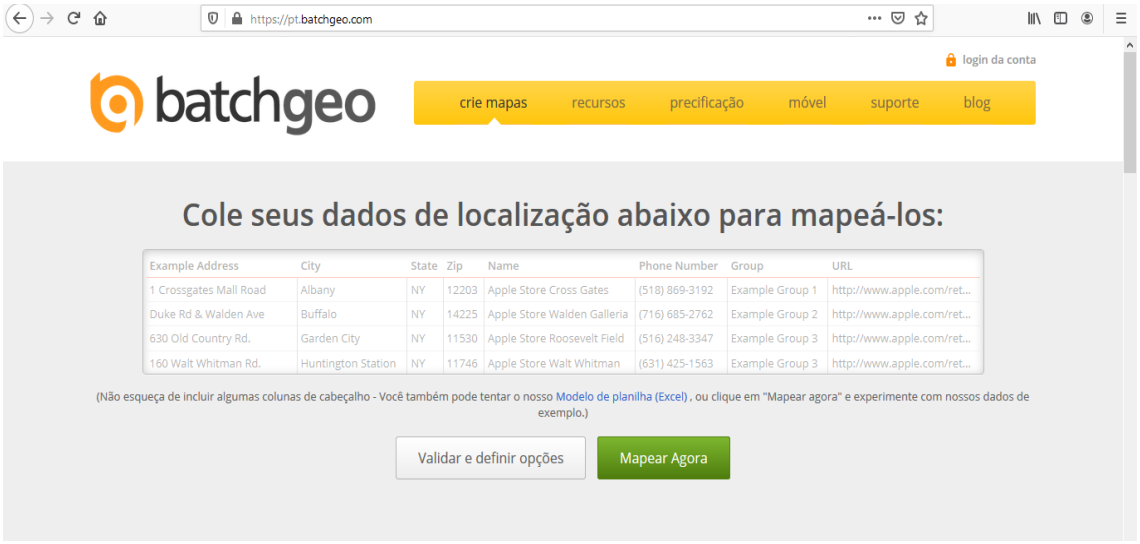
1.2.2 Batchgeo

Sistema que permite mapear dados da sua planilha do Excel (XLS). clientes da sua empresa, locais de férias da família, endereço de parentes, podendo visualizar rapidamente os locais num mapa e, em seguida, compartilhá-los. Quando seus dados estiverem em um mapa, você poderá explorá-los clicando nos marcadores, que mostrarão informações adicionais sobre o local. Esse mesmo processo também funciona com o Números e o Google Docs.

Apenas uma coluna precisa conter dados de localização. Por exemplo, você pode ter apenas CEP ou nomes de cidades, ou você pode ter um endereço completo. Quanto mais dados você tiver, melhor.

Nesse sistema tem a facilidade de copiar e colar a planilha Excel com dados necessários para definir seus pontos na geolocalização podendo criar um mapa particular ou público.

Figura 3 - Onde inserir dados planilha



Fonte: Https://pt.batchgeo.com/

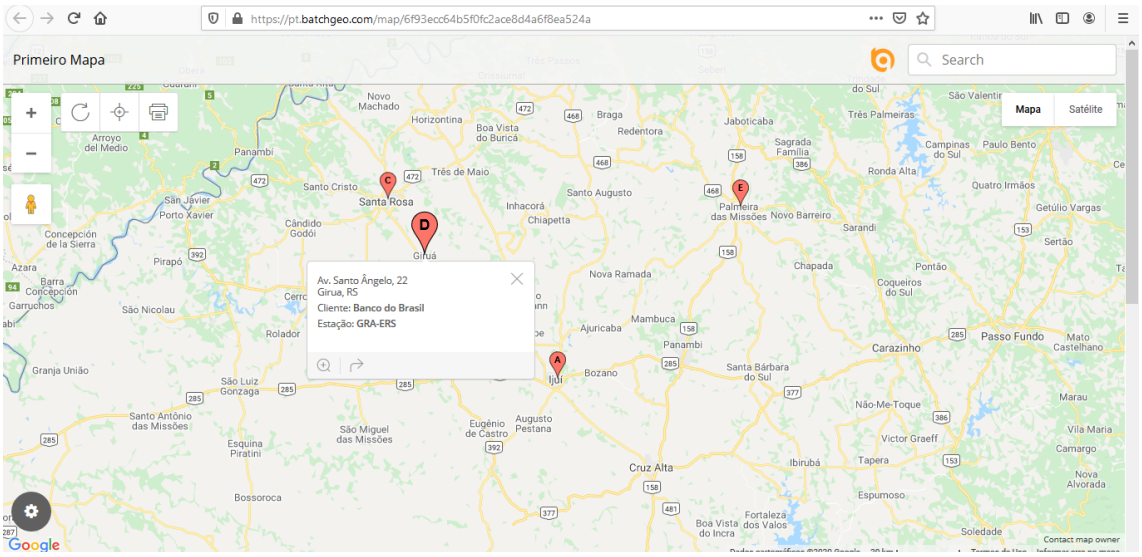
Após validar e definir as opções, abre o campo de salvar o mapa, onde você pode também, escolher em deixar público ou privado.

Figura 4 - Detalhamento para receber as URL

Fonte: Https://pt.batchgeo.com.

Salvamos o mapa e recebemos no e-mail cadastrado duas URL, sendo um particular que tem a possibilidade de editar.

Figura 5 – Resultado do mapeamento da planilha

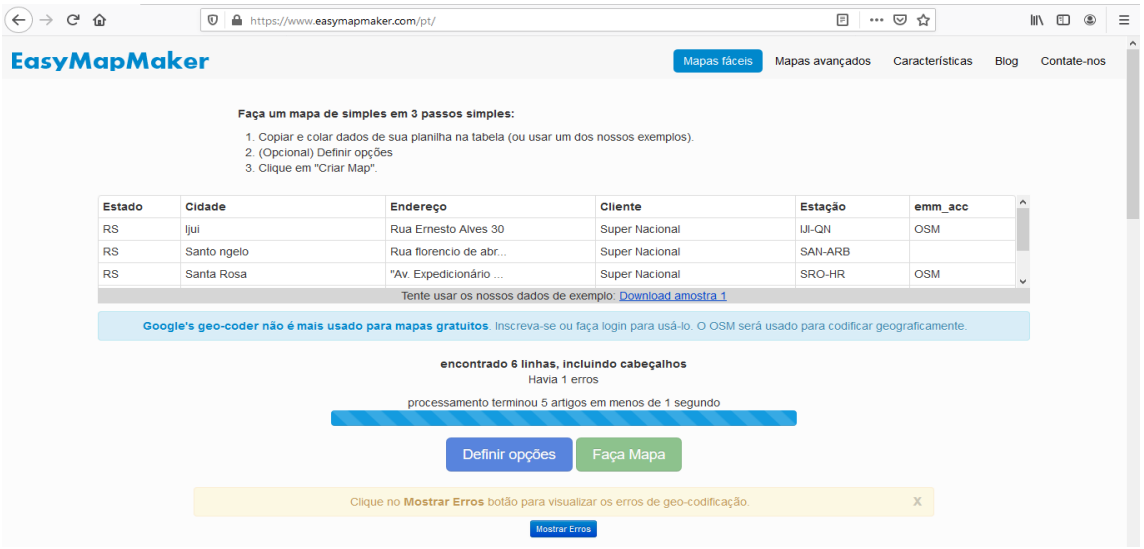


Fonte: Https://pt.batchgeo.com/.

1.2.3 Easymapmaker

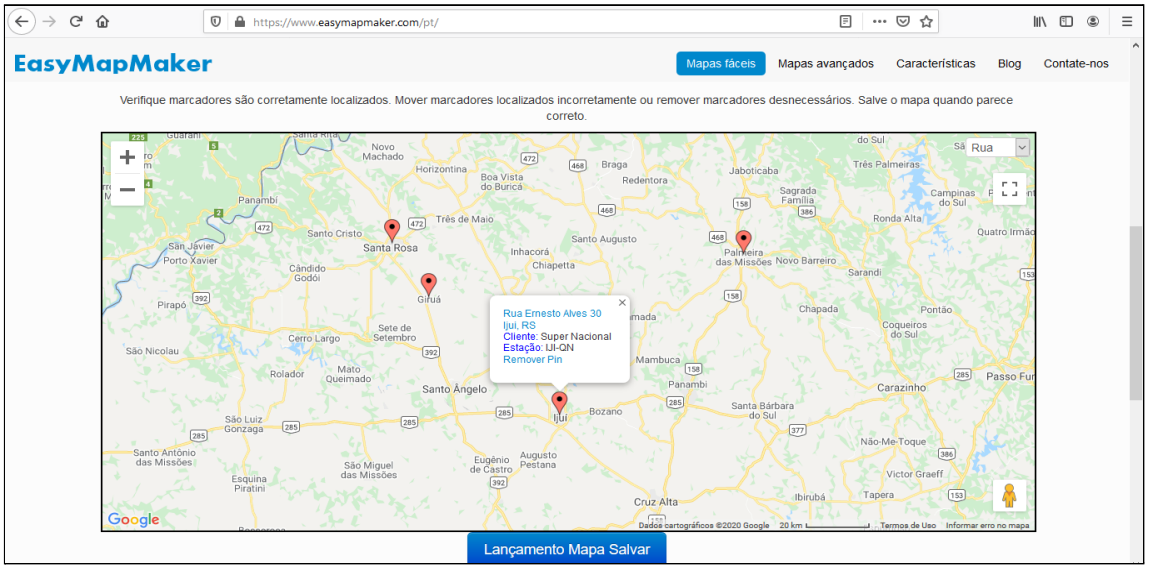
Sistema semelhante ao batchgeo tendo os mesmos princípios, usado para mapear dados da planilha em um mapa do google. Mapear e planejar manualmente vários locais em um mapa. O primeiro passo é copiar a planilha com dados incluindo os cabeçalhos das colunas colando no local indicado próximo passo executando definindo as opções ou fazendo o mapa.

Figura 6 - Onde insere dados na planilha



Fonte: https: Https://www.easymapmaker.com/pt/.

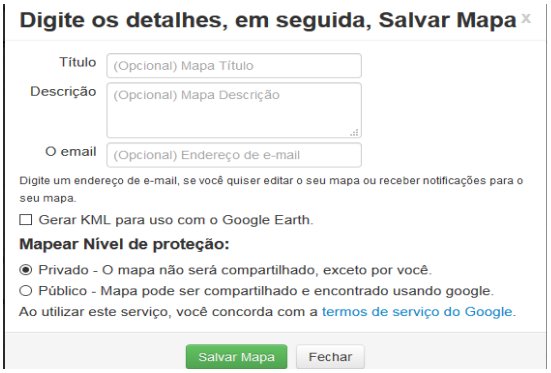
Figura 7 - Resultado após clicar em Faça Mapa



Fonte: https: Https://www.easymapmaker.com/pt/.

Salvamos o mapa e recebemos no e-mail cadastrado duas URL, sendo um particular que tem a possibilidade de editar.

Figura 8 - Detalhamento para receber a URL



Fonte: https: Https://www.easymapmaker.com/pt/.

1.3 FUNDAMENTAÇÃO TEÓRICA

Neste projeto foram detalhados os conceitos relevantes na apresentação deste trabalho. Para facilitar a compreensão de pontos importantes no desenvolvimento do projeto.

1.3.1 Localização

É a identificação ou estimativa da localização geográfica real de um país, estado, cidade, rua, e até o horário onde esse objeto se encontra, que através de um dispositivo com conexão à internet, radio frequência, e o GPS é capaz de gerar um conjunto de coordenadas geográficas, identificando a localização solicitada.

O GPS funciona a partir de uma rede de 24 satélites que ficam distribuídos em seis planos, próximos à órbita do planeta Terra. Estes satélites enviam sinais para o receptor (o aparelho de GPS, Smartphones), atualmente existem quatro sistemas que permitem a navegação por satélite, o GPS americano e o **GLONASS** russo, e outros dois sistemas que estão em fase final de desenvolvimento, o Galileo, da União Européia, e o Compass, da China.

- A triangulação

Os GPS usam um sistema chamado de triangulação para determinar a localização do receptor na Terra. A triangulação funciona da seguinte forma: três satélites enviam o sinal para o receptor, que calcula quanto tempo cada sinal demorou a chegar nele. Além da sua localização terrestre, o receptor GPS também consegue saber a altura do receptor em relação ao nível do mar, porém para isso é necessário um quarto satélite. Para entendermos melhor, veja a imagem abaixo e entenda como funciona o conceito.

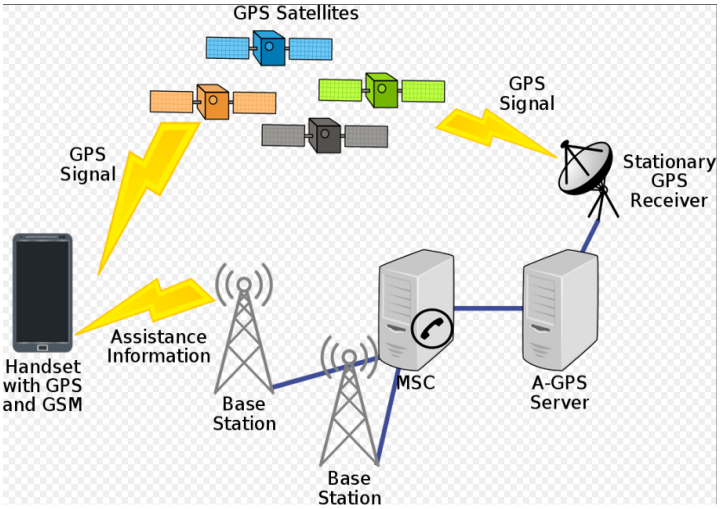
Figura 9 - Triangulação satélite GPS

Fonte:Oficinadanet.com.br.

O A-GPS (de Assisted GPS, ou GPS Assistido) é um sistema voltado a melhorar a precisão dos dados do GPS, utilizando para isso a rede de telefonia móvel terrestre e por isso mesmo, está presente em quase todos os smartphones do mercado.

A solução do A-GPS é utilizar até quatro antenas de telefonia para atualizar os dados de posição de modo a agilizar a triangulação do sinal, diminuindo o tempo de comunicação entre os satélites e oferecendo informações adicionais como a previsão do tempo, diretamente pelo pacote de dados.

Figura 10 - Funcionamento do AGPS



Fonte: Ffw Com.br.

1.3.2 Normalização do banco de dados

Normalização é uma aplicação de regras a todas as tabelas do banco de dados projetando a forma como as informações serão armazenadas, eliminando, ou minimizando, a redundância no banco, assim evitando falhas no projeto.

> O processo de normalização compreende o uso de um conjunto de regras, chamadas de formas normais. Ao analisarmos o banco de dados e verificarmos que ele respeita as regras da primeira forma normal, então podemos dizer que o banco está na “primeira forma normal”. Caso o banco respeite as primeiras três regras, então ele está na “terceira forma normal”. Mesmo existindo mais conjuntos de regras para outros níveis de normalização, a terceira forma normal é considerada o nível mínimo necessário para grande parte das aplicações. [Microsoft 2007]

- Primeira forma normal

Uma relação está na primeira forma normal quando todos os atributos contêm apenas um valor correspondente, a tabela não deve conter grupos repetidos e nem atributos com mais de um valor.

- Segunda forma normal

A tabela para estar a segunda forma normal obrigatoriamente tem que estar na primeira forma os atributos que não são chaves vão ter que depender unicamente da chave primária e não do seu valor.

- Terceira forma normal

Se analisarmos uma tabela e não encontrarmos um atributo não chave dependente de outro atributo não chave, podemos dizer que a entidade em questão está na terceira forma normal, elimine os campos que não dependem da chave.

- Quarta forma normal

Também chamado de Boyce Codd Normal (BCNF), para estar na quarta forma normal, tem que estar na terceira forma normal e não existir dependências multivaloradas entre seus atributos, ou seja, atributos que se repetem em relação a chave primária, gerando redundância nas linhas da entidade.

1.3.3 Tecnologias usadas

Neste capítulo abordaremos as ferramentas e técnicas para desenvolvimento do sistema proposto com as linguagens JAVA para web com uso também do JSF (Java Server Faces). A ferramenta utilizada para o desenvolvimento em JAVA foi o NetBeans IDE, que proporciona uma programação mais eficiente proporcionada por suas ferramentas internas que visam melhorar alguns aspectos na programação, dando assim um desempenho maior no momento do desenvolvimento. Além de por meio dele ser possível a utilizar o JSF, que nada mais é que um framework que visa a elaboração rápida de interfaces para sistemas web.

1.3.3.1 JAVA uma linguagem de programação foi desenvolvida por James Gosling, juntamente com outros colaboradores, no início da década de 1990, na empresa Sun Microsystems. A linguagem de programação Java é orientada a objetos, onde comportamento dos objetos são determinados por classes, e

compilada em *bytecode* (as instruções são executadas através de uma Máquina Virtual Java - JVM). A sintaxe da linguagem Java é similar às linguagens C e C + +, A linguagem Java é a mais importante da Plataforma Java, que agora pertence à empresa Oracle.

Figura 11 - Simbologia do JAVA

Fonte: Número Real.com.br.

1.3.3.2 NetBeans IDE foi iniciado em 1996 por dois estudantes tchecos na Universidade de Charles, em Praga, quando a linguagem de programação Java ainda não era tão popular como atualmente. Em 1999 com o nome de NetBeans Developer X2 Nessa época a empresa Sun Microsystems havia desistido de sua IDE Java Workshop e adquiriu o projeto NetBeans Developer X2 incorporando-o à sua linha de softwares. plataforma Open Source onde comunidade de desenvolvedores que utilizam e contribuem com o projeto que não parou de crescer, tornando-se uma das IDEs mais populares.

Figura 12 - NetBeans

Fonte: visualdicas.com.br

.

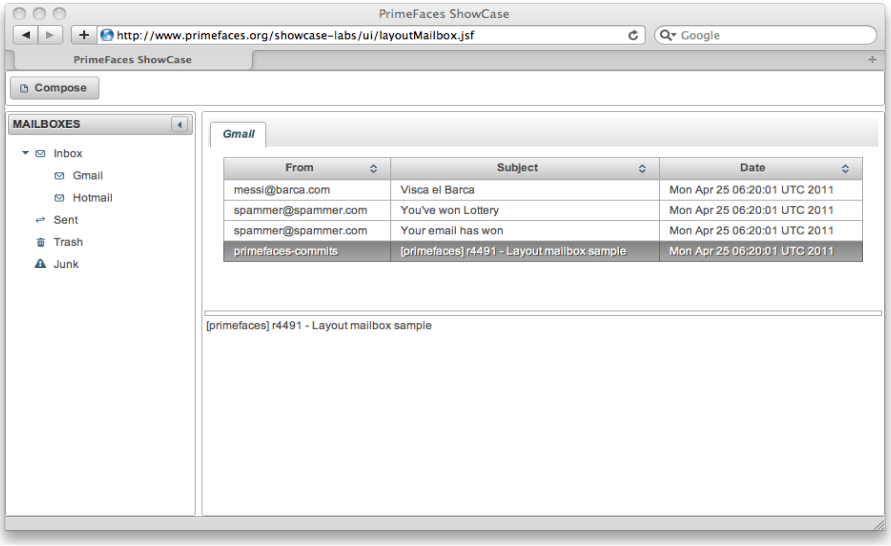
1.3.3.3 Java Server Faces, para quem nunca ouviu falar de JSF, é um framework onde é efetuada a construção de interfaces de usuários para sistema web, colocando componentes em um formulário e ligando os a objetos Java, sendo assim ele faz a separação entre a lógica e regras de negócio a navegação e conexões com serviços externos seguindo o padrão MVC.

Figura 13 - Java Server Faces (JSF)

Fonte: Devmedia.com.br.

1.3.3.4 Primefaces é um framework usado para desenvolver rapidamente aplicações sofisticadas ou no desenvolvimento de sites padrão, biblioteca de componentes de interface gráfica para as aplicações web baseadas em JSF.

Figura 14 - Layout básico do primefaces

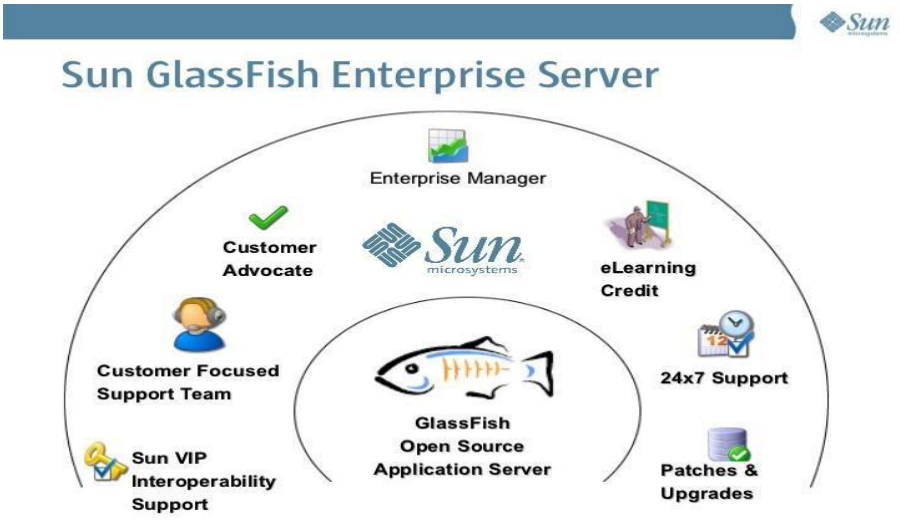


Fonte: Número Real.com.br.

1.3.3.5 Glassfish é um servidor de aplicação Open-Source. O projeto foi iniciado pela Sun Microsystems e agora é patrocinado pela Oracle Corporation.

Suporta todas as aplicações JAVA, permitindo aos desenvolvedores criar aplicativos que se integram com tecnologias legadas.

Figura 15 - Glassfish

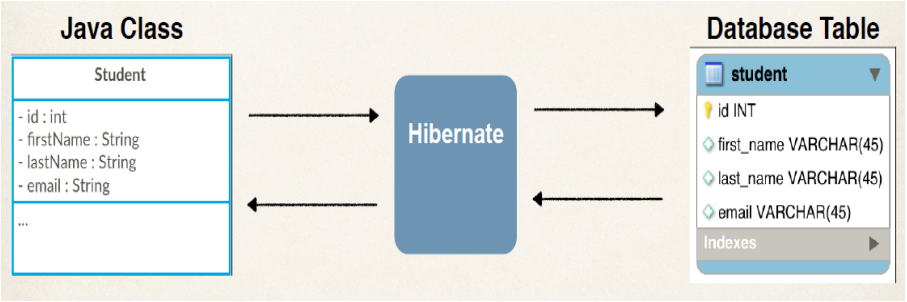


Fonte: Slideshare.net

1.3.3.6 Hibernate lançado em 2001, o Hibernate foi criado por desenvolvedores Java, juntamente com a liderança de Gavin King, fundador do projeto Hibernate É um framework ORM, com ele você faz operações CRUD (criação, leitura, atualização e exclui) em objetos no banco de dados,

"O Hibernate é um framework para mapeamento objeto-relacional para linguagem Java. Na prática, ele é um conjunto de classes, interfaces e arquivos de configuração pré acabados que permitem a criação de uma camada de serviço capaz de abstrair a existência do banco de dados para sistemas Java" [MRACK 2006].

Figura 16 - Exemplo básico da função do Hibernate



Fonte: dzone.com.

1.3.3.7 JavaScript é a linguagem de programação da Web. A ampla maioria dos sites modernos usa JavaScript e todos os navegadores modernos, criada em 1995 baseada em scripts que inicialmente era usado apenas para aplicação de efeitos visuais em páginas da internet, e com as evoluções tornou uma das linguagens mais utilizadas no mundo.

Na verdade, o nome "JavaScript" é um pouco enganoso. A não ser pela semelhança sintática superficial, JavaScript é completamente diferente da linguagem de programação Java. E JavaScript já deixou para trás suas raízes como linguagem de script há muito tempo, tornando-se uma linguagem de uso geral robusta e eficiente. [David Flanagan 2013]

Figura 17 - Código básico Javascript

Fonte: sujeito programador.com.

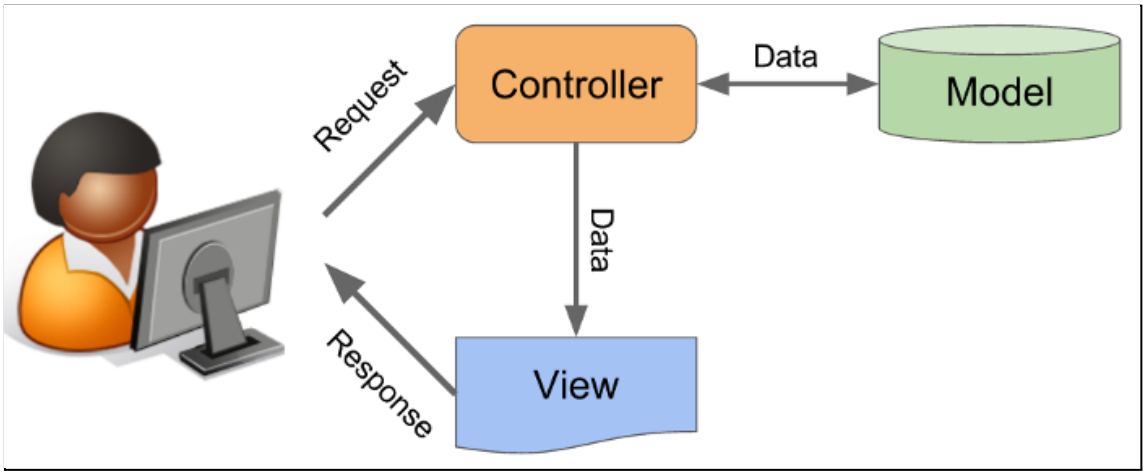
1.3.3.8 (MVC) **Model, View, Controller** padrão de arquitetura de software, separando sua aplicação em 3 camadas. A camada de interação do usuário (view), a camada de manipulação dos dados (modelo) e a camada de controle **(**controller).

Modelo parte da manipulação de dados, responsável pela leitura e escrita de dados, e também de suas validações**.**

View camada de interação com o usuário, realiza a exibição dos dados, sendo ela por meio de um html ou xml.

Controller recebe todas as requisições do usuário, controlando qual modelo usar e qual view será mostrado ao usuário.

Figura 18 - Ilustração do padrão MVC



Fonte: The model-view-controller pattern | Dave Hollingworth

1.3.3.9 Jasperreport é um framework open source inteiramente escrito em Java, tendo sido criado e mantido pela organização JasperForge. Com a sua poderosa API, é possível gerar e exportar relatórios altamente dinâmicos em PDF, XLSX, DOC, para aplicações Java, em uma forma intuitiva e escrevendo poucas linhas de código.

Figura 19 - Ilustração do Jasperreport

Fonte: compilação do autor

2 DESENVOLVIMENTO

Além das ferramentas mencionadas como Java, Java Web, Javascript, Frameworks e Jasperreports , foi usada as API do Google, Geocodificação e Geolocalização complementando o projeto e dando funcionalidades importantes.

2.1 API GOOGLE

2.1.1 Geocodificação ou Geocoding realiza a conversão dos endereços físicos como o endereço RS 218 km 5 Indubras Santo Ângelo RS em coordenadas geográficas -28.26680 e -54.22489 também podendo realizar a operação ao inverso, tornando possível a localização visual no mapa.

2.1.2 Geolocalização ou Geolocation se utiliza da informação de latitude e longitude disponibilizado pelo sistema de Decodificação, assim localizando no mapa as localizações.

A comunicação é feita por HTTPS usando bibliotecas java. Tanto a solicitação quanto a resposta são formatadas como JSON, e o tipo de conteúdo de ambas é aplicação/json.

Antes de começar a desenvolver com a API , você precisa de um projeto, criar uma conta de faturamento e com isso, ter em mão uma chave de API(KEY) e o uso da API e as informações de faturamento (você precisa habilitar o faturamento em seu projeto), tendo a acesso a todos as APIs google com a mesma chave.

2.2 MODELAGEM DO BANCO DE DADOS

É a forma que temos para especificar as funcionalidades de um software, como os casos de uso interagem entre si no sistema e com os usuários (atores), ou seja, como as funcionalidades se relacionam umas com as outras e como serão utilizadas pelo usuário.
Segundo Booch, Rumbaugh e Jacobson (2005) existem vários elementos que auxiliam um software a obter sucesso em sua criação e um desses é a modelagem, sendo que esse elemento é uma técnica de engenharia aprovada e bem aceita. A modelagem de software é necessária para que se possa ter um conhecimento pleno de todos os itens do software. Os itens a seguir descrevem essas técnicas de modelagem.

2.2.1 Diagrama de caso de uso

O diagrama de Casos de Uso, representa as funcionalidades do sistema, onde o usuário poderá interagir com sistema realizando o cadastramento das suas especificações, onde vai ser gerado mapa de geolocalização com gráfico estático, podendo realizar levantamento, gerando relatórios. Conforme pode ser visto na imagem 20, veja o apêndice A.

2.2.2 Especificações de Casos de Uso

É a descrição de uma sequência de ações que o sistema realiza para produzir uma resposta para um ator, uma interação entre o caso de uso e o ator. Conforme pode ser visto na imagem das tabelas, veja o apêndice B.

2.2.3 Diagrama de sequência

Mostra diferentes partes do sistema trabalhando em uma sequência para se fazer algo, modelam as interações entre objetos, ilustram como as diferentes partes do sistema interagem entre si para

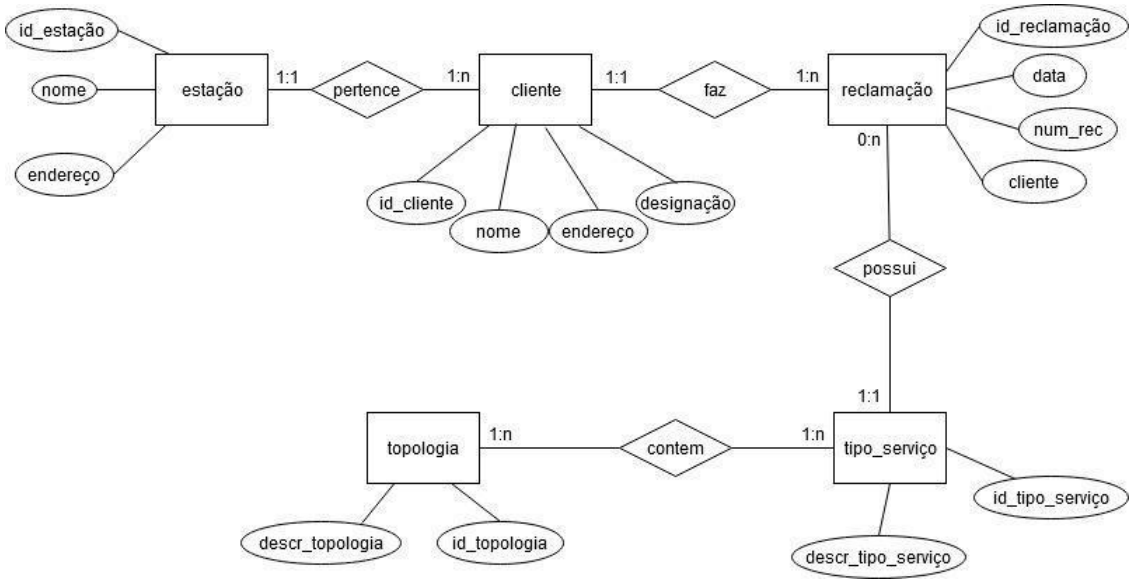
realizar a função, e a ordem em que as interações ocorrem quando um determinado caso de uso é executado. Conforme pode ser visto nos diagramas , veja o apêndice C.

2.3 BANCO DE DADOS

2.3.1 Modelo Entidade Relacionamento

O diagrama demonstrará como deve estar a relação de dados da base de BD, para que não ocorram inconsistências de informações.

Figura 26 – Modelo de ER



Fonte: Próprio autor

2.3.2 Modelo lógico

Representa a forma lógica como se dará a modelagem do sistema de Geolocalização.

Cliente (id_cliente, endereço, designação, estação, nome, id_estado)
Estação (id_estado, endereço, nome)

Reclamação (id_reclamacao, cliente, numeric, data, id_cliente, id_tipo_serviço)

Reclamação (referência tabela cliente

Id_tipo Serviço referência tabela tipo serviço

Tipo Serviço (id_tipo serviço, descr tipo serviço)

Topología (id topología, desc topología)

Tipo _serv Topología (id_tipo _serv topología, id topología, id_tipo_serviço)

Id Topologia referência tabela topologia

Id_tipo Serviço referência tabela tipo serviço

2.3.3 Modelo físico.

```
CREATE TABLE cliente (
id_cliente INT (10) PRIMARY KEY,
endereco VARCHAR (50),
designacao VARCHAR (12),
estacao VARCHAR (3),
nome VARCHAR (20),
id_estado INT (10),
FOREIGN KEY (id_estado) REFERENCES estação (id_estado)
)

CREATE TABLE reclamação (
id reclamação INT (10) PRIMARY KEY,
cliente VARCHAR (20),
num_recl VARCHAR (6),
data DATE,
id_cliente INT (10),
id_tipo serviço INT (10),
FOREIGN KEY (id_cliente) REFERENCES cliente (id_cliente),
FOREIGN KEY (id_tipo serviço) REFERENCES tipo serviço (id_tipo serviço)
)
CREATE TABLE estação (
id_estado INT (10) PRIMARY KEY,
endereco VARCHAR (50),
nome VARCHAR (50)
)

CREATE TABLE tiposervico (
id_tipo serviço INT (10) PRIMARY KEY,
descr tipo serviço VARCHAR (20)
)
```

```
CREATE TABLE topologia (
id topologia INT (10) PRIMARY KEY,
descr topologia VARCHAR (20)
)

CREATE TABLE contém (
id_tipo serviço INT (10),
id topologia INT (10),
FOREIGN KEY (id_tipo serviço) REFERENCES tipo serviço (id_tipo serviço),
FOREIGN KEY (KEY () REFERENCES topologia (id topologia)
)
```

2.4 DIAGRAMA DE CLASSES

No diagrama representam as classes as classes os atributos e os métodos utilizados no desenvolvimento do sistema, pois mapeiam de forma clara a estrutura do sistema ao modelar suas classes, seus atributos, operações e relações entre objetos. Conforme pode ser visto no diagrama, veja o apêndice D.

2.5 PROGRAMAÇÃO DO SISTEMA

Foi adotado a linguagem Java Web com FrameWork primefaces para cadastros, adotou-se o padrão MVC(model, view, control), sendo utilizado JavaScript para chamar a view, com o mapa da plotagem dos clientes.

Também é utilizado excel, pois temos como referência uma planilha de dados reais, onde o sistema chama o excel, para com isso possamos alimentá-la com as informações e fazer importação dos dados, assim cadastrando os bilhetes de reclamações, e pode cadastrar cliente automaticamente sem a necessidade de cadastrar um por um. Conforme pode ser visto o código fonte, veja o apêndice E.

2.6 TELAS DO SISTEMA

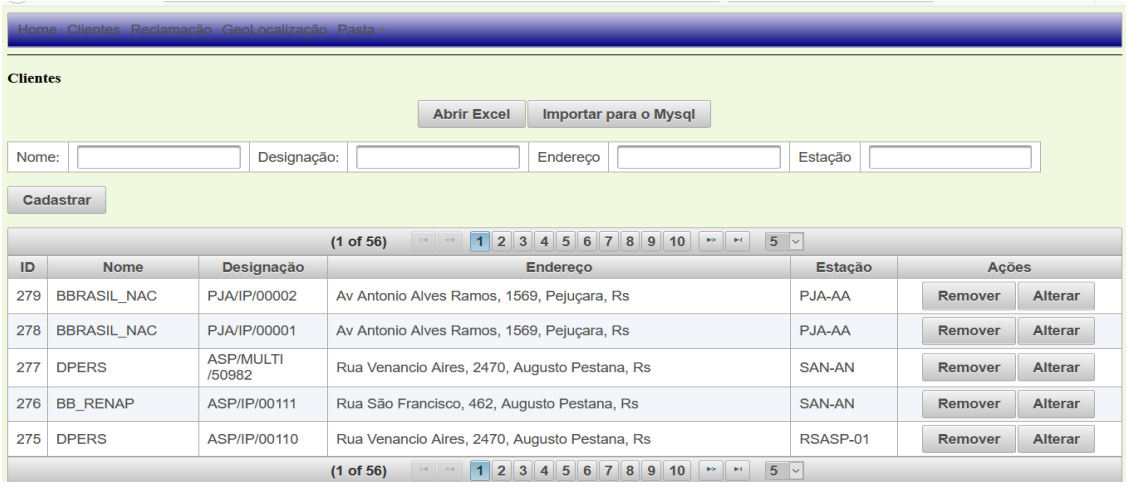
2.6.1 Tela de apresentação do projeto.

Figura 28 - Home.

Fonte: Próprio autor

2.6.2 Tela onde será feito o cadastramento do cliente, também ficará como base para dar referência a geolocalização, também poderemos percorrer a lista e verificar cada cliente.

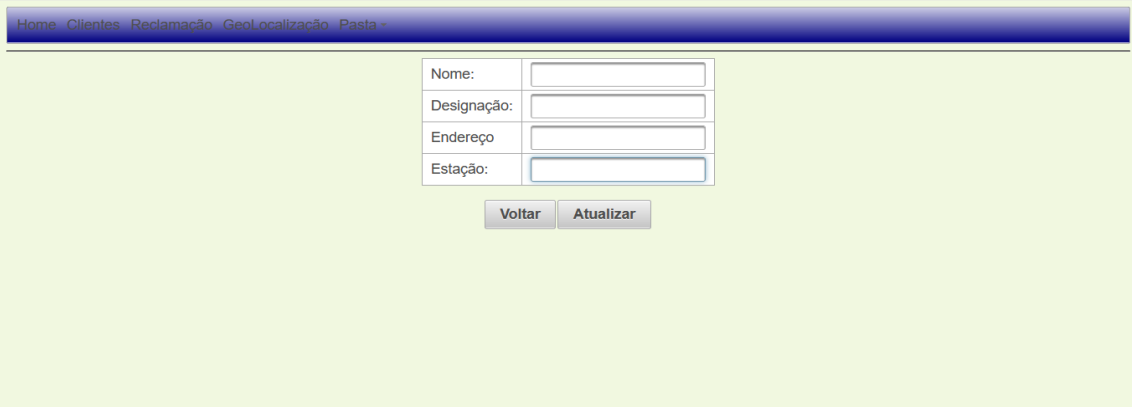
Figura 29 – Tela cadastro de cliente e lista dos clientes



Fonte: Próprio autor

2.6.3 Tela para realizar correções e alterações dos dados do cliente como nome, designação, endereço e estação, assim atualizando o cadastro.

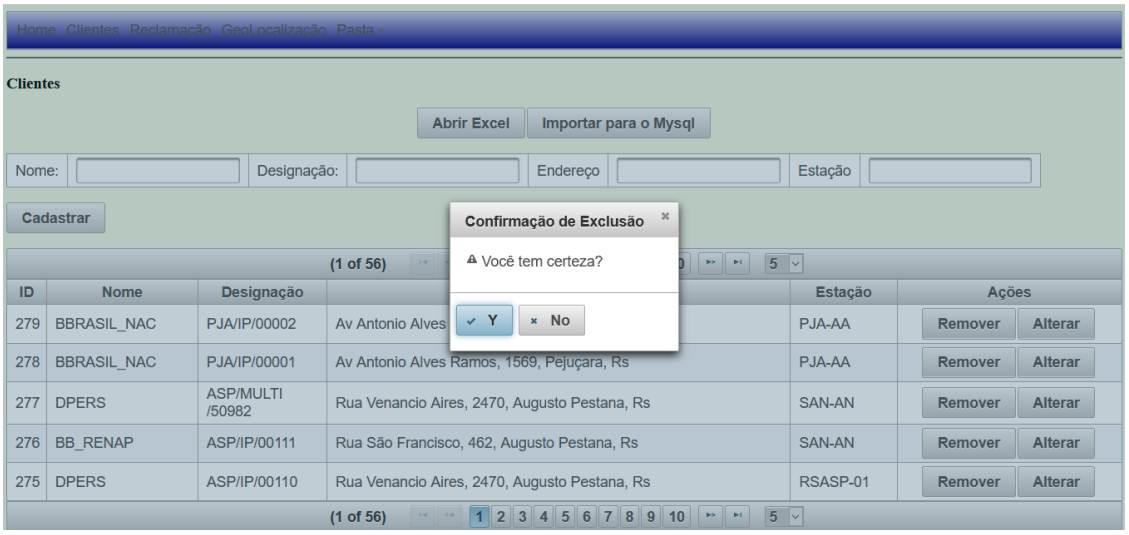
Figura 30 – Alterar os cadastros



Fonte: Próprio autor

2.6.4 Tela de exclusão de cliente onde é aberto um modal de de YES / NO.

Figura 31 – Tela de remover cadastro



Fonte: Próprio autor

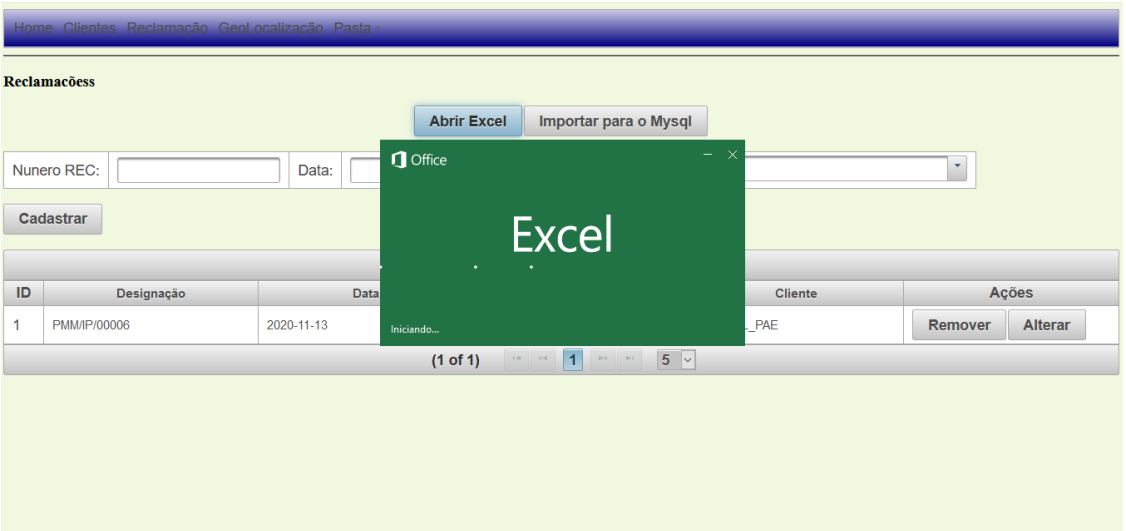
2.6.5 Tela onde se dará o cadastramento das reclamações, assim que cada reclamação for cadastrada aparecerá um ponto de referência no mapa de localização.

Figura 32 – Tela cadastrando as reclamações

Fonte: Próprio autor

2.6.6 Tela onde acionamos a planilha Excel, para importar os dados necessários assim realizando o carregamento no banco de dados das informações necessárias para alimentar na janela reclamações.

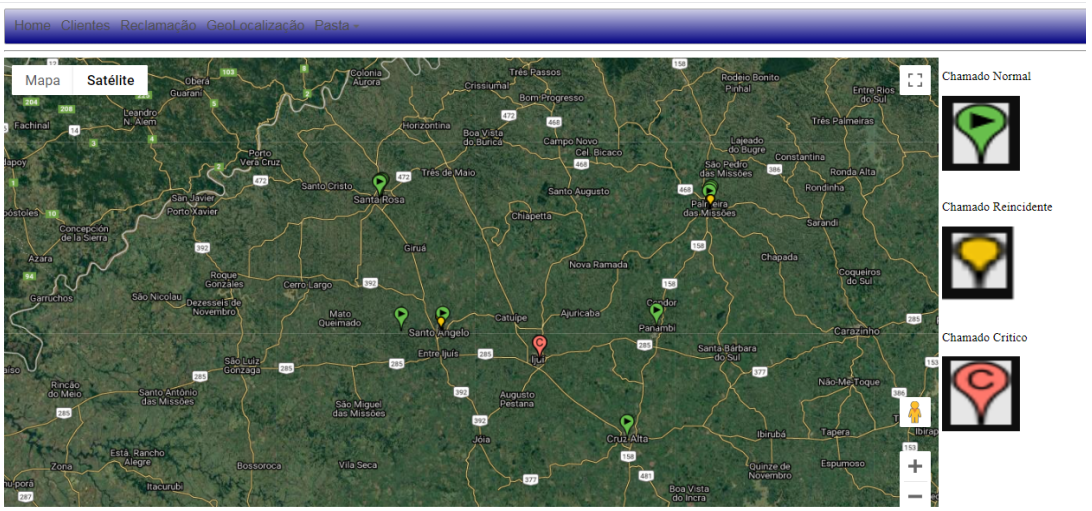
Figura 33 – Chamando o Excel



Fonte: Próprio autor

2.6.7 Tela onde será identificado cada reclamação, podendo aparecer vários pontos no mesmo local, a referência da localização dada por dados geográficos, latitude e longitude.

Figura 34 – Identificação das reclamações no mapa



Fonte: Próprio autor

3 RESULTADO E DISCUSSÕES

Os resultados obtidos neste projeto foram positivos e dentro do esperado. Para garantir de certa forma, que tudo foi realizado de forma correta, foram estudadas metodologias de desenvolvimento que serviram como guias, para assegurar a qualidade e produtividade de um processo de desenvolvimento de software, portanto foram discutidas algumas abordagens e componentes que fazem parte do desenvolvimento de software.

3.1 QUESTIONÁRIO DE AVALIAÇÃO E TESTES DO SISTEMA

O sistema pode ser acessado em qualquer tipo de navegador da Web, e independente do tipo de sistema operacional que será acessado pelo usuário. O sistema deverá ser instalado em qualquer servidor ou máquina que forneça suporte e acesso à Internet, foram utilizadas como ferramentas de teste: NetBeans, GlassFish, PHP MySQL, Jasperreport.

Avaliação realizada com as notas, sendo 1 para inaceitável, 2 para insatisfatório, 3 para médio, 4 para Bom, e 5 Excelente, escolhas a critério do usuário que testou o sistema.

Critérios Ergonômicos	1	2	3	4	5
1 – O software provê funcionalidades que satisfaçam as necessidades do usuário?				X	
2 – O software é facilmente compreendido e operado?					X
3 – O software deixa claro o que o usuário deve fazer?					X
4 – O software deixa claro cada item exibido na tela?				X	
5 – O software permite avançar e recuar durante o seu uso?					X
6 – O software tem um design atraente?			X		
7 – O uso de recursos da máquina é compatível com as funcionalidades?				X	
8 – O software pode ser executado em telas menores ou maiores?				X	
9 – Estou satisfeito com o desempenho do software?			X		

3.1.1 Resultado do sistema

O sistema criado, é possível coletar dados externos e importar para dentro do banco onde, endereços específicos são convertidos para latitude e longitude, assim podendo mapear as suas localizações e quantidades de chamados localizado na sua região, ganhando com isso o estudo o referencial teórico para o desenvolvimento do software, também foi recolhido os requisitos funcionais e não funcionais de uma planilha de dados específica, realizado um projeto de Banco de Dados Relacional, e implementado um sistema de software denominado Sistema de Gerenciamento de reclamações.

4 CONSIDERAÇÕES FINAIS

O compartilhamento do conhecimento é imprescindível para a aluno, no sentido de contribuir para o alcance do sucesso de suas estratégias, proporcionar a inovação contínua e contribuir para a aprendizagem coletiva. Num ambiente acadêmico, o eficiente compartilhamento do conhecimento promove o aprendizado em projetos, pois os conhecimentos adquiridos ao longo de um projeto poderão ser reutilizados, evitando-se assim problemas relacionados à o retrabalho ou repetição de erros.

A partir da utilização do sistema desenvolvido no decorrer deste projeto foi possível ter uma visão mais amigável de onde estão ocorrendo o maior número de reclamações, pois os clientes são mostrados no google mapas, sendo possível no futuro melhorar e ampliar as suas funcionalidades, como adicionar rotas e gráficos quantitativos.

REFERÊNCIAS

Batchgeo. Mapear seus Dados de Localização do Excel com BatchGeoDisponível em <https://pt.batchgeo.com/features/map-excel-data/>. Acessado em 02 abril. 2020.

Codificar. O que é geolocalização e como aplica-la no seu negócio. Disponível em <https://codificar.com.br/geolocalização-o-que-e/>. Acessado em 04 abril. 2020.

Devmedia. Gerando Relatórios com JasperReports. Disponível em <https://www.devmedia.com.br/gerando-relatorios-com-jasperreports/24798>. Acessado em 12 dezembro. 2020.

EasyMapMaker. Easymapmaker. Disponível em <https://www.easymapmaker.com/pt/>. Acessado em 02 abril. 2020.

F583j Flanagan, David JavaScript: o guia definitivo / David Flanagan; tradução: João Eduardo Nóbrega Tortello; revisão técnica: Luciana Nedel. – 6. ed. – Porto Alegre: Bookman, 2013. XVIII, 1062 p.: il.; 25 cm.

Fábrica de Software. O que é JSF (Java Server Faces)? Disponível em <fabrica.ms.senac.br/2013/06/o-que-e-jsf-java-server-faces>. Acessada em 09 maio. 2020.

Google Maps. Meus mapas. Disponível em < https://www.google.com/intl/pt-BR/maps/about/mymaps/>. Acessado em 06 abril. 2020.

Microsoft. Descrição das noções básicas de normalização de banco de dados. Disponível em <https://docs.microsoft.com/pt-br/office/troubleshoot/access/database-normalization-description>. Acessado em 11 fevereiro. 2021.

MRACK, M. (2006) "Hibernate, uma visão geral sobre o framework padrão de fato para mapeamento objeto-relacional", http://www.slideshare.net/mmrack/workshophibernate-com-comentarios, acessado em 02 -04 de 2020

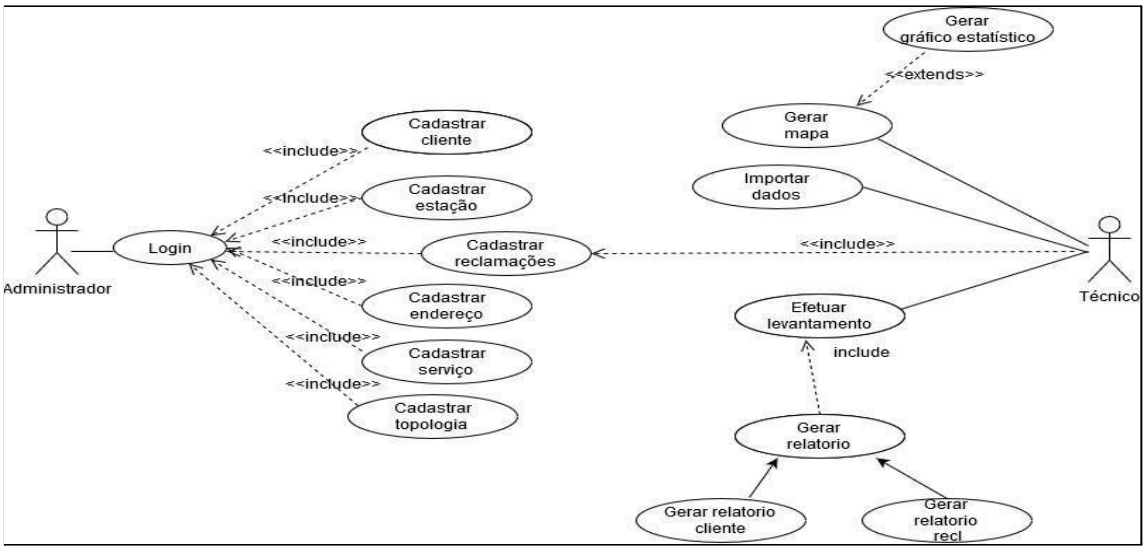
Significados. Significado de Java. Disponível em <https://www.significados.com.br/java/>. Acessado em 11 abril. 2020.

Tableless. O que é MVC? Disponível em <https://tableless.com.br/mvc-afinal-e-o-que/>. Acessado dia 13 junho. 2020.

Tecnoblog. Qual a diferença entre GPS, A-GPS e GLONASS? Disponível em <https://tecnoblog.net/265818/qual-a-diferenca-entre-gps-a-gps-e-glonass/>. Acessado em 04 abril. 2020.

APÊNDICE A - Imagem do diagrama de caso de uso

Figura 20 - Caso de Uso



Fonte: Próprio autor.

APÊNDICE B - Caso de uso Especificações

Tabela 1 - Especificação caso de uso Cadastrar Cliente

Identificação caso de uso: CC01	Nome caso de uso: Cadastrar cliente
Ator Principal: Técnico	
Resumo: descreve as sequências de passos que o técnico realizará para cadastrar cada cliente no sistema web.	
Pré-condições	Pós-condições: Mensagem de cadastro concluído
Fluxo Principal	
Ações do Ator: 1. Clicar em cliente 2. Inserir informações 3. Clicar em cadastrar	Ações do Sistema 4. Sistema valida informações 5. Sistema insere as informações no banco de dados
Fluxo Alternativo	
Ações do Ator	Ações do Sistema
	3.a) Dados inválidos 4.a) Sistema retorna ao passo 2.

Fonte: Próprio autor.

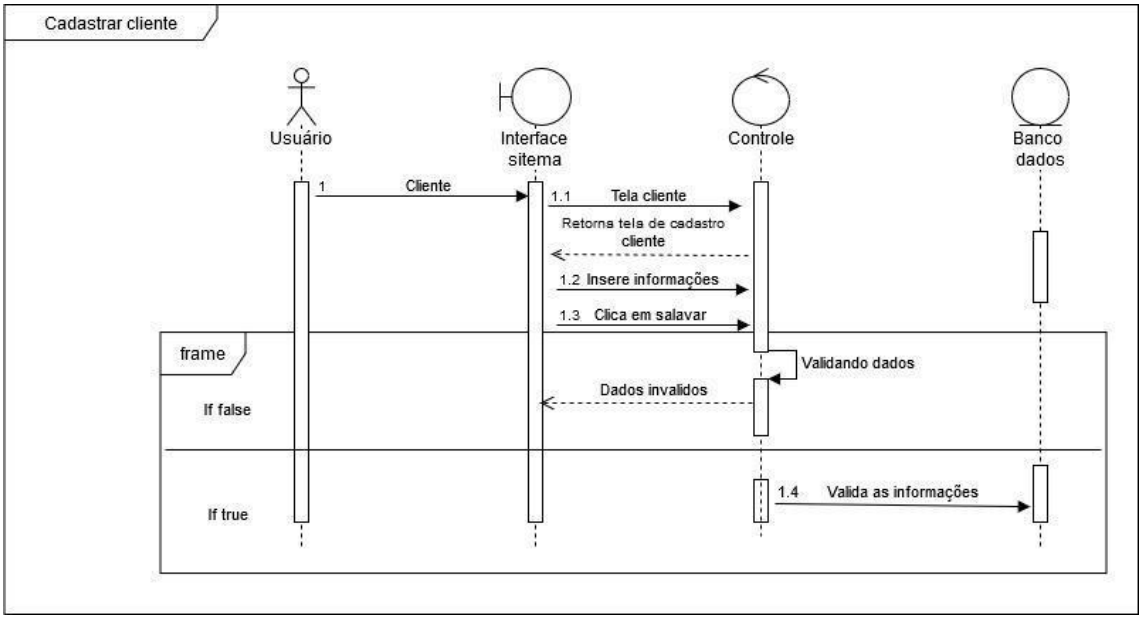
Tabela 2 - Especificação caso de uso Importar Dados

Identificação caso de uso: ID01	Nome caso de uso: Importar dados
Ator Principal: Técnico	
Resumo: Descreve as sequências de passos que o técnico realizará para importar dados para o sistema web.	
Pré-condições: planilha Excel com dados a ser copiados	Pós-condições: Mensagem de dados inseridos
Fluxo Principal	
Ações do Ator: 1. Clicar em importar 2. Copiar e colar dados 3. Valida informações	Ações do Sistema 4. Verificar e alterar informações 5. Sistema insere as informações no banco de dados
Fluxo Alternativo	
Ações do Ator	Ações do Sistema
	3.a) Dados inválidos 4.a) Sistema retorna ao passo 2.

Fonte: Próprio autor.

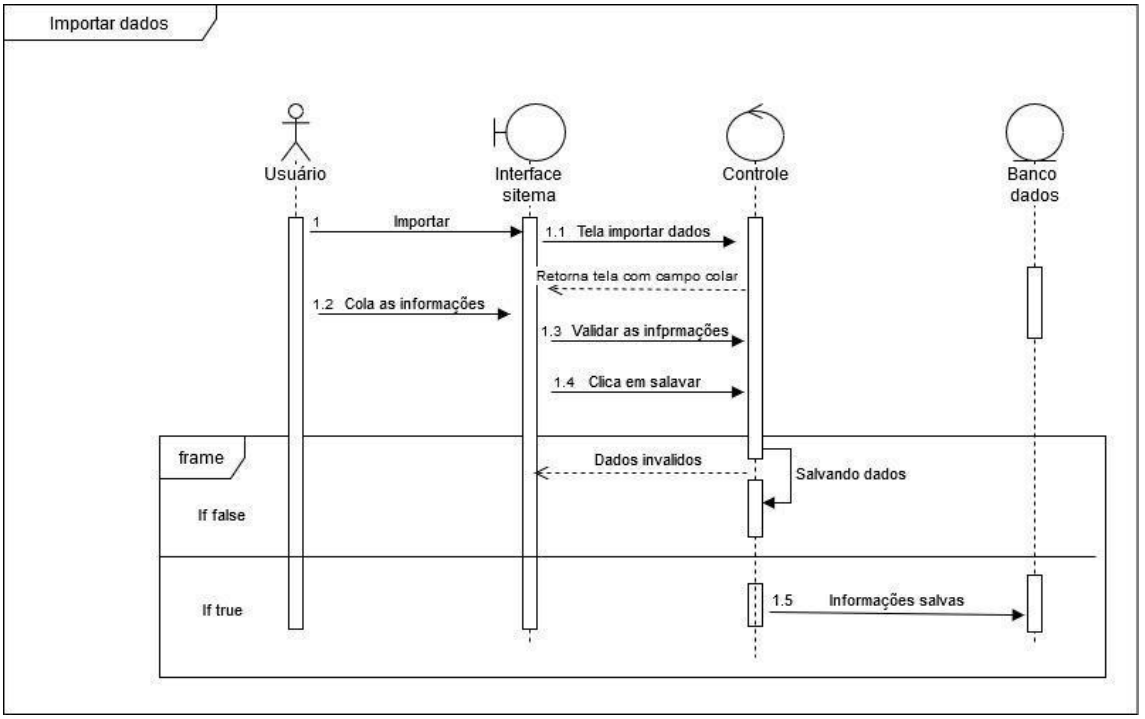
APÊNDICE C - Diagramas sequência

Figura 21 – Diagrama de sequência do caso de uso cadastrar cliente



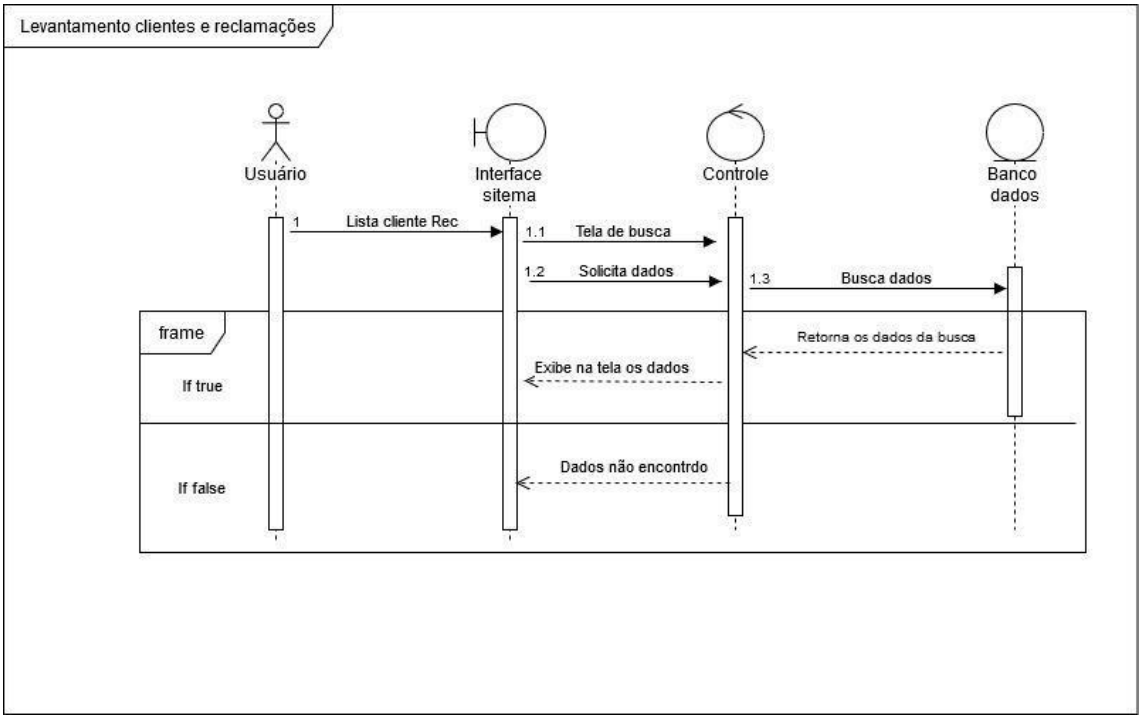
Fonte: Próprio autor

Figura 22 – Diagrama de sequência do caso de uso importar dados



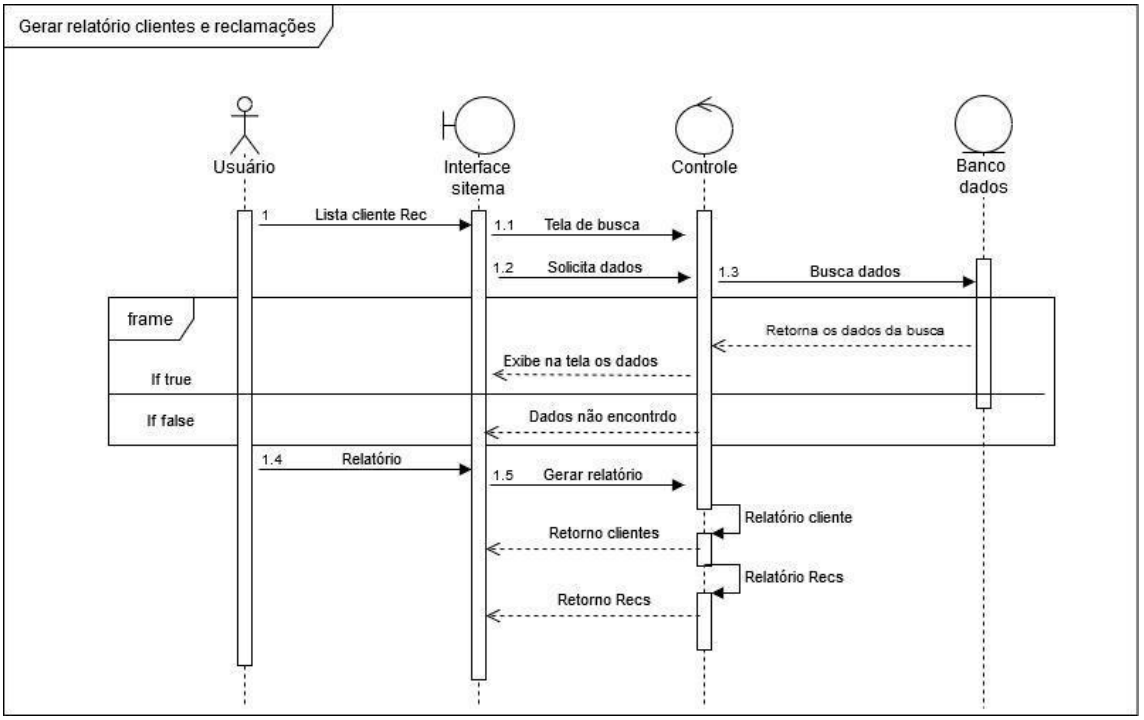
Fonte: Próprio autor

Figura 23 – Diagrama de sequência do caso de uso levantamento clientes e reclamações



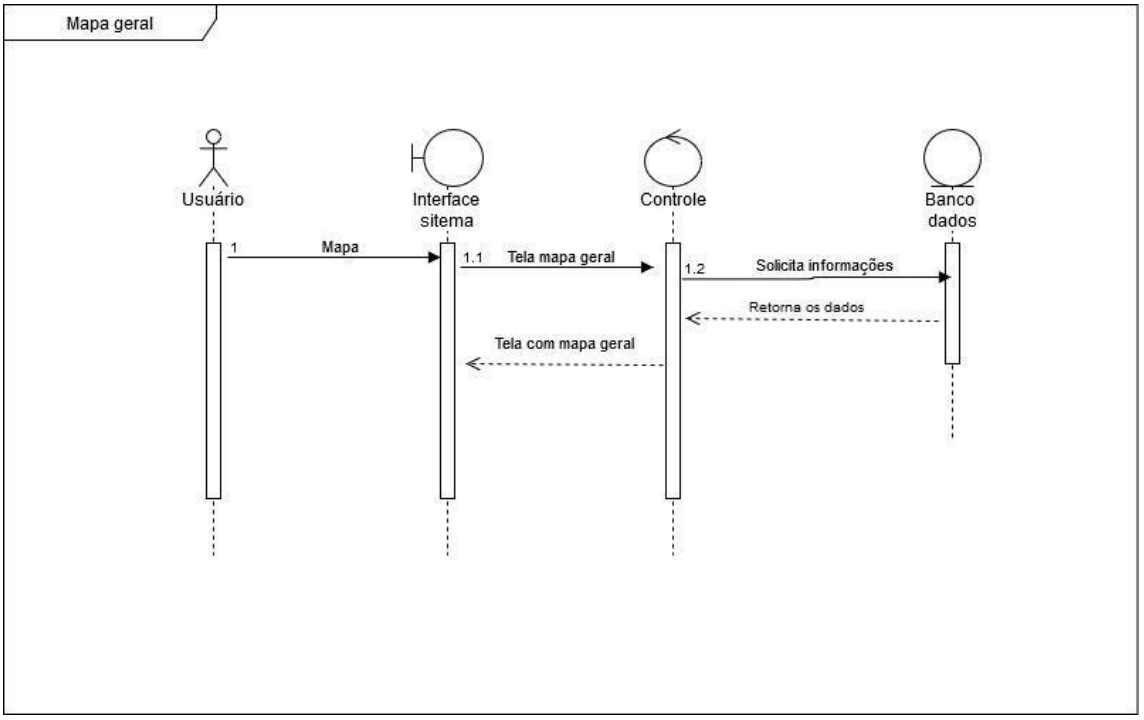
Fonte: Próprio autor

Figura 24 – Diagrama de sequência do caso de uso gerar relatório clientes e reclamações



Fonte: Próprio autor

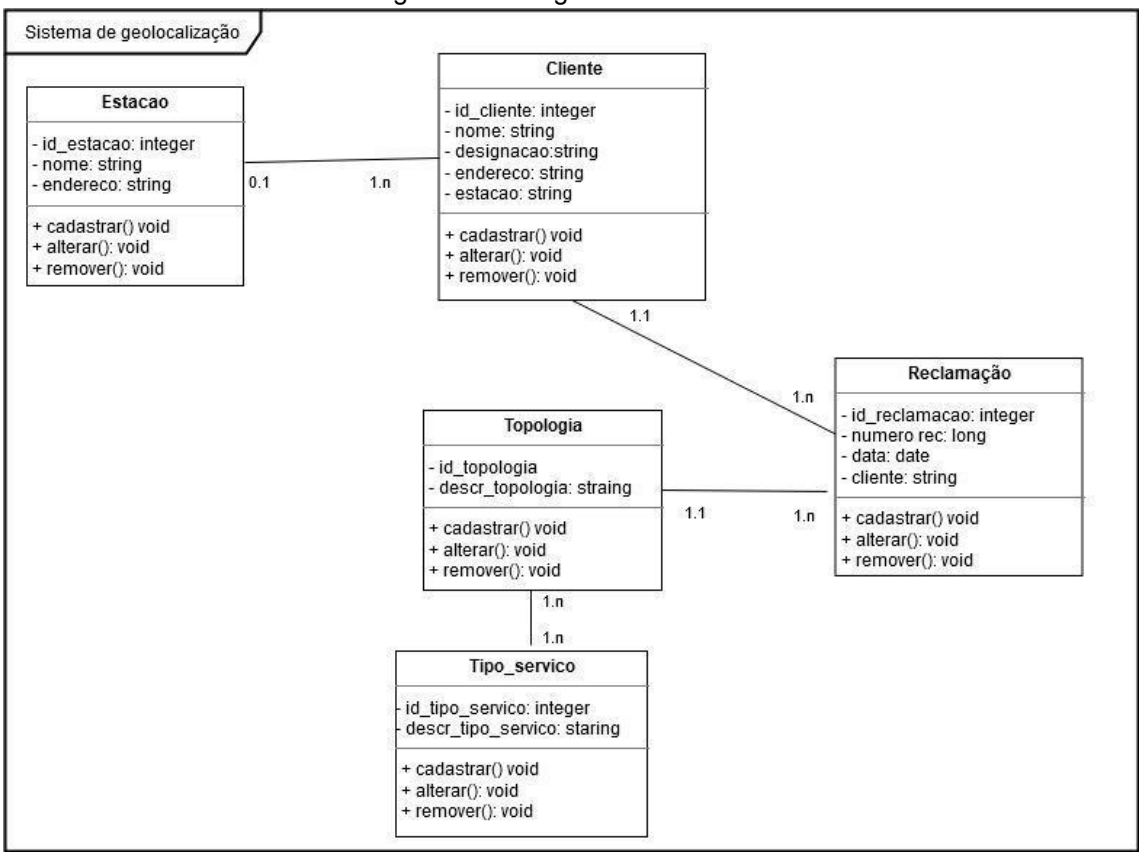
Figura 25 – Diagrama de sequência do caso de uso mapa geral



Fonte: Próprio autor

APÊNDICE D - Diagrama de classes

Figura 27 – Diagrama de classes



Fonte: Próprio autor

APÊNDICE E - Código fonte

Método para chamada do excel

```
public String ChamaExcel() throws IOException
   {
            Runtime.getRuntime().exec("C:\\Program Files (x86)\\Microsoft
      Office\\Office12\\EXCEL.EXE
      \"C:\\Users\\Luimar\\Documents\\NetBeansProjects\\VolumeRec1\\Tabcli
      ente.xls\"");
     return "cliente";
    }
```

Importação para o banco de dados do sistema.

MysqlReclamações:

```
package Controller;

import java.io.BufferedReader;
import java.io.File;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.URL;
import java.net.URLEncoder;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import jxl.Cell;
import jxl.Sheet;
import jxl.Workbook;
import jxl.read.biff.BiffException;
```

```
public class MysqlReclamacoes {


    ConexaoReclamacao con;  //Conexão com o banco de dados.
    String data;
    String designacao;
    String id_cliente;
    String nome;
    String num_rec;
    String endereco;


   //métodos para ler os arquivos no excel
         void ArquivosExcel(String arquivoDestino) throws IOException,
BiffException, SQLException
          {
       int contador = 1;
       int controle = 0;
       Workbook workbook = Workbook.getWorkbook(new File(arquivoDestino));
         //Le a quantidade de linha da planilha
       Sheet sheet = workbook.getSheet(0);
       int linhas = sheet.getRows();
       int colunas = sheet.getColumns();
       String atual;
       for (int i = 0; i < linhas; i++) {
           controle = i;
           for (int j = 0; j < colunas; j++) {


               Cell celula = sheet.getCell(j, i);
               atual = celula.getContents();
               System.out.println("Linha: " + i + " Coluna: " + j);
               System.out.println(atual);
```

```
            if (j == 0) {
                nome = atual;
            }
            if (j == 1) {
                designacao = atual;
            }
            if (j == 2) {
                id_cliente = atual;
            }
            if (j == 3) {
                endereco = atual;
            }
             if (j == 4) {
                num_rec= atual;
            }
            if (j == 5) {
                //nao faz nadda
            }


        } //terminou de percorrer as colunas
       // e faz a inserção no banco caso não seja linha em branco
        //se não for a primeira linha e o endereço sem espaços não for diferente de nada então insere no banco.

        if (((controle != 0) && (endereco!= ""))) {


            if (endereco.toString() == "%2B") {
                System.out.println("Fim de planilha ou endereço em branco.");
                System.exit(0);
            }
            if (endereco.toString() == "") {
```

```
            System.out.println("Fim de planilha ou endereço em branco.");
            System.exit(0);
        }
        if (endereco.length() < 5) {
            System.out.println("Fim de planilha ou endereço em branco");
            System.exit(0);
        }

        System.out.println("Mostra o endereço lido do EXCEL");
        System.out.println(endereco);
        endereco = endereco.replaceAll(" ", "+"); //realiza a leitura de linha por linha
        endereco = endereco.trim();  //elimina os espaços embranco
        endereco = URLEncoder.encode(endereco, "UTF-8"); //codifica as string
        URL url = new
URL("https://maps.googleapis.com/maps/api/geocode/json?address=" +
endereco + "&key=AIzaSyDQQuEd1FhWxuolCDxCWvZOUiNJbGW4Ih8");
//endereço da api do Google e chave KEY

        System.out.println("Mostra o endereço tratado, sem espaços com
UTF-8");
        System.out.println(endereco);

        String json;
        try (
            // Le todo o texto retornado pelo servidor
              BufferedReader in = new BufferedReader(new
InputStreamReader(url.openStream()))) {
            String str;
            json = "";
            int counter = 0;
            while ((str = in.readLine()) != null) {
                json = json + str;
```

```
                }
            }

            System.out.println("Vai mostrar o JSON");
            System.out.println(json);

            String location = json.substring(json.indexOf("\"location\" :"),
json.indexOf("\"location_type\"") - 13); //endereço convertido em latitude e longitude
            String lat = location.substring((location.indexOf("\"lat\"") + 8),
location.indexOf(","));
            String lng = location.substring((location.indexOf("\"lng\"") + 8),
(location.indexOf("}") - 12));

            System.out.println("Mostrará a latitude e longitude");
            System.out.println(lat);
            System.out.println(lng);

            System.out.println("\n");
            ConexaoReclamacao con = new ConexaoReclamacao();
            Connection cn = con.Conectar();
        //conectando a inserindo no banco de dados
            String setenca = "INSERT INTO reclamacao(data , designacao ,
id_cliente , nome , num_rec,latitude,longitude)VALUES(NOW(),'" + designacao
+ "','" + id_cliente + "','" + nome + "','" + num_rec + "','" + Float.valueOf(lat) + "','"
+ Float.valueOf(lng) + "')";
            System.out.println("Mostrará o SQL");
            System.out.println(setenca);
            PreparedStatement pst = cn.prepareStatement(setenca);
            pst.executeUpdate();
            con.EncerrarConexao();
        }
    } //end que percorre as linhas da planilha
  }
```

```
}
```

MapaBean

```
@PostConstruct
   public void init() {

       simpleModel = new DefaultMapModel();
       reclamacoes = reclamacaoDao.getList();//pega todos os registros da
tabela reclamacoes

       String marcador = "";
       int i = 0;
       int contador = 0;
       for (Iterator iterator = reclamacoes.iterator(); iterator.hasNext();) {

         String designacao = reclamacoes.get(i).getDesignacao();

          contador = 0;
         //testa a designação

         for(int j=0; j < reclamacoes.size(); j++){
          if (reclamacoes.get(j).getDesignacao().equals((designacao))){
              //inicia a comparação
              contador=contador+1;
             }
         }

           if(contador > 2){
               System.out.println("Vai colocar marcador vermelho");
               marcador = "http://www.google.com/mapfiles/markerC.png";
           }

           else
           {
              System.out.println("Vai colocar marcador Verde");
               marcador = "http://www.google.com/mapfiles/dd-start.png";
           }
           if(contador == 2)
           {


         System.out.println("vai colocar marcador laranja");
         marcador ="
http://labs.google.com/ridefinder/images/mm_20_orange.png";
```

```
            }

            Reclamacao obj = (Reclamacao) iterator.next();
            //abaixo, cria a coordenada, conforme lat e long do banco de dados
            LatLng coord = new LatLng(reclamacoes.get(i).getLatitude(),
reclamacoes.get(i).getLongitude());
            String nome = reclamacoes.get(i).getNome();
            designacao = reclamacoes.get(i).getDesignacao();
            simpleModel.addOverlay(new Marker(coord, nome, this, marcador));
//adiciona o marcador no mapa
          //adiciona o marcador
            //coord = null;*/

            i++;


    }
          }

    public MapModel getSimpleModel() {
        return simpleModel;
    }
}
```

Mapa.xhtml

```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<ui:decorate xmlns="http://www.w3.org/1999/xhtml"
             xmlns:h="http://xmlns.jcp.org/jsf/html"
             xmlns:t="http://xmlns.jcp.org/jsf/html"
             xmlns:f="http://xmlns.jcp.org/jsf/core"
             xmlns:p="http://primefaces.org/ui"
             xmlns:ui="http://xmlns.jcp.org/jsf/facelets"
             template="frente_2.xhtml">
```

```
//função que traz a localização do mapa do Rio Grande do sul
<script type="text/javascript">
        function
                onload() {

            map.addOverlay(new GMarker(new GLatLng(#{-28.2752662},
#{-54.251698}, {'asdfasdf'}
            , {
            }
            , {'http://www.google.com/mapfiles/markerM.png'
            }
            )));

        }
    </script>

    <h:body style="background-color: #A9A9A9">
        //centraliza o mapa do RS no google maps e mostra na tela a plotagem do pontos
designados.
        <script type="text/javascript"
src="https://maps.google.com/maps/api/js?key=AIzaSyDQQuEd1FhWxuolCDx
CWvZOUiNJbGW4Ih8"></script>
        <p:gmap center="-28.1757,-54.1547" zoom="9" type="HYBRID"
style="width:100%;height:630px" model="#{mapaBean.simpleModel}" />


    </h:body>

</ui:decorate>
```