



REPÚBLICA FEDERATIVA DO BRASIL
MINISTÉRIO DA ECONOMIA
INSTITUTO NACIONAL DA PROPRIEDADE INDUSTRIAL
DIRETORIA DE PATENTES, PROGRAMAS DE COMPUTADOR E TOPOGRAFIAS DE CIRCUITOS INTEGRADOS

Certificado de Registro de Programa de Computador

Processo Nº: **BR512019001682-3**

O Instituto Nacional da Propriedade Industrial expede o presente certificado de registro de programa de computador, válido por 50 anos a partir de 1º de janeiro subsequente à data de 20/07/2019, em conformidade com o §2º, art. 2º da Lei 9.609, de 19 de Fevereiro de 1998.

Título: MAPP - Uma aplicação para visualização de base de dados georreferenciados

Data de publicação: 20/07/2019

Data de criação: 15/01/2019

Titular(es): VICTOR MARCELO RODRIGUES SILVA; ERIC AMARAL FERREIRA

Autor(es): VICTOR MARCELO RODRIGUES SILVA; SEBASTIÃO EMÍDIO ALVES FILHO; ERIC AMARAL FERREIRA; HITALO VINÍCIUS ALVES DE ANDRADE

Linguagem: JAVA SCRIPT

Campo de aplicação: AD-04; AH-02; AH-05

Tipo de programa: AP-02; FA-01; GI-01; GI-04; TC-01

Algoritmo hash: SHA-512

Resumo digital hash:

e91d79c4105dac8ca8fbc2d28cd6216a55c6ff8d9a4a1690dd970614dff56a07c2dd2d533a68c8b1243343dee0a4095b562
a39ca9135d3c3f2fc3b2bea88fed2

Expedido em: 13/08/2019

Aprovado por:

Helmar Alvares

Chefe da DIPTO - Portaria/INPI/DIRPA Nº 09, de 01 de julho de 2019

UNIVERSIDADE DO ESTADO DO RIO GRANDE DO NORTE – UERN

FACULDADE DE CIÊNCIAS EXATAS E NATURAIS – FANAT

DEPARTAMENTO DE INFORMÁTICA – DI

Victor Marcelo Rodrigues Silva

MAPP - Uma plataforma para visualização de base de dados
georreferenciados

MOSSORÓ - RN
2019

MAPP - Uma plataforma para visualização de base de dados georreferenciados

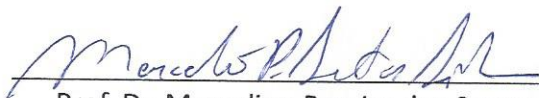
Monografia apresentada como pré-requisito para a obtenção do título de Bacharel em Ciência da Computação da Universidade do Estado do Rio Grande do Norte – UERN, submetida à aprovação da banca examinadora composta pelos seguintes membros:

Aprovada em: 26 / 09 / 2019

Banca Examinadora



Prof. Dr. Sebastião Emídio Alves Filho
Universidade do Estado do Rio Grande do Norte - UERN



Prof. Dr. Marcelino Pereira dos Santos Silva
Universidade do Estado do Rio Grande do Norte - UERN



Prof. Dr. Francisco Chagas de Lima Júnior
Universidade do Estado do Rio Grande do Norte - UERN

RESUMO

Os sistemas de informações geográficas vem ganhando cada vez mais espaço no âmbito da administração municipal, um dos fatores que têm grande influência na crescente busca por esse tipo de sistema é o também crescente número de dados associados a Big Data e a necessidade cada vez maior de manipular diversos dados que são atrelados a coordenada geográficas. Visando o auxílio na administração pública é que foi proposto o desenvolvimento deste trabalho, que tem como objetivo o desenvolvimento de uma plataforma para a visualização de base de dados georreferenciados. Quando consumindo dados fornecidos por um serviço externo, a aplicação faz parte de um SIG que pode ajudar diversos setores da administração pública, como por exemplo tomadas de decisões ou no armazenamento e manipulação de dados. A aplicação foi desenvolvida inteiramente utilizando a linguagem de programação JavaScript, e tendo como pilar principal a biblioteca ReactJS. Para a construção dos mapas presentes no sistema foi utilizada a biblioteca de código aberto LeafLet. A validação se deu com a aplicação consumindo serviços fornecidos pela MAPI, uma API REST voltada para a manipulação e consulta em bases de dados georreferenciados, onde a aplicação se comportou da forma desejada.

Palavras-chave: Georreferenciamento, sistemas WEB, SIG, mapa de cluster, mapa de calor.

Conteúdo

1. Referencial teórico	5
1.1 Georreferenciamento	5
1.2 Bancos de dados georreferenciados	5
2. Descrição geral do sistema	5
2.1 Tipos de Mapas	5
2.2 Auxílio na análise dos dados	8
3. Análise de requisitos	9
3.1 Casos de uso	9
3.2 Fluxo na aplicação	10
3.3 Requisitos funcionais e não funcionais	10
4. Desenvolvimento	15
3.1 Linguagens e ferramentas utilizadas	15
3.2 Paradigmas utilizados	16
4. Prova de conceito	17
3.1 MAPI	17
5. Referências	19

1. Referencial teórico

1.1 Georreferenciamento

O georreferenciamento de um dado com endereço é definido como o processo de associação desse dado a um mapa e pode ser efetuado de três formas básicas: associação a um ponto, a uma linha ou a uma área. O resultado desse processo é a criação de elementos gráficos que podem ser usados para a análise espacial (BARCELLOS, 2008).

1.1 Banco de dados Georreferenciados

Para Silva (2001) um banco de dados é georreferenciado, quando os dados constantes nele possuem atrelados a si informações que possibilitam saber a localização de sua ocorrência/existência. Por possuir tais informações, esses bancos de dados são comumente utilizados em análises complexas das informações obtidas, podendo ser crucial na designação de soluções para o gerenciamento do tráfego em avenidas ou na prevenção de acidentes de trânsito, por exemplo.

Alguns exemplos de bases georreferenciadas são: *acidentes de trânsito, eventos em geral, empreendimentos, vias urbanas, multas de trânsito, ocorrências médicas, etc.*

2. Descrição geral do sistema

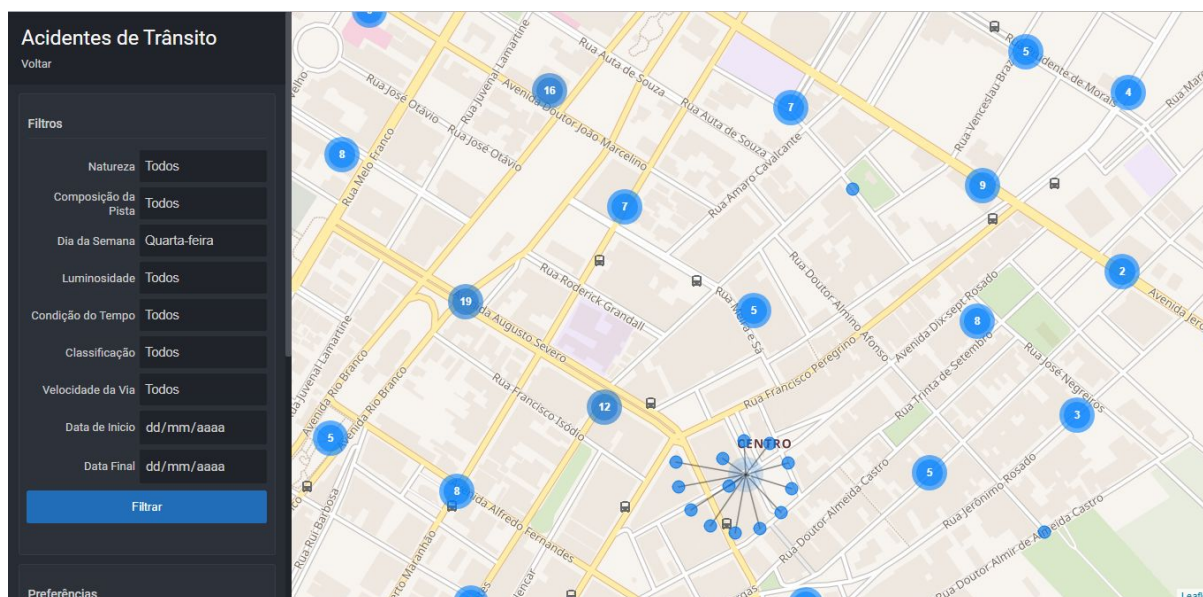
MAPP - Plataforma para visualização de base de dados georreferenciados, é uma ferramenta que visa facilitar a visualização e conseqüentemente a análise de bases de dados georreferenciados. Tendo como principal característica a disposição dos dados sob forma visual através de mapas, permitindo assim uma clara percepção das informações.

2.1 Tipos de mapas

O sistema conta com três tipos de mapas para a visualização, sendo eles:

Mapa de Cluster: bastante similar ao mapa de pontos, o mapa de *cluster* também dispõe cada registro do banco como um ponto no mapa, porém nesse mapa pontos próximos são agrupados em *clusters*, facilitando assim a visualização de grandes quantidades, uma vez que diminui-se consideravelmente a quantidade de registros apresentados ao mesmo tempo. O agrupamento dos pontos é determinado por um raio, e a amplitude deste pode ser controlada pelo usuário através da interface da aplicação.

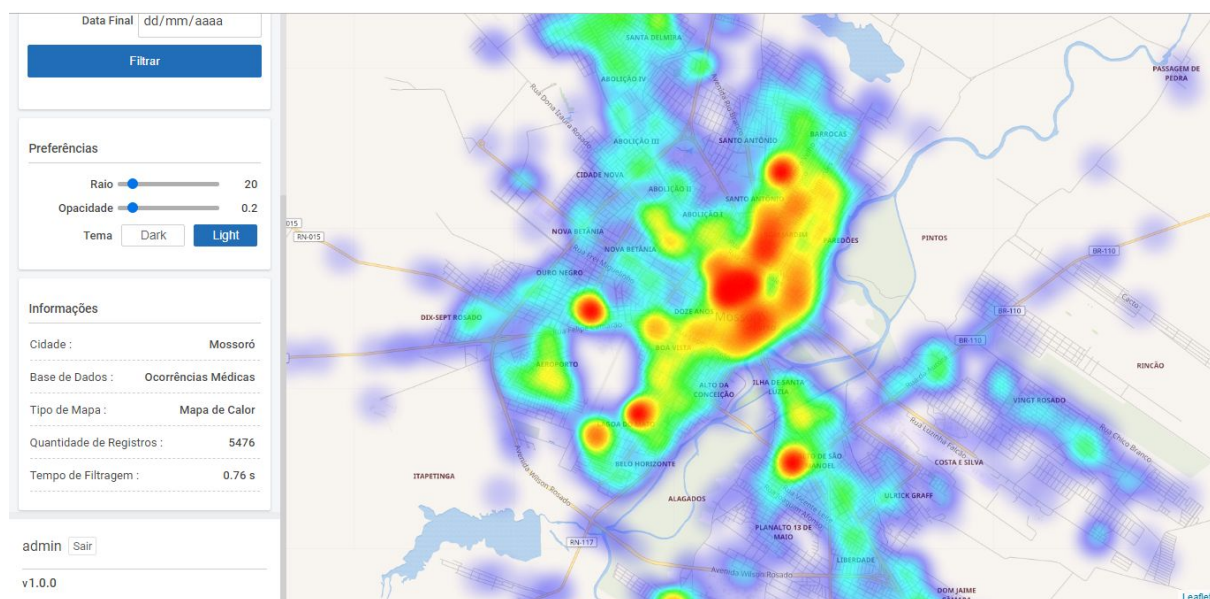
Figura 2: Mapa de Cluster



Fonte: Autoria própria

Mapa de Calor: mais voltado para a análise de dados, esse tipo de mapa mostra manchas de calor no mapa, onde cores mais próximas do vermelho indicam uma maior quantidade de registros no local, enquanto que cores mais frias indicam um menor número dos mesmos. Esse tipo de mapa pode auxiliar na detecção de pontos críticos em determinados setores, como uma grande quantidade de focos de mosquito da dengue em determinados bairros de uma cidade, permitindo assim que autoridades responsáveis possam analisar e tomar atitudes mais diretas no combate ou na prevenção, por exemplo.

Figura 3: Mapa de calor



Fonte: Autoria própria

2.2 Auxílio na análise dos dados

O sistema conta com um sistema de filtros, que permite ao usuário mensurar quais dados deseja visualizar. Esses filtros são determinados com base nas informações que cada registro do banco possui. Para uma melhor compreensão, imaginemos duas bases distintas, uma sendo referente à acidentes de trânsito e a outra aos pontos turísticos de uma cidade. Possíveis filtros para a primeira seriam: *ocorrências com vítimas fatais, horário da ocorrência, informações das vítimas, etc.* Enquanto que não faz sentido termos essas opções relacionadas aos pontos

turísticos, sendo os seguintes exemplos mais condizente para essa base de dados: *horário de funcionamento, valor de entrada, descrição do local, etc.*

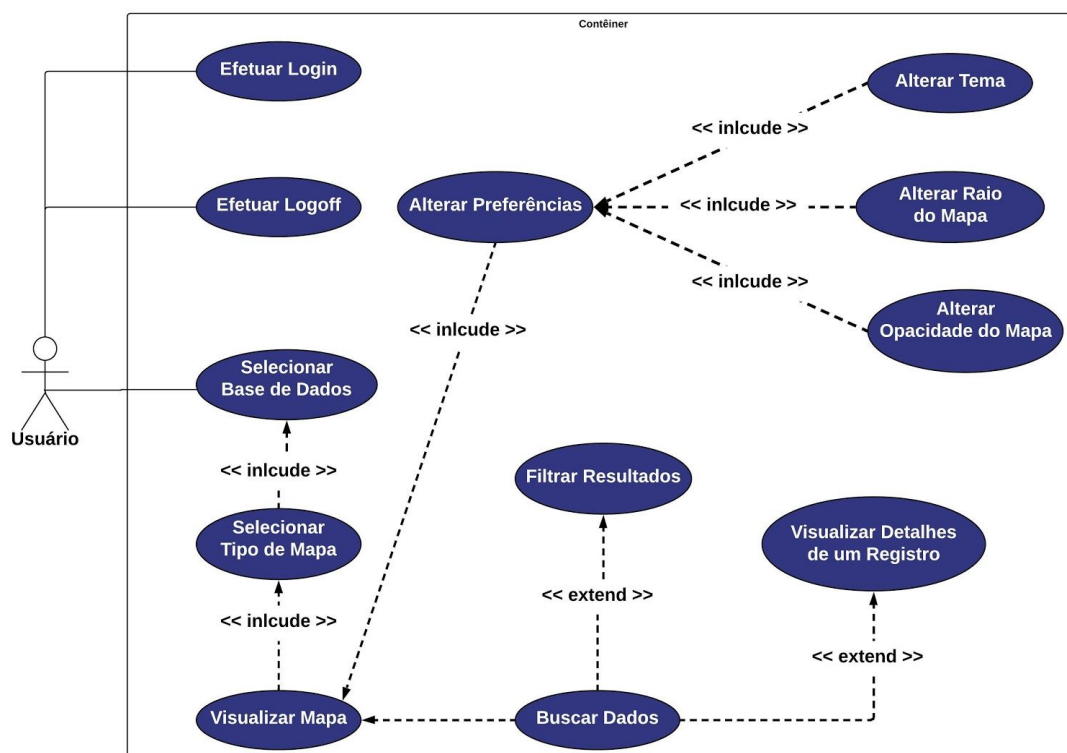
3. Análise de requisitos

A análise e elicitação de requisitos é o processo inicial no desenvolvimento de software. É nele onde é definido o que se deve fazer, quais necessidades o sistema deve atender e a quem deve suprir as necessidades.

3.1 Casos de uso

O diagrama de caso de uso expõe os requisitos funcionais, determinando tanto as funcionalidades, como as características do software sob o ponto de vista do usuário é desenvolvido durante a fase de levantamento e análise de requisitos para representar as principais funcionalidades (usos), e a interação dos usuários (atores) com o sistema (Sommerville, 2011).

Figura 4: Diagrama de caso de uso

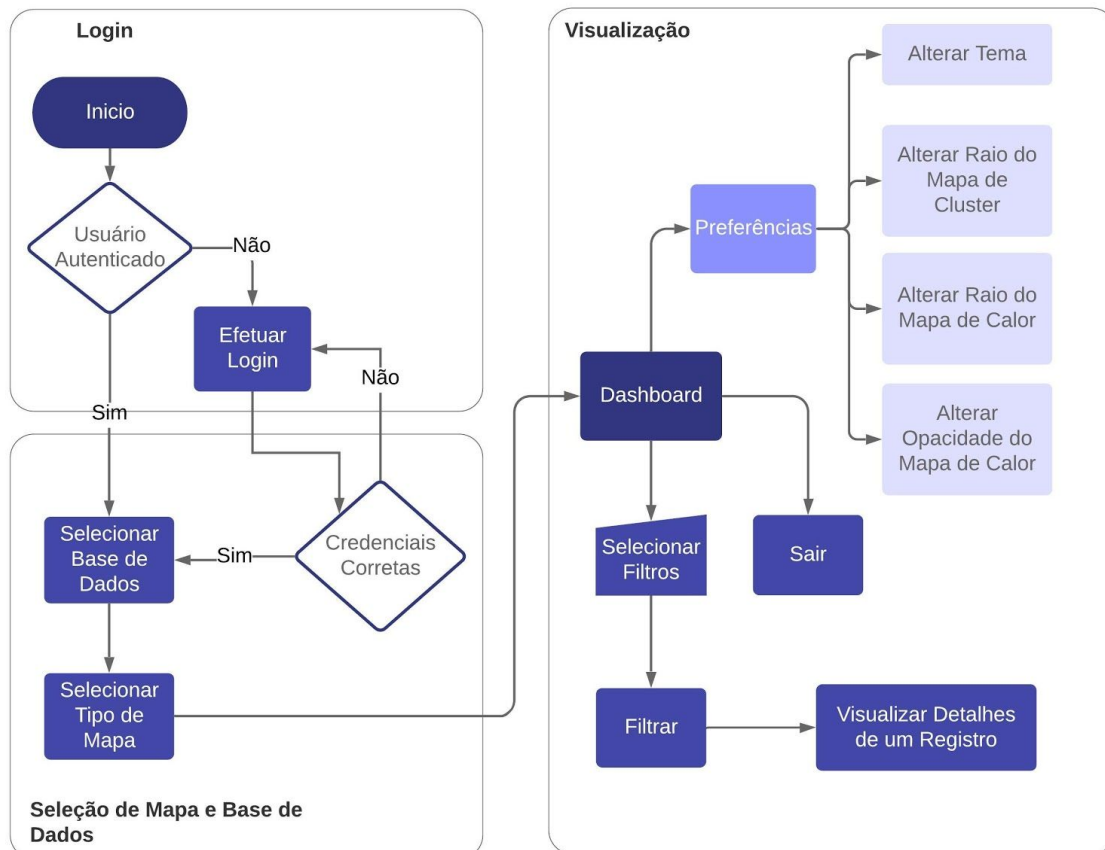


Fonte: Autoria própria

3.2 Fluxo na aplicação

O fluxograma apresentado a seguir exibe a trajetória do usuário no contexto de navegação do sistema, mostrando de forma cronológica quais passos o usuário deve seguir para a utilização da aplicação.

Figura 5: Diagrama de Fluxo



Fonte: Autoria própria

3.3 Requisitos funcionais e não funcionais

Requisitos são descrições de quais funcionalidade o sistema deve implementar e quais necessidades o mesmo deve ser capaz de atender e são divididos em dois grupos, os requisitos funcionais e não funcionais.

Os requisitos funcionais descrevem quais funcionalidades a aplicação deve possuir e como deve se comportar em determinadas situações. Por sua vez os

requisitos não funcionais referem-se às qualidades do sistema, podendo ser entendidos como não perceptíveis ao usuário final. Para estabelecer a prioridade dos requisitos, foram adotadas as denominações “essencial”, “importante” e “desejável”.

Essencial: é o requisito sem o qual o sistema não entra em funcionamento. Requisitos essenciais são requisitos imprescindíveis, que têm que ser implementados impreterivelmente.

Importante: é o requisito sem o qual o sistema entra em funcionamento, mas de forma não satisfatória. Requisitos importantes devem ser implementados, mas, se não forem, o sistema poderá ser implantado e usado mesmo assim.

Desejável: é o requisito que não compromete as funcionalidades básicas do sistema, isto é, o sistema pode funcionar de forma satisfatória sem ele. Requisitos desejáveis podem ser deixados para versões posteriores do sistema, caso não haja tempo hábil para implementá-los na versão que está sendo especificada.

A referência dos requisitos é feita através do nome da subseção onde eles estão descritos seguidos do identificador do requisito, de acordo com a especificação a seguir: [nome da subseção. identificador do requisito]

3.3.1 Requisitos funcionais

[RF001] Efetuar Login

Descrição do caso de uso: Este caso de uso permite que o usuário efetue login na aplicação.

Prioridade: Essencial Importante Desejável

Entradas e pré-condições: O usuário deve possuir um registro com nome de usuário e senha no banco de dados.

Saídas e pós-condição: O usuário efetua login na aplicação e pode ter acesso às suas funcionalidades.

[RF002] Selecionar Base de Dados

Descrição do caso de uso: Através desse caso de uso, o usuário seleciona qual base de dados irá visualizar.

Prioridade: Essencial Importante Desejável

Entradas e pré-condições: O usuário seleciona de uma lista, a base de dados desejada.

Saídas e pós-condição: O sistema interpreta qual base de dados deve carregar.

[RF003] Selecionar Tipo de Mapa

Descrição do caso de uso: Através desse caso de uso, o usuário seleciona qual deverá ser o mapa para a visualização da base de dados previamente selecionada.

Prioridade: Essencial Importante Desejável

Entradas e pré-condições: O usuário seleciona de uma lista, o tipo de mapa desejado.

Saídas e pós-condição: O sistema interpreta qual tipo de mapa deve usar para a visualização da base de dados.

[RF004] Selecionar Filtros

Descrição do caso de uso: O usuário informa quais devem ser os filtros utilizados na hora da recuperação dos dados.

Prioridade: Essencial Importante Desejável

Entradas e pré-condições: O usuário preenche um formulário com os filtros desejados.

Saídas e pós-condição: O sistema armazena quais devem ser os filtros utilizados.

[RF005] Filtrar

Descrição do caso de uso: O usuário submete o formulário previamente preenchido com os filtros e a aplicação faz uma requisição à API.

Prioridade: Essencial Importante Desejável

Entradas e pré-condições: Caso o usuário tenha selecionado algum filtro, este será utilizado durante a recuperação dos dados, caso contrário o sistema irá retornar todos os registros do banco de dados.

Saídas e pós-condição: O sistema mostra de forma visual os registros da base de dados.

[RF005] Visualizar Detalhes de um Registro

Descrição do caso de uso: O usuário deve ser capaz de visualizar de forma detalhada as informações de todos os registros do banco.

Prioridade: Essencial Importante Desejável

Entradas e pré-condições: O usuário informa qual registro deseja visualizar de forma detalhada.

Saídas e pós-condição: Uma página com os detalhes do registro selecionado é carregada.

[RF006] Alterar Tema da Aplicação

Descrição do caso de uso: O usuário deve ser capaz de alternar entre os temas da aplicação.

Prioridade: Essencial Importante Desejável

Entradas e pré-condições: O usuário informa qual tema prefere.

Saídas e pós-condição: O sistema alterna entre os temas.

[RF007] Alterar Preferências do Mapa de Calor

Descrição do caso de uso: O usuário deve ser capaz de informar qual será a opacidade e o raio no mapa de calor.

Prioridade: Essencial Importante Desejável

Entradas e pré-condições: O mapa selecionado deve ser o mapa de calor.

Saídas e pós-condição: O mapa é rapidamente renderizado com as novas preferências selecionadas.

[RF007] Alterar Preferências do Mapa de Cluster

Descrição do caso de uso: O usuário deve ser capaz de informar qual será o raio de agrupamento dos pontos no mapa de cluster.

Prioridade: Essencial Importante Desejável

Entradas e pré-condições: O mapa selecionado deve ser o mapa de cluster.

Saídas e pós-condição: O mapa é rapidamente renderizado com as novas preferências selecionadas.

[RF008] Efetuar Logoff

Descrição do caso de uso: O usuário efetua logoff na aplicação e é redirecionado para a tela inicial.

Prioridade: Essencial Importante Desejável

Entradas e pré-condições: O usuário deve estar autenticado.

Saídas e pós-condição: O token de autenticação é destruído e o usuário é redirecionado para a tela inicial da aplicação.

3.3.2 Requisitos não funcionais

[NF001] Usabilidade

Descrição do caso de uso: O sistema deve possuir uma interface simples e amigável ao usuário, sem muitas informações que venham a comprometer a utilização do mesmo.

Prioridade: Essencial Importante Desejável

[NF002] Desempenho

Descrição do caso de uso: Considerado como fator de qualidade de software, um bom desempenho, livre de longos períodos de carregamento e travamentos entre as telas, é essencial para uma melhor utilização da aplicação.

Prioridade: Essencial Importante Desejável

[NF003] Responsividade

Descrição do caso de uso: A interface de usuário(UI) deve ser capaz de se adaptar para os diferentes tamanhos de telas de todos os dispositivos capazes de acessar o aplicação.

Prioridade: Essencial Importante Desejável

[NF004] Hardware e Software

Descrição do caso de uso: A aplicação deve consumir uma API que seja capaz de fornecer os dados geográficos no formato GeoJson.

Prioridade: Essencial Importante Desejável

[NF005] Segurança

Descrição do caso de uso: Usuários não autenticados não devem estar habilitados a acessar as funcionalidade da aplicação, bem como o sistema deve impedir que informações de acesso sejam expostas.

Prioridade: Essencial Importante Desejável

4. Desenvolvimento

4.1 Linguagens e ferramentas utilizadas

O sistema foi desenvolvido no ambiente Web, utilizando JavaScript como linguagem de programação, tendo o ReactJS como pilar principal da aplicação. O ReactJS é uma biblioteca JavaScript voltada para o desenvolvimento de interfaces de usuário, sendo totalmente focada na criação da *View*, dentro do modelo *model-view-controller* MCV. A utilização desta biblioteca permite ao desenvolvedor focar-se totalmente no desenvolvimento *frontend* da aplicação, deixando com o *backend* diversas responsabilidades, como por exemplo a lógica de comunicação com o banco ou o tratamento e fornecimento de dados.

Na parte visual da aplicação foi utilizado o *Syntactically Awesome Style Sheets* SASS, um pré-processador de *Cascading Style Sheets* CSS. O mesmo conta com uma sintaxe muito parecida com o CSS, porém fornece algumas vantagens no desenvolvimento, como por exemplo a capacidade de “aninhar” componentes da interface dentro do mesmo bloco de código. O SASS também livra o desenvolvedor da obrigação de escrever códigos redundantes a fim de que sejam interpretados por todos os navegadores do mercado, gerando-os de forma automática durante o *build* da aplicação, tornando assim o desenvolvimento mais rápido e dinâmico.

Para a criação dos mapas, foi utilizado o LeafLet, uma biblioteca de código aberto voltada para a criação de mapas interativos. Além de possuir código aberto, permitindo assim adaptar o mapa da melhor forma possível para a aplicação, o LeafLet apresenta como grande vantagem sua leveza. Pesando apenas 38Kb, é

extremamente performático em dispositivos de menor capacidade de processamento, até mesmo quando lidando com grandes quantidades de dados.

4.2 Paradigmas utilizados

A aplicação foi desenvolvida utilizando o conceito de CSR - Client Side Rendering. Esse conceito apresenta algumas vantagens, bem como desvantagens. O principal benefício de sua utilização pode ser percebida na performance da aplicação. Todo o conteúdo é renderizado no lado do cliente, não sendo necessário uma requisição ao servidor todas as vezes que algum elemento da interface mude. Uma grande desvantagem da utilização deste paradigma é a indexação da aplicação nos motores de busca. Isso acontece porque, como o React busca as informações para exibição em tela de uma API externa, os mesmos são totalmente dependentes do Javascript, e este é automaticamente desabilitado pelo processo de indexação dos motores de busca.

5. Prova de conceito

Testes na aplicação ocorreram em dois ambientes distintos. Na primeira instância a aplicação consumiu dados fornecidos por uma API sendo executada em ambiente local, em segundo caso a aplicação consumiu a mesma API, porém executando na nuvem através de uma instância EC2 t2.micro, serviço fornecido pela AWS (Amazon Web Services).

5.1 MAPI

Todos os testes na aplicação foram feitos enquanto a mesma consumia dados fornecidos pela MAPI. MAPI é uma aplicação que compõe o mesmo SIG onde a aplicação MAPP está presente, ela manipula dados georreferenciados utilizando o modelo REST para efetuar consultas utilizando métodos de filtragem.

A figura 4 mostra os dados brutos retornados por uma requisição feita à MAPI, enquanto que a figura 5 mostra como a aplicação dispõe de forma gráfica os dados retornados pela consulta.

Figura 6: Dados brutos retornados por uma requisição à MAPI

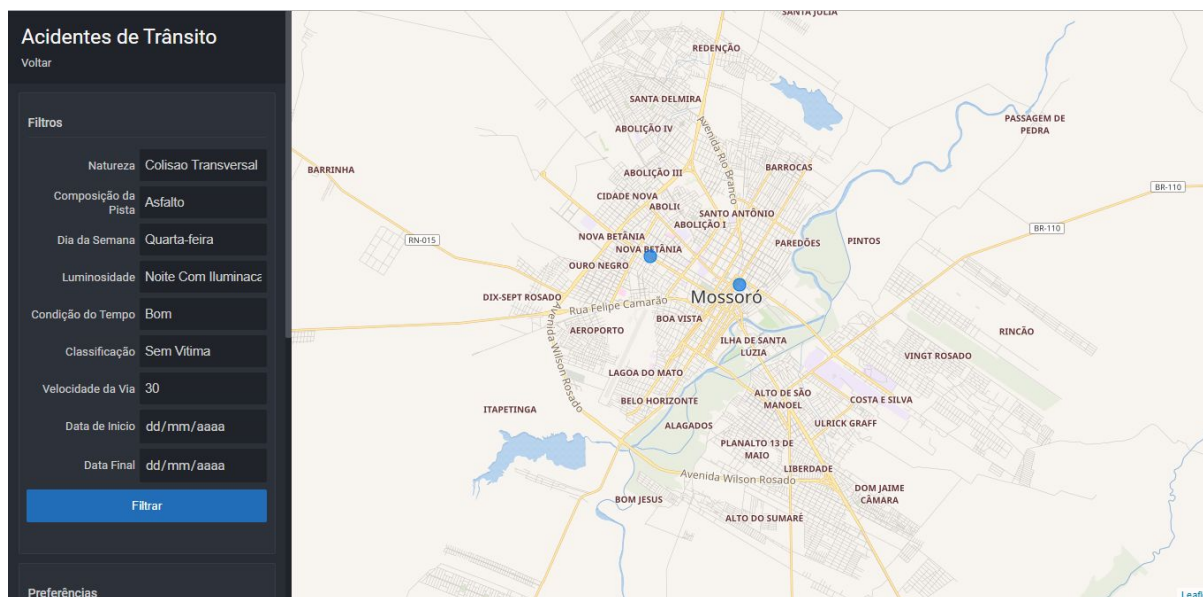
```

▼ (2) [{"..."}, {"..."}] ⓘ
  ▼ 0:
    ▶ geometry: {type: "Point", coordinates: Array(2)}
    ▶ properties: {numeroBOA: "1712018", latitude: "-5.18826678", longitude: "-37.34185187"}
      type: "Feature"
      _key: "695939"
    ▶ __proto__: Object
  ▼ 1:
    ▶ geometry: {type: "Point", coordinates: Array(2)}
    ▶ properties: {numeroBOA: "1802034", latitude: "-5.1827449", longitude: "-37.35911108"}
      type: "Feature"
      _key: "696389"
    ▶ __proto__: Object
  length: 2
  ▶ __proto__: Array(0)

```

Fonte: Autoria própria

Figura 7: Disposição visual de dados retornados pela MAPI



Fonte: Autoria própria

6. Referências

FACEBOOK, Inc.. **React - Uma biblioteca JavaScript para criar interfaces de usuário**, 2019. Disponível em: <https://pt-br.reactjs.org/>. Acesso em: 9 de Set. de 2019.

SOMMERVILLE, I. **Software engineering**. [S.l.: s.n.], 2011. v. 137035152.

AGAFONKIN, Vladimir. **Leaflet - a JavaScript library for interactive maps**, 2019. Disponível em: <https://leafletjs.com/examples.html>. Acesso em: 9 de Set. de 2019.

XAVIER-DA-SILVA, J. **Geoprocessamento para Análise Ambiental**. Rio de Janeiro: sn, 2001. 228 p.

BARCELLOS, Christovam de Castro et al. **Georreferenciamento de dados de saúde na escala submunicipal: algumas experiências no Brasil**. 2008.