

HEATMAP+ ADDRESS MONITOR

MANUAL DE INSTALAÇÃO

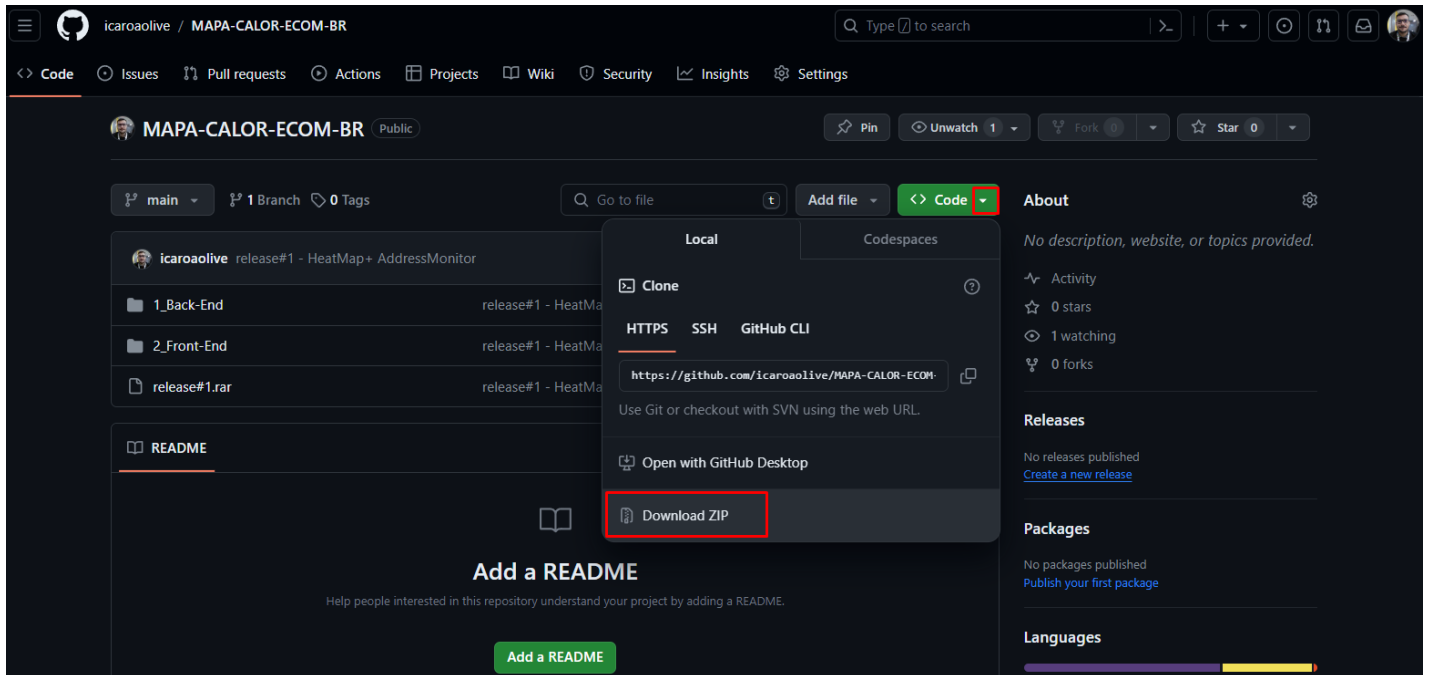


MINISTÉRIO DA
CIÊNCIA, TECNOLOGIA
E INOVAÇÃO



Agradecemos pela utilização do nosso projeto! E ficamos contentes de termos realizado este trabalho em parceria com vocês.

Ao receber os links de acesso do projeto, a ECOMBR poderá baixar os fontes através do link: <https://github.com/icaroaolive/MAPA-CALOR-ECOM-BR>, seguindo os passos ilustrados na imagem a seguir:

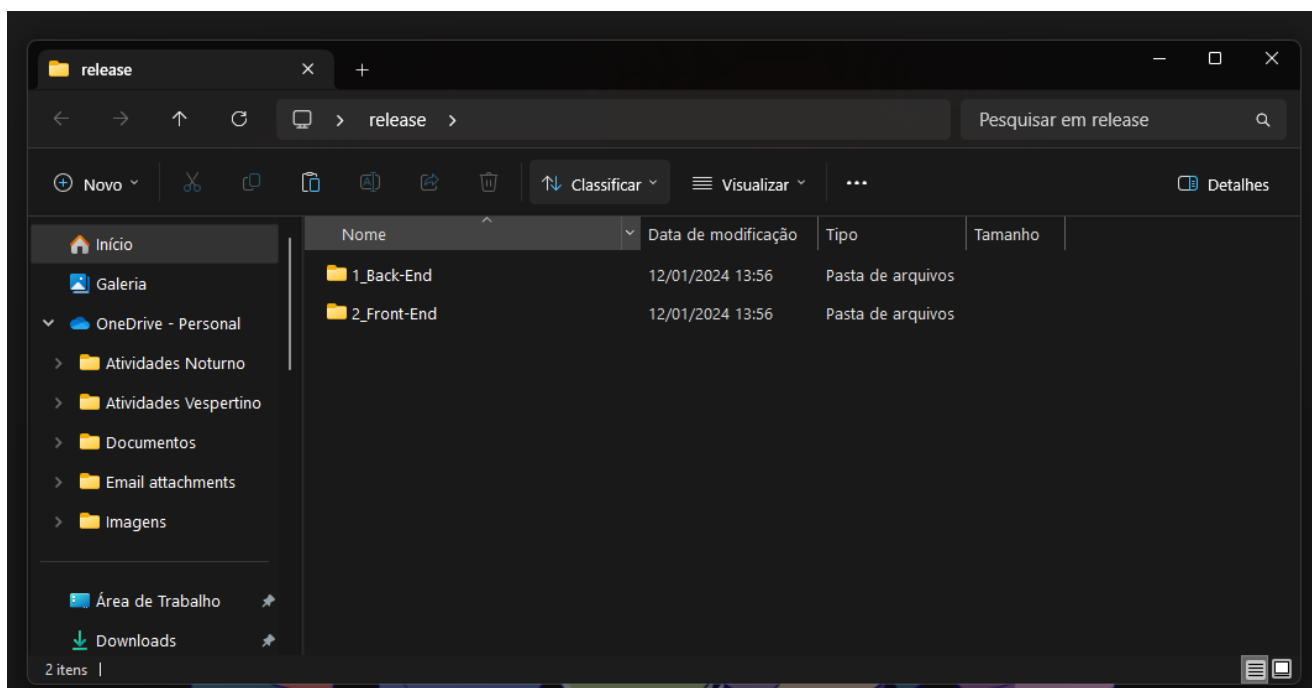
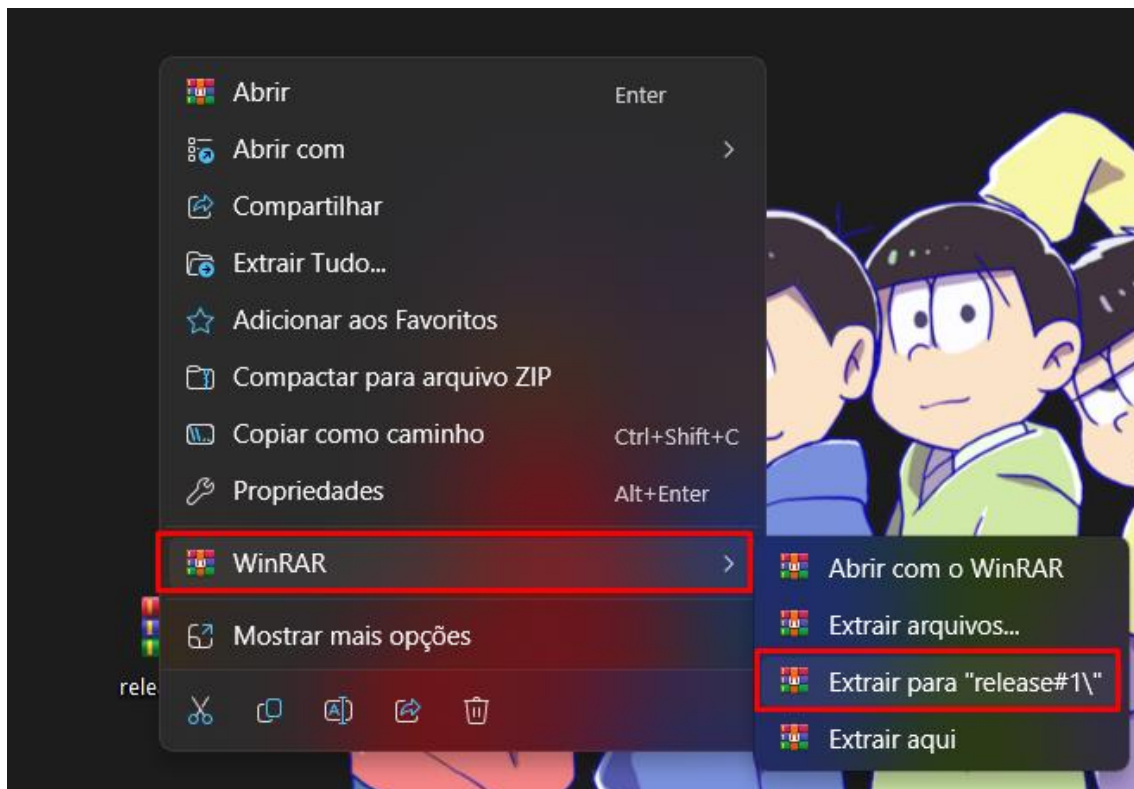


Ao clicar no símbolo de “seta para baixo” do botão verde: **“Code”**, realize o download do projeto clicando no botão: **“Download ZIP”**.

Concluindo o download, extraia o ZIP para um diretório do seu computador qual saiba que haja permissão de controle total do Windows.

A localização do projeto não importa muito, sendo assim, escolha um local de sua preferência, neste manual será executado diretamente da Área de Trabalho.

Para extrair, localize o zip baixado pelo navegador, clique com o botão direito e extraia no local de sua preferência.



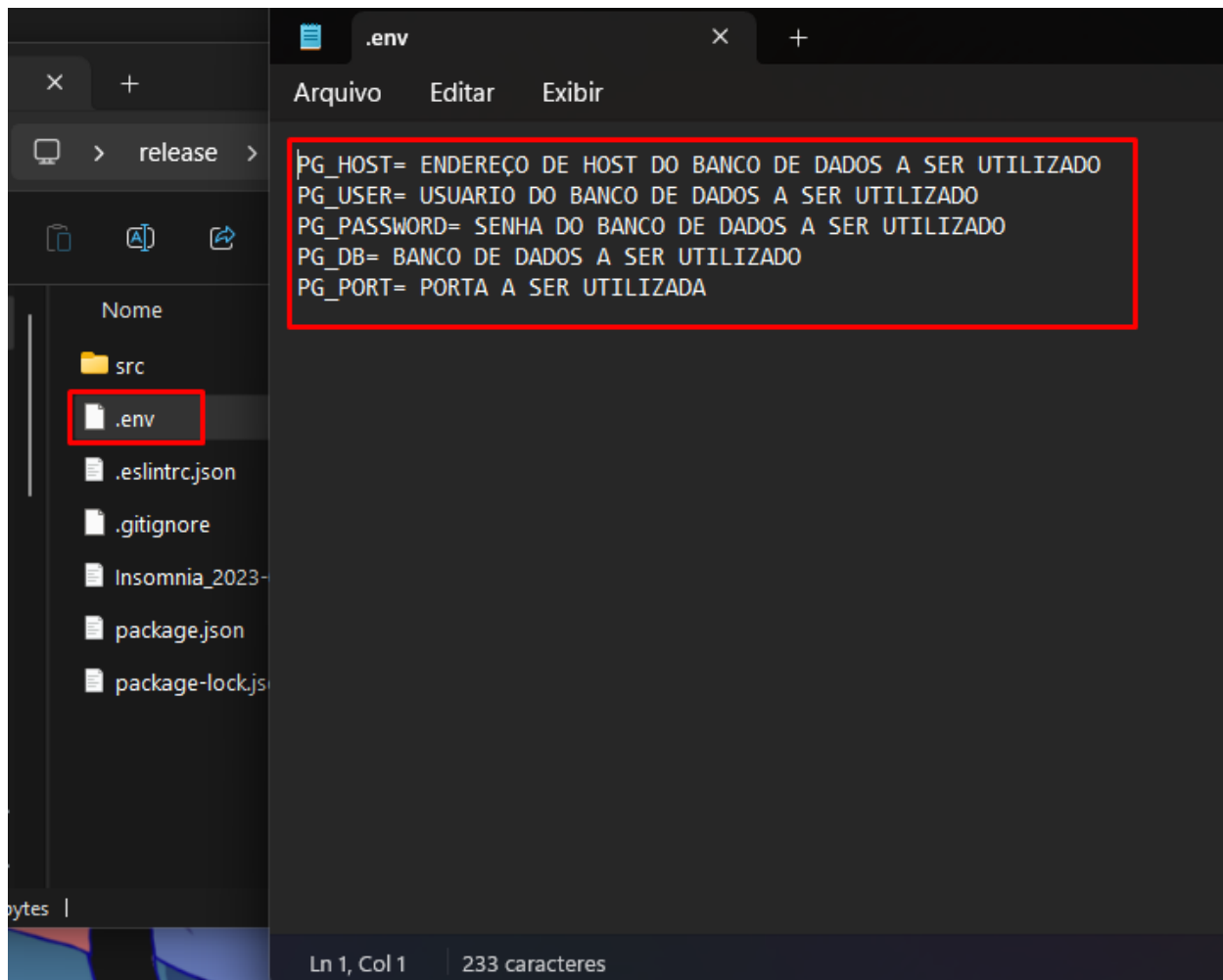
Ao concluir a extração dos arquivos, será criada a pasta: “release” contendo os diretórios:

- “1_Back-End”
- “2_Front-End”

Será necessário configurar os seguintes arquivos:

1_Back-End (Address Monitor): API com ligação ao banco de dados.

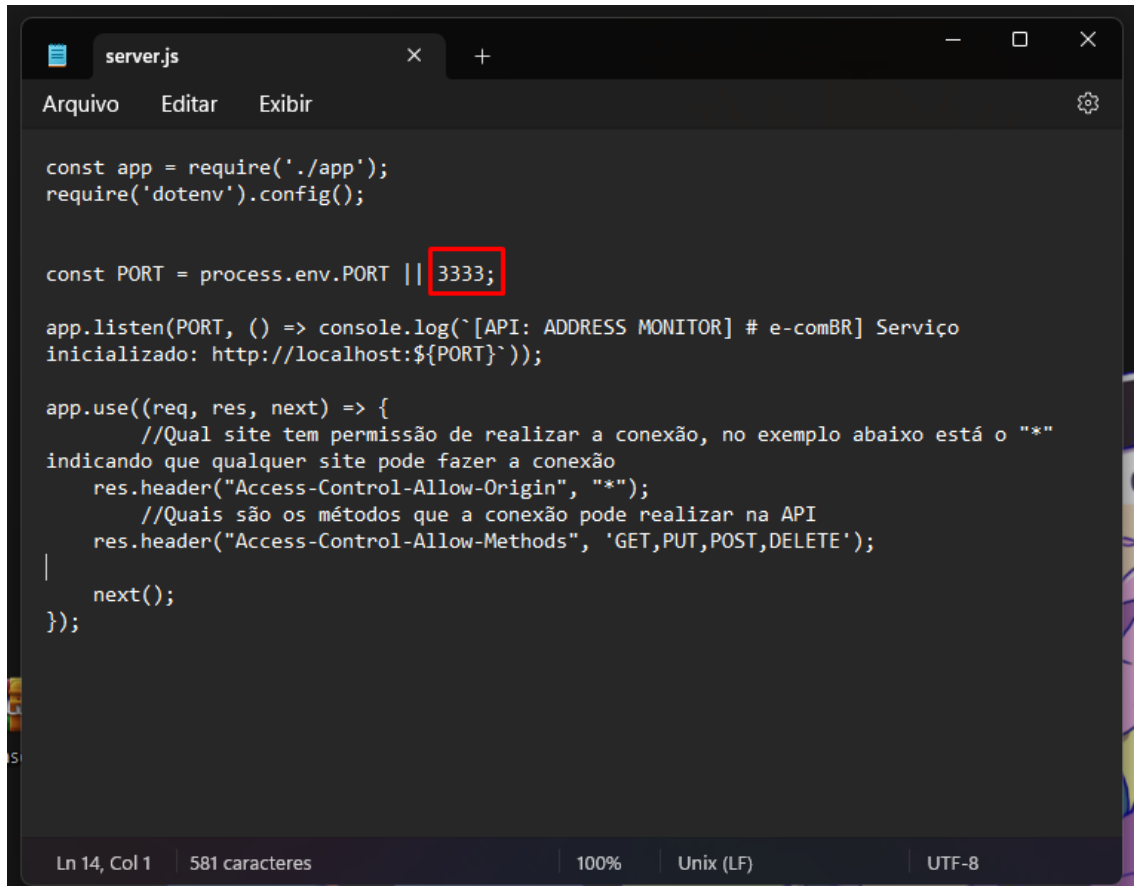
Abrir o arquivo .env presente no diretório: ..\release\1_Back-End\.env



- **PG_HOST:**
 - ENDEREÇO DE HOST DO BANCO DE DADOS A SER UTILIZADO;
- **PG_USER:**
 - USUARIO DO BANCO DE DADOS A SER UTILIZADO;
- **PG_PASSWORD:**
 - SENHA DO BANCO DE DADOS A SER UTILIZADO;
- **PG_DB:**
 - BANCO DE DADOS A SER UTILIZADO;
- **PG_PORT:**
 - PORTA A SER UTILIZADA.

..\release\1_Back-End\src\server.js

Caso seja necessário rodar a API em porta diferente do padrão configurado, basta alterar no arquivo “**server.js**” a porta necessária.



```
server.js
Arquivo  Editar  Exibir

const app = require('./app');
require('dotenv').config();

const PORT = process.env.PORT || 3333;

app.listen(PORT, () => console.log(`[API: ADDRESS MONITOR] # e-comBR] Serviço
inicializado: http://localhost:${PORT}`));

app.use((req, res, next) => {
  //Qual site tem permissão de realizar a conexão, no exemplo abaixo está o "*"
  indicando que qualquer site pode fazer a conexão
  res.header("Access-Control-Allow-Origin", "*");
  //Quais são os métodos que a conexão pode realizar na API
  res.header("Access-Control-Allow-Methods", 'GET,PUT,POST,DELETE');
  |
  next();
});

Ln 14, Col 1  581 caracteres  100%  Unix (LF)  UTF-8
```

Com a porta escolhida, será necessário configurar o sistema de SALT e tabela a ser utilizada pela API. Para isso, abrir o arquivo: “enderecosModel.js” presente no diretório:
..\release\1_Back-End\src\models

Todos os arquivos dentro deste diretório, possuem configuração de SALT, e de TABELA, com exceção de: “filtrosModel.js” que recebe a tabela por parâmetro da interface, sendo necessário abrir TODOS e alterar var key = “ para var key = ‘salt_desejado’ e var tabela = “ para var tabela = ‘tabela_desejada’.

Lembre-se: O salt deverá ser o mesmo em todos os arquivos, e posteriormente, haverá configuração do salt na pasta: “2_Front-End”, qual também deverá ser o mesmo.

```
JS main.js JS filtroValidate.js JS enderecosModel.js
C: > Users > mrica > Desktop > release > 1_Back-End > src > models > JS enderecosModel.js > ...
1  const connection = require('./connection')
2  const axios = require('axios')
3
4  var key = '' // SALT da API (Usado para ter permissão ao rodar os métodos de busca ou inserção)
5  const tabela = '' // Tabela a ser utilizada
6  var urlapi = 'https://nominatim.openstreetmap.org' //Endereço da API que fará a correção dos endereços não colocar / no final.
7  var tempoConsulta = 2000 //Intervalo em milisegundos para utilização da API gratuita.
```

Ainda no arquivo enderecosModel.js é necessário configurar a url da api que será utilizada na correção dos endereços, e um tempo de consulta, caso a api seja local deixe o tempoConsulta como 0 e altere a url da api para a hospedagem local.

Após configurar todos os “models” com o SALT, podemos começar a configurar o node para executar a API.

Para isso, acessar a página: <https://nodejs.org/en> e baixar a versão LTS.

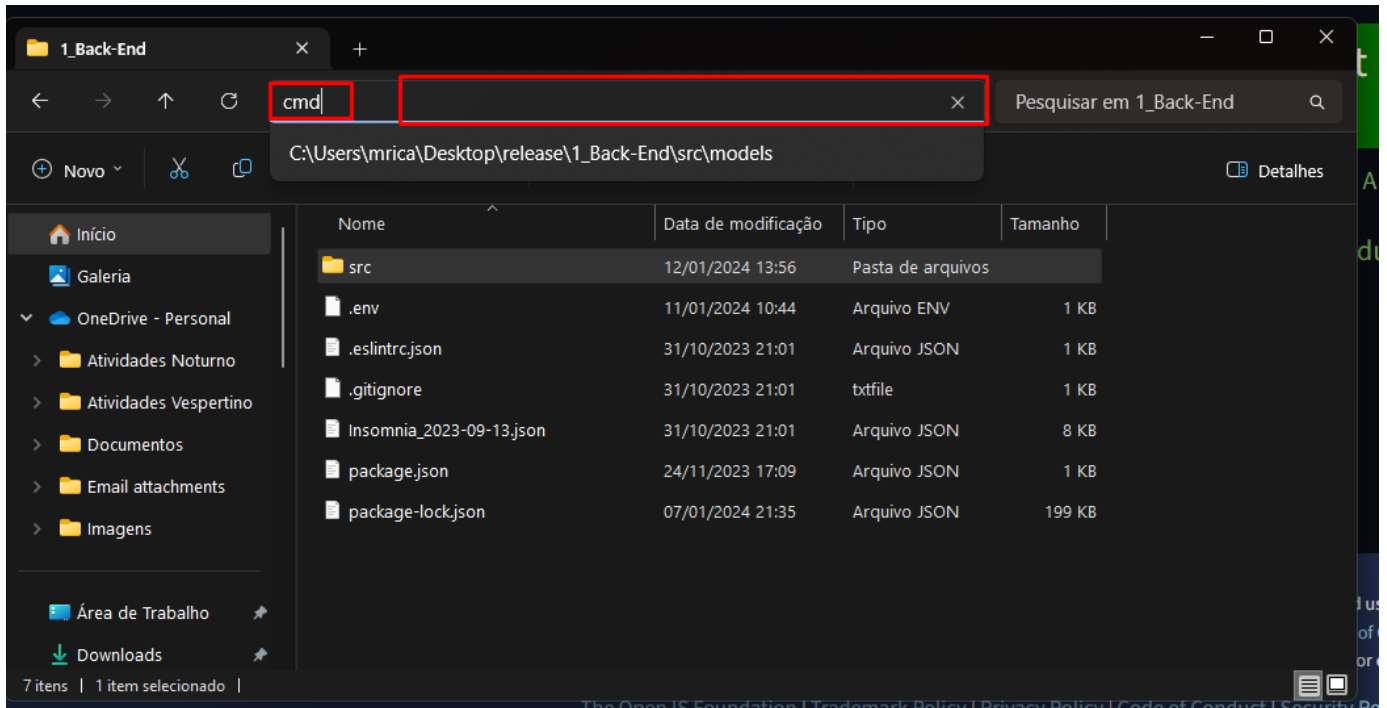
The screenshot shows the Node.js website with the following content:

- Header: node.js logo, navigation links (LEARN, ABOUT, DOWNLOAD, GUIDES, BLOG, DOCS, CERTIFICATION).
- Main text: Node.js® is an open-source, cross-platform JavaScript runtime environment.
- Section: Download Node.js®
- Buttons:
 - 20.11.0 LTS** (highlighted with a red box): Recommended For Most Users
 - 21.5.0 Current: Latest Features
- Footer links: Other Downloads | Changelog | API Docs (repeated twice).
- Text: For information about supported releases, see the [release schedule](#).

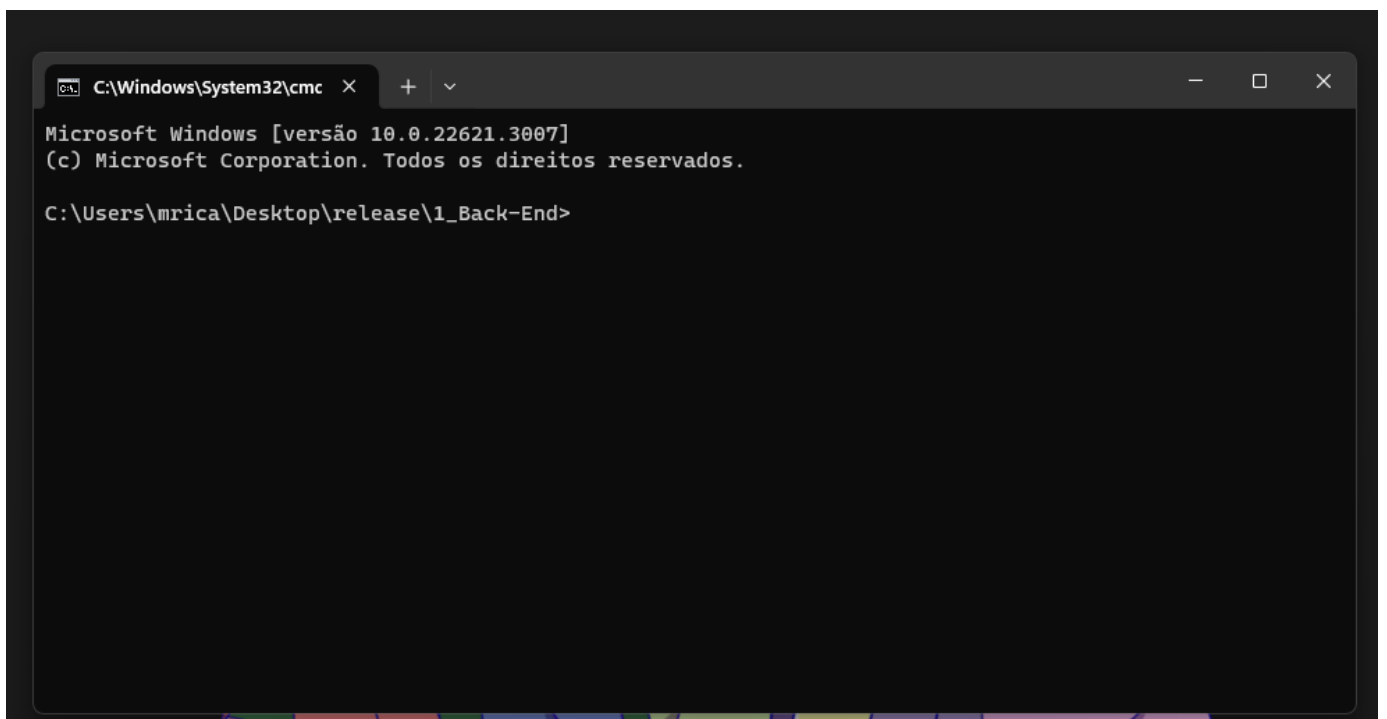
Para instalar o node, não há parametrização necessária, sendo somente avançar todas as etapas do instalador.

Ao concluir a instalação o computador será reinicializado.

Após a reinicialização, com tudo pronto, acesse o diretório: `..\release\1_Back-End\` clique na barra de endereços do explorador de arquivos do Windows e digite: `CMD`;



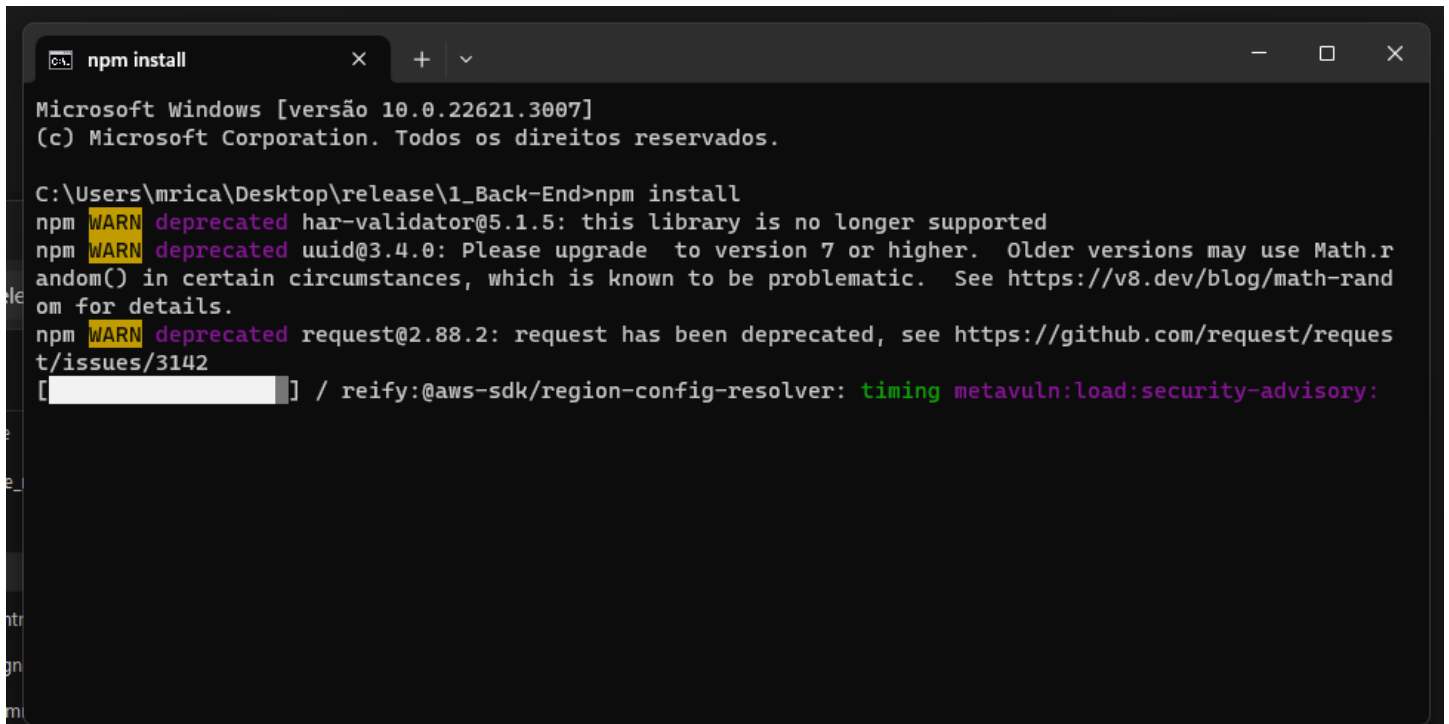
Realizar o procedimento acima fará com que seja inicializado o prompt de comando já com o diretório : `..\release\1_Back-End\` aberto nele.



No prompt de comando, cmd, digite o seguinte comando:

npm install

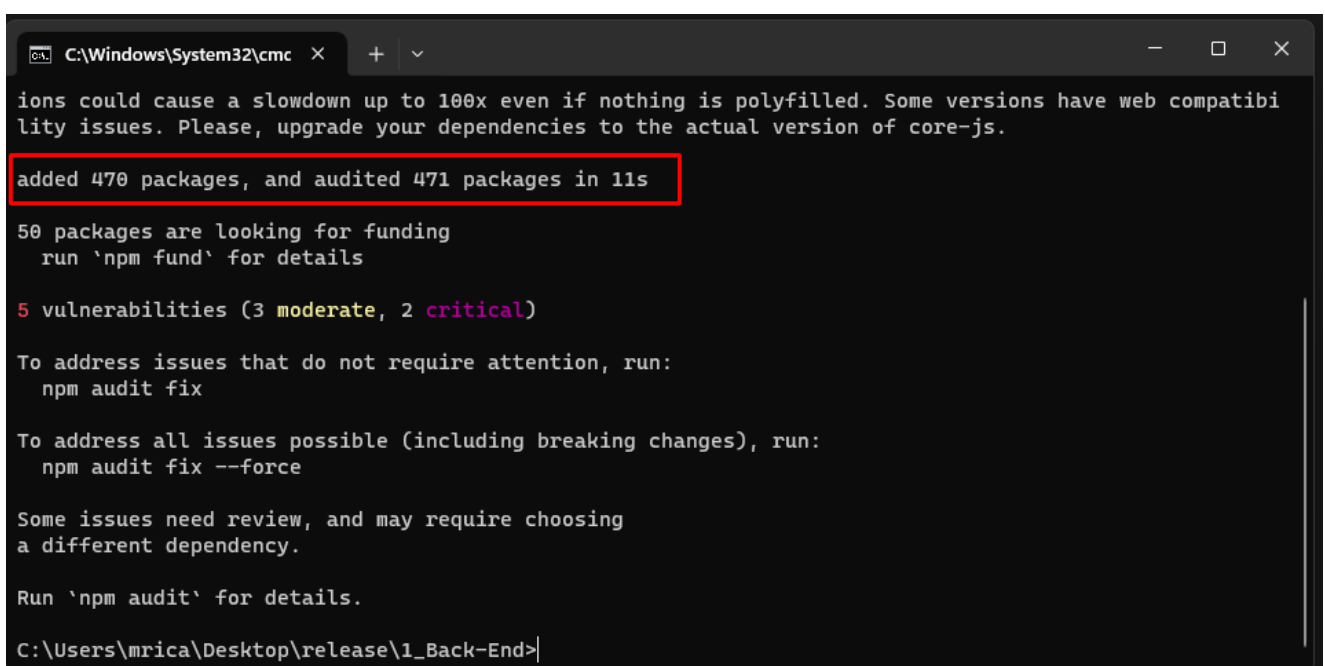
será necessário aguardar a resolução das dependências:



```
Microsoft Windows [versão 10.0.22621.3007]
(c) Microsoft Corporation. Todos os direitos reservados.

C:\Users\mrca\Desktop\release\1_Back-End>npm install
npm WARN deprecated har-validator@5.1.5: this library is no longer supported
npm WARN deprecated uuid@3.4.0: Please upgrade to version 7 or higher. Older versions may use Math.random() in certain circumstances, which is known to be problematic. See https://v8.dev/blog/math-random for details.
npm WARN deprecated request@2.88.2: request has been deprecated, see https://github.com/request/request/issues/3142
[REDACTED] / reify:@aws-sdk/region-config-resolver: timing metavuln:load:security-advisory:
```

A instalação estará completa após a seguinte mensagem, a quantidade de “packages” instaladas pode variar conforme o passar do tempo, sendo assim, desconsidere.

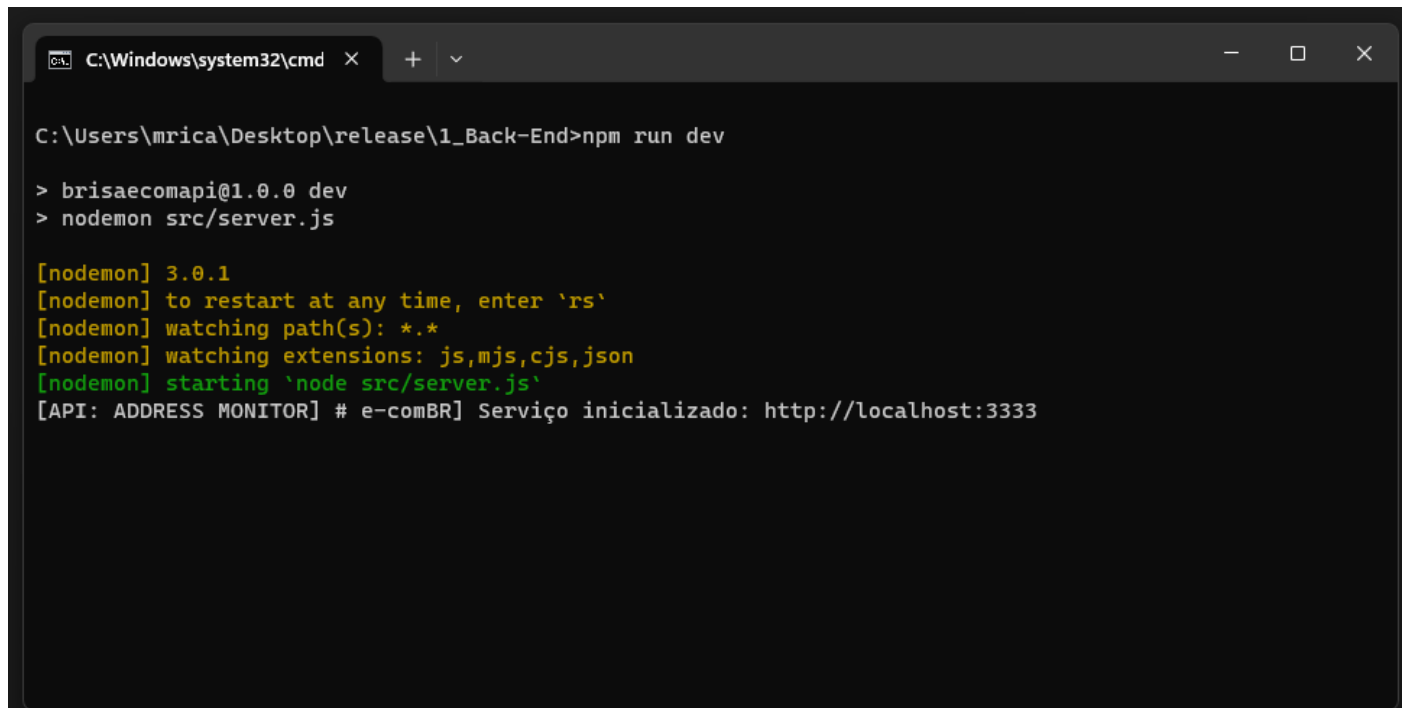


```
C:\Windows\System32\cmd > npm install
ions could cause a slowdown up to 100x even if nothing is polyfilled. Some versions have web compatibility issues. Please, upgrade your dependencies to the actual version of core-js.
added 470 packages, and audited 471 packages in 11s
50 packages are looking for funding
  run `npm fund` for details
5 vulnerabilities (3 moderate, 2 critical)
To address issues that do not require attention, run:
  npm audit fix
To address all issues possible (including breaking changes), run:
  npm audit fix --force
Some issues need review, and may require choosing
a different dependency.
Run `npm audit` for details.
C:\Users\mrca\Desktop\release\1_Back-End>
```


Ao concluir toda a instalação de dependências e pacotes, execute o seguinte comando para colocar a API no ar:

```
npm run dev
```

Ao executar o comando acima, a tela do prompt de comando deverá responder com:



```
C:\Windows\system32\cmd X + v

C:\Users\mrca\Desktop\release\1_Back-End>npm run dev

> brisaecomapi@1.0.0 dev
> nodemon src/server.js

[nodemon] 3.0.1
[nodemon] to restart at any time, enter 'rs'
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting 'node src/server.js'
[API: ADDRESS MONITOR] # e-comBR] Serviço inicializado: http://localhost:3333
```

Com isso os métodos que ficarão disponíveis são:

<http://host:porta/criarEstatisticas>

<http://host:porta/corrigerEnderecos/:pass>

<http://host:porta/buscarTodosSemLatLong>

<http://host:porta/buscarPorNumero/:numero>

<http://host:porta/buscarFiltro/:consulta/:pass>

<http://host:porta/buscarFiltros/:schema,:tabela,:pass>

Exemplo de uso

<http://localhost:3333/buscarFiltro/select> * from pessoas where nome like 'ana'/32952@@

Não é possível realizar consultas sem um salt configurado. Caso haja a necessidade de usar com um salt “fácil” considere colocar apenas um dígito de salt.

Para corrigir os endereços que não possuem latitude e longitude é utilizado o método:

corrigirEndereços que deve ser passado em conjunto do SALT configurado.

Exemplo: <http://localhost:3333/corrigirEnderecos/32952@@@>

Com isso o prompt de comandos responderá:

```
[nodemon] 3.0.1
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node src/server.js`
[API: ADDRESS MONITOR] # e-comBR] Serviço inicializado: http://localhost:3333
Solicitação de Correção para: 6477 registros.
```

```
(3/6474) - [REDACTED] RUA E - nº 26 - ,
Correção a ser Aplicada: Lat: -16.4683783, Lon: -54.6189735
Corrigido no banco de dados com sucesso.
Solicitação Realizada em: Fri, 12 Jan 2024 19:01:17 GMT
Resposta Status: 200
```

2_Front-End (HeatMap+): View com motor de resolução de coordenadas geográficas ligado ao OpenLayers.

Para a parte de Front-end toda a configuração estará em apenas dois arquivos, sendo eles: “main.js” e “filtroValidate.js”;

Abra o diretório: \release\2_Front-End\derek\ e o arquivo: “main.js” que estará em seu interior.

```
C: > Users > mrica > Desktop > release > 2_Front-End > derek > JS main.js > ...
1 //CONFIGURAÇÕES
2 const host = "" //host da API
3 const porta = "" //porta da API
4 const schema = "" //determina o schema do postgres para auxiliar em consultas globais
5 const tabela = "" //determina a tabela de onde os filtros serão montados
6 const key = "" //usado como salt na API para não ser explorado como vulnerabilidade
7 const coordenada_padrao_mapa = [-5855349.118211474, -3140369.6419760953] //Posicionamento padrão do mapa quando criada a visualização em tela.
8
```

Toda a configuração necessária estará logo no começo do arquivo, portanto leia atentamente aos comentários e preencha os campos:

- `const host = ""`
 - //host da API
- `const porta = ""`
 - //porta da API
- `const schema = ""`
 - //determina o schema do postgres para auxiliar em consultas globais o padrão é: **public**
- `const tabela = ""`
 - //determina a tabela de onde os filtros serão montados
- `const key = ""`
 - //usado como salt na API para não ser explorado como vulnerabilidade
- `const coordenada_padrao_mapa = [-5855349.118211474, -3140369.6419760953]`
 - //Posicionamento padrão do mapa quando criada a visualização em tela.

Feitas as configurações, o próximo passo será instalar todas as dependências e pacotes necessários para rodar o front-end, para isso, utilizaremos o vite através do node.

Navegue até o diretório: ..\release\2_Front-End clique na barra de endereços do explorador de arquivos do Windows e execute o seguinte comando:

npm install

```
C:\Windows\System32\cmd  x + v - □ x  
C:\Users\mrca\Desktop\release\2_Front-End>npm install
```

```
npm install  x + v - □ x  
C:\Users\mrca\Desktop\release\2_Front-End>npm install  
[ ] - reify:@esbuild/android-arm: timing reifyNode:node_modules/@esbuild
```

Ao concluir a instalação de todas as dependências, o prompt responderá:

```
C:\Windows\System32\cmd  x  +  v  -  □  x

C:\Users\mrca\Desktop\release\2_Front-End>npm install
npm WARN deprecated core-js@2.6.12: core-js@<3.23.3 is no longer maintained and not recommended for usage due to the number of issues. Because of the V8 engine whims, feature detection in old core-js versions could cause a slowdown up to 100x even if nothing is polyfilled. Some versions have web compatibility issues. Please, upgrade your dependencies to the actual version of core-js.

added 257 packages, and audited 258 packages in 9s

87 packages are looking for funding
  run `npm fund` for details

1 moderate severity vulnerability

To address all issues, run:
  npm audit fix

Run `npm audit` for details.

C:\Users\mrca\Desktop\release\2_Front-End>
```

Para subir a interface, utilize o comando:

npm start

O vite deverá responder da seguinte maneira no prompt de comandos:

```
C:\Windows\system32\cmd  x  +  v  -  □  x

VITE v3.2.7  ready in 210 ms

→ Local:    http://localhost:5173/
→ Network:  use --host to expose
```

IMPORTANTE

Caso opte por continuar usando o vite para executar a view, será necessário iniciar o serviço com o comando:

npm start -- --host

Esse comando fará com que o vite dê as devidas permissões para acessar o host de forma externa à rede LAN.

Ao utilizar o comando acima, o vite responderá da seguinte maneira no prompt de comandos:

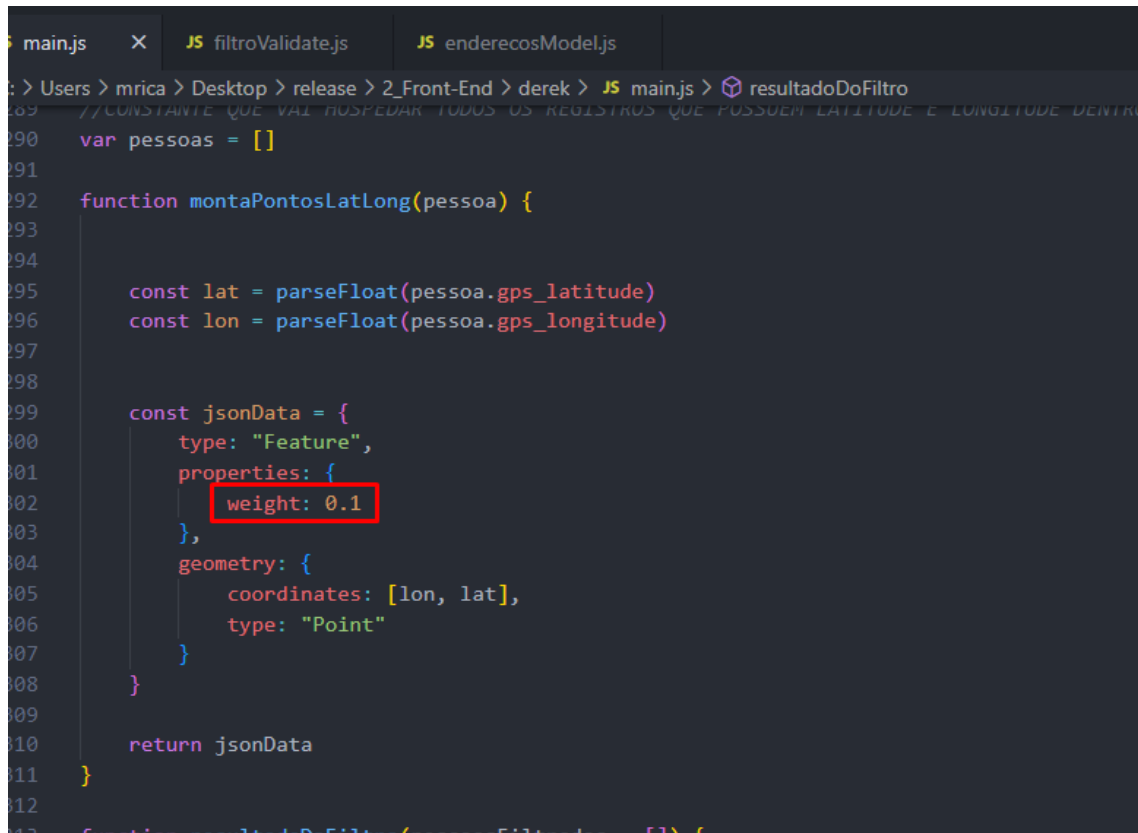
```
VITE v3.2.7 ready in 190 ms

→ Local:    http://localhost:5173/
→ Network:  http://26.17.8.178:5173/
→ Network:  http://192.168.56.1:5173/
→ Network:  http://192.168.0.104:5173/
→ Network:  http://192.168.196.213:5173/
→ Network:  http://172.31.32.1:5173/
```

NÃO É OBRIGATÓRIO USAR O VITE COMO SERVIDOR, SENDO ASSIM É POSSÍVEL CONFIGURAR OUTROS SERVIDORES WEB PARA RODAR A APLICAÇÃO, DESDE QUE COMPATÍVEIS COM JAVASCRIPT PURO.

PERSONALIZAÇÃO

Para alterar a densidade dos pontos de calor, procure por: “weight” no arquivo: “main.js” localizado no diretório ..\release\2_Front-End\derek\ e altere conforme sua necessidade.



```
main.js x JS filtroValidate.js JS enderecosModel.js
: > Users > mica > Desktop > release > 2_Front-End > derek > JS main.js > resultadoDoFiltro
189 //CONSTANTE QUE VAI HOSPEDAR TODOS OS REGISTROS QUE POSSUEM LATITUDE E LONGITUDE DENTRO
290 var pessoas = []
291
292 function montaPontosLatLong(pessoa) {
293
294
295     const lat = parseFloat(pessoa.gps_latitude)
296     const lon = parseFloat(pessoa.gps_longitude)
297
298
299     const jsonData = {
300         type: "Feature",
301         properties: {
302             weight: 0.1
303         },
304         geometry: {
305             coordinates: [lon, lat],
306             type: "Point"
307         }
308     }
309
310     return jsonData
311 }
312
313 function resultadoDoFiltro(pessoasFiltradas = []) {
```

Para alterar a circunferência dos pontos de calor, procure por: “radius” no arquivo: “main.js” localizado no diretório ..\release\2_Front-End\derek\



```
//CONFIGURA A APARÊNCIA DOS PONTOS DE CALOR
const heatMapPontosCalor = new ol.layer.Heatmap({
    source: new ol.source.Vector({
        features: [],
    }),
    radius: 20,
    blur: 12,
    gradient: ['#0000FF', '#00FF00', '#FFFF00', '#FFA500', '#FF0000'],
    title: 'PontosCalor',
    visible: true,
})
```

Para alterar a força do espectro do ponto de calor, procure por: “blur” no arquivo: “main.js” localizado no diretório ..\release\2_Front-End\derek\

```
//CONFIGURA A APARÊNCIA DOS PONTOS DE CALOR
const heatMapPontosCalor = new ol.layer.Heatmap({
  source: new ol.source.Vector({
    features: [],
  }),
  radius: 20,
  blur: 12,
  gradient: ['#0000FF', '#00FF00', '#FFFF00', '#FFA500', '#FF0000'],
  title: 'PontosCalor',
  visible: true,
})
```

Para alterar a paleta de cores dos pontos de calor, procure por: “gradient” no arquivo: “main.js” localizado no diretório ..\release\2_Front-End\derek\ e altere o vetor de cores utilizando o código hexadecimal da cor desejada.

```
//CONFIGURA A APARÊNCIA DOS PONTOS DE CALOR
const heatMapPontosCalor = new ol.layer.Heatmap({
  source: new ol.source.Vector({
    features: [],
  }),
  radius: 20,
  blur: 12,
  gradient: ['#0000FF', '#00FF00', '#FFFF00', '#FFA500', '#FF0000'],
  title: 'PontosCalor',
  visible: true,
})
```