IPV – Instituto Politécnico de Viseu

ESTGV – Escola Superior de Tecnologia e Gestão de Viseu

Departamento de Informática

Relatório do Projeto de Aplicações para a Internet II

Licenciatura em Engenharia Informática

Realizado na Unidade Curricular de

Aplicações para a Internet II

por

David Albuquerque, 18734

Alexandre Lourenço, 18735

André Marques, 18722

Hélder Lourenço, 18751

João Crespo, 18708

Orientadores

ESTGV: Fernando Almeida, Nuno Costa e Steven Abrantes

Viseu, 2021

IPV – Instituto Politécnico de Viseu

ESTGV – Escola Superior de Tecnologia e Gestão de Viseu

Departamento de Informática

Relatório do Projeto de Aplicações para a Internet II

Licenciatura em Engenharia Informática

Realizado na Unidade Curricular de

Aplicações para a Internet II

por

David Albuquerque, 18734

Alexandre Lourenço, 18735

André Marques, 18722

Hélder Lourenço, 18751

João Crespo, 18708

Orientadores

ESTGV: Fernando Almeida, Nuno Costa e Steven Abrantes

Viseu, 2021

Índice

1 Introdução

Este relatório visa o desenvolvimento do projeto da UC de Aplicações para a Internet II em que se pretende a criação de um website completo com backend, frontend e chamadas a uma API para obter dados de uma base de dados.

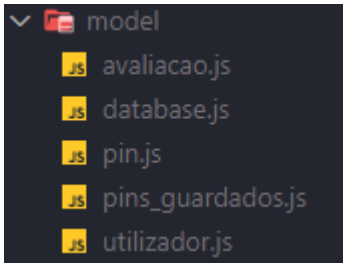
O projeto foi implementado com recurso à ferramenta Visual Studio Code e outras ferramentas menos relevantes.

2 Execução do Projeto

2.1 Backend

2.1.1 Modelos

Para este trabalho foram criados 5 modelos para armazenar os dados corretamente:



Exemplo do modelo pin:

```
var Sequelize = require('sequelize');
var sequelize = require('./database');

// importa o modelo - chave forasteira utilizadorId
var utilizador = require('./utilizador');
var pin = sequelize.define('pin', {
    id: {
        type: Sequelize.INTEGER,
        primaryKey: true,
        autoIncrement: true,
    },
    nivel: Sequelize.INTEGER,
    upvote: Sequelize.INTEGER,
    downvote: Sequelize.INTEGER,
    latitude: Sequelize.FLOAT,
    longitude: Sequelize.FLOAT,
    datap: Sequelize.DATEONLY,
    horap: Sequelize.TIME,
    utilizadorId: {
        type: Sequelize.INTEGER,
        // referência a outro modelo
        references: {
            model: utilizador,
            key: 'id'
        }
    }
},{timestamps: false});

pin.belongsTo(utilizador)
module.exports = pin
```

2.1.2 Controladores

Foram também criados vários controladores para cada modelo, controladores estes que vão realizar operações como inserir um novo utilizador ou um novo pin na base de dados. Na grande maioria, os controladores chamam funções da base de dados que vão realizar as operações, por motivos de eficiência.

Exemplo de uma função do Controlador Pin:

```
/* Inserir Upvote e Downvote ---------------------- */
controllers.inserir_up_down = async (req,res) => {
    const { email,id,upvote,downvote} = req.params;
    sequelize.query("select pin_vote_update('"+email+"','"+id+"','"+ upvote + "','"+downvote+"');")
    .then(data => {
      res.status(200)
        .json(data);
    })
    .catch(error => {
        console.log(error);
        next();
    })
}
```

2.1.3 Rotas

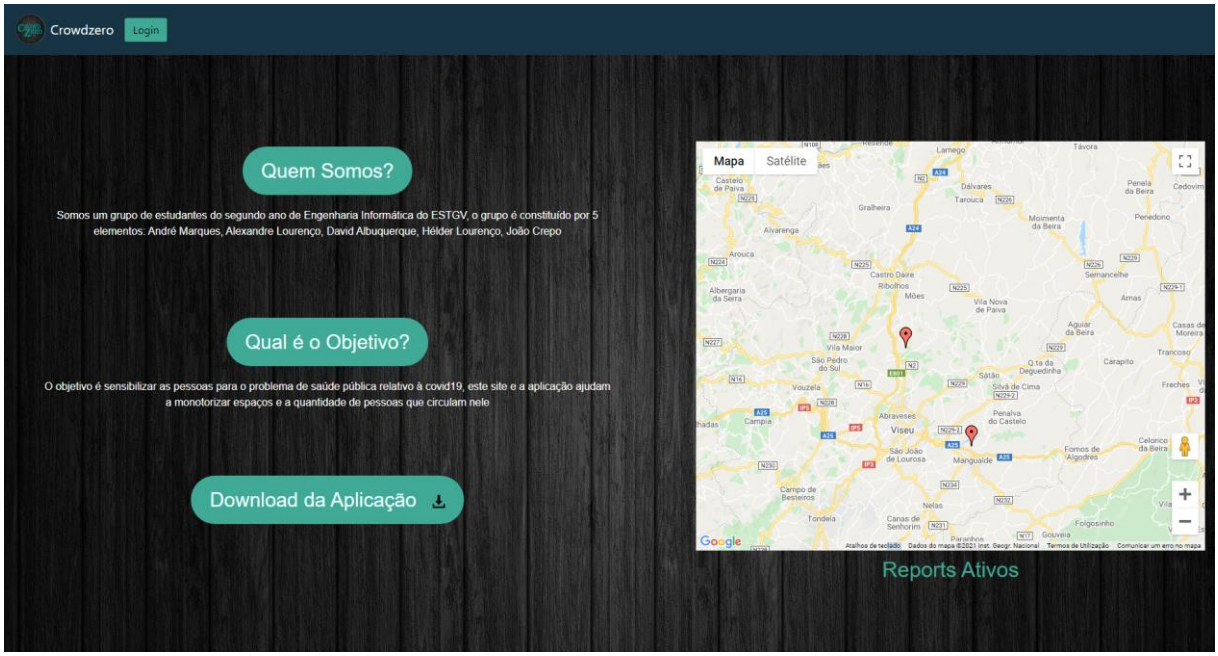
Rotas estabelecidas para o funcionamento do backend e utilização dos controladores e modelos definidos:

```
const express = require('express');
const app = express();
const middleware = require('./middleware');
const utilizadorRouters = require('./routes/utilizadorRoute.js')
const pinRouters = require('./routes/pinRoute.js')
const pins_guardadosRouters = require('./routes/pins_guardadosRoute.js')
const avaliacaoRouters = require('./routes/avaliacaoRoute.js')

app.set('port', process.env.PORT|| 3000);

app.use(express.json());
app.use((req, res, next) => {
    res.header('Access-Control-Allow-Origin', '*');
    res.header('Access-Control-Allow-Headers', 'Authorization, X-API-KEY, Origin, X-Requested-With, Content-Type, Accept, Access-Control-Allow-Request-Method');
    res.header('Access-Control-Allow-Methods', 'GET, POST, OPTIONS, PUT, DELETE');
    res.header('Allow', 'GET, POST, OPTIONS, PUT, DELETE');
    next();
    });
app.use('/utilizador',utilizadorRouters)
app.use('/pin',pinRouters)
app.use('/pins_guardados',pins_guardadosRouters)
app.use('/avaliacao',avaliacaoRouters)

app.listen(app.get('port'),()=>{
console.log("Start server on port "+app.get('port'))
})
```

2.2 Frontend

2.2.1 Homepage

Assim que o site inicia, somos levados para a Homepage, que contém um mapa com os pins ativos em tempo real, duas pequenas caixas de texto que explicam quem somos e qual é o objetivo deste projeto. A Homepage contém ainda um botão para fazer Login e outro botão para descarregar a aplicação aliada a este site.



A componente do Mapa que mostra os pins ativos de momento é também reutilizada na vista Home, renderizada com a API do GoogleMaps para JavaScript. Aqui pode ver-se como é retornado o JSX:

```
<h2 class='loading'>{data.loading ? "Loading..." : "Reports Ativos"}</h2>
<GoogleMap defaultZoom={10} defaultCenter={{ lat: 40.661037124, lng: -7.9120189460}}>

  {props.isMarkerShown && (data.markers || []).map((marker) => (
      <Marker
        onClick={()=>atribuilink(marker)}
        key={marker.id}
        position={{ lat: marker.latitude, lng: marker.longitude }}
        icon={dvnivel(marker.nivel)}
      />
    ))
  }

</GoogleMap>
```

A componente é construída com propriedades e possui uma constante “data” que contém os dados dos pins a mostrar, a tag “Loading” e ainda, se algo correr mal, armazena o erro:

```
const [data, setData] = useState({
  markers: [],
  loading: false,
  error: undefined
});
```

Esta constante é depois renderizada através da função map().

Esta componente do Mapa beneficia ainda da função dvnivel() que atribui a cor ao pin consoante o nível de concentração e a função atribuilink() que faz com que cada pin seja clicável e abra uma nova janela no google maps com o pin colocado no local exato, tornando o site mais ágil e fácil de usufruir das funcionalidades do Google Maps, como por exemplo marcar um trajeto para o local.

Função dvnivel():

```
function dvnivel(props){
  if(props === 1)
  {
    return "http://maps.google.com/mapfiles/ms/icons/green-dot.png"
  }
  if(props === 2)
  {
    return "http://maps.google.com/mapfiles/ms/icons/yellow-dot.png"
  }
  if(props === 3)
  {
    return "http://maps.google.com/mapfiles/ms/icons/red-dot.png"
  }
}
```

Função atribuilink():

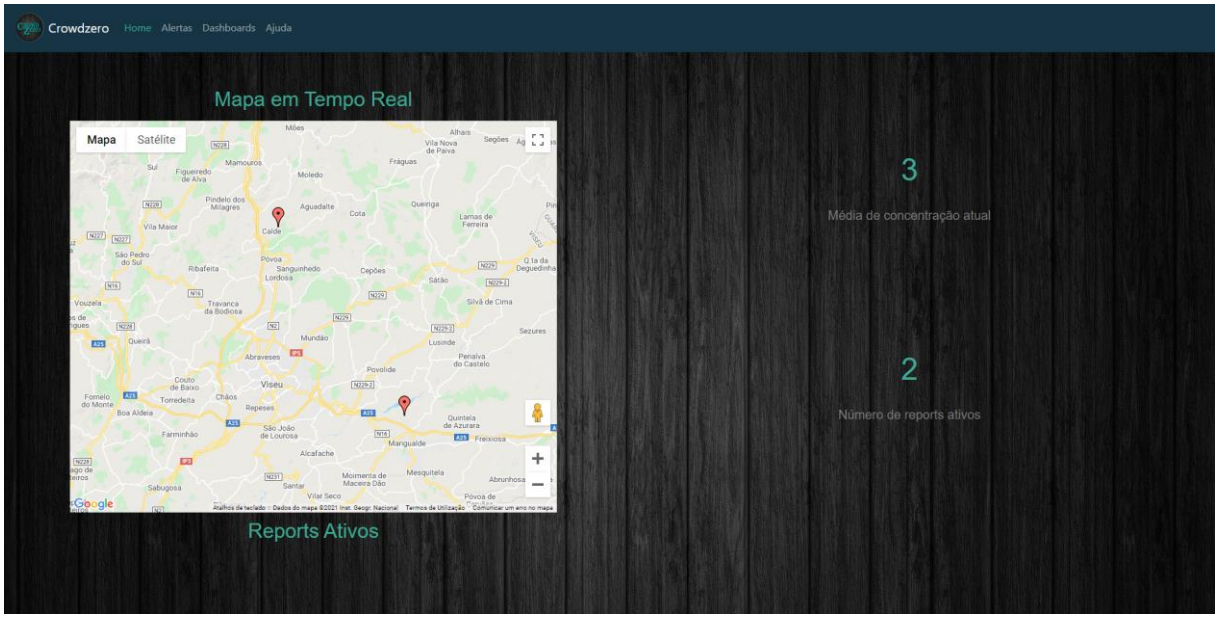
```
function atribuilink(data){
  var strLink= "https://www.google.com/maps/search/?api=1&query="+ data.latitude+"," + data.longitude
  window.open(strLink, "_blank");
}
```

2.2.2 Login

A vista de Login é uma vista simples, enquadrada no tema do website, em que o utilizador pode introduzir as suas credenciais e, se tiver privilégios de administrador, conseguir aceder ao resto do site.

2.2.3 Home

Depois de o utilizador com privilégios de administrador fazer login, será redirecionado para a página Home, que contém, mais uma vez, o mapa em tempo real com os reports/pins ativos. Esta página Home contém também o número de reports ativos a média de concentração atual.

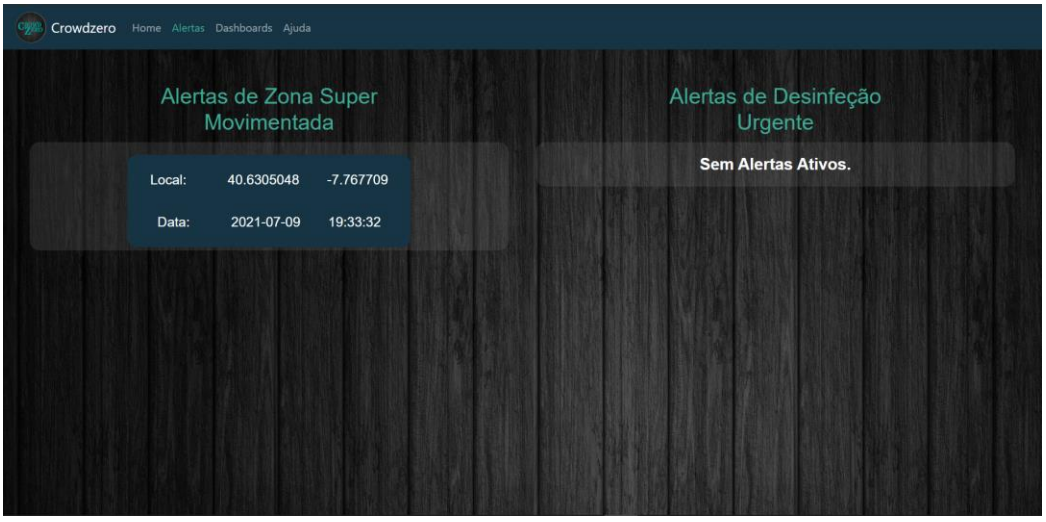


Estes dois números são calculados de forma simples usando o React Hook UseEffect, que quando recebe os JSON array com os pins ativos atribui os valores às variáveis a mostrar:

```
useEffect(()=>{
    axios.get('https://warm-hamlet-63390.herokuapp.com/pin/list')
    .then(res=>{
      console.log(res.data);

        var nivel = 0;
        setPins(res.data.data.length)
        for(var i = 0; i < res.data.data.length; i++){
          nivel = nivel + res.data.data[i].nivel
        }

        if(res.data.data.length === 0)
        {
          setLvl(0)
        }
        else{
          var coco = nivel/res.data.data.length
        setLvl(Math.round(coco * 100) / 100)
        }
    })
    .catch(err =>{
        console.log(err);
    })
}, [])
```

2.2.4 Alertas

Na vista de alertas serão mostrados os locais que são Super Movimentados ou que precisam de Desinfeção urgente. Estes dois conceitos estão explicados na vista Ajuda e são calculados também de forma relativamente simples: Se o pin corresponder a que foi definido para cada tipo de alerta, é colocado no array do respetivo alerta e depois colocado no ecrã através da função map:



Estes alertas são também clicáveis e redirecionam para o Google Maps, mais uma vez.

Exemplo do cálculo de Zona de Desinfeção Urgente:

```
useEffect(()=>{
    axios.get('https://warm-hamlet-63390.herokuapp.com/pin/list')
    .then(res=>{
        var vetor = [];
        for(var i = 0; i < res.data.data.length; i++)
        {
            var horaatual = moment().format('LTS')
            var startTime = moment(res.data.data[i].horap, "HH:mm:ss");
            var endTime = moment(horaatual, "HH:mm:ss");

            var duration = moment.duration(startTime.diff(endTime));

            var nivelp = res.data.data[i].nivel;

            if(nivelp === 3 && duration._data.hours > 3)
            {
                    vetor.push(res.data.data[i])
            }
        }
        setPins(vetor)
    })
    .catch(err =>{
        console.log(err);
    })
}, [])
```

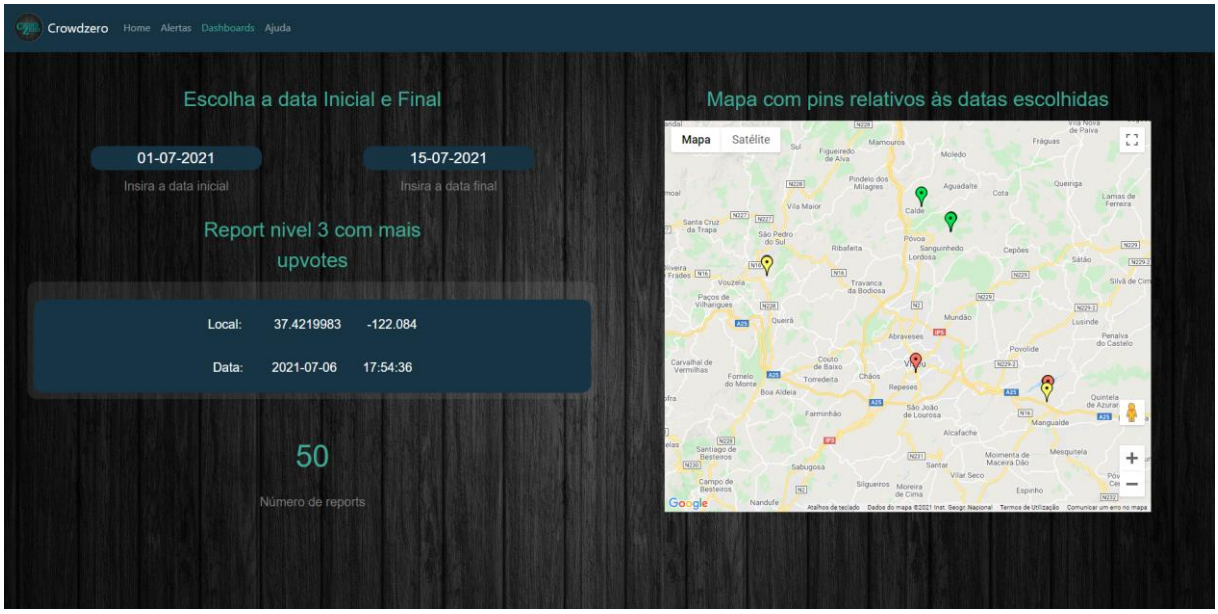
Renderização dos alertas:

```
if(pins.length === 0){
    return(
        <div className="tsp2">
            Sem Alertas Ativos.
       </div>
       )
}
else{
    return(
        pins.map(
            pins =>
                <a id="linkalertasingle" target="_blank" onClick={()=>atribuilink(pins)} href="">
                <div className="alerta_single" key={pins.id}>
                <div className="txtalerta">Local:      {pins.latitude}    {pins.longitude}</div>
                <br/>
                <div className="txtalerta">Data:     {pins.datap}    {pins.horap}</div>
                </div>
                </a>
            )
    )
}
```

2.2.5 Dashboards

A secção Dashboards do website contém também um mapa, desta vez um pouco diferente. Este mapa irá mostrar os pins que foram colocados entre as datas a selecionar pelo utilizador e o número de pins colocados nesse intervalo será também mostrado. Se o utilizador selecionar a mesma data em ambos os "DatePickers", serão mostrados os pins colocados nesse dia.

Esta vista mostra ainda o Pin de nível 3 com mais upvotes no intervalo de tempo selecionado, simbolizando assim a zona mais movimentada. Este pin simbolizado no ecrã através de texto é, mais uma vez, clicável.



Isto é feito através de uma função que recebe as duas datas selecionadas pelo utilizador e devolve um array com todos os dias nesse intervalo de tempo. De seguida o UseEffect, dependente das datas, irá colocar num array todos os pins cuja data está incluída no array de datas, ficando assim com um array que contém todos os pins colocados nesse intervalo de tempo

Depois de conseguirmos o array com os pins, estes são renderizados no mapa e são mostrados os dados mencionados acima.

```
function getDates (startDate, endDate) {

  const dates = []

  let currentDate = startDate

  function addDays(date, days) {
    var result = new Date(date);
    result.setDate(result.getDate() + days);
    result = moment(result).format("YYYY-MM-DD");
    return result;
  }

  while (moment(currentDate).isSameOrBefore(endDate)) {
    dates.push(currentDate)
    currentDate = addDays(currentDate, 1)
  }
  return dates
}
```

```
useEffect(()=>{
  axios.get('https://warm-hamlet-63390.herokuapp.com/pins_guardados/todospg')
  .then(res=>{

    maisup = res.data.data[0]

    var arraydatas = getDates(data1, data2)
    pinsdata = [];
    for(var i = 0; i < res.data.data.length; i++)
    {
      if(arraydatas.includes(res.data.data[i].data_pg))
      {
        pinsdata.push(res.data.data[i])
        if(maisup.upvote_pg < res.data.data[i].upvote_pg && res.data.data[i].nivel_pg === 3)
        {
          maisup = [];
          maisup = res.data.data[i]
        }
      }
    }
    setPorra(pinsdata)
  })
}, [startDate, startDate2])
```

2.2.6 Ajuda

Na vista “Ajuda”, como já foi mencionado, estão presentes definições de conceitos do Website, como por exemplo o que é uma zona super movimentada. Está ainda uma prevê informação acerca do que é o projeto Crowdzero e ainda contactos de ajuda.

3 Conclusão

Com a realização deste trabalho, adquirimos conhecimentos e capacidades no âmbito da UC de Aplicações para a Internet II, que com certeza nos serão úteis na nossa vida profissional.

Este trabalho também ajudou no desenvolvimento da nossa capacidade de trabalho em grupo e desenvolvimento de um backend e um frontend de um Website.