

MAPA DE CALOR BASEADO EM GEOLOCALIZAÇÃO INTERNA COM UTILIZAÇÃO DE BEACONS

Aluno: Leonardo Adriano Reichert

Orientadora: Luciana Pereira de Araújo

Roteiro

- Introdução
- Objetivos
- Fundamentação teórica
- Trabalhos correlatos
- Requisitos
- Especificação
- Implementação
- Resultados e discussões
- Conclusões

Introdução

- Obtenção da informação (ALCARÁ et al., 2010);
- Coleta, armazenamento e processamento (SILVA, 2011);
- Beacon Eddystone (PAIVA, 2016);
- Mapa de Calor (YAU, 2010).

Objetivos

Desenvolver uma ferramenta para mapear o uso de beacons em localização indoor

- Apresentar uma ferramenta para a montagem de páginas HTML de acordo com o beacon escolhido;
- Disponibilizar uma ferramenta para visualização dos beacons mais visitados através de um mapa de calor;
- Disponibilizar uma estrutura dinâmica de exibição de páginas HTML de acordo com o beacon acessado.

Fundamentação Teórica

- Sinal Bluetooth

- Classes

Classe	Potência(Mw)	Alcance(m)
1	100	100
2	2,5	10
3	1	1
4	0,5	0,5

- Versões

- RSSI, IPV6, FIPS;
 - BLE: baixo consumo de energia.

Fundamentação Teórica

- Beacons
 - BLE;
 - Somente envia dados;
 - Identificação;
 - iBeacons;
 - Eddystone
 - URLID;
 - Physical web local.

Fundamentação Teórica

- Monitoramento de pessoas
 - Privacidade;
 - Segurança;
 - Informação
 - Fluxo de pessoas;
 - Hot zones.
- Mapa de calor
 - Interpretação;
 - Unificação de informação.

Trabalhos Correlatos

- PayPal Beacon (PAYPAL, 2009);
- Microlocation (HANDCOM, 2014);
- Launch Here (AWWAPPS, 2013).

Trabalhos Correlatos: PayPal Beacon



- Pagamento;
- Identificação da Loja;
- Recorrência;
- Facilidade.

Trabalhos Correlatos: Microlocation

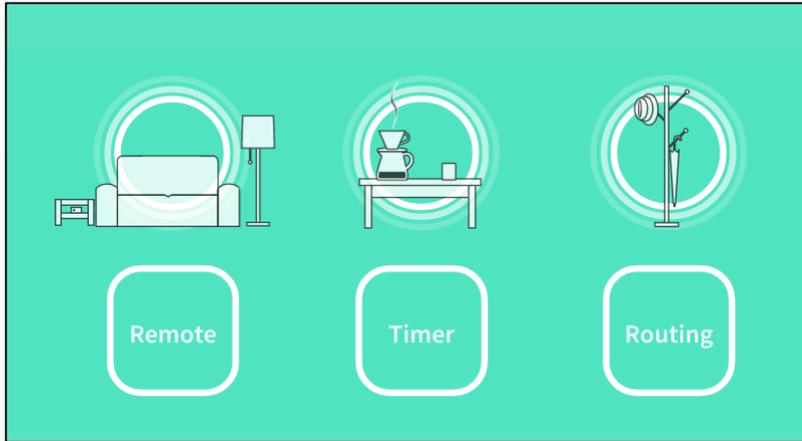


- Localização indoor;

- Cupons;

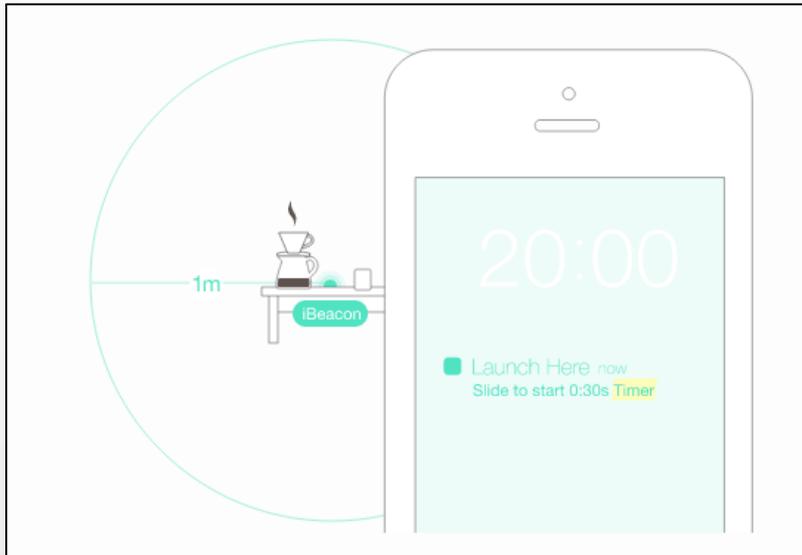
- Informações estatísticas.

Trabalhos Correlatos: Launch Here



- Marcação;

- Dia a dia;



- Ações customizadas.

Requisitos Funcionais

RF01: A ferramenta deverá exibir ao usuário uma página web vinculada ao beacon.

RF02: A ferramenta deverá enviar data e hora de acesso do beacon ao servidor Web.

RF03: A ferramenta deverá disponibilizar serviços para recepção de data e hora de acesso dos beacons.

RF04: A ferramenta deverá disponibilizar serviços de consulta a páginas editadas/estruturadas para cada beacon.

RF05: A ferramenta deverá conter uma autenticação de usuário.

RF06: A ferramenta deverá permitir manter cadastro de usuários.

RF07: A ferramenta deverá permitir manter o cadastro de beacons.

RF08: A ferramenta deverá manter o histórico de acesso dos beacons.

RF09: A ferramenta deverá permitir vincular o beacon a um lugar físico num mapa.

RF10: A ferramenta deverá permitir editar uma página HTML para exibição de informações e componentes web.

RF11: A ferramenta deverá permitir a pré-visualização da página estruturada.

Requisitos Não Funcionais

RNF01: A ferramenta deverá ser desenvolvida na linguagem de programação Java.

RNF02: A ferramenta deverá ser desenvolvida utilizando o *framework* Spring.

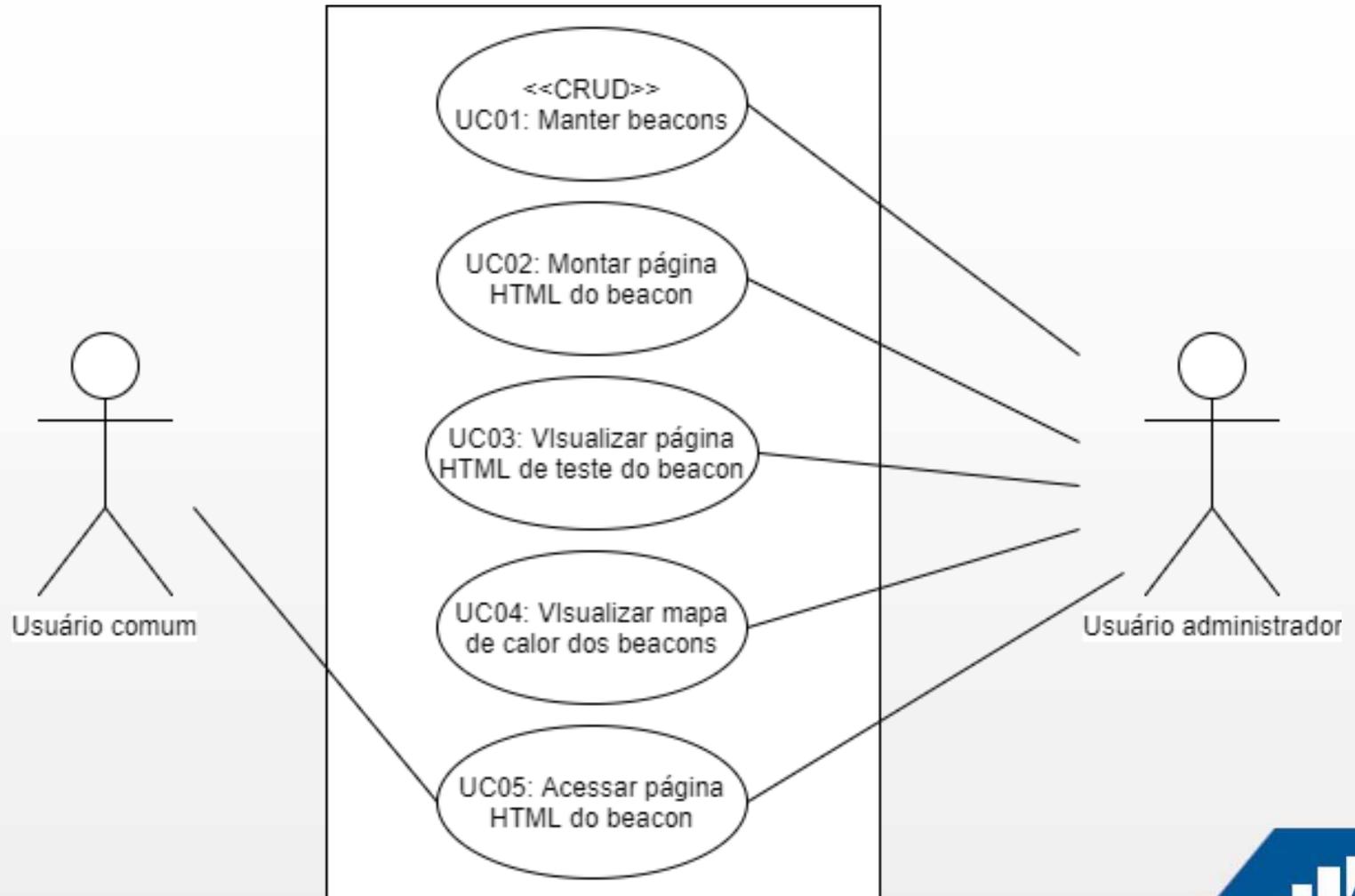
RNF03: A ferramenta deverá ser disponibilizada através de serviços no protocolo HTTP.

RNF04: A ferramenta deverá se comunicar com banco de dados PostgreSQL.

RNF05: A ferramenta deverá manter a estrutura das páginas HTML dinâmicas no banco de dados em formato JSON.

RNF06: A ferramenta deverá utilizar o *framework* AngularJS para interpretar o JSON e gerar páginas HTML.

Especificação: Casos de uso



Implementação

- Linguagem: Java;
- Ferramenta Front-end
 - Servidor: JSP;
 - Páginas: Angular JS;
- Ferramenta Back-end: Java Application;
- Banco de dados: PostgreSQL.

Implementação: Desenvolvimento da aplicação Front-end Servidor

```
54Ⓞ <form:form class="form-horizontal" method="post" modelAttribute="beaconForm" action="{beaconActionUrl}">
55
56     <form:hidden path="id" />
57
58Ⓞ     <spring:bind path="uid">
59Ⓞ         <div class="form-group ${status.error ? 'has-error' : ''}">
60             <label class="col-sm-2 control-label">ID</label>
61Ⓞ             <div class="col-sm-10">
62                 <form:input path="uid" type="text" class="form-control" id="uid" placeholder="uid" />
63                 <form:errors path="uid" class="control-label" />
64             </div>
65         </div>
66     </spring:bind>
67
68Ⓞ     <spring:bind path="urlid">
69Ⓞ         <div class="form-group ${status.error ? 'has-error' : ''}">
70             <label class="col-sm-2 control-label">URL</label>
71Ⓞ             <div class="col-sm-10">
72                 <form:input path="urlid" class="form-control" id="urlid" placeholder="urlid" />
73                 <form:errors path="urlid" class="control-label" />
74             </div>
75         </div>
76     </spring:bind>
77
78Ⓞ     <spring:bind path="estado">
79Ⓞ         <div class="form-group ${status.error ? 'has-error' : ''}">
80             <label class="col-sm-2 control-label">Estado</label>
81Ⓞ             <div class="col-sm-5">
82Ⓞ                 <form:select path="estado" class="form-control">
83                     <form:option value="NONE" label="--- Select ---" />
84                     <form:options items="{estadoList}" />
85                 </form:select>
86                 <form:errors path="estado" class="control-label" />
87             </div>
88             <div class="col-sm-5"></div>
89         </div>
90     </spring:bind>
```

Implementação: Desenvolvimento da aplicação Back-end Servidor

```
@RequestMapping(value = "/beacons", method = RequestMethod.POST)
public String saveOrUpdatebeacon(@ModelAttribute("beaconForm") @Validated Beacon beacon,
                                BindingResult result,
                                Model model,
                                final RedirectAttributes redirectAttributes) {
    if (result.hasErrors()) {
        populateDefaultModel(model);
        return "beacons/beaconform";
    } else {

        redirectAttributes.addFlashAttribute("css", "success");
        if (beacon.getId() == null) {
            redirectAttributes.addFlashAttribute("msg", "Beacon adicionado com sucesso!");
        } else {
            redirectAttributes.addFlashAttribute("msg", "Beacon atualizado com sucesso!");
        }

        beaconRepository.save(beacon);
        return "redirect:/beacons/" + beacon.getId();
    }
}
```

Implementação: Desenvolvimento da aplicação

Front-end Páginas

```
5  modelsNull = {
6      selected: null,
7      uid: "",
8      templates: [{
9          name: 'Rótulo',
10         type: "label",
11         id: 1,
12         fields: [{
13             description: "Texto",
14             value: ""
15         }]
16     },
17     {
18         name: 'Imagem',
19         type: "img",
20         fields: [{
21             description: "Caminho",
22             value: ""
23         }],
24         {
25             description: "Link",
26             value: ""
27         }
28     },
29     {
30         name: 'Container 1',
31         type: "container",
32         id: 4,
33         fields: [{
34             description: "Titulo",
35             value: ""
36         }],
37         columns: [
38             []
39         ]
40     },
41 }
```

Implementação: Desenvolvimento da aplicação Front-end Páginas

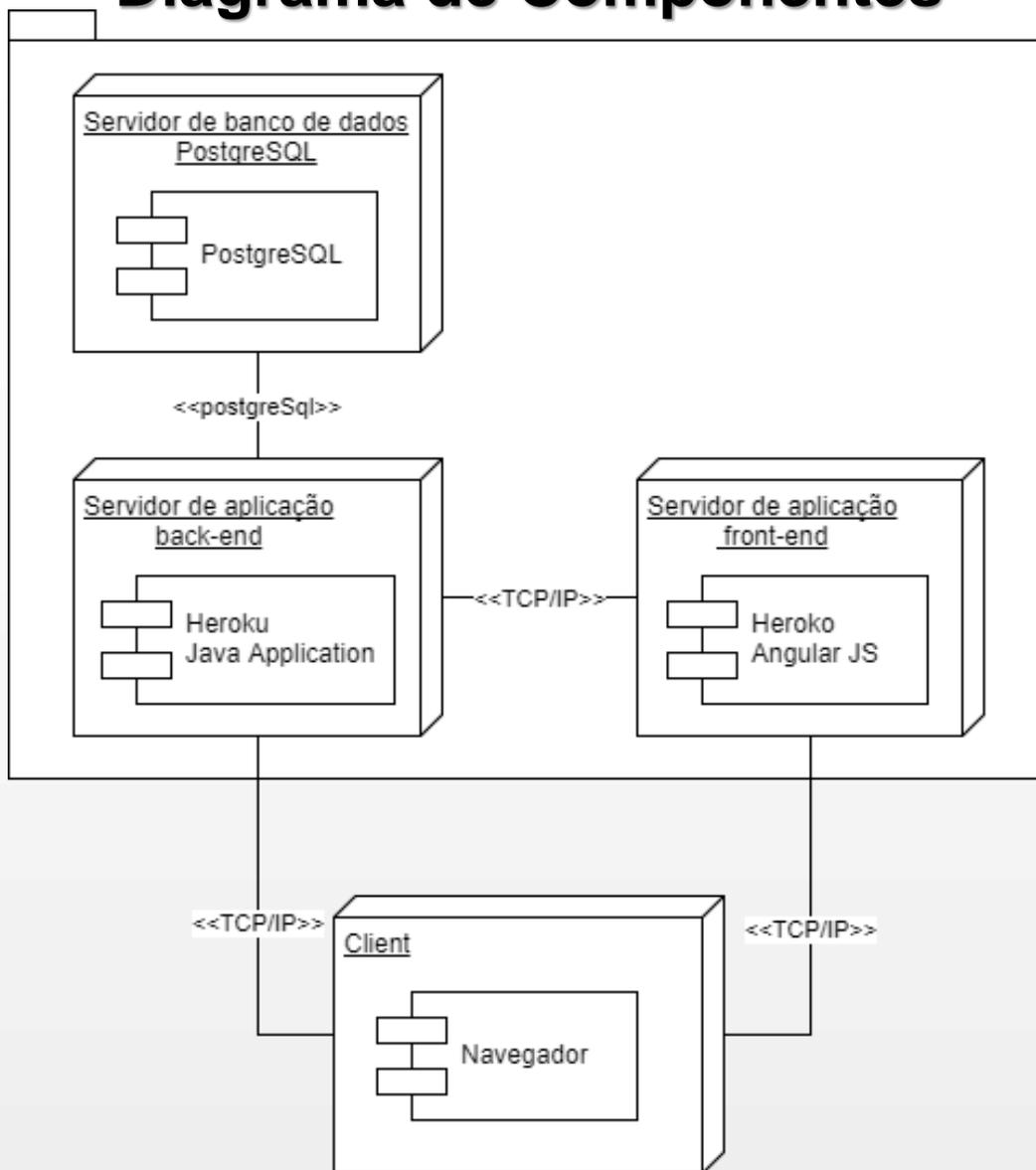
```
16 <script type="text/ng-template" id="container.html">
17   <div class="container box box-blue">
18     <h3 ng-hide="item.id == ''">{{item.id}}</h3>
19     <div class="column{{item.columns.length}}" ng-repeat="list in item.columns" ng-include="'list.html'"></div>
20     <div class="clearfix"></div>
21   </div>
22 </script>
23
24 <script type="text/ng-template" id="label.html">
25   <div class="item">{{item.fields[0].value}}</div>
26 </script>
27
28 <script type="text/ng-template" id="img.html">
29   <a href="{{item.fields[1].value}}">
30     
31   </a>
32 </script>
```

Implementação: Desenvolvimento da aplicação Front-end Páginas

```
62 <div class="col s7">
63   <div class="row">
64     <div ng-repeat="(zone, list) in models.dropzones" class="col s12">
65       <div class="dropzone box box-yellow editor">
66         <h3>Editor: {{zone}}</h3>
67         <div ng-include="'list.html'"></div>
68       </div>
69     </div>
70   </div>
71 </div>
```

Implementação: Desenvolvimento da aplicação

Diagrama de Componentes



Operacionalidade da Implementação

Cadastro

[Página Inicial](#) [Mapa de calor](#)

[Usuários](#) [Sair](#)

Beacons mapeados

UID	URLID	Estado	Ações				
1	https://page-maker.herokuapp.com/#/1	A	Visualizar	Atualizar	Deletar	Editar Página	Visualizar Página
2	https://page-maker.herokuapp.com/#/2	I	Visualizar	Atualizar	Deletar	Editar Página	Visualizar Página

[Novo Beacon](#)

Leonardo Adriano Reichert - FURB 2017/II

Operacionalidade da Implementação Cadastro

Página Inicial Mapa de calor Usuários Sair

Atualizar

ID

URL

Estado

X Y

The diagram shows a furniture layout on a grid. A large L-shaped sofa is labeled 'SOFA'. Two tables are labeled 'TABLE HW'. Two server racks are labeled 'SERVER'. A central display case is labeled 'SHOWCASE'. Two screens are labeled 'SCREEN'. The layout is positioned at X=152 and Y=255.

FURB
UNIVERSIDADE DE BLUMENAU

Operacionalidade da Implementação

Montagem de página

SALVAR

RÓTULO

IMAGEM

CONTAINER 1

CONTAINER 2

CONTAINER 3

Editor: Principal



Selecioneado

Tipo: Imagem - img

Arquivo

teste.jpg

Altura

500

Largura

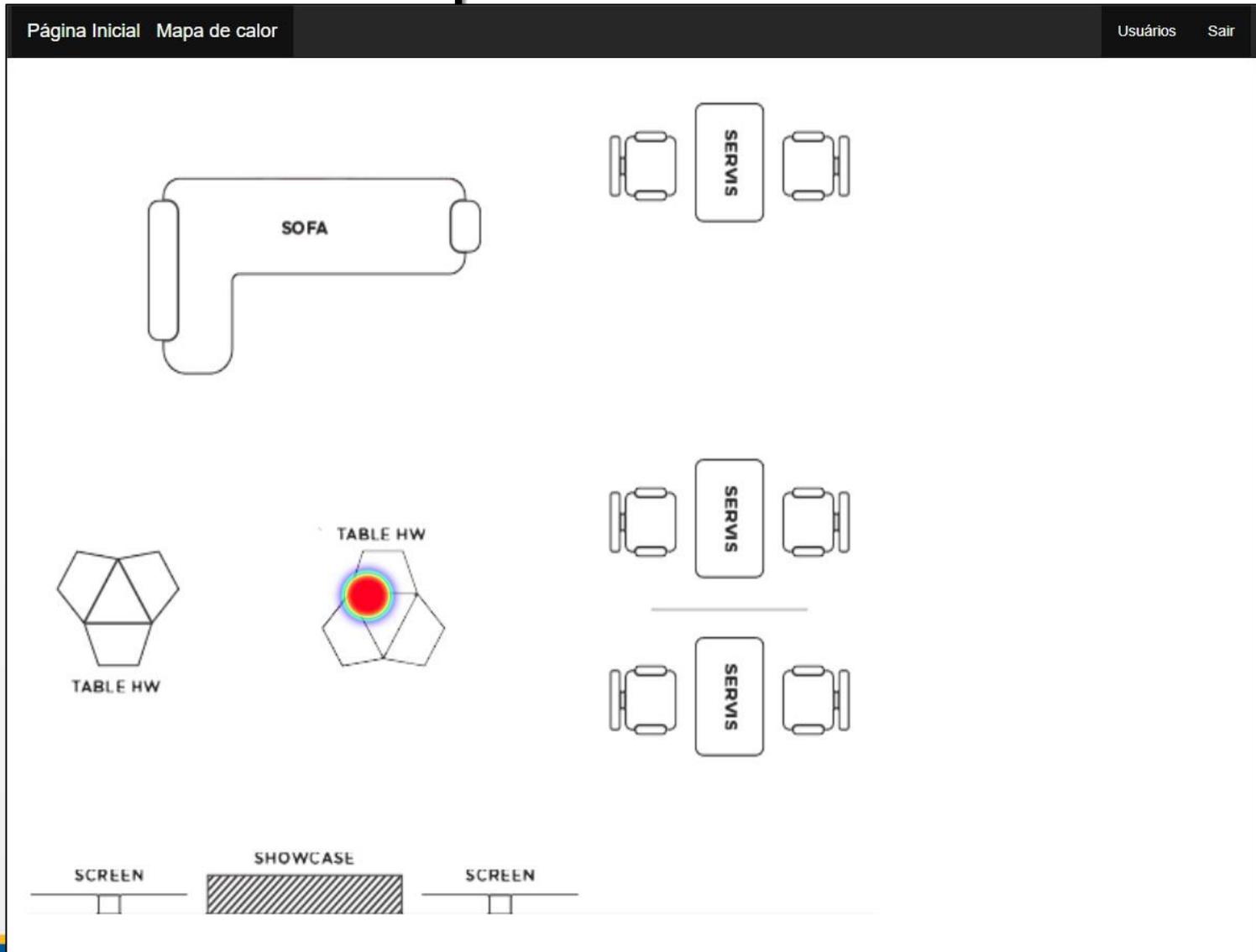
500

Link



Operacionalidade da Implementação

Mapa de calor



Resultados e Discussões

- Tempo de montagem e busca de template das páginas dinamicamente;

Tentativa	Tipo de Rede	Resposta requisição template (s)	Tempo Montagem HTML (s)
1	Móvel	0,824	1,459
2	Móvel	1,002	1,379
3	Móvel	0,832	1,414
4	Wifi	0.511	1,338
5	Wifi	0.436	1,479
6	Wifi	0.424	1,434

- Dispensa instalação de aplicativos no Android;
- Precisão do mapa de calor.

Resultados e Discussões

Características / Trabalhos correlatos	PayPal Beacon (2009)	Microlocation (2014)	Launch Here (2013)	Trabalho desenvolvido
Utiliza beacons como marcadores	X	X	X	X
Realiza ação após identificar beacon		X	X	
Permite customizar ação realizada			X	X
Geração de informações estatísticas	X	X		X
Armazena acessos e a localização do beacon		X		X
Não necessita de aplicativo instalado				X

Conclusões

- Ferramenta desenvolvida realiza mapeamento sem necessidade de instalação;
- Geração de página Dinamicamente
 - Facilidade e simplicidade na geração da página;
 - Efetividade na exibição.
- Expansível;

Sugestões

- a) permitir a possibilidade de cadastrar mais de um local;
- b) adicionar novos componentes web:
 - botão com evento de clique;
 - campo input para armazenamento de valor;
- c) tornar a montagem de página HTML parte da ferramenta principal.