



UNIVERSIDADE FEDERAL DO SUL E SUDESTE DO PARÁ - UNIFESSPA
FACULDADE DE COMPUTAÇÃO E ENGENHARIA ELÉTRICA - FACEEL

201940601001 - ALAIM DE JESUS LEÃO COSTA
201940601026 - HENRIQUE PEREIRA VIANA
201940601025 - MANOEL MALON COSTA DE MOURA

Trabalho Web Services (REST)

Marabá – PA
2022

Documentação API REST

1. Back-end

- **Model**

Classe das entidades do programa. Nessa classe é especificado todos os atributos do código, as anotações da API Rest e também os métodos de cada uma.

Classe Cliente - Na classe cliente observamos os imports que devem ser utilizados para implementação das anotações e métodos da API Rest, na linha 10 e 11 são declarados as anotações REST `@Entity` e `@Table`, essas anotações definem o mapeamento da classe Cliente na regra de negócio do programa e para posterior persistência no banco de dados. Nas linhas 14 a 35 são definidos os atributos da classe Cliente, cada atributo estão sendo anotados com suas respectivas anotações de estratégia de classe. No atributo Id, é feita as anotações `@Id`, `@GeneratedValue` e `@Column`, essas anotações irão fazer o mapeamento do atributo na classe do banco de dados automaticamente, a anotação `@GeneratedValue` é utilizada para informar que a geração do valor do identificador único da entidade será gerenciada pelo provedor de persistência. Essa anotação deve ser adicionada logo após a anotação `@Id`. Quando não anotamos o campo com essa opção, significa que a responsabilidade de gerar e gerenciar as chaves primárias será da aplicação, em outras palavras, do nosso código. Da linha 37 a 89 é feita a declaração dos métodos da nossa classe, esses métodos vão permitir o acesso aos atributos da nossa classe.

```

1 package br.com.viagem.projeto.model;
2
3 import javax.persistence.Column;
4 import javax.persistence.Entity;
5 import javax.persistence.GeneratedValue;
6 import javax.persistence.GenerationType;
7 import javax.persistence.Id;
8 import javax.persistence.Table;
9
10 @Entity
11 @Table(name = "usuario")
12 public class Cliente {
13
14     @Id
15     @GeneratedValue(strategy = GenerationType.IDENTITY)
16     @Column(name = "id")
17     private Integer id;
18
19     @Column(name = "nome_completo", length = 200, nullable = true)
20     private String nome_completo;
21
22     @Column(name = "email", length = 50, nullable = true)
23     private String email;
24
25     @Column(name = "senha", length = 15, nullable = true)
26     private String senha;
27
28     @Column(name = "telefone", length = 15, nullable = true)
29     private String telefone;
30
31     @Column(name = "cpf", length = 15, nullable = true)
32     private String cpf;
33
34     @Column(name = "idade")
35     private int idade;
36
37     public Integer getId() { return id; }
38
39     public void setId(Integer id) { this.id = id; }
40
41     public String getNome_completo() { return nome_completo; }
42
43     public void setNome_completo(String nome_completo) { this.nome_completo = nome_completo; }
44
45     public String getEmail() { return email; }
46
47     public void setEmail(String email) { this.email = email; }
48
49     public String getSenha() { return senha; }
50
51     public void setSenha(String senha) { this.senha = senha; }
52
53     public String getTelefone() { return telefone; }
54
55     public void setTelefone(String telefone) { this.telefone = telefone; }
56
57     public String getCpf() { return cpf; }
58
59     public void setCpf(String cpf) { this.cpf = cpf; }
60
61     public int getIdade() { return idade; }
62
63     public void setIdade(int idade) { this.idade = idade; }
64
65 }

```

Imagem (1). Cliente

Classe Hospedagem - Na classe de Hospedagem, na primeira linha é informado a hierarquia da classe na qual ela se encontra, das linhas 2 e 8 é declarado os imports que vão ser utilizados na classe, da linha 9 até a linha 33 é informado os atributos da classe hospedagem e suas respectivas anotações. Da linha 35 a 87 são feitas as declarações dos métodos Geters e Seteres, que serão utilizados nos acessos aos atributos da classe.

```
Hospedagem.java
1 package br.com.viagem.projeto.model;
2 import java.util.Date;
3 import javax.persistence.Column;
4 import javax.persistence.Entity;
5 import javax.persistence.GeneratedValue;
6 import javax.persistence.GenerationType;
7 import javax.persistence.Id;
8 import javax.persistence.Table;
9 @Entity
10 @Table(name = "hospedagem")
11 public class Hospedagem {
12     @Id
13     @GeneratedValue(strategy = GenerationType.IDENTITY)
14     @Column(name = "id")
15     private Integer id;
16
17     @Column(name = "nome_cidade", length = 100, nullable = true)
18     private String nome_cidade;
19
20     @Column(name = "data_entrada")
21     private java.sql.Date data_entrada;
22
23     @Column(name = "data_saida")
24     private java.sql.Date data_saida;
25
26     @Column(name = "numero_quartos")
27     private Integer numero_quartos;
28
29     @Column(name = "quantidade_pessoas")
30     private Integer quantidade_pessoas;
31
32     @Column(name = "valor")
33     private Integer valor;
34
35     public Integer getId() { return id; }
36
37     public void setId(Integer id) { this.id = id; }
38
39     public String getNome_cidade() { return nome_cidade; }
40
41     public void setNome_cidade(String nome_cidade) { this.nome_cidade = nome_cidade; }
42
43     public java.sql.Date getData_entrada() { return data_entrada; }
44
45     public void setData_entrada(java.sql.Date data_entrada) { this.data_entrada = data_entrada; }
46
47     public java.sql.Date getData_saida() { return data_saida; }
48
49     public void setData_saida(java.sql.Date data_saida) { this.data_saida = data_saida; }
50
51     public Integer getNumero_quartos() { return numero_quartos; }
52
53     public void setNumero_quartos(Integer numero_quartos) { this.numero_quartos = numero_quartos; }
54
55     public Integer getQuantidade_pessoas() { return quantidade_pessoas; }
56
57     public void setQuantidade_pessoas(Integer quantidade_pessoas) { this.quantidade_pessoas = quantidade_pessoas; }
58
59     public Integer getValor() { return valor; }
60
61     public void setValor(Integer valor) { this.valor = valor; }
62
63 }
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
```

Imagem (2). Hospedagem

Classe Passagem - Na respectiva classe, na primeira linha é informado o local que a classe se encontra na arquitetura do projeto. Das linhas 2 a 9 declaramos os imports que serão utilizados no decorrer da classe, da linha 10 até a linha 38, são declarados os atributos e as anotações REST da classe pessoa, da linha 34 até a linha 78, declaramos os métodos da nossa classe, esses métodos são parte estruturante da orientação a objetos da classe.

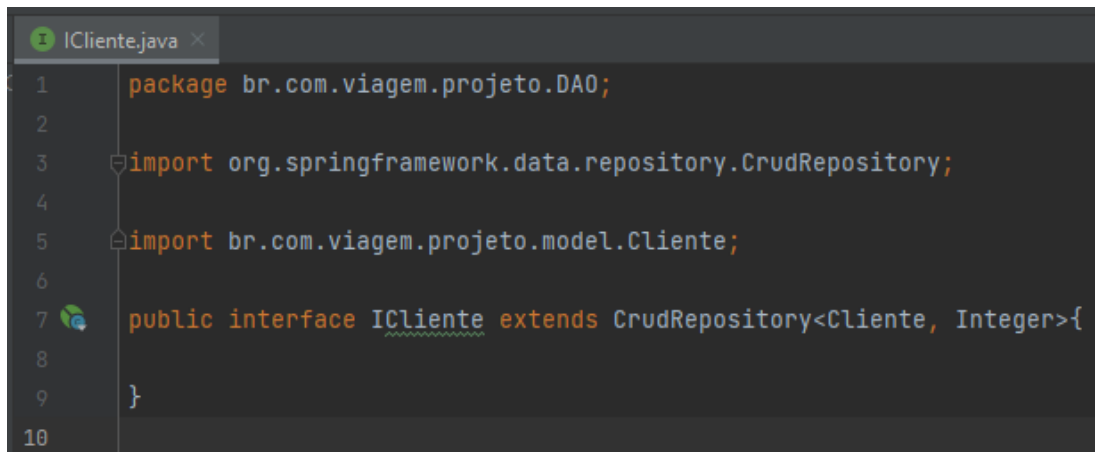
```
Passagem.java
1 package br.com.viagem.projeto.model;
2 import java.sql.Date;
3 import java.time.LocalDate;
4 import javax.persistence.Column;
5 import javax.persistence.Entity;
6 import javax.persistence.GeneratedValue;
7 import javax.persistence.GenerationType;
8 import javax.persistence.Id;
9 import javax.persistence.Table;
10 @Entity
11 @Table(name = "passagem")
12 public class Passagem {
13     |
14     @Id
15     @GeneratedValue(strategy = GenerationType.IDENTITY)
16     @Column(name = "id")
17     private Integer id;
18
19     @Column(name = "origem", length = 100, nullable = true)
20     private String origem;
21
22     @Column(name = "destino", length = 50, nullable = true)
23     private String destino;
24
25     @Column(name = "data_ida")
26     private java.sql.Date data_ida;
27
28     @Column(name = "data_volta")
29     private java.sql.Date data_volta;
30
31     @Column(name = "valor_passagem")
32     private Integer valor_passagem;
33
34     public Integer getId() { return id; }
35
36     public void setId(Integer id) { this.id = id; }
37
38     public String getOrigem() { return origem; }
39
40     public void setOrigem(String origem) { this.origem = origem; }
41
42     public String getDestino() { return destino; }
43
44     public void setDestino(String destino) { this.destino = destino; }
45
46     public java.sql.Date getData_ida() { return data_ida; }
47
48     public void setData_ida(java.sql.Date data_ida) { this.data_ida = data_ida; }
49
50     public java.sql.Date getData_volta() { return data_volta; }
51
52     public void setData_volta(java.sql.Date data_volta) { this.data_volta = data_volta; }
53
54     public Integer getValor_passagem() { return valor_passagem; }
55
56     public void setValor_passagem(Integer valor_passagem) { this.valor_passagem = valor_passagem; }
57
58 }
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82 }
```

Imagem (3). Passagem

- **DAO**

No pacote DAO, declaramos as interfaces que vão compor a estrutura do projeto, essas classes vão provê as interface que abstrai o acesso aos dados da nossa classe.

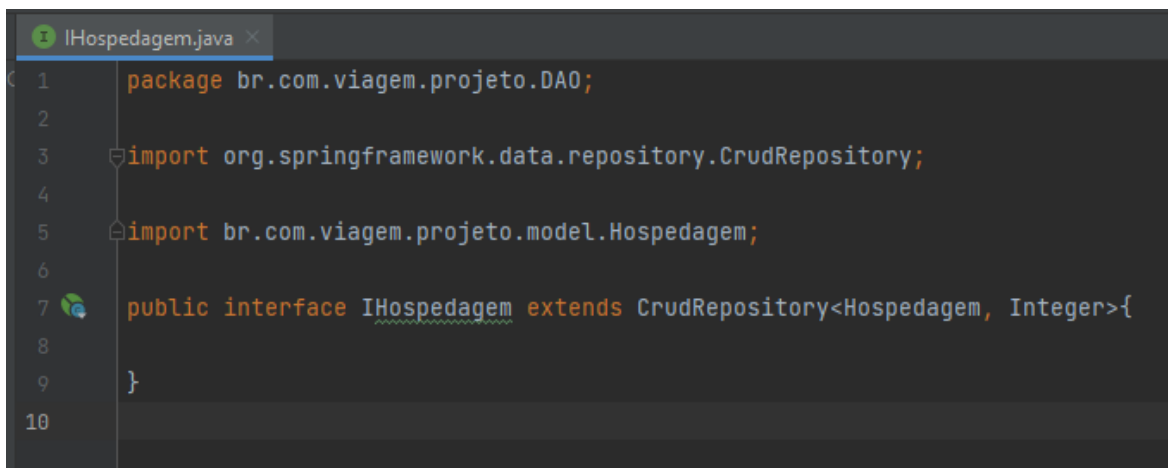
Interface ICliente - A interface ICliente vai abstrair o acesso dos dados da classe cliente.

A screenshot of an IDE showing the code for ICliente.java. The code is as follows:

```
1 package br.com.viagem.projeto.DAO;  
2  
3 import org.springframework.data.repository.CrudRepository;  
4  
5 import br.com.viagem.projeto.model.Cliente;  
6  
7 public interface ICliente extends CrudRepository<Cliente, Integer>{  
8  
9 }  
10
```

Imagem (4). ICliente.

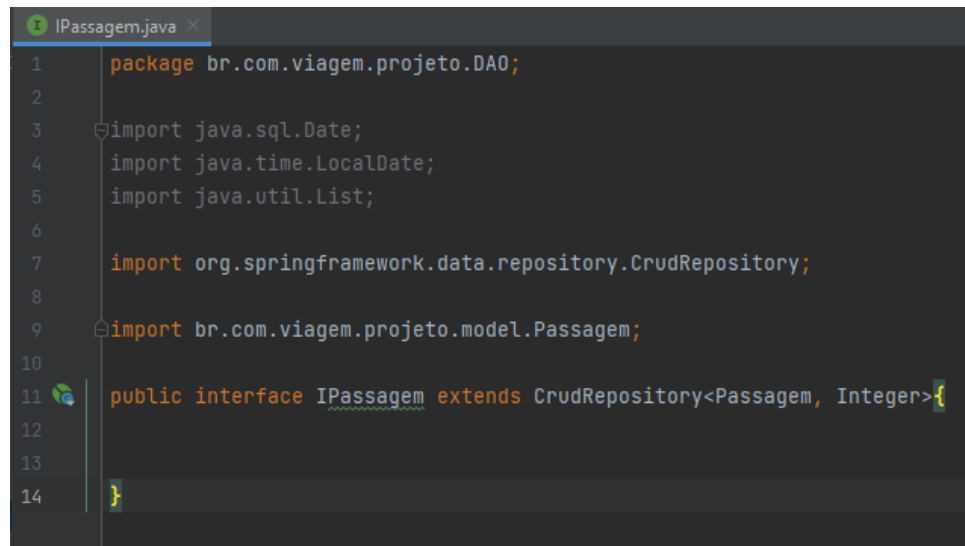
Interface IHospedagem - A interface IHospedagem vai abstrair o acesso dos dados da classe hospedagem.

A screenshot of an IDE showing the code for IHospedagem.java. The code is as follows:

```
1 package br.com.viagem.projeto.DAO;  
2  
3 import org.springframework.data.repository.CrudRepository;  
4  
5 import br.com.viagem.projeto.model.Hospedagem;  
6  
7 public interface IHospedagem extends CrudRepository<Hospedagem, Integer>{  
8  
9 }  
10
```

Imagem (5). IHospedagem.

Interface IPassagem - A interface IPassagem vai abstrair o acesso dos dados da classe passagem.



```
1 package br.com.viagem.projeto.DAO;
2
3 import java.sql.Date;
4 import java.time.LocalDate;
5 import java.util.List;
6
7 import org.springframework.data.repository.CrudRepository;
8
9 import br.com.viagem.projeto.model.Passagem;
10
11 public interface IPassagem extends CrudRepository<Passagem, Integer> {
12
13
14 }
```

Imagem (6). IPassagem.

- **Controller**

No pacote de controle são implementadas as classes dos responsáveis por intermediar as requisições enviadas pelo front com as respostas fornecidas pelo Model, processando os dados que o usuário informou e repassando para outras camadas.

Classe ClienteController - Na classe ClienteController, além de informar todos os imports que serão utilizados, também é necessário fazer a anotação da classe, essas anotações vão compor a regra de negócio do programa. O método HTTP é utilizado para determinar a operação a ser realizada em um determinado recurso do programa. Em geral, utiliza-se o GET para recuperar, POST para criar, PUT para alterar e DELETE para apagar.

As anotações `@RestController`, `@CrossOrigin` e `@RequestMapping`, irão mapear as requisições que serão feitas à classe cliente do programa. A anotação `@Autowired` permite a injeção automática de dependência fornecida pelo spring. Já a anotação `GetMapping` é para mapear GETsolicitações HTTP em métodos de manipulador específicos. Especificamente, `@GetMapping` é uma anotação composta que funciona como um atalho para arquivos `@RequestMapping(method = RequestMethod.GET)`.

```

1  package br.com.viagem.projeto.controller;
2  import org.springframework.beans.factory.annotation.Autowired;
3  import org.springframework.web.bind.annotation.CrossOrigin;
4  import org.springframework.web.bind.annotation.DeleteMapping;
5  import org.springframework.web.bind.annotation.GetMapping;
6  import org.springframework.web.bind.annotation.PathVariable;
7  import org.springframework.web.bind.annotation.PostMapping;
8  import org.springframework.web.bind.annotation.PutMapping;
9  import org.springframework.web.bind.annotation.RequestBody;
10 import org.springframework.web.bind.annotation.RequestMapping;
11 import org.springframework.web.bind.annotation.RestController;
12 import java.util.List;
13 import java.util.Optional;
14 import javax.websocket.server.PathParam;
15 import br.com.viagem.projeto.DAO.ICliente;
16 import br.com.viagem.projeto.model.Cliente;
17
18 @RestController
19 @CrossOrigin("*")
20 @RequestMapping("/clientes")
21 public class ClienteController {
22
23     @Autowired
24     private ICliente dao;
25
26     @GetMapping
27     public List<Cliente> listaClientes() { return (List<Cliente>) dao.findAll(); }
28
29     @PostMapping
30     public Cliente criarClientes(@RequestBody Cliente cliente) {
31         Cliente clienteNovo = dao.save(cliente);
32         return clienteNovo;
33     }
34
35     @PutMapping
36     public Cliente editarClientes(@RequestBody Cliente cliente) {
37         Cliente clienteNovo = dao.save(cliente);
38         return clienteNovo;
39     }
40
41     @DeleteMapping("/{id}")
42     public Optional<Cliente> excluirCliente(@PathVariable Integer id) {
43         Optional<Cliente> cliente = dao.findById(id);
44         dao.deleteById(id);
45         return cliente;
46     }
47 }

```

Imagem (7).Cliente Controller.

Classe HospedagemController - Na classe HospedagemController serão feitas as declarações da regra de negócio da classe hospedagem. Semelhante a classe anterior, a classe realiza as implementações semelhantes a classe anterior porém, implementando a regra de negócio da classe de hospedagem.

```
HospedagemController.java
1 package br.com.viagem.projeto.controller;
2 import org.springframework.beans.factory.annotation.Autowired;
3 import org.springframework.web.bind.annotation.CrossOrigin;
4 import org.springframework.web.bind.annotation.DeleteMapping;
5 import org.springframework.web.bind.annotation.GetMapping;
6 import org.springframework.web.bind.annotation.PathVariable;
7 import org.springframework.web.bind.annotation.PostMapping;
8 import org.springframework.web.bind.annotation.PutMapping;
9 import org.springframework.web.bind.annotation.RequestBody;
10 import org.springframework.web.bind.annotation.RequestMapping;
11 import org.springframework.web.bind.annotation.RequestParam;
12 import org.springframework.web.bind.annotation.RestController;
13 import java.util.ArrayList;
14 import java.util.List;
15 import java.util.Optional;
16 import javax.websocket.server.PathParam;
17 import br.com.viagem.projeto.DAO.IHospedagem;
18 import br.com.viagem.projeto.model.Hospedagem;
19 @RestController
20 @CrossOrigin("*")
21 @RequestMapping("/hospedagens")
22 public class HospedagemController {
23
24     @Autowired
25     private IHospedagem dao;
26
27     @GetMapping
28     public List<Hospedagem> listaPassagens(@RequestParam ("dataInicio") java.sql.Date dataInicio
29                                         @RequestParam ("dataFim") java.sql.Date dataFim){
30         List<Hospedagem> hospedagem = (List<Hospedagem>) dao.findAll();
31         List<Hospedagem> auxiliar = new ArrayList<>();
32
33         for(int i = 0; i < hospedagem.size(); i++) {
34             if((hospedagem.get(i).getData_entrada().getTime() >= dataInicio.getTime()) &&
35                 (hospedagem.get(i).getData_saida().getTime() <= dataFim.getTime())) {
36                 auxiliar.add(hospedagem.get(i));
37             }
38         }
39         return auxiliar;
40     }
41
42     @PostMapping
43     public Hospedagem criarHospedagem(@RequestBody Hospedagem hospedagem) {
44         Hospedagem hospedagemNova = dao.save(hospedagem);
45         return hospedagemNova;
46     }
47
48     @PutMapping
49     public Hospedagem editarHospedagem(@RequestBody Hospedagem hospedagem) {
50         Hospedagem hospedagemNova = dao.save(hospedagem);
51         return hospedagemNova;
52     }
53
54     @DeleteMapping("/{id}")
55     public Optional<Hospedagem> excluirHospedagem(@PathVariable Integer id) {
56         Optional<Hospedagem> hospedagem = dao.findById(id);
57         dao.deleteById(id);
58         return hospedagem;
59     }
60 }
```

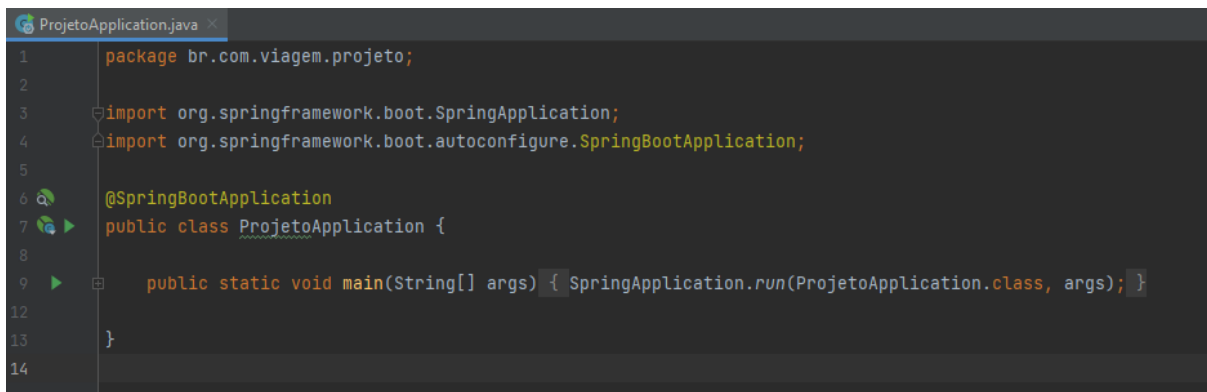
Imagem (8).Hospedagem Controller.

Classe Passagem Controller - Na classe PassagemController serão feitas as declarações da regra de negócio da classe passagem. Semelhante a classe controller do cliente, a classe realiza as implementações semelhantes a classe anterior porém, implementando a regra de negócio da classe de passagem, as funções das anotações também se aplicam na classe PassagemController.

```
PassagemController.java
1 package br.com.viagem.projeto.controller;
2 import org.apache.catalina.connector.Response;
3 import org.springframework.beans.factory.annotation.Autowired;
4 import org.springframework.http.ResponseEntity;
5 import org.springframework.web.bind.annotation.CrossOrigin;
6 import org.springframework.web.bind.annotation.DeleteMapping;
7 import org.springframework.web.bind.annotation.GetMapping;
8 import org.springframework.web.bind.annotation.PathVariable;
9 import org.springframework.web.bind.annotation.PostMapping;
10 import org.springframework.web.bind.annotation.PutMapping;
11 import org.springframework.web.bind.annotation.RequestBody;
12 import org.springframework.web.bind.annotation.RequestMapping;
13 import org.springframework.web.bind.annotation.RequestParam;
14 import org.springframework.web.bind.annotation.RestController;
15 import java.time.LocalDate;
16 import java.time.format.DateTimeFormatter;
17 import java.util.ArrayList;
18 import java.util.List;
19 import java.util.Optional;
20 import java.sql.Date;
21 import javax.websocket.server.PathParam;
22 import br.com.viagem.projeto.DAO.IPassagem;
23 import br.com.viagem.projeto.model.Passagem;
24 @RestController
25 @CrossOrigin("*")
26 @RequestMapping("/passagens")
27 public class PassagemController {
28     @Autowired
29     private IPassagem dao;
30     private LocalDate date = LocalDate.now();
31     @GetMapping
32     public List<Passagem> listaPassagens(@RequestParam("dataInicio") java.sql.Date dataInicio,
33                                         @RequestParam("dataFim") java.sql.Date dataFim){
34         List<Passagem> passagem = (List<Passagem>) dao.findAll();
35         List<Passagem> auxiliar = new ArrayList<>();
36
37         for(int i = 0; i < passagem.size(); i++) {
38             if((passagem.get(i).getData_ida().getTime() >= dataInicio.getTime()) &&
39                (passagem.get(i).getData_volta().getTime() <= dataFim.getTime())) {
40                 auxiliar.add(passagem.get(i));
41             }
42         }
43         return auxiliar;
44     }
45     @PostMapping
46     public Passagem criarPassagem(@RequestBody Passagem passagem) {
47         Passagem passagemNova = dao.save(passagem);
48         return passagemNova;
49     }
50     @PutMapping
51     public Passagem editarPassagem(@RequestBody Passagem passagem) {
52         Passagem passagemNova = dao.save(passagem);
53         return passagemNova;
54     }
55     @DeleteMapping("/{id}")
56     public Optional<Passagem> excluirPassagem(@PathVariable Integer id) {
57         Optional<Passagem> passagem = dao.findById(id);
58         dao.deleteById(id);
59         return passagem;
60     }
61 }
62
```

Imagem (9).Passagem Controller.

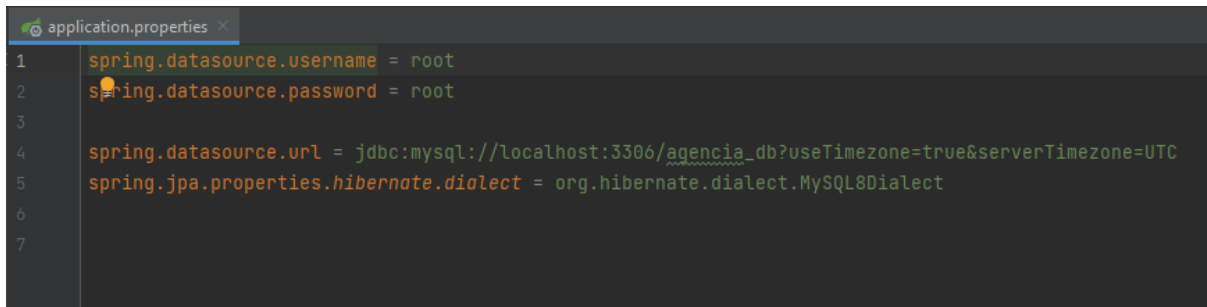
Aplicação do Projeto: Nesta classe teremos a implementação inicial do projeto, será a classe que vai executar o código.



```
1 package br.com.viagem.projeto;
2
3 import org.springframework.boot.SpringApplication;
4 import org.springframework.boot.autoconfigure.SpringBootApplication;
5
6 @SpringBootApplication
7 public class ProjetoApplication {
8
9     public static void main(String[] args) { SpringApplication.run(ProjetoApplication.class, args); }
10
11 }
12
13
14
```

Imagem (10).Projeto Application.

Propriedades do aplicativo:



```
1 spring.datasource.username = root
2 spring.datasource.password = root
3
4 spring.datasource.url = jdbc:mysql://localhost:3306/agencia_db?useTimezone=true&serverTimezone=UTC
5 spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.MySQL8Dialect
6
7
```

Imagem (11).Application properties.

Front-end : no front foi criada a interface gráfica para que o usuário possa interagir com o programa onde ele irá fazer suas consultas das passagens aéreas, onde ele possa comprar também tanto passagens de “ida” como de “volta” e fazer outras coisas como ver o destino da viagem, a Data, a quantidade de pessoas e suas idades mostrar as opções de pagamento.

Index: o index vai ser responsável por organizar os dados e mostrar os resultados de uma maneira mais eficiente, vai ser uma estrutura da interface onde os clientes possam acessar como uma matriz vai ser uma forma para aumentar o desempenho na execução de consultas onde o cliente possa acessar suas reservas, suas datas de check-in e check-out, as hospedagens entre muitas outras coisas.

```

1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5    <meta charset="UTF-8" />
6    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
7    <meta http-equiv="X-UA-Compatible" content="ie=edge" />
8    <script src="/script.js" defer></script>
9    <link rel="stylesheet" href="css/style.css" />
10   <title>Document</title>
11 </head>
12
13 <body>
14   <header>
15     <div class="container content">
16       
17       <p>
18         Reservations: 061/505-775 <br />
19         Info: 032/123-456
20       </p>
21       <p>
22         Reservas: alaim@unifesspa.com.br <br />
23         Info: palladium@gmail.com
24       </p>
25     </div>
26   </header>
27   <nav>
28     <div class="container">
29       <ul>
30         <li><a href="#welcome">Hospedagens</a></li>
31         <li><a href="#voos.html">Voos</a></li>
32         <li><a href="#staff">Equipe</a></li>
33         <!--<li><a href="#pricing">Pricing</a></li>-->
34         <li><a href="#gallery">Galeria</a></li>
35         <li><a href="#contact">Contato </a></li>
36       </ul>
37     </div>
38   </nav>
39
40   <main class="showcase">
41     <video width="100%" autoplay muted loop>
42       <source src="img/video.mp4" type="video/mp4" />
43     </video>
44
45     <form method="post">
46       <div class="forma" id="corpo">
47         Check in date: <br />
48         <input type="date" name="date" id="dateIni" />
49         Check out date: <br />
50         <input type="date" name="date" id="dateFim" />
51         <input type="button" class="button" id="receita1" value="Verificar Disponibilidade!" />
52       </div>

```

Imagem (12). Index parte 1

```

53     </form>
54 </main>
55
56 <style type="text/css">
57     .escondida {
58         display: none;
59     }
60 </style>
61
62 <script type="text/javascript">
63     function abrir() {
64         var main = document.getElementById("corpo");
65         var iten = document.querySelector("#receita1");
66
67         const button = document.querySelector(".button")
68
69         //var teste = document.querySelector("#botao")
70         //console.log(teste);
71
72         //var iten = teste;
73         //console.log(iten);
74
75         /*
76         if (iten) {
77
78             iten.onclick = function () {
79                 var el = document.getElementById(this.id.substr(7, 7));
80                 console.log(el);
81                 if (el.style.display == "block")
82                     el.style.display = "none";
83                 else
84                     el.style.display = "block";
85             }
86         }
87         */
88
89         button.addEventListener("click", function () {
90             event.preventDefault();
91             var el = document.getElementById(this.id.substr(7, 7));
92             console.log(el);
93             console.log(this.id);
94             if (el.style.display == "block")
95                 el.style.display = "none";
96             else
97                 el.style.display = "block";
98         });
99         /*
100         if (iten) {
101             for (var i = 0; i < iten.length; i++) {
102                 iten[i].onclick = function () {
103                     var el = document.getElementById(this.id.substr(7, 7));

```

Imagem (13). Index parte 2

```

104         if (el.style.display == "block")
105             el.style.display = "none";
106         else
107             el.style.display = "block";
108     }
109 }
110 }*/
111
112 }
113 window.onload = abrir;
114 </script>
115
116 <div id="central">
117     <!-- Botao -->
118     <!--<input type="button" value="Abrir / fechar" id="receita2">-->
119     <!-- conteudo escondido -->
120     <div id="1" class="escondida">
121         <!-- Aqui fica o seu form -->
122
123         <!-- INICIO DO FORMULÁRIO -->
124
125         <head>
126             <meta charset="UTF-8">
127             <meta name="viewport" content="width=device-width, initial-scale=1.0">
128             <title>Selecionar multiplos itens no elemento select</title>
129         </head>
130
131         <body>
132             <label for="ddlCores">Selecione sua Hospedagem:</label><br>
133             <select id="ddlCores" multiple="multiple" size="10" style="width:600px">
134
135
136
137         </select> <br>
138         <div class="box2">
139             <h2>Seus Dados</h2>
140             <form action="dadosCliente">
141                 <input type="text" class="nome" name="nome" placeholder="Nome">
142                 <input type="text" class="email" name="email" placeholder="Email">
143                 <input type="text" class="cpf" name="cpf" placeholder="CPF">
144                 <input type="text" class="tel" name="tel" placeholder="Telefone">
145                 <input type="number" class="idade" name="idade" placeholder="Idade">
146                 <br>
147                 <input type="number" class="quantPessoas" name="quantPessoas" placeholder="Quantidade de Pessoas">
148                 <input type="number" class="numQuartos" name="numQuartos" placeholder="Nº de Quartos">
149                 <div class="modoPagamento">
150                     <p>
151                         Cartão de Débito: <input type="radio" name="citizenship" checked="yes" /><br />
152                         Cartão de Crédito: <input type="radio" name="citizenship" /><br />
153                         Boleto: <input type="radio" name="citizenship" />
154                     </p>

```

Imagem (14). Index parte 3

```

155     </div>
156     <div class="tipodeViagem">
157         <p>
158             Somente ida: <input type="radio" name="tipo" checked="yes" /><br />
159             Ida e volta: <input type="radio" name="tipo" /><br />
160         </p>
161     </div>
162     <button id="botaoCadastrar">Prosseguir</button>
163 </form>
164 </div>
165 <!--<input type="button" id="btnExibir" value="Exibir itens selecionados" />-->
166 <h2 id="resultado"></h2>
167 <script>
168     document.getElementById("btnExibir").onclick = function () {
169         let ddl = document.getElementById('ddlCores');
170         let itensSelecionados = Array.from(ddl.options)
171             .filter(option => option.selected)
172             .map(option => option.value);
173         document.getElementById("resultado").innerHTML = "Você selecionou: " + itensSelecionados;
174     };
175 </script>
176 </body>
177 </div>
178 </div>
179 <main class="prelude" id="Bem-vindo">
180     <h1>Bem-vindo</h1>
181 </main>
182 <main class="glavniMain">
183     
184     <div class="text1">
185         <h1>Quartos incríveis</h1>
186         <p>
187             Para todos os tipos de viagem, encontre ofertas únicas para aproveitar o seu tempo de descanso.
188         </p>
189     </div>
190
191     
192     <div class="text2">
193         <h1>A maior</h1>
194         <p>
195             A maior Companhia aérea do país e a número 1 em pontualidade, conforto, atendimento
196             e número de passageiros transportados no Brasil. Tudo isso graças à dedicação do nosso
197             time de colaboradores, que trabalha para que possamos oferecer sempre a melhor experiência e
198             cumprir o nosso Propósito de Ser a Primeira para Todos.
199         </p>
200     </div>
201
202     
203     <div class="text3">
204         <h1>Vista Magnífica</h1>
205         <p>

```

Imagem (15). Index parte 4

```

205     <p>
206         Com uma ampla gama de serviços para famílias, casais e grupos, além de serviços
207         especialmente pensados <0x200b><0x200b>para eventos, reuniões de negócios, viagens de incentivo
208         e casamentos. Os resorts - todos localizados à beira-mar - oferecem instalações
209         luxuosas para adultos e crianças, com atendimento personalizado garantido. Há uma
210         variedade de opções gastronômicas, uma infinidade de opções de passeios de um dia
211         e áreas de recreação para os mais pequenos... essa enorme variedade de opções está
212         disponível em nosso pacote Infinite Indulgence com tudo incluído. E com nosso serviço
213         premium para famílias, em uma excelente localização dentro do resort - todos os
214         membros da família podem desfrutar de toques de luxo e serviços VIP,
215         para que ninguém perca sua estadia.
216     </p>
217 </div>
218 </main>
219
220 <main class="prelude" id="gallery">
221     <H1>Galeria</H1>
222 </main>
223
224 <main class="Gallery">
225     
226     
227     
228     
229     
230     
231     
232     
233     
234     
235     
236     
237 </main>
238 <!--
239 <main class="prelude" id="pricing">
240     <H1>Pricing</H1>
241 </main>
242 <main class="Pricing">
243     <div class="columns">
244         <ul class="price">
245             <li class="header">Standard</li>
246             <li class="grey">$300/ Night</li>
247             <li>Casino</li>
248             <li>Exclusive Lounge</li>
249             <li>Private Concerts</li>
250             <li>Special Kitchen</li>
251             <li class="grey"><a href="#" class="button">Reserve</a></li>
252         </ul>
253     </div>
254

```

Imagem (16). Index parte 5


```

280 <main class="prelude" id="staff">
281   <H1>Nossa equipe</H1>
282 </main>
283
284 <main class="cards">
285   <div class="flip-card">
286     <div class="flip-card-inner">
287       <div class="flip-card-front">
288         
289       </div>
290       <div class="flip-card-back">
291         <h1>Alaim de Jesus</h1>
292         <p>Analista de Sistema - CTIC</p>
293         <p>O Melhor! kkk</p>
294       </div>
295     </div>
296   </div>
297
298   <div class="flip-card">
299     <div class="flip-card-inner">
300       <div class="flip-card-front">
301         
302       </div>
303       <div class="flip-card-back">
304         <h1>Henrique Viana</h1>
305         <p>Suporte - CTIC</p>
306         <p>O Maior!</p>
307       </div>
308     </div>
309   </div>
310
311   <div class="flip-card">
312     <div class="flip-card-inner">
313       <div class="flip-card-front">
314         
315       </div>
316       <div class="flip-card-back">
317         <h1>Manoel Malon</h1>
318         <p>Estudante</p>
319         <p>O Melhorzinho que ta tendo !kk</p>
320       </div>
321     </div>
322   </div>
323 </main>
324
325 <main class="prelude" id="contact">
326   <H1>Entre em contato conosco</H1>
327 </main>
328
329 <main class="AboutUs">
330   <div class="wrapper">

```

Imagem (17). Index parte 6

```

339         <textarea placeholder="Mensagem"></textarea>
340         <div class="btn">Enviar</div>
341     </div>
342 </div>
343 </div>
344 </main>
345 <footer>
346     <div class="container">
347         <table>
348             <tr>
349                 <td class="prvi">Ofertas</td>
350                 <td class="prvi">VIP</td>
351                 <td class="prvi">Vantagens</td>
352                 <td class="prvi">Quartos</td>
353             </tr>
354
355             <tr>
356                 <td>Pacote Padrão</td>
357                 <td>Plano privado</td>
358                 <td>Piscina</td>
359                 <td>Laptop</td>
360             </tr>
361
362             <tr>
363                 <td>Pacote de luxo</td>
364                 <td>Cabana na floresta</td>
365                 <td>Boate</td>
366                 <td>TV a satélite</td>
367             </tr>
368
369             <tr>
370                 <td>Pacote Premium</td>
371                 <td>Casino</td>
372                 <td>Tour pela cidade</td>
373                 <td>Wi-fi grátis</td>
374             </tr>
375
376             <tr>
377                 <td>Presidente</td>
378                 <td>Lounge Exclusivo</td>
379                 <td>Open bar</td>
380                 <td>Ar condicionado</td>
381             </tr>
382         </table>
383     </div>
384 </footer>
385 </body>
386
387 </html>

```

Imagem (18). Index parte 7

```

1  const button = document.querySelector(".button")
2  var origem1 = document.querySelector("#origem");
3  var dataAtual = new Date(); // retornar a data atual (Date)
4  const forma = document.querySelector(".forma")
5  const IdataInicio = document.querySelector('#dateIni')
6  IdataInicio.onchange = (e) => {
7      console.Log(e.target.value)
8  }
9
10 const IdataFim = document.querySelector('#dateFim')
11 IdataFim.onchange = (e) => {
12     console.Log(e.target.value)
13 }
14
15 const showData = (result) => {
16     console.Log(result);
17
18     for (const campo in result) {
19         console.Log(campo)
20     }
21 }
22
23 function verificarDisponibilidade() {
24
25     forma.addEventListener("submit", function (event) {
26         event.preventDefault();
27
28         let dados = {
29             IdataInicio,
30             IdataFim
31         };
32         alert(dados);
33     })
34
35     fetch("http://localhost:8080/hospedagens/?dataInicio=" + IdataInicio.value + "&dataFim=" + IdataFim.value)
36     .then(response => response.json())
37     .then(data => {
38         data.map((item) => {
39             //const li = document.createElement("li");
40             //li.setAttribute('id', item.valor_passagem);
41             //console.log(item.valor_passagem);
42             //origem1.innerHTML = origem1.value = item.origem;
43
44             var cont = 0;
45             for (const campo in item) {
46                 cont++;
47             }
48
49             for (let index = 0; index < 1; index++) {
50                 var divEl = document.createElement("option")
51                 var textEl = document.createTextNode(item.nome_cidade+" "+
52                 item.data_entrada+" "+item.data_saida+" "+item.valor);
53                 divEl.appendChild(textEl)
54
55                 document.getElementById("ddlCores").appendChild(divEl)
56             }
57         })
58     })
59 };
60
61 function mostrarResposta(dados) {
62     console.Log(dados);
63     const mensagem = `
64     Cidade: ${dados.origem}
65     `;
66     console.Log(Cidade);
67     alert(mensagem);
68 }
69 button.addEventListener("click", function () {
70     event.preventDefault();
71
72     verificarDisponibilidade();
73 });

```

Imagem(19). script.

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Selecionar multiplos itens no elemento select</title>
7  </head>
8  <body>
9      <label for="ddlCores">Selecione uma ou mais cores:</label><br>
10     <select id="ddlCores" multiple="multiple" size="10" style="width:100px">
11         <option value="Azul">Azul</option>
12         <option value="Amarelo">Amarelo</option>
13         <option value="Verde">Verde</option>
14         <option value="Rosa">Rosa</option>
15         <option value="Branco">Branco</option>
16     </select> <br>
17     <input type="button" id="btnExibir" value="Exibir itens selecionados" />
18     <h2 id="resultado"></h2>
19     <script>
20         document.getElementById("btnExibir").onclick = function () {
21             let ddl = document.getElementById('ddlCores');
22             let itensSelecionados = Array.from(ddl.options)
23                 .filter(option => option.selected)
24                 .map(option => option.value);
25             document.getElementById("resultado").innerHTML = "Você selecionou: " + itensSelecionados;
26         };
27     </script>
28 </body>

```

Imagem(20). teste.

```

1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8" />
5     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6     <meta http-equiv="X-UA-Compatible" content="ie=edge" />
7     <script src="./scriptVoo.js" defer></script>
8     <link rel="stylesheet" href="css/styleVoo.css" />
9     <title>Document</title>
10  </head>
11  <body>
12    <header>
13      <div class="container content">
14        
15        <p>
16          Reservations: 061/505-775 <br />
17          Info: 032/123-456
18        </p>
19        <p>
20          Reservas: alaim@unifesspa.com.br <br />
21          Info: palladium@gmail.com
22        </p>
23      </div>
24    </header>
25    <nav>
26      <div class="container">
27        <ul>
28          <li><a href="index.html">Hospedagens</a></li>
29          <li><a href="#">Voos</a></li>
30          <li><a href="index.html#staff">Equipe</a></li>
31          <!--<li><a href="#pricing">Pricing</a></li>-->
32          <li><a href="index.html#gallery">Galeria</a></li>
33          <li><a href="index.html#contact">Contato </a></li>
34        </ul>
35      </div>
36    </nav>
37
38    <main class="showcase">
39      <!--<video width="100%" autoplay muted loop>
40        <source src="img/video.mp4" type="video/mp4" />
41      </video>
42      -->
43      
44
45      <form method="post">
46        <div class="forma">
47          Check in date: <br />
48          <input type="date" name="date" class="dataIn" id="dateIni" />
49          Check out date: <br />
50          <input type="date" name="date" class="dataFim" id="dateFim" />
51          <input type="button" class="button" id="receita1" value="Verificar Disponibilidade!" />
52

```

Imagem(21). voo parte 1.

```

53     </div>
54 </form>
55 </main>
56
57
58 <style type="text/css">
59     .escondida {
60         display: none;
61     }
62 </style>
63
64 <script type="text/javascript">
65     function abrir() {
66         var main = document.getElementById("corpo");
67         var iten = document.querySelector("#receita1");
68
69         const button = document.querySelector(".button")
70
71         //var teste = document.querySelector("#botao")
72         //console.log(teste);
73
74         //var iten = teste;
75         //console.log(iten);
76
77         /*
78         if (iten) {
79
80             iten.onclick = function () {
81                 var el = document.getElementById(this.id.substr(7, 7));
82                 console.log(el);
83                 if (el.style.display == "block")
84                     el.style.display = "none";
85                 else
86                     el.style.display = "block";
87             }
88         }
89         */
90
91         button.addEventListener("click", function () {
92             event.preventDefault();
93             var el = document.getElementById(this.id.substr(7, 7));
94             console.log(el);
95             console.log(this.id);
96             if (el.style.display == "block")
97                 el.style.display = "none";
98             else
99                 el.style.display = "block";
100         });

```

Imagem(22). voo parte 2

```

115     window.onload = abrir;
116 </script>
117
118 <div id="central">
119     <!-- Botao -->
120     <!--<input type="button" value="Abrir / fechar" id="receita2">-->
121     <!-- conteudo escondido -->
122     <div id="1" class="escondida">
123         <!-- Aqui fica o seu form -->
124
125         <!-- INICIO DO FORMULÁRIO -->
126
127         <head>
128             <meta charset="UTF-8">
129             <meta name="viewport" content="width=device-width, initial-scale=1.0">
130             <title>Selecionar multiplos itens no elemento select</title>
131         </head>
132
133         <body>
134             <label for="ddlCores">Selecione sua Hospedagem:</label><br>
135             <select id="ddlCores" multiple="multiple" size="10" style="width:600px">
136
137             </select> <br>
138             <div class="box2">
139                 <h2>Seus Dados</h2>
140                 <form action="dadosCliente">
141                     <input type="text" class="nome" name="nome" placeholder="Nome">
142                     <input type="text" class="email" name="email" placeholder="Email">
143                     <input type="text" class="cpf" name="cpf" placeholder="CPF">
144                     <input type="text" class="tel" name="tel" placeholder="Telefone">
145                     <input type="number" class="idade" name="idade" placeholder="Idade">
146                     <br>
147                     <input type="number" class="quantPessoas" name="quantPessoas" placeholder="Quantidade de Pessoas">
148                     <input type="number" class="numQuartos" name="numQuartos" placeholder="Nº de Quartos">
149                     <div class="modoPagamento">
150                         <p>
151                             Cartão de Débito: <input type="radio" name="citizenship" checked="yes" /><br />
152                             Cartão de Crédito: <input type="radio" name="citizenship" /><br />
153                             Boleto: <input type="radio" name="citizenship" />
154                         </p>
155                     </div>
156                     <div class="tipodeViagem">
157                         <p>
158                             Somente ida: <input type="radio" name="tipo" checked="yes" /><br />
159                             Ida e volta: <input type="radio" name="tipo" /><br />
160                         </p>
161                     </div>
162                     <button id="botaoCadastrar">Prosseguir</button>
163                 </form>
164             </div>
165             <!--<input type="button" id="btnExibir" value="Exibir itens selecionados" />-->
166             <h2 id="resultado"></h2>

```

Imagem(23). voo parte 3

```

172         .map(option => option.value);
173         document.getElementById("resultado").innerHTML = "Você selecionou: " + itensSelecionados;
174     });
175 </script>
176 </body>
177 </div>
178 </div>
179
180 <footer>
181     <div class="container">
182         <table>
183             <tr>
184                 <td class="prvi">0fertas</td>
185                 <td class="prvi">VIP</td>
186                 <td class="prvi">Vantagens</td>
187                 <td class="prvi">Quartos</td>
188             </tr>
189
190             <tr>
191                 <td>Pacote Padrão</td>
192                 <td>Plano privado</td>
193                 <td>Piscina</td>
194                 <td>Laptop</td>
195             </tr>
196
197             <tr>
198                 <td>Pacote de luxo</td>
199                 <td>Cabana na floresta</td>
200                 <td>Boate</td>
201                 <td>TV a satélite</td>
202             </tr>
203
204             <tr>
205                 <td>Pacote Premium</td>
206                 <td>Casino</td>
207                 <td>Tour pela cidade</td>
208                 <td>Wi-fi grátis</td>
209             </tr>
210
211             <tr>
212                 <td>Presidente</td>
213                 <td>Lounge Exclusivo</td>
214                 <td>Open bar</td>
215                 <td>Ar condicionado</td>
216             </tr>
217         </table>
218     </div>
219 </footer>
220 </body>
221 </html>
222

```

Imagem(24). voo parte 4