



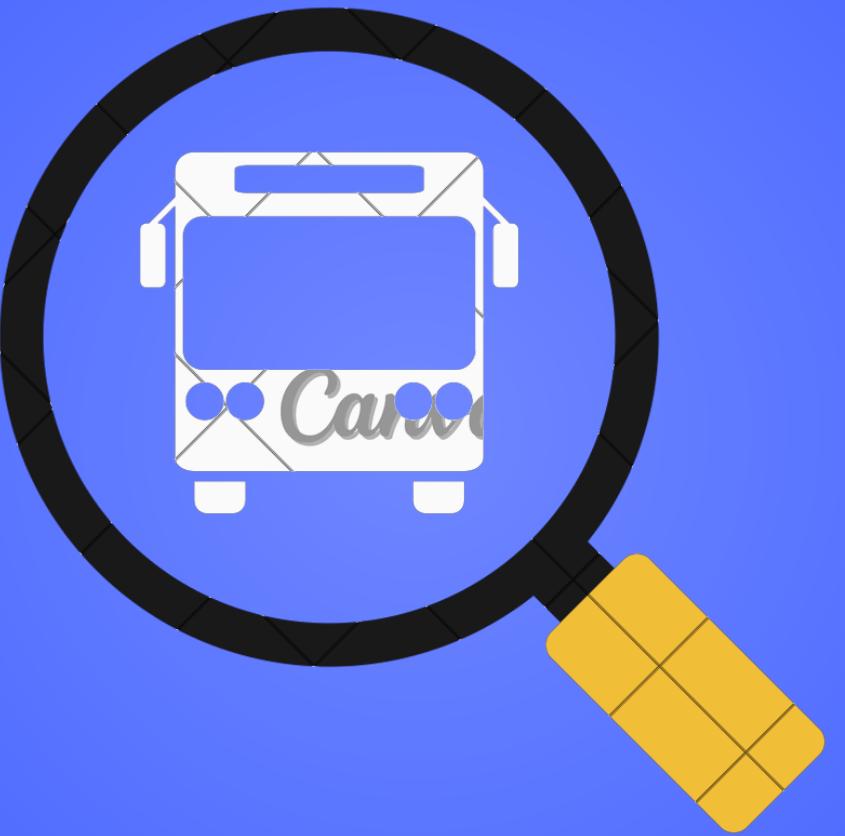
# MoveSmart

Grupo:  
Caio Faria Diniz;  
Felipe Bragança Godinho;  
Giuseppe Sena Cordeiro;  
Vinícius Miranda de Araújo;  
Yan Sabarense Coelho Silva.



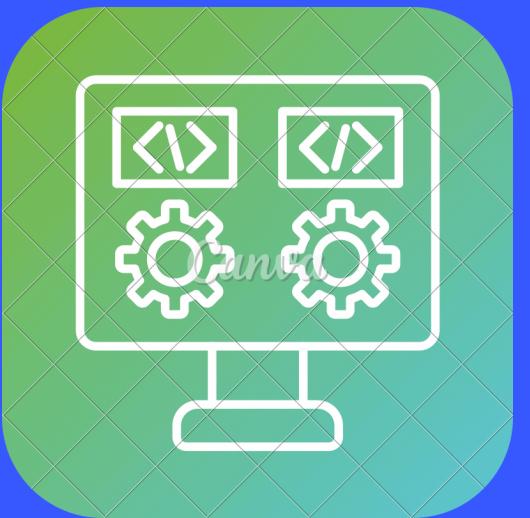
# Contextualização

- Aprimoramento da experiência do usuário no transporte público;
- Criação de notificações personalizadas;
- Fornecer informações em tempo real sobre a localização e o status dos transportes públicos.



# Ferramentas

Para o desenvolvimento de todas as partes do projeto foram utilizadas as seguintes ferramentas:



DER



Desenvolvimento



Versão de  
Código



Testes



# Artefatos

No BackEnd, utilizamos APIs para comunicação, bancos de dados para armazenar informações, autenticação e integração com serviços externos para funcionalidades adicionais.



Sistema de  
Banco de Dados



Armazenamento  
Azure



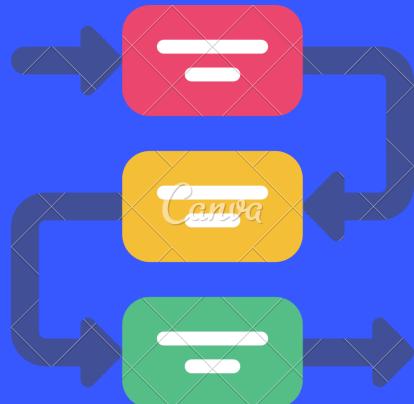
Conexão com o  
BackEnd com o  
SparkFramework



Principal  
Linguagem  
utilizada

# Dependências

As dependências utilizadas foram, em sua maioria, recomendadas pelos orientadores, com a adição das dependências google.gson, java.mail e azure.blob.

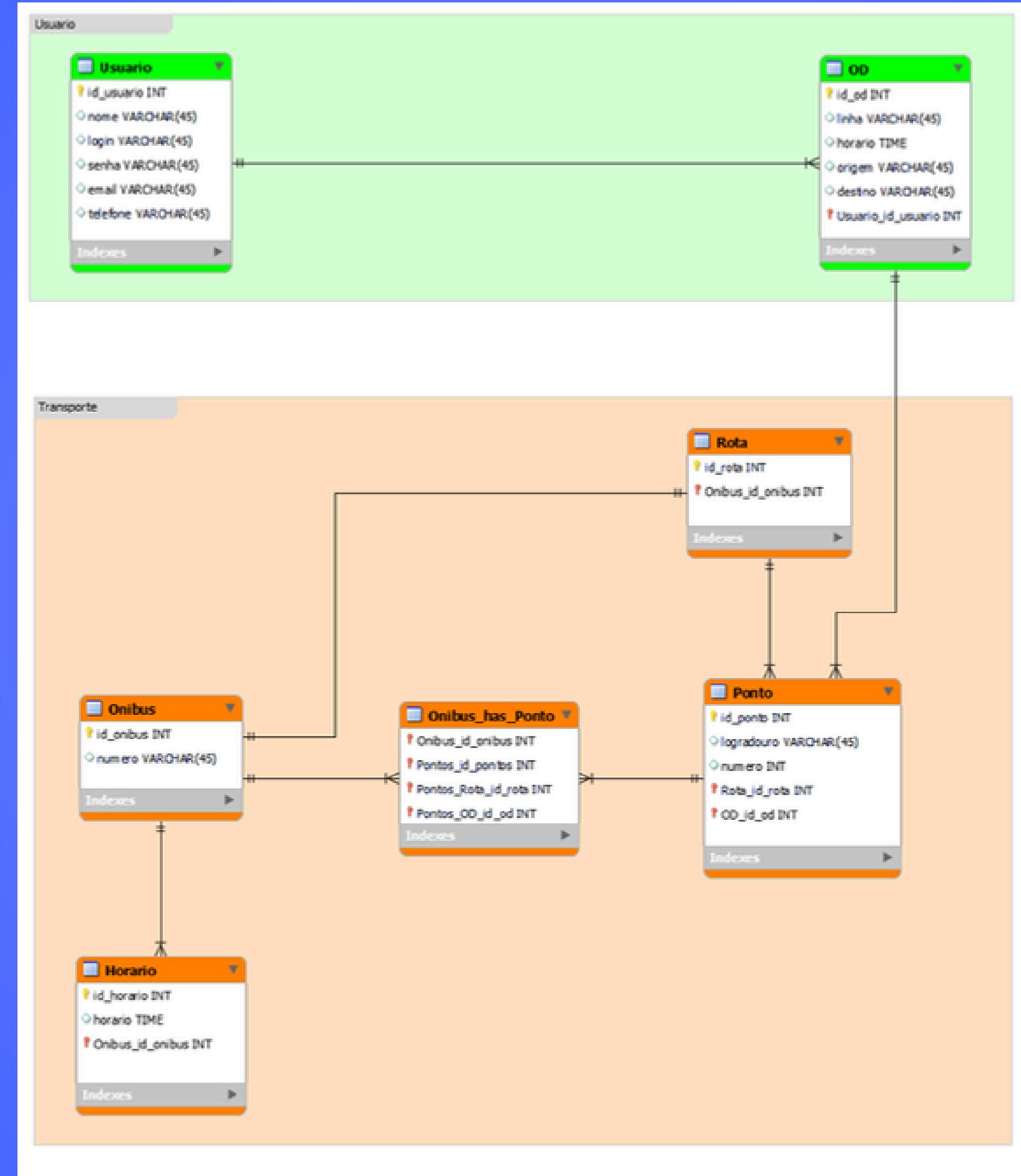


```
● ● ●
1 <!-- Dependência do Gson -->
2 <dependency>
3   <groupId>com.google.code.gson</groupId>
4   <artifactId>gson</artifactId>
5   <version>2.10.1</version>
6 </dependency>
7
8 <!-- Dependência do Java Mail -->
9 <dependency>
10  <groupId>org.simplejavamail</groupId>
11  <artifactId>simple-java-mail</artifactId>
12  <version>8.7.0</version>
13 </dependency>
14
15 <!-- Dependência da Azure -->
16 <dependency>
17   <groupId>com.azure</groupId>
18   <artifactId>azure-storage-blob</artifactId>
19   <version>12.14.0</version>
20 </dependency>
```

```
● ● ●
1 <!-- Dependência do Spark Framework -->
2 <dependency>
3   <groupId>com.sparkjava</groupId>
4   <artifactId>spark-core</artifactId>
5   <version>2.9.4</version>
6 </dependency>
7
8 <!-- Dependência do Maven -->
9 <dependency>
10  <groupId>org.apache.logging.log4j</groupId>
11  <artifactId>log4j-slf4j-impl</artifactId>
12  <version>2.14.1</version>
13 </dependency>
14
15 <!-- Dependência do PostgreSQL -->
16 <dependency>
17   <groupId>org.postgresql</groupId>
18   <artifactId>postgresql</artifactId>
19   <version>42.7.2</version>
20 </dependency>
```

# Banco de Dados

O Banco de Dados foi feito no MySQL, seguindo o modelo Relacional e notações Pé de Galinha

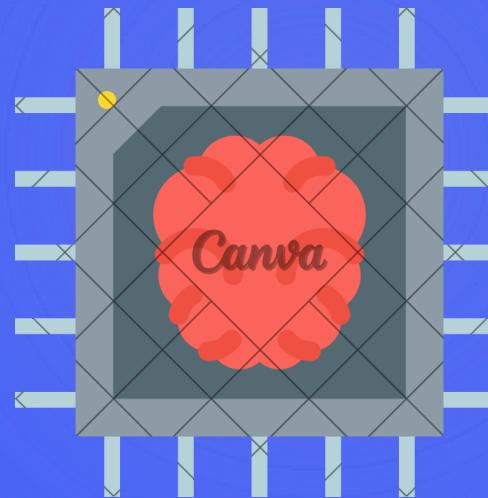


# Requisitos Funcionais



ID	Descrição
RF-001	O sistema deve permitir criar, atualizar e excluir um ponto de origem e de destino.
RF-002	O sistema deve enviar uma notificação ao usuário da linha que ele escolheu.
RF-003	O sistema deve permitir ao usuário visualizar rotas.
RF-004	O sistema deve planejar trajetos eficientes
RF-005	O sistema deve fornecer assistência por meio de chatbots em tempo real.

# Sistemas Inteligentes



Ferramental de IA	Entradas	Proposição de valor	Equipe	Cientes
<ul style="list-style-type: none"><li>• Sistemas inteligentes;</li><li>• Inteligência artificial;</li><li>• Chatbots de suporte em tempo real;</li><li>• Otimização dinâmica de rotas;</li><li>• Envio de notificações por e-mail.</li></ul>	<p><b>Entradas</b></p> <ul style="list-style-type: none"><li>• Condições de tráfego;</li><li>• Preferências do usuário;</li><li>• Dados de localização do usuário.</li></ul> <p><b>Saídas</b></p> <ul style="list-style-type: none"><li>• Rotas otimizadas;</li><li>• Respostas automatizadas via chatbots;</li><li>• Notificações por e-mail.</li></ul>	<ul style="list-style-type: none"><li>• Experiência personalizada e adaptável;</li><li>• Suporte em tempo real para usuários;</li><li>• Rotas otimizadas conforme condições de tráfego;</li><li>• Notificações programadas de acordo com a preferência do usuário.</li></ul>	<ul style="list-style-type: none"><li>• Desenvolvedores de IA;</li><li>• Especialistas em sistemas de tráfego;</li><li>• Suporte aos clientes;</li><li>• Analista de dados.</li></ul>	<ul style="list-style-type: none"><li>• Usuários de transporte urbano;</li><li>• Empresas de logística;</li><li>• Motoristas de aplicativos de transporte;</li><li>• Agências de transporte público.</li></ul>
<p><b>Custos</b></p> <ul style="list-style-type: none"><li>• Desenvolvimento e manutenção dos sistemas inteligentes;</li><li>• Infraestrutura tecnológica (servidores, bancos de dados);</li><li>• Licenciamento de dados de tráfego;</li><li>• Equipe de suporte e desenvolvimento.</li></ul>			<p><b>Receitas</b></p> <ul style="list-style-type: none"><li>• Assinaturas de usuários;</li><li>• Parcerias com empresas de transporte;</li><li>• Publicidade direcionada;</li><li>• Serviços premium (ex: relatórios avançados).</li></ul>	

# Aplicação

## Cadastro

## Usuário



**MOVE SMART**

### Faça seu cadastro!

Nome:

E-mail:

Telefone:

Login:

Senha:

Ao clicar no botão de cadastro, você concorda com os seguintes documentos:  
[Termos e Condições](#) e [Política de Privacidade](#).

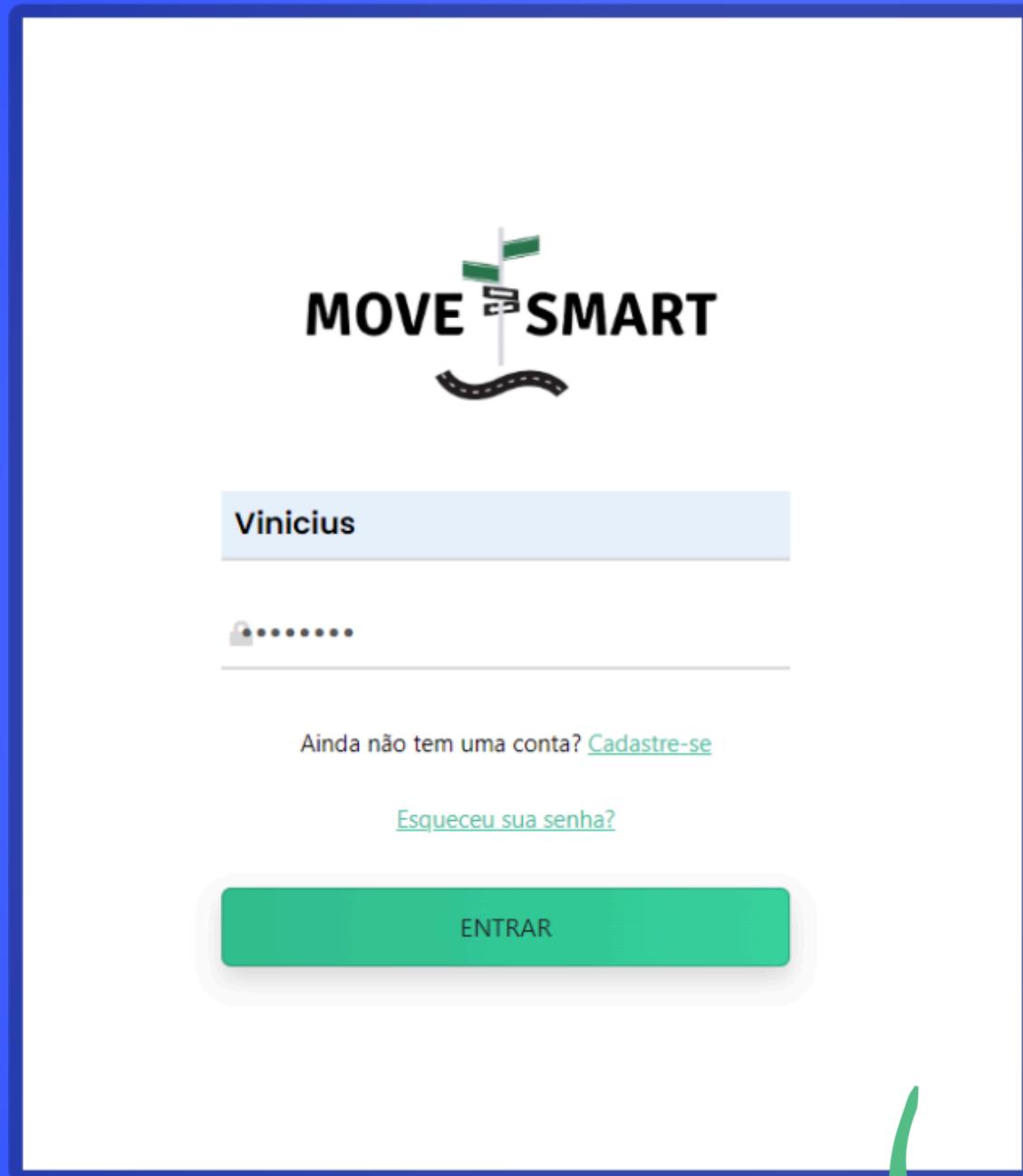
Já possui uma conta? [Acessar](#)

**Cadastrar**

```
1 // Insere um usuário
2     public Object insert(Request request, Response response) {
3
4         //verifica se o corpo da requisição é nulo
5         if(request.body() == null){
6             response.status(400); // 400 Bad Request
7             return "Corpo da requisição nulo!";
8         }
9
10        //transforma o json em um objeto do tipo Usuario
11        Usuario usuario = gson.fromJson(request.body(), Usuario.class);
12
13        if (usuarioDAO.exists(usuario.getIdUsuario())) {
14            response.status(409); // 409 Conflict
15            return "Usuário já existe!";
16        }
17
18        //verifica se a senha é valida
19        try {
20            String senhaMD5 = dao.toMD5Password(usuario.getSenha());
21            usuario.setSenha(senhaMD5);
22        } catch (Exception e) {
23            response.status(500); // 500 Internal Server Error
24            return "Erro ao converter a senha para MD5: " + e.getMessage();
25        }
26
27
28        //verifica se o usuário foi inserido com sucesso
29        if (usuarioDAO.insert(usuario)) {
30            response.status(201); // 201 Created
31            return "Usuario inserido com sucesso!";
32        } else {
33            response.status(500); // 500 Internal Server Error
34            return "Erro ao inserir contato!";
35        }
36    }
```

	<b>id_usuario</b> integer	<b>nome</b> character varying	<b>login</b> character varying	<b>senha</b> character varying	<b>email</b> character varying	<b>telefone</b> character varying
1	1	teste	teste	teste	teste@gmail.com	123456789
2	2	Vinicius	Vinicius	e7d80ffefafa212b7c5c55700e4f7193e	vinicius123miranda24@gmail.com	31982537764

# Autenticação do Usuário



```
● ● ●
1 public Object authenticate(Request request, Response response) {
2     // Verifica se o corpo da requisição é nulo
3     if (request.body() == null) {
4         response.status(400); // 400 Bad Request
5         return "Corpo da requisição nulo!";
6     }
7
8     // Transforma o json em um objeto do tipo Usuario
9     Usuario usuario = gson.fromJson(request.body(), Usuario.class);
10
11    // Pega o nome de usuário e a senha
12    String login = usuario.getLogin();
13    String senha = usuario.getSenha();
14
15    // Autentica o usuário
16    Usuario usuarioAutenticado = usuarioDAO.getByUsername(login);
17    if (usuarioAutenticado == null) {
18        response.status(401); // 401 Unauthorized
19        return "Falha na autenticação!";
20    }
21
22    String senhaMD5;
23    try {
24        senhaMD5 = dao.toMD5Password(senha);
25    } catch (Exception e) {
26        response.status(500); // 500 Internal Server Error
27        return "Erro ao converter a senha para MD5: " + e.getMessage();
28    }
29
30    if (usuarioAutenticado.getSenha().equals(senhaMD5)) {
31        response.status(200); // 200 OK
32        return "Usuário autenticado com sucesso!";
33    } else {
34        response.status(401); // 401 Unauthorized
35        return "Falha na autenticação!";
36    }
37 }
```

A terminal window displays the output of a command. It starts with '127.0.0.1:5500 diz' followed by the message 'Você foi autenticado com sucesso!' (You were successfully authenticated!). A large green curved arrow points from the bottom right of the code editor towards this terminal window.

```
127.0.0.1:5500 diz
Você foi autenticado com sucesso!
```

# Vídeo



MoveSmart - Cadastro

127.0.0.1:5500/Codigo/BackEnd/src/main/resources/public/cadastro.html

Google Email YouTube GitHub LinkedIn PUC Outros

**MOVE SMART**

**Faça seu cadastro!**

**Nome:**  
Vinícius M

**E-mail:**  
vinicius123miranda24  
vinicius123miranda24@gmail.c...

**Login:**

**Senha:**

Ao clicar no botão de cadastro, você concorda com os seguintes documentos:  
[Termos e Condições](#) e [Política de Privacidade](#).

Já possui uma conta?

**Cadastrar**

22:43 20/06/2024



# MUITO OBRIGADO!

Navegue com Eficiência, Viaje com Inteligência!

