

A black and white photograph of a multi-lane highway with several trucks in motion, blurred to indicate speed. A semi-transparent globe is overlaid on the image, showing a network of shipping routes with lines connecting various ports and cities around the world.

# TRUCK LOG

# 01 TIME



## MARCKLEN GUIMARÃES

Desenvolvedor Backend

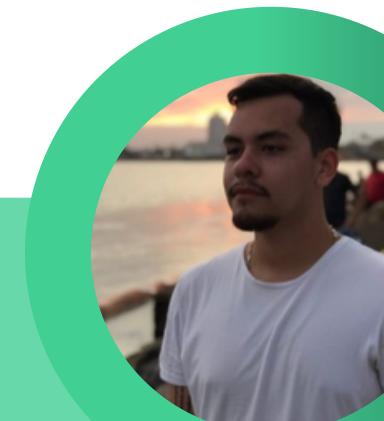
Atua no time da trucklog como um desenvolvedor backend utilizando Java e Spring Boot para o desenvolvimento de APIs Rest, responsável pela implementação da ultima versão do projeto aplicando Spring Security e JWT.



## MARIANA MACHADO

Desenvolvedora Backend

Atua no time da trucklog como um desenvolvedor backend utilizando Java e Spring Boot para o desenvolvimento de APIs Rest, responsável pela implementação da ultima versão do projeto aplicando Spring Security e JWT.



## LUCAS ALVES

Desenvolvedor Backend

Atua no time da trucklog como um desenvolvedor backend utilizando Java e Spring Boot para o desenvolvimento de APIs Rest, responsável pela implementação da ultima versão do projeto aplicando Spring Security e JWT.

## 02 AGRADECIMENTO AO TIME FRONTEND

Luan Lima

Leonardo  
Vasconcelos

Rafael Ramos

# SISTEMA LOGÍSTICO

## 01 Objetivo

Garantir a disponibilidade de insumos e produtos onde quer que eles sejam necessários, de forma eficiente e com o menor custo.

## 02 Função

Interligar as atividades logísticas, desde a administração de recursos até a entrega no cliente final. Com o máximo de precisão para otimizar a operação e eliminar falhas.

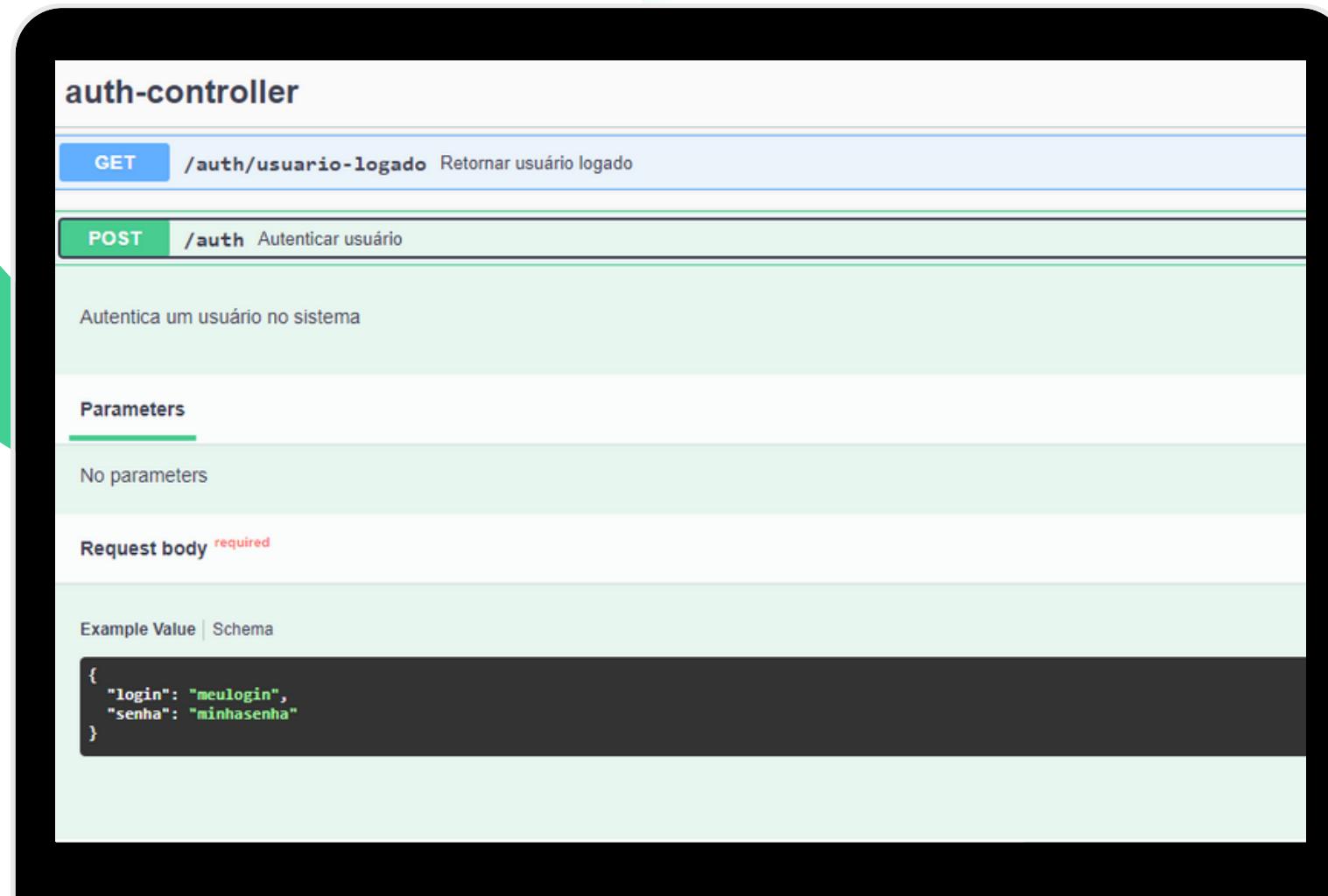
## 03 Vantagens

- Maior acessibilidade e disponibilidade das informações;
- Planejamento simplificado;
- Economia operacional;
- Integração entre softwares;
- Logística eficaz;
- Redução de despesas.



# REQUISITOS MINIMOS

O projeto segue o padrão MVC conforme padrão do projeto. Utiliza classes DTOs para entrada e saída de dados e conta com muitos endpoints paginados, além de relatórios que também utilizam paginação. A aplicação conta com pelo menos 4 APIs que realizam o CRUD de dados e todas suas regras de negócio estão contidas na service. A API esta no ambiente cloud (Jenkins) e na versão 5.0 conta com toda a segurança do **Spring Security** fazendo o permissionamento dos endpoints e a utilização de tokens para validação e integração ao sistema.



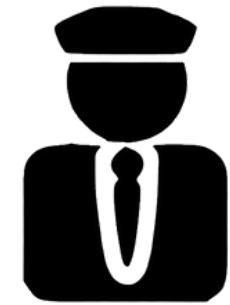
# PERMISSÕES CARGOS



ADMINISTRADOR



COLABORADOR



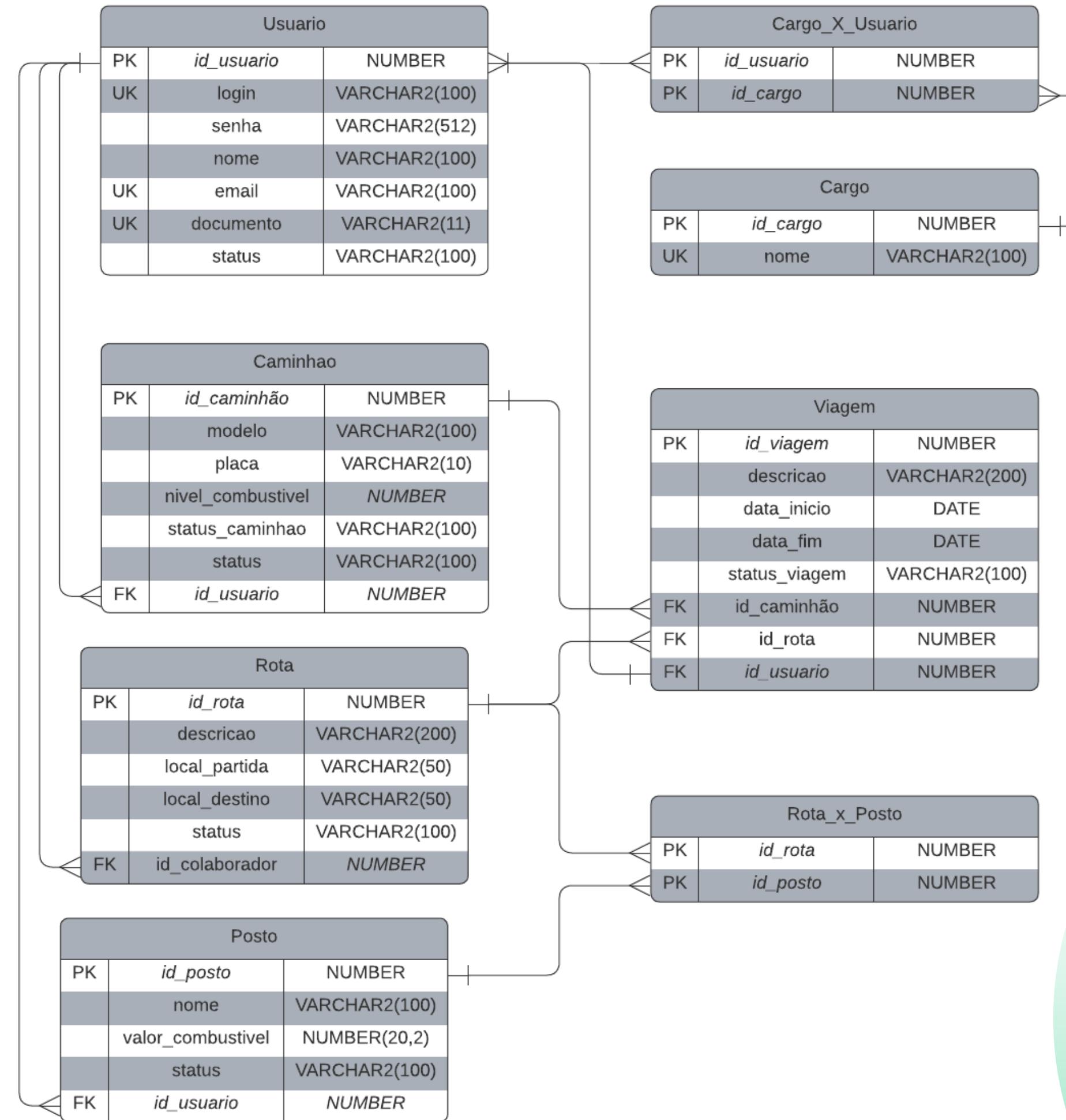
MOTORISTA

Acesso a todas as funcionalidades do sistema

Acesso a rotas, postos, caminhões e a seus dados

Acesso a viagens e caminhões para abastecimento e a seus dados

# DIAGRAMA ER



# SCRYPTPASSWORDENCODER

O SCryptPasswordEncoder é um algoritmo de hash de senhas utilizado em aplicações Java Spring Security. Ele é usado para armazenar senhas de usuários de forma segura e eficiente, protegendo-as de possíveis ataques de hackers.

A documentação oficial do **Spring Security** é uma excelente fonte para entender como o algoritmo funciona e como utilizá-lo em suas aplicações.

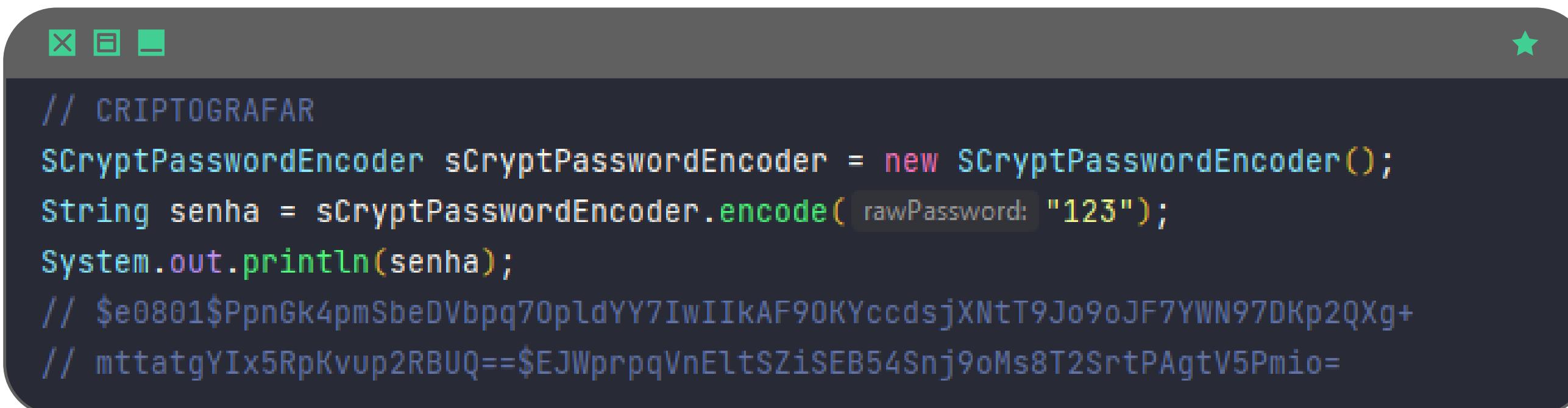
O **Bouncy Castle** é uma biblioteca de criptografia Java de código aberto que inclui suporte ao algoritmo SCryptPasswordEncoder.

## 01 Implementação

```
@Bean  
public PasswordEncoder  
passwordEncoder() {  
    // SCrypt algoritmo de criptografia  
    return new SCryptPasswordEncoder();  
}
```

## 02 Dependência

```
<dependency>  
    <groupId>org.bouncycastle</groupId>  
    <artifactId>bcpprov-jdk15on</artifactId>  
    <version>1.68</version>  
</dependency>
```



```
// CRIPTOGRAFAR  
SCryptPasswordEncoder sCryptPasswordEncoder = new SCryptPasswordEncoder();  
String senha = sCryptPasswordEncoder.encode("rawPassword: "123");  
System.out.println(senha);  
// $e0801$PpnGk4pmSbeDVbpq70pldYY7IwIIkAF9OKYccdsjXNtT9Jo9oJF7YWN97DKp2QXg+  
// mttatgYIx5RpKvup2RBUQ===$EJWprpqVnEltSZiSEB54Snj9oMs8T2SrtPAgtV5Pmio=
```

# E-MAIL



The image shows a simulated email interface. At the top, there's a header bar with standard window controls (X, square, minus). Below it is a teal-colored header section featuring the TruckLog logo, which consists of a stylized white truck icon with motion lines and the word "TruckLog" in a bold, sans-serif font.

The main content area starts with a greeting: "Olá, \${nomeUsuario}, seja bem vindo(a)!" followed by a blank line. Then, there's a paragraph of text: "É com grande prazer que lhe damos as boas-vindas à equipe da \${nome}! Nosso principal objetivo é fornecer aos nossos clientes os melhores serviços personalizados e soluções em logística. E isto não seria possível sem a sua contribuição. Você é peça chave neste processo!" Another blank line follows. The next section contains information about the user: "Seguem seus dados cadastrados:  
**Email:** \${emailUsuario}" followed by another blank line. Below that is a contact note: "Qualquer dúvida é só contatar o suporte pelo email -> \${emailContato}" followed by another blank line. The final section is a closing salutation: "Atenciosamente,  
**\${nome}**" followed by a final blank line.

In the bottom right corner of the email body, there's a large, bold, teal-colored text block containing the phrase: "**"Voamos por você!"**".

At the very bottom of the email interface, there are two teal-colored navigation arrows: a left arrow on the left and a right arrow on the right, positioned on either side of a large, rounded teal button.



**TruckLog**

# OBRIGADO!



**TELEFONE**

+123-456-7890



**EMAIL**

[trucklog@email.com](mailto:trucklog@email.com)



**LOCALIZAÇÃO**

Tv. São José, 455 -  
Navegantes - Porto Alegre