

TIME 07 - Backend

NOTIFICA



Renata Oliveira Schäfer

Leonardo Faneze

Andre Miranda

Introdução



NOTIFICA

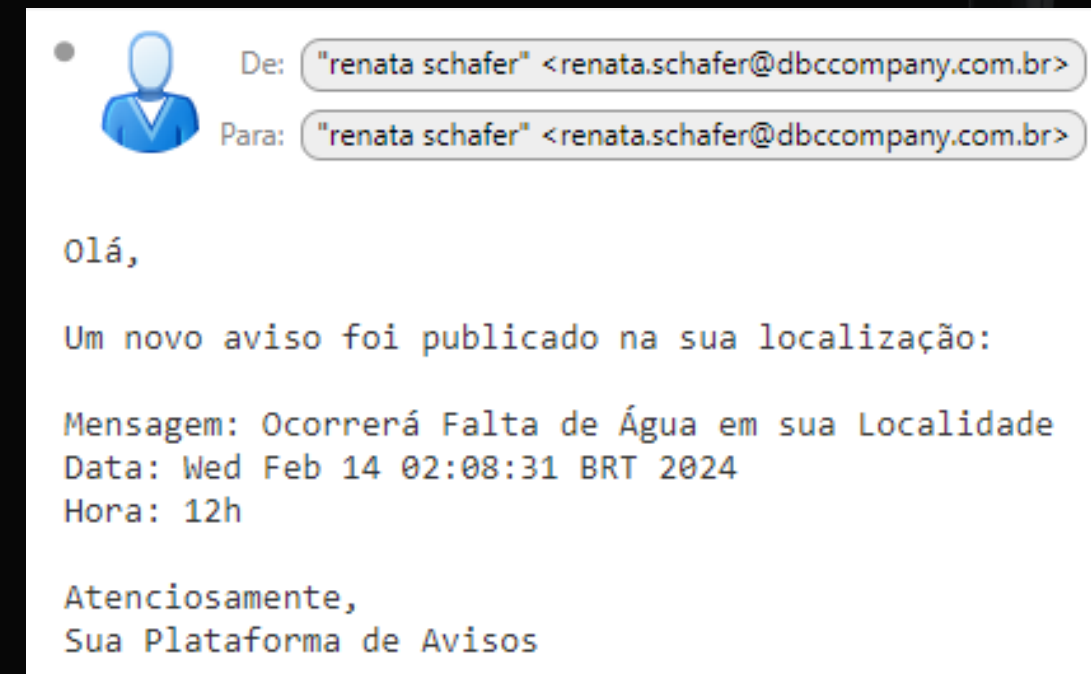
É um sistema de denúncias relacionado à ODS 6, que trata de saneamento e água potável. Essa ideia surgiu em resposta a diversos casos, como os ocorridos em Porto Alegre, nos quais comunidades ficaram semanas sem acesso à água.

O **Notifica** tem como objetivo centralizar e agilizar o processo de denúncias, visando dar voz a essas comunidades, buscando garantir o acesso adequado à água potável e saneamento básico.

Principais Funcionalidades Adicionadas

Usuário Instituição

Álem de Admin e Usuário, agora temos uma nova entidade, a Instituição, a ideia instalada é que essas instituições que forem criadas pelos Admins, possam criar Avisos para Usuários Comuns que adentraram a plataforma com endereço preenchido por meio de E-mails.

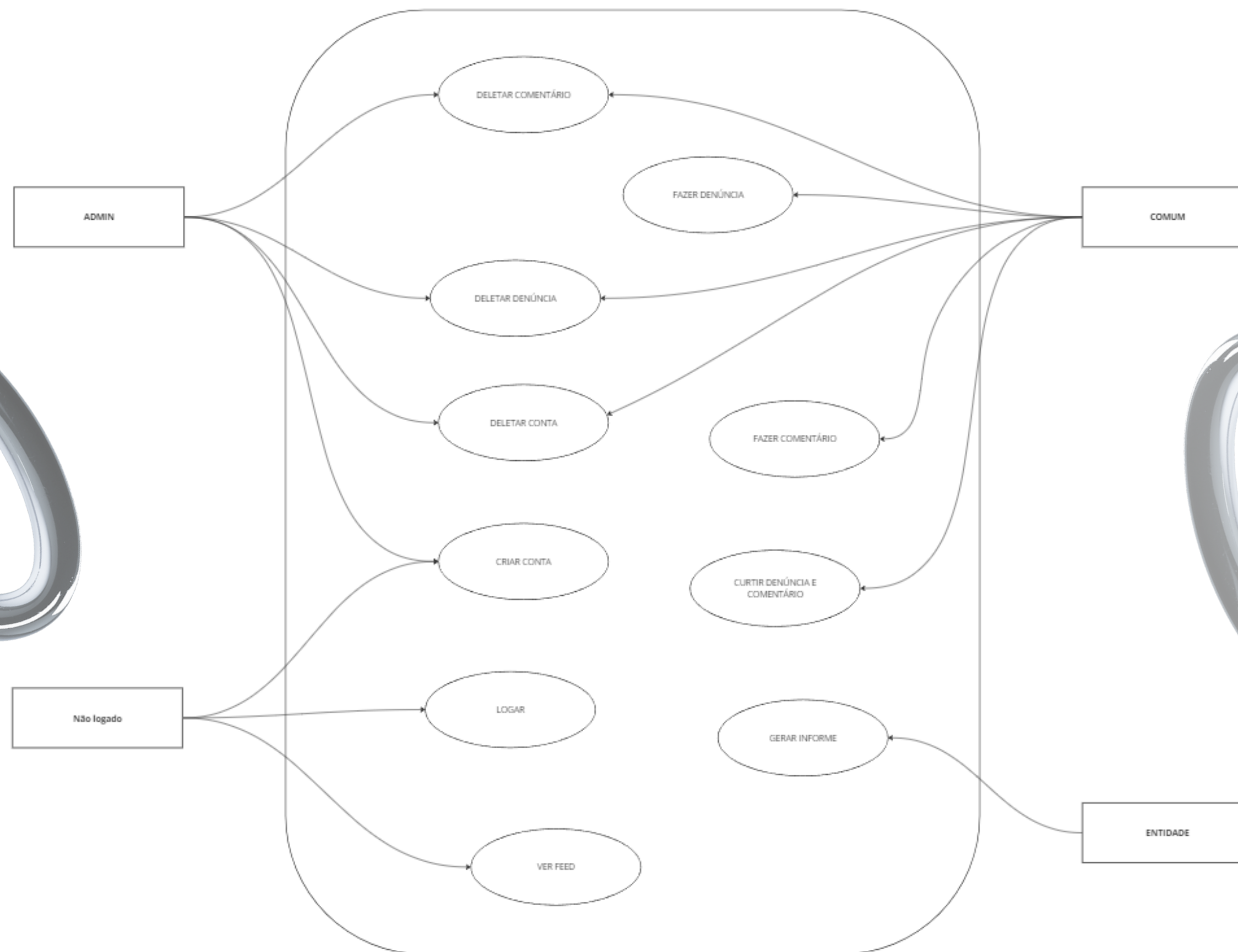


Criação de Usuário com Senha Argon2, Troca de Senha, Desativação, Atualização de Usuários e Login

Implementamos o Cadastro de Usuário, agora com senhas criptografadas usando Argos2. Adicionamos funcionalidades como Troca de Senha. O sistema também permite Desativação e Ativação de Conta. Além disso, possibilitamos a Atualização de Dados para manter as informações pessoais precisas. Para garantir uma entrada eficiente, foi implementado o procedimento de Login.

Usuário Controller	
PUT	/usuario/atualizar-senha
PUT	/usuario/atualizar-perfil
GET	/usuario/meu-perfil
DELETE	/usuario/desativar-perfil

Diagrama de Caso de Uso TIME 07



Rodando no Swagger

Agora iremos fazer uma breve demonstração de Usuário:

- Fazer Cadastro de Usuário
- Login nesse Cadastro, Inserir Token
- Troca de Senha
- Desativação (Exclusão)

PUT /usuario/atualizar-senha

Parameters

Name	Description
senhaAtual * required string (query)	Senha123@
novaSenha * required string (query)	Senha321@

DELETE /usuario/desativar-perfil

Parameters

No parameters

POST /login/cadastrar Fazer Cadastro

Fazer Cadastro com Email e Senha, se os dados não for

Parameters

No parameters

Request body **required**

```
{  "nomeUsuario": "Renata Oliveira",  "emailUsuario": "renata2.schafer@dbcc",  "numeroCelular": "998427710",  "senhaUsuario": "Senha123@",  "etniaUsuario": "BRANCO",  "dataNascimento": "1990-01-01",  "classeSocial": "A",  "generoUsuario": "MASCULINO"}
```

POST /login/usuario Fazer Login Usuario

Fazer Login com Email e Senha, se os dados estiverem no b

Parameters

No parameters

Request body **required**

```
{  "emailUsuario": "renata2.schafer@dbcccompany.com.",  "senhaUsuario": "Senha123@"}
```

Available authorizations

bearerAuth (http, Bearer)

Authorized

Value: *****

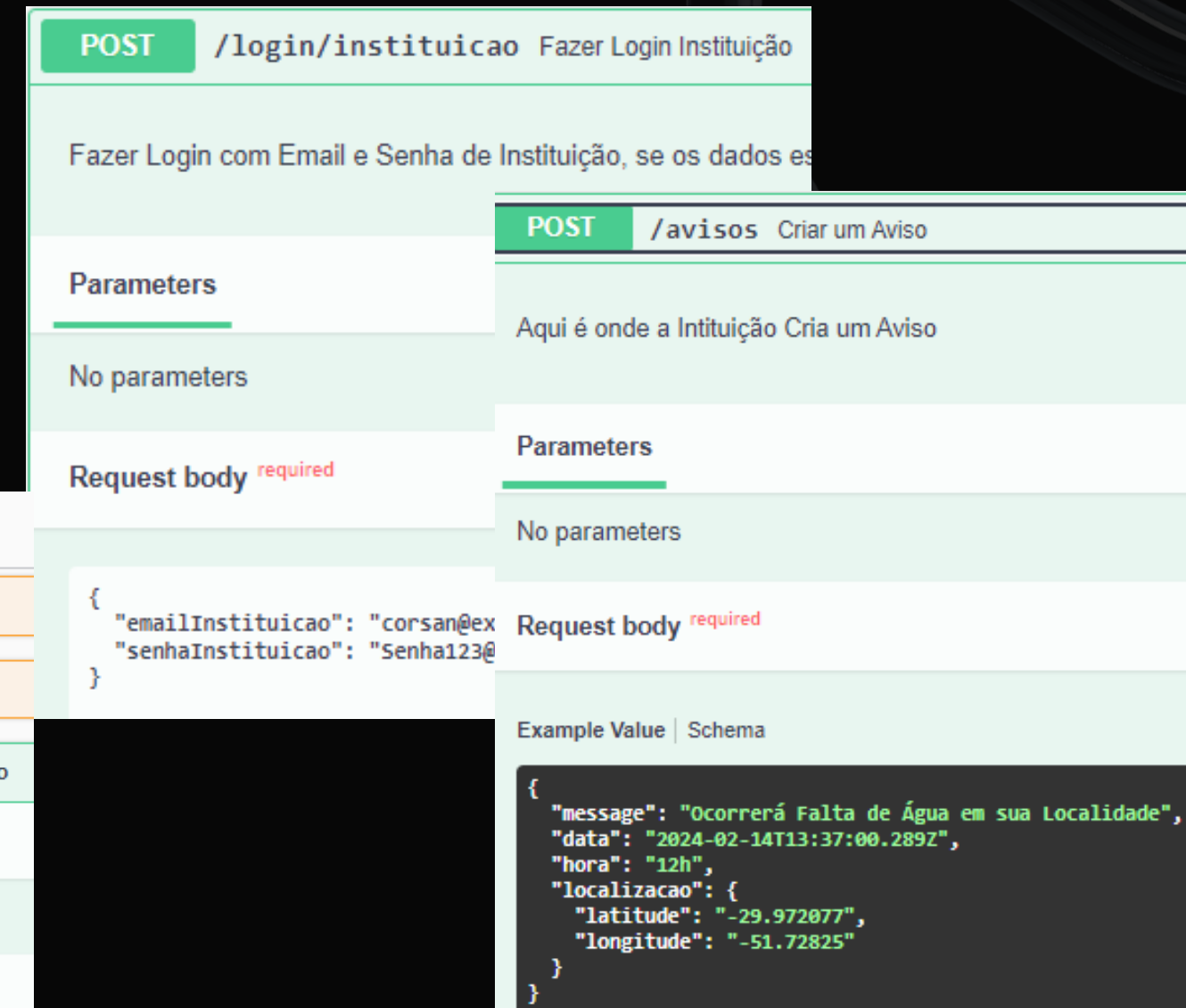
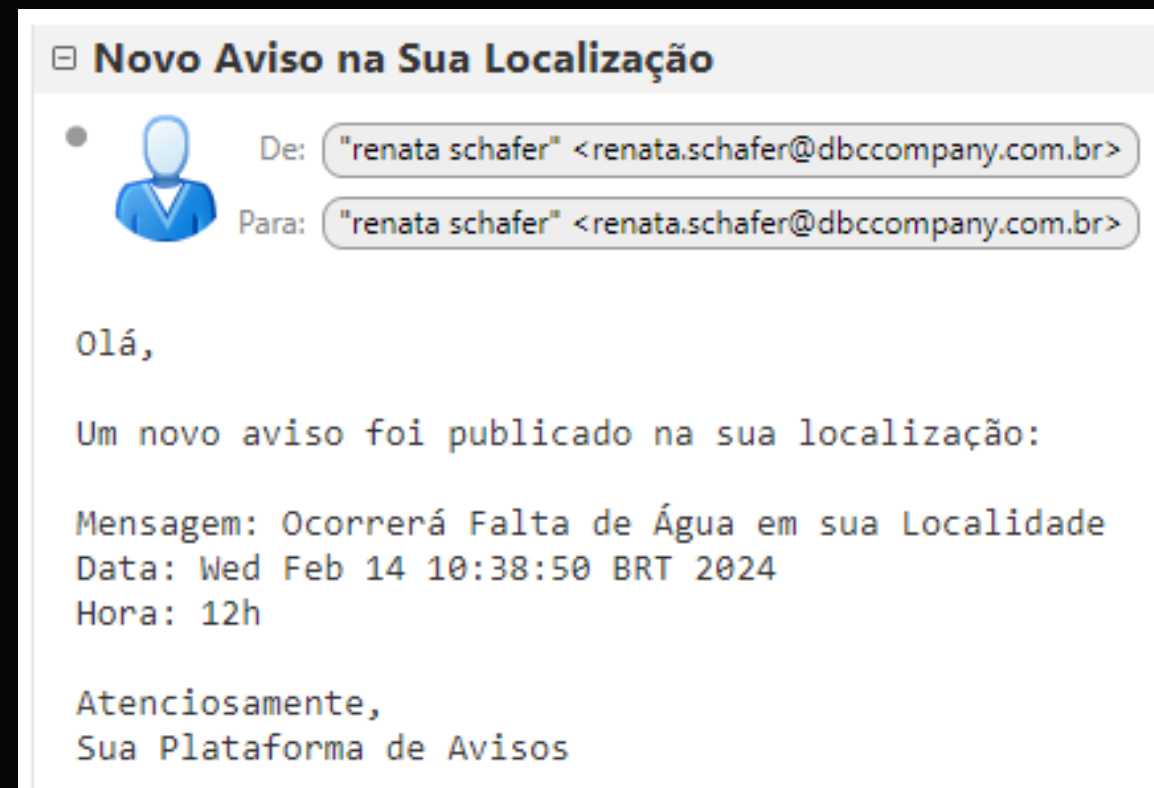
Logout

123 ID_USUARIO	ABC NOME_USUARIO	ABC EMAIL_USUARIO	ABC ATIVO
3	Andre	andre@dbcccompany.com.br	1
4	Usuario	usuario@example.com	1
5	Admin	admin@example.com	1
6	Renata Oliveira	renata2.schafer@dbcccompany.com.br	0

Rodando no Swagger

Agora iremos fazer uma breve demonstração de Instituição:

- Login como Instituição e Token
- Criação de Aviso
- Demonstração do Envio de E-mail para Localidades Iguais
- Mostre a Criação de Instituição dentro da Controller Admin



Criptografia com Argon2

O que é Argon2?

A função Argon2 foi selecionada como vencedora do Concurso de Função de Hash de Senha (Password Hashing Competition) em 2015. Ela tem várias características importantes que a tornam uma escolha sólida para armazenamento seguro de senhas.

Video Explicativo Akita



Como Funciona o Argon2?

1. O processo começa quando o Argon2 recebe a senha do usuário.
2. Argon2 é altamente configurável, permitindo ajustar três parâmetros:
 - **Tempo (t)**: Determina o número de iterações realizadas durante o processo de hash.
 - **Memória (m)**: Define a quantidade de memória utilizada durante o cálculo do hash.
 - **Paralelismo (p)**: Especifica o número de threads ou unidades de processamento paralelo.
3. Um salt aleatório é gerado para cada senha. O salt é uma sequência única de caracteres que é combinada com a senha original antes do processo de hashing. Isso garante que mesmo senhas idênticas tenham hashes diferentes devido aos salts únicos.
4. Argon2 executa iterações intensivas de computação, onde a senha e o salt são processados repetidamente. O número de iterações é determinado pelo parâmetro de tempo (t). Esse processo consome tempo e recursos computacionais, tornando ataques de força bruta demorados e custosos.
5. O resultado final do processo é o hash da senha, combinado com o salt. Ambos o salt e o hash resultante são armazenados no banco de dados.

ABC	SENHA_USUARIO	\$argon2<versão>\$<parâmetros>\$<salt>\$<hash_da_senha>
		\$argon2id\$v=19\$m=4096,t=3,p=1\$C07W8gudeN47XTxO7vrOHw\$JPkNbiSoHvFC0Wvh+BEe/UeXePs7eW2liA2L+O6+6VY
		\$argon2id\$v=19\$m=4096,t=3,p=1\$C07W8gudeN47XTxO7vrOHw\$JPkNbiSoHvFC0Wvh+BEe/UeXePs7eW2liA2L+O6+6VY
		\$argon2id\$v=19\$m=4096,t=3,p=1\$bglxa9DdU+RgNeTZmhnDvg\$JNmVNIVPvoGvaCTny1+coJRjHN+g+FC9caqQWMzizNw
		\$argon2id\$v=19\$m=4096,t=3,p=1\$M+aKVgXJv1fkojVxiFFLQw\$1+N/8BQsSvB3q1CvrF+ikesPXcbTHBwAL119VfOqkCE

TIME 07

NOTIFICA

TIME 07 AGRADECE A SUA ATENÇÃO!