



SENAI

ESCOLA SENAI "SANTO PASCHOAL CREPALDI"
UMA REALIZAÇÃO DA INDÚSTRIA

Técnico em Desenvolvimento de Sistemas

UI - Banco de dados

Criptografi



a



Bruno Ferreira | Matheus Oliveira

| O que é criptografia?

A criptografia é um dos elementos fundamentais da segurança cibernética. Ela é usada para proteger dados contra roubos, alterações ou comprometimentos. Ela funciona ao embaralhar dados em um código secreto que só pode ser desbloqueado com uma chave digital exclusiva. Os dados criptografados podem ser protegidos em repouso em computadores ou em trânsito entre eles ou enquanto são processados, independentemente de estarem no local ou em servidores de nuvem remotos.

O que é criptografia?

A criptografia codifica o "texto simples" em "texto criptografado", normalmente com o uso de modelos matemáticos criptográficos conhecidos como algoritmos. Para decodificar os dados de volta para texto simples, é necessário o uso de uma chave de descryptografia, uma string de números ou uma senha também criada por um algoritmo. Os métodos de criptografia segura têm um número tão grande de chaves criptográficas que uma pessoa não autorizada não consegue adivinhar qual delas está correta

História



Um dos primeiros exemplos de uma criptografia simples é a "Caesar crypt", em homenagem ao imperador romano Julius Caesar, porque ele a usou em sua correspondência particular. O método é um tipo de criptografia de substituição, em que uma letra é substituída por outra letra e um número fixo de posições é usado no alfabeto. Para descriptografar o texto codificado, o destinatário precisa saber a chave da criptografia, como mover o alfabeto para baixo quatro vezes e para a esquerda (um "shift de quatro para a esquerda"). Assim, cada "E" se torna um "Y" e assim por diante.

A criptografia moderna é muito mais sofisticada, usando strings de centenas (até milhares, em alguns casos) de caracteres gerados por computador como chaves de descriptografia.



Os dois tipos mais comuns de algoritmos de criptografia são simétricos e assimétricos.

A criptografia simétrica, também conhecida como chave compartilhada ou algoritmo de chave privada, usa a mesma chave para criptografia e descryptografia. As criptografias de chave simétrica são consideradas mais baratas para produzir e não usam tanta força de computação para criptografar e descryptografar, o que significa que há menos atraso na decodificação dos dados.

A desvantagem é que, se uma pessoa não autorizada colocar as mãos na chave, ela pode descryptografar todas as mensagens e dados enviados entre as partes. Por isso, a transferência da chave compartilhada precisa ser criptografada com uma chave criptográfica diferente, levando a um ciclo de dependência.

A criptografia assimétrica, também conhecida como criptografia de chave pública, usa duas chaves separadas para criptografar e descryptografar dados. Uma é uma chave pública compartilhada entre todas as partes para criptografia. Qualquer pessoa com a chave pública pode enviar uma mensagem criptografada, mas apenas os detentores da segunda chave privada poderão descryptografá-la.

A criptografia assimétrica é considerada mais cara para ser produzida e precisa de mais capacidade computacional para descryptografar, já que a chave pública costuma ser grande, entre 1.024 e 2.048 bits. Por isso, a criptografia assimétrica muitas vezes não é adequada para grandes pacotes de dados.

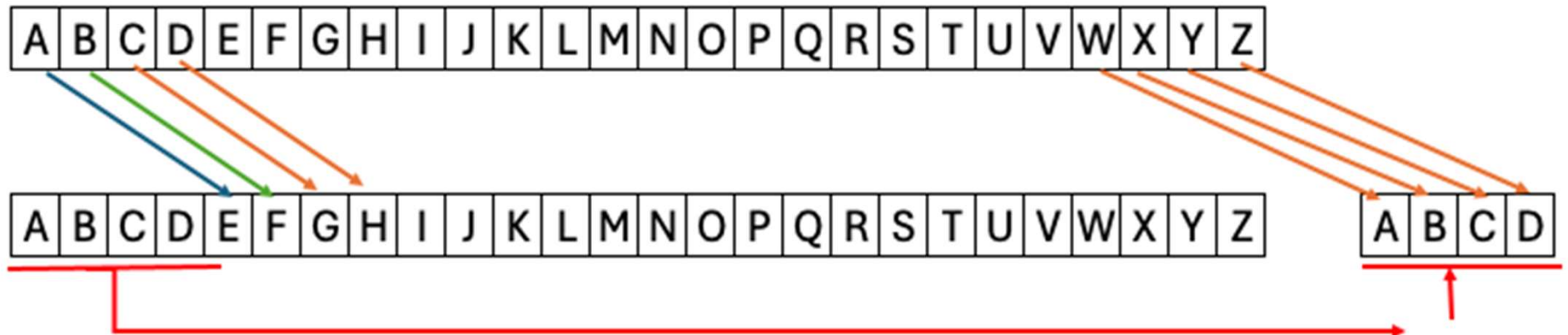
TRADUZAM A MENSAGEM!!

E	X	E	G	E	V		E	S		Q	T	M	S		H	M	E
---	---	---	---	---	---	--	---	---	--	---	---	---	---	--	---	---	---

TRADUZAM A MENSAGEM!!

E	X	E	G	E	V		E	S		Q	T	M	S		H	M	E
---	---	---	---	---	---	--	---	---	--	---	---	---	---	--	---	---	---

chave alfabeto + 4 posicoes





A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

E	X	E	G	E	V		E	S		Q	I	M	S		H	M	E
---	---	---	---	---	---	--	---	---	--	---	---	---	---	--	---	---	---

E	X	E	G	E	V		E	S		Q	I	M	S		H	M	E
---	---	---	---	---	---	--	---	---	--	---	---	---	---	--	---	---	---

A	T	A	C	A	R		A	O		M	E	I	O		D	I	A
---	---	---	---	---	---	--	---	---	--	---	---	---	---	--	---	---	---

Vamos para o



Criaremos em nosso banco a tabela **MySQL** **usuarios**

```
CREATE TABLE `usuarios` (  
  `id` INT NOT NULL AUTO_INCREMENT,  
  `usuario` VARCHAR(50) NOT NULL,  
  `nome_usuario` VARCHAR(100) NOT NULL,  
  `senha` VARCHAR(250) NOT NULL,  
  PRIMARY KEY (`id`)  
);
```

Vamos para o MySQL



Para trabalharmos com o armazenamento de senhas dos usuários da nossa aplicação
Não devemos salvá-las como os mesmos cadastram. por exemplo :

vendas.usuarios: 1 registros totais (exact)

#	id	usuario	nome_usuario	senha
1	1	bruno	BRUNO FERREIRA	1234

Recuperacao de senha:

Login

Usuario

Senha

☒ Lembrar neste computador

Sign In

[Esqueci minha senha!! Recuperar!](#)



Olá, Bruno Ferreira, você solicitou a recuperação da sua senha.

Para acessar nossos serviços utilize:

usuario: bruno

Senha: 1234



Olá, Bruno Ferreira, você solicitou a recuperação da sua senha.

Para acessar nossos serviços, utilize:

Clique no link, responda às confirmações de segurança e cadastre nova senha.

<https://ferreirabs.com.br/recuperarSenha>



md5

acento grave

```
sgs > exelmplos.sql
1  INSERT INTO `vendas`.`usuarios`
2  |      (`usuario`, `nome_usuario`, `senha`)
3  VALUES
4  |      ('ferreira', 'BRUNO FERREIRA', MD5('1234'));
5
```

Aspas
simples

Utilizando md5

ndas.usuarios: 2 registros totais (exibindo 2)

	id	usuario	nome_usuario	senha	
1	1	bruno	BRUNO FERREIRA	1234	incorreto
2	2	ferreira	BRUNO FERREIRA	81dc9bdb52d04dc20036dbd8313ed055	md5

SHA

```
sgc> exelmplos.sql
1  INSERT INTO `vendas`.`usuarios`
2    (`usuario`, `nome_usuario`, `senha`)
3  VALUES
4    ('ferreira', 'BRUNO FERREIRA', SHA1('1234'));
5
6
```

Utilizando

SHA1


vendas.usuarios: 3 registros totais Exatidão [» Próximo](#)

#	id	usuario	nome_usuario	senha	
1	1	bruno	BRUNO FERREIRA	1234	incorreto
2	2	ferreira	BRUNO FERREIRA	81dc9bdb52d04dc20036dbd8313ed055	md5
3	3	ferreira	BRUNO FERREIRA	7110eda4d09e062aa5e4a390b0a572ac0d2c0220	SHA1

SHA

2

```
sgt > exelmplos.sql
1  INSERT INTO `vendas`.`usuarios`
2    (`usuario`, `nome_usuario`, `senha`)
3  VALUES
4    ('ferreira', 'BRUNO FERREIRA', SHA2('1234', 256));
5
6
7
```

A red arrow points from the bottom right towards the number 256 in the SQL code, which is the second argument to the SHA2 function.

Utilizando SHA2

vendas.usuarios: 4 registros totais (exact) >> Próximo <◀ Mostrar todos

#	id	usuario	nome_usuario	senha	
1	1	bruno	BRUNO FERREIRA	1234	incorreto
2	2	ferreira	BRUNO FERREIRA	81dc9bdb52d04dc20036dbd8313ed055	md5
3	3	ferreira	BRUNO FERREIRA	7110eda4d09e062aa5e4a390b0a572ac0d2c0220	SHA1
4	4	ferreira	BRUNO FERREIRA	03ac674216f3e15c761ee1a5e255f067953623c8b388b4459e13f978d7c846f4	SHA2 (256 bits)

E como recuperar essa senha

A large, stylized red question mark is centered on the slide. It has a thick, rounded stroke and a separate circular dot at the bottom. A black question mark is placed inside the white circular area of the large red question mark.

?

A resposta é

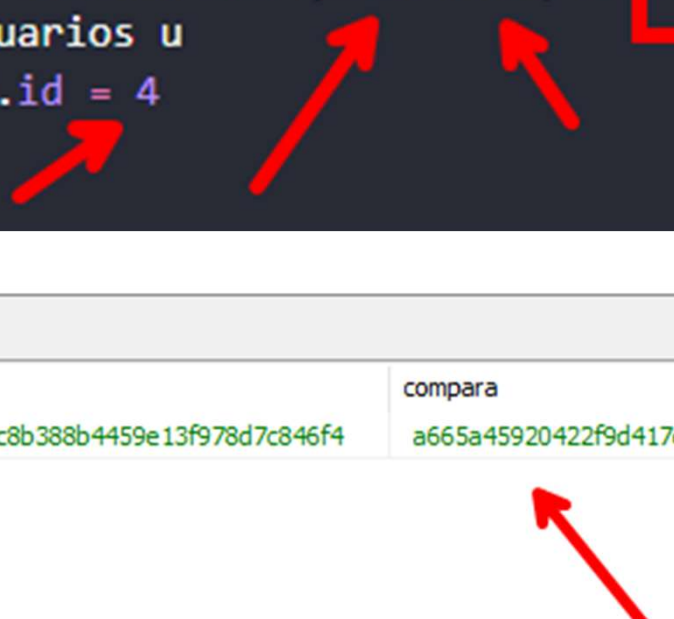


:




Para autenticar o usuário em seu aplicativo, você deve comparar o hash salvo no banco com o cálculo do hash que o usuário está informando no campo senha.

```
sgc > exelmplos.sql
1  SELECT u.senha, SHA2('123',256) AS compara
2  FROM usuarios u
3  WHERE u.id = 4
4
5
```

Four red arrows originate from the SQL query and point to the corresponding parts of the table result below. One arrow points from 'u.senha' to the first column header 'senha'. Another points from 'SHA2('123',256)' to the second column header 'compara'. A third points from 'AS compara' to the second column header. A fourth points from 'WHERE u.id = 4' to the first row of data.

usuarios (1r x 2c)		
#	senha	compara
1	03ac674216f3e15c761ee1a5e255f067953623c8b388b4459e13f978d7c846f4	a665a45920422f9d417e4867efdc4fb8a04a1f3fff1fa07e998e86f7f7a27ae3

Two red arrows point from the first row of the table result back to the SQL query. One points from the 'senha' column value to 'u.senha' in the SELECT statement. The other points from the 'compara' column value to 'SHA2('123',256)' in the SELECT statement.

```
sgc > exelmplos.sql
1  SELECT u.senha, SHA2('1234',256) AS compara
2  FROM usuarios u
3  WHERE u.id = 4
4
5
```

usuarios (1r x 2c)		
#	senha	compara
1	03ac674216f3e15c761ee1a5e255f067953623c8b388b4459e13f978d7c846f4	03ac674216f3e15c761ee1a5e255f067953623c8b388b4459e13f978d7c846f4


```
sgc > exelmplos.sql
1  SELECT CASE
2      WHEN EXISTS (
3          SELECT 1
4          FROM usuarios u
5          WHERE u.senha = SHA2('1234',256) AND u.id =4
6      ) THEN 'true'
7      ELSE 'false'
8  END AS senha_valida;
9
10
```

usuarios (1r x 1c)	
#	senha_valida
1	true

```
sgc > exelmplos.sql
1  SELECT CASE
2      WHEN EXISTS (
3          SELECT 1
4          FROM usuarios u
5          WHERE u.senha = SHA2('123',256) AND u.id =4
6      ) THEN 'true'
7      ELSE 'false'
8  END AS senha_valida;
9
10
```

usuarios (1r x 1c)	
#	senha_valida
1	false

AES

**Para utilizar criptografia do tipo AES é
necessario criar um campo do tipo
VARBINARY(100);**

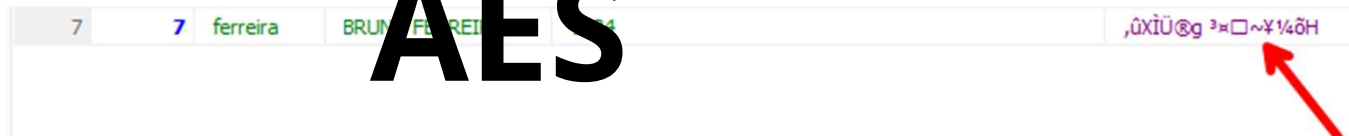
AES

```
ALTER TABLE `usuarios`  
  ADD COLUMN `senha_aes` VARBINARY(100) NULL DEFAULT NULL AFTER `senha`;
```

AES

```
sgc > exelmplos.sql
1  INSERT INTO usuarios
2      (`usuario`, `nome_usuario`, `senha_aes`, `senha`)
3  VALUES
4      ('ferreira', 'BRUNO FERREIRA', AES_ENCRYPT('1234', 'abacaxi'), 1234);
```

Utilizando AES



Decrypt AES

```
sgs > exelmplos.sql
1 SELECT id, nome_usuario, senha_aes, AES_DECRYPT(senha_aes, 'abacaxi') AS senha_decrypt
2 FROM usuarios
3 WHERE id = 7 ;
```

usuarios (1r x 4c)					
#	id	nome_usuario	senha_aes	senha_decrypt	
1	7	BRUNO FERREIRA	,ûXÏÜ@g ³¼□~¥¼õH	1234	

MD5 (Message Digest Algorithm 5):

- Produz um hash de 128 bits (ou 32 caracteres hexadecimais).
- Desenvolvido por Ronald Rivest em 1991.
- Considerado comprometido em termos de segurança devido a vulnerabilidades conhecidas, tornando-o inadequado para aplicações criptográficas modernas.
- É rápido, mas não é mais considerado seguro para criptografia.

SHA-1 (Secure Hash Algorithm 1):

- Produz um hash de 160 bits (ou 40 caracteres hexadecimais).
- Desenvolvido pela NSA (National Security Agency) dos EUA em 1993.
- Também foi considerado comprometido devido a vulnerabilidades conhecidas e é desaconselhado para uso em novos sistemas criptográficos.
- Embora ainda seja amplamente utilizado, seu uso é desencorajado em aplicações sensíveis à segurança.

SHA-2 (Secure Hash Algorithm 2):

- SHA-2 inclui várias funções hash, incluindo SHA-224, SHA-256, SHA-384 e SHA-512, que produzem hashes de 224, 256, 384 e 512 bits, respectivamente.
- Desenvolvido pela NSA e publicado pelo NIST (National Institute of Standards and Technology) dos EUA em 2001 como uma melhoria do SHA-1.
- Atualmente considerado seguro para uso geral e é amplamente utilizado em várias aplicações criptográficas, incluindo SSL/TLS, criptografia de senhas e autenticação de mensagens.

AES (Advanced Encryption Standard)

É uma cifra simétrica, o que significa que a mesma chave é usada tanto para criptografar quanto para descriptografar os dados. Ele é amplamente utilizado para proteger informações confidenciais em sistemas de computador e redes.

Afinal qual utilizar?

Ao armazenar senhas de usuários em um banco de dados, é geralmente recomendado usar um algoritmo de hash como SHA-2 (Secure Hash Algorithm 2) em vez de criptografia como AES (Advanced Encryption Standard)

Atividade



1:

Pesquise sobre bibliotecas de criptografia que podem ser utilizadas com node.js

Atividade

2:



Faca um exemplo em node.js utilizando a biblioteca escolhida na atividade 1.