

# HASHING

“

Cuando trabajamos con **datos sensibles** es fundamental almacenarlos **encriptados** para **preservar** la información en caso de que un tercero acceda a ella.



# QUÉ ES UN HASH

En informática las funciones de hasheo nos permiten **encriptar** datos. Es decir, **transformar** un texto plano en una *nueva serie* de caracteres -con una longitud fija- únicamente descifrable mediante un algoritmo complejo.

Dada la complejidad los métodos de hasheo vienen acompañados de dos características principales:

- la opción de **encriptar un dato**.
- la opción de **comparar** un dato entrante con un dato **hasheado** para verificar si coinciden o no.

El paquete `bcryptjs` nos permite incorporar estas funciones en nuestro proyecto de Node.

Para usarlo hay que instalarlo a través de npm.

```
npm install bcryptjs
```



# .hashSync()

Es el **método** que trae el paquete `bcrypt` para encriptar datos. Recibe dos parámetros:

- El **dato** que queremos encriptar.
- La **sal** que le queremos añadir a la encriptación

## ¿Qué es la sal?

Un pequeño dato añadido que hace que los hash sean significativamente más difíciles de crackear. En este contexto se le suele pasar 10 o 12.

```
const bcrypt = require('bcryptjs');  
let passEncriptada = bcrypt.hashSync('monito123', 10);
```

## .compareSync()

Es el **método** que trae el paquete `bcrypt` para **comparar** un texto plano contra un hash. `compareSync()` **retorna** un **booleano** y recibe dos parámetros:

- El primero: el **texto plano**.
- El segundo: el **hash** con el que lo queremos comparar.

```
{ }
```

```
let check = bcrypt.compareSync('monito123', passEncriptada);  
console.log(check); // true
```

# SEQUELIZE + BCRIPT

## Guardando de forma segura

Veamos un ejemplo guardando un usuario con su contraseña encriptada.

```
const bcrypt = require('bcryptjs');
const db = require('../database/models');

let passEncriptada = bcrypt.hashSync('monito123', 10);

db.Usuario.create({
  name: "Manuel",
  username: "manolito",
  password: passEncriptada
});
```

{}

# bcryptjs

## Documentación

Para saber más puedes acceder a la documentación oficial de bcryptjs haciendo click en el siguiente [Link](#)

