Desarrollo Backend con NodeJS

Hash de passwords con Bcrypt

course[It]

Bcrypt es una **función de cifrado** que se utiliza ampliamente para *hashear* o cifrar contraseñas y así evitar guardarlas en texto plano. Se encuentra disponible en diferentes implementaciones para todos los lenguajes populares, incluyendo Java, C++, Python y por supuesto... Javascript

https://en.wikipedia.org/wiki/Bcrypt

Instalamos la versión de javascript

- https://www.npmjs.com/package/bcrypt
- `npm install bcrypt`

Método hash

- El método hash recibe como parámetro nuestra contraseña en texto plano y un número de ‘saltRounds’ que indican la complejidad del cifrado

```
bcrypt.hash(data.password, 10).then(hash => {
// nuestras acciones
   });
```

Método compare

- El método compare recibe como parámetros nuestra contraseña en texto plano y en hash, realiza una comparación interna y nos avisa de forma booleana si son iguales o no

```
bcrypt.compare(password, hash);
```

Método hash - al crear (o modificar) un usuario

```
const User = require("./../models/userModel");
const bcrypt = require("bcrypt");

class UserService {
 create(data) {
   bcrypt.hash(data.password, 10).then(hash => {
     data.password = hash;
     const newUser = new User(data);

     return newUser.save();
   });
 }
}

module.exports = UserService;
```

Al loguear un usuario

- Sin bcrypt

```
    if (userData.password != password) {
      return cb(null, false);
    }
```

Método compare - Al loguear un usuario

- Con bcrypt

```
    bcrypt.hash(userData.password, 10).then(hash => {
      data.password = hash;
      bcrypt.compare(password, hash);
      return cb(null, false);
    });
```