

**Proyecto:** ACME Explorer (backend)

**Grupo:** ACME-Sandwich

**Autores:** Antonio Rodríguez Artacho, Jorge Gordo Aguilar y Rafael Fresno Aranda.

**Fecha:** 22-02-2020

Como A+ para la primera entrega de la asignatura, el 17 de febrero de 2020, se ha implementado la funcionalidad de hashear de nuevo la contraseña de un usuario ya existente, cuando éste intenta modificarla.

Previamente, si se modifica un Actor ya existente en el sistema, proporcionando una nueva contraseña, el hasheo de esta guardado en la base de datos no cambiaba. Para ello, el profesor propuso como A+ que implementáramos una forma de cambiar este comportamiento y actualizar también la contraseña del actor, hasheando la nueva y guardándola en la base de datos. Esto se ha logrado mediante el uso de un “middleware” del tipo “findOneAndUpdate” en ActorSchema, de forma que cada vez que se ejecuta dicho método para actualizar un usuario, también se ejecuta lo que ponemos dentro del middleware. En definitiva, lo que se hace es obtener la nueva contraseña escrita por el usuario, hashearla, y establecérsela al nuevo actor.

Anteriormente, este comportamiento no funcionaba debido a que la funcionalidad de hashear una contraseña se encontraba en el middleware “save” de ActorSchema, que sólo es ejecutado a la hora de crear un nuevo actor, y no en el caso de actualización.

El código que solventa el problema es el siguiente (se encuentra en el fichero ‘api/models/actorModel.js’):

```
var hashPassword = function (password) {
  return new Promise(function (resolve, reject) {
    if (password) {
      var salt = bcrypt.genSaltSync(5);
      var hash = bcrypt.hashSync(password, salt);
      resolve(hash);
    } else {
      var error = new Error('Password needed');
      reject(error);
    }
  });
};

ActorSchema.pre('save', async function (callback) {
  var actor = this;
  hashPassword(actor.password).then(function (myhash) {
    actor.password = myhash;
    callback();
  })
});
```

```
ActorSchema.pre("findOneAndUpdate", async function (callback) {  
  var actor = this;  
  var password = this.getUpdate().password;  
  hashPassword(password).then(function (myhash) {  
    actor.update({ password: myhash })  
    callback();  
  })  
});
```