

# Tarea de la sección 11 – Hospitales y Médicos

Basados en lo aprendido en la sección 10, sobre el CRUD y uso de Token de los usuarios, siga las siguientes instrucciones.

## Crear rutas, modelos y CRUD de hospitales

Opcional – Crear en MongoDB, una colección llamada hospitales. Es opcional, ya que cuando intentemos guardar desde un esquema de Mongoose, al detectar que no existe dicha colección, la creará por nosotros.

Las rutas de los servicios de hospitales, debe de ser esta:

**localhost:3000/hospital** para cualquier petición GET, PUT, POST y DELETE.

Crear el modelo **hospital** así:

```
var mongoose = require('mongoose');
var Schema = mongoose.Schema;

var hospitalSchema = new Schema({
  nombre: { type: String, required: [true, 'El nombre es necesario'] },
  img: { type: String, required: false },
  usuario: { type: Schema.Types.ObjectId, ref: 'Usuario' }
}, { collection: 'hospitales' });

module.exports = mongoose.model('Hospital', hospitalSchema);
```

### Tip:

Un tema que no vimos hasta el momento, es que el usuario es de tipo Schema.Types.ObjectId, esto es utilizado para indicarle a Mongoose, que ese campo es una relación con otra colección, y la referencia es Usuario. Al final este campo nos dirá **qué usuario creó el registro**.

Se guarda únicamente el usuario.\_id en ese campo. Es todo.

Otro tema que no vimos, es la parte de **{ collection: 'hospitales' }** esto simplemente es para evitar que Mongoose coloque el nombre a la colección como **hospitals**.

## Crear CRUD de Hospitales

Realizar todos los métodos GET, PUT, POST y Delete de la colección de Hospitales.

El GET, NO necesita el envío del token, pero las peticiones PUT, POST y DELETE deben de ser validadas por TOKEN.

No se preocupen por la paginación, eso lo haremos juntos.

## Crear rutas, modelos y CRUD de médicos

Las rutas de los servicios de médicos, debe de ser esta:

**localhost:3000/medico** para cualquier petición GET, PUT, POST y DELETE.

Crear el modelo **medico** así:

```
var mongoose = require('mongoose');
var Schema = mongoose.Schema;

var medicoSchema = new Schema({
  nombre: { type: String, required: [true, 'El nombre es necesario'] },
  img: { type: String, required: false },
  usuario: { type: Schema.Types.ObjectId, ref: 'Usuario', required: true },
  hospital: { type: Schema.Types.ObjectId, ref: 'Hospital', required: [true, 'El id hospital es un campo obligatorio'] }
});

module.exports = mongoose.model('Medico', medicoSchema);
```

### Tip:

Como pueden observar, este esquema requiere el usuario que lo creo, como un hospital... estas únicamente son los **\_id** para manejar la relación entre el usuario que lo creo y el hospital al que estamos queriendo asignar al médico.

## Crear CRUD de Médicos

Realizar todos los métodos GET, PUT, POST y Delete de la colección de médicos.

El GET, NO necesita el envío del token, pero las peticiones PUT, POST y DELETE deben de ser validadas por TOKEN.

No se preocupen por la paginación, eso lo haremos juntos.

## Nota final:

Recuerden guardar cada servicio que prueben en el Postman, ya que lo necesitaremos después en la parte de Angular.

Les pido que por favor lo intenten hacer ustedes mismos, les servirá mucho como un repaso de lo que aprendimos. Pero si ya sienten que no pueden hacerlo, no se preocupen, en las siguientes clases lo haremos juntos.

Pero sé que pueden hacerlo!  
Mucha suerte!