

Práctica 1

Desarrollo de una API REST para la gestión de una clínica de fisioterapia con Express y Mongoose

Parte 0. Estructura de la aplicación (0,25 puntos)

Crea una aplicación llamada **PhysioCare** e instala los módulos necesarios (*express* y *mongoose*). La aplicación se debe estructurar en las siguientes carpetas y archivos (además del archivo `package.json` que se generará con `npm init`):

- Archivo **index.js** en la carpeta principal del proyecto. Aquí definiremos el servidor principal Express.
- Carpeta **models** donde guardaremos los esquemas y modelos de la aplicación que se explicarán a continuación.
- Carpeta **routes** con los enrutadores para las diferentes colecciones que también se indicarán después.

Nota importante

Se utilizarán **tests automatizados** para la corrección de la práctica, por lo que es fundamental que se respete el nombre exacto de todas las propiedades especificadas en los modelos y esquemas, incluyendo el uso correcto de mayúsculas, minúsculas y la estructura de cada campo (por ejemplo: `name`, `surname`, `birthDate`, etc.). Si los nombres de las propiedades no coinciden con los definidos, los tests no podrán ejecutarse correctamente, lo que afectará el resultado de la evaluación.

Parte 1. Modelo de datos (2,25 puntos)

En esta parte vamos a crear los esquemas y modelos de datos necesarios para el sistema de gestión de una clínica médica. Utilizaremos una base de datos en MongoDB y la API que ofrece la librería Mongoose.

1. Esquemas y modelos

En la carpeta `models` iremos guardando los diferentes esquemas y modelos de nuestra base de datos. Concretamente, definiremos tres modelos principales con los campos que se indican a continuación.

Modelo Paciente (0,5 puntos)

En el archivo `models/patient.js` definiremos el esquema y modelo para la colección de pacientes. Para cada paciente recogeremos la siguiente información:

- **name:** de tipo `string`, obligatorio. Debe tener una longitud mínima de 2 caracteres y una longitud máxima de 50.

- **surname:** de tipo string, obligatorio. Debe tener una longitud mínima de 2 caracteres y una longitud máxima de 50.
- **birthDate:** de tipo fecha, obligatorio.
- **address:** de tipo string, opcional. Debe tener una longitud máxima de 100 caracteres.
- **insuranceNumber:** de tipo string, obligatorio. Debe cumplir con una expresión regular que valide exactamente 9 caracteres alfanuméricos. Además, el valor de esta propiedad debe ser único en la base de datos, garantizando que no se repita entre los pacientes registrados.

Recuerda definir el modelo asociado al esquema de pacientes para que los datos se guarden en una colección llamada **patients**, y exporta el contenido necesario para el resto de los archivos del proyecto.

Modelo Fisio (0,5 puntos)

En el archivo `models/physio.js` se registrará la información sobre los fisioterapeutas que trabajan en la clínica. Para cada uno, recogeremos la siguiente información:

- **name:** de tipo string, obligatorio. Debe tener una longitud mínima de 2 caracteres y una longitud máxima de 50.
- **surname:** de tipo string, obligatorio. Debe tener una longitud mínima de 2 caracteres y una longitud máxima de 50.
- **specialty:** de tipo string, obligatorio. Solo debe aceptar los siguientes valores predefinidos: Sports, Neurological, Pediatric, Geriatric and Oncological.
- **licenseNumber:** de tipo string, obligatorio. Debe cumplir con una expresión regular que valide exactamente 8 caracteres alfanuméricos. Además, el valor de esta propiedad debe ser único en la base de datos, garantizando que no se repita entre los pacientes registrados.

Recuerda definir el modelo asociado al esquema de fisios para que los datos se guarden en una colección llamada **physios**, y exporta el contenido necesario para el resto de los archivos del proyecto.

Modelo Expediente médico (1,25 puntos)

En el archivo `models/record.js` definiremos el esquema y modelo para la colección de expedientes médicos, donde se almacenará el historial clínico de cada paciente. Este esquema incluye subdocumentos que representan las consultas que el paciente ha tenido. Los campos del expediente serán:

- **patient:** referencia al ID del paciente, obligatorio.
- **medicalRecord:** de tipo string, opcional. Debe tener una longitud máxima de 1000 caracteres.

- **appointments:** array de subdocumentos que contendrán la información de cada consulta médica realizada al paciente.

Subdocumento Consulta

Cada consulta médica dentro del expediente incluirá la siguiente información:

- **date:** de tipo fecha, obligatorio.
- **physio:** referencia al ID del fisioterapeuta, obligatorio.
- **diagnosis:** de tipo string, obligatorio. Debe tener una longitud mínima de 10 caracteres y una longitud máxima de 500 caracteres.
- **treatment:** de tipo string, obligatorio.
- **observations:** de tipo string, opcional. Debe tener una longitud máxima de 500 caracteres.

Recuerda definir el modelo asociado al esquema de expedientes médicos para que los datos se guarden en una colección llamada **records**, y exporta el contenido necesario para el resto de los archivos del proyecto.

Carga de datos predefinidos

Se os proporcionará un `script.js` para que carguéis en la base de datos un conjunto de pacientes, fisios y expedientes con los que poder hacer una serie de pruebas homogéneas en todas las entregas.