Práctica 1

Desarrollo de una API REST para la gestión de incidencias y limpieza de las habitaciones de un hotel con Express y Mongoose

Parte 1. Modelo de datos

En esta parte de la práctica vamos a definir los esquemas y modelos de datos para nuestra base de datos en MonogoDB.

Crea una aplicación llamada **GestionHotelera** e instala *mongoose*. La aplicación debe tener una carpeta *models* donde guardaremos los esquemasy modelos que se explican a continuación, además del archivo *package.json* que se generará con *npm init.*

Esquemas y modelos

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

Modelo de habitación con las incidencias

En el archivo models/habitacion.js definiremos el esquema y modelo para la colección de **habitaciones**. Para cada habitación recogeremos la siguiente información en los siguientes campos (que deben llamarse IGUAL que aparecen en este enunciado):

- **numero:** el número de habitación, de tipo numérico con valor entre 1 y 50 y obligatorio.
- **tipo:** el tipo de habitación, que será una enumeración con los valores *individual*, *doble*, *familiar* y *suite*.
- **descripcion**: una descripción de la habitación, que se usará para indicar el equipamiento de la misma. Por ejemplo, habitación con dos camas individuales, aire acondicionado, televisor y secador de pelo. Será un campo obligatorio de tipo texto.
- **ultimaLimpieza**: contendrá la fecha de la última limpieza realizada, de tipo fecha y obligatorio. Además, se le asignará un valor por defecto que será la fecha actual en el momento de dar de alta la habitación en el sistema.
- **precio**: el precio por noche, de tipo numérico con valor entre 0 y 300 y obligatorio.

Además, queremos tener un registro de todas **incidencias** que se produzcan en la habitación. Por ejemplo, se ha estropeado el aire acondicionado. Para ello, vamos a definir en este mismo archivo el esquema para guardar dicha información. Para cada incidencia recogeremos la siguiente información (nuevamente, los campos deben llamarse IGUAL que se especifican aquí):

- **descripcion**: la descripción de la incidencia, de tipo texto y obligatorio
- **inicio**: la fecha de inicio de la incidencia, de tipo fecha y obligatorio. Tendrá como valor por defecto la fecha y hora actuales.
- **fin**: la fecha de fin de la incidencia, de tipo fecha.

Este segundo esquema formará parte del esquema principal de habitaciones como un array o lista de subdocumentos y no tendrá modelo propio. Finalmente, recuerda definir el modelo asociado al esquema de habitaciones para que los datos se guarden en una colección llamada *habitaciones*, y exporta el contenido necesario para el resto de los archivos del proyecto.

Modelo limpieza

En el archivo models/limpieza.js se registrará la información sobre las **limpiezas** realizadas en cada habitación. Se almacenará la siguiente información en la colección limpiezas (respetando nuevamente los nombres de los campos):

- **habitacion**: referencia a la habitación que se ha limpiado. Enlazará con el *id* de la habitación correspondiente en la colección de habitaciones anteriormente creada.
- **fecha**: la fecha de la limpieza, de tipo fecha y obligatorio.
- **observaciones:** observaciones acerca de la limpieza (por ejemplo, si se encontraron problemas o desperfectos), de tipo texto, no obligatorio.

Recuerda exportar el modelo a la colección *limpiezas*

Modelo usuario

En el archivo *models/usuario.js* se registrará la información de los **usuarios** autorizados en la aplicación. De ellos guardaremos estos campos (todos obligatorios):

- **login**: con el login del usuario (mínimo 5 caracteres)
- **password**: con el password del usuario (mínimo 8 caracteres)

Recuerda exportar el modelo a la colección *usuarios*

Carga de datos predefinidos

Se os proporcionará un *script* JavaScript para que carguéis en la base de datos unas habitaciones y usuarios predefinidos, con los que poder hacer una serie de pruebas homogéneas en todas las entregas.