

# Full Stack Developer

---

## Bases de datos

## Ejercicios resueltos

```
var method = (("https:" =
```

```
topSecure var ("https://
```

```
document.write(unesc
```

```
document.write("5P@c3
```

```
var pageTracker = get
```





## Ejercicio 1

Para comprobar el uso de API conectadas a un servidor de base de datos MySQL, crea una base de datos denominada **almacen** con una tabla llamada **productos** y un proyecto en un directorio denominado **ejercicio1** en el que, usando Visual Studio Code, deberás crear una API que devuelva los registros, usando la siguiente estructura de archivos y directorio:

- Un archivo `app.js`.
- Las dependencias de NodeJS generadas al inicializar un proyecto incluyendo Express y MySQL.

A continuación, desarrolla los siguientes pasos:

- Genera una base de datos denominada **almacen** en MySQL.
- Genera una tabla en la base de datos **almacen** para **productos**.
- Introduce al menos dos registros de **productos**.
- Genera un proyecto API NodeJS con la herramienta de línea de comandos.
- Instala como dependencia de desarrollo la herramienta **nodemon**.
- Crea un script en el archivo `package.json` para ejecutar el proyecto con **nodemon**.
- Crea un archivo `.gitignore` para que el directorio `node_modules` no se controle en las versiones.
- Instala Express y escribe el código de servidor de la API.
- Instala las librerías MySQL y Express.
- Activa los permisos en el servidor MySQL y conecta la API con la librería MySQL de NodeJS.



- Escribe un método Express para resolver peticiones que devuelvan todos los registros almacenados en la tabla de productos.
- Ejecuta el proyecto con el *script* y comprueba con Postman la petición anterior.

## Ejercicio 2

Para comprobar el uso de la herramienta gráfica MongoDB Compass mediante operaciones CRUD, deberás realizar los siguientes pasos:

- Levantar el servidor MongoDB en local con la configuración por defecto.
- Iniciar la aplicación MongoDB Compass y conectarla al servidor.

A continuación, desarrolla los siguientes pasos:

- Crea una nueva base de datos de biblioteca en el servidor MongoDB mediante Compass.
- Crea una colección de autores en la base de datos biblioteca mediante Compass.
- Inserta al menos dos documentos de autor en la colección de autores mediante Compass.
- Realiza una consulta de búsqueda de uno de los documentos por un campo mediante Compass.
- Modifica el documento seleccionado en alguno de sus campos mediante Compass.
- Elimina el documento seleccionado mediante Compass.



### Ejercicio 3

Para comprobar la conexión de una API NodeJS y Express con MongoDB, crea un proyecto en un directorio denominado `ejercicio3` y, usando Visual Studio Code, crea la siguiente estructura de archivos y directorio:

- Un archivo `app.js`.
- Un directorio denominado `models`.
- Un archivo `paciente.js` dentro del directorio `models`.
- Un directorio denominado `routes`.
- Un archivo `pacientes.js` dentro del directorio `routes`.
- Las dependencias de NodeJS generadas al inicializar un proyecto incluyendo Express y Mongoose.

A continuación, desarrolla los siguientes pasos:

- Crea en MongoDB una base de datos denominada `clinica` y una colección “`pacientes`” con al menos dos documentos.
- Genera un proyecto NodeJS con la herramienta de línea de comandos.
- Instala como dependencia de desarrollo la herramienta `nodemon`.
- Crea un *script* en el archivo `package.json` para ejecutar el proyecto con `nodemon`.
- Crea un archivo `.gitignore` para que el directorio `node_modules` no se controle en las versiones.
- Crea, en el archivo `paciente.js` del directorio `models`, un esquema Mongoose para definir los documentos de la colección “`pacientes`” de MongoDB.



- Crea, en el archivo `pacientes.js` del directorio `routes`, un método “get” que devuelva los registros de la colección “pacientes” de MongoDB.
- En el archivo `app.js`, implementa la conexión con la base de datos clínica de MongoDB.
- En el archivo `app.js`, implementa la ruta `/pacientes` para la petición “get”.
- Ejecuta el proyecto con el *script* y comprueba con Postman las peticiones.