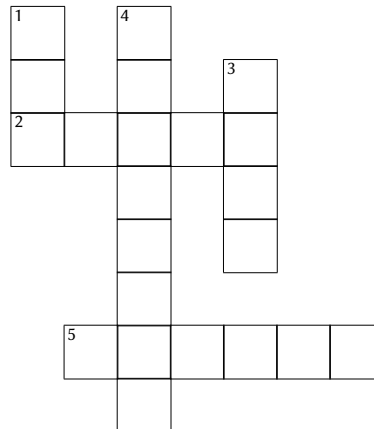


1. Completar el crucigrama.



Vertical 1 Programa que usamos para crear un proyecto en NodeJS 3 Método HTTP para crear un recurso 4 Método de un modelo para realizar búsquedas en la base de datos

Horizontal 2 Base de datos NoSQL 5 En el main de Express usamos este método al final con un número de puerto

2. Indicar los métodos HTTP que se usan para las cuatro operaciones del CRUD.

.....

3. Escribir en un sólo archivo (`index.js`) una API simple para consultar y crear documentos en una colección de países. Tiene que tener rutas para consultar todos los países y para crear un país. Es deseable tener rutas para consultar la base de datos por ID o para buscar por algún campo. Usar MongoDB. Un posible esquema para los documentos podría ser `{ id, name, capital, continent, flag }`.
 4. Escribir un *middleware* para manejar errores de “no encontrado” y pasarlo a la cadena de *middlewares*.
 5. Escribir un *router* para un recurso llamado `books`. Las rutas posibles son `/books`, `/books/search` y `/books/id`.
 6. Escribir un *middleware* para registrar las peticiones que llegan al servidor. Indicar método, ruta y *timestamp*.
 7. Escribir una API con dos colecciones: usuarios y comentarios. La API debe tener rutas para crear, consultar, modificar y borrar ambos recursos. Separar el código en rutas, controladores y modelos, cada uno en su archivo separado.
 8. Escribir modelos de Mongoose para una API sobre películas con opiniones de usuarios. Los modelos serían `Movie` y `Review`.
 9. Escribir una API para una *app* de lista de tareas. Crear modelos para las tareas y para los usuarios. Como ejercicio integrador implementar el *frontend* en React.
10. Completar los espacios en blanco.
- (a) `const User = _____ .model('User', _____)`
 - (b) `_____.route('/users/:id').delete(controller._____);`
 - (c) En Express, un _____ es una función que tiene acceso a la petición, a la _____ y al próximo _____.
 - (d) `mongoose._____('mongodb://localhost:27017/db_name').catch(_____)`
 - (e) Una ruta con parámetros se define usando _____ antes del nombre del parámetro. Se pueden leer los parámetros de la ruta en el objeto _____.

- (f) _____.route('/users/:id')._____(controller._____).
- (g) En Mongoose se puede usar el método _____ para buscar todos los documentos que cumplan cierta _____.
11. Indicar verdadero o falso.
- (a) El método `app.use()` de Express aplica *middleware* a todas las rutas globalmente. ____
 - (b) El objeto `req.body` se llena automáticamente con los datos de la petición, sin intervención de ningún *middleware* en Express. ____
 - (c) Cuando un cliente envía una entrada inválida el servidor responde con código 500. ____
 - (d) En Mongoose, al usar `save()` en un nuevo documento se realizan chequeos automáticos de validación. ____
 - (e) El método `findOneAndUpdate()` de Mongoose devuelve el documento actualizado por defecto. ____
 - (f) En Express se puede usar `app.all()` para manejar una ruta con cualquier método HTTP. ____
 - (g) En el archivo `controllers/user.js` generalmente asociamos un método HTTP y una ruta con una función de *middleware*. ____
 - (h) En Mongoose antes de crear un modelo debemos crear un *schema*. ____
 - (i) Las variables de entorno sirven para no exponer datos como las credenciales para el servidor de la base de datos. ____
 - (j) Los *handlers* en Express deben devolver una respuesta al cliente o el cliente quedaría esperando indefinidamente. ____
12. ¿Cuál de las siguientes tecnologías no pertenece al *stack* MERN?
- ☐ Express ☐ React ☐ MySQL ☐ NodeJS
13. ¿Dónde se guardan los paquetes que instalamos como dependencias de un proyecto?
- ☐ En el `package.json` ☐ En `./node_modules` ☐ En `/usr/bin` ☐ En el `.gitignore`
14. ¿Qué IP agregamos a la *whitelist* de Atlas para poder conectar nuestra app a la base de datos?
- ☐ 127.0.0.1 ☐ 0.0.0.0/0 ☐ 192.168.0.0/24 ☐ 255.255.255.0
15. ¿Qué librería se usa comunmente para conectar NodeJS con MongoDB?
- ☐ Sequelize ☐ Axios ☐ Mongoose ☐ Lodash
16. ¿Cuál es el código de estado en HTTP para indicar que un recurso fue creado?
- ☐ 200 ☐ 204 ☐ 201 ☐ 400
17. ¿Qué método HTTP se usa para actualizar un recurso?
- ☐ GET ☐ PUT ☐ POST ☐ DELETE
18. ¿Qué objeto podemos usar en Express para acceder a un ID en la ruta `/products/:id`?
- ☐ `req.params.id` ☐ `req.query.id` ☐ `req.body.id` ☐ `req.fields.id`
19. ¿Qué método de Mongoose asociarían con una petición HTTP de tipo PUT?
- ☐ `find()` ☐ `findByIdAndUpdate()` ☐ `findById()` ☐ `findByIdAndDelete()`
20. ¿Qué *middleware* usamos en Express para leer los datos en el cuerpo de la petición?
- ☐ `mongoose.connect()` ☐ `express.json()` ☐ `morgan('dev')` ☐ `express.urlencoded()`
21. ¿Qué método se usa en Mongoose para eliminar un documento por ID?
- ☐ `deleteOne()` ☐ `removeById()` ☐ `findByIdAndDelete()` ☐ `findByIdAndRemove()`

22. ¿Cómo indicar en Mongoose que un atributo de un modelo no puede ser nulo?
- ☐ { type: String, validate: true }
 - ☐ { type: String, required: true }
 - ☐ { type: String, default: '' }
 - ☐ { type: String, exists: true }
23. ¿Qué *middleware* de Express puede usarse para manejar errores 404 para las rutas no definidas?
- ☐ app.use((req, res) => res.sendStatus(404));
 - ☐ app.use((req, res, next) => res.status(404));
 - ☐ app.get('*', (req, res) => res.sendStatus(404));
 - ☐ app.use((req, res, next) => res.status(404).json({ error: 'Not Found' }));
24. ¿Qué es lo que hace el método `populate()` en Mongoose?
- ☐ Elimina un documento relacionado.
 - ☐ Carga documentos referenciados en los campos de un documento.
 - ☐ Actualiza múltiples campos de un documento al mismo tiempo.
 - ☐ Valida las relaciones entre esquemas.
25. ¿Qué método usamos para crear un nuevo recurso?
- ☐ GET ☐ PUT ☐ POST ☐ DELETE
26. ¿Qué significa la sigla CRUD?
- ☐ *Create, Read, Update, Delete*
 - ☐ *Copy, Read, Update, Drop*
 - ☐ *Create, Retrieve, Upload, Delete*
 - ☐ *Copy, Retrieve, Upload, Download*
27. ¿Qué código HTTP se usa para indicar que un recurso fue eliminado con éxito?
- ☐ 200 ☐ 204 ☐ 400 ☐ 500
28. Si estamos desarrollando una API tipo REST. ¿Qué ruta es la más adecuada para buscar al usuario con ID 10?
- ☐ /fetchUser/10 ☐ /users?id=10 ☐ /getUser/10 ☐ /users/10
29. ¿Cuál es el valor del *header* Content-Type en una petición que envía JSON en el cuerpo?
- ☐ application/x-www-form-urlencoded
 - ☐ multipart/form-data
 - ☐ text/plain
 - ☐ application/json
30. ¿Qué formato usa una API REST para enviar información al cliente?
- ☐ JSON ☐ JavaScript ☐ HTTP ☐ HTML