1.

¿Para qué se utilizan las peticiones HTTP?

Para comunicarse con servidores en internet

2.

¿Qué método HTTP se usaría para cambiar la foto de un usuario?

PATCH

3.

¿Cómo sabemos qué tipo de contenido hay en el cuerpo de una petición HTTP?

Por la cabecera Content-type

4.

¿Qué línea de código tenemos que añadir para que el servidor escuche en el puerto 3000?

app.listen(3000)

5.

¿Cómo le indicamos a nuestra app que use el router?

express.Router(app)

REPASAR CLASE

6.

En una petición hacia /user?uid=123, ¿cómo recogemos el valor de "uid"?

req.query.uid

7.

¿Cómo recogemos la cabecera Authorization de la petición?

req.headers['Authorization']

8.

¿Se puede enviar sólo un código de estado 200 en la response?

Si, con res.status(200).send()

9.

¿Cuál es la principal ventaja de responder las peticiones de manera coherente y uniforme?

Facilita trabajar con nuestra API desde cualquier cliente

10.

¿Se pueden servir los archivos de una carpeta con nuestra app de express?

Sí, usando el middleware express.static

11.

¿Es una buena práctica devolver la traza del error y una descripción detallada?

No, es un riesgo para la seguridad, ya que le estás dando información de tu código al atacante

12.

¿Es una buena práctica separar las rutas, la funcionalidad y el almacenamiento?

Sí, y te permite separar la lógica de la interfaz, y abstraer tu código de la base de datos que uses

13.

¿Debería la capa de red interactuar con la base de datos?

No, solamente con el controlador

14.

En un controlador, ¿puede haber funciones internas?

Sí, en cualquier parte puede haber funciones internas que no se expongan

15.

¿Quién es el encargado de llamar a la capa de datos?

El controlador

16.

¿Cuántas entidades podemos tener en nuestra aplicación?

No hay límite

17.

¿Las bases de datos no relacionales pueden consultarse con SQL?

Sí, SQL es un lenguaje universal

REPASAR CLASE

18.

¿Hay que crear manualmente un ID cada vez que añadamos un registro a MongoDB?

No, lo hace automáticamente

19.

¿Se puede eliminar un registro de Mongo?

Sí, con el método remove

20.

¿De qué formas podemos conectar nuestra API con la base de datos?

Conexiones puntuales que se abren y se cierran, o una conexión permanente

21.

¿Se pueden enviar ficheros a nuestra API, y recibirlos desde el servidor?

Sí, usando el módulo multer

22.

Una vez subido un archivo, ¿se puede hacer algo más con él?

Sí, se puede mover a una carpeta de estáticos, subir al cloud, o generar miniaturas

23.

¿Cuál es la principal ventaja de los WebSockets sobre las peticiones HTTP?

Los WebSockets se quedan abiertos y permiten una comunicación en tiempo real.

24.

¿Se puede recibir más de un evento desde una conexión WebScoket?

Sí, se pueden recibir tantos eventos como el servidor envíe

25.

¿Se puede mandar más de un evento a la misma conexión?

Sí, se pueden enviar tantos eventos como se desee

26.

¿Podemos tener en la misma aplicación HTTP y WebSockets?

Sí, se pueden tener en el mismo servidor, y mandar eventos cuando surjan acciones