**¿Qué es un middleware?**

Es el software que brinda servicios y funciones comunes a las aplicaciones, además de lo que ofrece el sistema operativo. Generalmente, se encarga de la gestión de los datos, los servicios de aplicaciones, la mensajería, la autenticación y la gestión de las API.

Ayuda a los desarrolladores a diseñar aplicaciones con mayor eficiencia. Además, actúa como hilo conductor entre las aplicaciones, los datos y los usuarios.

**¿Cuál es el significado del término “MEAN stack”?**

Se conoce con el acrónimo de MEAN (formado por MongoDB, Express, AngularJS y NodeJS) al stack o compilación de estas tecnologías para desarrollar aplicaciones web usando como único lenguaje de programación javascript en el servidor, en el cliente y en la base de datos.

**¿Cuáles son las características/funcionalidades que Express ofrece como**

**middleware?**

Escritura de manejadores de peticiones con diferentes verbos HTTP en diferentes caminos URL (rutas).

Integración con motores de renderización de "vistas" para generar respuestas mediante la introducción de datos en plantillas.

Establecer ajustes de aplicaciones web como qué puerto usar para conectar, y la localización de las plantillas que se utilizan para renderizar la respuesta.

Añadir procesamiento de peticiones "middleware" adicional en cualquier punto dentro de la tubería de manejo de la petición.

Permite crear APIs y aplicaciones web fácilmente, provee un conjunto de características como manejo de rutas (direccionamiento), archivos estáticos, uso de motor de plantillas, integración con bases de datos, manejo de errores, middlewares entre otras.

**¿Qué es una ruta en express?**

El direccionamiento hace referencia a la determinación de cómo responde una aplicación a una solicitud de cliente en un determinado punto final, que es un URI (o una vía de acceso) y un método de solicitud HTTP específico (GET, POST, etc.).

Cada ruta puede tener una o varias funciones de manejador, que se excluyen cuando se correlaciona la ruta.

**¿Qué es un “route handler” en express y para qué se utiliza?**

Puede proporcionar varias funciones de devolución de llamada que se comportan como middleware para manejar una solicitud. La única excepción es que estas devoluciones de llamada pueden invocar next('route') para omitir el resto de las devoluciones de llamada de ruta. Puede utilizar este mecanismo para imponer condiciones previas en una ruta y, a continuación, pasar el control a las rutas posteriores si no hay motivo para continuar con la ruta actual.

**¿Cuáles son los 2 parámetros necesarios para crear una ruta para el método GET con express, ejemplo: app.get()?**

1. url o ruta
2. función con petición y respuesta

**¿Qué es lo que hace la instrucción res.send?**

Es la respuesta que brindara al recibir una petición.

**¿Cuál es el motivo por el que express es tan popular?**

Como Node.js está enfocado en aplicaciones en tiempo real y con uso intensivo de datos es una herramienta bastante demandada, que generó una gran comunidad, además de cantidad de paquetes de código abierto, muchas herramientas y soporte de diferentes proveedores, todo esto está generando un buen ambiente para Node.js ya que está demostrando ser una buena tecnología para generar aplicaciones de backend en cuanto a rendimiento. Gigantes como LinkedIn, Netflix, PayPal, Uber, Ebay entre otros expusieron resultados del uso de Node.js en sus ambientes y los resultados fueron realmente buenos.

Node.js se ha convertido en una de las herramientas más usadas para crear aplicaciones en la nube, y es usada de forma masiva en entornos como Amazon Web Services por ejemplo.

¿Cuáles son las ventajas de utilizar express?

Análisis del cuerpo de las peticiones HTTP.

Análisis de las cookies.

Determinación de las cabeceras apropiadas para las respuestas.

Parámetros URL análisis del query string.

Cabeceras automáticas en las respuestas.

Enrutado y mejor organización del código.

Uso de middlewares.

Identificación, validación sesiones.