**Calificado del Módulo 1**

**CALIFICACIÓN DEL ÚLTIMO ENVÍO**

100%

1.

Pregunta 1

**¿Qué componentes conforman la estructura básica de un proyecto web?**

**1 / 1 puntos**



**La base de datos y el browser.**



**Un servidor, el cliente y la base de datos.**



**El servidor web y la base de datos.**



**El servidor web de nodejs.**

**Correcto**

**¡Sí, correcto! Un servidor con la lógica del backend, un browser para acceso del cliente y la base de datos para persistir los datos. Las arquitecturas actuales son un poco más complejas, ya que se vinculan con muchos servicios de la nube. Año tras año van incrementando sus prestaciones, lo que permite que cada vez puedas hacer más cosas sin invertir demasiado en servidores, pagando sólo lo que necesites.**

2.

Pregunta 2

**¿Qué es una API?**

**1 / 1 puntos**



**Una librería de programación.**



**Una tecnología de servidores.**



**Una interfaz de software para interactuar con otro software.**

**Correcto**

**¡Sí, muy bien, correcto! Te permite conectarte o trabajar con componentes de software sin necesidad de conocer su implementación. Solo basta conocer su interfaz, es decir, su API.**

3.

Pregunta 3

**¿Qué es NodeJs?**

**1 / 1 puntos**



**Una librería de programación.**



**Un motor de datos que permite realizar consultas de big data.**



**Un entorno en tiempo de ejecución multiplataforma de código abierto.**



**Un lenguaje de programación.**

**Correcto**

**¡Sí, correcto! Se ocupa de la capa del servidor basado en el lenguaje de programación ECMAScript y utiliza el motor V8 de Google.**

4.

Pregunta 4

**¿Qué es NPM?**

**1 / 1 puntos**



**La herramienta de sincronización de eventos I/O.**



**Un lenguaje de programación.**



**El gestor de librerías de NodeJs.**



**Un sistema de configuración de servidores.**

**Correcto**

**¡Sí, correcto! Además permite manejar las dependencias de tu proyecto**.

5.

Pregunta 5

**¿Para qué te sirve el archivo package.json de un proyecto basado en NodeJs?**

**1 / 1 puntos**



**Para guardar la configuración de Git.**



**Para cualquier fin, siempre que esté definido en el proyecto.**



**Para guardar la configuración del proyecto y sus dependencias.**

**Correcto**

**Correcto. Además te permite definir tareas mediante sencillos scripts.**

6.

Pregunta 6

**¿Qué es Express?**

**1 / 1 puntos**



**Un conjunto de librerías de CSS muy utilizadas en la industria.**



**Una metodología de programación.**



**Un servidor web de Nodejs minimalista.**



**Un entorno de programación que compite con NodeJS.**

**Correcto**

**¡Sí, correcto! Muchas librerías se programan por encima de Express agregando módulos y extensiones.**

7.

Pregunta 7

**¿Qué es un middleware?**

**1 / 1 puntos**



**Una función que se ubica entre el request y el response.**



**Una librería de desarrollo web.**



**Una ruta a un endpoint.**

**Correcto**

**¡Sí, correcto! Tiene múltiples finalidades: chequear seguridad, cambiar una ruta, loguear, etc.**

8.

Pregunta 8

**¿Cuáles son los 3 estados principales de los archivos manejados por Git?**

**1 / 1 puntos**



**Modified, Staged , Commited.**



**Open, closed , read-only.**



**Permanente, temporal, oculto.**

**Correcto**

**Sí, muy bien! Estos 3 estados permiten gestionar los estadíos de tu desarrollo, a fin de controlar los cambios que deseas en cada etapa.**

9.

Pregunta 9

**Verdadero o falso. Las APIs en NodeJS se utilizan de forma asíncrona. Si bien pueden gestionarse síncronamente, se recomienda su uso asíncrono.**

**1 / 1 puntos**



**Falso**



**Verdadero**

**Correcto**

**¡Sí, muy bien! Usar APIs asíncronas sin bloqueos es importante en Node, ya que es un entorno de ejecución controlado por eventos de un solo hilo. "Un solo hilo" quiere decir que todas las peticiones al servidor son ejecutadas en el mismo hilo,en vez de dividirse en procesos separados.**

10.

Pregunta 10

**¿Por qué se sugiere incluir en el .gitignore el directorio node\_modules?**

**1 / 1 puntos**



**Porque nadie necesita utilizar el node\_modules.**



**Porque al estar definidas las dependencias en el package.json, el usuario puede descargarlas con el comando npm install.**



**Porque el contenido del node\_modules por defecto es suficiente.**

**Correcto**

**¡Sí, muy bien! Así se genera el mismo contenido de node\_modules.**