**1. ¿Qué es Node.js?**

Node.js es un entorno de ejecución de JavaScript que permite ejecutar código JavaScript en el servidor. Está construido sobre el motor JavaScript V8 de Google Chrome y permite a los desarrolladores usar JavaScript para la programación del lado del servidor, lo que antes solo era posible en el lado del cliente (navegadores web).

**2. ¿Cómo se crea un proyecto en Node?**

Para crear un proyecto en Node.js, sigue estos pasos:

1. **Instalar Node.js**: Descarga e instala Node.js desde [nodejs.org](https://nodejs.org/).
2. **Inicializar el proyecto**:

* mkdir nombre-del-proyecto
* cd nombre-del-proyecto
* npm init -y

**3. ¿Para qué es el package.json?**

El archivo package.json es un archivo de configuración que contiene información sobre el proyecto y sus dependencias. Incluye detalles como el nombre del proyecto, la versión, el autor, las dependencias necesarias, los scripts de ejecución, entre otros.

**4. ¿Qué es Express y para qué se utiliza?**

Express es un framework web para Node.js que facilita la creación de aplicaciones web y APIs. Proporciona una serie de herramientas y funcionalidades para manejar rutas, solicitudes y respuestas HTTP, middleware, y más.

**5. Escriba el código de un servidor básico en Express**.

const express = require('express');

const app = express();

const port = 3000;

app.get('/', (req, res) => {

res.send('¡Hola, mundo!');

});

app.listen(port, () => {

console.log(`Servidor escuchando en http://localhost:${port}`);

});

**6. ¿Qué es una API?**

Una API (Application Programming Interface) es un conjunto de definiciones y protocolos que permite a diferentes aplicaciones comunicarse entre sí. Las APIs definen la forma en que los desarrolladores pueden interactuar con una aplicación o servicio.

**7. ¿Cuáles son los métodos HTTP?**

Los métodos HTTP más comunes son:

* GET: Solicita datos del servidor.
* POST: Envía datos al servidor para crear un recurso.
* PUT: Actualiza un recurso existente en el servidor.
* DELETE: Elimina un recurso del servidor.
* PATCH: Aplica modificaciones parciales a un recurso.

**8. ¿Qué es API REST?**

REST (Representational State Transfer) es un estilo de arquitectura para diseñar APIs. Las APIs RESTful utilizan HTTP para realizar operaciones CRUD (Create, Read, Update, Delete) sobre los recursos, que son representados en formatos como JSON o XML.

**9. ¿Cómo se instala Express en un proyecto Node.js?**

* npm install express

**10. Le solicitan crear los métodos POST y PUT para una API REST. ¿Cómo testear su funcionamiento rápidamente?**

Puedes usar herramientas como Postman o curl para testear los métodos POST y PUT de una API REST. Estas herramientas permiten enviar solicitudes HTTP con datos y ver las respuestas del servidor.

**11. Le solicitan que realice una API que devuelva el promedio de temperaturas de un array en el que se encuentran almacenadas las temperaturas por día. ¿Cuál método HTTP emplearía?**

Usaría el método GET para devolver el promedio de temperaturas, ya que se trata de obtener información del servidor.

**12. Responda Falso o Verdadero:**

* JSON es un lenguaje de programación para crear APIs: **Falso**
* Express es un framework para Node.js: **Verdadero**
* JSON es un formato empleado para intercambiar datos entre lenguajes de programación: **Verdadero**

**13. La siguiente API** [**https://www.datos.gov.co/resource/sdmr-tfmf.json**](https://www.datos.gov.co/resource/sdmr-tfmf.json) **contiene datos de los Medicamentos Vitales No Disponibles en Colombia.**

a. Teste el método GET de la API. ¿Qué obtuvo, un array simple o un array de objetos?

* Al hacer una solicitud GET a la API, obtendrás un array de objetos.

b. Su jefe le solicita que realice inserciones a la API de medicamentos, si lo podría realizar. Justifique la respuesta.

* No se puede realizar inserciones a la API de medicamentos, ya que es una API pública de solo lectura (GET) y no permite operaciones de escritura (POST, PUT, DELETE).

c. Le solicitan que analice si el formato del valor de la clave cantidad\_solicitada, requiere alguna conversión para realizar cálculos en el frontend. ¿Qué respondería?

* Revisaría el formato del valor de la clave cantidad\_solicitada. Si es una cadena de texto que representa un número, se necesitaría convertirla a un tipo de dato numérico para realizar cálculos en el frontend.

d. Le solicitan listar en un HTML los valores del nombre\_comercial pero en minúsculas. ¿Qué haría, agregar código al frontend que programará o solicitaría a los proveedores de la API que conviertan los valores del campo a minúsculas?

* Agregaría código al frontend para convertir los valores de nombre\_comercial a minúsculas, ya que es una transformación sencilla y no requiere modificar la API original.