

Taller Express y otras cosas



Nombre:

José Yael Varela García

Matrícula:

A01645324

Película favorita y razón:

John Wick 4, me encantaron las escenas de acción y la historia, aunque básica supo ser finalizada de una muy impresionante manera.

El objetivo de la actividad es explorar y conocer más sobre el desarrollo web del lado del backend.

Al final de la actividad deberás hacer un PDF con este documento. El PDF deberá tener un enlace en supertarea.html

- **Ejercicio 0: El pasado**

En las semanas previas vimos la forma de consumir un API desde el cliente (el navegador). Ahora consumirás el **API desde el servidor**. Usa javascript y nodejs.

Selecciona un API que no requiera autenticación.

Cataas.

<https://cataas.com>

El servidor deberá consumir el API, procesarlo de alguna manera (ejemplo: seleccionar un dato) y posteriormente entregar ese dato (o datos) al cliente (navegador).

Sube tu archivo a tu repositorio.

Escribe aquí abajo el enlace al archivo en tu repositorio:

https://WhatsNewsYoyo.github.io/Yael_Varela_A01645324/ApiServidor.js

¿Cuál fue el principal problema para resolver este ejercicio? ¿Cuál fue el problema que tuvo una de las personas en tu equipo? Explica cómo resolvió esa persona ese problema.

El mayor error que encontré fue el buscar una librería para trabajar como lo hacía con los fetchs en el cliente, pero al utilizar node-fetch fue solucionado rápidamente. La única problemática que vi que tuvieron mis compañeros de mesa fue que les faltó añadir la línea "type": "module" en el json para poder utilizar import.

- **Ejercicio 1: Simplicidad**

¿Recuerdas cómo **instalar un módulo** en node.js? Escribe la instrucción que se usa:

En la terminal se usa npm install -nombre del módulo- para instalar un determinado módulo.

- **Ejercicio 2: Express es**

Investiga qué es express (en el contexto de desarrollo web en nodejs). Explica a un **niñ@** qué es express, cómo se usa, para qué sirve. Escribe ese texto aquí. Incluye también **las referencias** de tu investigación.

Dile a una persona de la clase (que no esté en tu mesa) que te explique qué es express y qué ventajas te podría dar su uso. Graba el audio de la explicación.

Imaginando que tienes una tienda y cada vez que un cliente entra debes saber que quiere de la tienda, dónde llevarlo (en que sección/pasillo) o que información darle. Express es un ayudante de dicha tienda que sirve al encargarse de organizar todo. Creando el acomodo de la tienda, sabiendo que responder cuando alguien te pide un producto. Usando Express todo es más sencillo ya que si se tiene un caos dentro de la tienda. Para usarlo debes contratarlo (instalar Express), abrir la tienda con su ayuda (Crear la página web usando App = Express ()), darle instrucciones al ayudante de que hacer y finalmente pones el letrero de abierto para recibir clientes (Prender el puerto para recibir visitantes a la página web).

Como una definición más técnica Express es un framework que ahorra trabajo al construir una página web. Pues te ayuda a crear rutas, manejar lo que piden los usuarios y como responderles, manteniendo todo ordenado para su correcto funcionamiento.

Kinsta. (2025, March 5). *¿Qué es Express.js? Todo lo que Debes Saber*. Kinsta®. <https://kinsta.com/es/base-de-conocimiento/que-es-express/>

Codecademy. (n.d.). *What is Express.js?* Codecademy. <https://www.codecademy.com/article/what-is-express-js>

¿Crees que realizó una buena explicación? ¿Por qué?

Audio Sebas:

<https://drive.google.com/file/d/1As5cg7w9AlQHp5WzDVgfnOZAPctqt9jj/view?usp=sharing>

Yo considero si fue buena porque explicó lo más importante que engloba a express, lo que es y su funcionamiento además mencionó con su analogía como es que express disminuye el trabajo del desarrollador.

- **Ejercicio 3: Desde un libro**

Investiga una definición en un libro (digital o físico) **de una API**. Incluye la referencia del libro.

“Una API es una forma para que dos aplicaciones de computadora se comuniquen entre sí a través de una red (predominantemente Internet) utilizando un lenguaje común que ambas entienden.”

Jacobson, D., Brail, G., Woods, D. (2011). APIs: A Strategy Guide. Estados Unidos: O'Reilly Media.

Escribe un ejemplo de un endpoint de tu proyecto.
/buscar/escuelas:id

- **Ejercicio 4: Instala express**

Escribe la instrucción que se usa para **instalar el módulo de express**:

npm install express

- **Ejercicio 5: Uso de express**

Ejecuta y explica que hace el siguiente código:

```
import express from 'express';

const app = express();

app.listen(1984, () => {
  console.log('Up and up');
});
```

Importa el módulo express, crea la aplicación con la función express (), enciende el servidor y finalmente muestra en la consola el mensaje “Up and up”.

- **Ejercicio 6: Uso de express**

Ejecuta y explica que hace el siguiente código:

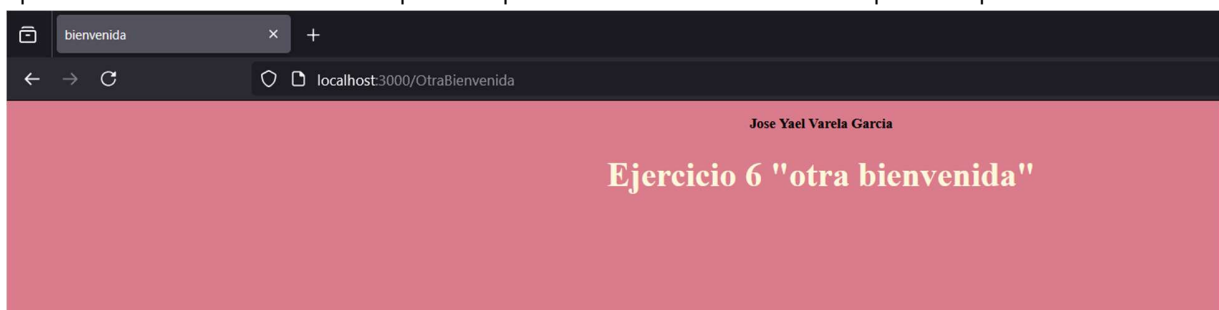
```
app.get('/bienvenida', (req, res) => {
  res.send('Esto no es una página html');
});

app.get('/otraBienvenida', (req, res) => {
  res.sendFile('bienvenida.html');
});
```

Lo que realiza el código (utilizando un puerto adicional) es que cuando se entra al localhost:puerto/bienvenida muestra un mensaje en pantalla que dice “Esto no es una página html”. Si se buscará acceder a localhost:puerto/OtraBienvenida debería mostrar el archivo bienvenida.html sin embargo marca un error indicando que el path debe ser absoluto (indicando que no hay un archivo bienvenida.html en la misma carpeta que el servidor, es decir, donde se están realizando los app.get)

En caso de que alguno de estos te marque error, soluciona el problema y explica la forma de solucionarlo:

Marca error porque no existe un archivo bienvenida.html en la misma carpeta que mi servidor, lo que requeriría la creación de un archivo estático llamado bienvenida.html y al momento de usar la app (la función de express) expresar que sea express.static y con ello servir el archivo estático creado llamado “bienvenida.html” al cual se tiene que ubicar en una carpeta “public” en el mismo path que el servidor.



https://WhatsNewsYoyo.github.io/Yael_Varela_A01645324/express.js

- **Ejercicio 7: Recuerdo de imagen**

Recuerdas que la imagen no se podía ver en <https://github.com/sgiomatec/act2025>. Un servidor entrega respuestas a solicitudes realizadas por un cliente. Cuando el cliente solicita una imagen (u otro tipo de archivo) el servidor tiene que saber responder.

Express nos propone crear un directorio para los archivos que queremos usar, por ejemplo, imágenes o archivos de hojas de estilo.

Investiga qué es express.static. Escribe tu investigación.

Escribe tu investigación

express.static es una función que se encarga de servir archivos estáticos, es decir que no cambian en el servidor, a los clientes. Como suelen ser imágenes, hojas de estilo, etc.

- **Ejercicio 8: Película**

Pregunta a una persona de otra mesa su película favorita y sus razones.

¿Ya viste esa película? ¿Te gusta? ¿La verías si no la has visto? ¿Por qué?

Interestelar, le gustó por el drama y ciencia ficción que tiene la película, no la he visto pero la vería por las buenas reseñas que he escuchado de ella.

- **Ejercicio 9: Transformación**

Usa express para transformar todo el código de servidor.js, nombra el archivo ahora servidor_express.js

Escribe aquí abajo el enlace al archivo en tu repositorio:

https://WhatsNewsYoyo.github.io/Yael_Varela_A01645324/servidor_express.js

- **Ejercicio 10: Verbos**

En el contexto de desarrollo web, explica los siguientes verbos GET, POST, PUT, DELETE

GET: Solicita u obtiene recursos o información desde el servidor.

POST: Envía datos al servidor para crear un nuevo recurso o ejecutar alguna acción que realice cambios en el sistema.

PUT: Actualiza completamente un recurso existente en el servidor, es decir se usa cuando se quiere reemplazar un recurso ya existente.

DELETE: Elimina el recurso existente especificado en la URL

Explica los siguientes códigos, en el contexto de desarrollo web:

1xx (Informativos): Indican que la solicitud ha sido recibida y el proceso continúa.

2xx (Éxito): Confirman que la solicitud fue recibida, entendida y procesada correctamente (Por ejemplo, la respuesta 200 que indica que todo OK)

3xx (Redireccionamiento): Indican que se requiere una acción adicional para completar la solicitud.

4xx (Error del cliente): Señalan que hubo un problema con la solicitud del cliente.

5xx (Error del servidor): Indica que el servidor encontró una situación que le impide cumplir con la solicitud del cliente.

Vas, V. P. (2022, May 19). *HTTP Status Codes for SEO | Best Practices (3xx, 4xx, 5xx errors)*. FandangoSEO. <https://www.fandangoseo.com/seo/http-status-codes/>

Explica con un ejemplo (relacionado con tu proyecto/reto) cómo se atendería una solicitud de cada uno de esos verbos usando express. Incluye al menos una respuesta 404, 201 y 200.

GET: Usar GET /escuelas/:id podría servir para mostrar una escuela en específico y devolvería una respuesta 200 si todo salió bien y con ello devolviendo la escuela o un 404 si no se encontró la escuela solicitada.

POST: POST /escuelas se usaría para crear una nueva escuela en la página y si todo sale bien respondería con 201 (Created), indicando que la escuela se creó correctamente.

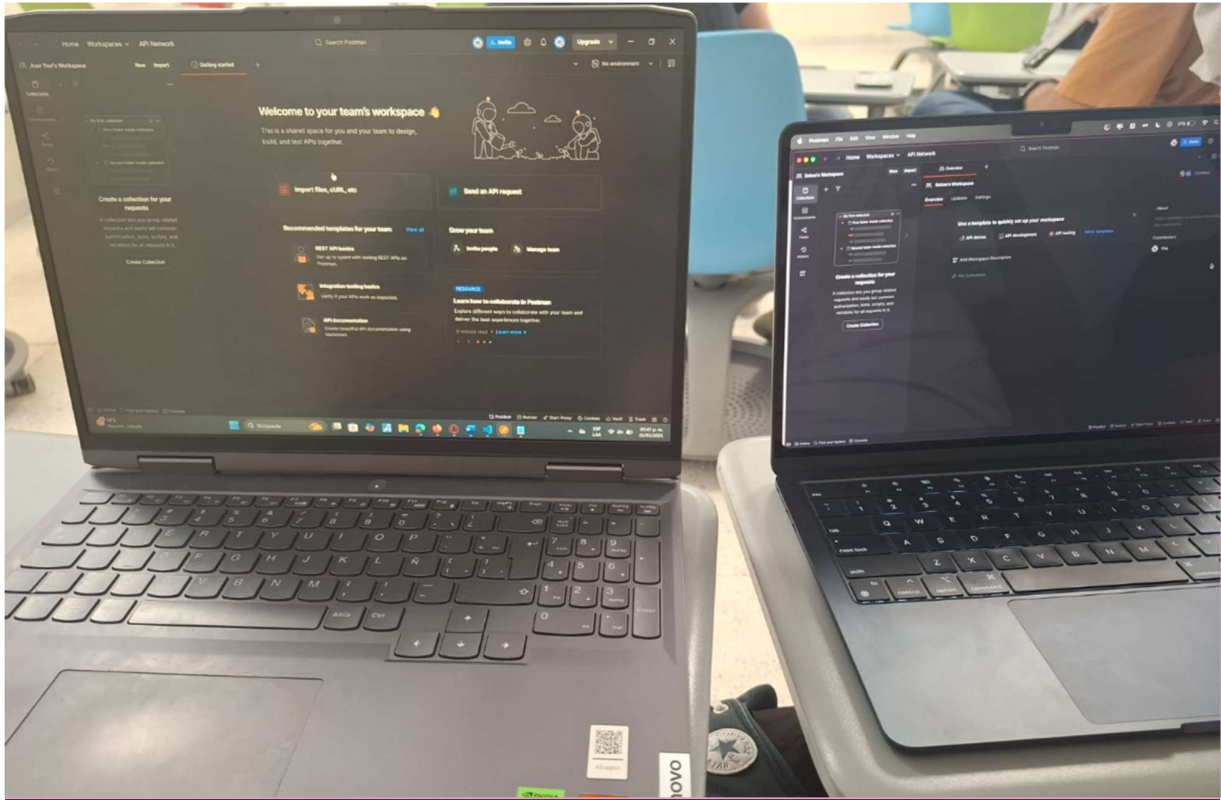
PUT: PUT /escuelas/:id para modificar los datos de una escuela en específico y remplazar todos sus datos. Si se completó se podría devolver un 200 (ok) para indicar el éxito de la operación.

DELETE: DELETE /escuelas/:id para eliminar una escuela específica y podría devolver un código 200 si la eliminación fue realizada con éxito.

- **Ejercicio 11: Postman**

En equipo con tu mesa, instalen postman en cada una de sus computadoras. Tomen una misma foto en la que aparezcan todas las pantallas con postman abierto.

Foto aquí:

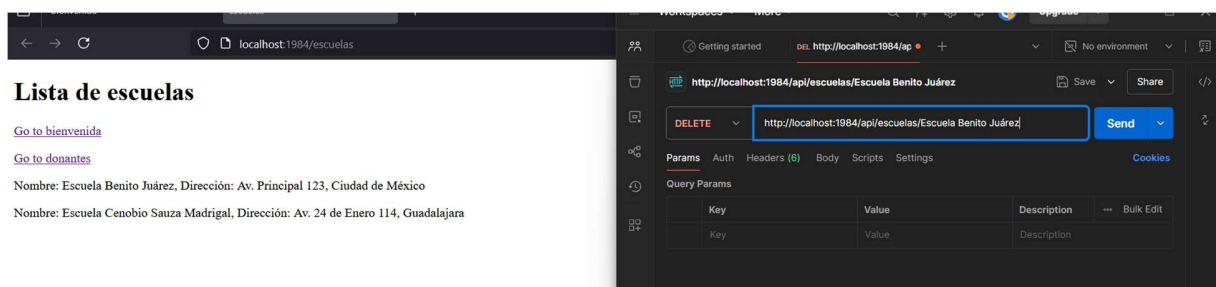
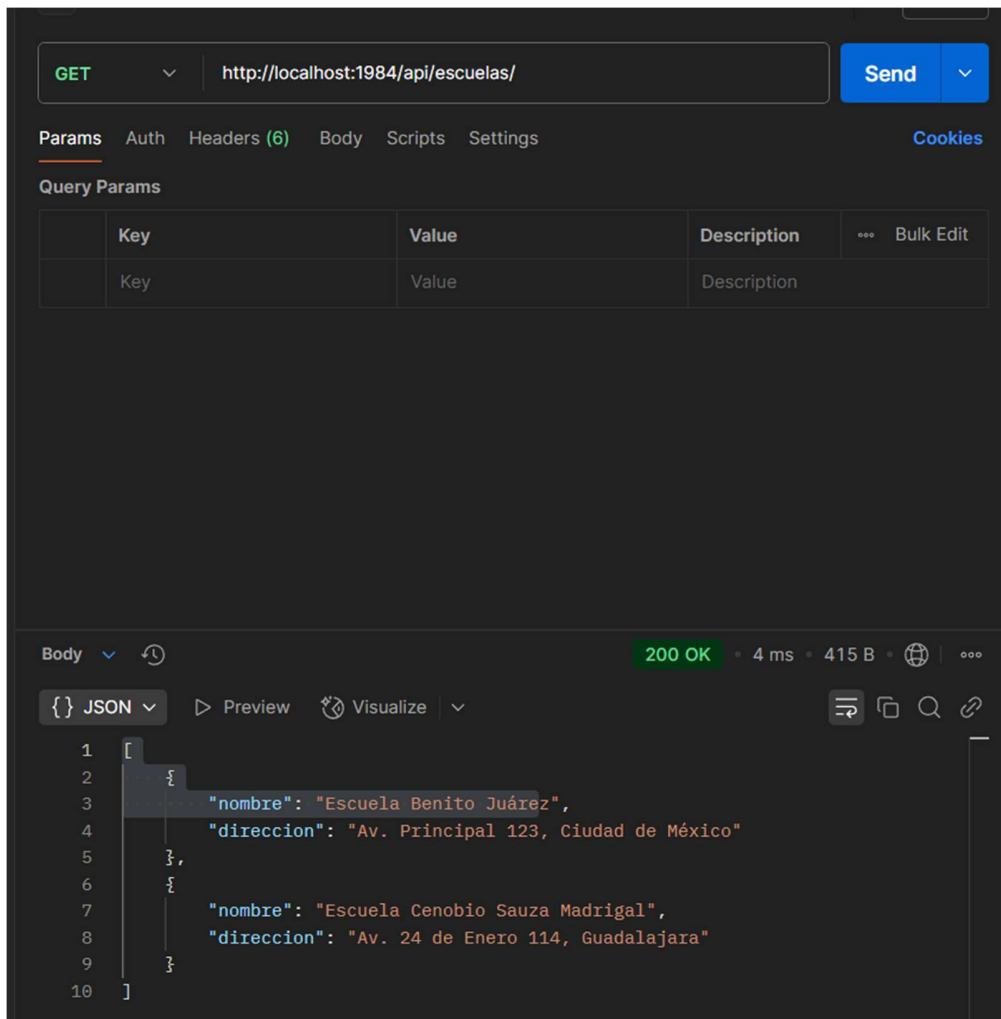


Explica para qué sirve Postman

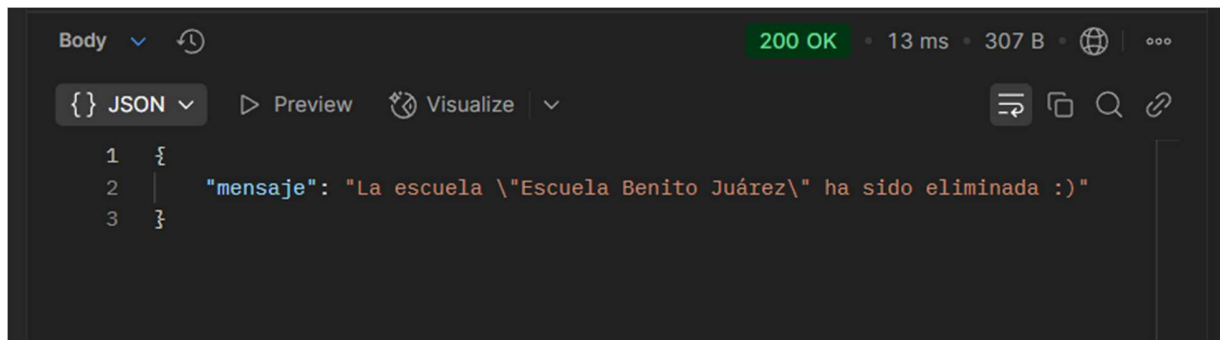
Postman es una herramienta que sirve para probar y trabajar con APIs. Permite verificar su funcionamiento al enviar solicitudes HTTP y revisar de manera detallada las respuestas que el servidor devuelve. Es esencial para agilizar la verificación, la colaboración y la automatización de los servicios web.

Prueba lo que hiciste en el ejercicio 10 con Postman. Pon una captura de pantalla usando el DELETE.

Obteniendo las escuelas con GET



Aplicando DELETE



```
Body 200 OK • 13 ms • 307 B
{} JSON Preview Visualize
1 {
2   "mensaje": "La escuela \"Escuela Benito Juárez\" ha sido eliminada :)"
3 }
```

- **Ejercicio 12: Parámetros**

Inventa un ejercicio (que esté relacionado con escuelas/donantes) para que uses parámetros en las rutas (por ejemplo /getEscuelas)

El ejercicio será para obtener las escuelas de acuerdo con la parte que pertenezcan de la ciudad. Es decir, se tendrán que filtrar si son de Zapopan, Guadalajara y Tonalá.

Soluciona ese ejercicio

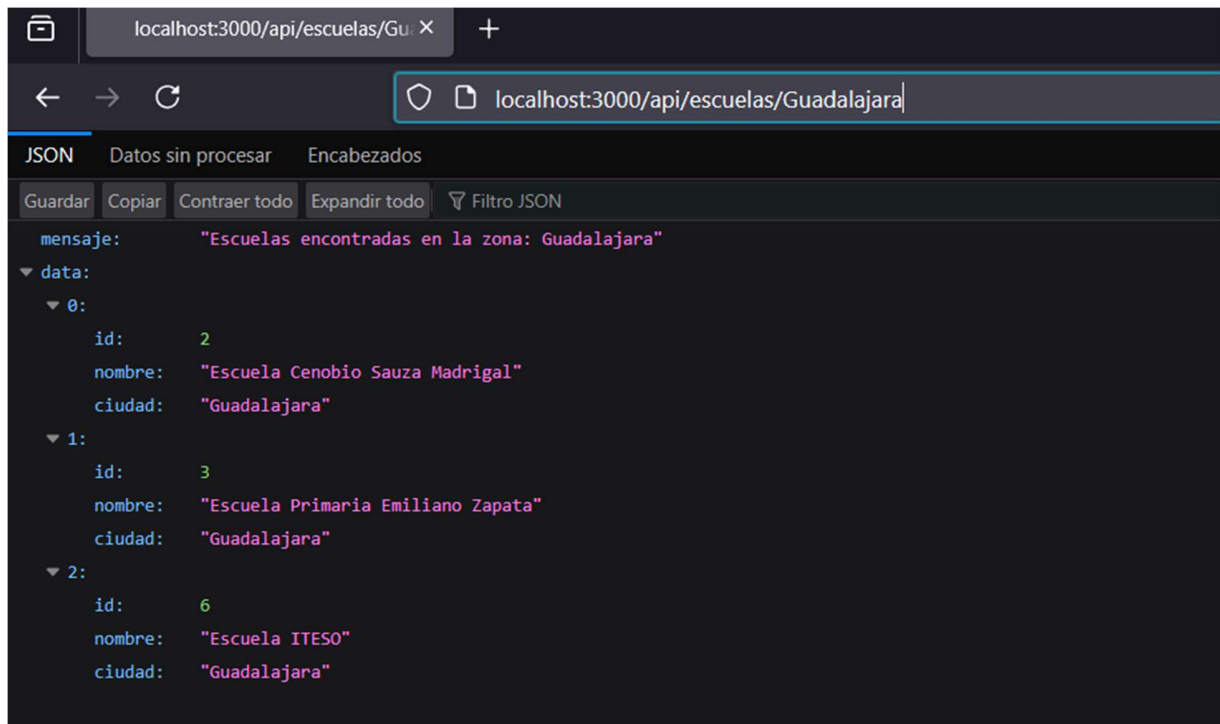
Escribe aquí abajo el enlace al archivo en tu repositorio:

https://WhatsNewsYoyo.github.io/Yael_Varela_A01645324/parametros.js

Compara ese ejercicio con el de otra persona de la clase. ¿Qué ejercicio ayuda mejor a comprender el uso de parámetros y por qué?

Considero el de mis compañeros fue mejor porque fue más detallado, mi intención era mostrar únicamente los datos (filtrarlos) por medio de un GET mientras las implementaciones de mis compañeros fueron más extensas.

```
let escuelas = [
  { id: 1, nombre: 'Escuela Benito Juárez', ciudad: 'Tonalá' },
  { id: 2, nombre: 'Escuela Cenobio Sauza Madrigal', ciudad: 'Guadalajara' },
  { id: 3, nombre: 'Escuela Primaria Emiliano Zapata', ciudad: 'Guadalajara' },
  { id: 4, nombre: 'Escuela Secundaria Revolución', ciudad: 'Tonalá' },
  { id: 5, nombre: 'Escuela ITESM', ciudad: 'Zapopan' },
  { id: 6, nombre: 'Escuela ITESO', ciudad: 'Guadalajara' },
  { id: 7, nombre: 'Escuela UPM', ciudad: 'Zapopan' },
];
```



- **Ejercicio 13: npx**

Investiga con alguien más lo siguiente: `npx express-generator`. Importante haz un directorio diferente si quieres ejecutar el comando.

Escribe el nombre de ese alguien más:

Sebastián Alejandro Soria Piñuela

Explica lo que genera y hace ese comando.

Crea la estructura base de una aplicación de Express de manera automática, primero lo que hace el comando es pedir tu confirmación sobre la instalación de determinados packages y con ello se terminaran instalando las dependencias necesarias.

Explica de manera básica y conceptualmente qué archivos se generaron.

Genera una estructura de carpetas, un archivo `app.js` y un archivo `package.json` con los scripts de inicio (start) y sus respectivas dependencias.

- **Ejercicio 14: Explicar**

Explica a una niña o un niño de 10 años para qué sirven frameworks como Angular, React, Vue, o Svelte.

Imaginando un LEGO, si no se utilizarán frameworks sería como tener que cortar y pegar cada ladrillo manualmente, es decir, crearlos a mano. Pero con un framework es utilizar piezas prefabricadas para solo tener que ensamblarlas, lo que reduce enormemente el trabajo. Estos frameworks hacen que las piezas se peguen solas, que tengan instrucciones paso a paso para reducir la complejidad y con ello se disminuyen los errores.

Son herramientas que se utilizan para crear la parte visual de las aplicaciones web que ayudan a organizar y reutilizar código y facilitar la creación de interfaces y la interacción con los usuarios.