

## Taller Express y otras cosas



Nombre: Angel Gabriel Camacho Pérez

Matrícula: A01743075

Película favorita y razón:

Una de mis favoritas es “Spider-Man into the spider-verse” ya que toma a mi personaje favorito y presenta su esencia de una grandiosa manera, mostrando valores, sentires e historias que le forman de una forma nueva con personajes no antes vistos en el cine de la mano también de un gran arte en la animación y en la música.

**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.

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:

¿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 principal problema fue comprender cómo manejar los llamados a las rutas, mandar los JSON y deconstruirlos correctamente para mostrar la información desglosada en el HTML.

### • Ejercicio 1: Simplicidad

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

Se debe escribir "npm install " y luego el nombre del 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.

¿Te ha pasado que quieres armar un lego y llega a tener muchas partes chiquitas y termina siendo muy difícil de armar? Bueno, imagina que express es

alguien que arma las piezas chiquitas en unas más grandes y te las pasa para que tú armes las cosas ahora más fácil. No te quita todo el trabajo, solo te lo facilita para que no pierdas tanto tiempo en hacer las cosas tan detalladas.

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

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.

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

Creo que se confundió un poco en el qué es, pero tiene una buena idea básica de lo que proporciona express y las ventajas de usarlo. Considero que si fue buena.

### • Ejercicio 3: Desde un libro

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

“The charm of APIs is that they are simple, clean, clear and approachable... APIs operate under the hood and are only directly called by other applications. APIs are used for machine to machine communication and for the integration of two or more software systems.” (Biehl, 2015)

Biehl, M. (2015). API architecture (Vol. 2). API-University Press.

Escribe un ejemplo de un endpoint de tu proyecto.

GET /convenio/{id}

Descripción: Obtiene al convenio que cumpla con ese id.

Mensaje de confirmación: Retorna la lista de los atributos publicos del convenio correspondiente al 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');
});
```

Levanta un servidor en el puerto 1984 y manda el mensaje “Up and up” en consola refiriendo que salió bien.

### • 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');
});
```

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

### • 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

Es un middleware de express que permite utilizar archivos como imágenes, CSS, JavaScript, etc. almacenarlos en una carpeta a la que se tendría acceso el navegador.

### • 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?

El silencio de los inocentes, le gusta porque es un trailer de terror que explora los miedos y la inseguridad con un buen guión y buenos personajes. No la he visto, no soy muy fan de las películas de terror entonces no creo que la vería.

### • 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://github.com/AngelGabrielCP5TEC/AngelGabrielCP5TEC.github.io/blob/main/node/sem4\\_p2/act2025-main/servidor\\_express.js](https://github.com/AngelGabrielCP5TEC/AngelGabrielCP5TEC.github.io/blob/main/node/sem4_p2/act2025-main/servidor_express.js)

### • Ejercicio 10: Verbos

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

-

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

1xx

2xx

3xx

4xx

5xx

GET, POST, PUT, DELETE son acciones del protocolo HTTP que permiten conocer cómo la instrucción actuará con la API.

- GET: Obtiene datos del servidor.
- POST: Envía datos que se guardan en el servidor.
- PUT: Actualiza datos ya existentes en el servidor.
- DELETE: Elimina datos del servidor.

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 /escuela/{id}

404: Objeto no encontrado.

200: Operación exitosa (objeto escuela obtenido).

500: Error en el servidor.

POST /escuela/{datos de la escuela}

404: Error al crear el objeto.

200: Operación exitosa (crear registro de escuela).

201: Objeto creado correctamente

500: Error en el servidor.

PUT /escuela/{id}/{datos de la escuela}

404: Objeto no encontrado.

200: Operación exitosa (actualizar datos de la escuela).

500: Error en el servidor.

DELETE /escuela/{id}

404: Objeto no encontrado.

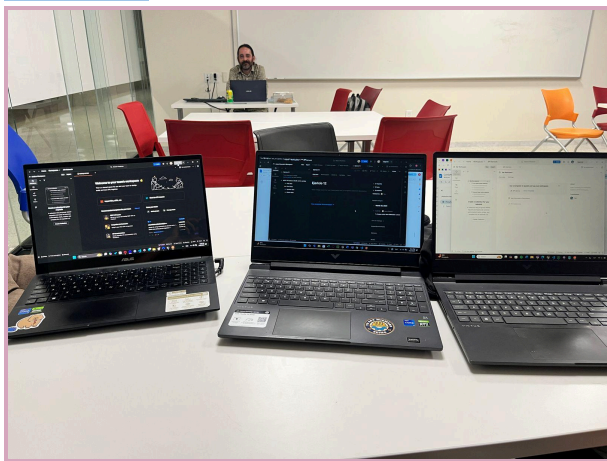
200: Operación exitosa (borrar escuela).

500: Error en el servidor.

### • 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.

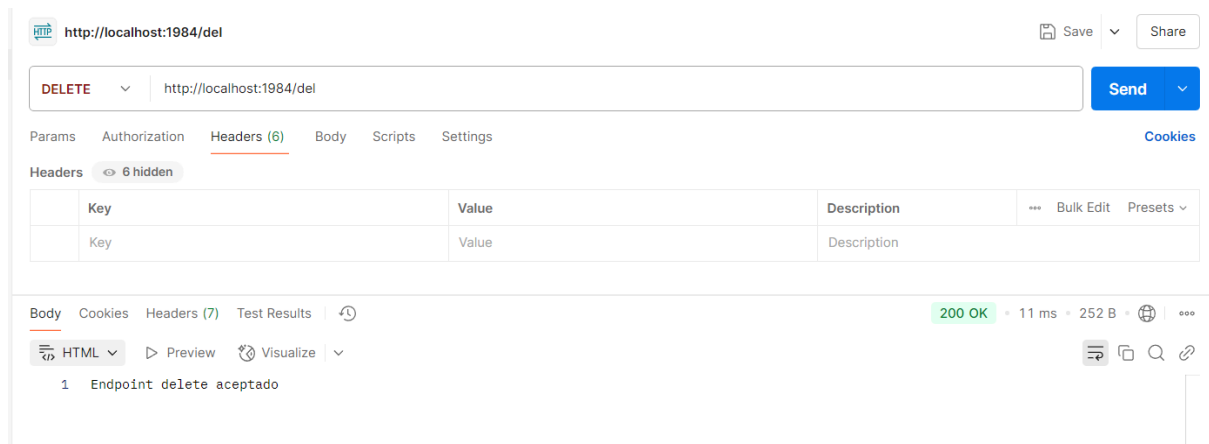
Foto aquí:



Explica para qué sirve Postman

Permite probar y desarrollar una API de manera eficiente sin necesidad del frontend.

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



## • Ejercicio 12: Parámetros

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

Soluciona ese ejercicio

Para mi aplicación web necesito que los donantes sean capaces de visualizar las escuelas registradas y su información para poder seleccionar la escuela a la que querrán donar.

Mediante el endpoint GET /api/escuelas puedo obtener un objeto JSON que contiene una lista de escuelas con su información como nombre y dirección. Será cuestión de mostrar esta información en el frontend de la manera deseada.

Otro caso sería, una escuela la cual quiere borrar su cuenta de la aplicación web. Esta escuela entra a la parte de configuración de cuenta y presiona borrar cuenta.

Mediante el endpoint DELETE /api/escuela/del/{nombre} la aplicación obtiene el nombre de la escuela con que se registró y manda al endpoint este nombre como parámetro en el url para borrar la información de dicha escuela del sistema.

Escribe aquí abajo el enlace al archivo en tu repositorio:

[https://github.com/AngelGabrielCP5TEC/AngelGabrielCP5TEC.github.io/blob/main/node/sem4\\_p2/act2025-main/servidor\\_express.js](https://github.com/AngelGabrielCP5TEC/AngelGabrielCP5TEC.github.io/blob/main/node/sem4_p2/act2025-main/servidor_express.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 que el ejercicio en el que utilicé el endpoint DELETE `/api/escuela/del/{nombre}` da un buen ejemplo de cómo nuestra aplicación utilizará los endpoints automáticamente restringiendo solo a lo que el usuario hace y la información que el usuario conoce ya que no debería conocer la información de los demás ni poder modificar la de otros usuarios.

### • 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:

Emilio Flores

Explica lo que genera y hace ese comando.

Genera un conjunto de carpetas y archivos que funcionan como una estructura para desarrollar aplicaciones con express. Contiene desde paquetes, rutas, controladores, imágenes, archivos de configuración, etc.

Explica de manera básica y conceptualmente qué archivos se generaron.

Están archivos como `app.js` que es el principal de la aplicación que permite configurar el servidor, el archivo `package.json` que contiene metadatos del proyecto para tener buen control de versión y dependencias, una carpeta de rutas para la aplicación, una carpeta para la vista de la aplicación en el navegador, una carpeta para contener imágenes, el `css`, `javascripts` usados en frontend, y luego está una carpeta que contiene las dependencias de la aplicación.

### • Ejercicio 14: Explicar

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



Los frameworks son como una plantilla sobre la que puedes trabajar. Así como si tuvieras que hacer un dibujo a color con sombras que se vea muy bien, y el framework sería como un libro de colorear que tiene las líneas del dibujo ya marcado y te dice de donde viene la luz, entonces solo tienes que pintar sin salirse de las líneas y sabiendo de donde viene la luz sabrías más o menos donde poner las sombras. Es como si te hicieran las bases para que tu puedas trabajar más sencillamente a tu modo y aun necesitando saber un poco de como funcionan la luz y las sombras.