

Taller Express y otras cosas



Nombre: José Pedro Gastélum Beltrán

Matrícula: A00227608

Película favorita y razón:
Interstellar
Banda Sonora, trama, ambientación.

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:

<https://github.com/A00227608/TallerExpress2025/tree/main/ejercicio0>

¿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 encontrar los headers y permisos necesarios para que pudiera recibir la información.

- **Ejercicio 1: Simplicidad**

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

npm install

- **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.

Express es un herramienta para hacer aplicaciones web de forma más fácil.

[¿Qué es Express.js? Todo lo que Debes Saber](#)

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.

https://drive.google.com/file/d/1n1lIThz4m59Rt1QUAnOngqWT_rZSaPyp/view?usp=sharing

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

Creo que es una explicación muy general, aunque no diga específicamente en que puede servir creo que esta bien para tener express en consideración a la hora de desarrollar una aplicación web.

• **Ejercicio 3: Desde un libro**

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

Es una aplicación con la cuál nos podemos comunicar para obtener información.

Ramírez Caballero, A. (2022). Aprende a trabajar con web APIs (1.2).

<https://cosasdedevelopers.com/static/dist/files/guia-para-aprender-a-trabajar-con-apis-versión-1.pdf>

Escribe un ejemplo de un endpoint de tu proyecto.

'api/usuarios'

• **Ejercicio 4: Instala express**

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

Unset

```
$ npm install express --save
```

• 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');
});
```

Crea un servidor express escuchando en el puerto 1984 e imprime 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');
});
```

Creamos métodos GET para manejar solicitudes al servidor mediante URLs,

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

Para solucionarlo necesitamos darle una ruta absoluta del archivo, para esto podemos usar librerías como path y url

• 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 facilita el manejo de archivos estáticos como imágenes, html,css, etc.

Para usarlo se utiliza app.use(express.static("dirname")) de esta manera evitamos tener que poner la ruta específica de cada archivo.

• 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?

The batman no me gustó, se me hacía muy oscura y apenas podía ver algo.

• 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/A00227608/TallerExpress2025/blob/main/ejercicio9/servidor.js>

• Ejercicio 10: Verbos

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

Post envía datos al servidor

Get solicita datos al servidor

Put, crear o reemplaza un elemento, a diferencia de post, utilizarlo sucesivamente tiene el mismo efecto,

Delete, solicita que se elimine un elemento.

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

1xx : indica que una solicitud esta en proceso

2xx: Son codigos de éxito, significan que la solicitud fue procesada correctamente

3xx : Indican redirección, es decir, le dicen al cliente que tiene que buscar en otro lado pero que sí existe.

4xx : Codigos de error por parte del cliente, ejemplo, not found, bad request, unauthorized

5xx: codigos de error por parte 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.

102: EL usuario escribe su usuario y contraseña, da click en iniciar sesión y recibe código 102, es decir que la solicitud está siendo procesada.

201: Se crea un nuevo usuario y el servidor regresa un código 201 indicando que fue registrado exitosamente.

200:después de que el usuario es mandado a otra pagina con éxito y regresa un código 200

302: la página de publicar donación no esta disponible y se le redirecciona a la página principal o página de error.

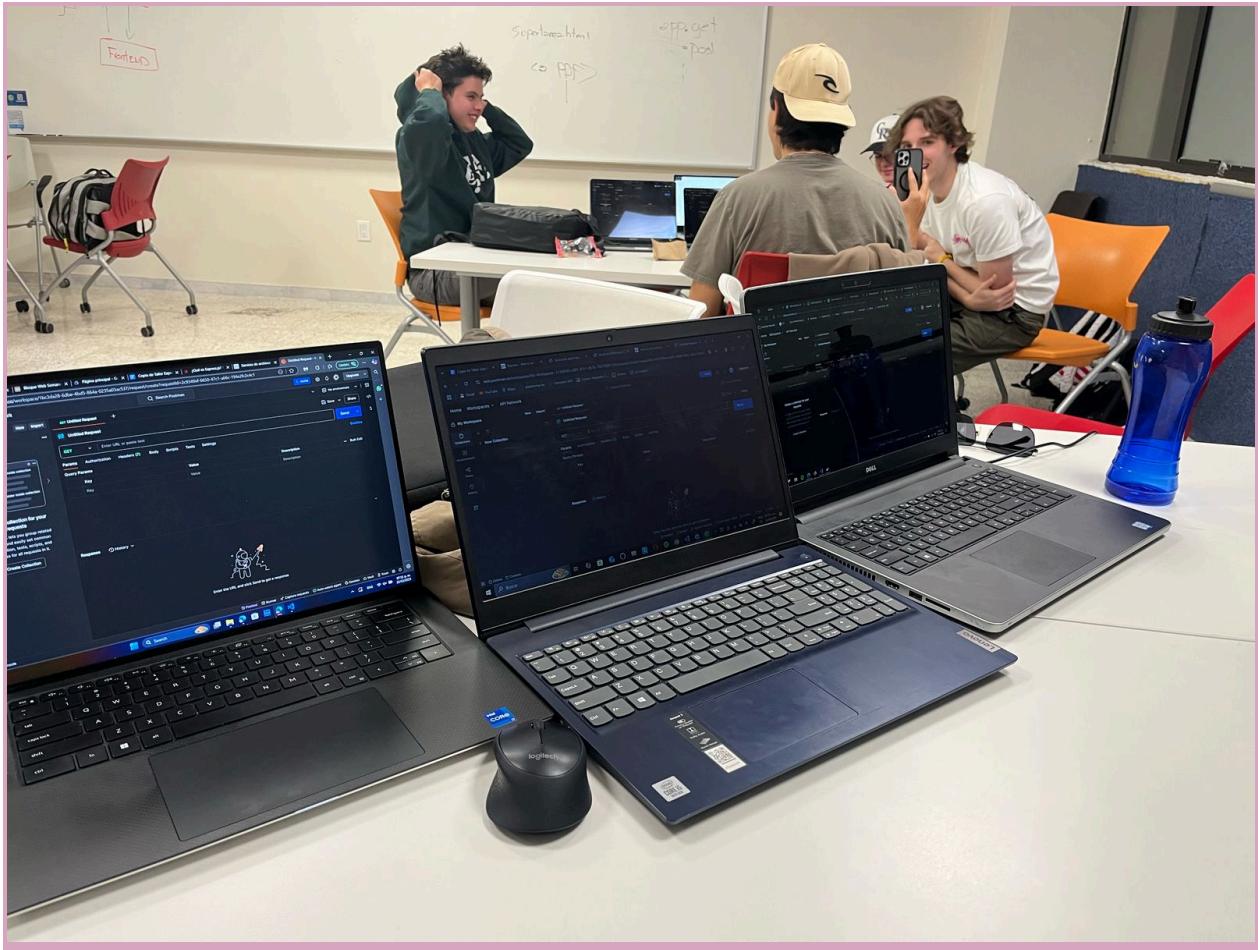
401: el cliente intenta modificar su perfil sin haber iniciado sesión, se le responde con un código de no autorizado

404: el usuario introdujo una dirección equivocada y no encontró ninguna página.

• 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 sirve para probar APIs de una forma más sencilla que creando un código, postman utiliza interfaz gráfica para facilitar llenar los parámetros de nuestra solicitud.

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

The screenshot shows a Postman interface with a DELETE request to `http://localhost:1984/api/escuelas`. The 'Headers' tab is selected, showing a Content-Type of `application/json`. The 'Body' tab contains the JSON payload: `{ "id": "1", "name": "Escuela A", "address": "Calle 1, Ciudad A", "lat": 40.7128, "lon": -74.0060 }.`. The 'Test Results' tab shows a status of 404 Not Found, time 14 ms, and size 280 B. The response body is: `1 Aqui no esta el tesoro, busca la x en el mapa`.

• 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

Escribe aquí abajo el enlace al archivo en tu repositorio:

<https://github.com/A00227608/TallerExpress2025/blob/main/ejercicio12/servidor.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?

Creo que mi ejercicio ayuda mejor a comprender el uso, ya que usamos el parámetro que se da en la url y lo utilizamos junto con el json en donde está la información de las escuelas para filtrar.

GET | http://localhost:1984/api/escuelas/Krypton323 | Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings

Code Cookies

Query Params

Key	Value
Key	Value

Body Cookies Headers (7) Test Results

Status: 200 OK Time: 16 ms Size: 289 B

Pretty Raw Preview JSON

```
1 {  
2   "nombre": "Escuela Superman",  
3   "direccion": "Krypton323"  
4 }
```

• 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:

Alejandro Gutierrez

Explica lo que genera y hace ese comando.

Este comando genera la estructura básica de una aplicación express, anteriormente cuando queríamos mostrar los archivos estáticos como html e imágenes, tuvimos que crear una carpeta en donde estuvieran guardados, este comando genera todas esas carpetas automáticamente y de forma bien estructurada, dividida por imágenes, javascripts, styleSheets, etc.

Explica de manera básica y conceptualmente qué archivos se generaron.

Se crearon archivos para definir las dependencias del proyecto, un archivo para configurar express donde se especifican las rutas y directorios con archivos estaticos(app.js), archivos con las rutas de la app, como index y usuarios , archivo que inicia el servidor.

• 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 sirven para facilitar el desarrollo de una aplicación, básicamente son como si al momento de construir una casa, tuvieras piezas ya hechas que simplemente tienes que colocar, en cambio si no usaras frameworks, tendrías que hacer todo desde 0.