

# Taller Express y otras cosas



Nombre: Patricio Blanco Rafols

Matrícula: A01642057

Película favorita y razón: Naruto The Last porque me gusta como es muy resiliente y la trama con su mejor amigo es muy buena.

**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.

<https://github.com/PatWhite29/TallerExpress>

¿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.

- **Ejercicio 1: Simplicidad**

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

- **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 conjunto de herramientas que se utiliza con **Node.js**, una plataforma que permite usar JavaScript para crear servidores y aplicaciones en la computadora. Express simplifica tareas comunes en el desarrollo web, como manejar rutas y gestionar las solicitudes y respuestas entre el usuario y el servidor.

Imagina que tienes un restaurante de hamburguesas. **Node.js** es como la cocina, donde se preparan todas las hamburguesas, pero necesitas alguien que reciba los pedidos y entregue la comida a los clientes.

Aquí es donde entra **Express**. Express es como un mesero muy rápido y organizado que toma los pedidos de los clientes, los lleva a la cocina (Node.js) y luego les entrega su comida.

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?

<https://drive.google.com/file/d/1NDFGVME7P9CVooDtPp28In-UslMorOnC/view?usp=sharing>

Si, creo que una persona que sepa muy poco de computacion podria entender a grandes rasgos esta explicacion.

### • Ejercicio 3: Desde un libro

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

Una interfaz que permite que un software interactúe con otro software de manera programática, exponiendo funciones o datos específicos para su uso externo de una manera controlada.

Jacobson, D., Brail, G., & Woods, D. (2011). APIs: A strategy guide. O'Reilly Media.

Escribe un ejemplo de un endpoint de tu proyecto.

• **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');
});
```

Este código crea un servidor utilizando **Express** en Node.js. Importa Express, luego crea una instancia de la aplicación con const app = express();. Después, configura el servidor para que escuche en el puerto 1984. Cuando el servidor se inicia correctamente, muestra el mensaje "Up and up" en la consola.

• **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 el segundo res.send no pones el path del archivo.

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

express.static es una función en Express que se usa para servir archivos estáticos, como imágenes, hojas de estilo (CSS) o archivos JavaScript, directamente desde una carpeta en tu servidor.

Cuando usas express.static, le dices a tu servidor que pueda acceder a archivos que no cambian y que los entregue a los usuarios cuando los pidan, sin tener que hacer procesamiento adicional.

### • 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, porque es un thriller criminal muy bueno y los personajes son interesantes no la he visto, nose si me guste, pero si la veré porque suena muy interesante..

### • Ejercicio 9: Transformación

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

<https://github.com/PatWhite29/TallerExpress>

### • Ejercicio 10: Verbos

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

**GET:** Se utiliza para solicitar datos de un servidor. No modifica el recurso, solo lo recupera.

**POST:** Se usa para enviar datos al servidor, generalmente para crear un nuevo recurso.

**PUT:** Se utiliza para actualizar o reemplazar un recurso existente en el servidor con nuevos datos. Si el recurso no existe, PUT puede crear uno nuevo en algunos casos, pero su propósito principal es la actualización.

**DELETE:** Sirve para eliminar un recurso del servidor.

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

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

2xx: Indican que la solicitud fue recibida, entendida y procesada correctamente.

3xx: Indican que se requiere una acción adicional para completar la solicitud

4xx: Indican que la solicitud contiene errores o el cliente no tiene permisos para acceder al recurso.

5xx: Indican que la solicitud contiene errores o el cliente no tiene permisos para acceder al recurso.

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.

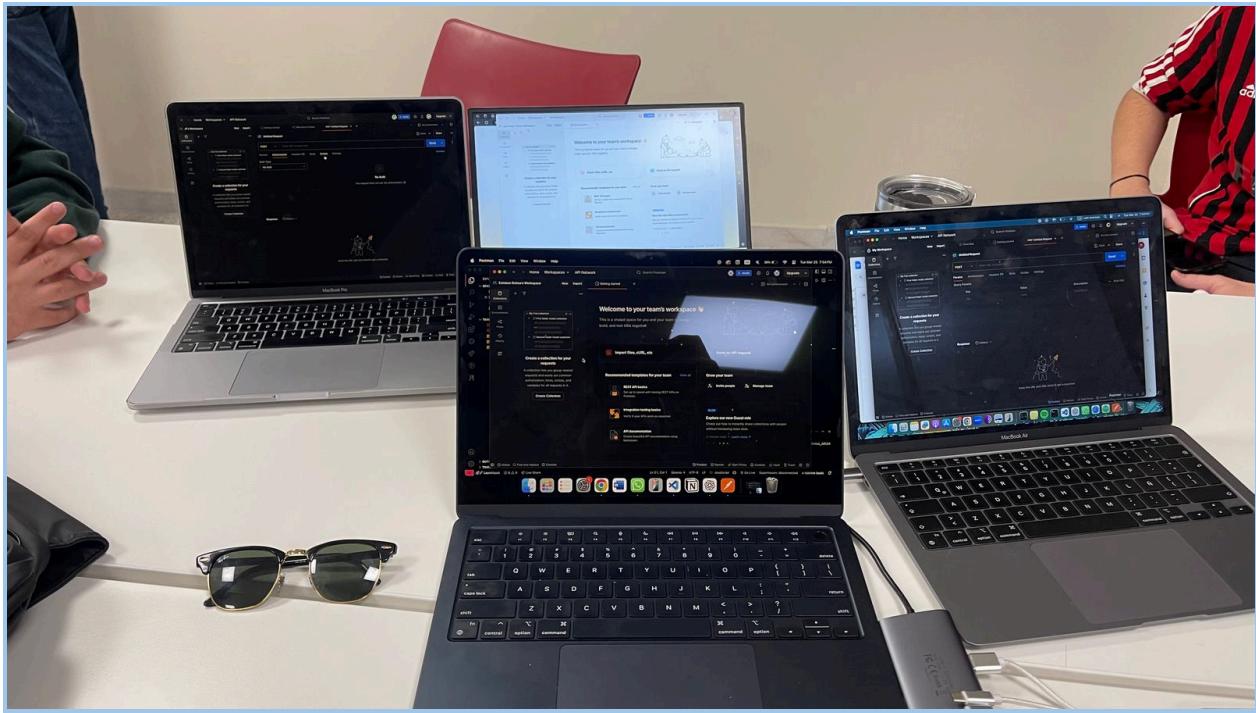
En la página web de donaciones para escuelas, un **GET** a /escuelas puede devolver una lista de escuelas con un **200 OK** o un **404** si no hay escuelas. Un **POST** a /donaciones crea una donación, respondiendo con un **201 Created** si es exitosa o un **400 Bad Request** si faltan datos. Un **PUT** en /donaciones/:id actualiza una donación, respondiendo con un **200 OK** si se actualiza correctamente o un **404** si no

se encuentra. Finalmente, un **DELETE** en `/donaciones/:id` elimina una donación, respondiendo con un **200 OK** o un **404** si no existe.

### • 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 de desarrollo que permite realizar pruebas y gestionar APIs de manera sencilla. Facilita el envío de solicitudes HTTP, como GET, POST, PUT y DELETE, y permite analizar las respuestas de los servidores.

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

Escribe aquí abajo el enlace al archivo en tu repositorio:

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

### • 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: Juan Pablo Torres

Explica lo que genera y hace ese comando.

El comando `npx express-generator` crea plantilla básica de una aplicación web usando Express.js, incluyendo los archivos y directorios necesarios para configurar el servidor, las rutas, las vistas y los archivos estáticos de la aplicación.

Explica de manera básica y conceptualmente qué archivos se generaron.

Los archivos generados incluyen el archivo principal `app.js`, el script de inicio `bin/www`, el archivo de configuración `package.json`, directorios como `public/` para archivos estáticos, `routes/` para definir rutas, y `views/` para plantillas 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.

Son conjuntos de herramientas que facilitan la creación de sitios web. Son como kits de construcción, con piezas ya listas para usar, como botones, menús y formularios.

En lugar de tener que construir todo desde cero, puedes usar estas herramientas para hacer el trabajo más rápido y con menos errores.