

Taller Express y otras cosas



Nombre: Juan Pablo Buenrostro

Matrícula: A01645981

Película favorita y razón:

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://jp-buenrostro.github.io/server.js>

- **Ejercicio 1: Simplicidad**

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

- **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 una herramienta que ayuda a construir páginas web de manera más fácil y rápida usando un programa llamado Node.js.

Node.js por sí solo puede hacer que una página funcione, pero a veces es complicado y se repite mucho código. Ahí es donde entra Express, que es como un ayudante inteligente que organiza todo y te ahorra trabajo.

Con Express puedes decirle fácilmente a tu página qué hacer cuando alguien la visita, como:

- Mostrar un mensaje
- Enviar una lista de cosas
- Guardar información

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?

Muy bien porque uso ejemplos sencillos de entender

- **Ejercicio 3: Desde un libro**

Una API (Application Programming Interface) es un conjunto de funciones y procedimientos que permiten a los desarrolladores acceder a ciertas funcionalidades o datos de una aplicación o servicio sin necesidad de conocer los detalles internos de su implementación."

Libro: "Node.js Design Patterns"

Ejemplo de endpoint en mi proyecto:

GET /getEscuelas/1

- **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');
});
```

Primero import Express

Luego crea una app con express()

Crea un servidor en el puerto 1984 con app.listen

Imprime en consola el “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');
});
```

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

Cuando visitas la pagina `http://localhost:1984/bienvenida`, el servidor te manda el mensaje “Esto no es una pagina html”

Cuando visitas la ruta de “otra bienvenida”, el servidor te manda el archivo “bienvenida.html”

- **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.

`express.static` es una función de Express que deja al servidor mostrar archivos como imágenes, hojas de estilo (CSS) directamente desde una carpeta específica, sin necesidad de crear rutas para cada archivo. Al usarla, se indica a Express que todos los archivos dentro de esa carpeta están disponibles para que el navegador los pueda ver o descargar. Esto es útil para mostrar imágenes o estilos en una página sin complicaciones.

Escribe tu investigación

- **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?

Spiderverse

Sí, ya la vi. Está bien hecha, la animación esta muy padre y la historia es buena. Si no la hubiera visto, probablemente la vería.

- **Ejercicio 9: Transformación (No hice este)**

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:

- Ejercicio 10: Verbo

```
// GET
app.get('/escuelas', (req, res) => {
  res.status(200).json({ mensaje: 'Lista de escuelas' });
});

// POST
app.post('/login', (req, res) => {
  const { email, password } = req.body;
  if (email === 'admin@ejemplo.com' && password === '1234') {
    res.status(200).json({ mensaje: 'Inicio de sesión exitoso' });
  } else {
    res.status(401).json({ error: 'Credenciales inválidas' });
  }
});

// PUT
app.put('/escuelas/:id', (req, res) => {
  const id = parseInt(req.params.id);
  const { nombre, direccion } = req.body;
  // Lógica simulada
  const escuela = escuelas.find(e => e.id === id);
  if (escuela) {
    escuela.nombre = nombre;
    escuela.direccion = direccion;
    res.status(200).json({ mensaje: 'Escuela actualizada' });
  } else {
    res.status(404).json({ error: 'Escuela no encontrada' });
  }
});
```

```
// DELETE
app.delete('/escuelas/:id', (req, res) => {
  const id = parseInt(req.params.id);
  const index = escuelas.findIndex(e => e.id === id);
  if (index !== -1) {
    escuelas.splice(index, 1);
    res.status(200).json({ mensaje: 'Escuela eliminada correctamente' });
  } else {
    res.status(404).json({ error: 'Escuela no encontrada' });
  }
});
```

- **Ejercicio 11: Postman (No supe como usar 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

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.

Esteban Ochoa

Este comando crea las carpetas y archivos básicos para empezar un proyecto con Express, sin tener que hacer todo tú. Es como un esqueleto para que empieces más rápido y ordenado.

- **Ejercicio 14: Explicar**

Frameworks como Angular, React, sirven para facilitar la creación de sitios web. En lugar de escribir todo desde cero, te dan herramientas y estructuras ya listas para organizar mejor tu código, y permitir que tu aplicación sea más dinámica y atractiva. Son una ayuda para trabajar de forma más rápida y sencilla.