

## Ejercicio Práctico.

### Ejercicio 1 - Configurando el Proyecto Base

Matías te observa con una sonrisa intrigada mientras sostiene una taza de café. Está contento de lo que has logrado hasta el momento y aún más por tu reciente incorporación al equipo.



“Bueno, esto está tomando forma, pero necesitamos establecer una base sólida para nuestro proyecto integrador”, comenta mientras señala tu laptop.



Sabrina asiente. “Crear un proyecto bien estructurado es como construir los cimientos de un edificio. Queremos que este proyecto sea el punto de partida para algo más grande.”

#### Misión:

1. Crea un nuevo proyecto con el comando `npm init -y` para generar un archivo `package.json` básico.
2. Inicializa un repositorio de Git en el proyecto con `git init`.
3. Crea un archivo `.gitignore` que excluya la carpeta `node_modules`.
4. Configura el archivo `package.json` para que soporte el estándar de imports ES Modules. Modifica el archivo añadiendo `"type": "module"`.
5. Define un script en `package.json` para ejecutar el proyecto con Node.js usando `"start": "node index.js"`.



## Ejercicio 2 - Tu Primer Servidor Web

Impresionados por tu habilidad para establecer las bases del proyecto, Matías y Sabrina deciden aumentar el desafío.

“Ahora es momento de que entres al mundo real del desarrollo web con Express,” dice Matías, colocándose frente a la pizarra. “Queremos que configures un servidor básico que pueda manejar solicitudes.”



Sabrina toma la palabra. “Recuerda, la clave está en lo simple pero funcional. Con una ruta bien configurada, demostraremos que nuestro servidor está listo para crecer.”

### Misión 2:

1. Usa la estructura creada en el ejercicio anterior para iniciar un servidor web.
2. Instala Express con `npm install express`.
3. Configura un servidor básico que corra en el puerto `3000` usando Express.
4. Agrega una ruta `/ping` que responda con el texto plano `/pong` cuando sea visitada desde un navegador.

Matías concluye: “Queremos ver un servidor funcional, código limpio y bien comentado. Este servidor será el corazón de nuestro proyecto integrador, así que pon tu mejor esfuerzo.”

## Materiales y Recursos Adicionales:

[Documentación oficial de Node.js](#) para profundizar en el uso del módulo `path`.

Documentación de [Express.js](#) para explorar sus características avanzadas.

Entornos para probar API REST, como [Postman](#) y [Insomnia](#).

---

## Preguntas para Reflexionar:

- ¿Qué ventajas y desventajas encuentras en configurar un servidor nativo con Node.js en comparación con usar un framework como Express.js?
  - ¿Por qué es importante organizar los archivos estáticos en una carpeta como `public`, y cómo mejora esto la seguridad y accesibilidad de un servidor?
  - Al usar herramientas como Express Generator, ¿en qué casos prefieres configurar manualmente un proyecto y en cuáles optarías por su uso?
  - ¿Qué elementos considerarías al estructurar un servidor web que escale para manejar miles de usuarios simultáneamente?
- 

## Próximos Pasos:

- **Modelando una API Rest:** Nos iniciaremos en la metodología para crear nuestra primera API Rest.
- **Request y Response:** Llegó el momento de aplicar de forma práctica todo nuestro conocimiento sobre la comunicación web.
- **Capa lógica:** controlando la respuesta de nuestra aplicación.



**Buenos Aires**  
*aprende*  
Agencia de Promoción y Desarrollo

**BA** Buenos  
Aires  
Ciudad