


10.7 Clase en vivo - Profesores

 Status	Done
--	------

Unidad 10: JavaScript y Frameworks

Repaso Unidad 10

Duración: 5 minutos

Consigna: Repaso de lo visto en la Unidad 10

Actividad: Introducción a Node.js y su Instalación

Instrucciones para el Profesor:

Duración: 20 minutos

- Demuestra cómo instalar Node.js desde la página oficial, si aún no lo tienen instalado.

Duración: 10 minutos

- Muestra en vivo cómo crear y ejecutar el archivo `script.js` desde la terminal.

Posible Resolución: Ejecutar en terminal:

```
// script.js
console.log('Hola Mundo con Node.js');
```

```
node script.js
```

Actividad: Uso de NPM para Instalar Paquetes

Instrucciones para el Profesor:

Duración: 15 minutos

- Explica el propósito del archivo `package.json` que se genera con `npm init`.
- Muestra cómo instalar un paquete y dónde se almacena en el proyecto.

Posible Resolución: Resultado: Un archivo `package.json` con las dependencias listadas y una carpeta `node_modules` que contiene los paquetes instalados.

```
npm init -y  
npm install axios
```

Actividad: Implementación de una Función de Petición HTTP con Axios

Duración: 20 minutos

Instrucciones para el Profesor:

- Muestra cómo importar el paquete `axios` en un archivo JavaScript.
- Realiza en vivo una petición GET a una API y muestra los resultados en la consola.

Posible Resolución:

```
const axios = require('axios');  
  
axios.get('https://jsonplaceholder.typicode.com/posts')  
  .then(response => {  
    console.table(response.data);  
  })  
  .catch(error => {  
    console.error('Error al hacer la petición', error);  
  });
```

Find the Bug: Depuración de un Script Node.js

Instrucciones para el Profesor:

Duración: 20 minutos

- Presenta el código con los errores y guía a los estudiantes para identificar los problemas.
- Realiza en vivo la corrección de los errores y muestra cómo depurar de manera efectiva.

Posible Resolución:

```
const axios = require('axios');

axios.get('https://jsonplaceholder.typicode.com/posts') //
URL corregida
  .then(response => { // Nombre de variable corregido
    console.table(response.data); // Propiedad corregida
  })
  .catch(error => {
    console.error('Error al realizar la petición', error.message); // Mensaje detallado
  });
```

Actividad: Crear una Aplicación Simple con Node.js

Instrucciones para el Profesor:

Duración: 25 minutos

- Explica cómo utilizar `process.argv` para capturar argumentos desde la línea de comandos en Node.js.
- Demuestra en vivo cómo construir y ejecutar esta aplicación básica.

Posible Resolución:

```

const axios = require('axios');

const userId = process.argv[2];

if (!userId) {
  console.error('Por favor, proporciona un ID de usuario');
  process.exit(1);
}

axios.get(`https://jsonplaceholder.typicode.com/posts?userId=${userId}`)
  .then(response => {
    console.table(response.data);
  })
  .catch(error => {
    console.error('Error al obtener los posts del usuario', error.message);
  });

```

Cierre, Revisión de Proyecto Final y Preguntas

Duración: 15 minutos

Consigna: Resumen de la clase repasando los puntos clave vistos durante la unidad.

- Espacio para consultas sobre el Proyecto Final.

Preclase - 30 minutos previos

Repaso de Conceptos Clave (10 minutos)

- **Introducción a Frameworks:** Explicación de qué son los frameworks, cómo ayudan a estandarizar el desarrollo y sus principales usos en aplicaciones Frontend y Backend. Revisión rápida de frameworks populares como React, Angular, Vue y Svelte.

- **NodeJS y NPM (Node Package Manager):** Introducción a NodeJS como entorno de ejecución de JavaScript fuera del navegador y cómo NPM facilita la instalación y gestión de librerías y paquetes.
- **Uso de NPM y `package.json`:** Cómo inicializar proyectos con `npm init`, gestionar dependencias y scripts de ejecución en `package.json`.
- **Concepto de SPA (Single Page Application):** Explicación de las aplicaciones de una sola página, su arquitectura y el rol de los ruteadores en la actualización dinámica del DOM sin recargar la página.

Espacio para Preguntas y Respuestas (10 minutos)

- Abrir el espacio para que los estudiantes hagan preguntas sobre los conceptos de NodeJS, NPM, y la utilización de frameworks en proyectos de Frontend.
- Resolver dudas con ejemplos en vivo si es necesario, como la instalación de un paquete con NPM o cómo configurar un proyecto básico con `npm init`.
- Si no hay preguntas, sugerir temas comunes como la diferencia entre librerías y frameworks, o cómo elegir el framework adecuado para un proyecto.

Consulta y Apoyo al Proyecto Final (10 minutos)

- **Revisión de Requisitos del Proyecto Final:** Repasar brevemente los objetivos y requisitos clave del proyecto final (uso de librerías, estructura de código, lógica de negocio, etc.).
- **Resolución de Dudas del Proyecto:** Abrir el espacio para que los estudiantes consulten dudas específicas sobre sus proyectos, como la implementación de funcionalidades, integración de librerías o corrección de errores comunes.
- **Sugerencias y Consejos Prácticos:** Proporcionar recomendaciones sobre la presentación del proyecto, el uso de herramientas de depuración (`console.log`, `alert` vs. librerías de notificaciones) y la importancia de un código limpio y bien documentado.