



Actividad: Creación de un Gestor de Tareas con Módulos Propios y Librerías Externas

Título: "Desarrollando un Gestor de Tareas en Node.js con Módulos y Librerías"

Objetivo:

Aplicar conocimientos sobre importación y exportación de módulos en Node.js (CommonJS y ES Modules), utilizando módulos propios y librerías de terceros para construir una aplicación de consola funcional.

Instrucciones:

1. Crear un módulo para manejar tareas (`gestorTareas.js`)

En este archivo, implementarás las siguientes funciones:

- `agregarTarea(tareas, titulo, descripcion):`
 - Agrega una nueva tarea a la lista de tareas (un array).
 - Cada tarea debe ser un objeto con las propiedades: `id`, `titulo`, `descripcion`, `completada` (por defecto `false`).
- `listarTareas(tareas):`
 - Muestra todas las tareas con su estado (pendiente o completada).
- `marcarCompletada(tareas, id):`
 - Marca una tarea como completada según su `id`.
- `eliminarTarea(tareas, id):`
 - Elimina una tarea de la lista según su `id`.

Exporta estas funciones usando **CommonJS** (`module.exports`).

2. Crear un archivo principal (`app.js`)

- Importa las funciones del módulo `gestorTareas.js` usando `require`.
- Utiliza la librería [chalk](#) para colorear la salida de la consola:
 - Tareas completadas en **verde**.
 - Tareas pendientes en **amarillo**.



- Mensajes de error en **rojo**.
- Crea un array vacío `tareas` que contendrá las tareas.
- Usa la librería `readline` de Node.js para capturar la entrada del usuario.
 - Opciones disponibles:
 - 1: Agregar tarea.
 - 2: Listar tareas.
 - 3: Marcar tarea como completada.
 - 4: Eliminar tarea.
 - 0: Salir de la aplicación.

Ejemplo de interacción:

Gestor de Tareas:

1. Agregar tarea
2. Listar tareas
3. Marcar tarea como completada
4. Eliminar tarea
0. Salir

3. Crear una versión con ES Modules

- Convierte el archivo `gestorTareas.js` en `gestorTareas_esm.js`, utilizando **ES Modules** (`export` y `export default`).
- Crea un archivo `app_esm.js` para ejecutar el gestor de tareas usando `import`.
- Configura el proyecto con `"type": "module"` en el archivo `package.json` o utiliza la extensión `.mjs`.

Criterios de Evaluación:

1. **Módulo personalizado (`gestorTareas.js` y `gestorTareas_esm.js`):**
 - Funciones correctamente implementadas con manejo de errores.
2. **Interacción con el usuario (`app.js` y `app_esm.js`):**
 - Menú funcional y dinámico en la consola.
 - Operaciones correctamente realizadas (agregar, listar, marcar completada, eliminar).
3. **Uso de librerías de terceros:**
 - Mensajes coloreados con `chalk` según el estado de las tareas o errores.

TALENTO TECH VALLE	
ACTIVIDAD: Creación de un Gestor de Tareas con Módulos Propios y Librerías Externas con node.js	
103 – 104	
CLASE 16	



Entrega:

Sube los siguientes archivos:

- Al repositorio de github