

Actividades – Para realizar las actividades, has de tener en cuenta:

- Usar propiedades privadas y métodos privados en las clases cuando sea necesario.
- Cuando se devuelvan valores, si son objetos y arrays de las propiedades, devolver copias de los mismo, para evitar que puedan ser modificados.
- Utiliza módulos (EMS).
- Realiza la instalación global de lite-server. Configura la misma en el archivo package.json

Como te habrás dado cuenta, lo correcto es no tener todo el código en un único fichero javascript sino cada cosa en su fichero correspondiente. Para definir la estructura de archivos es imprescindible utilizar un patrón como MVC (Modelo Vista Controlador): <https://www.natapuntos.es/patron-mvc-en-vanilla-javascript/>

Este patrón separa estas entidades:

- Modelo: La representación de los datos, lo que interactúa con la base de datos, independientemente del tipo que sea.
- Vista: Lo que ve y con lo que interactúa el usuario.
- Controlador: El que gestiona las peticiones y comunicaciones entre las entidades.

De momento, vamos a crear una estructura de archivos en la que tendremos diferentes componentes que se encargarán de parte de este patrón y que poco a poco iremos perfilando.

Puedes escoger la estructura de archivos con la que te sientas más cómodo/a, pero para mantener los proyectos lo más simple posible para facilitar la comprensión, crea un directorio llamado */src*, que contendrá todos los directorios y archivos de cada proyecto.

Actividad a realizar en tu proyecto intermodular

Sobre tu PROYECTO realiza las acciones necesarias para trabajar con módulos utilizando el patrón MVC. Utiliza todo lo aprendido para organizar tu proyecto.

Se adjuntan los ficheros y la estructura, para que sirva como guía, de cómo debería quedar organizado un proyecto.

```
tienda-mvc/  
|  
├─ index.html  
├─ index.js  
├─ models/  
|   └─ Product.js  
├─ views/  
|   ├── ProductView.js  
|   └─ CartView.js  
└─ controllers/  
    └─ ShopController.js
```