

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

- Usar propiedades privadas y métodos privados en las clases, cuando no sean necesarios.
- Cuando se devuelvan valores, si son objetos y arrays de las propiedades, devolver copias de los mismo, para evitar que puedan ser modificados.
- Incluye ejemplos de uso.
- 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.

Carrito con Productos (2 puntos)

Sobre **Carrito con Productos**, realiza la separación de las clases Producto (producto.class.js) y Carrito (carrito.class.js):

```
▼ model
  JS carrito.class.js
  JS producto.class.js
  > node_modules
  <> index.html
  JS index.js
  {} package-lock.json
  {} package.json
```

Eventos (4 puntos)

Sobre tu **Formulario de Eventos**, realiza:

- Genera la clase Eventos (eventos.class.js), con la siguiente estructura:

```
constructor({id, name, date, description, image, price})
```

- Estructura mejor el código de eventos_formularios_1.js, separando:
 - o la lógica de negocio para guardar los eventos en un Array llamado **eventData []** (events.js)
 - o la gestión de la visualización de estos datos en index.html (main.js)
- La estructura del proyecto debería quedar como mínimo así:

```
JS evento.class.js
JS events.js
JS main.js
<> index.html
{} package.json
# style.css
```

Diccionario de Rimas (4 puntos)

Sobre tu **Diccionario de Rimas** realiza las acciones necesarias para trabajar con módulos utilizando el patrón MVC. Utiliza todo lo aprendido en tu proyecto.