

# Práctica 01 – Open events

## Objetivos académicos

El fin de la práctica es que demuestres que eres capaz de resolver un problema antes no propuesto de que requiera el uso de tecnologías HTML, CSS, Node.js y NPM. En concreto se evaluarán los siguientes conocimientos teóricos y prácticos:

### HTML5

- Crear *mockups* de páginas web
- Demostrar el uso de tags semánticos
- Demostrar el uso de tablas con cabecera, cuerpo y pie
- Demostrar el uso de listas ordenadas o no ordenadas
- Aplicar *anchoring* como método de navegación entre páginas web
- Fundamentar las decisiones estructurales en los desarrollos
- Maquetar una página web en HTML5 y CSS a partir de *mockups*
- Sugerir cuáles son los tags HTML5 adecuados para el diseño
- Fundamentar las propuestas de tags HTML5 a partir de un diseño
- Aplicar la validación HTML5 W3C a los desarrollos como requisito

### CSS

- Crear el CSS a partir de *mockups*
- Demostrar el uso del hashtag , selectores, selectores anidados y custom properties
- Demostrar el uso del display y sus posibilidades
- Desarrollar sitios web que usen clases dentro de archivo externos. css
- Desarrollar sitios web *Responsive* mobile first
- Integrar *Meda Quieres* al desarrollo web *Responsive*
- Aplicar la validación CSS W3C a los desarrollos como requisito

### Nodo.js y NPM

- Demostrar el uso del servidor node.js para levantar un servidor local con servicio estático de archivos
- Demostrar capacidad de configuración de un proyecto con package.json
- Aplicar npm para la instalación de dependencias
- Utilizar scripts para tareas de automatización como la creación de distribuibles

## Aplicación de eventos

Se pide desarrollar una aplicación web social donde los usuarios puedan gestionar, buscar y participar en eventos de cualquier ámbito. Como aplicación de índole social los usuarios podrán interactuar entre sí haciéndose amigos y comunicándose dentro de la plataforma.

## Las funcionalidades

La aplicación debe contemplar:

- Sesión de usuario
  - Registro
  - Inicio
  - Cerrar
- Gestionar Perfil de usuario
  - Datos
  - Eliminar cuenta
  - Ver estadísticas: puntuación media dada, número de comentarios realizados, porcentaje de usuarios con menos comentarios que este
- Gestionar Amistades
  - Listar
  - Ver perfil
  - Enviar petición
  - Petición pendiente
- Gestión de Eventos
  - Crear
  - Listar
    - Los eventos deben mostrarse con su nombre, la fecha y la ubicación.
    - Se debe poder ver el evento en cuestión detallada al clicar sobre él.
    - Ordenar por puntuación del creador del evento (media de las puntuaciones de los diferentes eventos que éste ha creado)
  - Buscar por palabra clave del nombre del evento, ubicación y fecha de inicio.
  - Compartir
  - Participar
  - Timeline de eventos participantes

## Webservices

La aplicación, para funcionar, necesita interactuar con la base de datos, para nutrirse y por el otro almacenarlos. Estos dos requisitos se ven cubiertos gracias a la disposición de una API ( *back-end* ) que

permitirá consultar y almacenar datos haciendo que la aplicación web tenga estado. Este *back-end* estará disponible para poder desarrollar el *front-end* con tecnologías web HTML, CSS y JavaScript . Puede consultar la documentación de la OpenEvents API en eStudy .

## Evaluación

Para que la práctica pueda evaluar debe cumplir los siguientes requisitos: los archivos HTML y CSS deben validar por W3C sin errores. Se aceptan warnings . Si se cumplen los requerimientos anteriores se evaluará, teniendo en cuenta que:

### Mockup

- Se pide el *mockup* de la versión desktop y móvil. Es necesario indicar la resolución móvil.

### Nodo.js

- Tiene que haber una carpeta *src* con el código fuente de la práctica ( *mockups* , *htmls* , *css* ...)
- Debe haber una carpeta *dist* con el código maquetado distribuible y generado a partir de un script automatizado en el *package.json* con nombre " *build* ". Por tanto, la carpeta *dist* estará vacía y el pedido " *npm run build* " le nutrirá.
- La práctica debe venir acompañada de un servidor local *node.js* que publique *dist*. Deberá utilizarse " *npm run start* " .
- El *package.json* debe tener un script de desarrollo " *npm run watch* " que permita detectar cambios en archivos del *src* para crear archivos *dist* automáticamente.
- Tiene que haber carpetas que organicen los diferentes archivos ( *html* , *css* , *media* ...).

### HTML

- Codificación de caracteres UTF8.
- Indicar meta de autor, descripción y palabras clave.
- Indicar que los motores de búsqueda indexen la página pero no sigan los enlaces.
- Indicar comentarios suficientes y claros que justifiquen estructura y tags . Sólo hacerlo en dos archivos.

### CSS

- La web debe ser *Responsive Mobile first* . *Adaptar a desktop* y móvil (indicar medidas).
- Utilizar *Normalize* , *Reset* o similar que sea propuesta externa.
- Uso de *display* ( *inline-block* , *flex* o *grid* ).
- Enlazar archivos de estilos . *css* .
- Integrar *Media Queries* al desarrollo web *Responsive*.
- Uso del *hashtag* , selectores, selectores anidados y *Custom Properties*.