

U06-P01: Formularios para visualizar mapas

Requisitos

- Uso de la API para representar mapas con JS: leaflet.js
- Web con formulario Login
- Web con formulario para establecer criterios y filtrar elementos a representar
- Utilizar el [JSON siguiente](#) como origen de los datos a representar

Descripción

Web con formulario

Inicialmente se debe partir de una web de login que cuente como mínimo de los campos usuarios y contraseña. Se propone usar la herramienta Firebase de Google para su implementación.

- El campo usuario:
 - puede ser un email o un número de teléfono (con sus formatos correspondientes)
 - estar validado mediante una expresión regular acorde a si es un email o número de teléfono
- El campo contraseña:
 - Debe cumplir algunas restricciones en cuanto al número y tipos de caracteres. Se debe informar al usuario del formato a cumplir.

Una vez el usuario esté validado se accederá a la Web para buscar y mostrar elementos en el mapa

Web con filtros y mapa

Se debe disponer varios filtros (máx 3). A partir de los criterios seleccionados en el formulario, el usuario pulsará buscar y se mostrarán en un mapa ubicado en esa misma Web las marcas correspondientes a los elementos filtrados.

Entrega

- El proyecto debe tener la siguiente estructura de carpetas y ficheros:
 - U06P01_Apellido1_Nombre1_Apellido2_Nombre2 (carpeta del proyecto)
 - img (carpeta con las imágenes empleadas)
 - js (carpeta con los JS creados)
 - css (carpeta con los ficheros css creados)
 - index.html (html que estará enlazado con un único fichero JS minificado y ofuscado)
- Sube el proyecto comprimido en formato ZIP.
- El nombre del fichero debe seguir el siguiente formato:
 - U06P01_Apellido1_Nombre1_Apellido2_Nombre2.zip
- Subir el enlace a GitHub pages del proyecto creado.
- Indenta adecuadamente el código y usa los comentarios que consideres oportunos.
- Elimina las importaciones de librerías que no uses.

Calificación

- Requisitos mínimos (70%):
 - El formulario de login (con los requisitos solicitados)
 - Es implementado con Firebase: máx 2,5 pto
 - Es implementado sin Firebase: máx 1 pto
 - Por cada criterio proporcionado (hasta 3 criterios) para filtrar los elementos del mapa se valorará su:
 - creación (se crea dinámicamente mediante herramientas DOM o estáticamente),

- su funcionalidad (que funcione correctamente)
 - y la dificultad de su implementación.
- Los filtros pueden ser:
 - filtros que encadenen criterios (por ejemplo: al seleccionar una provincia, se mostrarán únicamente las islas de dicha provincia)
 - elemento tipo select con los datos del JSON
 - cajas de texto
- Cada criterio para filtrar implementado se ha de valorar independientemente y su valor es de un máximo de 3 pts.
- Se representan los elementos resultantes de aplicar los filtros mediante marcas en el mapa: máx 0,5 pts.
 - Cada una de las marcas representadas en el mapa ha de: máx 1 pto.
 - tener un icono
 - se valorará mostrar iconos que no sea el establecido por defecto
 - responder al evento click para mostrar y cerrar la ventana con información del elemento representado
- Buenas prácticas de programación (20%):
 - Sólo se valorará si se obtienen al menos 5 puntos de 7 en el apartado anterior.
 - Código mínimo imprescindible.
 - Código optimizado.
 - Comentarios necesarios y suficientes.
 - Variables y métodos con nombres significativos y autoexplicativos.
 - uso de ESLint (incluir al menos 3 capturas en las que se pueda distinguir que se usado): 0.5 pts.
 - notación camelCase, funciones arrows, métodos map, filter, reduce,...: 0.5 pts.
 - uso de JSDocs: 0.5 pts.
 - código refactorizado: 0.5 pts.
- Originalidad y creatividad (10%). Se valorarán positivamente las aportaciones creativas o innovadoras para enriquecer la aplicación:
 - Sólo se valorará si se obtienen al menos 7 puntos de 9 entre los dos apartados anteriores.
 - Se valorarán como máximo dos aportaciones.

El plagio o la incapacidad de defender la entrega se calificará con un cero.