| **Desarrollo de Aplicaciones en Entorno Cliente**  **2º Ciclo Superior de Desarrollo de Aplicaciones Web**  **2023-2024** | **barcoFomento.png** |
| --- | --- |
| **Tarea 4. jQuery. AJAX. React** |  |

**Observaciones:**

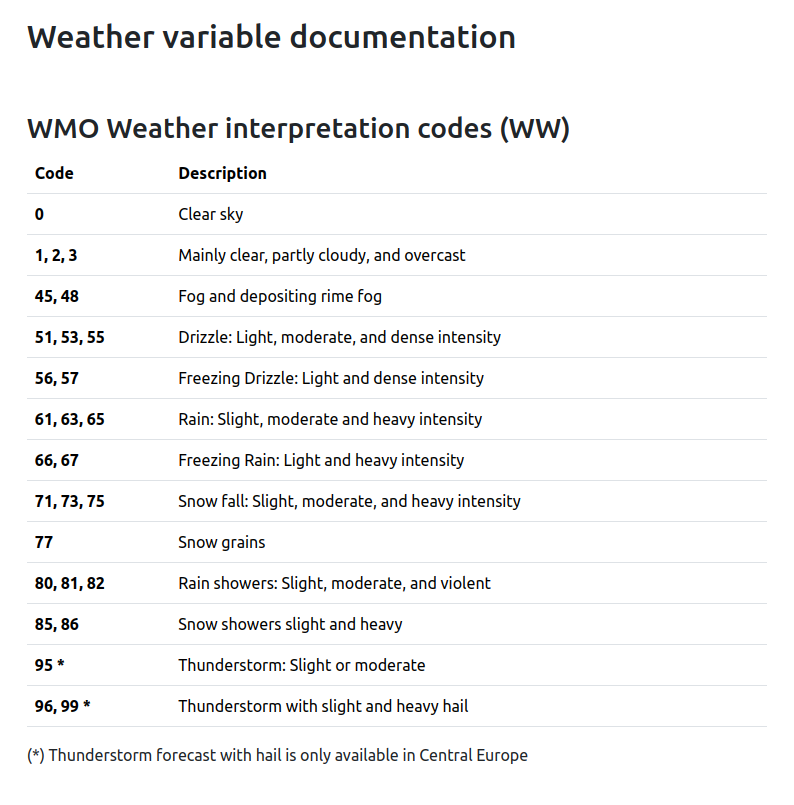
* Entrega solamente 04\_parte1.js y 04\_parte2.jsx con las soluciones solicitadas, comprimido en formato ZIP con el formato apellido1\_apellido2\_nombre.zip sustituyendo los valores por los del alumno(a) en cuestión. Ejemplo: perez\_lopez\_juan.zip

**Parte 1: jQuery y AJAX**

1. **(2 puntos)** Implementa la funcionalidad utilizando jQuery para que al hacer clic en los elementos *li* que pertenezcan a la clase *lugar* se conmuta la clase *enabled*. Es decir, se añade la clase *enabled* al elemento si no está presente y se elimina si lo está.
2. **(2 puntos)** Implementa la funcionalidad utilizando jQuery para que cuando se escriba en la caja de texto con el id *entradaBusqueda* y se haga clic en el botón con el id *botonBusqueda*, realice una petición AJAX a la API de [open-meteo.com](https://open-meteo.com/en/docs/) utilizando las coordenadas GPS de los lugares definidos. La petición se realizará solo si el texto introducido coincide con el de algún elemento *li* que tenga activa la clase *enabled*. En caso contrario, se ha de mostrar una alerta indicando que el lugar no está disponible.
3. **(2 puntos)** Implementa la funcionalidad para que se realice la petición AJAX correctamente al script *clima.php* de modo que debe se envía el parámetro *lugar* (por ejemplo, "Santiago de Compostela", "A Coruña" o "Vigo") y devuelva los datos climáticos en formato JSON. Actualiza la información en el elemento con id *infoClima* con los datos recibidos (ciudad, temperatura y condiciones climáticas en español).

**Guías y pistas para la parte 1:**

* Emplea la función **$.get()** para gestionar la petición AJAX y **JSON.parse()** para interpretar la respuesta. Revisa los ejemplos que emplean los métodos anteriores.
* En jQuery, puedes acceder a los atributos **data-<algo>** de un elemento HTML usando **$(selector).data("algo")**, donde "algo" es el nombre del atributo después de data-.
* La consulta AJAX será la siguiente (sustituyendo **<latitud\_GPS\_lugar>** y **<longitud\_GPS\_lugar>** por las coordenadas GPS oportunas:  
  [**https://api.open-meteo.com/v1/forecast?latitude=<latitud\_GPS\_lugar>&longitude=<longitud\_GPS\_lugar>&current=apparent\_temperature,weather\_code**](https://api.open-meteo.com/v1/forecast?latitude=%3Clatitud_GPS_lugar%3E&longitude=%3Clongitud_GPS_lugar%3E&current=apparent_temperature,weather_code)
* La interpretación del **<weather\_code>** seguirá la correspondencia de la siguiente tabla:



**Parte 2: React**

1. **(2 puntos)** Reescribe el archivo 04\_parte2.jsx para que sólo se renderice un componente llamado *gestorEventos* al que se le pasa como propiedad el objeto *eventos*. Además, en ese mismo archivo declara el componente *gestorEventos* con la funcionalidad inicial del archivo 04\_parte2.jsx.

**Parte común**

1. **(1.25 puntos)** Optimiza todo el código que has implementado tanto en velocidad como en espacio.
2. **(0.75 puntos)** Explica con comentarios embutidos en el código, todas las funcionalidades y justifica la estructuras de sintaxis utilizadas en cada caso.