```
/**
 * API para obtener datos con AJAX en JSON de una forma fácil y sencilla. Admite tanto GET como POST.
 * Para comprender cómo funciona el método que acontece, es necesario ver cómo funcionan las promesas de JavaScript:
 * https://desarrolloweb.com/articulos/introduccion-promesas-es6.html
 * @author Santiago San Pablo Raposo
 * @version 25.02.2023
 */
/**
 * Función que permite obtener un JSON desde un servidor web. Sin los parámetros opcionales, el comportamiento por defecto es usar el
método GET sin parámetros.
 * @param {string} url Especifica la URL desde la que se quiere obtener el JSON.
 * <code>@param {string}</code> metodo (Opcional) Especifica mediante una cadena de texto el metodo que se quiere emplear en la cabecera de la petición
HTTP.
 * Oparam {object} datos (Opcional) Especifica un objeto literal con los parámetros que deseas enviar al servidor.
 * @returns Un JSON / objeto literal con los datos que devuelva la API en el servidor en la ruta especificada.
*/
function obtenerJSON(url, metodo = "GET", datos = null) {
    return new Promise((resolve, reject) => {
        /*-- Verificar datos de los parámetros --*/
        if (metodo != "GET" && metodo != "POST" && metodo != "PUT" && metodo != "DELETE") {
            reject("El método especificado no es correcto. Especifica GET, POST, PUT o DELETE");
        if (datos != null && typeof(datos) != "object") {
            reject("Especifica los parámetros de la petición en un objeto literal/JSON");
        /*-- Efectúa la petición al servidor --*/
        switch (metodo) {
            case "GET":
                /*-- Prepara los parámetros en un objeto de tipo URL --*/
                let direccion = new URL(url);
                for (let key in datos) {
                    direccion.searchParams.append(key, datos[key]);
                }
                /*-- Realiza la petición al servidor --*/
                fetch(direccion, {method: metodo})
                    .then(response => {
                        console.dir(response);
```

```
if (response.ok) {
                return response.text();
            } else {
                reject("No se ha podido acceder a ese recurso. Status: " + response.status);
        })
        .then(responseText => {
            let objJson = JSON.parse(responseText);
            resolve(objJson);
        })
        .catch(err => reject(err));
    break;
case "POST":
   /*-- Prepara los parámetros en un objeto de tipo URLSearchParams --*/
    let parametros = new URLSearchParams();
    for (let key in datos) {
        parametros.append(key, datos[key]);
   /*-- Realiza la petición al servidor --*/
    fetch(url, {method: metodo, body: datos})
        .then(response => {
            if (response.ok) {
               return response.text();
            } else {
                reject("No se ha podido acceder a ese recurso. Status: " + response.status);
        })
        .then(responseText => {
           let objJson = JSON.parse(responseText);
            resolve(objJson);
        })
        .catch(err => reject(err));
    break;
default:
   /*-- PUT y DELETE no están implementados, ya que no he estudiado cómo funcionan --*/
    reject("No implementado el método " + metodo + ".");
    break;
```

});