



DOCUMENTACIÓN DWEC



DIEGO FREILE GARCÍA

Índice

Introducción.....	3
Script búsqueda de rutas	3
Script card rutas	5
Carga del header y del footer.....	6
Comprobar formularios	8
Script enviar datos formularios.....	10
Script mapa de leaflet	12
Script mis rutas	13
Script ventanas modales	15

Introducción

En este documento voy a explicar los scripts escritos en JavaScript para hacer funcional la aplicación web “Allroutes”. Estos scripts realizan funciones como: carga de elementos html, comprobación de formularios, carga de mapas, etc.

Script búsqueda de rutas

Script:

```
mainHeader();
mainBody();
mainList();

// Carga el header con los inputs del formulario
function mainHeader() {
    let main_header = document.querySelector(".main__header");

    main_header.innerHTML = `
        <div class='main__filters'>
            <form action="#" method="post"
onsubmit="functionSubmit(event)">
                <input class='inputs2' type='text'
name='nombre_ruta' id='nombre_ruta' onkeyup='mainList()'
placeholder='Nombre ruta...'>
                <input class='inputs2' type='number'
name='dist_min' id='dist_min' onkeyup='mainList()' min='0'
placeholder='Distancia mínima...'>
                <input class='inputs2' type='number'
name='dist_max' id='dist_max' onkeyup='mainList()' min='0'
placeholder='Distancia máxima...'>
            </form>
        </div>`;
}

// Carga en body los div para la lista de rutas y para el mapa de Leaflet
function mainBody() {
    let main_body = document.querySelector(".main__body");

    main_body.innerHTML = `
        <div class='main__list'></div>
        <div class='main__map'>
            <div id='map'></div>
        </div>
    `;
}
```

```

// Carga la lista de las rutas en em main list
function mainList() {
    let main_list = document.querySelector(".main__list");
    let inputNombreRuta = document.querySelector("#nombre_ruta").value;
    let inputDistMin = document.querySelector("#dist_min").value;
    let inputDistMax = document.querySelector("#dist_max").value;

    let url = `http://localhost/proyecto-
allroutes/php/api/routes?name=${inputNombreRuta}`;

    inputDistMin != '' ? url += `&min_dist=${inputDistMin}` : '';
    inputDistMax != '' ? url += `&min_dist=${inputDistMax}` : '';

    fetch(url, {
        method: 'GET',
        headers: {
            'Content-Type': 'application/json;charset=utf-8'
        },
    },
    ))
    .then(response => {
        return response.json();
    })
    .then(data => {
        let datos = "";
        if (data.length > 0) {
            data.forEach(item => {
                datos += `
                    <a class='enlaces' href='detalle-
ruta.php?id=${item['id']}'>
                        <div class='trail'>
                            <div class='trail__header'>
                                <span>${item['route_name']}</span>
                            </div>
                            <div class='trail__body'>
                                <div class='trail_item'>
                                    <small>Distancia</small>
                                    <h4>${item['distance']}</h4>
                                </div>
                                <div class='trail_item'>
                                    <small>Altura Máxima</small>
                                    <h4>${item['max_height']}m</h4>
                                </div>
                                <div class='trail_item'>
                                    <small>Altura Mínima</small>
                                    <h4>${item['min_height']}m</h4>
                                </div>
                            </div>
                            <div class='trail__image'><img
src='imgs/default-image.jpg' alt='imagen'></div>

```

```

        </div>
    </a>
    `;
    });

    } else {
        datos = "<h2 style='text-align: center;'>No se
encontraron rutas</h2>";
    }
    main_list.innerHTML = datos;
})
}

```

Este script tiene la función de cargar dinámicamente los elementos html de la página de búsqueda de rutas. Hay tres funciones: mainHeader, mainBody y mainList. La primera se encarga de cargar los inputs del formulario para filtrar las rutas, la segunda carga la estructura que tendrá la lista de las rutas y el mapa, y la última se encarga de hacer una consulta a la api para obtener las rutas, si las obtiene las inserta en el div main list y si no encontrara se insertaría en main list un mensaje de que no se encuentran rutas.

Script card rutas

Script:

```

let containerRutas = document.querySelector('.container-rutas');

cargarRutas();

function cargarRutas() {
    let url = `http://localhost/proyecto-allroutes/php/api/routes/`;

    fetch(url, {
        method: 'GET',
        headers: {
            'Content-Type': 'application/json;charset=utf-8'
        },
    },
    })
    .then(response => {
        return response.json();
    })
    .then(data => {
        if (data) {
            let cont = 0;
            let datos = "";
            if (data.length > 0) {
                data.forEach(item => {

```

```

        cont++;
        if (cont <= 6) {
            datos += `
                <div class="card-ruta">
                    
                    <h4>${item['route_name']}

```

Se encarga de cargar seis rutas en el div “container-rutas”, hace una consulta a la api y si hay rutas las añade al div y si no hubiera muestra un mensaje de que no hay rutas.

Carga del header y del footer

Script:

```

let paginaHeader = "";

if(!localStorage.getItem('token')) {
    paginaHeader = "layout/header.html";
} else {
    paginaHeader = "layout/header-inicioSesion.html";
}

const paginaFooter = "layout/footer.html";

loadHeader(paginaHeader);
loadFooter(paginaFooter);

```

```

function loadHeader(url) {
    fetch(url)
        .then( (response) => {
            return response.text()
        })
        .then( (data) => {
            document.querySelector('#encabezado').innerHTML = data;
        })
}

function loadFooter(url) {
    fetch(url)
        .then( (response) => {
            return response.text();
        })
        .then( (data) => {
            document.querySelector('#piePagina').innerHTML = data;
        })
}

function addNavList() {
    let ulSecond = document.querySelector('#ul-second');

    if(ulSecond.classList.contains('fondo_invisible')) {
        ulSecond.classList.remove('fondo_invisible');
        if(window.innerWidth <= 768) {
            document.getElementsByTagName('body')[0].classList.add('quitar_scroll');
        }
    } else {
        ulSecond.classList.add('fondo_invisible');
        document.getElementsByTagName('body')[0].removeAttribute('class')
    }
}

```

Carga el header y el footer dinámicamente en las páginas de la aplicación. Primero se comprueba si hay algún token existente en el LocalStorage, si no lo hubiera carga el header con los botones de iniciar sesión y registrarse, si por el contrario hubiera token carga el header con el botón del perfil del usuario.

Las funciones loadHeader y loadFooter se encargan de cargar el header y el footer a dos divs.

La función addNavList se encarga de añadir y quitar la clase “fondo_invisible” al div ulSecond. Y si la ventana es menos o igual a 768px también quita el scroll al body.

Comprobar formularios

Script:

```
function comprobarCamposVaciosSesi() {
    /* CAMPOS DEL FORMULARIO DE INICIAR SESION */
    let usermaneSesi = document.querySelector('#usernameSesi');
    let passSesi = document.querySelector('#passSesi');
    let btnFormSesi = document.querySelector('#btnFormSesi');

    /* COMPROBAR INPUTS FORMULARIO INICIO DE SESION */
    if(usermaneSesi.value == '' || passSesi.value == '') {
        btnFormSesi.disabled = true;
    } else {
        btnFormSesi.disabled = false;
    }
}

function comprobarCamposVaciosReg() {
    /* CAMPOS DEL FORMULARIO DE INICIAR SESION */
    let fullname = document.querySelector('#fullname');
    let usermaneReg = document.querySelector('#usernameReg');
    let email = document.querySelector('#email');
    let estatura = document.querySelector('#estatura');
    let peso = document.querySelector('#peso');
    let passReg = document.querySelector('#passReg');
    let repPass = document.querySelector('#repPass');
    let btnFormReg = document.querySelector('#btnFormReg');

    /* COMPROBAR INPUTS FORMULARIO DE REGISTRO */
    if(fullname.value == '' || usermaneReg.value == '' || email.value == ''
    || estatura.value == '' || peso.value == '' || passReg.value == ''
    || repPass.value == '') {
        btnFormReg.disabled = true;
    } else {
        btnFormReg.disabled = false;
    }
}

function comprobarPassReg() {
    let passReg = document.querySelector('#passReg');
    let passRegError = document.querySelector('#passRegError');

    // COMPRUEBA QUE LA CONTRASEÑA SEA MAYOR A 8 CARACTERES
    if(passReg.value.length < 8) {
        passReg.classList.remove('inputs');
```



```

passReg.classList.add('inputs2');

passRegError.classList.remove('fondo_invisible');
passRegError.classList.add('divPassRegError');

passRegError.innerHTML = `
    <span class='spanError'>La contraseña no es segura</span>
`;
} else {
    passReg.classList.remove('inputs2');
    passReg.classList.add('inputs');

    passRegError.classList.remove('divPassRegError');
    passRegError.classList.add('fondo_invisible');
}

if(passReg.value.length == 0) {
    passReg.classList.remove('inputs2');
    passReg.classList.add('inputs');

    passRegError.classList.remove('divPassRegError');
    passRegError.classList.add('fondo_invisible');
}
}

function comprobarRepPass() {
    let passReg = document.querySelector('#passReg');
    let repPass = document.querySelector('#repPass');
    let repPassError = document.querySelector('#repPassError');

    // COMPRUEBA QUE EL CAMPO DE REPETIR CONTRASEÑA SEA IGUAL QUE EL CAMPO DE LA CONTRASEÑA
    if(passReg.value != repPass.value) {
        repPass.classList.remove('inputs');
        repPass.classList.add('inputs2');

        repPassError.classList.remove('fondo_invisible');
        repPassError.classList.add('divRepPass');

        repPassError.innerHTML = `
            <span class='spanError'>Las contraseñas tienen que ser
iguales</span>
        `;
    } else {
        repPass.classList.remove('inputs2');
        repPass.classList.add('inputs');

        repPassError.classList.remove('divRepPass');
        repPassError.classList.add('fondo_invisible');
    }
}

```

```

    }

    if(repPass.value.length == 0) {
        repPass.classList.remove('inputs2');
        repPass.classList.add('inputs');

        repPassError.classList.remove('divRepPass');
        repPassError.classList.add('fondo_invisible');
    }
}

```

Este script se encarga de comprobar los formularios de las ventanas modales para iniciar sesión y para registrar usuarios. Las dos primeras funciones se encargan de comprobar que los campos no estén vacíos, si alguno esta lo está, desactiva los botones para enviar los formularios.

Las otras dos funciones se encargan de comprobar la seguridad de la contraseña y de que los dos campos sean idénticos.

Script enviar datos formularios

Script:

```

/* Funcion para quitar el comportamiento por defecto de los formularios
*/
function funcionSubmit(event) {
    event.preventDefault();
}

/* Funcion para registrar a un usuario */
function recogerDatos() {

    let fullname = document.querySelector('#fullname');
    let username = document.querySelector('#usernameReg');
    let email = document.querySelector('#email');
    let estatura = document.querySelector('#estatura');
    let peso = document.querySelector('#peso');
    let fechaNac = document.querySelector('#fechaNac');
    let pass = document.querySelector('#passReg');
    let senderismo = document.querySelector('#senderismo');
    let bicicleta = document.querySelector('#bicicleta');
    let running = document.querySelector('#running');
    let alpinismo = document.querySelector('#alpinismo');

    let actividades = [];

```

```

senderismo.checked ? actividades.push(senderismo.value) : '';
bicicleta.checked ? actividades.push(bicicleta.value) : '';
running.checked ? actividades.push(running.value) : '';
alpinismo.checked ? actividades.push(alpinismo.value) : '';

let datosUsuario = {
  "username": `${username.value}`,
  "fullname": `${fullname.value}`,
  "email": `${email.value}`,
  "pass": `${pass.value}`,
  "height": `${estatura.value}`,
  "weight": `${peso.value}`,
  "birthday": `${fechaNac.value}`,
  actividades,
};

let url = "http://localhost/proyecto-allroutes/php/api/users/";

fetch(url, {
  method: 'POST',
  headers: {
    'Content-Type': 'application/json;charset=utf-8'
  },
  body: JSON.stringify(datosUsuario)
})
.then(response => {
  return response.json();
})
.then(data => {
  if (data['success']) {
    alert(data['msg']);
  } else {
    alert(data['msg']);
  }
})
}

/* Funcion para iniciar sesion */
function recogerDatosInicio() {
  let nombreUsu = document.querySelector('#usernameSesi');
  let pass = document.querySelector('#passSesi');

  let datosUsuario = {
    username: `${nombreUsu.value}`,
    pass: `${pass.value}`
  };

  let url = "http://localhost/proyecto-allroutes/php/api/users/";

```

```

    fetch(url, {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json'
      },
      body: JSON.stringify(datosUsuario)
    })
    .then(response => {
      return response.json();
    })
    .then(data => {
      if (data['success']) {
        localStorage.setItem('token', `${data['token']}`);
        localStorage.setItem('username', `${data['username']}`);
        localStorage.setItem('id', `${data['id']}`);
        alert(data['msg']);
        location.reload();
      } else {
        alert(data['msg']);
      }
    })
  }

function cerrarSesion() {
  localStorage.clear();
  window.location = "index.html";
}

```

Este script tiene principalmente dos funciones que se encargan de recoger los campos de los formularios de inicio de sesión y de registro de usuarios. La función para registrar los usuarios obtiene los valores de los inputs y lo guarda en un objeto, luego realiza una consulta a la api para insertar ese objeto en la base de datos. La función para iniciar sesión es igual, recoge los campos del formulario, los inserta en un objeto y realiza una consulta a la api para comprobar que exista el usuario, si existe se guarda en el localStorage el token generado, el id y el username.

Script mapa de leaflet

Script:

```

let map = L.map('map')
    .setView([40.463667, -3.74922], 6);
L.tileLayer('https://tile.thunderforest.com/outdoors/{z}/{x}/{y}.png?apikey=8fea5469271547bdaa2bc623555e4432', {
  maxZoom: 19,

```

```

        attribution: '&copy; <a
href="https://www.thunderforest.com/terms/">ThunderForest</a>'
    }).addTo(map);

addMarkers();

function addMarkers() {
    let inputNombreRuta = document.querySelector("#nombre_ruta").value;
    let inputDistMin = document.querySelector("#dist_min").value;
    let inputDistMax = document.querySelector("#dist_max").value;

    let url = `http://localhost/proyecto-
allroutes/php/api/routes?name=${inputNombreRuta}`;

    inputDistMin != '' ? url += `&min_dist=${inputDistMin}` : '';
    inputDistMax != '' ? url += `&min_dist=${inputDistMax}` : '';

    fetch(url, {
        method: "GET",
        headers: {
            "Content-Type": "application/json;charset=utf-8"
        }
    })
        .then(response => {
            return response.json();
        })
        .then(data => {
            let marker = '';
            data.forEach(item => {
                marker += L.marker([item['start_lat'],
item['start_lon']], {
                    title: `${item['route_name']}`,
                    opacity: 1,
                    draggable: false,
                }).addTo(map);
            });
        })
    }
}

```

Añade a la página de búsqueda de rutas el mapa y con la función addMarkers añade los marcadores de las rutas encontradas.

Script mis rutas

Script:

```

/* Funcion para quitar el comportamiento por defecto de los formularios
*/
function funcionSubmit(event) {
    event.preventDefault();
}

let mr__header = document.querySelector('.mr__header').clientHeight;
let mr__body = document.querySelector('.mr__body');

mr__body.setAttribute('style', `max-height: calc(765px - ${mr__header} + 25px)`);

cargarRutas();

function cargarRutas() {
    let id_user = localStorage.getItem('id');
    let inputNombreRuta = document.querySelector("#nombre_ruta").value;
    let inputDistMin = document.querySelector("#dist_min").value;
    let inputDistMax = document.querySelector("#dist_max").value;

    let url = `http://localhost/proyecto-allroutes/php/api/routes?id=${id_user}`;

    inputNombreRuta != '' ? url += `&name=${inputNombreRuta}` : '';
    inputDistMin != '' ? url += `&min_dist=${inputDistMin}` : '';
    inputDistMax != '' ? url += `&max_dist=${inputDistMax}` : '';

    fetch(url, {
        method: 'GET',
        headers: {
            'Content-Type': 'application/json; charset=utf-8'
        },
    })
    .then(response => {
        return response.json();
    })
    .then(data => {
        let datos = "";
        if (data.length > 0) {
            data.forEach(item => {
                datos += `
                    <a class='enlaces' href='detalle-ruta.php?id=${item['id']}'>
                        <div class='trail' style='border: .1rem solid var(--blanco_oscuro)'>
                            <div class='trail__header'>
                                <span>${item['route_name']}</span>
                            </div>
                            <div class='trail__body'>

```

```

        <div class='trail_item'>
            <small>Distancia</small>
            <h4>${item['distance']}

```

Este script obtiene las de un usuario en concreto haciendo una consulta a la api y las inserta en un div de la pagina de mis rutas, si no hubiera rutas subidas por ese usuario mostraría un mensaje de que no se encuentran rutas.

Script ventanas modales

Script:

```

function iniciarSesion() {
    crearVentana();
    formularioLogin();
    let ventana = document.querySelector('#ventana');
    ventana.classList.remove('ventana_registro');
}

function registrarse() {
    crearVentana();
    formularioRegistro();
    let ventana = document.querySelector('#ventana');
    ventana.classList.add('ventana_registro');
}

```

```

}

function crearVentana() {
    let div = document.querySelector('#formularios');
    document.getElementsByTagName('body')[0].classList.add('quitar_scroll');
    div.classList.add('modal-show');
}

function cerrarVentana() {
    let div = document.querySelector('#formularios');
    document.getElementsByTagName('body')[0].removeAttribute('class');
    div.classList.remove('modal-show');
}

function formularioLogin() {
    document.querySelector('#ventana_contenido').innerHTML = `
        <h3>Inicio de sesión en AllRoutes</h3>
        <form action='#' method='post' onsubmit='funcionSubmit(event)'>
            <label for='username'>Nombre de usuario:</label><br>
            <input class='inputs' type='text' name='usernameSesi'
id='usernameSesi' onkeyup='comprobarCamposVaciosSesi()'>
            <br>
            <label for='passSesi'>Contraseña:</label><br>
            <input class='inputs' type='password' name='passSesi'
id='passSesi' onkeyup='comprobarCamposVaciosSesi()'>
            <br>
            <input class='boton_formulario' id='btnFormSesi'
onclick='recogerDatosInicio()' type='button' disabled='' value='Iniciar
sesión'>
        </form>
    `;
}

function formularioRegistro() {
    document.querySelector('#ventana_contenido').innerHTML = `
        <h3>Registrate en AllRoutes</h3>
        <form action='#' method='post' onsubmit='funcionSubmit(event)'>
            <label for='fullname'>Nombre y apellidos:</label><br>
            <input class='inputs' type='text' name='fullname'
id='fullname' onkeyup='comprobarCamposVaciosReg()'>
            <br>
            <label for='user'>Nombre de usuario:</label><br>
            <input class='inputs' type='text' name='usernameReg'
id='usernameReg' onkeyup='comprobarCamposVaciosReg()'>
            <br>
            <label for='email'>Correo electrónico:</label><br>
            <input class='inputs' type='email' name='email' id='email'
onkeyup='comprobarCamposVaciosReg()'>
    `;
}

```



```

        <br>
        <label for='estatura'>Estatura:</label><br>
        <input class='inputs' type='number' name='estatura'
id='estatura' onkeyup='comprobarCamposVaciosReg()'>
        <br>
        <label for='peso'>Peso:</label><br>
        <input class='inputs' type='number' name='peso' id='peso'
onkeyup='comprobarCamposVaciosReg()'>
        <br>
        <label for='fechaNac'>Fecha de nacimiento:</label><br>
        <input class='inputs' type='date' name='fechaNac'
id='fechaNac'>
        <br>
        <label for='passReg'>Contraseña:</label><br>
        <input class='inputs' type='password' name='passReg'
id='passReg' onkeyup='comprobarCamposVaciosReg(); comprobarPassReg();'>
        <br>
        <div id='passRegError' class='fondo_invisible'></div>
        <label for='repPas'>Repetir contraseña:</label><br>
        <input class='inputs' type='password' name='repPass'
id='repPass' onkeyup='comprobarCamposVaciosReg(); comprobarRepPass();'>
        <br>
        <div id='repPassError' class='fondo_invisible'></div>
        <label for='actividades'>Actividades:</label><br>
        <input type="checkbox" id="senderismo" name="senderismo"
value="senderismo">
        <label for="senderismo">Senderismo</label><br>
        <input type="checkbox" id="bicicleta" name="bicicleta"
value="bicicleta">
        <label for="bicicleta">Bicicleta</label><br>
        <input type="checkbox" id="running" name="running"
value="running">
        <label for="running">Running</label><br>
        <input style="margin-bottom: 3rem;" type="checkbox"
id="alpinismo" name="alpinismo" value="alpinismo">
        <label for="alpinismo">Alpinismo</label>
        <br>
        <input class='boton_formulario' id='btnFormReg'
onclick='recogerDatos()' type='submit' disabled='' value='Registrarse'>
    </form>
    `;
}

```

Se encarga de mostrar y darles funcionalidad a las ventanas modales de inicio de sesión y de registro. Las primeras cuatro funciones se encargan de crear las ventanas y de cerrarlas cuando se le de click a los respectivos botones. Y las dos últimas funciones se encargan de generar los respectivos formularios dinámicamente.