



# Pràctica: CRUD

ID	NOM	COGNOM	EDAT	MAGIA	CARACTERISTIQUES	HABILITAT	IMATGE
1	Vanessa	Enoteca	24	Otros	Timido-Feroz	<a href="#">Veure habilitat</a>	 <a href="#">Eliminar</a> <a href="#">Modificar</a>
2	Yami	Sukehiro	28	Agua	Valiente	<a href="#">Veure habilitat</a>	 <a href="#">Eliminar</a> <a href="#">Modificar</a>

[Nou Personatge](#)

Selena Catena Morales  
Gerard Martinez  
UF3: Esdeveniments. Management de  
formularis. DOM  
Mòdul 6: Desenvolupament web en entorn  
client  
Curs 2020-2021

# Índex

[Parts més importants del projecte](#)

[Dificultats trobades](#)

[Possibles ampliacions](#)

[Conclusions](#)

# Parts més importants del projecte

El nostre projecte es divideix en tres grans blocs: mostrar la taula de personatges, modificar un personatge i crear un personatge nou.

Començarem explicant com hem mostrat la taula dels personatges:

```
function genera_taula(personatges) {  
    // Obtener la referencia del elemento body  
    var div = document.getElementsByTagName("div")[0];  
  
    // Crea un elemento <table> y un elemento <tbody>  
    var tabla = document.createElement("table");  
    var tblBody = document.createElement("tbody");  
  
    // Crea les files  
    for (var i = 0; i <= personatges.length; i++) {  
        // Crea les celes de la fila  
        var hilera = document.createElement("tr");  
  
        if (i != 0) {  
            //comprovamos la hilera para incluir el css  
            hilera.setAttribute("class", "defTr");  
        }  
  
        //comprovem si estem en la primera fila  
        if (i == 0) {  
            //si es així, el que fem es indicar les claus dels objectes JSON  
            for (let propiedad in personatges[i]) {  
                var celda = document.createElement("th");  
  
                var textoCelda = document.createTextNode(propiedad.toUpperCase());  
                celda.appendChild(textoCelda);  
                hilera.appendChild(celda);  
            }  
        } else {  
            //si no estem en la primera fila, comencem a intruir les dades de cada paersoantge
```

En aquesta primera part de la funció, el que podem veure és la creació de la taula. Primerament, seleccionem el div on volem crear la taula. Seguidament, creem tantes file com personatges + 1 tenim a l'array JSON. Si ens trobem a la primera fila, el que fem és posar com a títol de columna les claus de l'objecte JSON. Sí no en trobem a la primera fila, el que realitzem és començar a crear els camps on mostrem la informació de cada personatge.

```
        for (let propiedad in personatges[i - 1]) {  
            if (propiedad == "imatge") {  
                //si la propietat correspon a la imatge, el que fem es crear un camp de imatge  
                //el src, el qual esta guardat en un string  
                var celda = document.createElement("td");  
                var textoCelda = document.createElement("img");  
                textoCelda.setAttribute("src", personatges[i - 1][propiedad]);  
                textoCelda.setAttribute("class", "classImg");  
                textoCelda.setAttribute("width", "75");  
                textoCelda.setAttribute("height", "75");  
                celda.appendChild(textoCelda);  
                hilera.appendChild(celda);  
  
                //creem tambe els enllaços de eliminar  
                var celda = document.createElement("td");  
                var textoCelda = document.createElement("a");  
                textoCelda.appendChild(document.createTextNode("Eliminar"));  
                textoCelda.setAttribute("class", "eliminar");  
  
                celda.appendChild(textoCelda);  
                hilera.appendChild(celda);  
  
                //creem tambe els enllaços de modificar  
                var celda = document.createElement("td");  
                var textoCelda = document.createElement("a");  
                textoCelda.appendChild(document.createTextNode("Modificar"));  
                textoCelda.setAttribute("class", "modificar");  
  
                celda.appendChild(textoCelda);  
                hilera.appendChild(celda);  
            }  
        }  
    }  
    div.appendChild(tabla);  
}
```

Sí el nom de la propietat que volem guardar és imatge, el que realitzem és crear un tag on guardarem el src de la imatge. A més a més creem els botons d'eliminar i modificar de cada personatge.

```
    } else if (propiedad == "habilitat") {  
        //si la propietat correspon a la habilitat, el que fem es crear uu enllaç que ens portara a una altre taula,  
        //on podrem veure l'habilitat  
        var celda = document.createElement("td");  
        var textoCelda = document.createElement("a");  
        textoCelda.appendChild(document.createTextNode("Veure habilitat"));  
        textoCelda.setAttribute("class", "habilitat");
```

Sí el nom de la propietat és habilitat, el que fem és crear un element de tipus A, que ens permetrà veure les habilitats de cada personatge. Aquesta part del codi estarà comentada més endavant, en l'apartat de les ampliacions.

```
    } else {  
        //la resta de celes son nomes td amb la informació que esta guardad en l'objecte JSON  
        var celda = document.createElement("td");  
        var textoCelda = document.createTextNode(personatges[i - 1][propiedad]);  
    }  
  
    celda.appendChild(textoCelda);  
    hilera.appendChild(celda);
```

Sí la propietat no correspon amb cap dels noms abans esmentats, el que fem és crear un td amb la informació corresponent.

```
    // agrega la hilera al final de la tabla (al final del elemento tblbody)  
    tblBody.appendChild(hilera);  
}  
  
// posiciona el <tbody> debajo del elemento <table>  
tabla.appendChild(tblBody);  
// appends <table> into <body>  
div.appendChild(tabla);  
  
//creem el boto a sota de la taula que ens servirà per crear un nou personatge  
var boto = document.createElement("button");  
boto.appendChild(document.createTextNode("Nou Personatge"));  
boto.setAttribute("id", "nou_personatge");  
boto.setAttribute("class", "lc");  
div.appendChild(boto);  
  
//funcio que serveix per afegir a cada boto de modificar el seu event de click  
generar_add_event_listener_modificar();  
  
//funcio que serveix per afegir a cada boto de eliminar el seu event de click  
generar_add_event_listener_eliminar();  
  
//funcio que serveix per afegir a cada boto de veure habilitat el seu event de click  
generar_add_event_listener_habilitat();
```

A continuació, creem un botó de nou personatge i cridem a 3 funcions que ens generen un event listener per a cada tipus de botó.

```

//afegim un event de click al boto de crear nou personatge
var nou_personatge = document.getElementById("nou_personatge");
nou_personatge.addEventListener("click", crear_nou_personatge);

set_eliminar_imatge();
document.getElementById("Taula").classList.add("Taula");
document.getElementById("Nou_personatge").classList.remove("nou_personatge");
document.getElementById("Actualitza").classList.remove("actualitza");
document.getElementById("TaulaH").classList.remove("Taula");

//realitzem un evente que al fer click a la n del teclat, creem un personatge nou
document.addEventListener("keydown", function (event) {
    if (event.key == "n") {
        crear_nou_personatge();
    }
});
//funcio per canviar el color de les capceleres de cada columna
set_multicolor_th();
}

```

Seguidament, creem l'esdeveniment de click per al botó del nou personatge. També creem un event listener de teclat: si piquem a la lletra "n", cridem a la funció `crear_nou_personatge()`

Per últim, cridem a una funció que és un event de ratolí, que quan passem per sobre del primer th canvia el color de la lletra aleatòriament.

Seguidament, parlarem sobre la funció que utilitzem per modificar:

```

5
6 function crear_formulari_modificar(posicio) {
7     let crear_div = document.createElement("div");
8
9     let div = document.getElementById("Actualitza");
10
11     div.appendChild(crear_div);
12
13     document.getElementById("Actualitza").classList.add("actualitza");
14     document.getElementById("Nou_personatge").classList.remove("nou_personatge");
15
16     for (let propiedad in personatges[0]) {
17         if (propiedad == "habilitat") {
18         } else {
19             let p = document.createElement("p");
20             p.appendChild(document.createTextNode(propiedad.toUpperCase()));
21             crear_div.appendChild(p);
22         }
23
24         if (propiedad == "caracteristiques") {
25             let caracteristicas_marcadas = personatges[posicio][propiedad].split("-");
26
27             crear_estadisticas(crear_div);
28
29             for (let i = 0; i < caracteristicas_marcadas.length; i++) {
30                 let input = document.getElementById(caracteristicas_marcadas[i]);
31                 input.setAttribute("checked", true);
32             }
33         }
34     }
35 }

```

Sí el nom de la propietat és habilitat no realitzem res.

Sí el nom de la propietat és característiques, el que fem és agafar la informació que hi ha guardada en aquesta clau i la guardem en un array separat per (-). Seguidament el que fem és recórrer totes les característiques que podem marcar. Comprovem sí l'array que hem creat anteriorment té alguna de les característiques. Si les té les mostrem marcades.

```
    } else if (propiedad == "imatge") {  
        let img = document.createElement("img");  
        img.setAttribute("id", "myImg");  
        img.setAttribute("class", "classImg");  
        img.setAttribute("src", personatges[posicio][propiedad]);  
        img.setAttribute("width", "100");  
        img.setAttribute("height", "100");  
        crear_div.appendChild(img);  
  
        let br = document.createElement("br");  
        crear_div.appendChild(br);  
  
        let br2 = document.createElement("br");  
        crear_div.appendChild(br2);  
  
        let input = document.createElement("input");  
        input.setAttribute("type", "file");  
        crear_div.appendChild(input);  
  
        let br3 = document.createElement("br");  
        crear_div.appendChild(br3);  
        let br4 = document.createElement("br");  
        crear_div.appendChild(br4);  
  
    } else if (propiedad == "id") {  
        crear_id(personatges[posicio][propiedad], crear_div);  
    }
```

Sí el nom de la propietat és imatge, el que fem és crear un input de tipus file, on agafem el src de l'array JSON

Sí el nom de la propietat és id, el que fem és crear a una funció que ens porta un comptador de personatges creats, fent així que aquest camp sigui un autonumeric.

```

    } else if (propiedad == "magia") {
        let magia = "";
        let magia_seleccionada = personatges[posicio][propiedad];
        let p = document.createElement("p");
        crear_div.appendChild(p);

        let select = document.createElement("select");
        crear_div.appendChild(select);

        for (i = 0; i < 5; i++) {
            magia = crear_magia(i);

            let option = document.createElement("option");
            option.appendChild(document.createTextNode(magia));
            if (magia_seleccionada == magia) {
                option.setAttribute("selected", "true");
            }
            select.appendChild(option);
        }
    } else if (propiedad == "habilitat") {
    } else {
        let input = document.createElement("input");
        crear_div.appendChild(input);

        input.value = personatges[posicio][propiedad];
    }
}

```

Sí el nom de la propietat és màgia, el que fem és crear 5 tags d'option amb 5 noms diferents. Per determinar aquests noms el que fem és crear a la funció crear\_magia(i) on li passem la i, la qual determinarà quin nom tindrà el tag. Sí el nom de la màgia que tenim guardat a l'objecte JSON és el mateix que el que anem a crear, el mostrem marcat.

```

crear_botones_aceptar(crear_div);

boto = document.getElementById("Aceptar");

boto.addEventListener("click", function () {
    errors = comprobar_dades();
    if (errors == false) {
        aceptar_modificacio();
    } else {
        alert(cadena_errors);
    }
});

let boto2 = document.createElement("button");

boto2.appendChild(document.createTextNode("Cancelar"));
boto2.setAttribute("id", "Cancelar");
crear_div.appendChild(boto2);

boto2.addEventListener("click", cancelar_modificacio);

generarImg();
guardarImg();

document.addEventListener("keydown", function (event) {
    if (event.key == "Escape") {
        cancelar_modificacio();
    }
});
}

```

Seguidament, cridem a la funció per a crear el botó d'acceptar. En el botó d'acceptar hi afegim un esdeveniment listener quan el cliquem. En fer clic el que fem és comprovar si hi ha algun error en el formulari, com que les expressions regulars no s'estan respectant o que estan introduint algun camp en blanc.

Per últim, el que realitzem és crear el botó de cancel·lar la modificació i crear un esdeveniment de teclat que ens fa tornar a la taula inicial quan fem clic a l'esc.

Per últim, explicarem la funció de crear un nou personatge.

```
function crear_nou_personatge() {  
    buidar_taula();  
    crear_formulari();  
  
    let id = (document.getElementsByTagName("input")[0].value = contador + 1);  
}
```

Quan li donem al botó de crear un nou personatge, aquesta funció crida a dues funcions, les quals són treure el div de la taula, barrant el seu contingut, i ficar-hi el div del formulari, creant un formulari de creació de personatge des de 0.

Al ser un nou personatge, el que fem és donar-li un nou valor en la ID, que equival al següent número del comptador.

Crear un nou personatge, cridant a la funció de creació de formulari, no canvia massa de la modificació d'aquest.

Els únics aspectes que varien, són l'autoincrementació numèrica i els camps buits. Quan modifiquem un camp, tots els **inputs** estan omplerts, però quan creem un, tan sols creem el formulari amb tots els camps buits.

I l'acceptació del nou personatge, és la mateixa de quan es modifica. Agafem els valors dels camps, i mirem si no estan buits. En cas de que el camp estigui mal escrit, o no s'hagi seleccionat un camp, ens dona un alert en el que ens demana que omplim els camps correctament. Finalment, fa un push amb els nous valors, i la nova ID, per a introduir un nou personatge a la taula inicial.

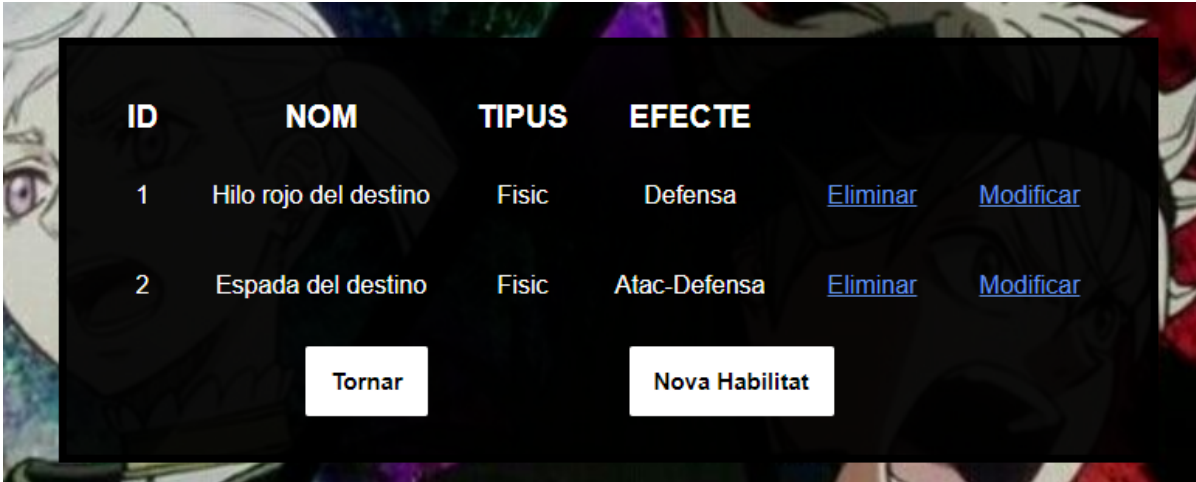
## Dificultats trobades

Les dificultats que hem trobat han estat a l'hora d'utilitzar un objecte JSON, ja que normalment no treballem amb ells. Una altra dificultat important ha estat com gestionar la posició de l'array JSON y del seu subarray, on tenim guardades les habilitats dels personatges.

Per una altra banda, també tenim la dificultat de gestionar la informació que guardem en diferents tipus d'inputs a l'hora de crear un nou personatge o de modificar un existent.



## Possibles ampliacions



ID	NOM	TIPUS	EFECTE		
1	Hilo rojo del destino	Fisic	Defensa	<a href="#">Eliminar</a>	<a href="#">Modificar</a>
2	Espada del destino	Fisic	Atac-Defensa	<a href="#">Eliminar</a>	<a href="#">Modificar</a>

Tornar

Nova Habilidad

La taula d'habilitats, no dinsta de la taula inicial. Dins de la taula inicial, afegim una nova subtaula JSON, que correspon amb una taula única per a cada un dels personatges, la qual, organitza les habilitats que té del personatge. Donant les mateixes opcions que amb els personatges, tal que son, eliminar habilitats, modificar-les o crear-les.

A l'hora de crear la taula, fem servir el mateix procediment que quan creem la taula de personatges. Donant-li una ID, un nom i les opcions d'eliminar i modificar. En aquesta, afegim dos camps els quals són tipus i efecte.

El camp de tipus és creat a partir de camps de tipus **radio**. En el qual o és un o és l'altre. El camp d'efecte és un checkbox, entre 4 elements a elegir.

Per a viatjar entre taules, creem un botó de tornar, que elimina el contingut de la taula d'habilitats i la reemplaça per la de personatges.

Diferències de la taula inicial:

```

    crear_div.appendChild(input);

} else if (propiedad == "efecte") {

    for (let i = 0; i < 4; i++) {
        switch (i) {
            case 0:
                caracteristica = "Atac";
                break;
            case 1:
                caracteristica = "Defensa";
                break;
            case 2:
                caracteristica = "Suport";
                break;
            case 3:
                caracteristica = "Cura";
                break;
        }
        let input = document.createElement("input");
        input.setAttribute("type", "checkbox");
        input.setAttribute("id", caracteristica);
        input.setAttribute("class", "checkbox");

        let label = document.createElement("label");
        label.setAttribute("for", caracteristica);
        label.innerText = caracteristica;
        crear_div.appendChild(input);
        crear_div.appendChild(label);
    }
} else if (propiedad == "tipus") {

    for (let i = 0; i < 2; i++) {

        switch (i) {
            case 0:
                tipus = "Fisic";
                break;
            case 1:
                tipus = "Magic";
                break;
        }
    }
}

```

Anteriorment, hem treballat amb el camp de checkbox, així que no ha sigut difícil implementar aquest camp en la nova taula, i guardar els seus elements. El de tipus **radio** sí que ha sigut una implementació nova en la que varem pensar, i hem implementat en la nova part. La creació és senzilla:

```

let input = document.createElement("input");
input.setAttribute("type", "radio");
input.setAttribute("id", tipus);
input.setAttribute("class", "radio");
input.setAttribute("name", "efecte");

if (i == 0) input.setAttribute("checked", true);

let label = document.createElement("label");
label.setAttribute("for", tipus);
label.appendChild(document.createTextNode(tipus));
crear_div.appendChild(input);
crear_div.appendChild(label);

```

Fem servir el mateix *modus operandi* que en el checkbox, amb una petita variació. Ja que forma part del mateix camp, li donem un name "tipus" als dos inputs, per a que tan sols un es pugui seleccionar.

Per a estalviar-nos errors i codi, de primeres, com s'ha de seleccionar un si o si. Fem que el primer input radio sigui seleccionat de manera predeterminada.

```

function comprobar_dades_habilitats(){

    let efectes = document.getElementsByClassName("checkbox");
    let errors = false;
    let cadena = "";
    cadena_errors = "";
    let contador = 0;

    if (document.getElementsByTagName("input")[1].value == ""){
        errors = true;
        document.getElementsByTagName("input")[1].style.borderColor = "red";
        cadena += "El nom és obligatori";
    }

    for (let i = 0; i < efectes.length; i++){
        if (efectes[i].checked == false) contador++;
    }

    if (contador == 4){
        errors = true;
        cadena += "\nHas de marcar una efecte mínim";
    }

    cadena_errors = cadena;
    return errors;
}

```

Aleshores, tan sols hem de indicar si el que fa falta és el nom de l'habilitat o l'efecte que té.

```

function acceptar_modificacio_habilitat() {

    let id = document.getElementsByTagName("input")[0].value;
    let nom = document.getElementsByTagName("input")[1].value;
    let efecte = document.getElementsByClassName("checkbox");
    let efectes_marcats = "";

    for (let i = 0; i < efecte.length; i++) {
        if (efecte[i].checked == true) {
            efectes_marcats += efecte[i].id + "-";
        }
    }

    if (efectes_marcats.charAt(efectes_marcats.length - 1) == "-") {
        efectes_marcats = efectes_marcats.slice(0, -1);
    }

    let tipus = document.getElementsByClassName("radio");
    let valTipus = "";

    if (tipus[0].checked){
        valTipus = tipus[0].id;
    } else if (tipus[1].checked){
        valTipus = tipus[1].id;
    }

    personatges[posicio_global].habilitat[pos_global].id = id;
    personatges[posicio_global].habilitat[pos_global].nom = nom;
    personatges[posicio_global].habilitat[pos_global].tipus = valTipus;
    personatges[posicio_global].habilitat[pos_global].efecte = efectes_marcats;

    let taula = document.getElementById("Actualitza").firstChild;
    taula.parentNode.removeChild(taula);

    genera_tabla_habilitat();
}

```

A l'hora d'acceptar aquesta, els únics canvis que hem de fer del codi original, és on guardar els nous valors, de la sub-taula, i verificar el valor de l'input tipus radio.

Si el primer valor de tipus radio és seleccionat, es guardarà la id del camp seleccionat, que en aquest cas és el valor del camp. Ja que si cridem al **value**, ens retorna **on** o **off**. Donada la id, que porta el nom del camp, ens serà més fàcil dir quin es el radio seleccionat.

Donats els valors que tenim de la creació de la nova habilitat, tan sols hem de implementar-los en el fitxer JSON. El qual es troba dins del fitxer JSON de personatges. Així que donat el el personatge en la posició actual, donada per la variable de posició global. Ens adentrem dins del fitxer habilitat, en la posició donada per la ID de la qual agafem els nous valors. I donats els camps predeterminats, hi afegim els valors corresponents als camps.

Sabent com agafar els valors dels camps, en la modificació de l'habilitat es fa el mateix, per a mostrar els camps seleccionats.

```
let tipus_marcados = personatges[posicio_global].habilitat[posicio][propiedad].split("-");
console.log(tipus_marcados);

for (let i = 0; i < tipus_marcados.length; i++) {
  let input = document.getElementById(tipus_marcados[i]);
  input.setAttribute("checked", true);
}

let efecte_seleccionado = personatges[posicio_global].habilitat[posicio][propiedad];
console.log("Tipus: " + efecte_seleccionado);
```

I finalment, el botó de cancel·lar tan sols treiem el div actual i fiquem el de la taula d'habilitats.

Possibles extensions, seria trastejar més el nou fitxer JSON d'habilitats, afegint més camps, imatges potser, i altres elements de formularis no utilitzats en la pràctica.

## Conclusions

En aquesta pràctica hem après a gestionar correctament un array JSON. També hem après a crear diferents elements des de JS i també a eliminar-los. Per una altra banda, també hem fet servir expressions regulars amb JS per validar els inputs a l'hora de crear un personatge o modificar un que ja tenim un creat.