

WHAT STRAIN?

PROYECTO INTEGRADO

DESARROLLO WEB EN ENTORNO CLIENTE



ALUMNO: *Alberto García Sola*

CURSO: *2 DAW*

ÍNDICE

¿EN QUÉ CONSISTE Y A QUIÉN VA DIRIGIDA?.....	3
OBJETIVO.....	3
¿CÓMO SE USA?.....	3
REQUISITOS.....	6
1. Eventos de ratón (2 además de onclick).....	6
2. Eventos de teclado.....	8
3. Expresiones regulares (3 como mínimo).....	10
4. Empleo de cookies.....	11
5. DOM: creación de varios elementos de forma dinámica en el documento utilizando las propiedades y métodos necesarios para ello.....	12
¿CÓMO FUNCIONA?.....	13
ESTRUCTURA DEL PROYECTO	22
CONCLUSIÓN Y OTROS DETALLES	22

¿EN QUÉ CONSISTE Y A QUIÉN VA DIRIGIDA?

Hay muchísima gente que consume cannabis, pero gran parte de esta gente no sabe qué está consumiendo en realidad, por otro lado hay muchísima gente que nunca a consumido y que quiere animarse a probarlo pero no sabe que variedad elegir, qué efectos tiene cada una, qué sabores, qué tipos hay... Esto fue lo que me animó a realizar este proyecto.

Actualmente existen más de dos-mil variedades de cannabis, haciendo muy difícil la tarea de elegir una. Mi página web está pensada para ayudarte con la decisión.

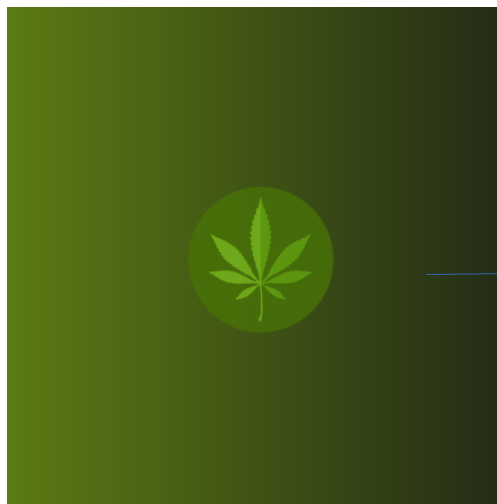
Por lo que esta pagina web está dirigida tanto para gente que nunca a consumido y quiere iniciarse probando exactamente lo que el quiera, como para veteranos que quieran curiosear y descubrir variedades que no conocían.

OBJETIVO

El objetivo es que un usuario, teniendo claro que sabor y efecto quiere concretamente pueda encontrar todas las variedades que contienen ese efecto y sabor.

¿CÓMO SE USA?

Lo primero que el usuario ve cuando entra en la página web es un preloader que se mostrará únicamente mientras se carga la página, seguido del preloader se mostrará un formulario en el que se pide el nombre, la edad y el correo.

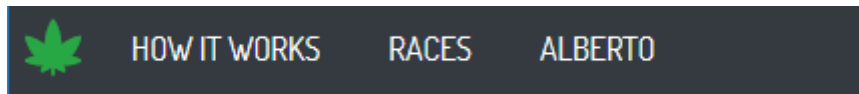


PRELOADER

A login form with a dark green background. It contains three input fields: 'NAME' with the value 'Alberto', 'AGE' with the value '21', and 'EMAIL' with the value 'alberto@alberto.com'. Below these fields is a green button labeled 'VALIDATE'.

FORMULARIO DE INICIO

Una vez completado el formulario accederá a la pantalla principal(que además es la única pantalla), en dicha pantalla podemos observar un menú en la parte superior en el que tendremos tres pestañas: **How it works**, **Races** y **Usuario**.



How it works: Esta pestaña nos indica de forma breve como usar la página web.

Races: Esta pestaña nos muestra información a cerca de las razas de marihuana que hay.

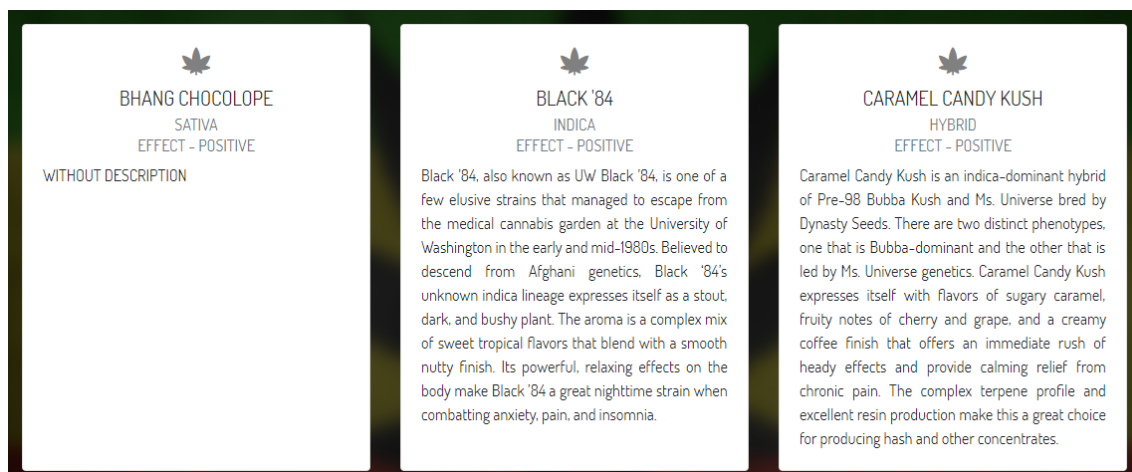
Usuario: El nombre de esta pestaña será el nombre que haya puesto el usuario en el input del Nombre del formulario. Y la función de esta pestaña es que si no eres el usuario que pone en la pestaña puedas "Iniciar sesión" con otro usuario.

Pongo "Iniciar sesión" entre comillas porque en realidad no inicias sesión ya que este no es uno de los objetivos de la página web. Me pareció que no hay información relevante que guardar. No hay cuentas ni nada por el estilo.

La primera vez que el usuario entra en la página se quedan guardados sus datos en las cookies, de manera que la siguiente vez que el usuario entre en la página si ya hay información en las cookies querrá decir que ya había entrado antes y por lo tanto se saltará directamente a la pantalla principal sin tener que volver a pasar por el formulario.

The main screen has a dark green background with a faint cannabis leaf pattern. At the top, it asks 'WHAT EFFECT ARE YOU LOOKING FOR?' with a dropdown menu showing 'HAPPY'. Below that, it asks 'WHAT FLAVOR DO YOU WANT?' with a dropdown menu showing 'COFFEE'. At the bottom, there are three checkboxes: 'SATIVA', 'INDICA', and 'HYBRID', all of which are checked. A green 'SEARCH' button with a magnifying glass icon is at the very bottom.

En la pantalla principal tenemos dos selects, uno para seleccionar el sabor y otro para seleccionar el efecto, además tenemos tres checkboxes que por defecto estarán marcados para seleccionar la o las razas de marihuana que estás buscando. Una vez seleccionados solo resta clicar en el botón **SEARCH** para ver las marihuanas que coinciden con el sabor, efecto y razas seleccionados



RESULTADOS

Cada variedad obtenida contiene:

- Un logo que, si se muestra de color verde significa que esa variedad está añadida a tus favoritas. Para añadir o quitar una variedad a tus favoritas hay que hacer doble clic sobre la variedad deseada. Y para ver las variedades que tenemos agregadas a favoritas podemos pulsar **Ctrl + I**.
- Un título con el nombre de la variedad.
- La raza a la que pertenece (sativa, indica o híbrida).
- El tipo de efecto (positivo, negativo o medicinal).
- Una breve descripción sobre la variedad.

REQUISITOS

1. Eventos de ratón (2 además de onclick)

dblclick: Utilizado para agregar o eliminar una variedad a nuestra lista de favoritas.

Se añade un evento de doble clic que disparará la función **addAndRemoveFavorite** a cada elemento que contenga la clase "card", que son los elementos que contienen cada variedad obtenida.

```
function addEvents() {  
  let classname = document.getElementsByClassName("card");  
  Array.from(classname).forEach(function(element) {  
    ////////////////////////////////// EVENTO DE RATÓN //////////////////////////////////  
    element.addEventListener('dblclick', addAndRemoveFavorite);  
  });  
}
```

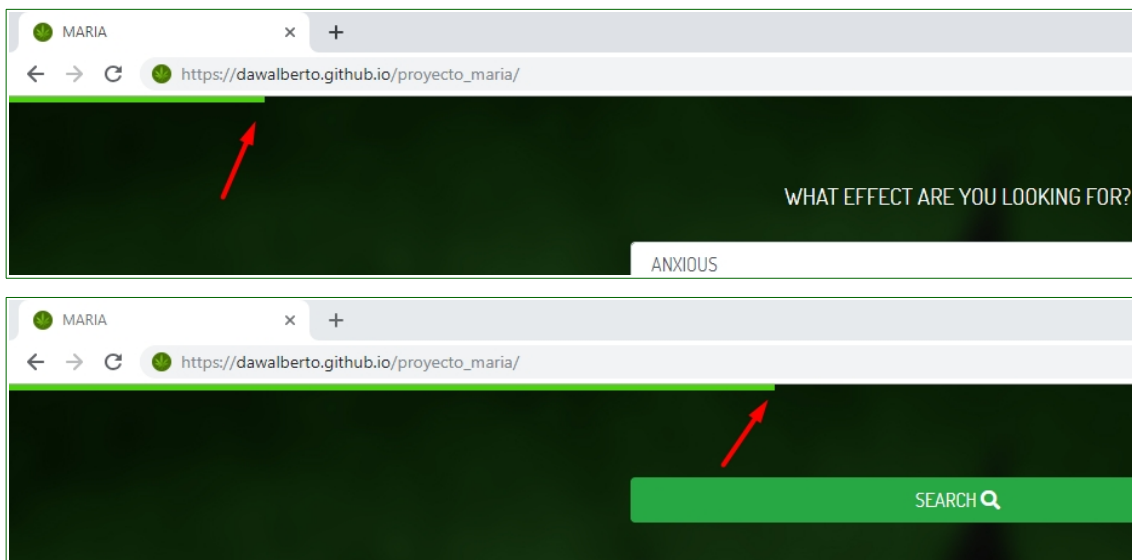
addAndRemoveFavorite es la función que añade o elimina variedades a nuestra lista de favoritos, se explicará con más detalle en la sección ¿CÓMO FUNCIONA? donde se explican todas las funciones.

Línea: [617](#)

scroll: Utilizado para actualizar la barra de progreso horizontal situada en la parte superior de la página que nos indica en todo momento cuanto queda para llegar al inicio o final de la página.

```
let barProgress = document.getElementById('progress');  
window.addEventListener('scroll', () => {  
  let max = document.body.scrollHeight - innerHeight;  
  barProgress.style.width = `${(pageYOffset / max) * 100}%`;  
  showAndHiddenElements();  
});
```

Se añade un evento **scroll** al elemento **window**, de manera que cada vez que hacemos scroll en la página se calcula el alto total de la página(**scrollHeight**) menos el alto que se muestra actualmente(**innerHeight**), obteniendo el alto total en el que se puede hacer scroll. Si dividimos el alto actual en el que nos encontramos(**pageYOffset**) entre el alto total sobre el que podemos hacer scroll y lo multiplicamos entre cien, obtenemos el porcentaje exacto de la altura en la que nos encontramos en la página. Establezco la propiedad **style.width** de la barra de navegación con este porcentaje obteniendo como resultado una barra de progreso dinámica.



Línea: [582](#)

2. Eventos de teclado

keydown: Utilizado para mostrar todas las variedades agregadas a tus favoritas.

En el momento en que se presione **Ctrl+I** o **Ctrl+L** aparecerán todas las variedades que hayas agregado a tus favoritas.

```
function addEventToShowFavorites() {  
  window.addEventListener('keydown', function(event) {  
    if (event.ctrlKey && (event.code === 'KeyI' || event.code === 'KeyL')) {  
      event.preventDefault();  
  
      if (arrayFavoritesCookies.length > 0) {  
        document.getElementById('containerRes').style.display = 'block';  
        document.getElementById('footer').style.display = 'block';  
        generateCardsMarias('toFavorites');  
        window.scrollTo(0, window.innerHeight);  
      } else {  
        barProgress.style.width = '0';  
        document.getElementById('containerRes').style.display = 'none';  
        document.getElementById('footer').style.display = 'none';  
        alert("YOU DON'T HAVE WEED FAVORITES");  
      }  
    }  
  });  
}
```

Se añade el evento **keydown** al elemento **window**, de manera que si presionamos las teclas **Ctrl+I** o **Ctrl+L** y tenemos agregadas variedades a favoritas, se ejecutará la función encargada de generar las variedades favoritas, la función **generateCardsMarias('to favorites')** la cual explicaré mas adelante.

Línea: [824](#)

onkeypress: Evento usado para disparar una función en la cual, si se a pulsado la tecla **enter** se ejecutará la misma función que se ejecuta para validar el formulario de inicio, la función **validateForm()**. La utilidad de este evento, usado en cada input del formulario de inicio es que podamos validar el formulario de inicio presionando la tecla enter desde cualquier input, al igual que en muchos formularios que encontramos por internet, y no únicamente haciendo clic sobre el propio botón para validar el formulario.

```
function validateFormWithKeyEnter(event) {
    if (event.keyCode == 13) {
        validateForm();
    }
}
```

```
<div class="col-12 divLabelForm">
  <label for="inputNombre">NAME</label>
</div>
<div class="col-md-8 col-lg-6">
  <input type="text" id="inputNombre" class="form-control" onkeypress="validateFormWithKeyEnter(event)" />
</div>
<div class="col-12 divSpanForm">
  <span id="spanNombre">ERROR</span>
</div>
</div>
<div class="row justify-content-center" id="divEdad">
  <div class="col-12 divLabelForm">
    <label for="inputEdad">AGE</label>
  </div>
  <div class="col-md-8 col-lg-6">
    <input type="tel" id="inputEdad" class="form-control" onkeypress="validateFormWithKeyEnter(event)" />
  </div>
  <div class="col-12 divSpanForm">
    <span id="spanEdad">ERROR</span>
  </div>
</div>
<div class="row justify-content-center" id="divCorreo">
  <div class="col-12 divLabelForm">
    <label for="inputCorreo">EMAIL</label>
  </div>
  <div class="col-md-8 col-lg-6">
    <input type="text" id="inputCorreo" class="form-control" onkeypress="validateFormWithKeyEnter(event)" />
  </div>
  <div class="col-12 divSpanForm">
    <span id="spanCorreo">ERROR</span>
  </div>
</div>
```

Líneas: js [498](#) html [44](#), [55](#), [66](#)

3. Expresiones regulares (3 como mínimo)

Todas las expresiones regulares usadas en esta página se usan concretamente en la parte del formulario de inicio y son usadas para validar, el nombre, la edad y el email:

1. Expresión regular para el nombre: `/^[A-Za-z]+$`

Únicamente se podrán introducir letras mayúsculas o minúsculas y sin ningún espacio, el símbolo + indica que al menos debe haber una letra. La idea de este campo es que el usuario ponga un nombre cualquiera simplemente para posteriormente dirigirnos a él en la página web por su nombre. Por eso no se requieren apellidos ni por lo tanto espacios.

Línea: [366](#)

2. Expresión regular para la edad: `/^\d{1,}$`

Únicamente se podrán introducir dígitos(`\d`) y como mínimo tendrá que introducir uno(`{1,}`)

Línea: [400](#)

3. Expresión regular para el email: `/^\w+([\.-]?\w+)*@\w+([\.-]?\w+)*(\.\w{2,3})+$`

La parte del email antes de la @ requiere que empiece al menos por un carácter alfanumérico(`^\w+`) seguido o bien ya por la @ o por un . o un -, en cuyo caso deberá ir seguido de otro carácter alfanumérico(`([\.-]?\w+)*@`). La parte de detrás de la @ es exactamente igual que la de delante a excepción de que debe terminar con un . seguido de como mínimo dos caracteres alfanuméricos y como máximo tres(`(\.\w{2,3})+$`).

Línea: [445](#)

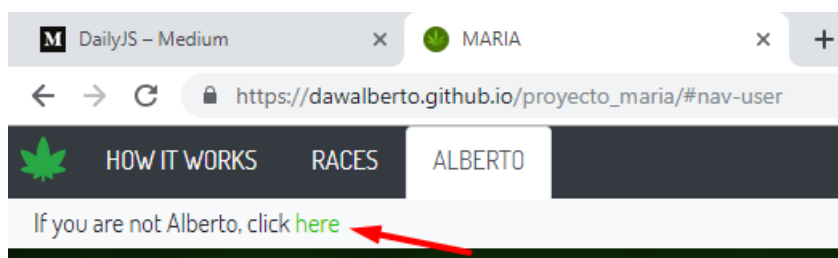
4. Empleo de cookies.

Las cookies tienen un gran papel en esta página web, en ellas recojo el **nombre** del usuario para dirigirme a él en otros lugares de la página, la **edad** para saber si es un adulto y mostrarle entonces el contenido, el **email**, la variedades **agregadas a favoritas** y un valor para saber si debo **reiniciar las cookies**. Tienen esta pinta:

```
> document.cookie  
< "age=21; nombre=Alberto; email=alberto@alberto.com; restartcookies=no; favorites=["hicon478","hicon549","hicon12"]"  
>
```

Como se puede observar este usuario tiene agregadas a favoritas tres variedades, que en realidad son ids que luego utilizo para encontrar las variedades de verdad. La utilidad de **restartCookies** es que si está puesta en **no**, el usuario saltará directamente a la pantalla de inicio sin pasar por el formulario, de lo contrario significará que el usuario a clicado en la pestaña de su nombre y a salido de su cuenta(entiéndase salir de su cuenta como resetear las cookies y por lo tanto perder la información anterior), poniendo **restartCookies** a **si** y teniendo que volver a pasar por el formulario de inicio.

La primera vez que se guardan cookies en la página web es en el formulario de inicio, donde se guarda el nombre, edad y email puestos por el usuario, también se guarda la cookie **favorites** que será un array vacío, y **restartCookies** con el valor **no**, que únicamente cambiará al valor **si** teniendo que volver a pasar por el formulario, cuando el usuario actual clique explícitamente en:



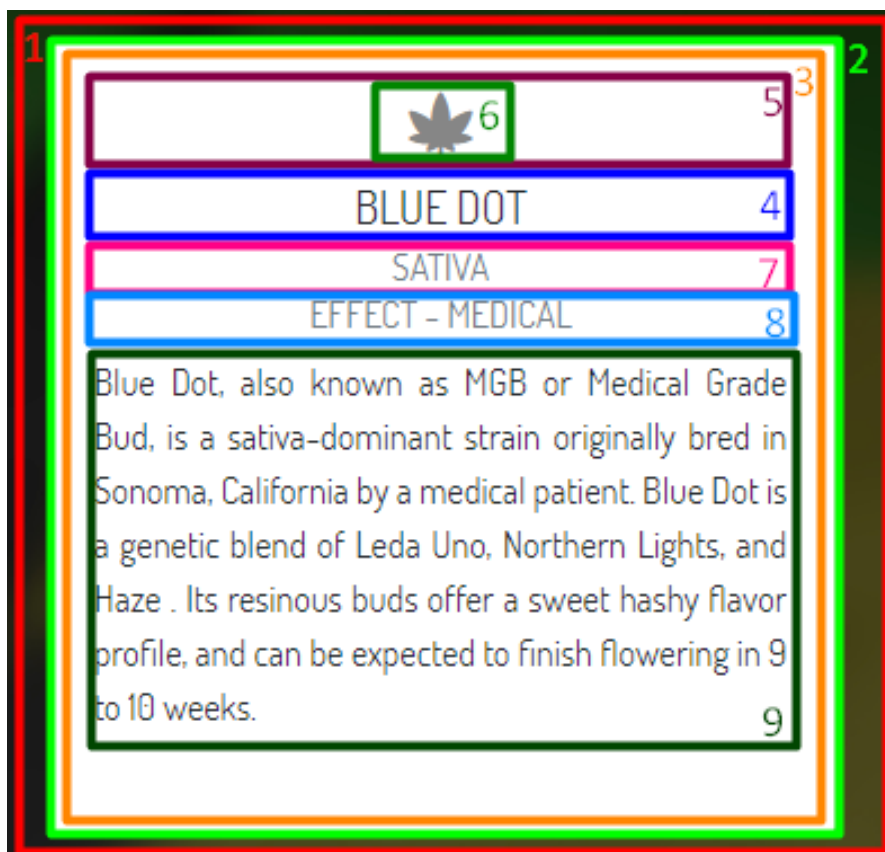
Las demás cookies se quedan intactas a excepción de **favorites**, la cual va aumentando o disminuyendo su valor(ya que es un array) dependiendo de si el usuario agrega o elimina variedades a su lista de favoritas.

Antes de guardar el array como cookie lo transformo con **JSON.stringify** y cuando lo recupero para trabajar con él en JavaScript uso **JSON.parse**.

Líneas: [515](#), [541](#), [553](#)

5. DOM: creación de varios elementos de forma dinámica en el documento utilizando las propiedades y métodos necesarios para ello.

Todos los variedades resultantes de la búsqueda del usuario se crean de forma dinámica. Se recorre un bucle a través de las variedades encontradas para los requisitos del usuario y para cada variedad encontrada se crea un elemento **div(1)** que será la columna que contenga el card, otro **div(2)** que será la card que contenga la variedad, otro **div(3)** que será el cuerpo de la card, un **h5(4)** que contendrá el nombre de la variedad, un **h3(5)** para contener el icono, un **i(6)** que será el icono, un **h6(7)** que será la raza, otro **h6(8)** que será el tipo de efecto y un **p(9)** para la descripción. Todo esto se hace en la función **generateCardsMarias()**, la cual se explicará mas adelante.



Línea: [237](#)

¿CÓMO FUNCIONA?

En esta sección se explicará que realiza cada función y para que sirve cada variable global definida en el archivo app.js, el JavaScript principal de la página web. No es necesario leer esta sección para usar la página web, pero te puede servir como referencia para ver alguna variable o función que no te haya quedado claro como funciona o para que sirve.

VARIABLES

VARIABLE	DESCRIPCIÓN	LÍNEA
marias	Usada para recoger la respuesta con los datos de la llamada a la API	1
arrayMarias	Guarda todas las variedades obtenidas de la llamada a la API en forma de objeto.	2
arrayEffectsAll	Contiene todos los efectos de todas las variedades.	3
arrayFlavorsAll	Contiene todos los sabores de todas las variedades.	4
arrayMariasSearch	Esta variable cambia su valor cada vez que el usuario busca variedades, en ella se almacenan las variedades que coinciden con la búsqueda del usuario.	5
arrayFavoritesCookies	Guarda los ids de las variables agregadas a favoritos para poder localizarlas cuando sea necesario.	6

FUNCIONES

FUNCIÓN	DESCRIPCIÓN	LÍNEA
getData()	<p>Esta función obtiene todos los datos de la API y recorre un bucle sobre la respuesta transformando cada variedad en un objeto (gracias a la función Maria()) que voy almacenando en la matriz arrayMarias. Además se encarga de llamar a las funciones:</p> <p>generateDescMarias()</p> <p>effectAll()</p> <p>flavorAll()</p> <p>fillSelects()</p>	10
Maria()	<p>Utilizada para crear los objetos antes de introducirlos en la matriz desde getData(). Los objetos creados con esta función tendrán las propiedades: id, nom, race, effectsM, effectsN, effectsP, effectsSum(que es un suma de todos los efectos anteriores) y description (propiedad a la cual se le dará valor posteriormente)</p>	63
effectsAll()	<p>Encargada de agregar a la matriz arrayEffectsAll todos los efectos de todas las variedades. Además los ordena alfabéticamente.</p>	78
flavorsAll()	<p>Encargada de agregar a la matriz arrayFlavorsAll todos los sabores de todas las variedades. Además los ordena alfabéticamente.</p>	98

fillSelects()	Esta función se encarga de rellenar los selects de los sabores y los efectos de la página principal.	118
searchMarias()	Encargada de buscar e ir añadiendo en la matriz arrayMariasSearch todas las variedades que coinciden con el efecto, sabor y razas seleccionados por el usuario.	146
clickButtonSearch()	Esta función se ejecuta cuando se clicla en el botón SEARCH de la pantalla principal y ejecuta las funciones: searchMarias() generateCardsMarias()	180
generateDescMarias()	Para entender la utilidad de esta función primero es necesario entender lo siguiente. La llamada a la API donde obtengo todas las variedades me devuelve de cada variedad su id, nombre, raza, sabores y efectos. Pero no me devuelve su descripción. El punto final de la API para obtener todas las variedades es uno, pero el punto final de la API para obtener la descripción de una variedad en concreto es otro, esto me obliga a realizar un bucle sobre todas las variedades y en cada iteración hacer una llamada al punto final de la API para la variedad de esa iteración en concreto, añadiendo a mi objeto ya creado la única propiedad que le falta y que no obtuvo en la primera llamada a la API, la descripción. Así que esta función se encarga de añadir las descripciones a las variedades.	203

generateCardsMarias()	<p>Esta función ya está explicada en el apartado de requisitos, concretamente en la parte de creación de elementos en el DOM. Solo me queda una cosa que explicar y es que esta función se usa tanto para generar las variedades encontradas según los requisitos del usuario como para generar solo las variedades agregadas a favoritos. Si se le pasa cualquier argumento generará las variedades favoritas, y si no, generará las variedades que coinciden con la búsqueda del usuario</p>	237
validateForm()	<p>Esta función valida el formulario de inicio, para ello hace uso de tres funciones internas: validateNom(), validateEdad(), validateEmail(). Las cuales validan mediante expresiones regulares que los campos introducidos sean los deseados. Además cada una de estas funciones se encarga de mostrar u ocultar los errores para informar al usuario y de retornar un valor true en caso de que haya errores o false en caso de que no hayan. De esta manera al final de la función validateForm() si las tres funciones internas me devuelven false significará que el formulario es valido y entonces estableceremos el valor de las cookies y rellenaremos las pestañas del menú y el footer con las funciones fillTextContentTabsNav() filltextContentFooter().</p> <p>Después de esto pasaremos a la pantalla principal.</p>	353

validateFormWithKeyEnter()	Esta función está explicada en el apartado de requisitos, concretamente en el requisito de eventos de teclado. Si se pulsa la tecla enter ejecutaremos la función validateForm()	498
backToTop()	Se ejecuta cuando clicamos en el botón en forma de flecha que podemos encontrar en la esquina inferior derecha, y lo único que hace es volver al inicio de la página.	507
setCookies()	A esta función podemos pasarle un o dos argumentos. Si le pasamos solo el primero(los días para que expire la cookie) se establecerán las cookies: nombre, age, email y restartcookies con la fecha de expiración que terminará en los días que le hayas pasado. Mientras que si le pasamos el segundo (una cookie con su valor. Ejemplo: cookieName=Alberto) se guardará esa cookie también con la fecha de expiración con los días que le hayamos pasado. Utilizo esta función con dos argumentos siempre para establecer el valor de la cookie favorites (en la que se guardan los ides de las variedades favoritas) cada vez que el usuario agrega o elimina una variedad a favoritos.	515
setCookieRestartCookies()	También con dos argumentos, en este caso ambos necesarios siempre. El primer argumento son los días en los que queremos que expire la cookie restartcookies y el segundo debe ser un string 'si' o 'no' el cual será el valor de la cookie;	541

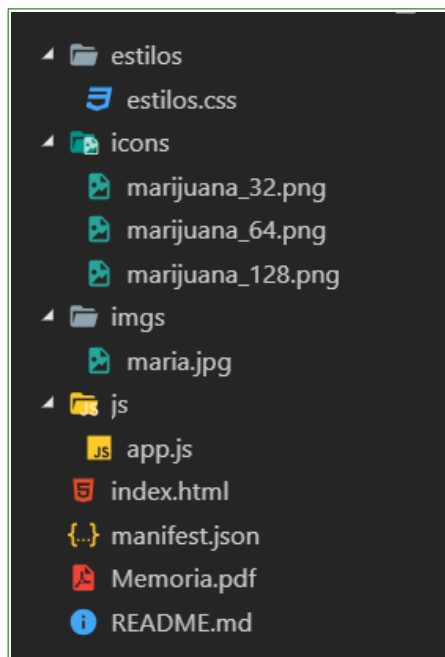
<code>getCookie()</code>	Esta función está sacada de la página w3schools y nos sirve para obtener el valor de una cookie pasándole su nombre.	553
<code>showAndHiddenElements()</code>	Esta función se encarga de mostrar u ocultar elementos de la página como la barra de navegación y el botón para volver al inicio de la página. Dicha función se llama dentro del evento scroll	592
<code>addEvents()</code>	Función usada para añadir el evento doble click a las variedades. Esta función se ejecuta una vez se han generado todas las descripciones de las variedades	617
<code>addAndremoveFavorite()</code>	Encargada de poner en verde o gris el color del icono que indica si está agregada a favoritas y de llamar a la función addAndRemoveFavoritesCookies() que es la que realmente las elimina o agrega.	631
<code>addAndremoveFavoritesCookies()</code>	A esta función se le pueden pasar uno o dos argumentos. El primero siempre es la id del icono de la variedad que se quiere eliminar o añadir. El segundo es opcional y si no se le pasa se borrará la variedad de favoritos, y si se le pasa un argumento la añadirá a favoritos. Cada vez que se usa esta función se llama a la función setCookies() estableciendo un nuevo valor para la cookie favorites	654
<code>getFavorites()</code>	Obtiene el valor actual de la cookie favorites y se lo pasa a la matriz global arrayFavoriteCookies	674

animar()	Te habrás dado cuenta a lo largo del código que se hace uso de esta función para hacer desaparecer, aparecer y animar elementos. Esta función hace uso de la librería animate.css, la cual nos proporciona un montón de clases css para animar elementos de nuestra página. Lo que hace esta función es animar el elemento(agregandole clases y quitandoselas en el momento oportuno) que le pasemos(primer argumento) con el efecto que queramos(segundo argumento). Y opcionalmente si vamos a usar una animación para hacer desaparecer el elemento tenemos que pasarle un tercer argumento	685
fillTextContentTabsNav()	Podría haber puesto el contenido de las pestañas directamente en el html, sin embargo esto no era posible ya que yo quería mostrar información de acuerdo a la hora o el usuario por ejemplo. La pestaña de HOW IT WORKS te dará los buenos días si son entre 00 y las 14, te dará las buenas tardes si son entre las 15 y las 20 y de lo contrario te dará las buenas noches. La pestaña de USUARIO lleva tu nombre. Por eso el contenido lo relleno desde JavaScript y para eso sirve esta función.	702
filltextContentFooter()	Exactamente por el mismo motivo que la función de arriba. Necesito mostrar el año y no quiero tener que cambiarlo manualmente el año que viene. Por lo que obtengo el año y contenido que quiero en mi footer desde esta función.	737

showAndHiddenTab()	<p>Esta función tuve que crearla porque me di cuenta después de haber estado usando una barra de navegación de Bootstrap de que esta no cerraba las pestañas una vez abiertas. Podía clicar otras pestañas pero siempre había alguna abierta. Entonces me di cuenta de que la manera que tiene Bootstrap para abrir o cerrar pestañas de navegación es agregandoles o quitandoles las clases 'active' y 'show'. Así que esta función se encarga de buscar en las clases de la pestaña clicada las clases 'active' y 'show' y si las tiene se las elimino, cerrando la pestaña clicada.</p>	749
signOut()	<p>Esta función se encarga de poner el valor de la cookie restartcookies en si y recargar la página llevandote al formulario de inicio.</p>	776
generateArrayFavs()	<p>Esta función obtiene todos los ids de las variedades favoritas desde la matriz arrayFavoritesCookies separando el texto de la id(Por ejemplo la id original tiene el valor hicon46 del cual a mi solo me interesa el 46 que es el id de la variedad) y los almaceno en una matriz llamada arrayIdFavorites. Después itera la matriz donde se guardan todas las variedades(arrayMarías) y va guardando en otra matriz(arrayFavorites) las variedades que coinciden con los ids obtenidos anteriormente(arrayIdFavorites). Finalmente devuelve la matriz arrayFavorites con todas las favoritas</p>	788

addEventToShowFavorites()	Esta función se encarga de añadir el evento keydown para cuando se pulsen las teclas Ctrl+I o Ctrl+L ya explicado en la parte de los requisitos. Esta función se llama una vez se han generado todas las descripciones de las variedades.	822
----------------------------	---	---------------------

ESTRUCTURA DEL PROYECTO



En la raíz se encuentran cuatro archivos:

index.html: el cual contiene el html de la página principal, el formulario de inicio y el preloader.

manifest.json: el cual es utilizado en dispositivos móviles para, entre otras cosas, agregar la página al menú de inicio y asignarle unos iconos para que de la sensación de que sea una app nativa en lugar de una página web.

Memoria.pdf: Memoria del proyecto.

README.md: archivo de markdown creado para dar información sobre el proyecto a los visitantes del mismo en GitHub.

En la carpeta **estilos** encontramos el archivo **estilos.css** en el cual defino mis estilos personalizados.

En la carpeta **icon** encontramos tres iconos iguales pero de distintos tamaños que son usados por el **manifest.json**.

En la carpeta **imgs** tenemos la imagen **maria.jpg** que se usa como background en la página principal.

Y en la carpeta **js** tenemos el archivo **app.js** el cual contiene todo el JavaScript de la página web.

CONCLUSIÓN Y OTROS DETALLES

- Para obtener una experiencia optima al usar mi página web me gustaría que la abrieras desde el servidor de GitHub:

https://dawalberto.github.io/proyecto_maria/ y no desde local. Si aún así prefieres abrirla en local hazlo desde Firefox ya que en Chrome las cookies no funcionan como uno desearía.

- La razón de que toda la página web esté en inglés es porque todos los datos de la API están en inglés. Y obviamente me resulta mucho más cómodo escribir el resto de la página en Inglés que tener que traducir todos los datos de la API.

- Si el proyecto no fuese solo de entorno cliente hubiese implementado un servidor pudiendo crear cuentas de usuario para poder guardar tus variedades favoritas siempre en tu cuenta y para que el correo tenga algún sentido.

- Me hubiese gustado poder hacer mas tipos de filtrados y obtener más detalles para cada variedad como por ejemplo el lugar de donde proviene la variedad, tiempo aproximado de plantación, imagen ...etc

Pero estoy limitado a los datos que me ofrece la API que uso, que en este caso son el id, nombre, raza, efectos, sabores y descripción.

Página de la API: <http://strains.evanbusse.com//>

- Casi todo el estilo de la página está realizado con Bootstrap 4.2.1

Página de Bootstrap: <https://getbootstrap.com/>

- Los iconos de marihuana usados para identificar las variedades favoritas son de Font Awesome

Página de Font Awesome: <https://fontawesome.com/>

- Las animaciones están hechas con la librería animate.css

Página de Animate.css: <https://daneden.github.io/animate.css/>