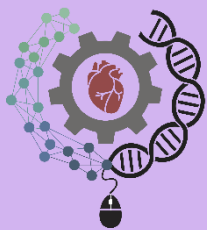




ESCUELA
POLITÉCNICA
SUPERIOR

Práctica 5.



Grado en Ingeniería Biomédica

Glucómetro.

Esther Bermejo.

Ingeniería Biomédica. 2023-2024



Tabla de contenido

Tabla de contenidos de imágenes.....	2
Introducción	4
Funcionamiento del glucómetro	4
Manejo de errores. Función: DatosValidos()	5
Funciones, estructura y ejecución del código fuente.	8
Funciones	9
Estructura del código fuente	12
Diseño	13
Ejecución	15
Ejemplos de ejecución	15
Conclusión	18
Bibliografía.	18

Tabla de contenidos de imágenes.

Ilustración 1. Tabla Criterios Diagnóstico.....	4
Ilustración 2. Función datos validos parte I.....	5
Ilustración 3. Función datos validos parte II.....	5
Ilustración 4. Alerta valor no válido DNI.....	6
Ilustración 5. Alerta campo DNI vacío.....	7
Ilustración 6. Alerta campo nombre vacío.....	7
Ilustración 7. Alerta valor no válido para nombre.....	8
Ilustración 8. Alerta valor no válido para fecha.....	8
Ilustración 9. Función Edad().....	9
Ilustración 10. Función Sexo().....	9
Ilustración 11. Función NivelGlucosa().....	10
Ilustración 12. Función ImprimirRegistro().....	10
Ilustración 13. Función Reset().....	11
Ilustración 14. Función Diagnóstico().....	11
Ilustración 15. Estructura página web I.....	12
Ilustración 16. Estructura página web II.....	13
Ilustración 17. Interfaz sin css I.....	13
Ilustración 18. Interfaz sin css II.....	13
Ilustración 19. Interfaz con css I.....	14
Ilustración 20. Interfaz con css II.....	14
Ilustración 21. Interfaz de introducción de datos.....	15
Ilustración 22. Diagnostico hiperglucemia I.....	15
Ilustración 23. Diagnóstico hiperglucemia II.....	16
Ilustración 24. Diagnostico normoglucemia I.....	16
Ilustración 25 Diagnostico normoglucemia II.....	16
Ilustración 26. Diagnostico normoglucemia III.....	17
Ilustración 27. Diagnóstico hipoglucemia I.....	17



Ilustración 28. Diagnóstico hipoglucemia II.....	17
Ilustración 29. Diagnostico hipoglucemia III	18

Introducción

Esta práctica es la número 5 del curso 2023-2024 de la asignatura de Infraestructura de sistemas clínicos del grado de Ingeniería Biomédica en la Universidad de Alicante. Se abordan diferentes aspectos relacionados con el lenguaje de programación Javascript con el objetivo de aprender a utilizarlo dentro de páginas webs desarrolladas con el lenguaje de marcado HTML. Este ejercicio está orientado además a afianzar los conocimientos anteriormente adquiridos. Tanto el código fuente como el diseño con el archivo .css se ha desarrollado en Visual Studio Code, se adjunta posteriormente volcados de pantalla del código fuente y ejemplos de ejecución tras a ver sido explicado el desarrollo del código fuente.

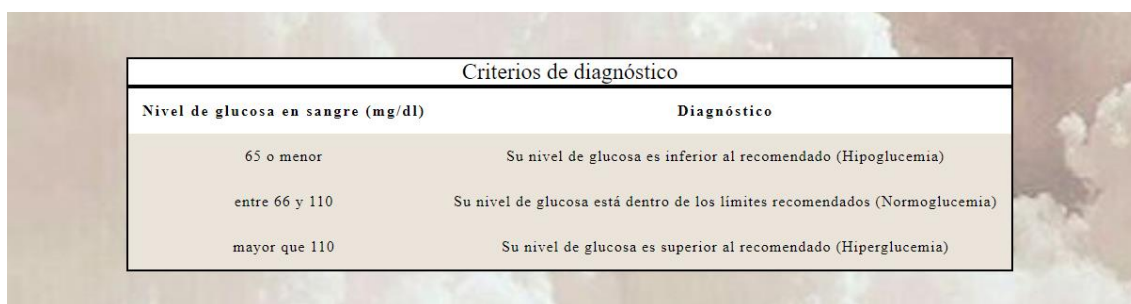
La memoria de esta práctica en concreto se va a organizar de la siguiente forma:

- Breve introducción sobre el funcionamiento del glucómetro.
- Manejo de errores.
- Funciones, estructura y ejecución del código fuente.
- Ejemplos de ejecución.

Se puede consultar una [guía de las ilustraciones](#) y del [contenido del documento](#).

Funcionamiento del glucómetro

El glucómetro se presenta en primer lugar con una interfaz en la que el usuario debe introducir: DNI(sólo números), nombre, apellidos, fecha de nacimiento, sexo biológico y nivel de glucosa en sangre. Una vez se han introducido todos los datos, debajo de estos haciendo 'click' sobre el botón diagnóstico se mostrará un cuadro de texto en el que se mostrará si el usuario tiene hipoglucemia, normoglucemia o hiperglucemia según la Ilustración 1. Esta tabla es en la que se el programa para realizar el diagnóstico y está presente en la página al final.



Criterios de diagnóstico	
Nivel de glucosa en sangre (mg/dl)	Diagnóstico
65 o menor	Su nivel de glucosa es inferior al recomendado (Hipoglucemia)
entre 66 y 110	Su nivel de glucosa está dentro de los límites recomendados (Normoglucemia)
mayor que 110	Su nivel de glucosa es superior al recomendado (Hiperglucemia)

Ilustración 1. Tabla Criterios Diagnóstico.

Cabe mencionar que las entradas en todos los campos son obligatorias, además el DNI del usuario tiene que ser introducido sin letra y el valor de la glucosa debe ser un número entero. Sin embargo para introducir el nombre y los apellidos del usuario se pueden introducir en mayúsculas, minúsculas, nombres compuestos y más de un apellido.

Manejo de errores. Función: DatosValidos()

El código fuente a partir del cual se implementa el ejercicio anterior está estructurado en funciones tal y como se explicará en [funciones, estructura y ejecución del código fuente](#). Por otro lado para que la ejecución del código fuente dé lugar al resultado esperado se han de manejar los errores que pueden surgir en la introducción de los datos por pantalla para realizarlo se ha definido la función `DatosValidos()`.

```

13 function DatosValidos(){ //Los datos son validos devuelve true
14     var dni = document.getElementById("dni") //DNI:
15     var name = document.getElementById("name") //Nombre:
16     var surname = document.getElementById("surname") //Apellidos:
17     var date_born = document.getElementById("date_born") //Fecha de nacimiento:
18     var sexo = document.getElementsByName("sexo"); //Sexo
19     var glucosa = document.getElementById("glucosa") //Nivel de glucosa
20     var datos_validos=true; //variable resultado
21
22     var sololetras = /^[A-Za-zÁÉÍÓÚáéíóú\s-]*$/;
23     var soloNumeros = /^[0-9]*$/;
24     if (dni.value.trim() == ""){ //Se comprueba que el valor del DNI es un campo no vacío
25         dni.focus() //Se pone el foco en el campo DNI
26         alert("El campo DNI no puede estar vacío.") //Alerta correspondiente
27         datos_validos = false;
28     }
29     else if(!soloNumeros.test(dni.value.trim())){//Controlar valor del DNI
30         dni.focus() //Se pone el foco en el campo DNI
31         alert("Introduzca un valor válido DNI") //Alerta correspondiente
32         datos_validos = false;
33     }
34     else{ //Si el DNI está correcto sigo evaluando el siguiente dato
35         if (!sololetras.test(name.value.trim())) { //se comprueba que el nombre introducido es de tipo string
36             alert("El valor del nombre no es un valor válido"); //Alerta correspondiente
37             name.focus(); //Se pone el foco en el campo Nombre
38             datos_validos = false;
39         }
40         else if (name.value.trim() == "") { //se comprueba que el campo nombre no este vacío
41             alert("El campo de nombre no puede estar vacío"); //Alerta correspondiente
42             name.focus(); //Se pone el foco en el campo Nombre
43             datos_validos = false;
44         }
45         else{ // Si el campo nombre está correcto sigo evaluando
46             if (!sololetras.test(surname.value.trim())) { //se comprueba que los datos de surname sean correctos
47                 alert("El valor de los apellidos no es un valor válido"); //Alerta correspondiente
48                 surname.focus() //Se pone el foco en el campo surname
49                 datos_validos=false;
50             }
51         }
52     }
53 }

```

Ilustración 2. Función datos validos parte I.

```

30     else if (surname.value.trim() === "") { //comprobacion de que el campo apellidos no esté vacío
31         alert("El campo de apellidos no puede estar vacío"); //Alerta correspondiente
32         surname.focus(); //Se pone el foco en el campo surname
33         datos_validos = false;
34     }
35     else { //Si el campo surname está correcto sigo evaluando. Se va a evaluar si los datos introducidos en la fecha de nacimiento son correctos
36         if ((new Date()-new Date(date_born.value))<=0){ //Se compara que la diferencia entre la fecha actual y la introducida no sea negativa
37             alert("La fecha de nacimiento no es válida") //Alerta correspondiente
38             date_born.focus() //Se pone el foco en el campo fecha de nacimiento
39             datos_validos=false;
40         }
41         else{ //Si el campo fecha de nacimiento está correcto sigo evaluando
42             if ((lsexo[0].checked) && (lsexo[1].checked)) { //comprobacion de que el campo sexo esté marcado
43                 alert("Por favor completa el campo sexo.") //Alerta correspondiente
44                 datos_validos=false;
45             }
46             else{ //Si el campo sexo está correcto sigo evaluando
47                 if (glucosa.value.trim() === ""){ //Se comprueba que el valor de la glucosa no es vacío
48                     alert("Lo siento si no introduce sus valores de glucosa en sangre no podemos realizar un diagnóstico") //Alerta correspondiente
49                     glucosa.focus(); //Se pone el foco en el campo de la glucosa
50                     datos_validos = false;
51                 }
52                 else if((glucosa.value<0) || (glucosa.value>1000) || (!soloNumeros.test(glucosa.value.trim()))){ //Controlar el rango de la glucemia
53                     alert("Introduzca un valor válido de glucosa en sangre")
54                     datos_validos = false;
55                 }
56             }
57         }
58     }
59 }
60
61 return datos_validos;
62
63 }
64

```

Ilustración 3. Función datos validos parte II

Desde la línea 13 hasta la 19 se obtienen los datos introducidos en los campos del Glucómetro y posteriormente se define la variable `datos_validos = true`. Esta función devuelve un valor booleano `true` o `false` que será el valor de la variable anterior según si absolutamente todos los datos introducidos son correctos, es decir, han introducido datos con valor y tipo adecuado.

Se analiza cada entrada de forma que en cadena se comprueba que el valor y el tipo de dato introducido es correcto. Se hace en orden y si algún dato resulta erróneo salta una alerta explicando el error producido, además se pone el foco el campo determinado y la función devuelve el valor `false`. Únicamente si esta función devuelve el valor `true` se realiza el diagnóstico y se imprime el registro, de lo contrario después de salir la alerta informando del error no ocurre nada.

Se comprueba en primer lugar que el valor del campo 'DNI' no esté vacío y en segundo lugar que sea un valor válido, es decir, que sean números, si no está vacío y es un valor válido se pasa a examinar el campo 'Nombre', para este se comprueban las mismas condiciones que el campo no esté vacío y que el valor sean letras, si se cumple a su vez se evalúan las mismas condiciones para el campo 'Apellidos'. Así sucesivamente para el campo de fecha de nacimiento, sexo y glucosa. En la fecha de nacimiento se evalúa que la fecha no sea una fecha anterior a la actual, en el campo de sexo se evalúa que haya un campo marcado y en el de la glucosa sea un número y que este a su vez no sea ni mayor de 1000 ni negativo. Algunas de las alertas definidas para los errores ya mencionados se muestran a continuación.

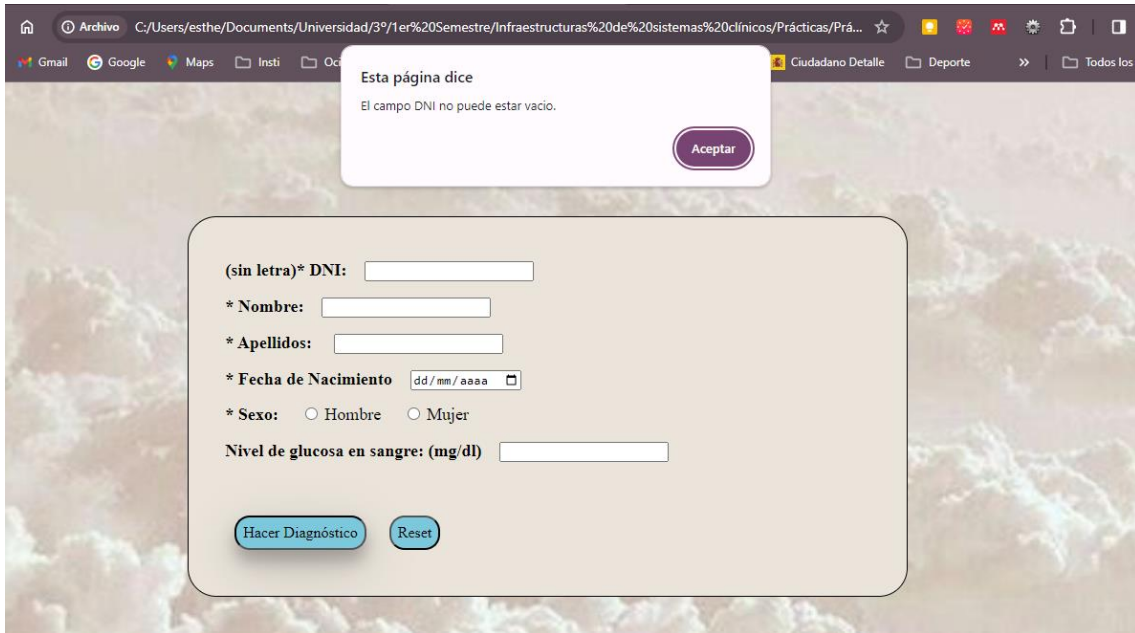


The screenshot shows a web browser window with a URL bar containing a file path. A modal alert box is displayed in the center, titled "Esta página dice" (This page says) with the message "Introduzca un valor válido DNI" (Enter a valid DNI value) and an "Aceptar" (Accept) button. Below the alert, a form is visible with the following fields and labels:

- (sin letra)* DNI:
- * Nombre:
- * Apellidos:
- * Fecha de Nacimiento:
- * Sexo: ☐ Hombre ☐ Mujer
- Nivel de glucosa en sangre: (mg/dl)

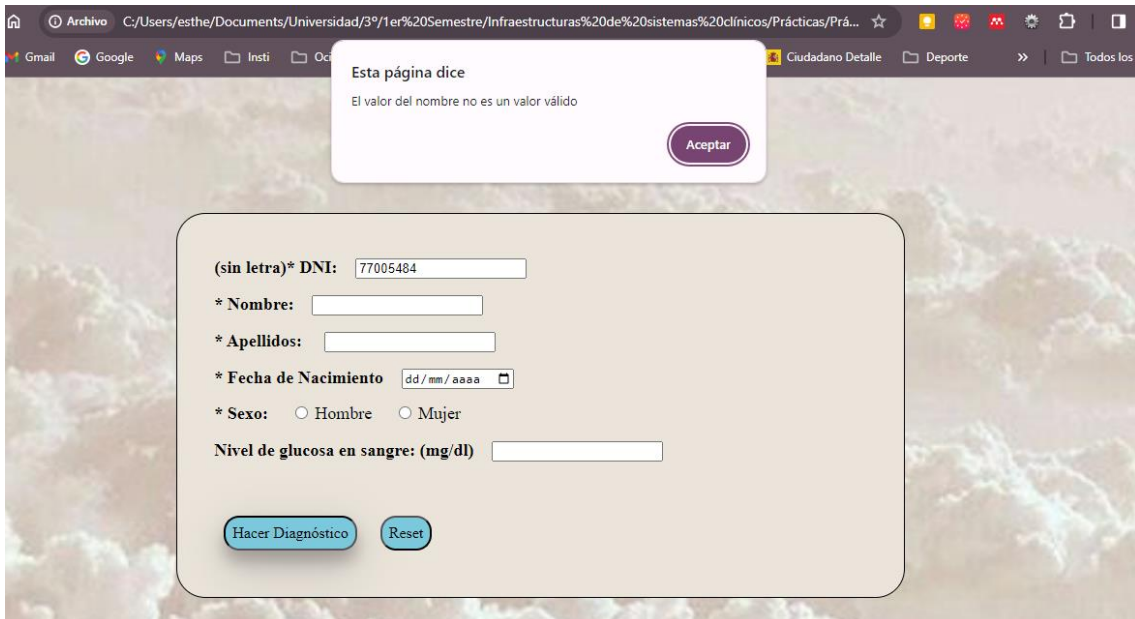
At the bottom of the form are two buttons: "Hacer Diagnóstico" (Make Diagnosis) and "Reset".

Ilustración 4. Alerta valor no válido DNI.



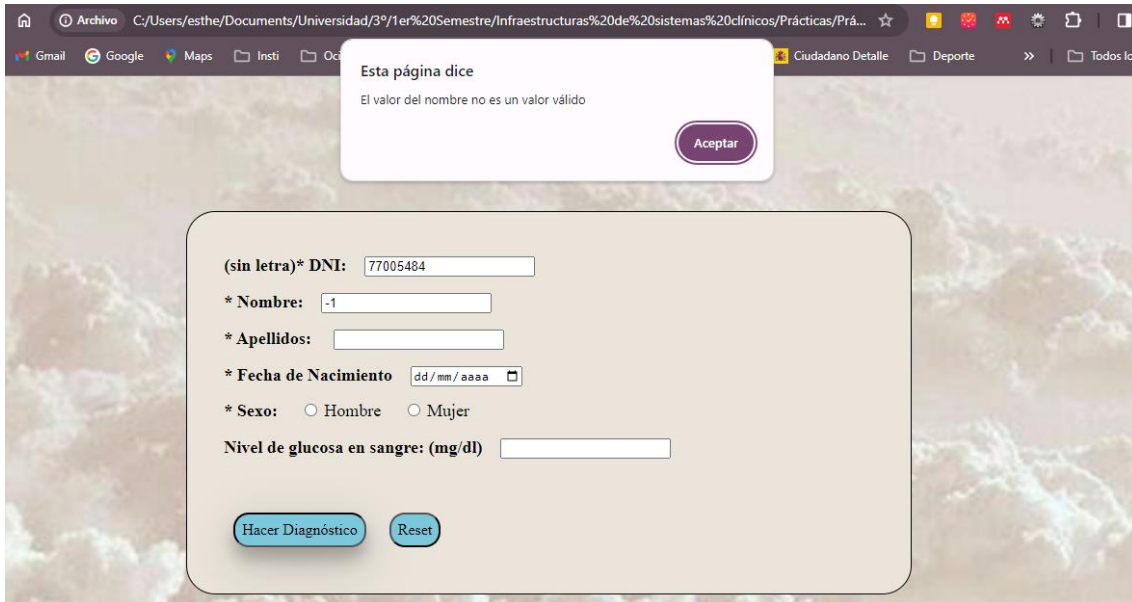
The screenshot shows a web browser window with a URL bar containing a file path. A modal dialog box is displayed in the center, titled "Esta página dice" (This page says), with the message "El campo DNI no puede estar vacío." (The DNI field cannot be empty). Below the message is an "Aceptar" (Accept) button. The background form is partially visible, showing fields for DNI, Name, Surname, Birth Date, Sex, and Blood Glucose Level, along with "Hacer Diagnóstico" (Make Diagnosis) and "Reset" buttons.

Ilustración 5. Alerta campo DNI vacío.



The screenshot shows the same web browser window, but the modal dialog box now displays the message "El valor del nombre no es un valor válido" (The name value is not a valid value). The "Aceptar" button is still present. In the background form, the DNI field now contains the value "77005484". The other fields (Name, Surname, Birth Date, Sex, Blood Glucose Level) and buttons ("Hacer Diagnóstico", "Reset") remain the same.

Ilustración 6. Alerta campo nombre vacío.



Esta página dice
El valor del nombre no es un valor válido

Aceptar

(sin letra)* DNI: 77005484

* Nombre: -1

* Apellidos:

* Fecha de Nacimiento dd/mm/aaaa

* Sexo: ☐ Hombre ☐ Mujer

Nivel de glucosa en sangre: (mg/dl)

Hacer Diagnóstico Reset

Ilustración 7. Alerta valor no válido para nombre.



Esta página dice
La fecha de nacimiento no es válida

Aceptar

(sin letra)* DNI: 77005484

* Nombre: esther

* Apellidos: bermejo

* Fecha de Nacimiento 04/05/2030

* Sexo: ☐ Hombre ☒ Mujer

Nivel de glucosa en sangre: (mg/dl) 50

Hacer Diagnóstico Reset

Ilustración 8. Alerta valor no válido para fecha.

Funciones, estructura y ejecución del código fuente.

Con el objetivo de estructurar el código fuente, hacerlo más eficiente y óptimo se han utilizado funciones, en concreto siete, la primera `DatosValidos()` ya se ha explicado anteriormente en [el manejo de errores](#), la siguientes son: `Edad()`, `Sexo()`, `NivelGlucosa()`, `ImprimirRegistro()`, `Reset()` y `Diagnostico()`. Los beneficios de estructurar el código en funciones han sido varios; ese hecho ha permitido reutilizar código llamando a unas funciones dentro de otras, aumentar la legibilidad, facilitar las pruebas en cada función por separado y facilitar también la depuración.

En la función de `Diagnostico()` según si el paciente tiene hiperglucemia, normoglucemia o hipoglucemia, aparece en el cuadro de texto valoración un diagnóstico u otro, además cabe mencionar

que si el paciente es mayor de 65 años, dato que se obtiene a partir de la función `Edad()`. Por otro lado se tiene la función auxiliar `Sexo()` que se utiliza en la función `ImprimirRegistro()` para imprimir el valor "femenino" o "masculino" según si se ha marcado un botón u otro. A continuación, se explica detalladamente cada función.

Funciones

Primero que todo se va a explicar la función `Edad()`, se obtiene el dato del campo de la fecha de nacimiento y se crea un nuevo objeto tipo `Date()` con el que se obtiene la fecha actual. Para restar ambas fecha y obtener la diferencia se crea otro objeto `Date()` para la fecha de nacimiento y después se restan, cabe mencionar que el resultado de esta diferencia se devuelve en milisegundos, se pasa a días y después a años. La diferencia entre la fecha actual y la fecha de nacimiento de la persona resulta la edad actual de la persona en años cumplidos. Este es el valor que devuelve la función.

```
85 function Edad() { //Devuelve la edad actual del paciente
86 //Primero se consigue conocer la edad de la persona
87 var date_born = document.getElementById("date_born").value;
88
89 // Crear un nuevo objeto Date
90 var fechaActual = new Date();
91
92 // Obtengo los componentes de la fecha
93 var dia = fechaActual.getDate(); //Se obtiene el dia completo
94 var mes = fechaActual.getMonth() + 1; // Sumo uno porque los meses en JavaScript van de 0 a 11
95 var anio = fechaActual.getFullYear(); //Se obtiene el año completo
96
97 // Formateamos la fecha
98 var fechaFormateada = dia + "-" + mes + "-" + anio;
99 var fechaNacimiento = new Date(date_born);
100
101 // Convertir las fechas a milisegundos para poder compararlas y calcular Edad en ms
102 var Diferencia_en_ms = fechaActual - fechaNacimiento;
103 var DiasTranscurridos = Diferencia_en_ms / (1000 * 60 * 60 * 24); //Se calcula en dias
104 var EdadActual = Math.floor(DiasTranscurridos / 365.25); //Se pasa a años
105 return EdadActual //Devuelve la edad actual de la persona teniendo en cuenta si ha cumplido ya o no años
106 }
```

Ilustración 9. Función `Edad()`.

Después se ha definido la función `Sexo()` en ella se obtiene los valores de los input de tipo radio y se comprueba cuál está marcado, si es el primero el sexo biológico que ha marcado el usuario es el que corresponde con masculino y si ha marcado el segundo el sexo biológico es femenino, entonces la función devuelve la cadena de texto correspondiente según si es masculino o si es femenino. La salida de esta función es un valor `string` que determina cuál es el valor del sexo biológico del usuario según la casilla que haya marcado.

```
108 function Sexo() { //Devuelve 'false' si el genero es masculino y 'true' si es femenino.
109 var sexo = document.getElementsByName("sexo"); //se obtiene el radio button con name="sexo"
110 if (sexo[0].checked) { //Si el primero está marcado que es el que corresponde con masculino
111     sexo_biologico="masculino"; //sexo biológico es masculino
112 }
113 else if (sexo[1].checked) { //Si el segundo está marcado que es el que corresponde con femenino
114     sexo_biologico="femenino"; //sexo biológico es femenino
115 }
116 return sexo_biologico
117 }
```

Ilustración 10. Función `Sexo()`

En tercer lugar, se ha definido la función más importante de todo el código fuente y esto es porque es en esta función en la que se determina cuál es el estado del paciente según su nivel de glucosa en sangre (mg/dl).

```

118
119
120+ function NivelGlucosa() { //Devuelve si es hiperglucemia, normoglucemia, hipoglucemia
121+   var glucosa = document.getElementById("glucosa").value; //Se obtiene los datos de glucosa introducidos por pantalla
122+   var estado_Paciente; //Se inicializa la variable estado paciente a la que se le asignará si es normoglucemia, hiperglucemia o hipoglucemia
123
124   if (glucosa > 110) { //Si el valor de la glucosa es mayor de 110mg/dl
125     estado_Paciente = "Hiperglucemia"
126   }
127   else if (glucosa <= 110 && glucosa >= 66) { //Si el valor de la glucosa esta entre 110mg/dl y 66mg/dl
128     estado_Paciente = "Normoglucemia"
129   }
130   else { //Si el valor de la glucosa es menor de 66mg/dl. No se controla si es positivo porque contaría como valor no válido y no se permitiría su introduccion
131     estado_Paciente = "Hipoglucemia"
132   }
133   return estado_Paciente //Devuelve el estado del paciente
134 }

```

Ilustración 11. Función NivelGlucosa()

Esta función obtiene el nivel de glucosa en sangre que introduce el usuario en la pantalla y si es mayor de 110 mg/dl se devuelve el estado del paciente de hiperglucemia, si es mayor de 66mg/dl y menor de 110mg/dl se devuelve el estado del paciente de normoglucemia y si es menor de 66mg/dl devuelve que el estado del paciente es hipoglucemia. Cabe destacar que se ha establecido un rango para los valores de la glucemia en el manejo de errores, este valor debe de estar dentro del intervalo [0,1000] para que el valor de la glucosa sea un valor válido.

Otra parte esencial del funcionamiento del glucómetro es el registro de las entradas, se ha definido una función para imprimir el registro de entradas y a la cuál se llama dentro de la función Diagnóstico() entre otras, que es la que se activa al darle al botón “Diagnóstico”.

```

135
136
137 function ImprimirRegistro() { //Imprime el registro por pantalla
138+   var registro = document.getElementById("registro"); //Se obtiene el elemento contenedor registro
139+   var datos = [ //se crea una lista para poder almacenar todos los datos del paciente
140+     "Nombre:", document.getElementById("name").value, //Se obtiene el valor de el NOMBRE
141+     "Apellidos:", document.getElementById("surname").value, //Se obtiene el valor de los APELLIDOS
142+     "Fecha de nacimiento:", document.getElementById("date_born").value, //Se obtiene el valor de la FECHA DE NACIMIENTO
143+     "Edad:", Edad(), //Se obtiene el valor de la EDAD
144+     "Sexo:", Sexo(), //Se obtiene el valor de el SEXO
145+     "Nivel de glucosa:", document.getElementById("glucosa").value, //Se obtiene el valor de la GLUCOSA
146+     "Valoración:", document.getElementById("valoracion").value //Se obtiene el valor de la VALORACION
147+   ];
148   var titulo = document.createElement("h3"); //Se crea un elemento titulo que será el valor del DNI para que cada conjunto de datos quede identificado
149   titulo.textContent = "Registro - DNI: " + document.getElementById("dni").value; // Añadir el valor del DNI al título
150   registro.appendChild(titulo) //Se añade el título al contenedor
151
152   var lista = document.createElement("ul"); //se crea un elemento lista en html donde se tendrán todos los registros
153   for (var i = 0; i < datos.length; i += 2) { //se recorre la lista con los datos almacenado
154     var dato = datos[i]; //Se almacena la etiqueta junto a su valor correspondiente
155     var valor = datos[i + 1];
156
157     var elementoLi = document.createElement("li"); //Se crea una lista
158     elementoLi.innerHTML = "<u>${dato}</u>: ${valor}"; //se añade en formas de pareja 'etiqueta' y 'valor' mostrando la etiqueta subrayada
159     lista.appendChild(elementoLi); //se añade cada dato como elemento de la lista para ese paciente
160   }
161   registro.appendChild(lista); //Se añade la lista que contiene todo el conjunto de datos al contenedor registro
162 }

```

Ilustración 12. Función ImprimirRegistro()

Esta función es más compleja que las anteriores y funciona de la siguiente manera. Se obtiene el contenedor registro donde se publicará una lista correspondiente a cada entrada que contendrá los datos de cada paciente. Se crea una lista y se añade cada pareja de dato y valor, consiguiendo la edad y el sexo con las funciones auxiliares anteriormente definidas. Posteriormente se crea un elemento título de html cuyo contenido será “Registro – DNI: dniUsuario” y después se añade ese título al contenedor registro. Tras eso se crea la lista que contendrá todos los datos en el contenedor, y se recorre la lista que almacena las parejas etiqueta y valor para introducir cada pareja dentro de un elemento de la lista de html. Una vez se ha recorrido la lista, se añade al contenedor registro y el resultado será tener dentro de un contenedor tantas listas como veces se haya dado al botón diagnóstico

que contenga cada pareja etiqueta y valor como un solo elemento, además mostrando la etiqueta subrayada para distinguir con mayor facilidad etiqueta y contenido.

No solo se tiene en pantalla el botón para realizar el diagnóstico también incluye un botón para hacer `Reset()` y esta funcionalidad está definida con una función con el mismo nombre. En esta se limpian todos los campos igualando a cadenas vacías y los campos de tipo radio se igualan al valor `false` para que estén desmarcados. Cada vez que se hace 'click' en el botón `Reset` desaparece el campo valoración y aparece cuando se realiza un diagnóstico con el botón `Diagnóstico`.

```
164 + function Reset() { //Limpia todos los campos del formulario
165     document.getElementById("dni").value = "";
166     document.getElementById("name").value = "";
167     document.getElementById("surname").value = "";
168     document.getElementById("date_born").value = "";
169     document.getElementById("hombre").checked = false;
170     document.getElementById("mujer").checked = false;
171     document.getElementById("glucosa").value = "";
172     document.getElementById("valoracion").value = "";
173     document.querySelector('label[for="valoracion"] strong').style.display = 'flex'; //No se muestra en la pantalla la etiqueta valoracion
174     document.getElementById("valoracion").style.display="none";
175 }
176
```

Ilustración 13. Función `Reset()`

Tal y como se ha ido mencionando la función `Diagnóstico()` es bastante completa se desglosará a continuación:

```
177 function Diagnostico(){
178     var edad;
179     var estado_Paciente;
180     var valoracion = document.getElementById("valoracion")
181
182     if(DatosValidos()==true){ //Si los datos introducidos por el usuario son correctos
183         estado_Paciente=NivelGlucosa();//Se determina si el paciente tiene hiperglucemia, normoglucemia, o hipoglucemia
184 +         edad = Edad(); //Se determina la edad del paciente
185         document.querySelector('label[for="valoracion"]').style.display = 'inline'; //Se muestra en la pantalla la etiqueta valoracion
186         valoracion.style.display="block"; //Se muestra en la pantalla el campo de texto donde se podrá ver la valoración
187         switch(estado_Paciente){
188             case "Hiperglucemia":
189                 if (edad>=65){
190                     valoracion.value="El usuario al que corresponde los datos introducidos presenta hiperglucemia, dado que es usted mayor de 65 años entra usted en el
191                 }
192                 else{
193                     valoracion.value="El usuario al que corresponde los datos introducidos presenta hiperglucemia."
194                 }
195                 break;
196             case "Normoglucemia":
197                 if (edad>=65){
198                     valoracion.value="El usuario al que corresponde los datos introducidos presenta normoglucemia, dado que es usted mayor de 65 años entra usted en el
199                 }
200                 else{
201                     valoracion.value="El usuario al que corresponde los datos introducidos presenta normoglucemia."
202                 }
203                 break;
204             case "Hipoglucemia":
205                 if (edad>=65){
206                     valoracion.value="El usuario al que corresponde los datos introducidos presenta hipoglucemia, dado que es usted mayor de 65 años entra usted en el
207                 }
208                 else{
209                     valoracion.value="El usuario al que corresponde los datos introducidos presenta hipoglucemia."
210                 }
211                 break;
212         }
213         ImprimirRegistro();
214     }
215 }
```

Ilustración 14. Función `Diagnóstico()`

Sólo y únicamente cuando los datos introducidos son válidos, se obtiene el nivel de glucosa y se almacena en la variable `estado_Paciente`, análogamente se obtiene la edad del paciente y se almacena en la variable `edad`. Aparece la etiqueta y el campo de valoración para mostrar en él el texto correspondiente. Si la persona tiene más de 65 años el paciente se considera de riesgo. Los diagnósticos posibles son los siguientes:

- Hiperglucemia. El usuario al que corresponde los datos introducidos presenta hiperglucemia.
- Hiperglucemia mayor de 65 años. El usuario al que corresponde los datos introducidos presenta hiperglucemia, dado que es usted mayor de 65 años entra usted en el grupo de riesgo.

- Normoglucemia. El usuario al que corresponde los datos introducidos presenta normoglucemia.
- Normoglucemia mayor de 65 años. El usuario al que corresponde los datos introducidos presenta normoglucemia, dado que es usted mayor de 65 años entra usted en el grupo de riesgo.
- Hipoglucemia. El usuario al que corresponde los datos introducidos presenta hipoglucemia.
- Hipoglucemia mayor de 65 años. El usuario al que corresponde los datos introducidos presenta hipoglucemia, dado que es usted mayor de 65 años entra usted en el grupo de riesgo.

En la parte inferior se presenta el registro con la lista de datos de cada paciente identificada cada una con el DNI introducido tal y como se podrá apreciar en los [ejemplos de ejecución](#).

Estructura del código fuente

El código fuente ha seguido una estructura determinada, en primer lugar se ha tratado el manejo de errores, posteriormente se ha definido las funciones auxiliares de menor a mayor complejidad: Edad(), Sexo(), NivelGlucosa(), ImprimirRegistro(), y por último se han definido las funciones que se activan con los botones Reset() y Diagnóstico().

Los campos en los que se introducen los datos están agrupados en un contenedor de fieldset y se ha desarrollado con HTML no con JAVASCRIPT que es el lenguaje de programación que sí se ha utilizado para la implementación del código anterior.

La tabla en la que se informa de los criterios de diagnóstico se ha creado también con HTML.

```

218 <h1 id="titulo">GLUCÓMETRO</h1>
219 <fieldset>
220 <label for="dni"><strong>(sin letra)</strong> DNI:</label><input type="text" id="dni" name="dni" required><br><!--DNI-->
221 <label for="name"><strong>* Nombre:</strong></label><input type="text" id="name" name="name" required><br><!--Nombre-->
222 <label for="surname"><strong>* Apellidos:</strong></label><input type="text" id="surname" name="surname" required><br><!--Apellido-->
223 <label for="date_born"><strong>* Fecha de Nacimiento:</strong></label><input type="date" id="date_born" name="fecha_nacimiento" required><br><!--Fecha nacimiento-->
224 <label><strong>* Sexo:</strong></label>
225 <input type="radio" id="hombre" name="sexo">
226 <label for="hombre">Hombre</label>
227 <input type="radio" id="mujer" name="sexo">
228 <label for="mujer">Mujer</label><br>
229 <label for="glucosa"><strong>Nivel de glucosa en sangre: (mg/dl)</strong></label><input type="text" id="glucosa" required><br>
230 <label for="valoracion"><strong>Valoración:</strong></label><input type="text" id="valoracion" name="valoracion" required><br><!--Valoracion-->
231 <button onclick="Diagnostico()">Hacer Diagnóstico</button>
232 <button onclick="Reset()">Reset</button><!--Boton resetear-->
233 </fieldset>
234 <p>Aquellos campos que presentan un asterisco * son obligatorios</p><br>
235 <div class="registro" id="registro"><h2>Registro</h2>
236 <p>A continuación se presenta un registro de todos los datos introducidos previamente.</p>
237 </div>
238
239 <h2>¿Qué criterios se siguen para realizar el diagnóstico?</h2>
240 <p>Entre otros criterios podemos apreciar en la Tabla 1 los niveles preestablecidos para determinar
241 si un paciente presenta: hipoglucemia, normoglucemia o hiperglucemia.
242 Si el paciente es considerado de riesgo por la edad o por algún factor se notificará cuando se haga el diagnóstico.
243 </p>
244 <table id="tabla">
245 <caption>Criterios de diagnóstico</caption>
246 <tr>
247 <th scope="col">Nivel de glucosa en sangre (mg/dl)</th>
248 <th scope="col">Diagnóstico</th>
249 </tr>
250 <tr>
251 <td>65 o menor</td>
252 <td>Su nivel de glucosa es inferior al recomendado (Hipoglucemia)</td>
253 </tr>

```

Ilustración 15. Estructura página web I.

```
254      <tr>
255      <td>entre 66 y 110</td>
256      <td>Su nivel de glucosa está dentro de los límites recomendados (Normoglucemia)</td>
257      </tr>
258      <tr>
259      <td>mayor que 110</td>
260      <td>Su nivel de glucosa es superior al recomendado (Hiperglucemia)</td>
261      </tr>
262    </table>
263  </body>
264  <footer>
265    <p>Este recurso web está desarrollado por Esther Bermejo Martínez alumna de Ingeniería Biomédica.</p>
266    <p>Creado en noviembre de 2023. Universidad de Alicante.</p>
267  </footer>
268 </html>
```

Ilustración 16. Estructura página web II.

Diseño

La página web tiene un determinado diseño que se ha implementado con CSS3. La página sin css queda como se ve en la ilustración 17 y 18 e insertando al inicio esta línea de código `<link href="estilo.css" rel="stylesheet" media="screen">` queda tal y como se aprecia en la ilustración 19 y 20.

GLUCÓMETRO

(sin letra)* DNI:	<input type="text"/>
* Nombre:	<input type="text"/>
* Apellidos:	<input type="text"/>
* Fecha de Nacimiento:	<input type="text"/> dd/mm/aaaa <input type="checkbox"/>
* Sexo:	<input type="radio"/> Hombre <input type="radio"/> Mujer
Nivel de glucosa en sangre: (mg/dl)	<input type="text"/>
Valoración:	<input type="text"/>
<input type="button" value="Hacer Diagnóstico"/> <input type="button" value="Reset"/>	

Aquellos campos que presentan un asterisco * son obligatorios

Registro

A continuación se presenta un registro de todos los datos introducidos previamente.

¿Qué criterios se siguen para realizar el diagnóstico?

Entre otros criterios podemos apreciar en la Tabla 1 los niveles preestablecidos para determinar si un paciente presenta: hipoglucemia, normoglucemia o hiperglucemia. Si el paciente es considerado de riesgo por la edad o por algún factor se notificará cuando se haga el diagnóstico.

Ilustración 17. Interfaz sin css I

Aquellos campos que presentan un asterisco * son obligatorios

Registro

A continuación se presenta un registro de todos los datos introducidos previamente.

¿Qué criterios se siguen para realizar el diagnóstico?

Entre otros criterios podemos apreciar en la Tabla 1 los niveles preestablecidos para determinar si un paciente presenta: hipoglucemia, normoglucemia o hiperglucemia. Si el paciente es considerado de riesgo por la edad o por algún factor se notificará cuando se haga el diagnóstico.

Criterios de diagnóstico	
Nivel de glucosa en sangre (mg/dl)	Diagnóstico
65 o menor	Su nivel de glucosa es inferior al recomendado (Hipoglucemia)
entre 66 y 110	Su nivel de glucosa está dentro de los límites recomendados (Normoglucemia)
mayor que 110	Su nivel de glucosa es superior al recomendado (Hiperglucemia)

Este recurso web está desarrollado por Esther Bermejo Martínez alumna de Ingeniería Biomédica.

Creado en noviembre de 2023. Universidad de Alicante.

Ilustración 18. Interfaz sin css II

GLUCÓMETRO

(sin letra)* DNI:

* Nombre:

* Apellidos:

* Fecha de Nacimiento

* Sexo: ☐ Hombre ☐ Mujer

Nivel de glucosa en sangre: (mg/dl)

Hacer Diagnóstico

Reset

Aquellos campos que presentan un asterisco * son obligatorios

Ilustración 19. Interfaz con css I

Registro

A continuación se presenta un registro de todos los datos introducidos previamente.

¿Qué criterios se siguen para realizar el diagnóstico?

Entre otros criterios podemos apreciar en la Tabla 1 los niveles preestablecidos para determinar si un paciente presenta: hipoglucemia, normoglucemia o hiperglucemia. Si el paciente es considerado de riesgo por la edad o por algún factor se notificará cuando se haga el diagnóstico.

Criterios de diagnóstico	
Nivel de glucosa en sangre (mg/dl)	Diagnóstico
65 o menor	Su nivel de glucosa es inferior al recomendado (Hipoglucemia)
entre 66 y 110	Su nivel de glucosa está dentro de los límites recomendados (Normoglucemia)
mayor que 110	Su nivel de glucosa es superior al recomendado (Hiperglucemia)

Este recurso web está desarrollado por Esther Bermejo Martínez alumna de Ingeniería Biomédica.

Creado en noviembre de 2023. Universidad de Alicante.

Ilustración 20. Interfaz con css II

Ejecución

La ejecución se lleva a cabo haciendo ‘click’ sobre el botón Diagnóstico y después sobre el botón Reset.



GLUCÓMETRO

(sin letra)* DNI: 77002020

* Nombre: Esther

* Apellidos: Bermejo

* Fecha de Nacimiento: 04/08/2003

* Sexo: ☐ Hombre ☒ Mujer


Nivel de glucosa en sangre: (mg/dl) 150

Hacer Diagnóstico Reset

Ilustración 21. Interfaz de introducción de datos.

Ejemplos de ejecución

Se muestran ejemplos de ejecución. Al pulsar el botón ‘Hacer Diagnóstico’ para casos de hiperglucemia, normoglucemia e hipoglucemia mostrando la valoración y el registro respectivamente:



GLUCÓMETRO

(sin letra)* DNI: 77002020

* Nombre: Esther

* Apellidos: Bermejo

* Fecha de Nacimiento: 04/08/2003

* Sexo: ☐ Hombre ☒ Mujer

Nivel de glucosa en sangre: (mg/dl) 150

Valoración:
El usuario al que corresponde los datos introducidos presenta hiperglucemia.

Hacer Diagnóstico Reset

Ilustración 22. Diagnostico hiperglucemia I.



Ilustración 23. Diagnóstico hiperglucemia II.



Ilustración 24. Diagnostico normoglucemia I



Ilustración 25 Diagnostico normoglucemia II

• **Nivel de glucosa:** 150

• **Valoración:** El usuario al que corresponde los datos introducidos presenta hiperglucemia.

Registro - DNI: 77550021

• **Nombre:** Dani

• **Apellidos:** Bermejo

• **Fecha de nacimiento:** 1993-02-01

• **Edad:** 30

• **Sexo:** masculino

• **Nivel de glucosa:** 68

• **Valoración:** El usuario al que corresponde los datos introducidos presenta normoglucemia.

¿Qué criterios se siguen para realizar el diagnóstico?

Entre otros criterios podemos apreciar en la Tabla 1 los niveles preestablecidos para determinar si un paciente presenta: hipoglucemia, normoglucemia o hiperglucemia. Si el paciente es considerado de riesgo por la edad o por algún factor se notificará cuando se haga el diagnóstico.

Ilustración 26. Diagnostico normoglucemia III

GLUCÓMETRO

(sin letra)* **DNI:** 77884449

* **Nombre:** Paula

* **Apellidos:** Guilló Giménez

* **Fecha de Nacimiento:** 26 / 09 / 1940

* **Sexo:** ☐ Hombre ☒ Mujer

Nivel de glucosa en sangre: (mg/dl) 18

Valoración:

El usuario al que corresponde los datos introducidos presenta hipoglucemia, dado que es usted mayor de 65 años entra usted en el grupo de riesgo.

Hacer Diagnóstico **Reset**

Ilustración 27. Diagnóstico hipoglucemia I

Registro

A continuación se presenta un registro de todos los datos introducidos previamente.

Registro - DNI: 77002020

• **Nombre:** Esther

• **Apellidos:** Bermejo

• **Fecha de nacimiento:** 2003-08-04

• **Edad:** 20

• **Sexo:** femenino

• **Nivel de glucosa:** 150

• **Valoración:** El usuario al que corresponde los datos introducidos presenta hiperglucemia.

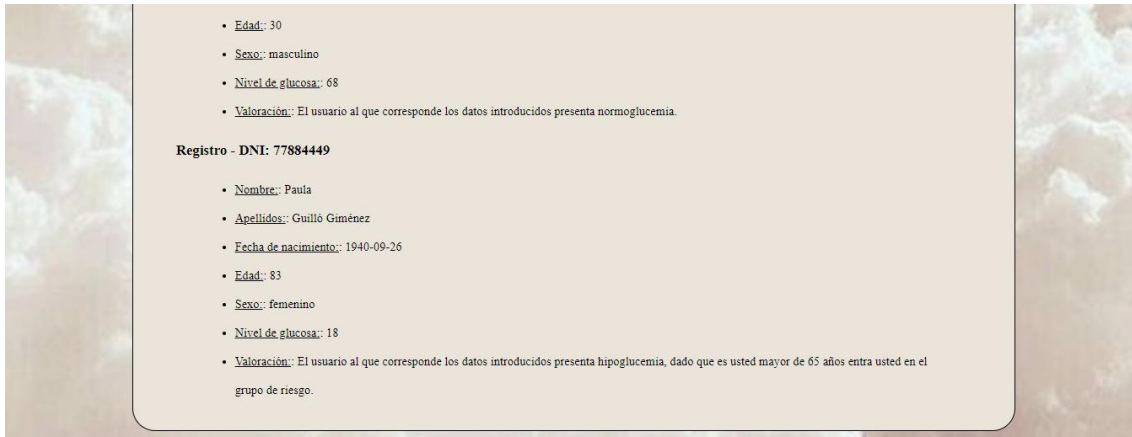
Registro - DNI: 77550021

• **Nombre:** Dani

• **Apellidos:** Bermejo

• **Fecha de nacimiento:** 1993-02-01

Ilustración 28. Diagnóstico hipoglucemia II



• **Edad:** 30

• **Sexo:** masculino

• **Nivel de glucosa:** 68

• **Valoración:** El usuario al que corresponde los datos introducidos presenta normoglucemia.

Registro - DNI: 77884449

• **Nombre:** Paula

• **Apellidos:** Guilló Giménez

• **Fecha de nacimiento:** 1940-09-26

• **Edad:** 83

• **Sexo:** femenino

• **Nivel de glucosa:** 18

• **Valoración:** El usuario al que corresponde los datos introducidos presenta hipoglucemia, dado que es usted mayor de 65 años entra usted en el grupo de riesgo.

Ilustración 29. Diagnostico hipoglucemia III

Conclusión

En síntesis, esta práctica ha consolidado los conocimientos previamente adquiridos en prácticas anteriores y ha proporcionado una comprensión más profunda de conceptos avanzados. Los objetivos establecidos se han alcanzado exitosamente, resultando en la creación de un glucómetro con control sobre los datos ingresados. Se han empleado los lenguajes HTML, JavaScript y CSS, abordando temas que van desde tipos de datos hasta bucles, pasando por la implementación de funciones y la extracción de datos desde HTML.

El enfoque principal ha sido la definición de funciones, explorando su aplicación y destacando su importancia en la estructuración, eficacia, eficiencia y optimización del código fuente. En consecuencia, esta práctica ha sentado una base sólida que abarca conocimientos básicos y, en cierta medida, avanzados sobre el desarrollo web, proporcionando confianza y habilidades específicas en el uso de JavaScript.

Bibliografía.

Se presentan a continuación todos los recursos que han sido consultados para el desarrollo de la práctica. Se ha consultado desde los apuntes proporcionados por el profesor de la asignatura hasta documentación oficial del propio lenguaje Javascript.

1. <https://www.w3schools.com/js/default.asp>
2. <https://www.w3schools.com/html/default.asp>
3. <https://www.w3schools.com/css/default.asp>
4. <https://developer.mozilla.org/es/docs/Web/JavaScript>
5. [Javascript 1 Lenguaje.pdf](#)
6. <https://devdocs.io/javascript/>