Trabajo voluntario

**José López Arribas**

Grado Superior en Desarrollo de Aplicaciones Web

Desarrollo Web en Entorno Cliente

Índice

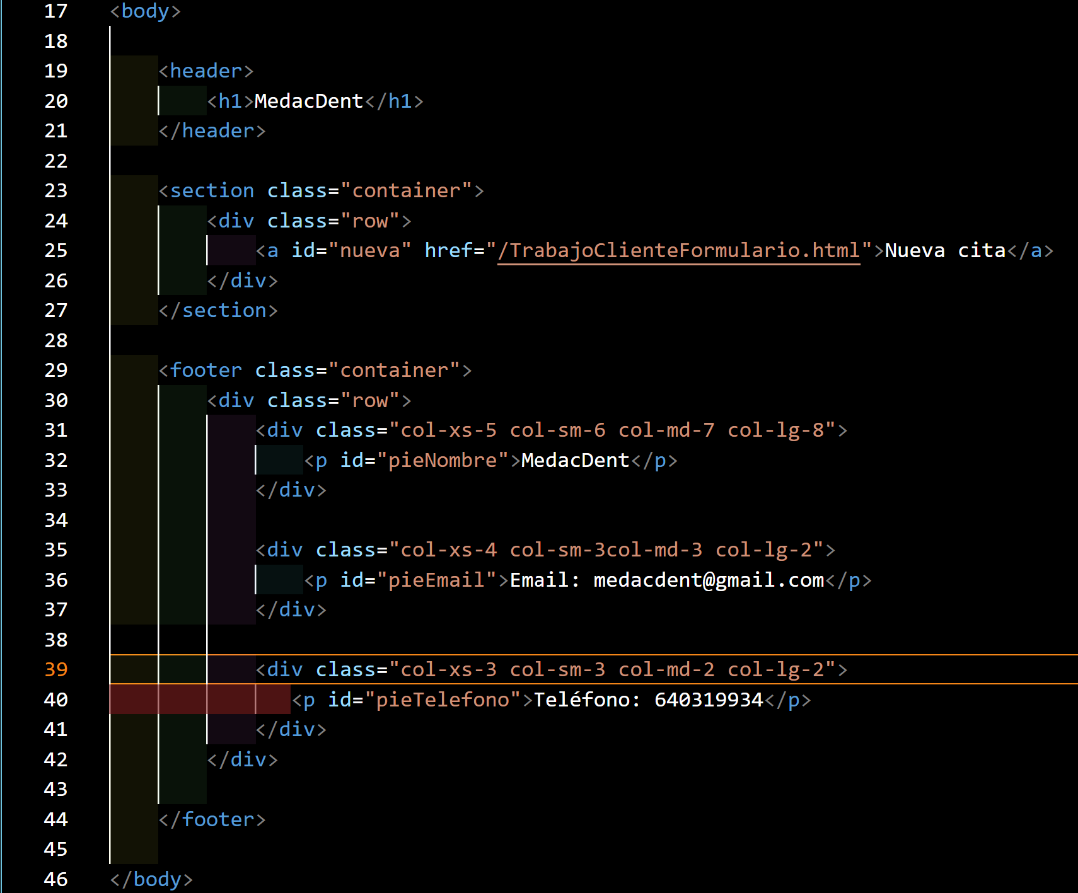
# Contenido ……………………………………….3

# Ejecución ………………………………………15

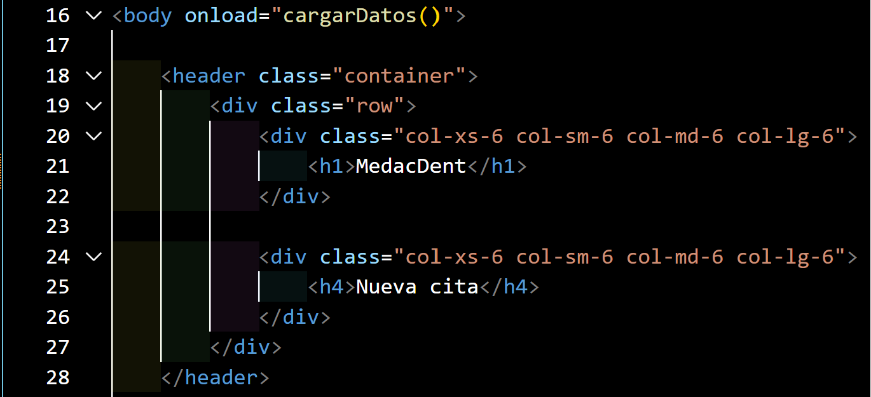
1. **Problemas ……………………………………..18**
2. **Bibliografía ……………………………………19**
3. Contenido

El ejercicio ha sido resuelto en dos documentos HTML. De modo externo, han sido incluidos la funcionalidad (archivo JavaScript) y el estilo (dos archivos CSS, uno para cada HTML). Decidí mostrar una primera página de inicio que enlazara a la página de formulario. En esta última se carga la tabla de las citas.

**El HTML** de la página de inicio es simple. Cuenta con una cabecera con el nombre de la empresa a modo de título, una sección que contiene un enlace a la página de formulario y un pie de página con los datos de contacto.



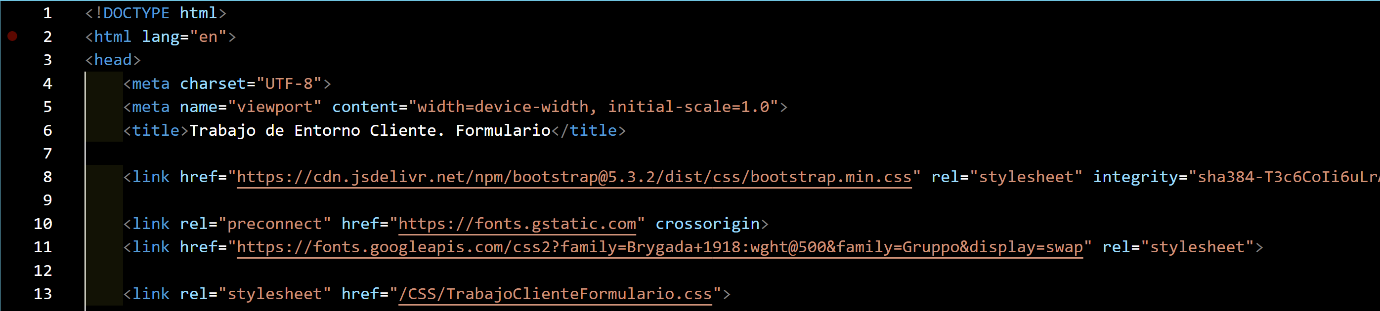
La página del formulario tiene una pequeña cabecera que nos sitúa. La parte principal es una gran sección que embute al formulario. Con dos partes: ‘Día de la cita’ e ‘Información del paciente’. En total, siete campos a rellenar por el usuario. Un pie de página con el contacto termina por definir la estructura.





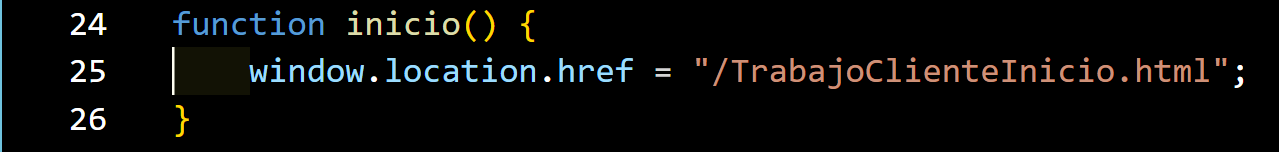


Se puede observar en las imágenes anteriores el uso de las clases **Bootstrap** para estilizar el proyecto y el responsive. He usado este framework en su versión última, la 5.3, junto a **CSS**. Los dos archivos HTML comparten el siguiente ‘head’ con los enlaces al framework, a las fuentes y al fichero de estilo, respectivamente.

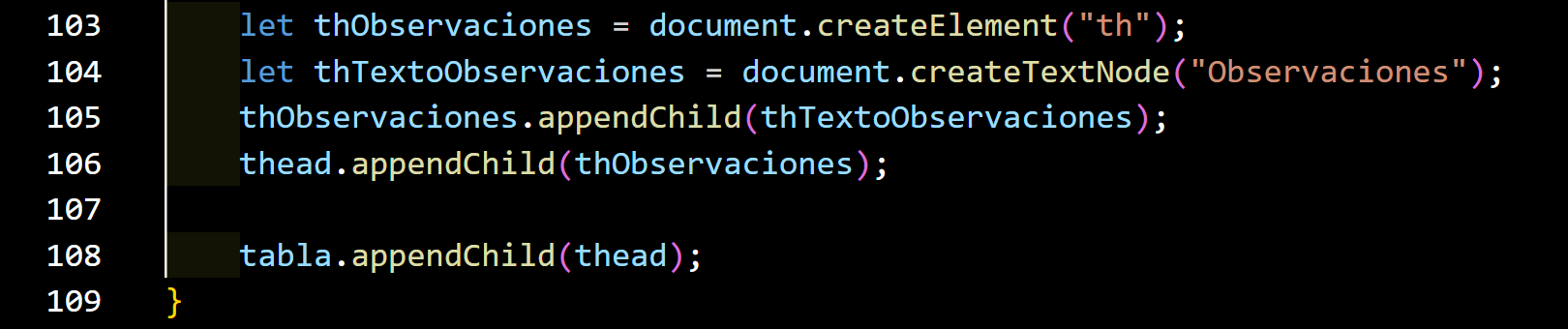
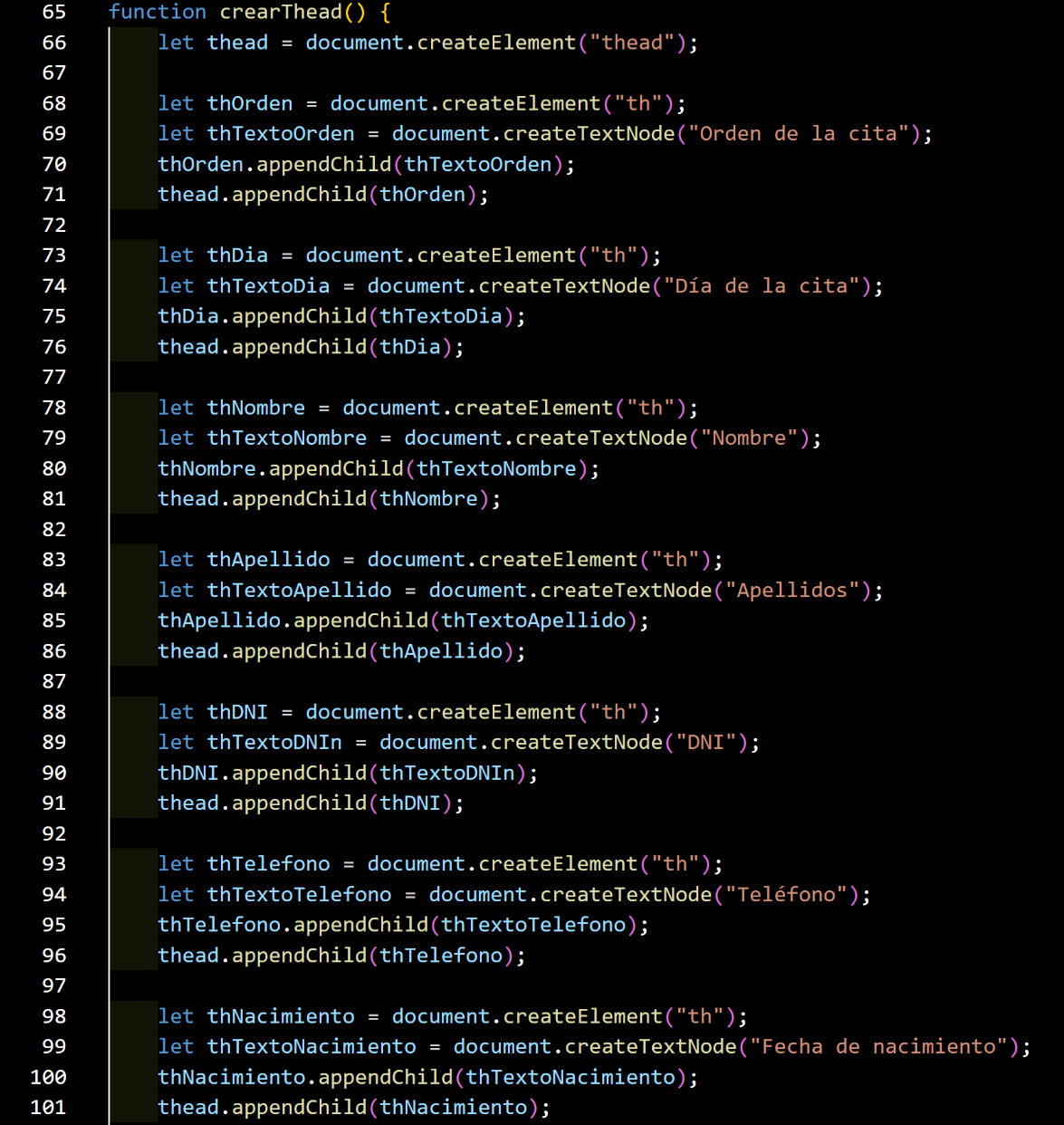
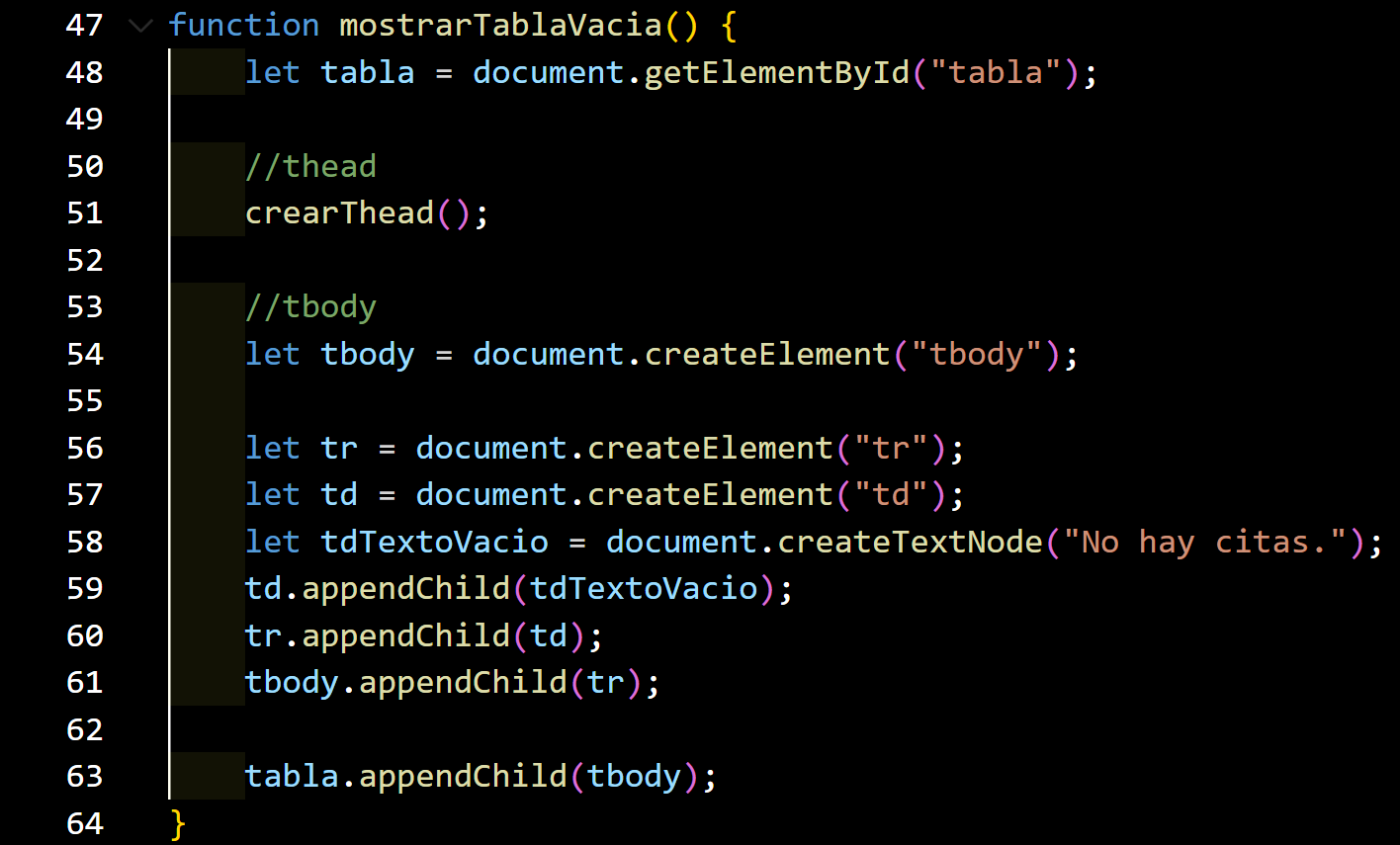
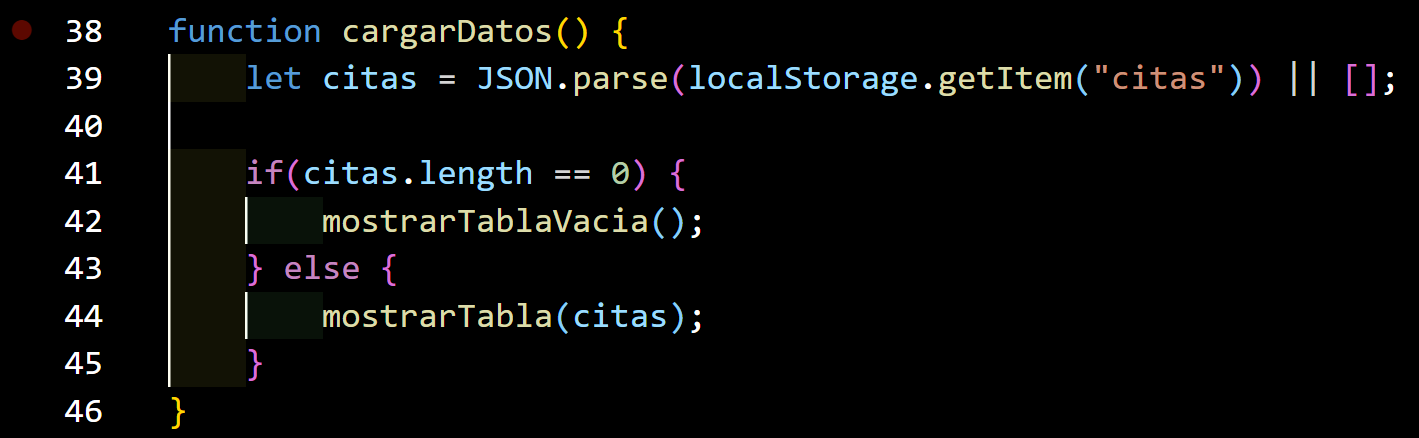
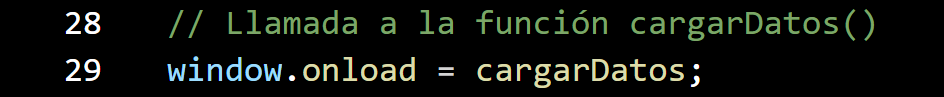


El **JS** es largo. Contabilizo 16 funciones, un número considerable de líneas de código y un espacio que penaliza y excede el recomendado en las instrucciones de entrega.

La función ‘inicio’ nos da acceso desde la página de inicio al formulario.



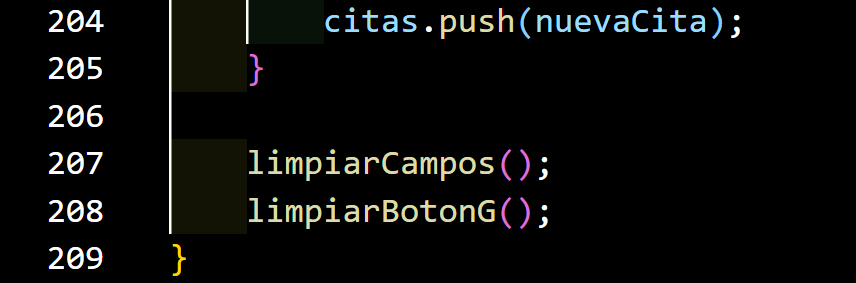
‘CargarDatos’ es llamada desde el ‘body’ de la página formulario. Muestra la tabla con las citas o una tabla vacía, en función de si tenemos o no datos almacenados.



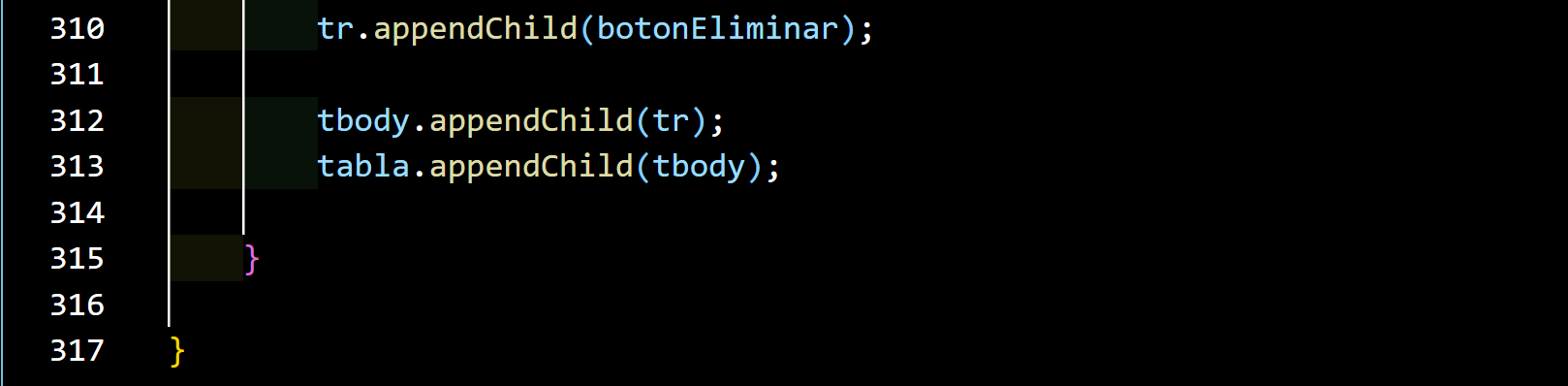
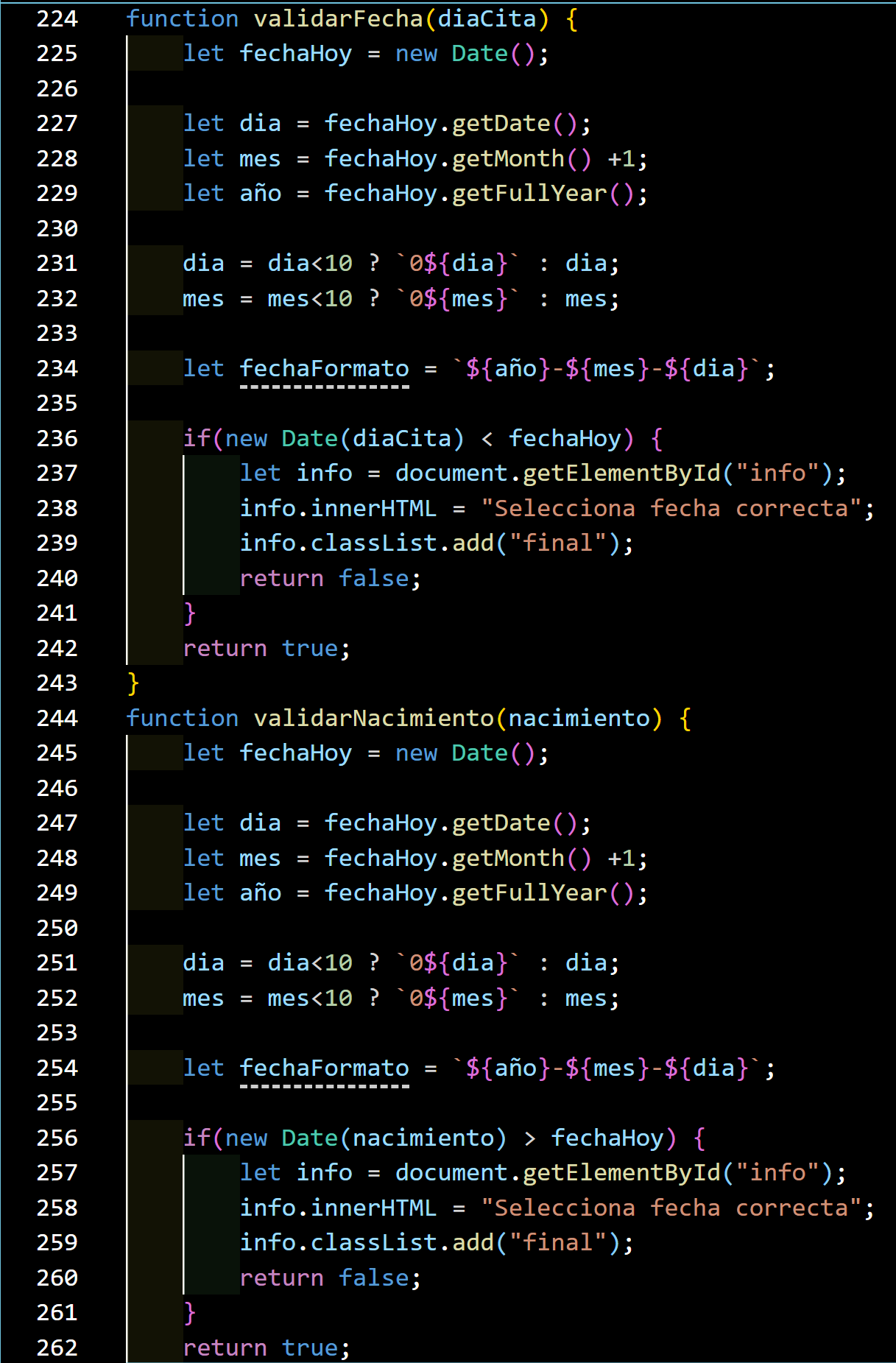
‘CrearTabla’ es una de las funciones principales. Es llamada desde un evento ‘onsubmit’ en el formulario. Crea y renderiza la tabla con las citas almacenadas. Incorpora otras funciones para almacenar la cita (‘pecirCita’[[1]](#footnote-1)) y mostrarlas en una tabla (‘mostrarTabla’).



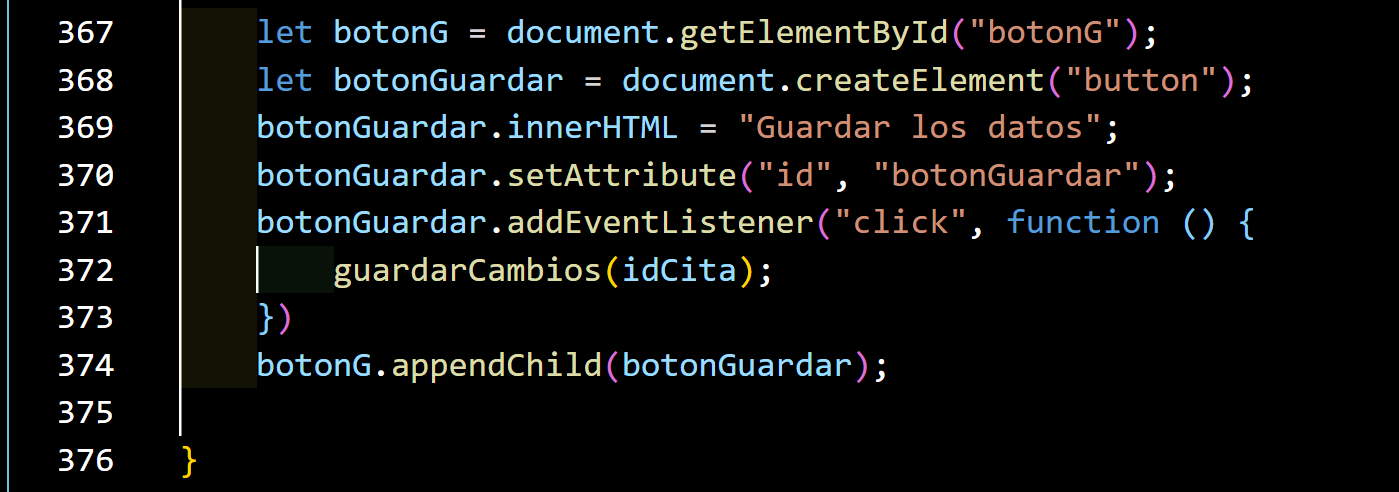
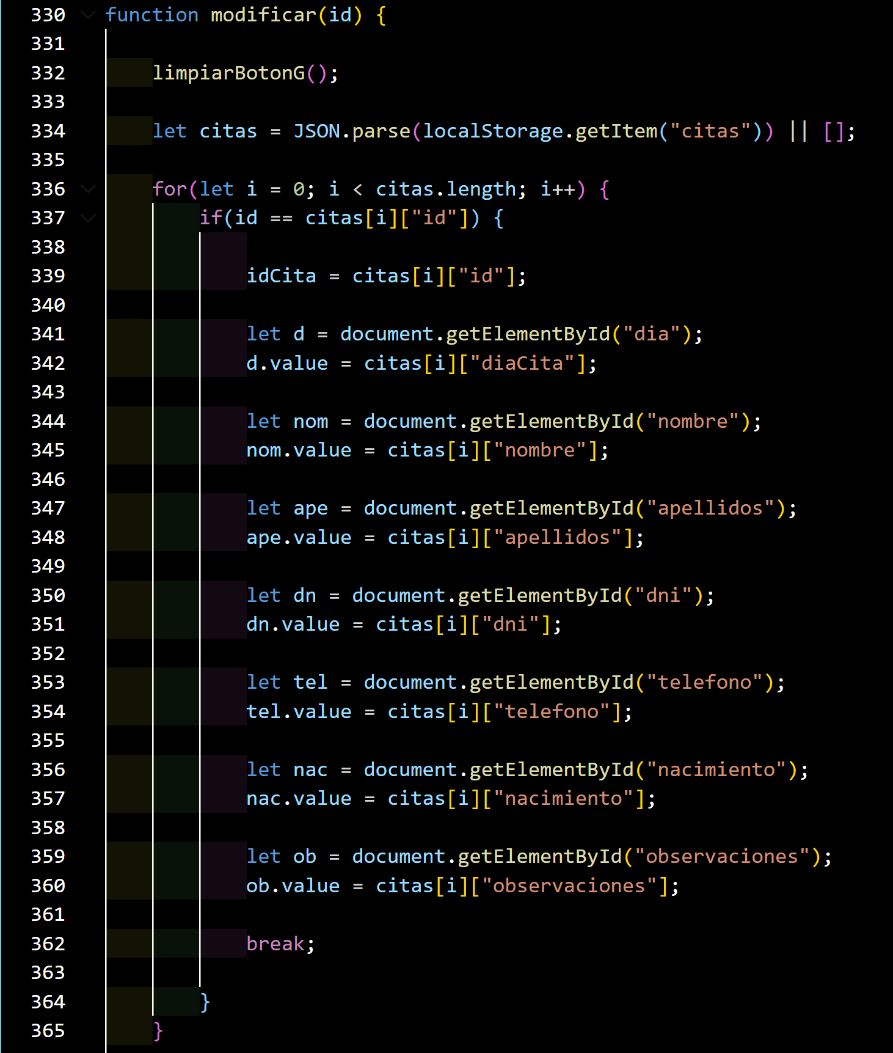




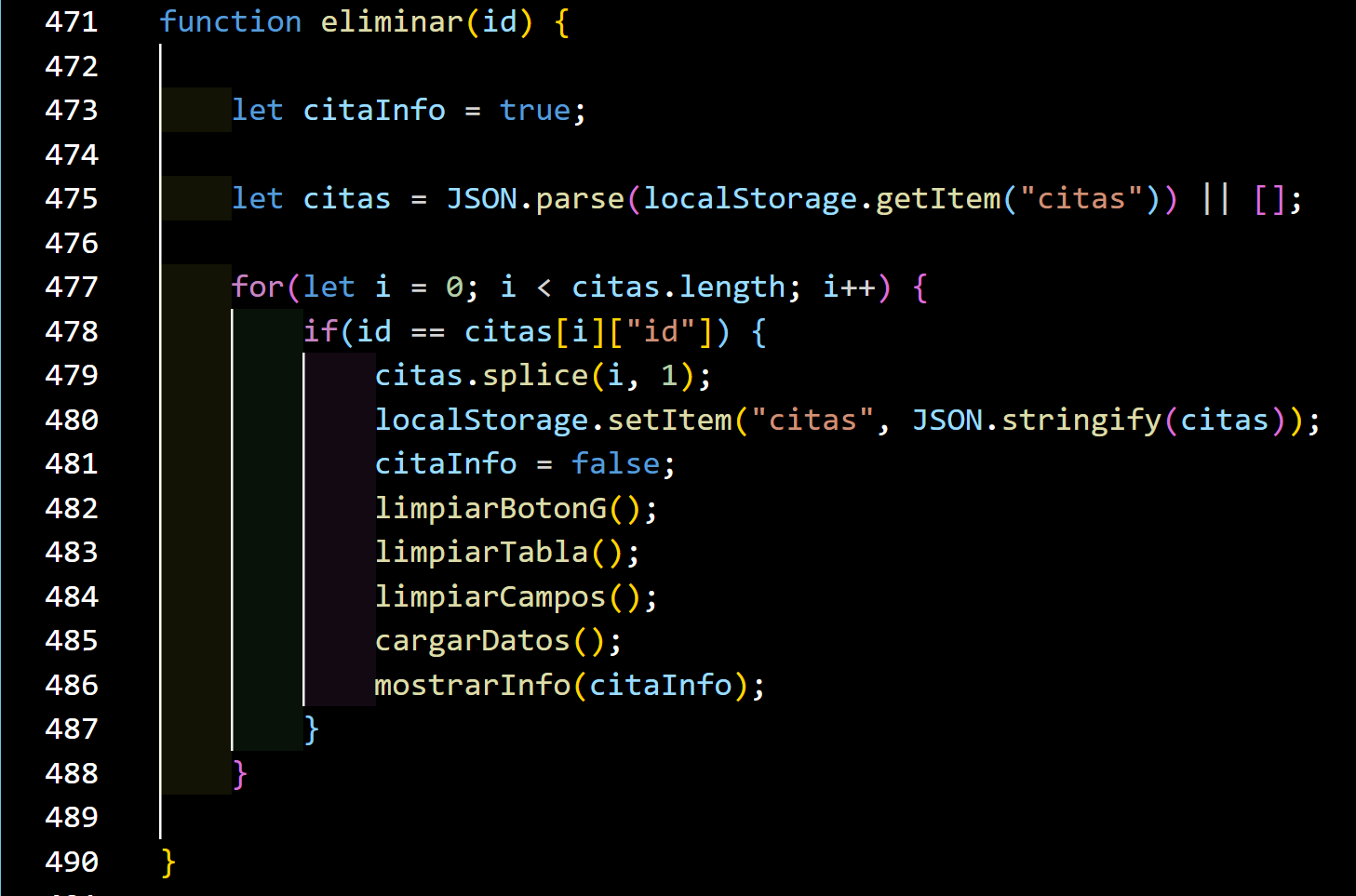
‘LimpiarCampos’ elimina de la pantalla las letras de los campos de formulario. ‘LimpiarBotonG’ hace lo mismo con el botón ‘Guardar los datos’ que aparece el modificar una cita. En las siguientes tomas, muestro las únicas validaciones que hago en JS, la del día de la fecha (‘validarFecha()’) y la del día de nacimiento (‘validarNacimiento()’).



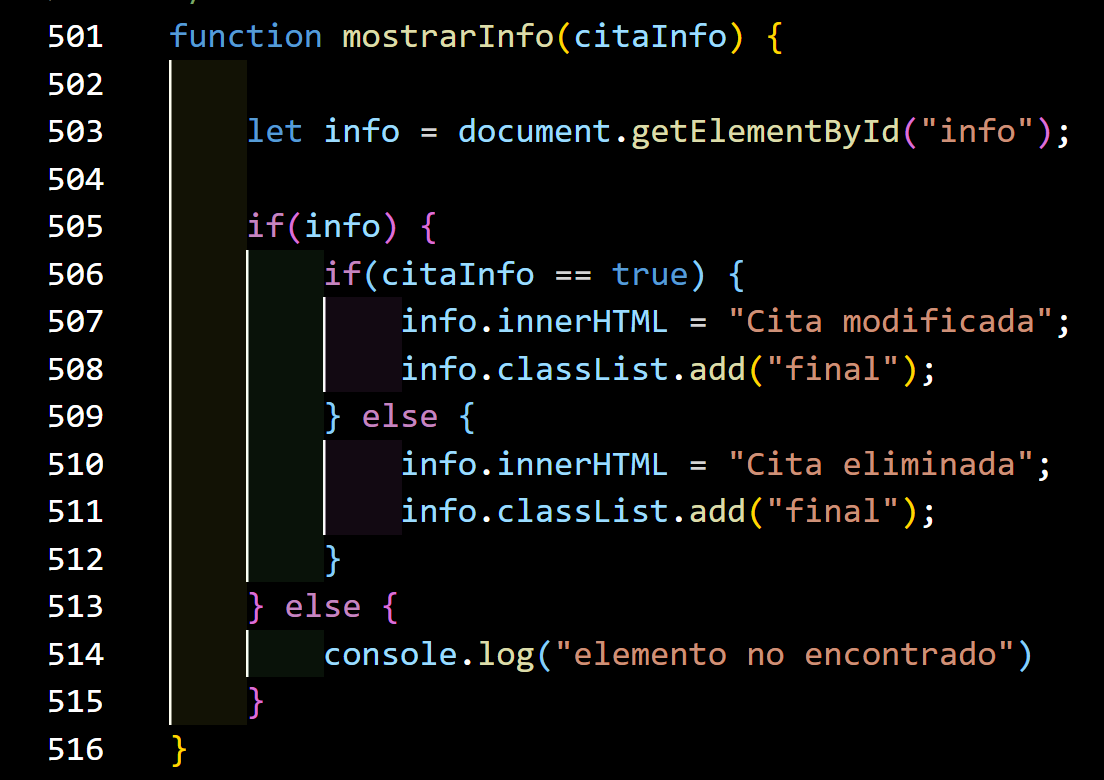
Con ‘modificar’ el usuario puede alterar sus datos almacenados de cita en base a su ‘id’ único. Va unida a ‘guardarCambios’, que es la función que guarda las modificaciones.



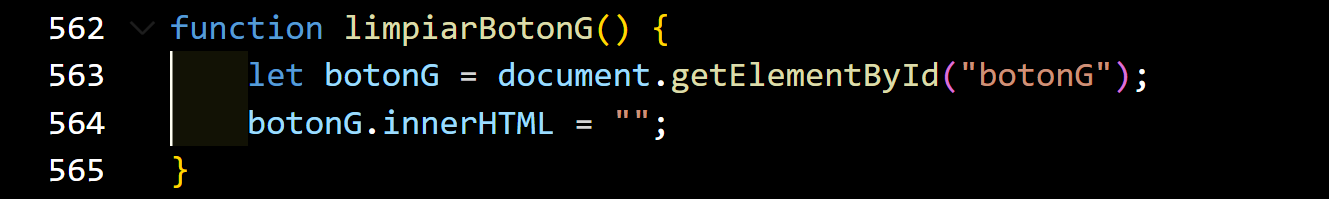
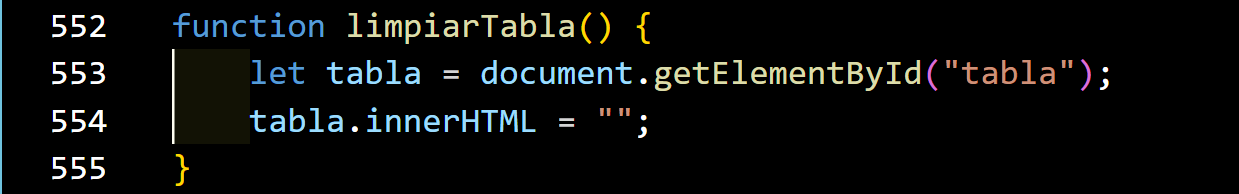
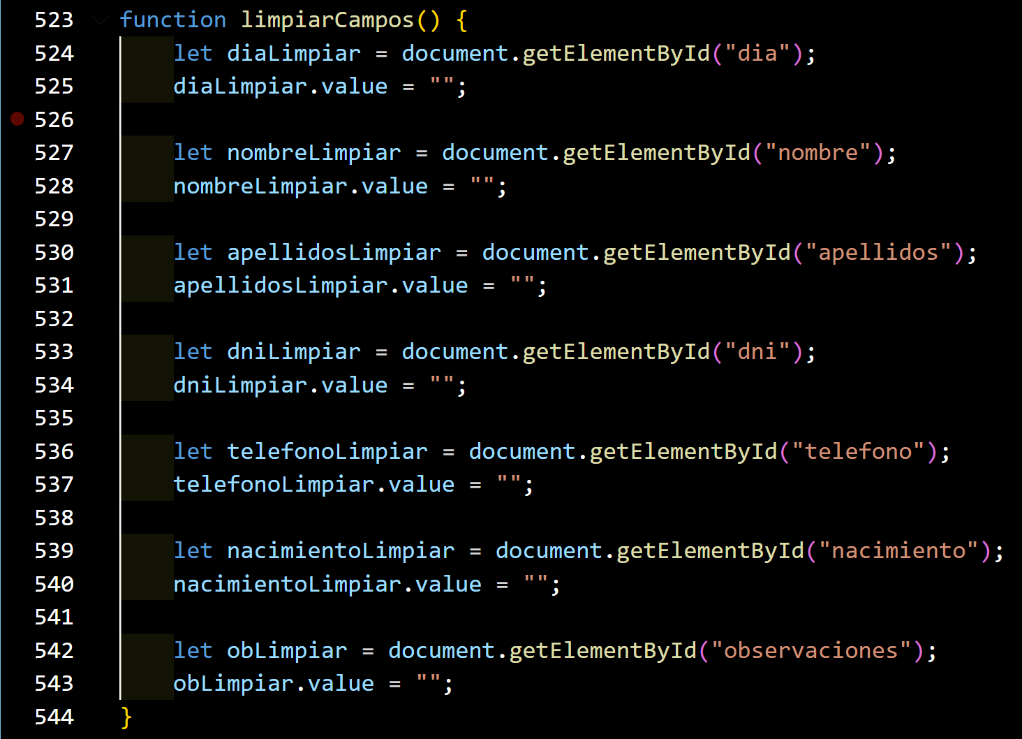
‘Eliminar’ borra la cita almacenada.



‘MostrarInfo()’[[2]](#footnote-2) añade una información en pantalla para alertar al usuario que sus datos de cita han sido modificados o eliminados.



Por último, las tres funciones de limpieza que eliminan de la pantalla algunos elementos.

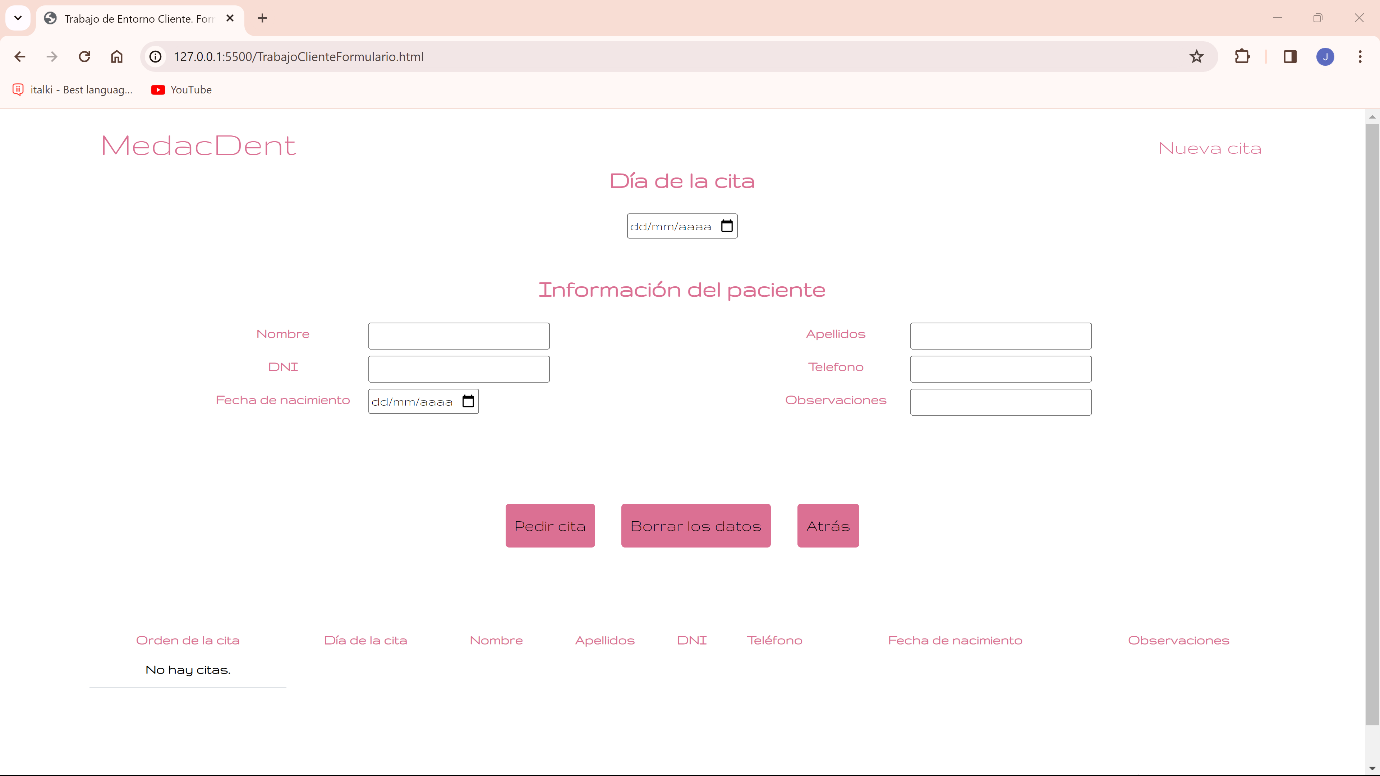


1. **Ejecución**

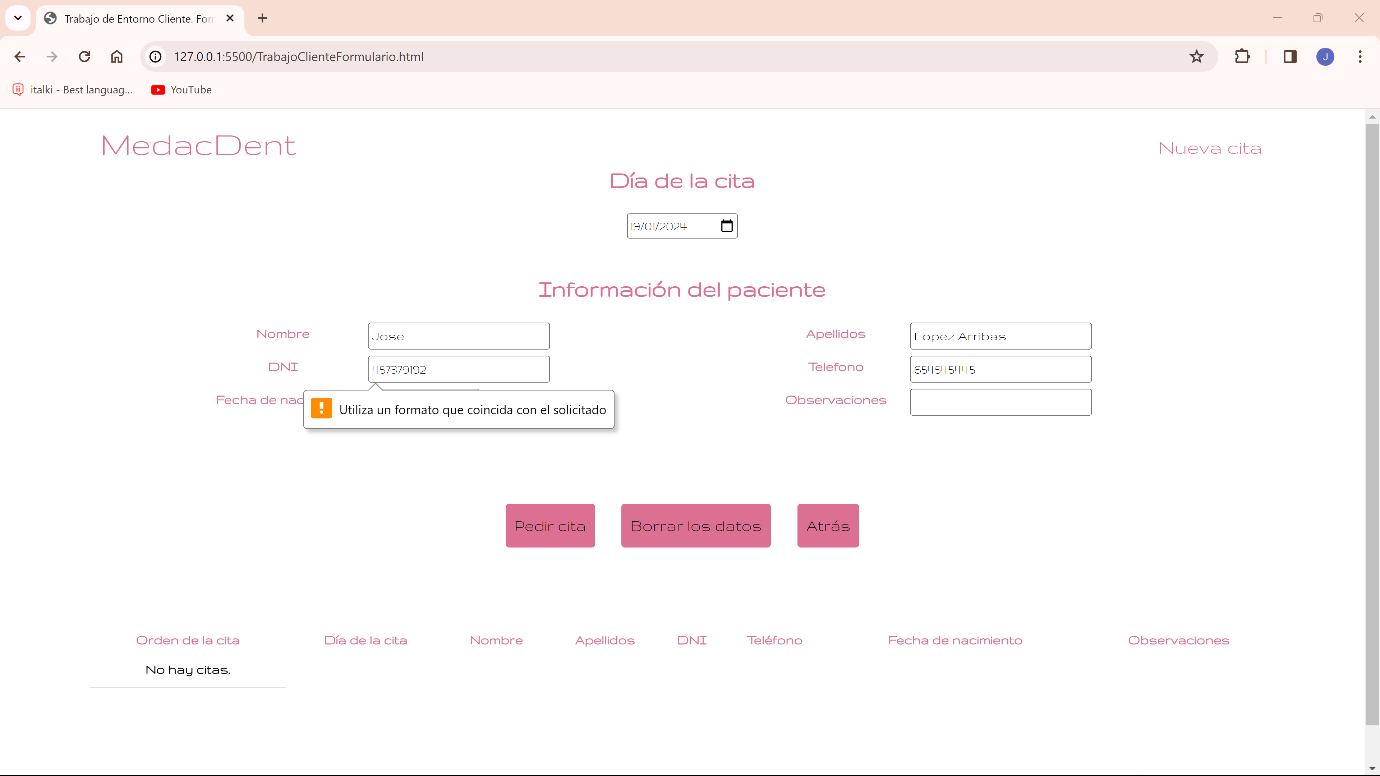
El usuario entra en la web y ve la siguiente pantalla:



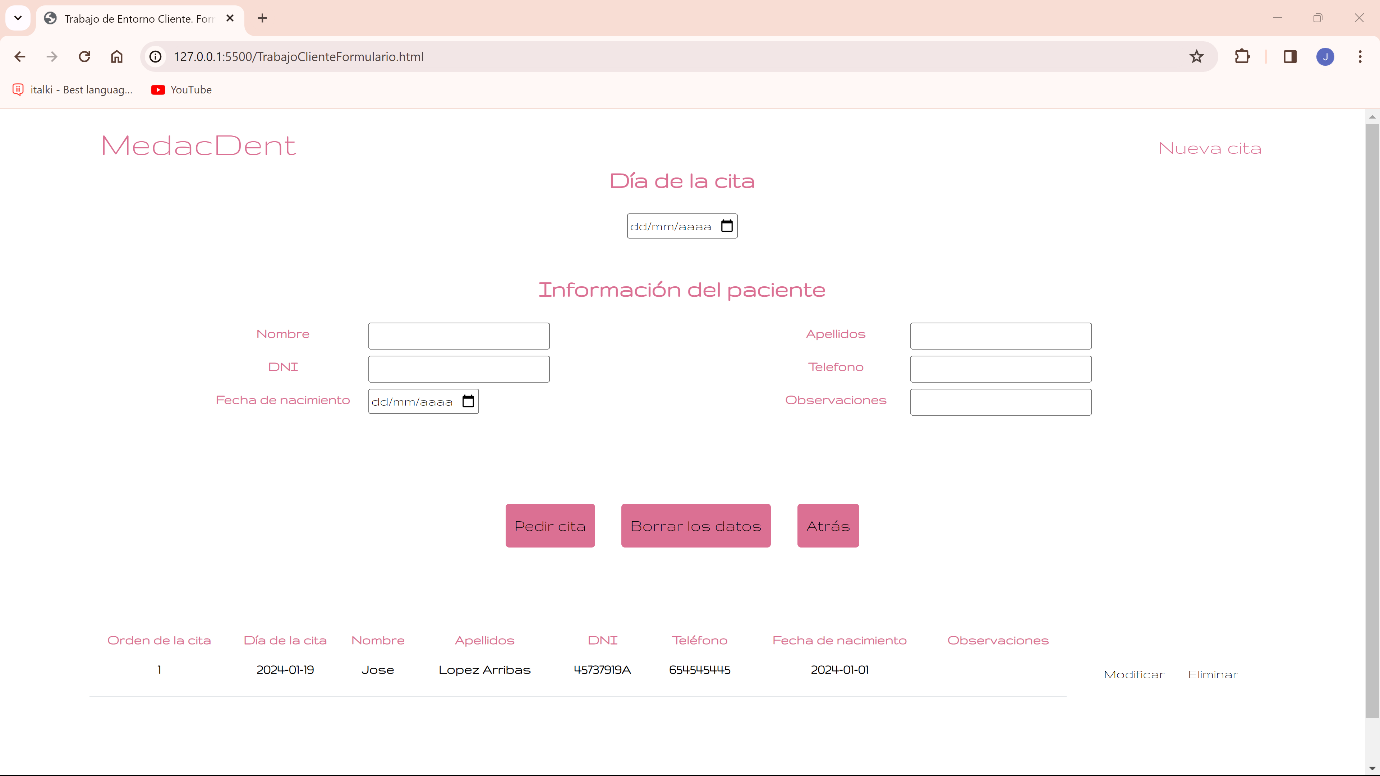
Entra en ‘**Nueva cita’**.



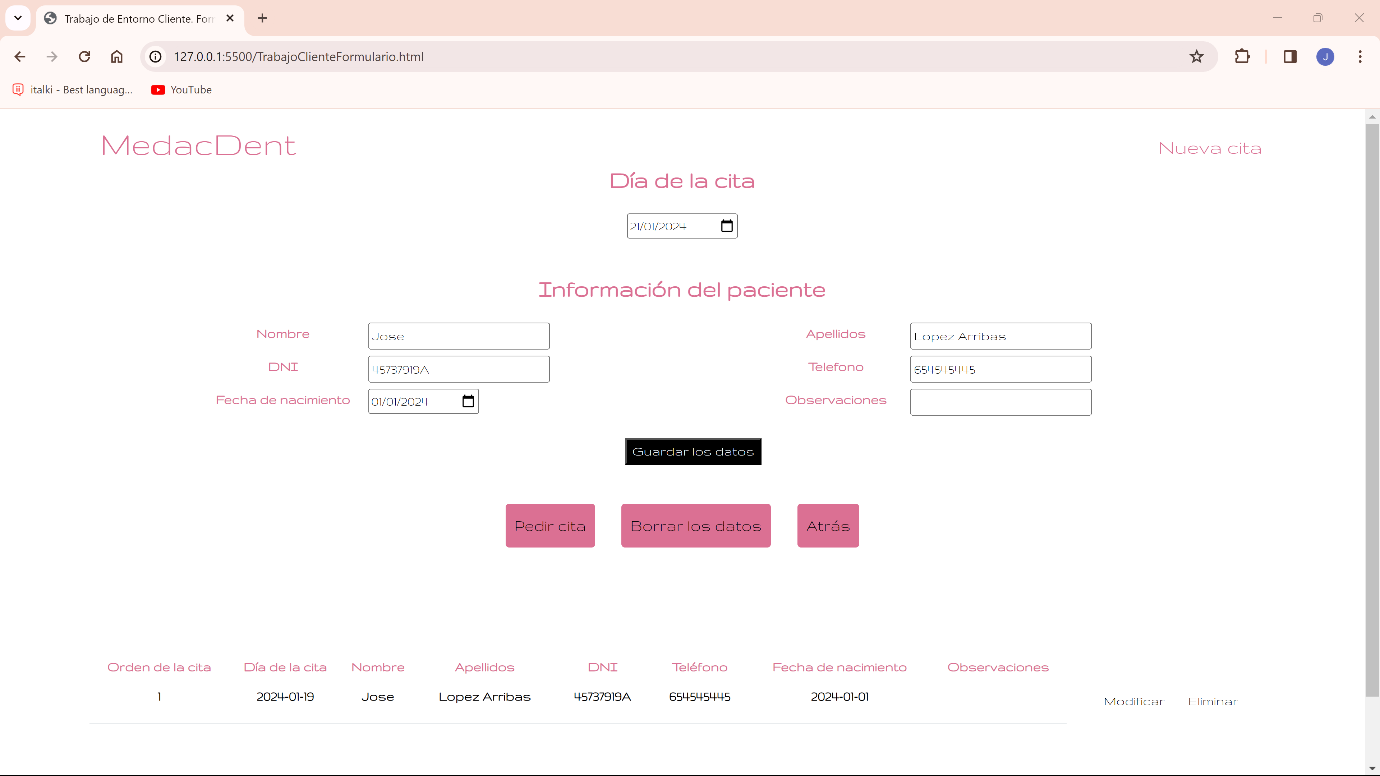
Observa el **formulario** con los datos para pedir la cita, **tres botones** (uno para confirmar, otro para borrar los datos y otro para volver a inicio), y la **tabla** de citas, donde aún no hay citas almacenadas. El usuario introduce sus datos.



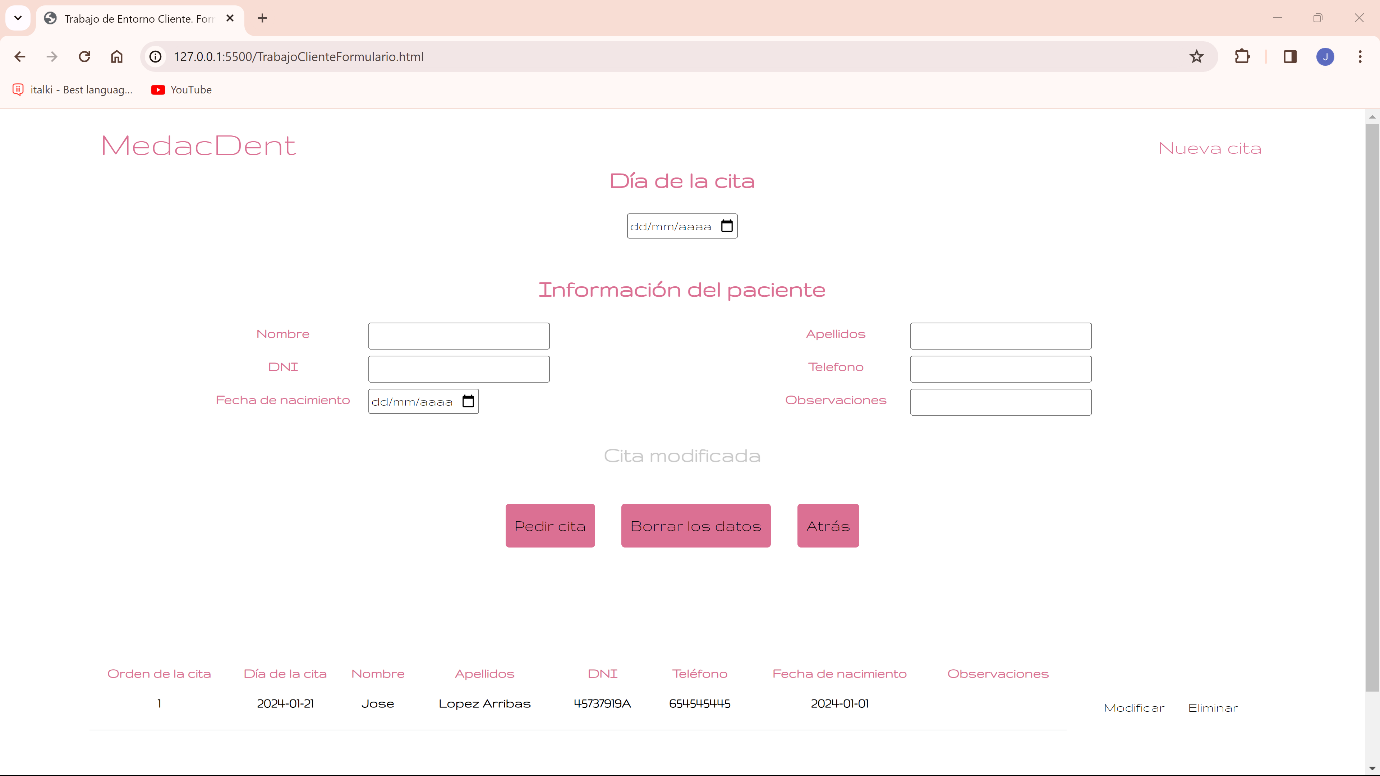
Parece que está teniendo problemas. Debe introducir un DNI con el formato correcto.



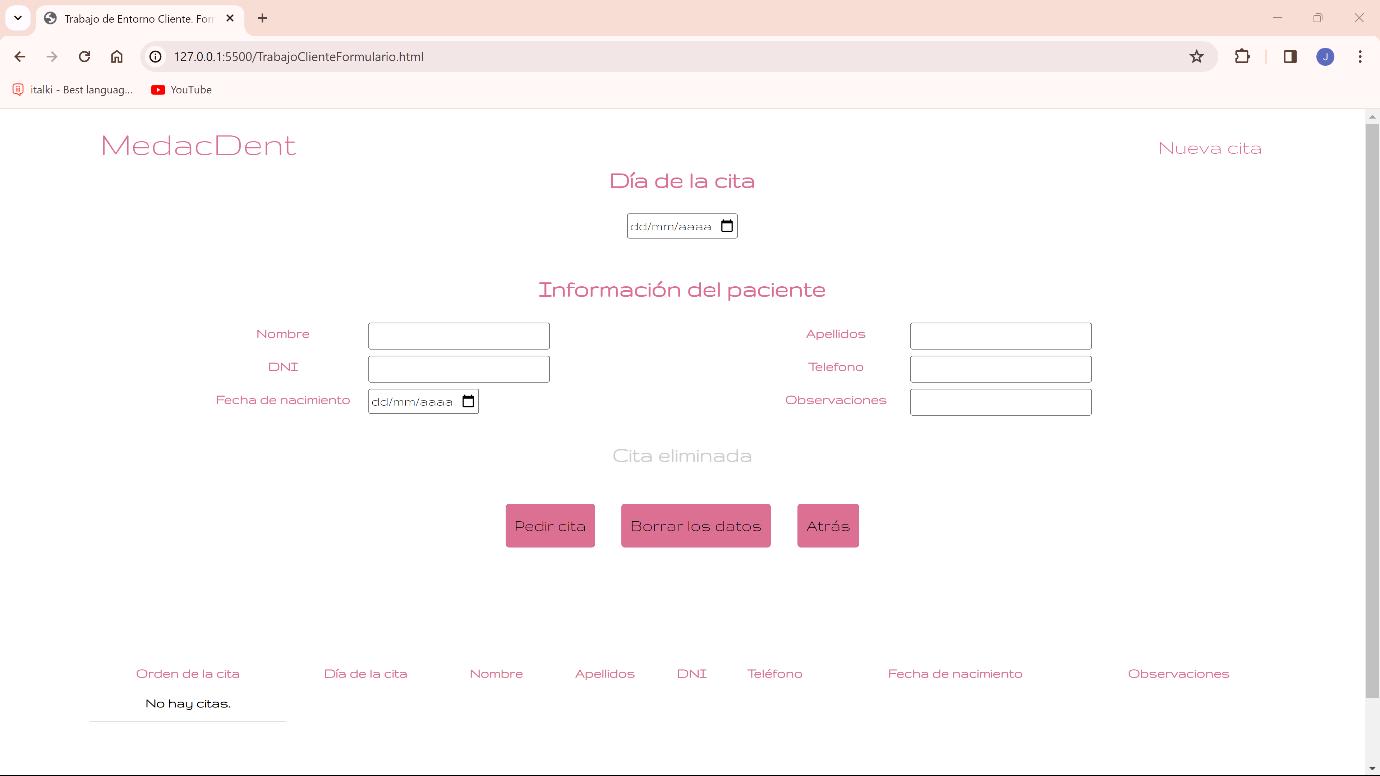
Lo hace y, automáticamente, su cita aparece en la tabla como almacenada. Resulta que quiere modificar el día de la cita. Para ello clica en ‘**Modificar’**. Los datos de la cita vuelven a los campos del formulario listos para ser cambiados. Tan solo debe seleccionar el día y **guardar los datos** en el nuevo botón que aparece.



La cita del paciente ha sido modificada.



¿Y si se arrepiente y la quiere **eliminar**?



Cita eliminada. No hay citas.

1. **Problemas**

A pesar de que el programa funciona: almacena, valida, modifica y elimina; hay un tipo de interferencia en la función ‘pedirCita’ a la hora de modificar los datos. Creo que no debiera porqué haberlo. No obstante, no he acertado a resolver. Mi solución ha sido de pega, pues el problema sigue estando; solo eliminé un ‘alert’ que informaba de que un mismo id no podía pedir más de una cita para el mismo día. Hasta esa línea profunda de función distinta penetra la aplicación a la hora de modificar.

¿Por qué la información que quiero mostrar al modificar y al eliminar una cita solo aparece la primera vez que se hacen tales acciones? Tampoco lo he podido detectar.

Ambos defectos permanecen.

1. **Bibliografía**

* Temario asignatura ‘Desarrollo Web en Entorno Cliente’, temas 1-9.
* W3School. <https://www.w3schools.com/>
* MDN web docs. https://developer.mozilla.org/es/
* OpenAI. (2023). ChatGPT.

1. Ver apartado **Problemas**, página . La función pedirCita interfiere al modificar una cita. [↑](#footnote-ref-1)
2. Ver apartado **Problemas**, página .. . La función ‘mostrarInfo’ no actúa todas las veces que es llamada. [↑](#footnote-ref-2)