28-2-2021

Víctor Cobo y David Ramos

Nicolau Copernic

UF3 Esdeveniments. Manegament de formularis. DOM

Proyecto UF3 20/21

Índice

[Requisitos obligatorios para optar a un 5 2](#_Toc65446415)

[Archivo index.html 3](#_Toc65446416)

[Archivo main.js 3](#_Toc65446417)

[Conclusiones 12](#_Toc65446418)

[ Víctor Cobo 12](#_Toc65446419)

[ David Ramos 12](#_Toc65446420)

[Problemas y partes complicada 12](#_Toc65446421)

# Requisitos obligatorios para optar a un 5

Al abrir la aplicación web, sólo será visible el contenido del primer DIV que contendrá una tabla con los registros guardados, si no hay elementos para mostrar se mostrará un texto indicándolo. También habrá un botón que eliminará (NO ocultar) todo el contenido del primer DIV y añade el segundo DIV un formulario con el que debe de constar como mínimo los siguientes elementos HTML;

* 2 inputs de tipo Texto
* 1 input de tipo numérico
* 1 input de selección múltiple
* 1 input que nos permita seleccionar una imagen
* 1 input de tipo menú desplegable con 4 opciones

Se valorará positivamente añadir más elementos HTML en el formulario de otros tipos o funcionamientos. Toda la información generada por el formulario será guardada en un array donde cada posición será un objeto JSON con los datos rellenados. Uno de los campos del objeto JSON debe tener un valor único autoincrementar gestionado por la aplicación.

Algunos campos del formulario serán obligatorios (3 como mínimo) y otros serán opcionales. Se validará desde código JavaScript que estén todos los campos obligatorios correctamente rellenado y, en caso contrario, no se permitirá el envío del formulario y los elementos HTML que no están correctamente cumplimentados deben resaltar modificando sus propiedades.

Al menos 2 elementos HTML del formulario deben tener una validación por expresión regular hecha desde código JavaScript.

Se deben utilizar al menos 3 tipos diferentes de eventos (dos de ratón y un teclado). Una vez cumplimentado correctamente el formulario se guardará la información en un objeto JSON al array, eliminará todo el contenido del DIV donde está el formulario y al primer DIV se volverá a crear una tabla con los registros guardados.

En la tabla se debe mostrar la información de todos los registros, incluida la imagen y enlaces para poder modificar y eliminar cada uno de los registros. Además, al hacer doble clic sobre la imagen ésta debe ser eliminada de la mesa y del registro (JSON) correspondiente del array.

El enlace de modificar no debe hacer nada por ahora, pero el enlace de eliminar si debe eliminar el registro de la tabla y del array. Al eliminar la imagen de un registro o el registro completo de la mesa, NO se puede eliminar la tabla y volver a crearla, se afectará sólo a los elementos implicados únicamente.

NO se puede hacer ningún botón o enlace para cerrar y volver a abrir la aplicación o para recargarla.

Se debe realizar una memoria obligatoria explicando las partes principales de la aplicación, las dificultades encuentros y una conclusión individual.

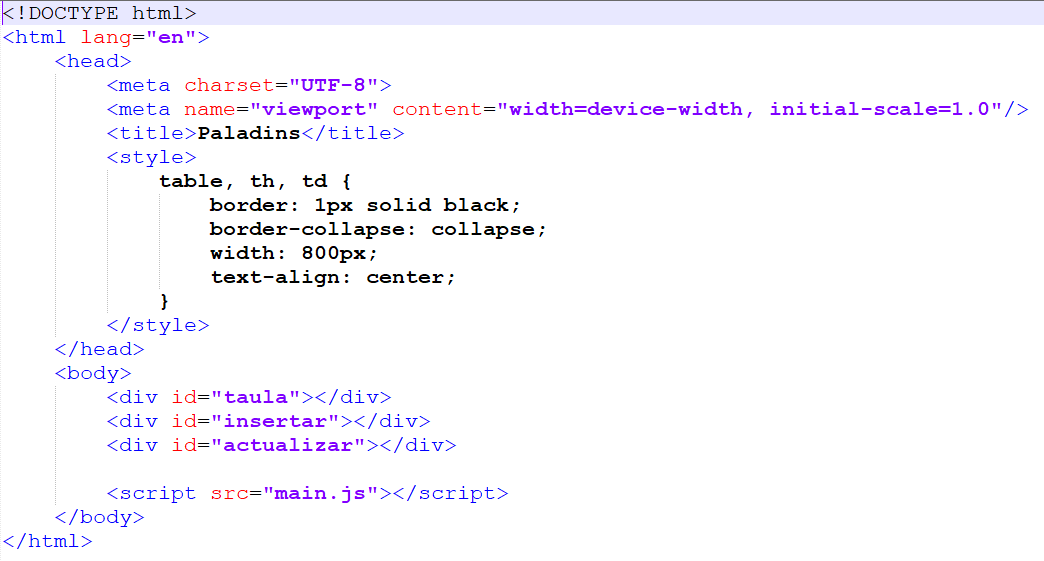
● Mostrar la mesa con toda la funcionalidad: 1,5 puntos.

● Formulario para insertar con toda la funcionalidad: 2,5 puntos.

● Memoria: 1 punto.

# Archivo index.html

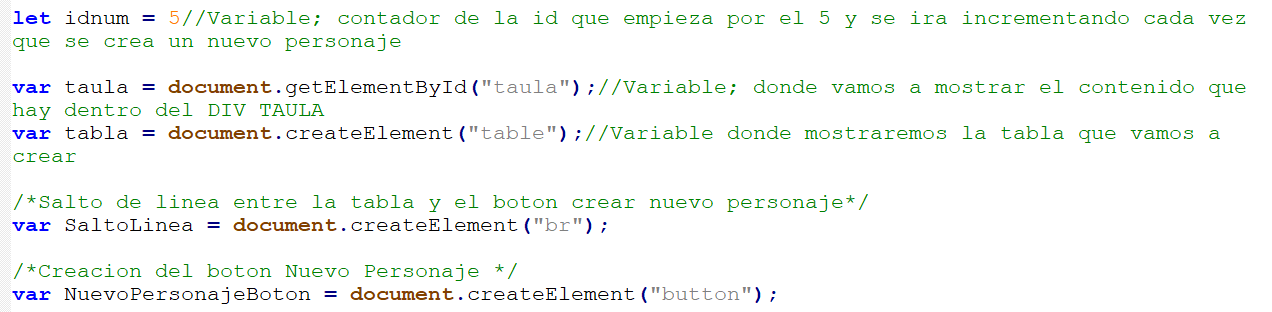
En nuestro archivo HTML encontraremos una estructura básica donde tenemos tres DIV’S que no tienen ningún tipo de configuración asignada dentro de los DIVS porque toda esa configuración se la hemos dado al archivo js que explicaremos a continuación.

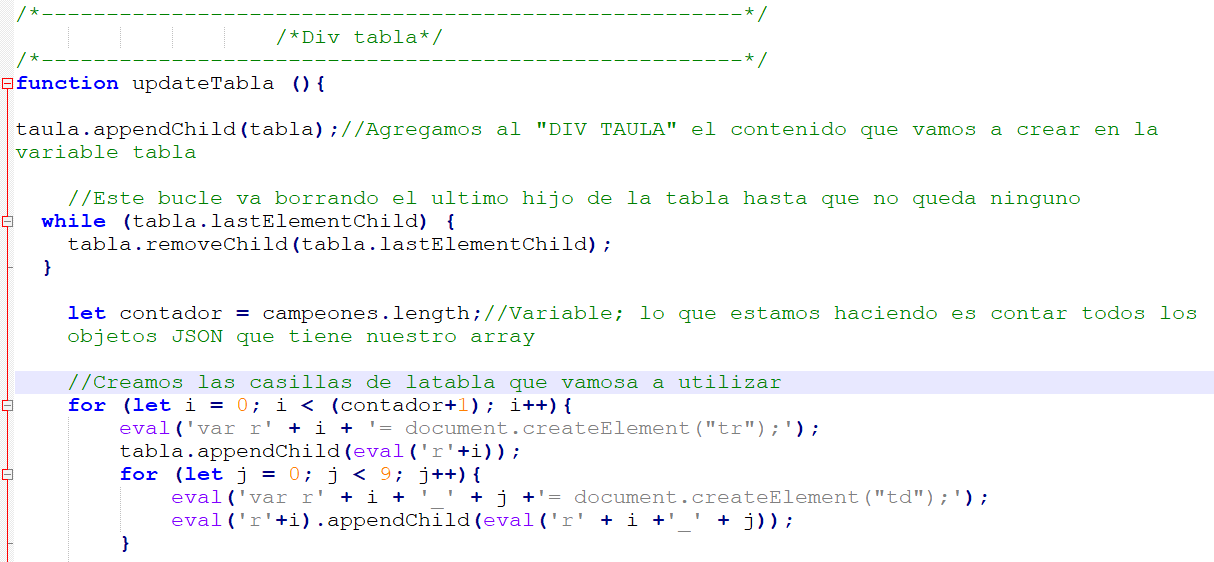


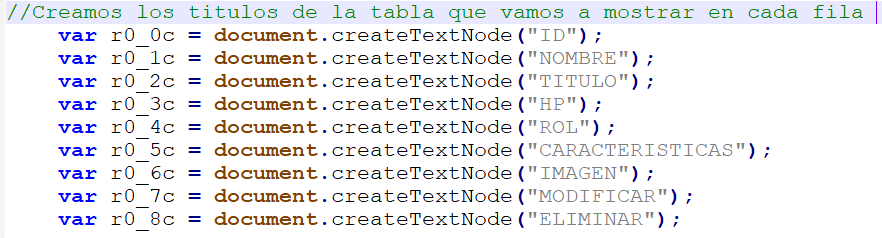
# Archivo main.js

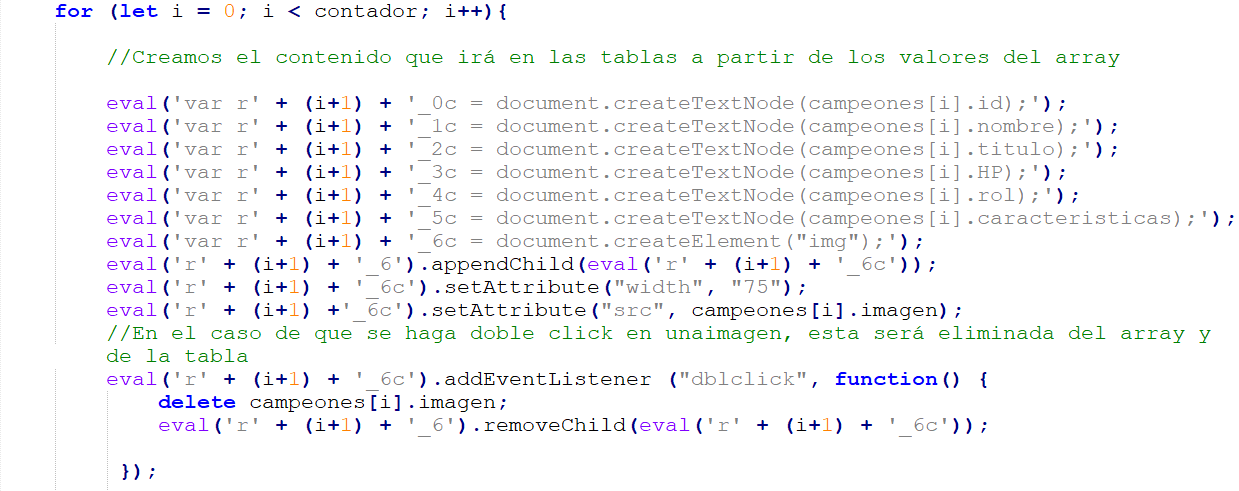
Estamos declarando nuestro array llamada “campeones “, dentro de este array vamos a alojar los objetos JSON, que nosotros vamos a definir para que se muestren en el “DIV tabla” que hay en nuestro archivo html.

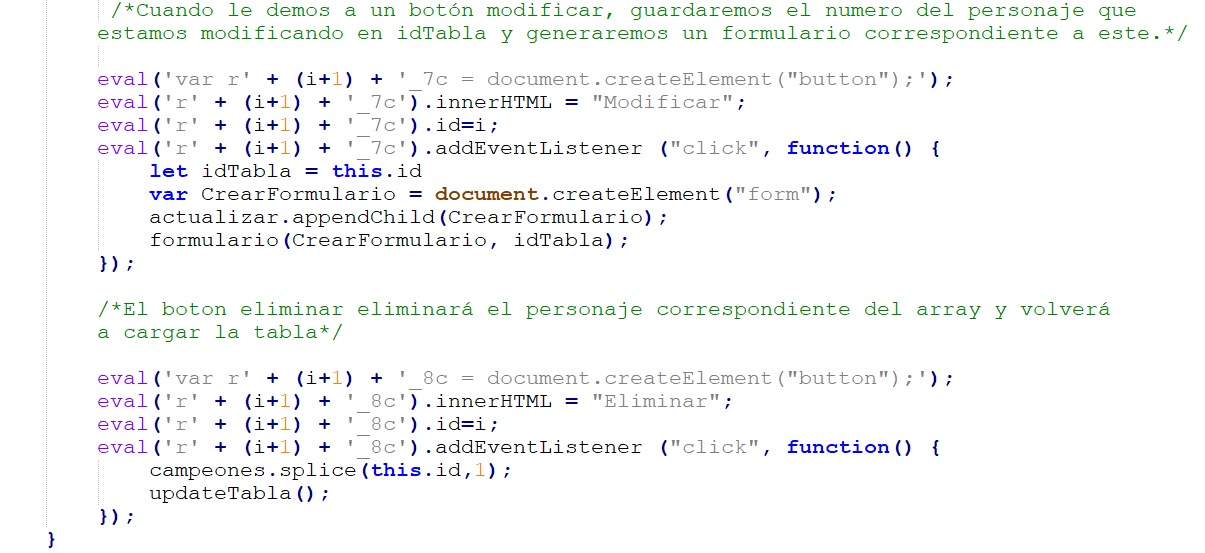


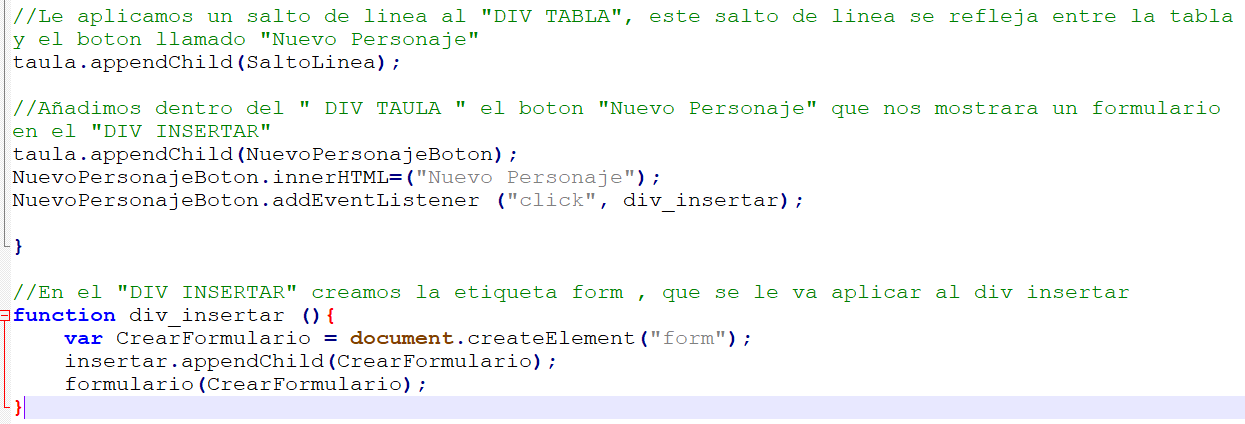


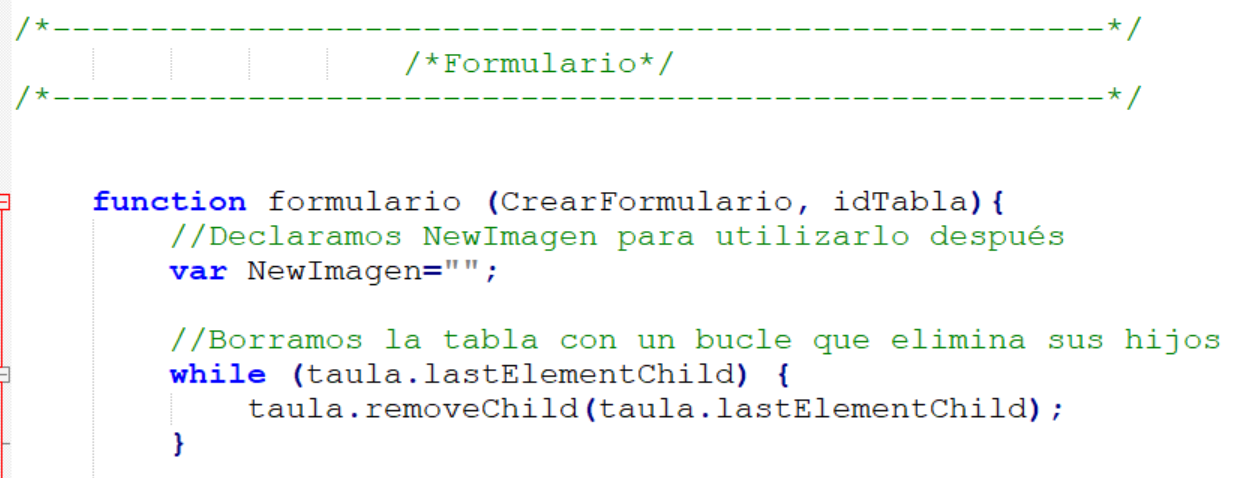


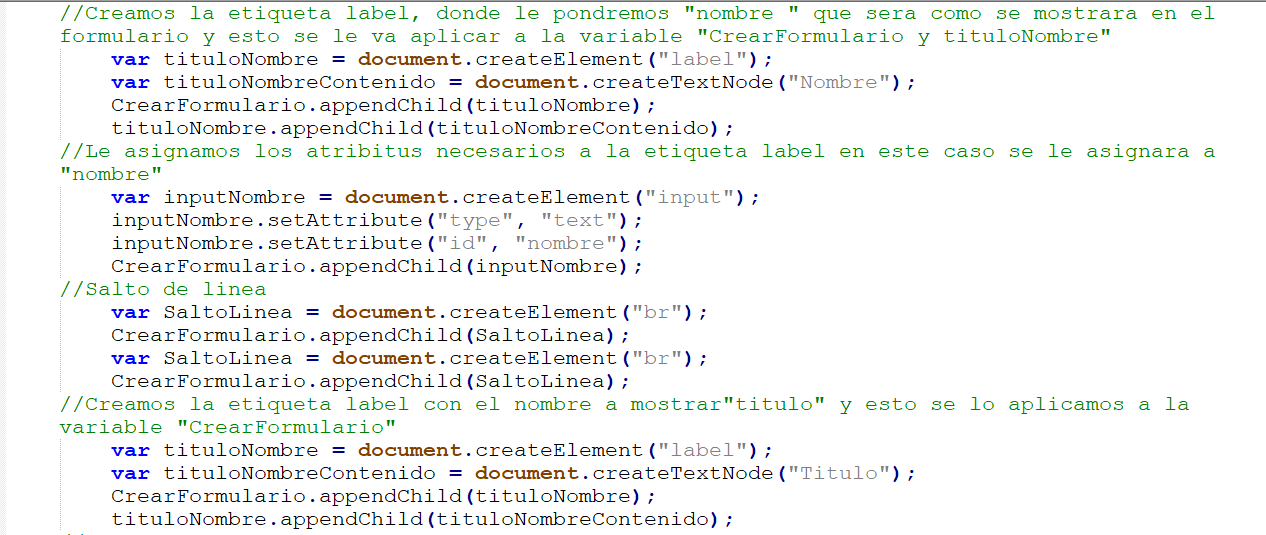


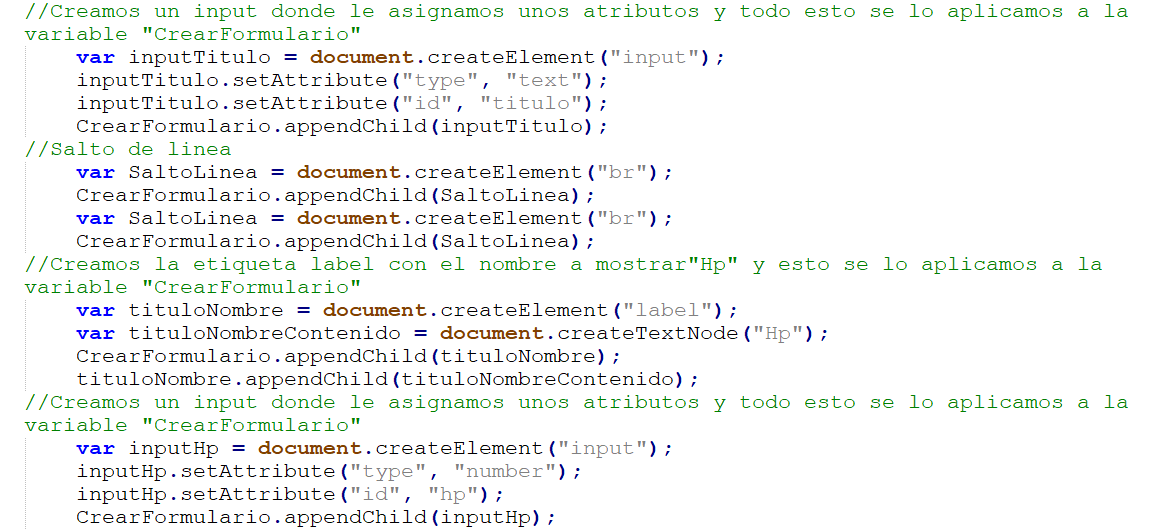


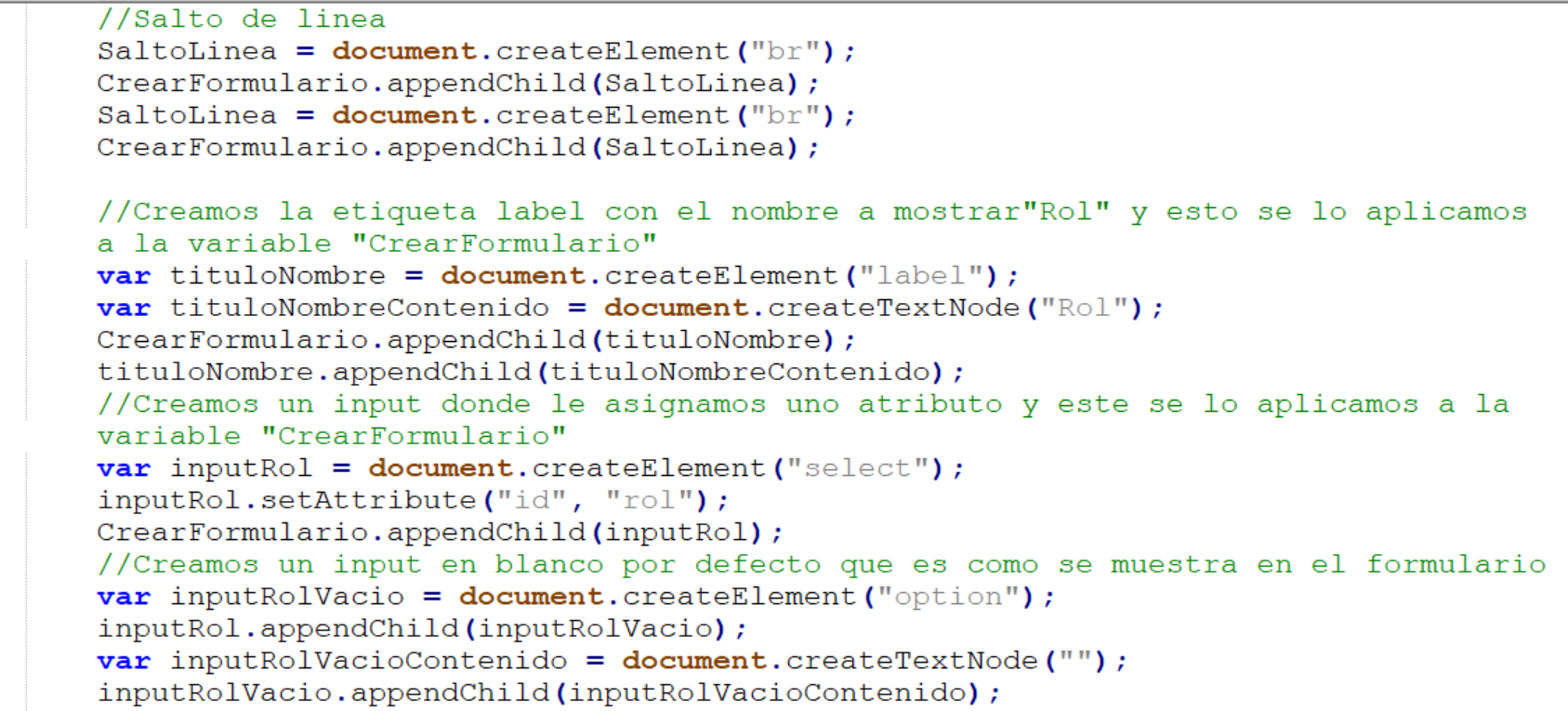


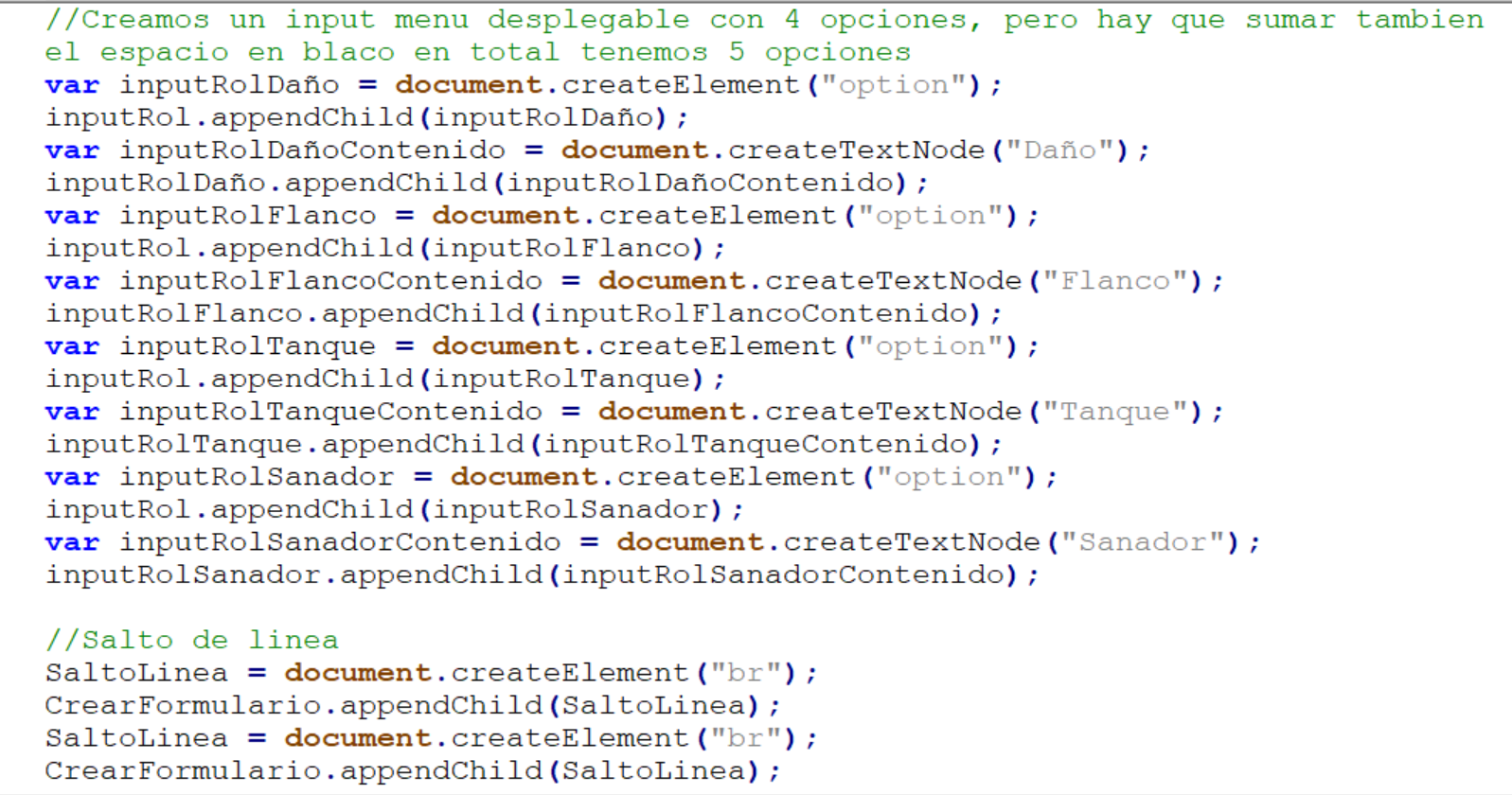


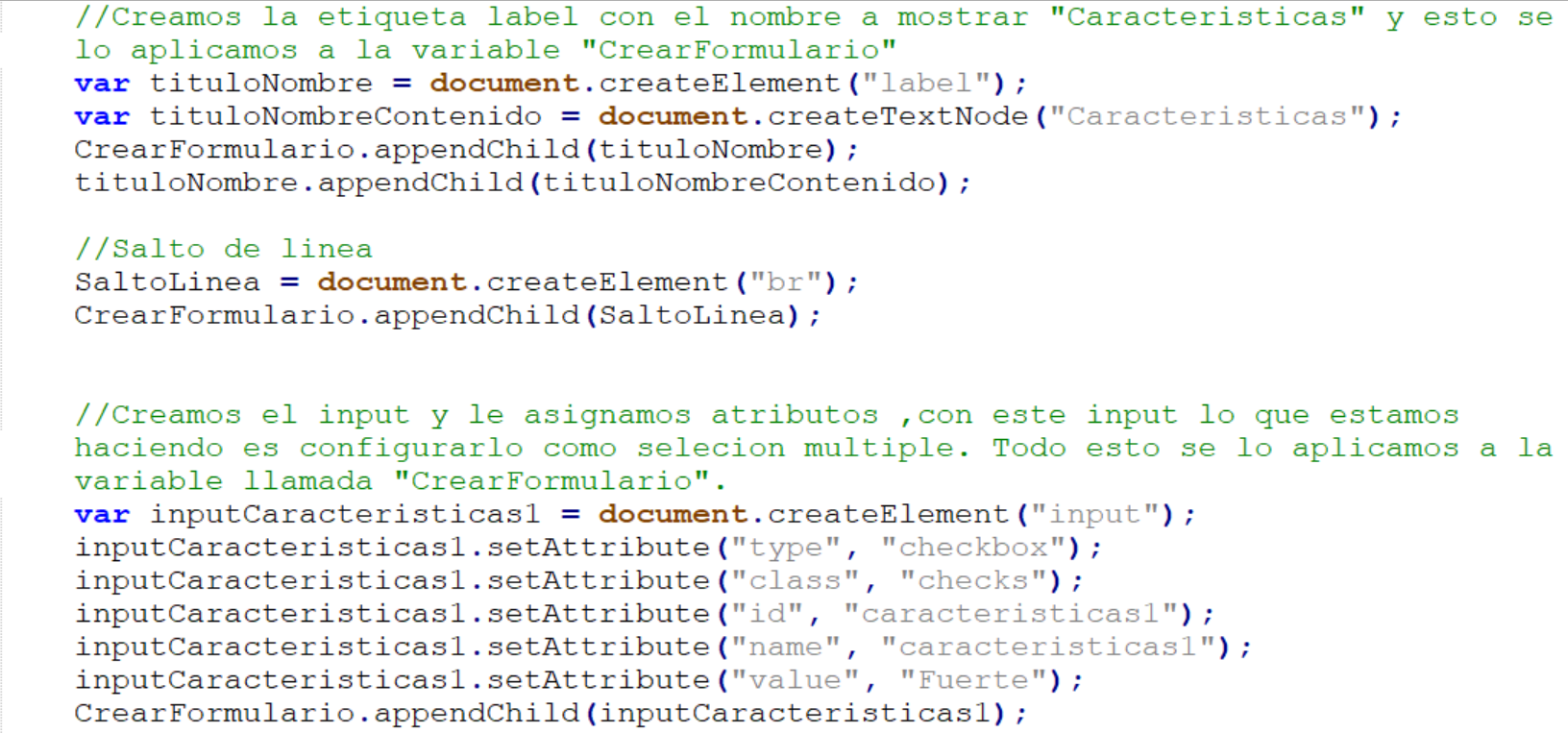


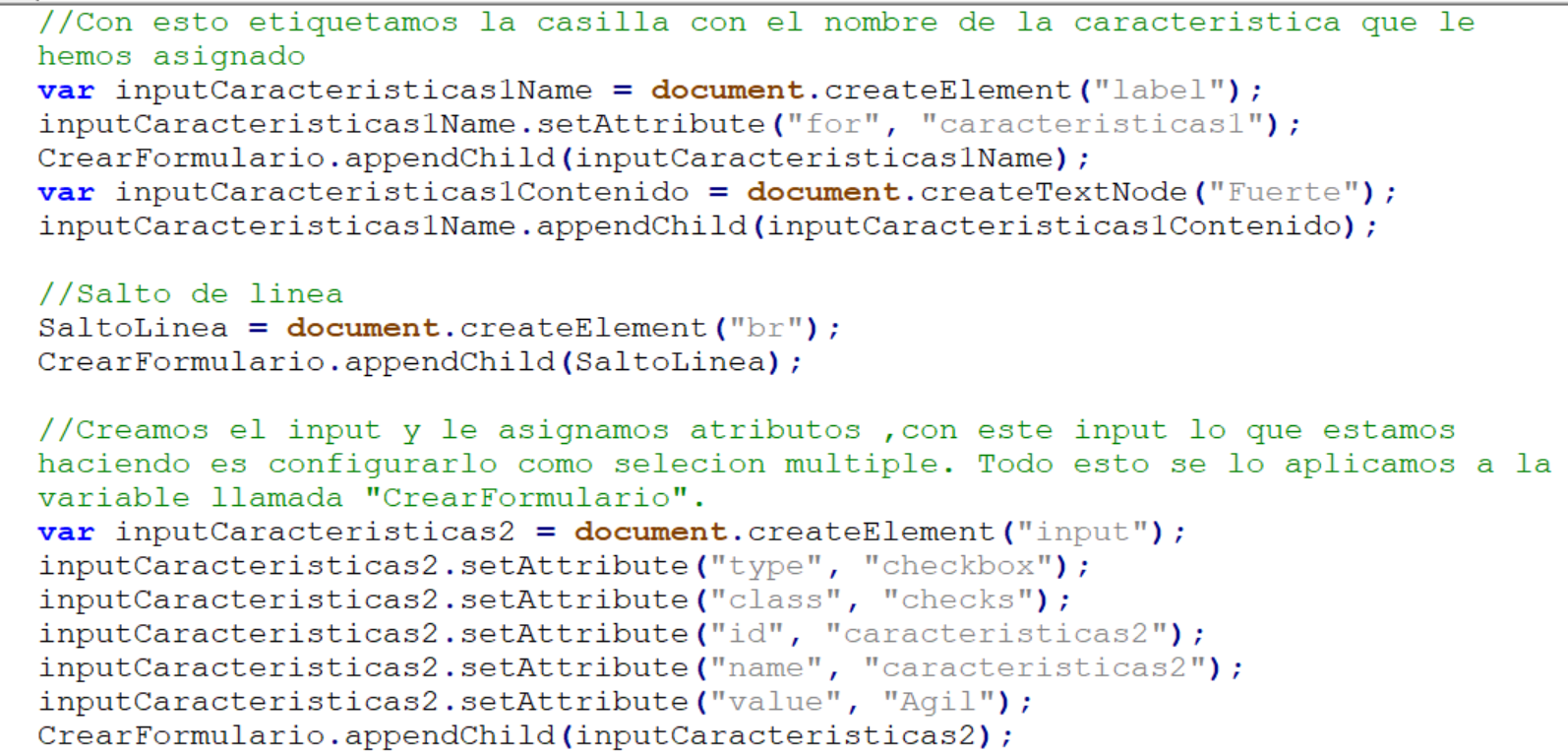


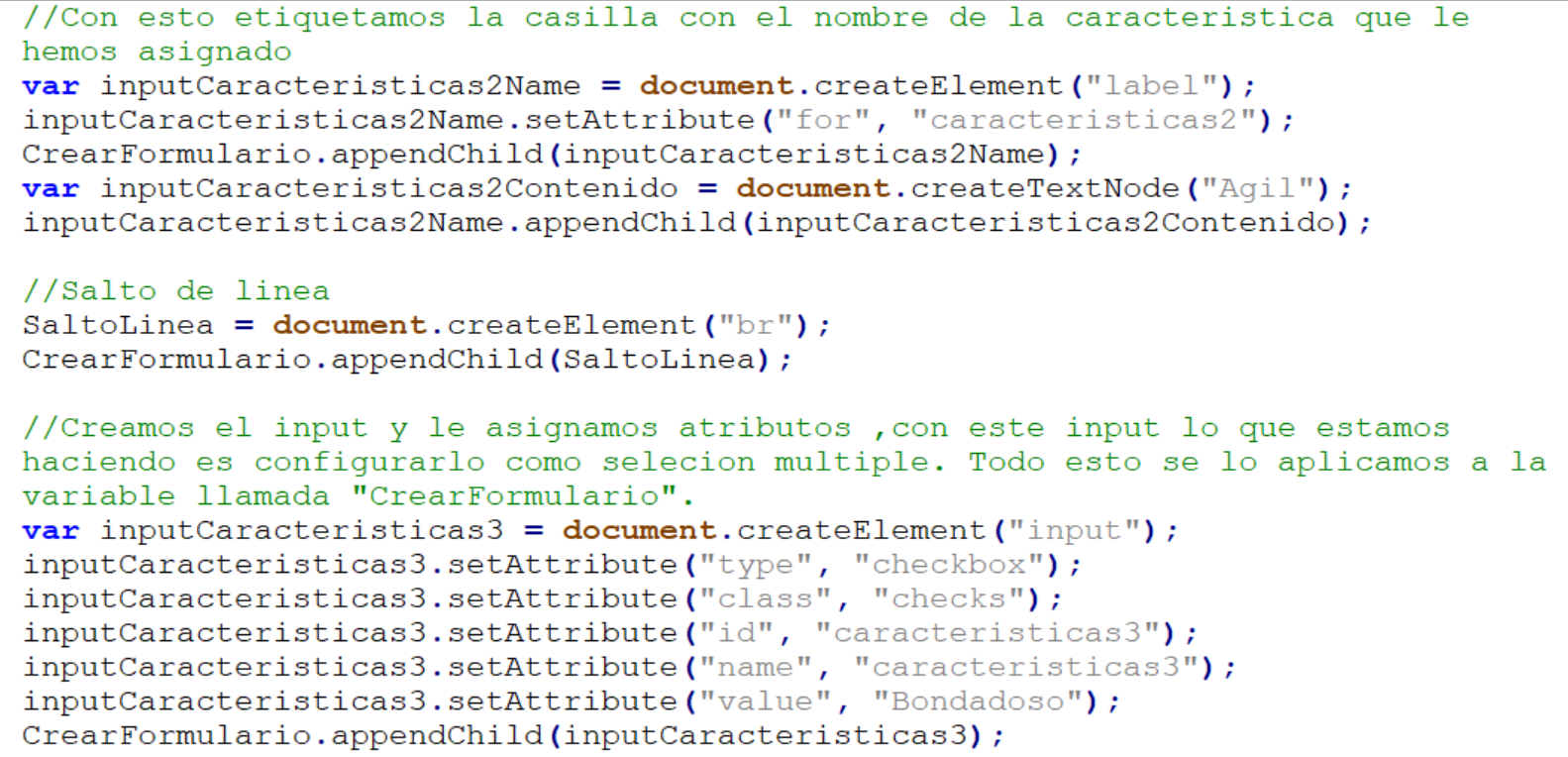


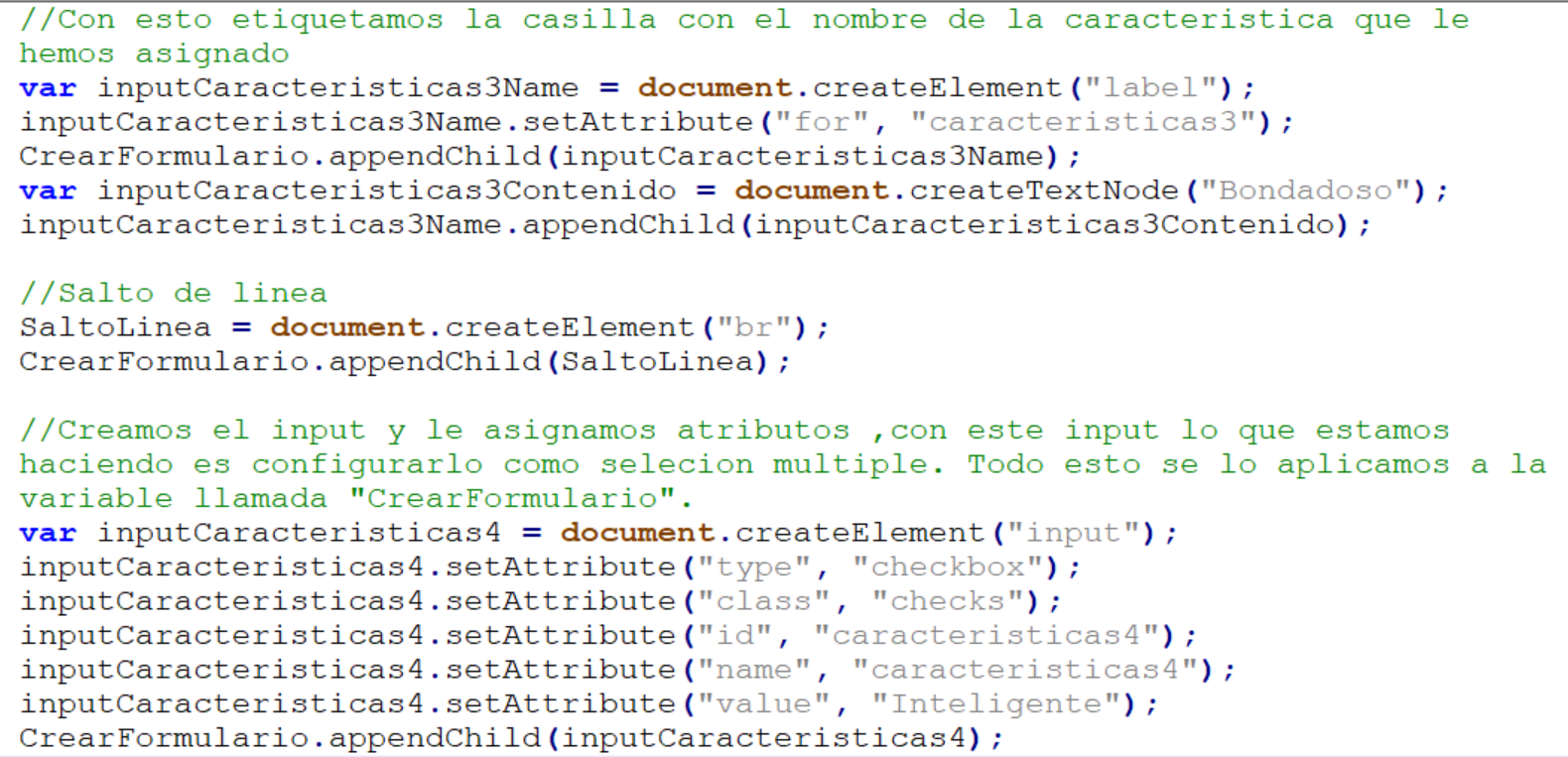


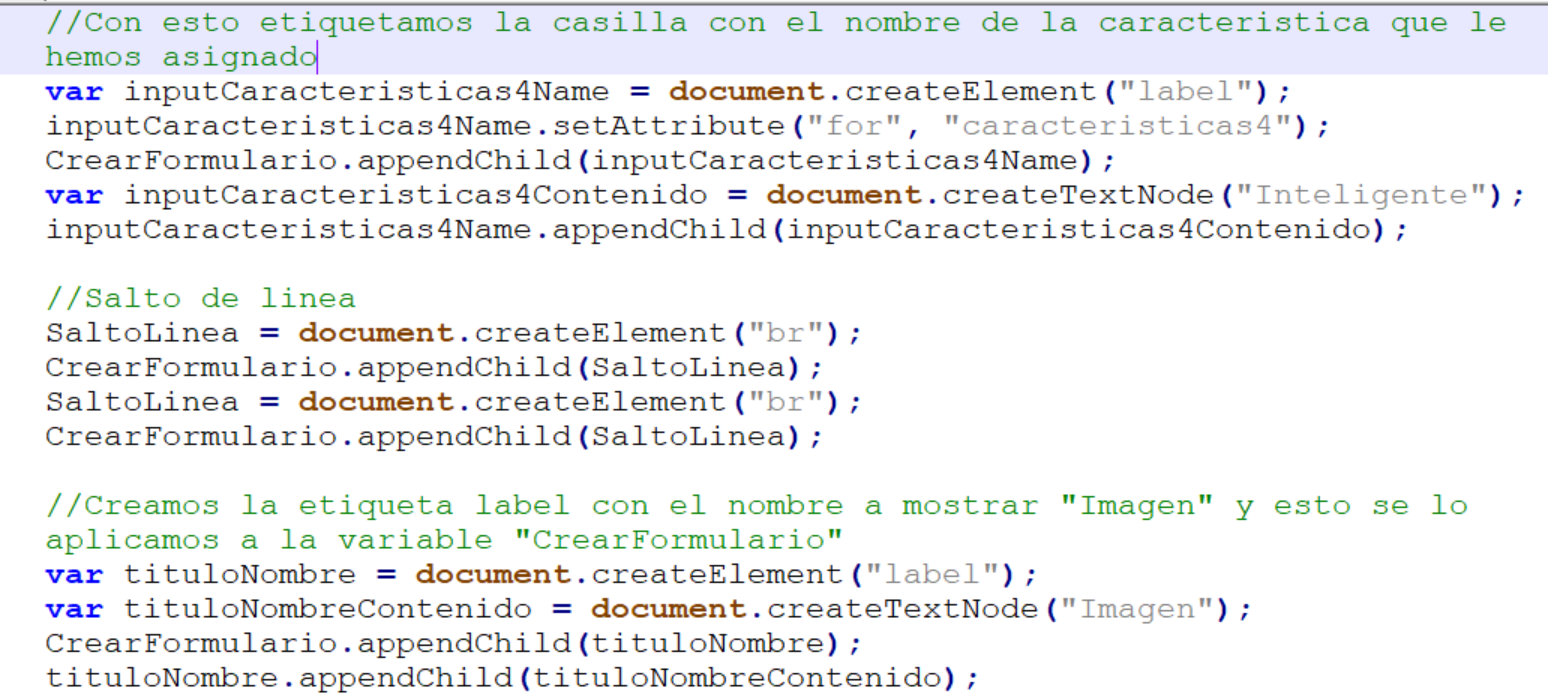


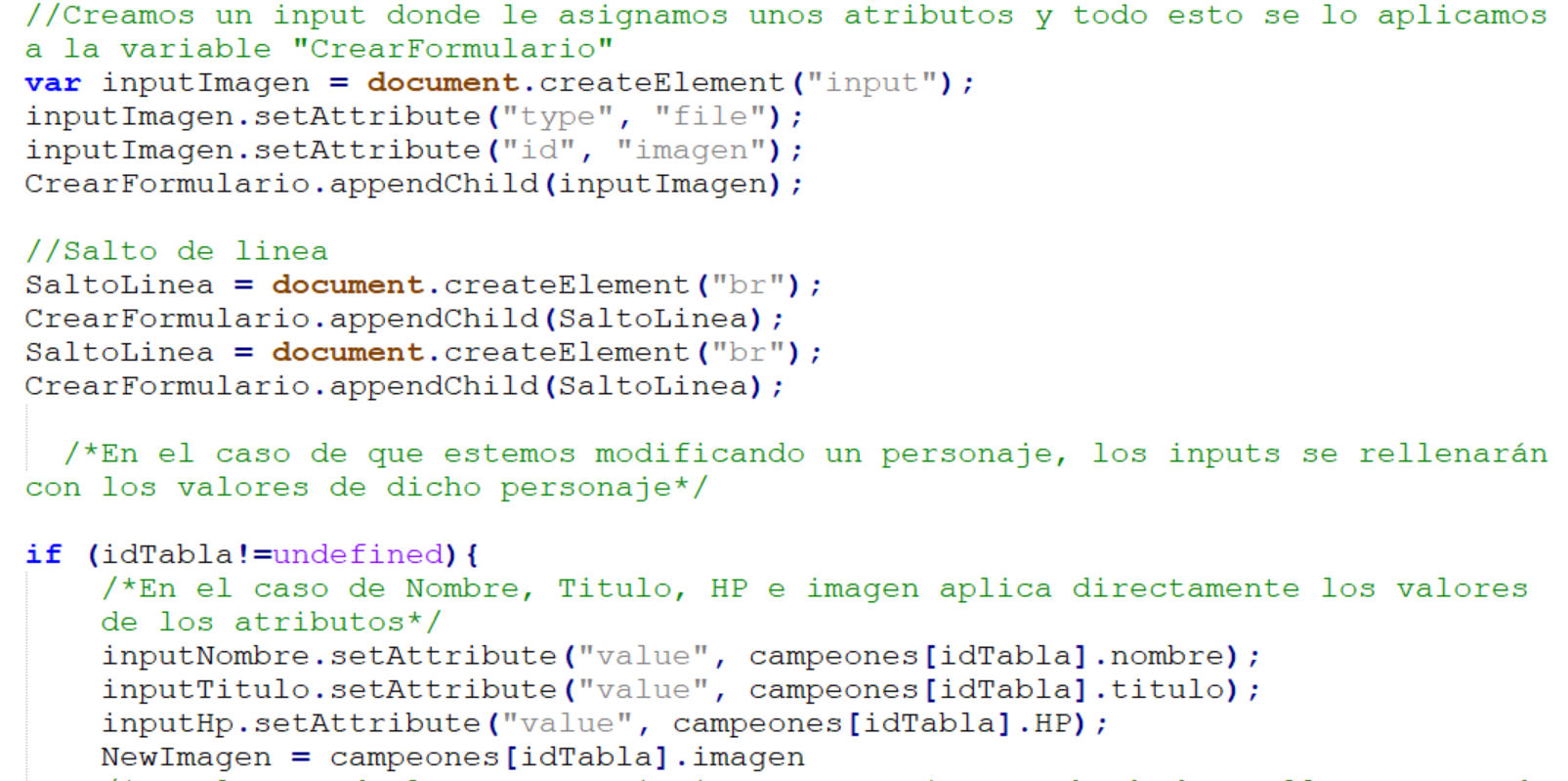


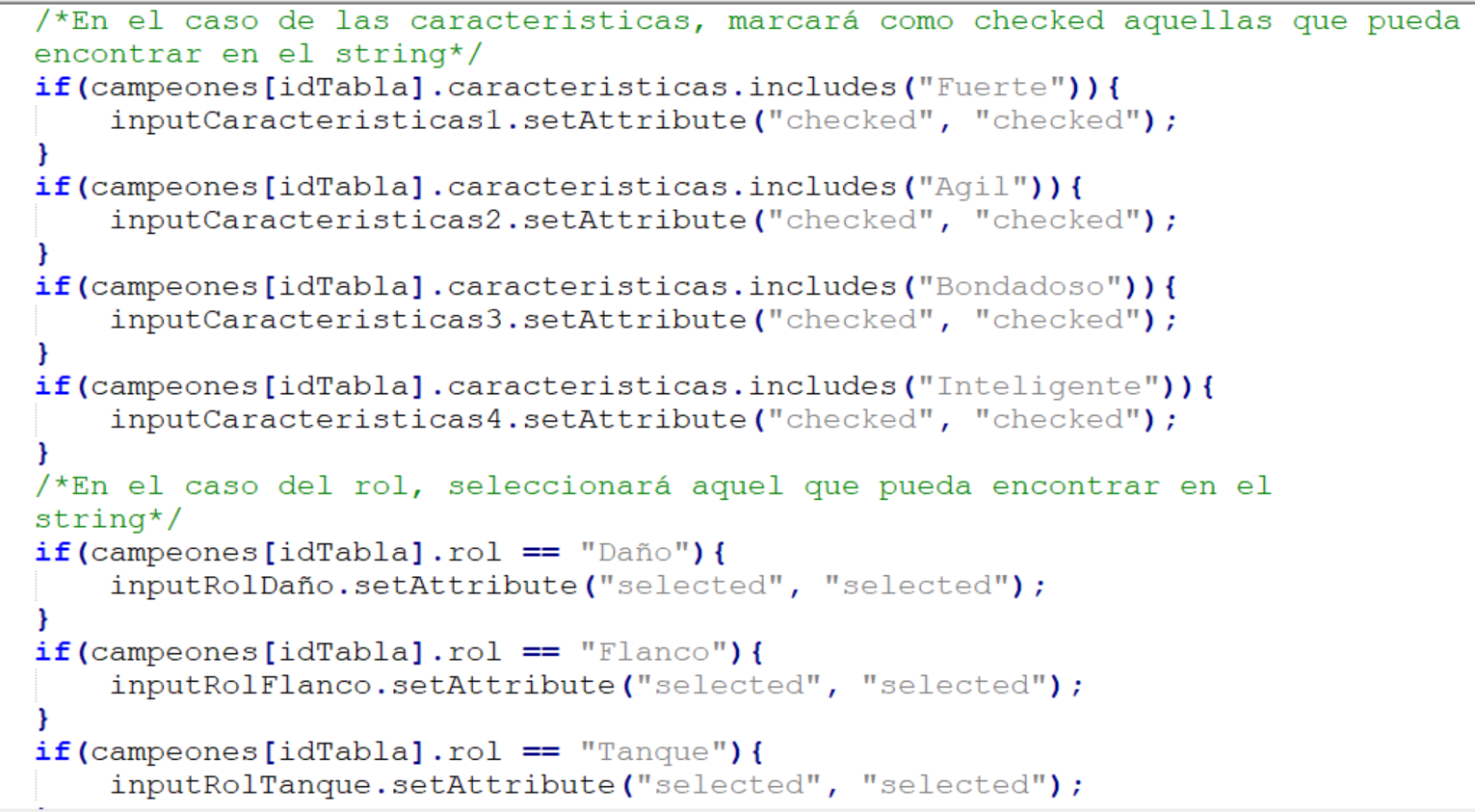


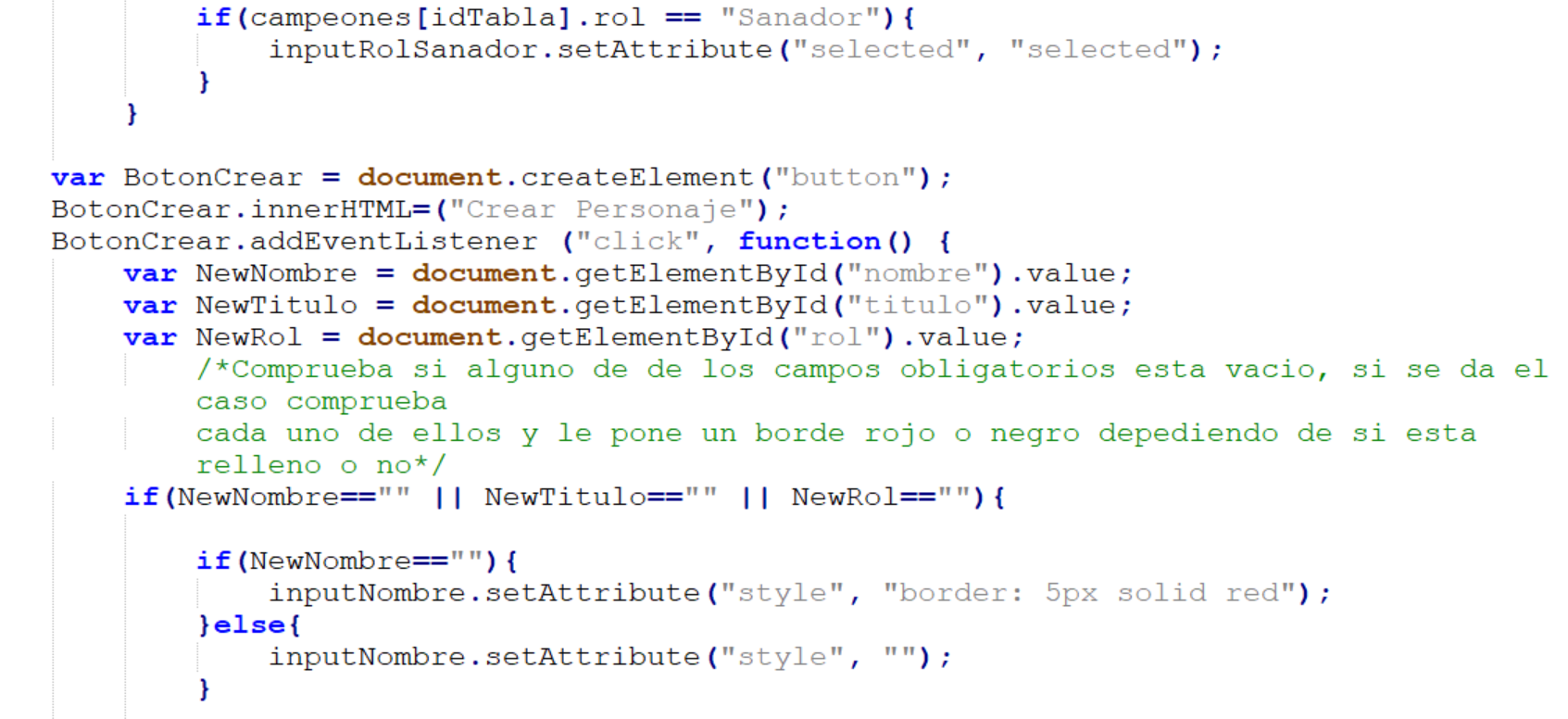


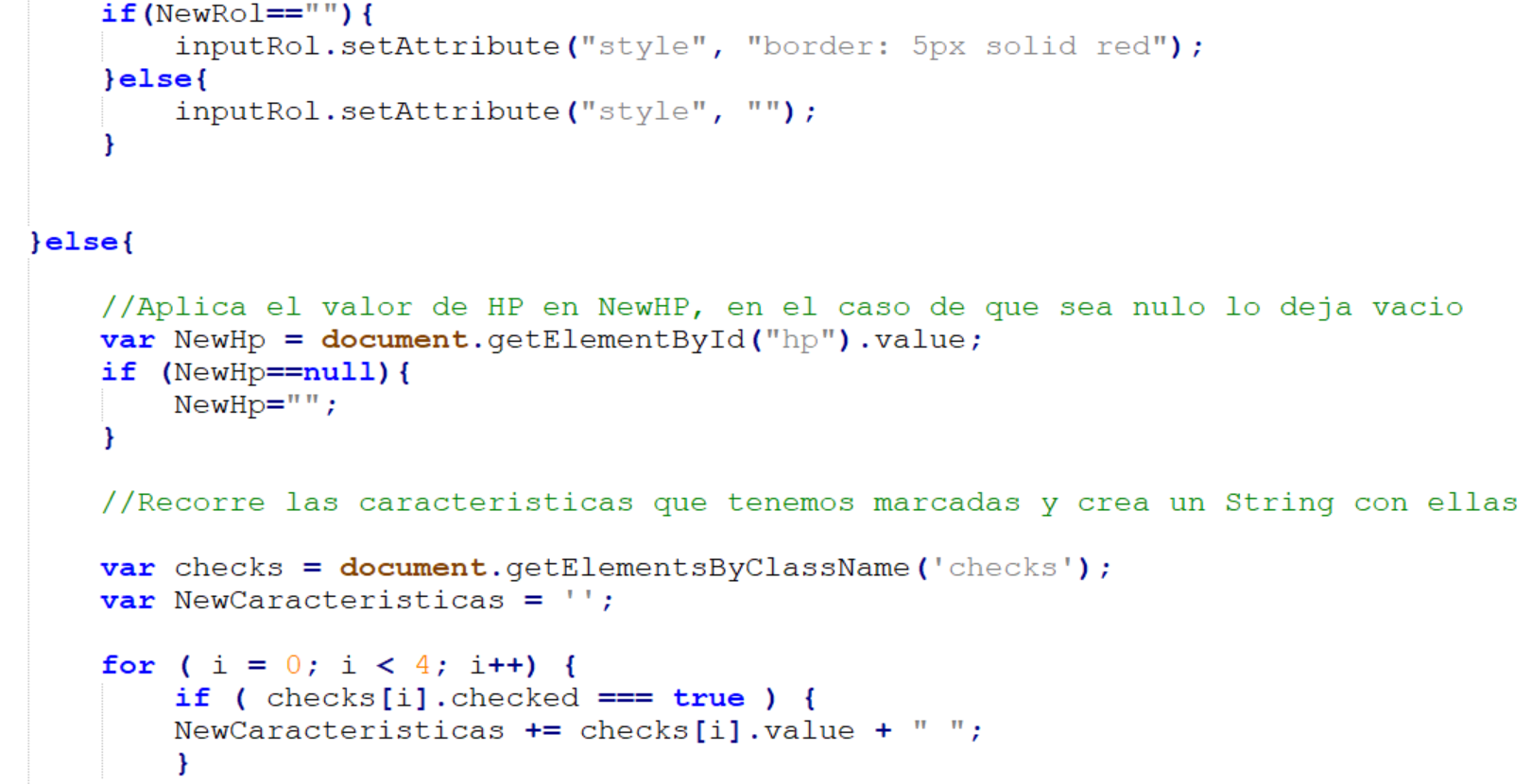


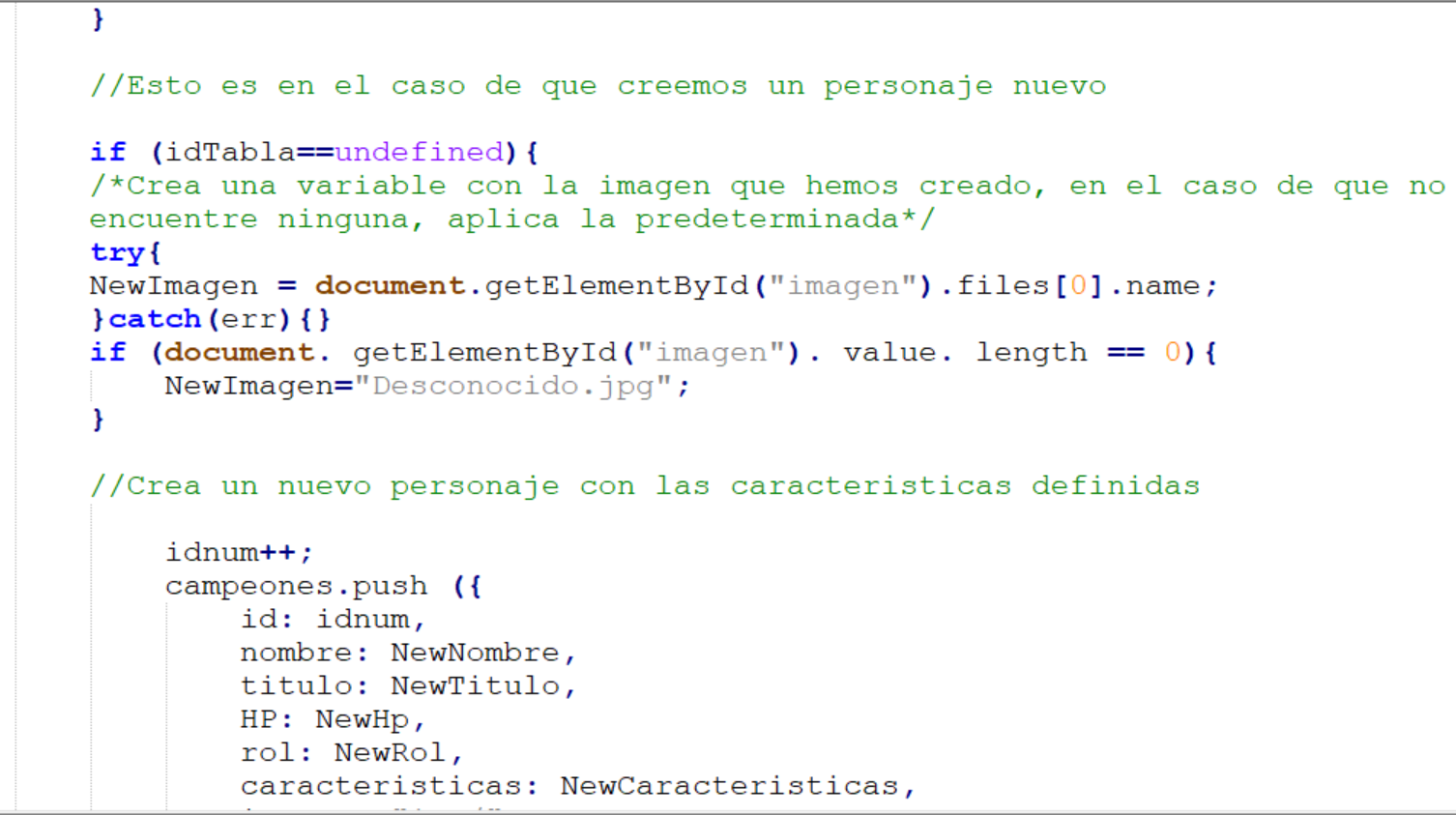


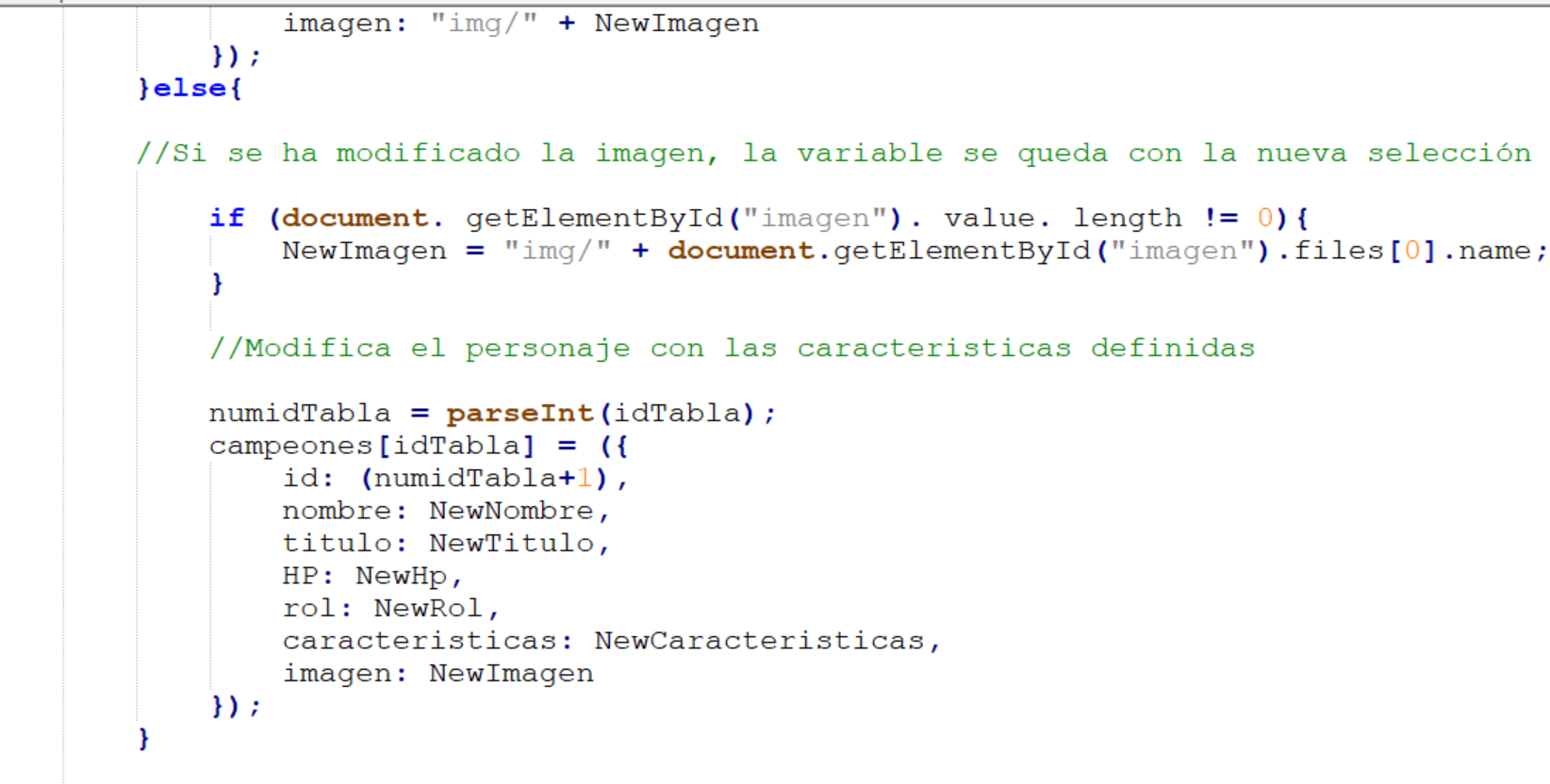


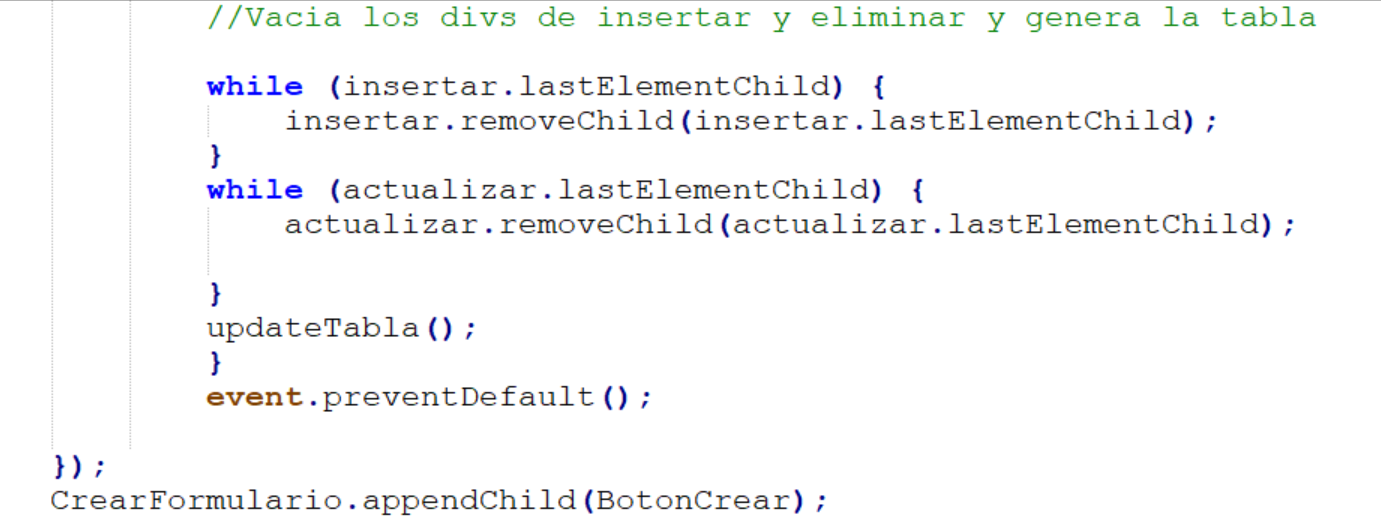


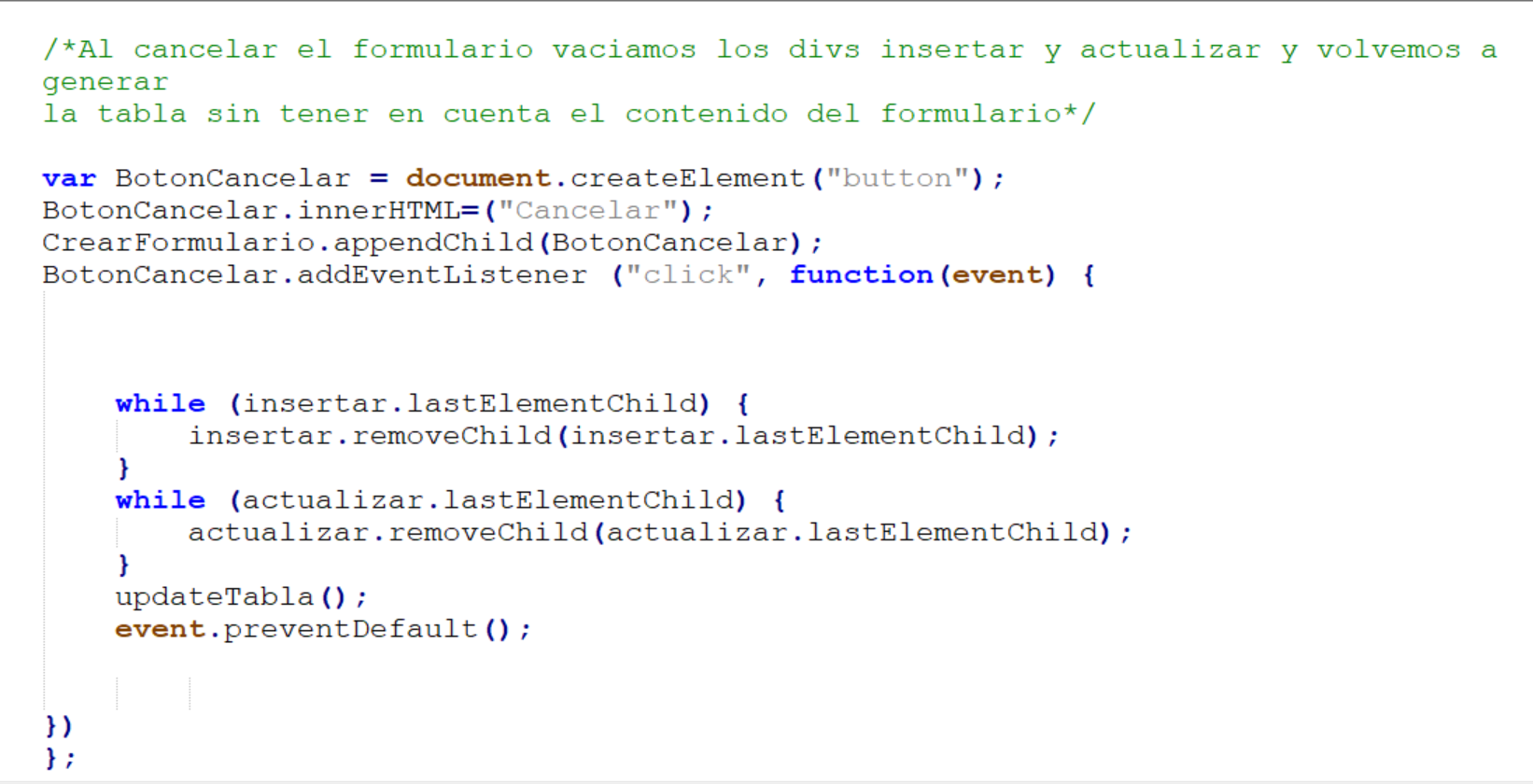












# Conclusiones

## Víctor Cobo

Este trabajo ha sido bastante útil a la hora de mostrar las virtudes de DOM, sobre todo al manipular elementos previamente creados. Eso sí, la gestión de algunos archivos como imágenes puede ser un poco confuso.

## David Ramos

Este trabajo me ha ayudado a aprender mucho mas y a superarme, pero aun así he encontrado algunas dificultades que entre los dos componentes del grupo los hemos sabido solucionar y salir de algún apuro que hemos tenido en el proyecto. Tras dedicarle tiempo a solucionar problemas que hemos tenido se nos a echo imposible llegar a los otros objetivos , pero estoy satisfecho con el trabajo sacado con Víctor ya que nos entendemos bien y sabemos trabajar en equipo.

# Problemas y partes complicada

-Generación automática de variables con eval()

-La manipulación de las variables anteriores

-Manipulación de identificadores

-Gestión de las acciones del formulario con prevenDefault