**Comentario CSS/HMTL:**

En primer lugar a la hora de trabajar con el HTML la cuestión fundamental que teníamos que plantear era: el número de contenedores totales con los que íbamos a operar y cómo los íbamos a ordenar para darle el aspecto de la calculadora.

En un primer momento iniciamos el proyecto con la idea de tres contenedores anidados. Uno principal, donde entraría toda la forma de calculadora, uno para la pantalla y otro para los botones de la misma. Comenzamos dándole esta forma e inmediatamente tuvimos que cambiar de idea puesto que, en nuestro modelo, cada uno de los botones tenía una clase única que, por implementación lógica en JS tuvimos que descartar. Algo similar ocurrió con la pantalla, la habíamos planteado en un solo contenedor, pero pronto nos dimos cuenta de que, por implementación de nuestra idea en JS esto era inviable y pasamos a dos. De esta forma el resultado final que obtuvimos tenía cuatro contenedores, uno principal queincluía la pantalla y los botones que estaba alineado con grid y dentro del de la pantalla, alineado con flex box, dos contenedores más. En el primero se mostraría el último dato introducido mediante una impresión y en el segundo, en la parte superior, el resultado.

Para los botones optamos por encuadrarlos en dos grupos, uno de numeros, para pasar los valores a JS y otro de operadores, donde se incluían (+,-,x,% e =), en un momento nos planteamos la inclusión de un botón de borrado pero suponía hacer gran cantidad de cambios en el orden HTML y algunos en el CSS.

A nivel CSS la idea era bastante más clara, simplemente había que darle a cada contenedor HMTL la alineación debida, colocando la pantalla con flex y los botones con grid (con tamaños diferentes para algunos de ellos) en una cuadrícula de 5x4 y 100 px estos últimos. Para hacer el diseño más interactivo también añadimos un pequeño sombreado para distinguir bien los botones (de no incluirlo solamente se vería el número pero no el contorno del botón) y una fuente y colores que generasen suficiente contraste para que se vieran tanto en botones como en pantalla de forma clara. Además añadimos un breve cambio de color en la activación de cada uno de los botones para que el usuario pudiere identificar de forma clara cuando se usa cada uno de los mismos.

**Comentario JavaScript:**

Para el JavaScript decidimos que lo mejor sería crear 2 clases, una para las operaciones y otra para la pantalla de la calculadora. La clase de las operaciones fue muy sencilla, consta de 4 funciones las cuales realizan las operaciones fundamentales de las matemáticas: sumar, restar, multiplicar y dividir. El verdadero problema vino en la segunda clase, la cual se encargaba de: mostrar por pantalla los números, ver qué operación se estaba haciendo, introducir los valores en la calculadora y realizar la operación en sí. Para esto tuvimos que consultar páginas como StackOverflow y muchas páginas de internet ya que estábamos bastante perdidos, aunque gracias a nuestros conocimientos en Java no se hizo tan difícil ya que había bastante similitud en muchos metodos y funciones, como el parseFloat y el constructor, que ambos funcionan igual en Java.

Otro gran problema fue la última parte de JavaScript, el apartado index.js, en el cual declaramos básicamente los valores de los números que introduces, los botones de los números, y, los botones de los operadores, y a parte, tiene 2 bucles for each los cuales, el primero, se encarga de que cuando pulses el boton de un numero, este se muestre en la pantalla, y la otra hace que el método computar funcione básicamente.

Hemos tenido que consultar muchas páginas de internet ya que había muchas dudas de como realizar muchos métodos, pero nos ha servido para aprender bastante y reforzar nuestros conocimientos.

**Comentario tests:**

Para asegurarnos de que las funciones del programa se ejecutan correctamente, previamente escribir el programa en JavaScript realizamos los tests correspondientes a las funciones de las operaciones. Para comprobar la función de la suma utilicé los números 1 y 2 y si recibe como resultado 3 (1+2) el test se completa correctamente. En el caso de la resta hice lo mismo, pero en este caso utilizando los números 2 y 1 esperando como resultado 1 (2-1). Para la división utilicé los números 4 y 2, esperando como resultado un 2 (4/2). Y por último en el caso de la multiplicación hice lo mismo que el los tres anteriores pero utilizando los números 3 y 2 esperando como resultado 6 (3\*2). Una vez terminamos el código en JavaScript lo sometimos a dichos test y los pasaron todos exitosamente.