

Memory v2.0

DESARROLLO WEB EN ENTORNO CLIENTE

SEBASTIÁN BOSCH CLADERA
2º DAW

Tabla de contenido

1-PROYECTO.....	2
2-PLANIFICACIÓN	2
Crear archivos y directorios.....	2
Código de la aplicación.....	2
CSS para la aplicación	4
Documentación.....	4
Modificaciones de la planificación.	4
Link de la planificación en Google Drive.	5
3- REGISTRO DE ERRORES	6
Link al registro de errores en Google Drive.	7
4- REGISTRO DE ACLARACIONES	8
Link al registro de aclaraciones en Google Drive.....	8
5- README	9
OBSERVACIONES	9
CONTENIDO DEL ARCHIVO.....	9
MODO DE JUEGO: ELECCIÓN DE TABLERO.....	9
MODO DE JUEGO	10
PUNTUACIÓN	10
Link al readme en Google Drive.	10
6- ÁRBOL DE DIRECTORIOS	11
7- DIAGRAMA DE CLASES.....	12
Link al árbol de directorios en Google Drive.	12

1- PROYECTO

[Link](#) al Proyecto JuegoMemoria_v2.0 en Google Drive

2- PLANIFICACIÓN

1. (10/01/2023)

Crear archivos y directorios

1. Crear HTML (index.html).
2. Crear archivo JS (JS/memory.js)
3. Crear archivo CSS (CSS/styles.css)
4. Crear directorio con imágenes para los iconos de las parejas

Código de la aplicación

1. (10/01/2023)

Añadir campos necesarios en el archivo index.html para maquetación de la aplicación.

2. (10/01/2023 a 11/01/2023)

A partir del proyecto Memory entregado al cliente, modificar en el archivo memory.js contenido en el directorio js:

- a) En la clase **Juego**, cambiar el valor de maxFilas y maxColumnas con el valor recogido del atributo max del *input text* con id "**inputFilas**" e "**inputColumnas**" mediante DOM.
- b) Modificar método *pedirDatosUsuario()*: anular condiciones ya innecesarias.
- c) Modificar método *inputDatoUsuario()*: anular condiciones ya innecesarias.
- d) Anular método *condicionesDatoInput()*.

- e) Anular método *inputTematicaParejas()*.
- f) Modificar método *arrayDeParejas()*: adaptarlo al nuevo formato DOM (clase **Memory**).
- g) Modificar método *pintarTableroEnPantalla()*: adaptarlo al nuevo formato DOM (clase **Memory**).

3. (12/01/2023)

Crear método para mostrar el tablero sin contenido: *dibujarTableroDOM()* en la clase **Juego**.

4. (13/01/2023)

Crear método para añadir un evento a cada celda/ficha del tablero en la clase **Memory**: *colocarEventoCasilla()*.

5. (13/01/2023 a 16/01/2023)

Crear método para mostrar el contenido de la casilla seleccionada tras clickar en ella (en la clase **Memory**): *mostrarContenido()*.

Añadir condiciones necesarias para comprobar si la elección es correcta o no.

6. (17/01/2023)

Crear método para el contador de tiempo del juego: *contadorSegundos()*.

7. (18/01/2023 al 19/01/2023)

Añadir control para la puntuación de la partida (por determinar).

CSS para la aplicación

1. (20/01/2023)

Maquetación de la aplicación, en el fichero estilos.css del directorio CSS, según necesidades del cliente.

Documentación (durante todo el proyecto)

- a) Documentar errores, aclaraciones con el cliente y posibles modificaciones de la planificación.
- b) Readme con aclaraciones al cliente final.

Modificaciones de la planificación.

1. (15/01/2023)

- a) Debido a las dificultades para implementar la entrada de datos mediante DOM, se opta por seguir el desarrollo de la aplicación solicitando al jugador el número de filas, columnas y la temática del tablero mediante "prompts", tal y como se realizaba en la versión anterior de la aplicación.
- b) En la clase **Juego**, se retorna al valor de maxFilas y maxColumnas en el constructor de la clase.
En el método *pedirDatosUsuario()* e *inputDatoUsuario()* se reincorporan las condiciones necesarias para la creación del tablero.
Se reincorporan los métodos *condicionesDatoInput()* e *inputTematicaParejas()*.
- c) En la clase **Memory**, se añade el método *despejar(parámetro)*.
- d) Se modifica el método *pintarTableroEnPantalla()*.

2. (16/01/2023)

- a) En la clase **Memory**, se añade el método *mostrarContenido(parametro)*.
- b) Modificado el método *arrayDeParejas()*; previsto para el (10-11/01/2023).

3. (17/01/2023)

- a) En la clase **Memory**, se añade el método *pararContadorSegundos()*.
- b) En la clase **Memory**, se añade el método *reiniciar()*.
- c) En la clase **Memory**, se añade el método *puntuacionParejas()*.
- d) Añadido botón de reinicio en el fichero index.html.

4. (21/01/2023)

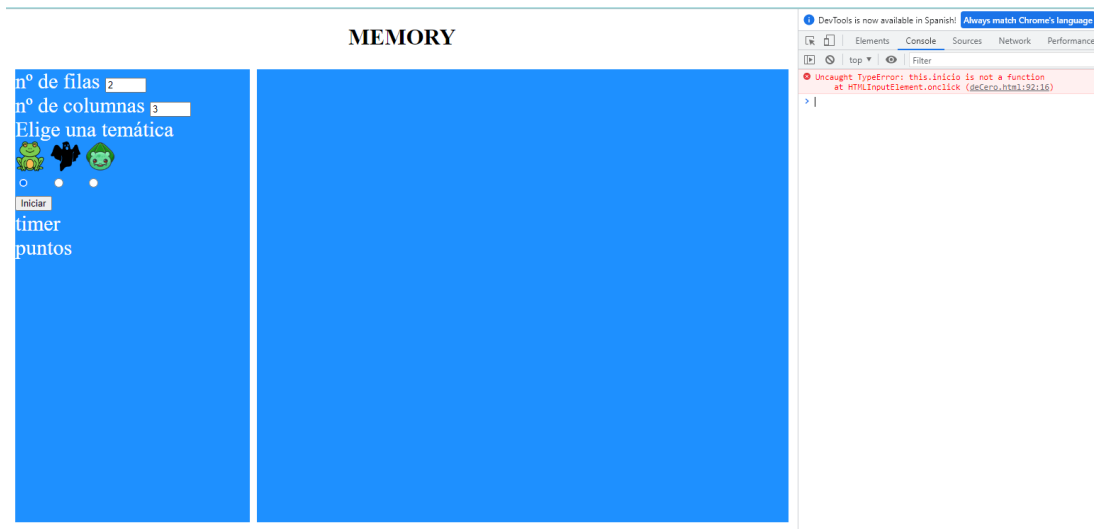
- a) En la clase **Memory**, se añaden los métodos *sonIguales(param1, param2)*, *ocultarCeldas(param1, param2)* y *partidaGanada()*, para modularizar más el código.
- b) En la clase **Juego**, se modifica la nomenclatura de los métodos *inputDatoUsuario()*, *pedirDatosUsuario()* y *condicionesDatoInput()*, para clarificar y posibilitar una mejor lectura del código, así quedan los métodos:
 - a. *inputDatoUsuario()* pasa a *filasColumnasTablero()*.
 - b. *pedirDatosUsuario()* pasa a *comprobarNumeroParejas()*.
 - c. *condicionesDatoInput()* pasa a *condicionesFilasColumnas()*.

[Link](#) de la planificación en Google Drive.

3- REGISTRO DE ERRORES

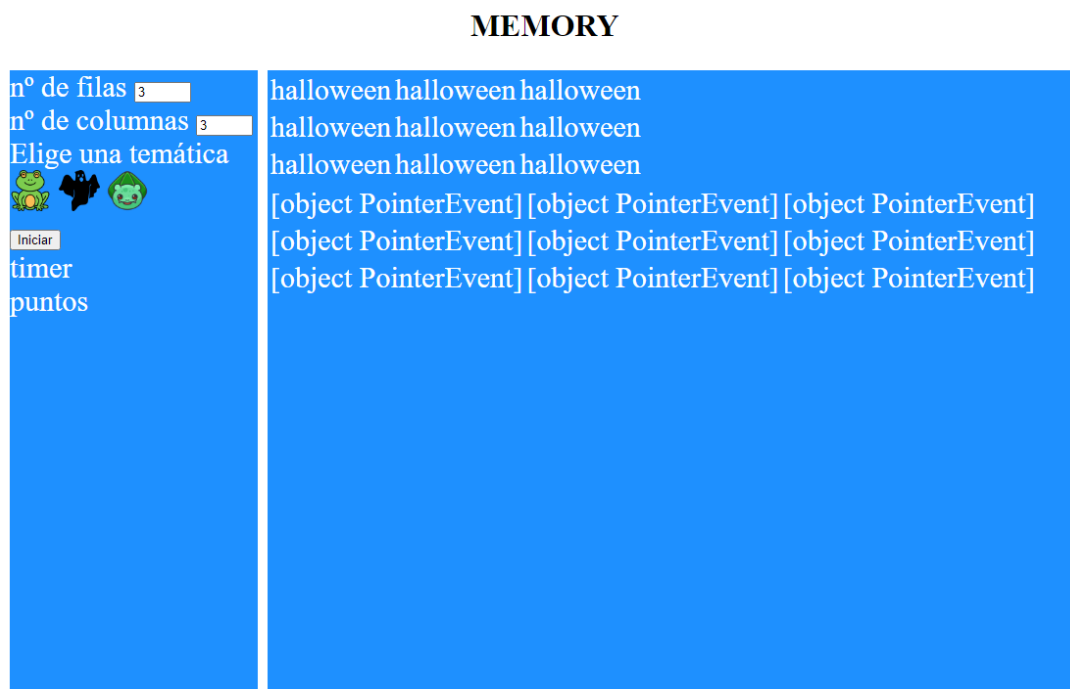
1. (09/01/2023)

No se consigue extraer los datos sobre la temática de la partida correctamente. Problemas en el ámbito de la función.



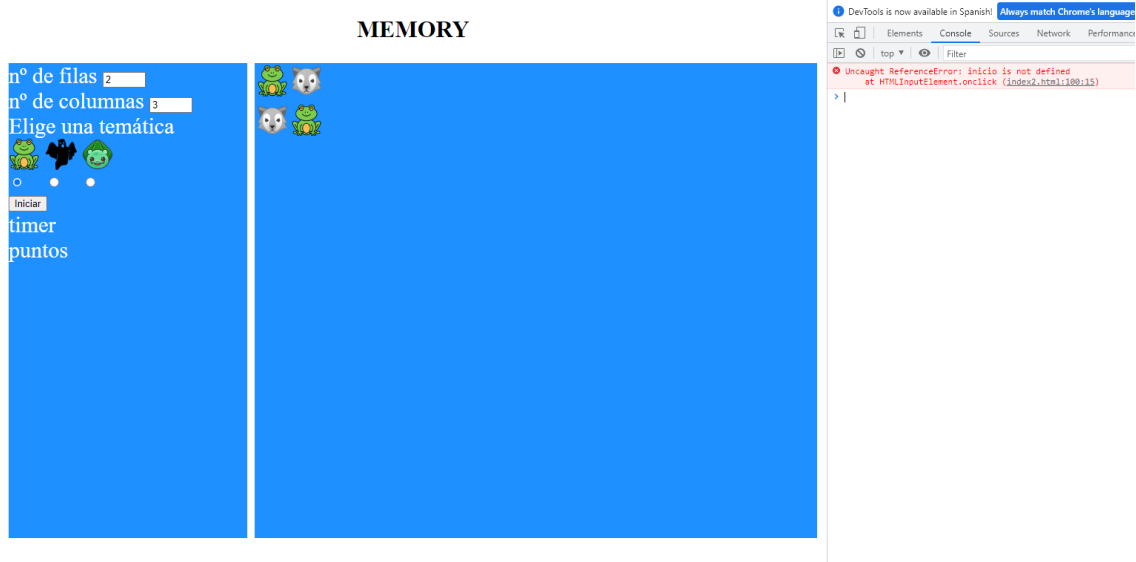
2. (10/01/2023)

No se consigue imprimir la tabla en pantalla tras pulsar el botón "iniciar".



3. (11/01/2023 al 13/012023)

Se consigue imprimir la tabla en pantalla tras pulsar el botón “iniciar”, pero no recoge los valores de los inputs de filas y columnas.



4. (14/01/2023)

Debido a la imposibilidad de solucionar los errores, se opta por continuar con el desarrollo de la aplicación en términos de funcionalidad.

[Link](#) al registro de errores en Google Drive.

4- REGISTRO DE ACLARACIONES

1. (09/01/2023)

Presentar un boceto de cómo será la presentación de la aplicación. (El cliente está de acuerdo con el diseño y solicita que la entrada de datos del usuario sea recogida a través de DOM).



2. (20/01/2023)

Debido a problemas de implementación, se consulta con el cliente si es posible dejar la nueva interfaz para la entrada de datos de la aplicación para futuras actualizaciones, a lo que éste responde afirmativamente. Se mantiene el formato el formato de la versión anterior de la aplicación.

[Link](#) al registro de aclaraciones en Google Drive.

5- README

OBSERVACIONES

Esta aplicación está configurada para el navegador Google-Chrome (probado también en Microsoft Edge y Mozilla), con una configuración para pantallas de 15 pulgadas o superior.

El número máximo de filas y columnas queda determinado por dicha medida de pantalla y el número de píxeles que mide cada imagen.

CONTENIDO DEL ARCHIVO

El archivo comprimido contiene 1 aplicaciones Memory:

1) "ProyectoMemory_v2.0": contiene la versión del juego.

MODO DE JUEGO: ELECCIÓN DE TABLERO

En el primer prompt (número de filas), el usuario debe entrar como valor un número positivo entre 1 y 4 (ambos incluidos). Cualquier otro valor diferente a éstos, provocará que se vuelva a pedir un número de filas.

En el segundo prompt (número de columnas), el usuario debe entrar como valor un número positivo entre 1 y 10 (ambos incluidos).

Cualquier otro valor diferente a éstos, provocará que se vuelva a pedir un número de columnas.

Si el número de celdas del tablero es impar (por la multiplicación de filas y columnas) o el número de parejas es inferior a 2, el juego volverá a iniciar el proceso de entrada de datos.

En el tercer prompt (temática de parejas), el usuario debe entrar un valor para la elección de las parejas, siendo aceptados; "1", "2", "3", "animales", "halloween" y "pokemon" (en estos dos últimos casos la

aplicación NO es "case sensitive"). Cualquier otro valor de entrada iniciará el proceso de entrada de datos para la temática de parejas.

Si todas las entradas de datos son correctas, la aplicación mostrará el tablero con las opciones elegidas por el usuario.

MODO DE JUEGO

El juego consiste en descubrir dos casillas, si ambas casillas coinciden, éstas quedan vistas durante el resto del juego o partida.

Si las casillas descubiertas no son iguales, en este caso, las casillas se vuelven a ocultar y el jugador deberá descubrir otras dos casillas.

Una vez que el jugador consiga descubrir todo el tablero, el juego finalizará, dando la opción de iniciar otra partida.

PUNTUACIÓN

Cuando el jugador descubra dos casillas en el primer intento, es decir, cuando dichas casillas descubiertas no coincidan en ningún caso con las dos anteriores seleccionadas, se le otorgarán 10 puntos.

En el caso que las casillas descubiertas coincidan en algún caso (ya sea la primera o segunda casilla descubierta), le serán otorgados 5 puntos.

Finalmente, si al descubrir una pareja de casillas éstas no coinciden, el jugador volviera a descubrir dos casillas y una de ellas coincidiera con algunas de las anteriores y no fueran iguales y se realizara un tercer intento, en el que alguna de las casillas seleccionadas coincidiera con las descubiertas en el intento anterior, el jugador recibiría 2.5 puntos por el acierto.

En cualquier otro caso, el jugador no recibirá ningún punto.

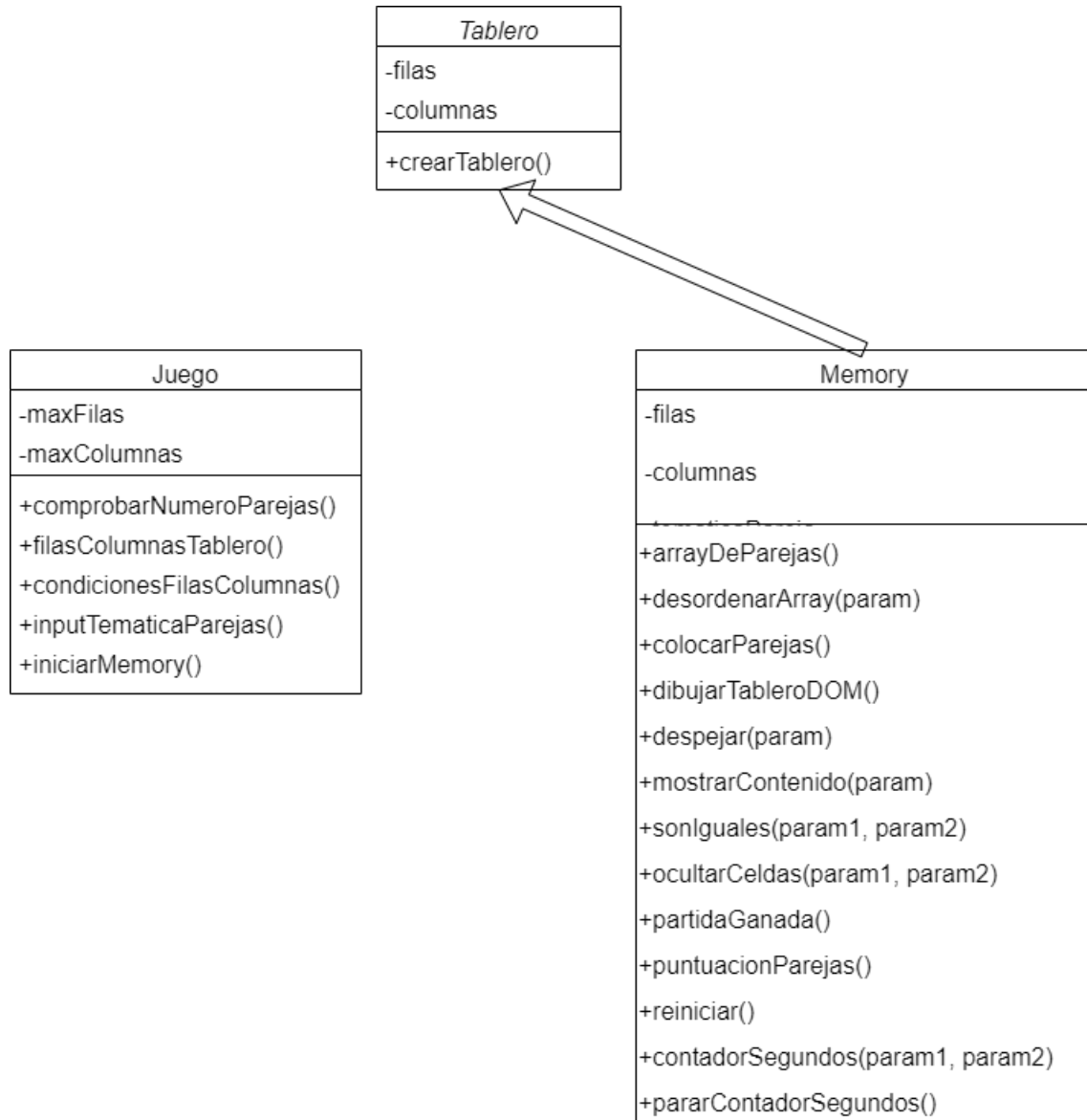
[Link](#) al readme en Google Drive.

6- ÁRBOL DE DIRECTORIOS

```
C:\Users\sebas>tree /F Desktop\ProyectoMemory_v2.0
Listado de rutas de carpetas
El número de serie del volumen es 00000004 7612:29EF
C:\USERS\SEBAS\DESKTOP\PROYECTOMEMORY_V2.0
    index.html
    README.txt
    -css
        DS-DIGIT.TTF
        estilos.css
    -imagenes
        -animales
            0.png
            1.png
            2.png
            3.png
            4.png
            5.png
            6.png
            7.png
            8.png
            9.png
        -halloween
            0.png
            1.png
            2.png
            3.png
            4.png
            5.png
            6.png
            7.png
            8.png
            9.png
        -pokemon
            0.png
            1.png
            2.png
            3.png
            4.png
            5.png
            6.png
            7.png
            8.png
            9.png
    -js
        memory.js
```

[Link](#) al árbol de directorios en Google Drive.

7- DIAGRAMA DE CLASES



[Link](#) al árbol de directorios en Google Drive.