

MEMORIA PROYECTO DIRIGIDO

FECHA	29-04-2023
APELLIDOS	Biurrun Lorda
NOMBRE	Luar
APELLIDOS	Ferrer Ferrández
NOMBRE	Miguel

ESPACIO COMPARTIDO A EVALUAR (SOLO UNO!)	<i>Ferrer Ferrández</i>
--	-------------------------

A la hora de contestar, extiende los cuadros destinados a rellenar tanto como sea necesario. Sed concisos. Podéis incluir (de hecho, resulta conveniente) diagramas/imágenes o copiar y pegar pequeños segmentos del código para apoyar la explicación.

A. Describe el grado de cumplimiento de los requisitos del enunciado.

Se han implementado todas las funciones que se exigen en el guión del proyecto.

B. Explica cómo habéis estructurado el modelo de datos

La organización de los datos de la página web se ha realizado de la siguiente manera.

Para los ficheros externos como imágenes se ha creado una carpeta que contiene todas la imágenes que se utilizan en la web, tanto png, cómo svg.

Para el flujo entre páginas se ha creado un array que contiene el orden de apertura de las aplicaciones. Cuando una aplicación se abre se añade a la cola si no existía en la lista. Si la aplicación ya estaba en la lista entonces la desplazamos a la última posición del array. Guardamos el array en sessionStorage serializado como un json.

Para las aplicaciones de editar, tanto para texto como para imagen se crea un objeto en memoria que contiene cinco parámetros (nombre, extensión, contenido, contenido no guardado y fecha). Este objeto lo guardaremos en memoria (sessionStorage) serializado como json y lo usaremos para comprobar si se ha guardado el contenido o no y para guardarlo en memoria cuando el usuario lo requiera. Tendremos un objeto para el editor de texto con extensión “txt” y otro para el editor de imagen con extensión “img”.

Para el almacenamiento de los ficheros de texto e imagen se han creado cuatro arrays de objetos. Dos arrays corresponden a los objetos de texto o imagen guardados en la memoria. Los otros dos arrays corresponden con los objetos de texto o imagen eliminados. Se han tratado por separado cada uno para simplificar el código. Los objetos guardados en los arrays contarán con cuatro parámetros:

1. Nombre
2. Extensión
3. Contenido
4. Fecha y hora

Se ha utilizado una clase con un constructor para crear los objetos.

```
class fileTextImage {  
    constructor(extension) {  
        this.name = "";  
        this.content = "";  
        this.date = "";  
        this.extension = extension;  
    }  
}
```

En cuanto al contenido de las imágenes será un array de 64 valores que contendrá el color del cuadrado correspondiente.

C. Describe las técnicas usadas y el estilo de programación

Hemos orientado la programación a objetos ya que es más cómodo para acceder a variables sin tener que declararlas previamente y nos permite tener un código más limpio y depurado. Además que para guardar los datos en memoria es recomendable utilizar objetos serializados como json.

Como los métodos no se conservan cuando guardamos un objeto en memoria hemos prescindido de crear métodos para trabajar solamente con los atributos necesarios e imprescindibles para el buen funcionamiento del programa.

La técnica utilizada para las alertas es ocultar y mostrar código de la página web en vez de añadir código en la página como se hace a la hora de mostrar ficheros de texto o imagen tanto en el buscador como en la papelera. Los elementos de texto e imagen que aparecen tanto en el buscador como en la papelera se generan creando componentes y asignándoles atributos en vez de mediante código html puro utilizando innerHTML, esto nos permite mayor flexibilidad y segmentación del código.

D. Explica cómo habéis trabajado el diseño y maquetación de la página

Para el diseño y maquetación se ha seguido el siguiente procedimiento.

MAQUETACIÓN

Cada aplicación se encuentra en una página html independiente.

En cuanto a la maquetación la página de escritorio cuenta con 3 secciones:

1. Sección con título, slogan y participantes del proyecto.
2. Sección de navegación con botones
3. Sección con la información de la fecha y hora actual

Para la maquetación de las aplicaciones de buscar ficheros y papelera contamos con 6 secciones :

1. Sección con nombre de la aplicación y botón de salir de la aplicación.
2. Sección con las pestañas de texto o imagen
3. Sección con los ficheros de texto e imagen creados o eliminados. Los ficheros mostrarán el contenido en pequeña escala.
4. Sección con una barra de progresión con la memoria que está ocupada.
5. Sección con la barra de navegación y los botones
6. Sección con la información de fecha y hora actual.

Para la maquetación de las aplicaciones de editar ficheros de texto o imagen se han utilizado 5 secciones:

1. Sección con nombre de la aplicación y botón de salir de la aplicación.
2. Sección con las opciones de edición de cada aplicación
3. Sección con el editor de texto o imagen

Para el caso del texto, el editor es un área de texto

Para el caso de imagen, el editor es un grid de 8x8 cuadrados

4. Sección con la barra de navegación y botones.
5. Sección con la información de fecha y hora.

DISEÑO

Para el diseño de la página web se ha separado en diferentes ficheros de estilo, de más general a más específico.

Todas las aplicaciones comparten general.css y allAppsStyle.css y luego cada aplicación cuenta con sus estilos más específicos.

El estilo general se ha realizado mediante el uso de un grid de 4x3 (4 filas, 3 columnas), para mantener la uniformidad de la página web en las distintas aplicaciones.

Todas las alertas están ocultas y se muestran y ocultan utilizando la clase `hiddenObject` conforme se van necesitando. Lo mismo sucede a la hora de mostrar y ocultar los ficheros de texto o imagen.

Todos los botones se cambian de color cuando el raton está puesto encima y en el caso del editor de imagen, el ratón cambia de forma para el caso de estar usando la brocha o el cubo. Y la paleta cambia de color en función del color seleccionado.