DWC\examenFInal\webAPIS\fileREader.html

```html
<!DOCTYPE html>
<html lang="es">

<head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <style>
        main {
            display: flex;
            flex-direction: column;
            gap: 1rem;
            align-items: center;
        }

        #dropZone {
            width: 300px;
            height: 150px;
            border: 2px dashed #007bff;
            display: flex;
            align-items: center;
            justify-content: center;
            text-align: center;
            font-size: 16px;
            color: #007bff;
            cursor: pointer;
            padding: 10px;
        }

        #dropZone.dragover {
            background-color: #e0f7fa;
        }

        #preview {
            margin-top: 10px;
            max-width: 100%;
            text-align: center;
        }

        img {
            max-width: 300px;
            max-height: 300px;
        }

        pre {
            white-space: pre-wrap;
            background: #f4f4f4;
            padding: 10px;
            border-radius: 5px;
            max-width: 90%;
            overflow-x: auto;
            text-align: left;
        }
    </style>
</head>

<body>
    <main>
```

```
        <h2>Adjuntar archivo</h2>

        <!-- Input para seleccionar archivo -->
        <input type="file" id="fileInput" accept="image/*, .txt">

        <!-- Área de Drag and Drop -->
        <div id="dropZone">Arrastra y suelta un archivo aquí</div>

        <!-- Vista previa del archivo -->
        <div id="preview"></div>
    </main>

    <script>
        const fileInput = document.querySelector("#fileInput");
        const dropZone = document.querySelector("#dropZone");
        const preview = document.querySelector("#preview");

        // Función para leer y previsualizar archivos
        function handleFile(file) {
            const reader = new FileReader();//crea instancia de filerader que nons permite leer ficheros

            reader.onload = function(event) {//cuando la lectura se completa
                const result = event.target.result;//Obtiene el contenido leído del archivo
                /*
                1️⃣ event
                Cuando se ejecuta la función reader.onload, se dispara un evento.
                Este evento (event) contiene información sobre la operación de lectura.

                2️⃣ event.target
                El target del evento es el objeto que disparó el evento, en este caso, el FileReader (reader).
                Es decir, event.target hace referencia a reader.

                3️⃣ event.target.result
                La propiedad result contiene los datos leídos del archivo.
                El formato de result depende del método de lectura usado:
                readAsDataURL(file) → Devuelve una cadena Base64 (para imágenes).
                readAsText(file) → Devuelve una cadena de texto (para archivos .txt).
                */

                if (file.type.startsWith("image/")) { // Si el archivo es una imagen, lo muestra en una etiqueta <img>
                    preview.innerHTML = `<img src="${result}" alt="Vista previa">`;
                } else if (file.type === "text/plain") {// Si el archivo es un texto plano, lo muestra dentro de una etiqueta <pre>
                    preview.innerHTML = `<pre>${result}</pre>`;
                } else {// Si el archivo no es compatible, muestra un mensaje de error
                    preview.innerHTML = `<p>Tipo de archivo no soportado para vista previa.</p>`;
                }
            };

            if (file.type.startsWith("image/")) {// Si el archivo es una imagen, se lee como Data URL para que pueda mostrarse en <img>
                reader.readAsDataURL(file);
            } else if (file.type === "text/plain") {// Si el archivo es de texto, se lee como texto plano
                reader.readAsText(file);
            }
        }
```

```
112
113         // Evento para input file
114         fileInput.addEventListener("change", function(event) {//el evento change se activa cuando el
    usuario selecciona un archivo.
115             const file = event.target.files[0];//event.target hace referencia a fileInput.
116                                                //files es una lista de archivos seleccionados.
117                                                //[0] obtiene el primer archivo (en caso de que se
    permita subir varios).
118             if (file) handleFile(file); //handleFile(file), que se encarga de leer y mostrar el
    archivo.
119
120         });
121
122         // Eventos de Drag & Drop
123         dropZone.addEventListener("dragover", function(event) {
124             // Previene el comportamiento por defecto del navegador
125             // para permitir soltar archivos en la zona de drop.
126             event.preventDefault();
127             // Agrega una clase CSS ("dragover") para resaltar la zona de drop
128             // mientras el usuario arrastra un archivo sobre ella.
129             dropZone.classList.add("dragover");
130         });
131
132         dropZone.addEventListener("dragleave", function() {// Se ejecuta cuando el usuario saca el
    archivo arrastrado
133                                                            // fuera de la zona de drop.
134
135             dropZone.classList.remove("dragover");// Remueve la clase "dragover" para restaurar el
    estilo original de la zona de drop.
136         });
137
138         dropZone.addEventListener("drop", function(event) {//Se agrega un evento drop a dropZone, que
    se activa cuando el
139                                                            // usuario suelta un archivo dentro de la
    zona de drop.
140             event.preventDefault(); // Evita que el navegador abra el archivo
141             event.stopPropagation(); // Evita propagación de eventos
142
143             dropZone.classList.remove("dragover");//// Elimina la clase "dragover" para restaurar el
    estilo original de la zona de drop.
144
145             const file = event.dataTransfer.files[0];// Obtiene el archivo arrastrado desde la
    propiedad dataTransfer.
146             if (file) handleFile(file);// Si hay un archivo válido, lo envía a la función handleFile()
    para su procesamiento.
147         });
148
149         // Previene que el navegador abra archivos arrastrados en cualquier parte de la página
150         document.addEventListener("dragover", (event) => event.preventDefault());
151         document.addEventListener("drop", (event) => event.preventDefault());
152
153         /*
154         🔍 ¿Por qué no solo drop?
155             Si solo usáramos drop, el navegador no sabría que el usuario está arrastrando un archivo
    hasta que realmente lo suelte.
156             No podríamos cambiar el estilo de la zona cuando el usuario arrastra un archivo sobre ella.
157             No podríamos manejar correctamente la interacción cuando el usuario saca el archivo sin
    soltarlo.
158             El navegador podría bloquear la funcionalidad de drop si dragover no está presente.
159
160             1️⃣ dragover → Se activa cuando el usuario arrastra un archivo sobre la zona de drop
```

```
            ✅ ¿Por qué es necesario?
                event.preventDefault() es obligatorio para permitir que el evento drop funcione.
                Cambia el estilo de la zona para indicar al usuario que puede soltar el archivo (por ejemplo, resaltando el área).

            2️ dragleave → Se activa cuando el usuario saca el archivo de la zona sin soltarlo
            ✅ ¿Por qué es necesario?
                Evita que la zona de drop se quede resaltada cuando el usuario saca el archivo sin soltarlo.
                Mejora la experiencia visual, mostrando que la zona ya no está activa.

            3️ drop → Se activa cuando el usuario suelta el archivo en la zona
            ✅ ¿Por qué es necesario?
                Evita que el navegador abra el archivo (comportamiento predeterminado en algunos navegadores).
                Maneja la carga y previsualización del archivo.
                Limpia el efecto visual después de soltar el archivo.



        */
    </script>
</body>

</html>
```