# Ricardo Alfonso Casanova Lozano.

Cristian Camilo Morales Tapias.

# Apartado 4. No cacheable

Usando manifest se ha configurado el servidor para que no sea cacheable. El archivo de texto manifest contiene los recursos necesarios para habilitar o deshabilitar la función de cache.

Desde todos los archivos .html se ha incluido el atributo manifest en la etiqueta html del documento. <html manifest=”cachemanifest”></html> para que accediera a la configuración del archivo de texto manifest y pudiera deshabilitar el almacenamiento en cache.

## Estructura del archivo manifest.

**CACHE MANIFEST**

La cadena “CACHE MANIFEST” es obligatoria y tiene que aparecer en la primera línea, después se divide en 3 secciones:

1. CACHE: Todos los archivos que se incluyan en esta sección serán almacenados en cache después de ser descargados por primera vez.
2. NETWORK: En el caso de la página web a configurar se utilizara esta sección dado que los archivos incluidos requieren conexión al servidor. En todas las solicitudes enviadas a estos recursos se omite la caché.
3. FALLBACK: Es una sección opcional que otorga la oportunidad de especificar paginas alternativas en caso de que no haya acceso a un recurso.

CACHE:

NETWORK:

Index.html

Config.html

Minombre.html

/css/iroom.css

FALLBACK:

# Apartado 6. Eventos

SSE (Server Sent Event). El cliente solicitara una página web al servidor en este caso iroom usando el protocolo HTTP, una vez recibida el cliente ejecuta JavaScript en la página creando una conexión con el servidor. Cada vez que haya generado un cambio en los sensores, el servidor se lo notificara al cliente. Los pasos a seguir para implementar esta función son:

1. Se creara un nuevo EventSource objeto con la URL del script que genera el evento para abrir conexión con el servidor:

“const evtSource = new EventSource("sse\_sensores.php");”

1. Una vez incorporado el “event source” ya se puede recibir mensajes por el servidor adjuntando un controlador para el evento mensaje. El siguiente código muestra un ejemplo de cómo escuchar mensajes del servidor:

evtSource.onmessage = function(event) {

const newElement = document.createElement("li");

const eventList = document.getElementById("list");

newElement.innerHTML = "message: " + event.data;

eventList.appendChild(newElement);

}

1. Desde el servidor se creara un script que envié eventos cada vez que perciba cambio en los sensores.