

GUIA DEL VIDEO DE RECUPERACIÓN: JSON

Nombre: Lander Sánchez

Para realizar un ejemplo de “Json” se desarrolló los siguientes códigos con la ayuda de la herramienta “Visual Studio Code”. A continuación, se explica cada parte detalladamente.



```
{
  "nombre": "Lander",
  "apellido": "Sanchez",
  "edad": 18,
  "precioAsiento": 100.25,
  "asiento": {
    "numero": 8,
    "letra": "A",
    "cargador": false
  },
  "historial": [
    {
      "salida": "California",
      "destino": "NY",
      "fecha": "22/08/2023"
    },
    {
      "salida": "Madrid",
      "destino": "Barcelona",
      "fecha": "12/06/2023"
    }
  ]
},
{
  "nombre": "Jair",
  "apellido": "Rodriguez",
  "edad": 21,
  "precioAsiento": 120.22,
  "asiento": {
    "numero": 5,
    "letra": "B",
    "cargador": true
  },
  "historial": []
}
]
```

Este código representa una lista de objetos en formato JSON, donde cada objeto describe información de un pasajero que ha viajado en diferentes vuelos. Cada objeto en la lista tiene varios campos que contienen información sobre el pasajero, sus nombres, apellidos, edades, precios de asientos, detalles de asientos (número, letra y si tienen cargador), y un historial de vuelos anteriores en forma de objetos JSON. Cada objeto representa un pasajero individual y su información asociada. Vamos a desglosar cada parte:

```
{ } Pasajeros.json X  <> index.html  JS main.js

{ } Pasajeros.json > ...
1  [{
2      "nombre": "Lander",
3      "apellido": "Sanchez",
4      "edad": 18,
5      "precioAsiento": 100.25,
6      "asiento": {
7          "numero": 8,
8          "letra": "A",
9          "cargador": false
10     },
11
12     "historial": [
13         {
14             "salida": "California",
15             "destino": "NY",
16             "fecha": "22/08/2023"
17         },
18         {
19             "salida": "Madrid",
20             "destino": "Barcelona",
21             "fecha": "12/06/2023"
22         }
23     ]
24 },
```

En este objeto, se describe un pasajero llamado "Lander Sanchez", de 18 años de edad. Se le asignó un asiento en la fila 8, letra A, y su asiento no tiene cargador. Además, tiene un historial de dos vuelos: uno desde California a Nueva York el 22 de agosto de 2023, y otro desde Madrid a Barcelona el 12 de junio de 2023.

2. Segundo objeto:

```
25  {
26    "nombre": "Jair",
27    "apellido": "Rodriguez",
28    "edad": 21,
29    "precioAsiento": 120.22,
30    "asiento": {
31      "numero": 5,
32      "letra": "B",
33      "cargador": true
34    },
35
36    "historial": []
37  }
38
```

En este objeto, se describe un pasajero llamado "Jair Rodriguez", de 21 años de edad. Se le asignó un asiento en la fila 5, letra B, y su asiento tiene un cargador. A diferencia del primer pasajero, este pasajero no tiene ningún registro en su historial de vuelos.

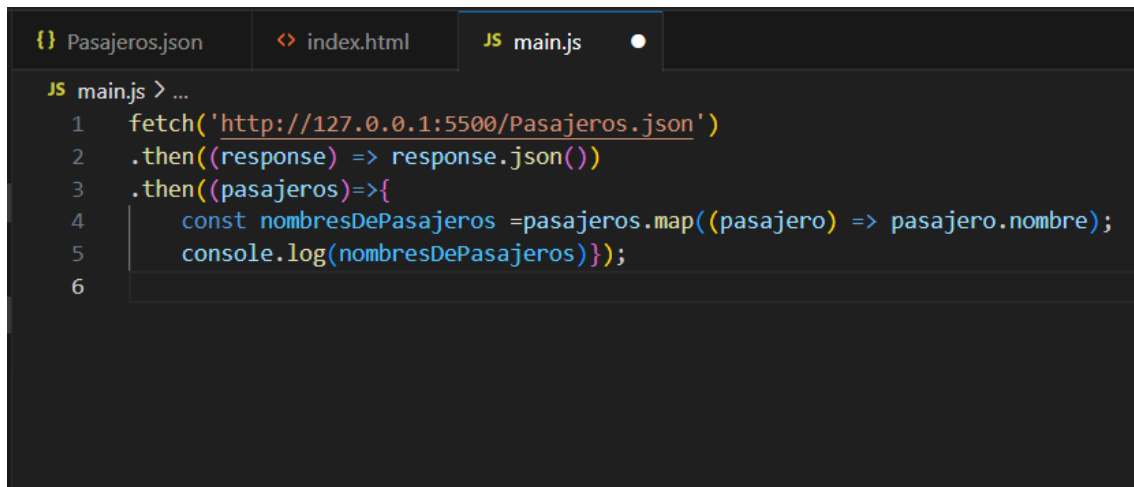
Index.html

```
{ } Pasajeros.json  <> index.html X  JS main.js

<> index.html > html
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <meta name="viewport" content="width=device-width, initial-scale=1.0">
6    <script src="./main.js" defer></script>
7    <title>PracticaJson</title>
8  </head>
9  <body>
10
11 </body>
12 </html>
```

Creamos un nuevo archivo, pero ahora un "index.html" donde escribimos un "!" para que nos cree automáticamente la estructura básica html. Entonces simplemente cambiamos el título a "PracticaJson" y enlazamos a un archivo "Main.js" que vamos a crear posteriormente usando la etiqueta "<Script>" con "defer".

Main.js



```
JS main.js > ...
1  fetch('http://127.0.0.1:5500/Pasajeros.json')
2  .then((response) => response.json())
3  .then((pasajeros)=>{
4      const nombresDePasajeros =pasajeros.map((pasajero) => pasajero.nombre);
5      console.log(nombresDePasajeros)});
6
```

Creamos un archivo JavaScript “main.js”. Eel código realiza lo siguiente:

1. Utiliza la función “fetch” para hacer una solicitud a la URL “http://127.0.0.1:5500/Pasajeros.json”, que es la dirección de nuestro navegador y el archivo Json de “Pasajeros” que creamos.
2. El método “.then()” se encadena después de la llamada “fetch” para manejar la respuesta de la solicitud. En este caso, se convierte la respuesta en formato JSON usando “response.json()”.
3. Luego, se encadena otro “.then()” que maneja los datos JSON obtenidos. Aquí, se utiliza el método “map” para recorrer la lista de pasajeros y extraer sus nombres en un nuevo array llamado “nombresDePasajeros”.
4. Finalmente, los nombres de los pasajeros se imprimen en la consola del navegador utilizando “console.log(nombresDePasajeros)”.

Link del video: <https://www.youtube.com/watch?v=fcUv9PxHEql>