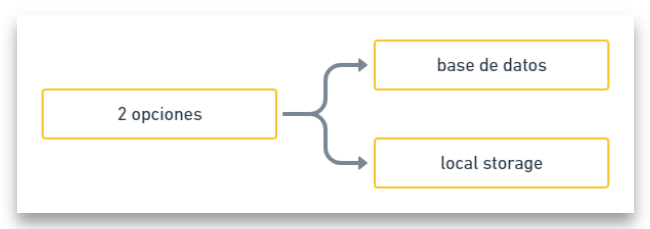
**LOCAL STORAGE**

**LocalStorage** es una herramienta compatible con html5 y los navegadores que lo soportan, incluso explorer desde su versión 8 hacia arriba. La gracia **de** ésta es la capacidad **para** almacenar datos **de** manera local en el navegador que utiliza el usuario mientras navega

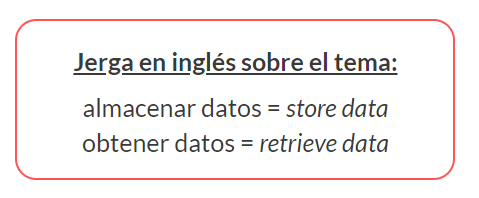
LocalStorage guarda cadena de texto y estas son guardadas con una clave => valor

Set => guarda información

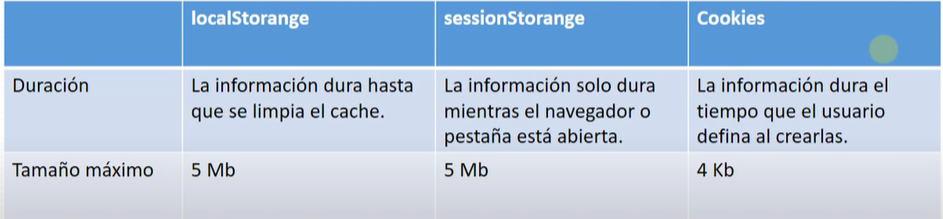
Get => Obtener información



Utilizando **local storage**, la información queda almacenada en el **navegador**del usuario. Esa información, al igual que en una base de datos, podremos rescatarla y manipularla como queramos.

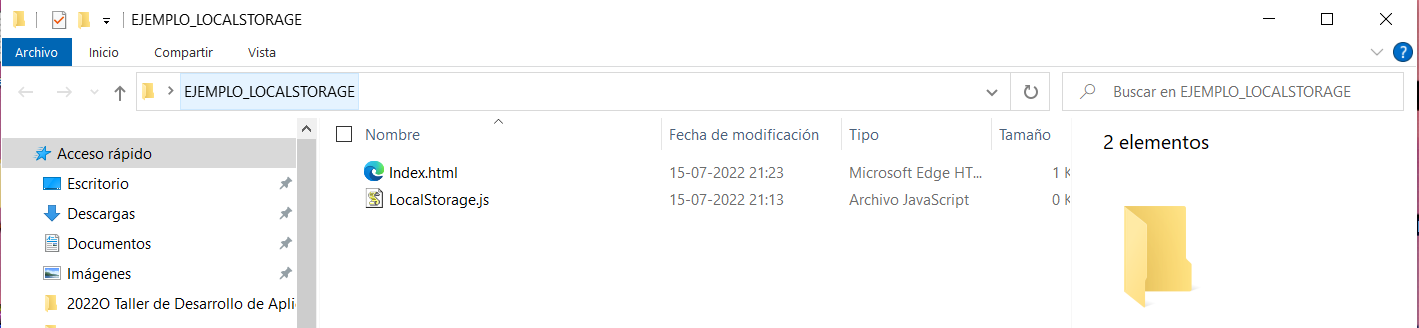


**Diferencias de almacenamiento:**



Ejemplo

Creamos una carpeta llamada **EJEMPLO\_LOCALSTORAGE**



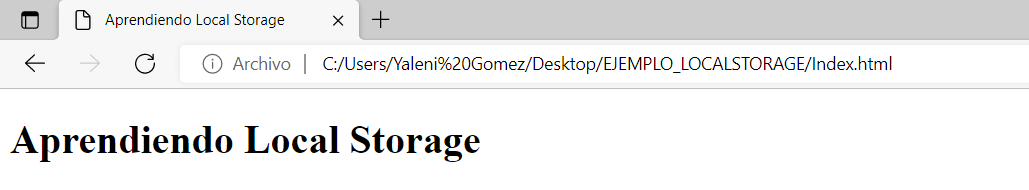
Creamos 2 archivos :

1.- Index.html

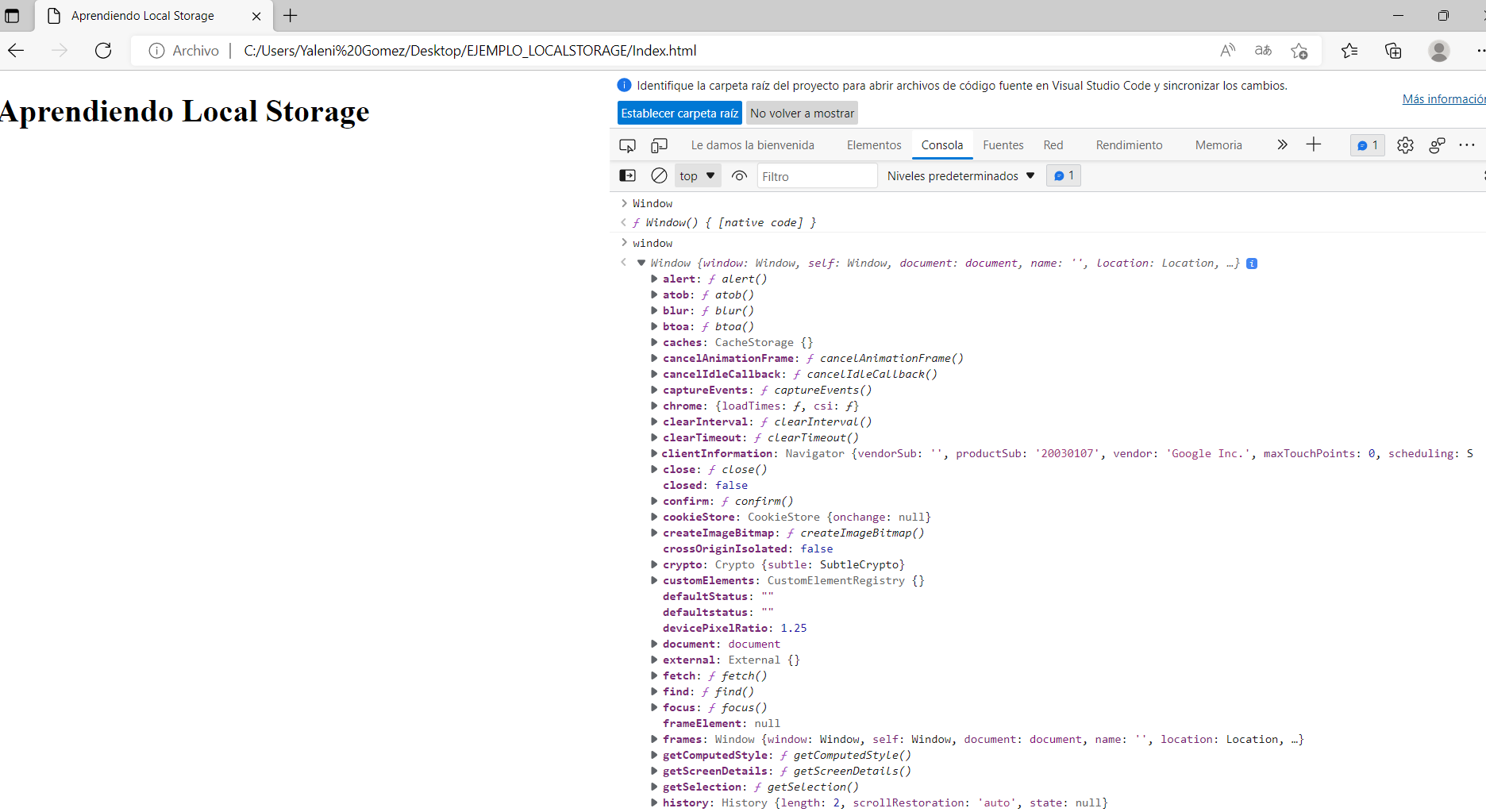
2.- LocalStorage.js

Nota: En el archivo Index.html referenciamos el archivo **LocalStorage.js.**

Lo ejecutamos y este levanta el browser Edge (puede ser otro navegador es opcional)



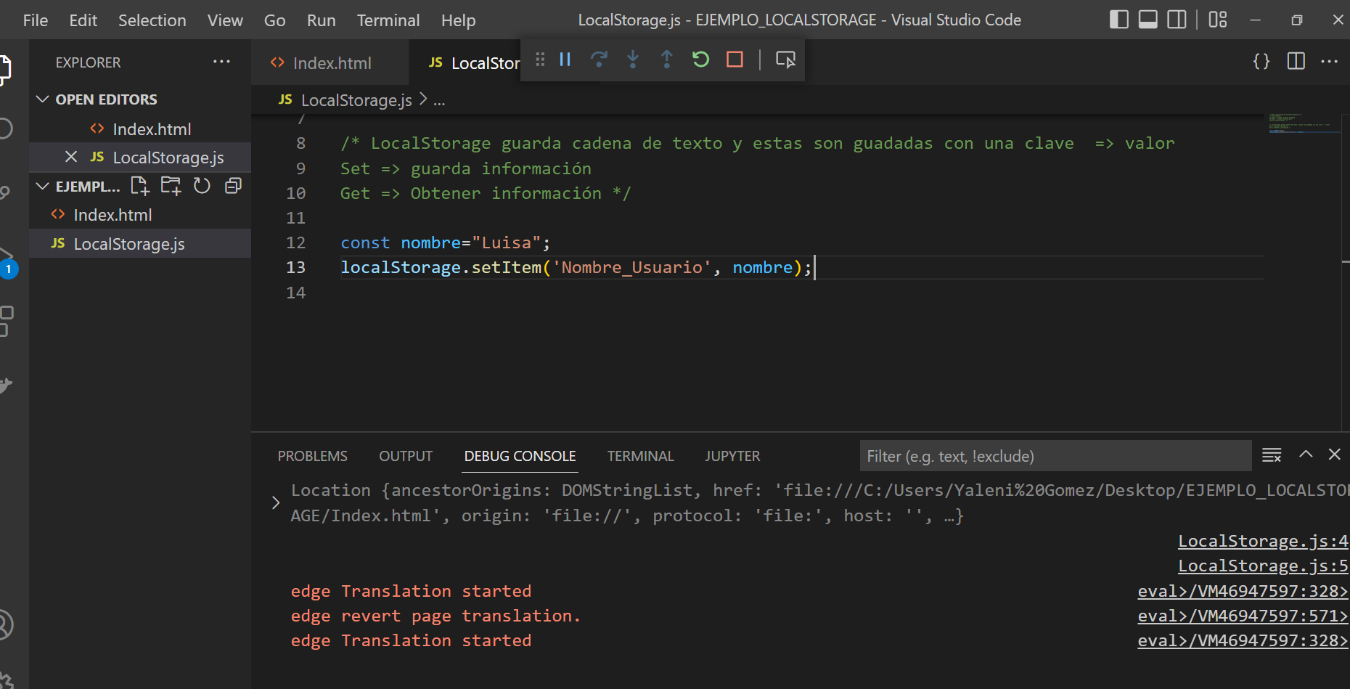
En el browser damos click derecho e inspeccionamos la página.



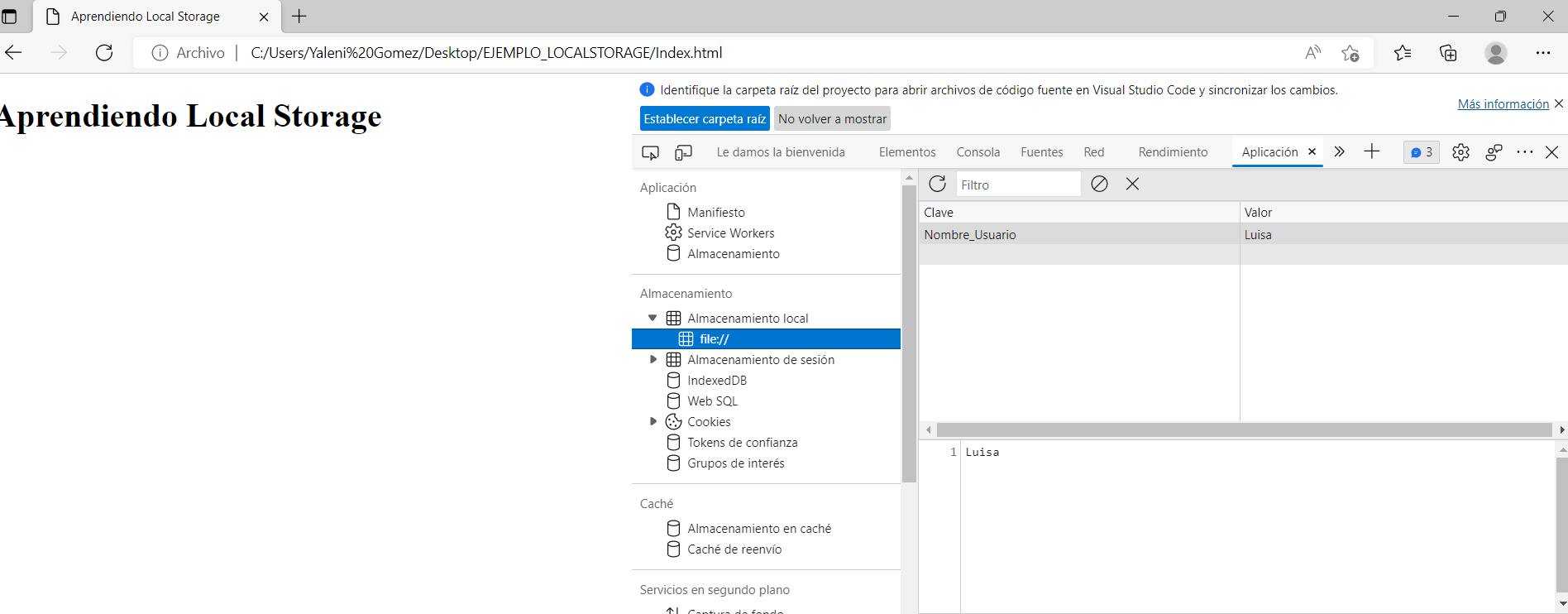
Aquí se despliega una lista de objetos, pero la que nos enfocaremos es en el LocalStorage.



Se escribe esta línea de código



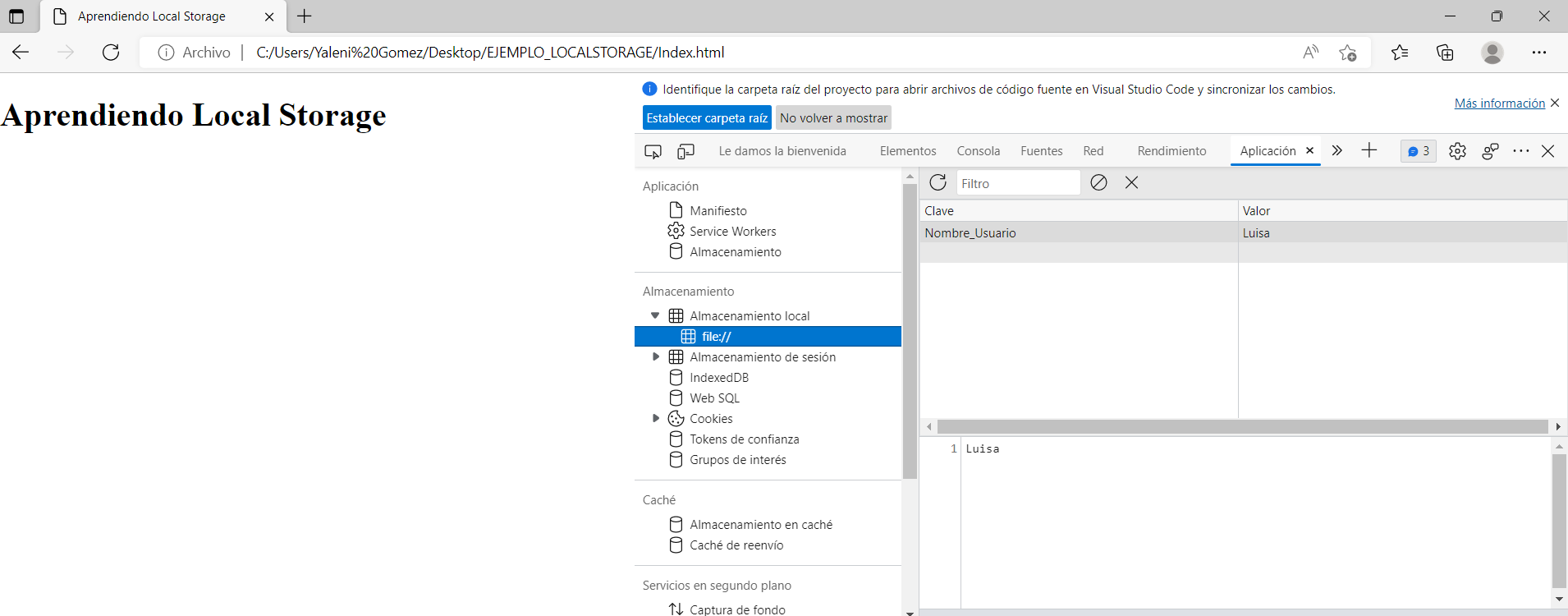
El resultado:



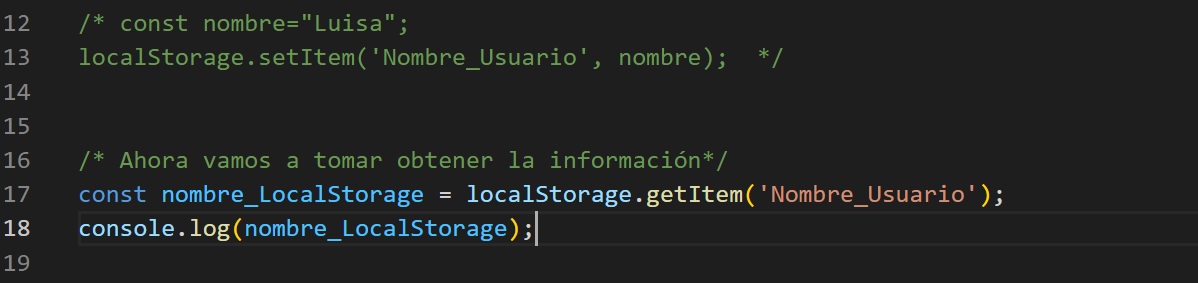
La diferencia entre el LocalStorage (almacenamiento local) a Sesión Storage (almacenamiento de sesión).

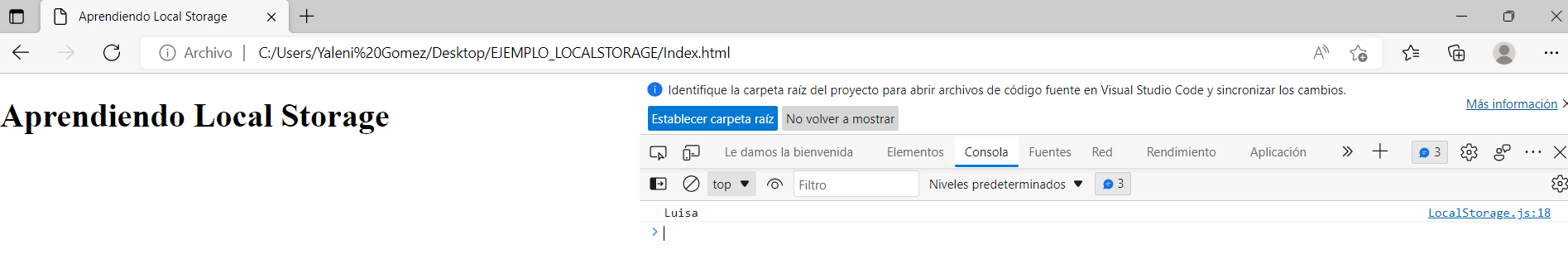
Es que cuando se cierra el navegado en el LocalStorage no se pierde la información a menos que limpie la caché.

IndexDB es una base de datos más compleja porque es más avanzado por lo que no lo abordaremos en esta clase.

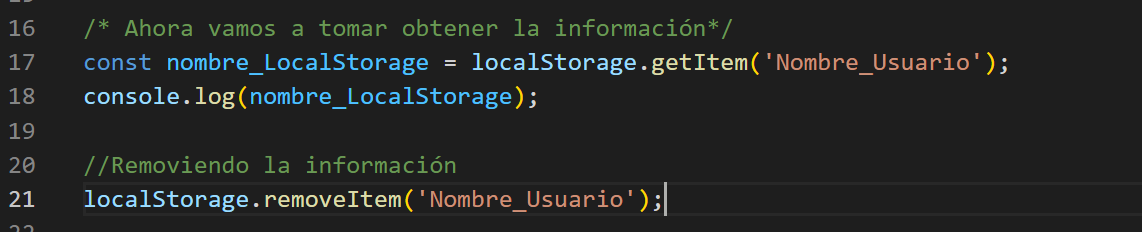


Ahora tomaremos la información:

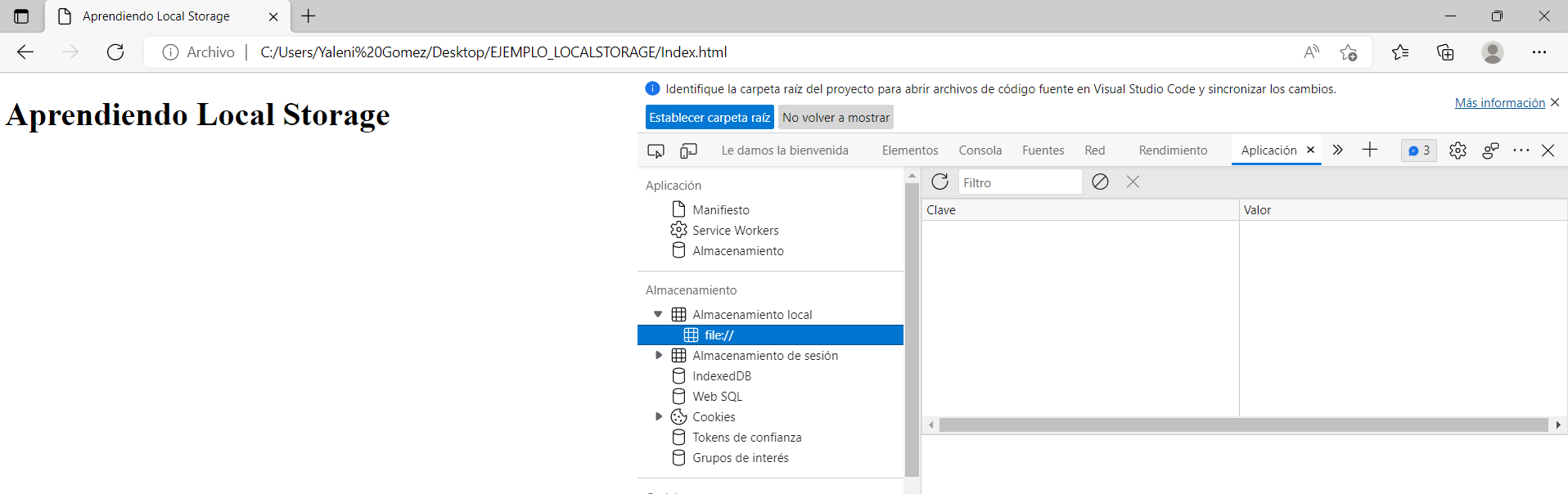




Si queremos remover la información:



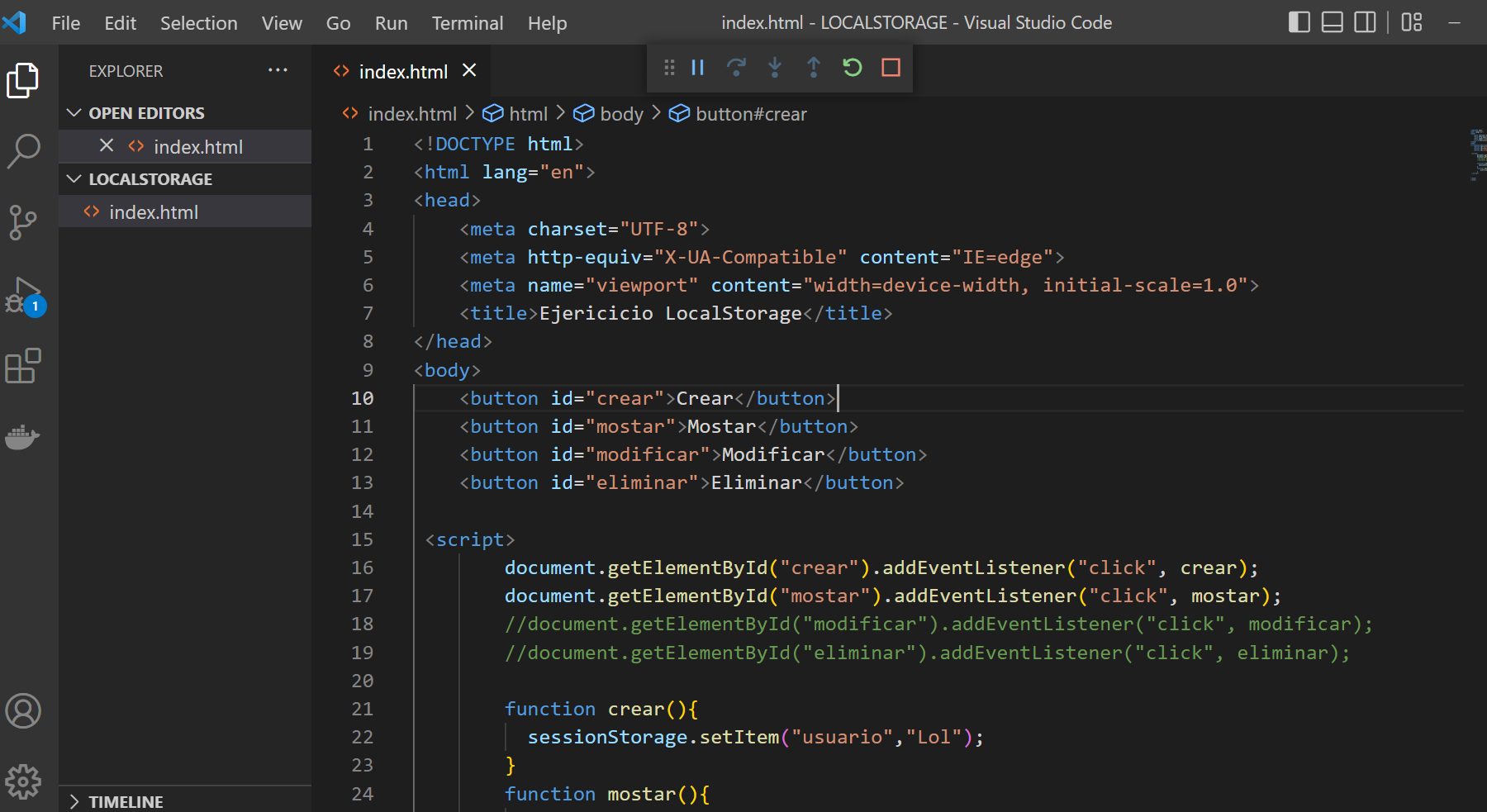
En el browser vamos a la viñeta “Aplicación”, ahora ya no se encuentra la información.



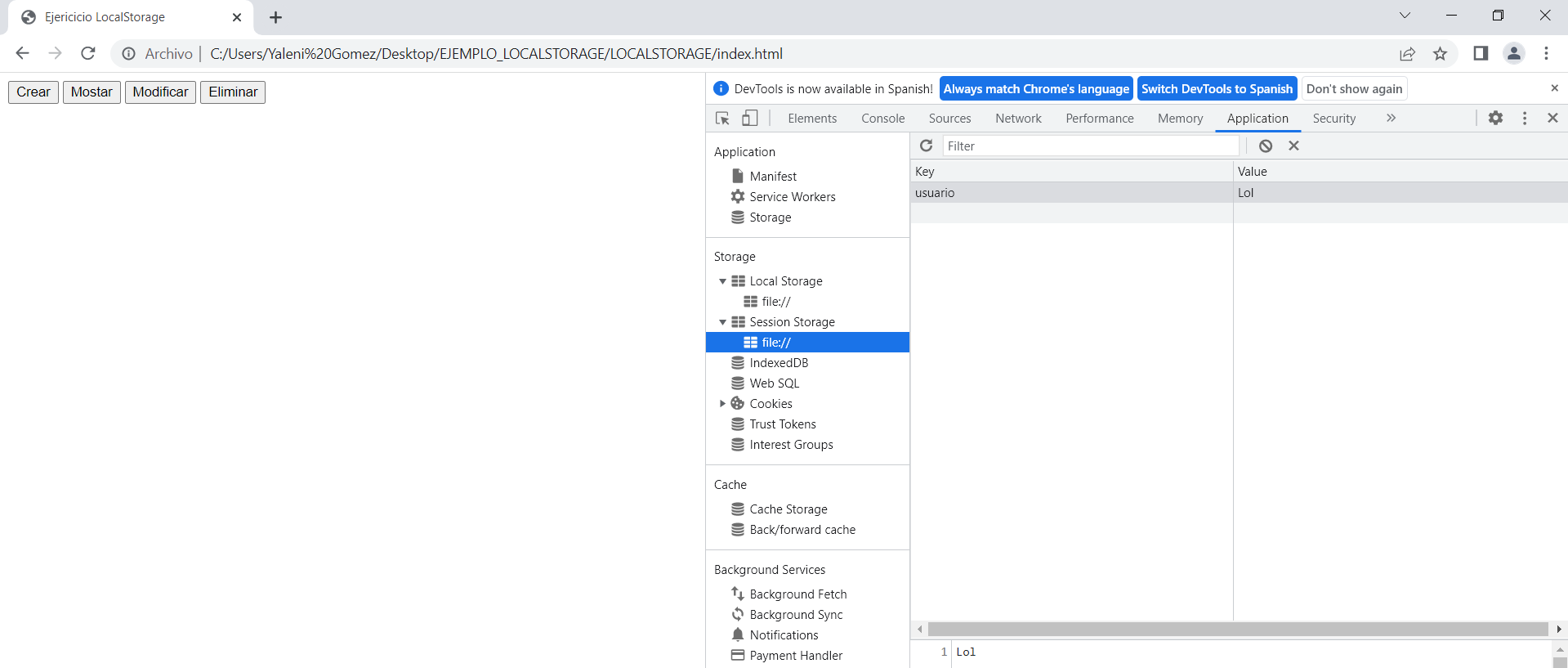
**Ahora vamos a realizar un ejercicio.**

1.- Crearemos una carpeta

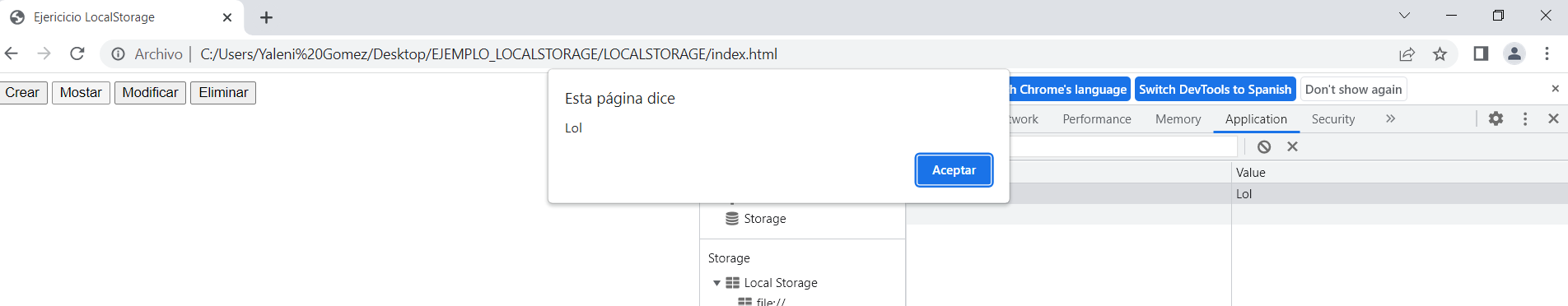
2.- Dentro de la carpeta crearemos un archivo para este ejemplo **”index.html”**



**Resultado de crear:**



**Resultado de mostrar:**



En el caso de querer mostrar la información por consola solo basta con utilizar “console.log”

  function mostar(){

            //alert(localStorage.getItem("usuario"));

            //alert(localStorage.getItem("carrera"));

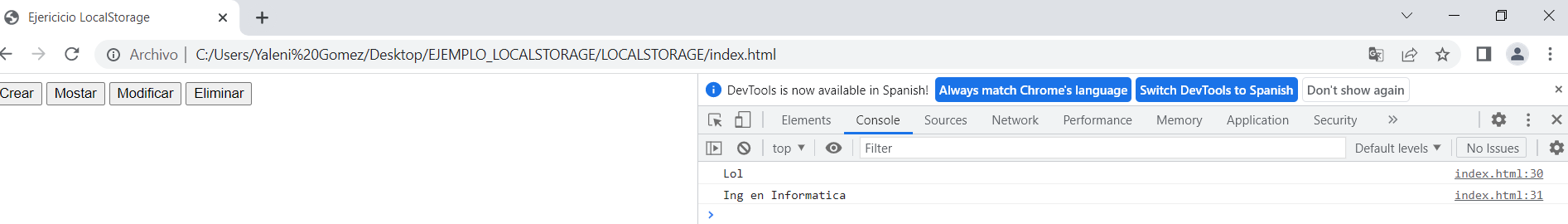
            const user= localStorage.getItem("usuario");

            const carre=localStorage.getItem("carrera");

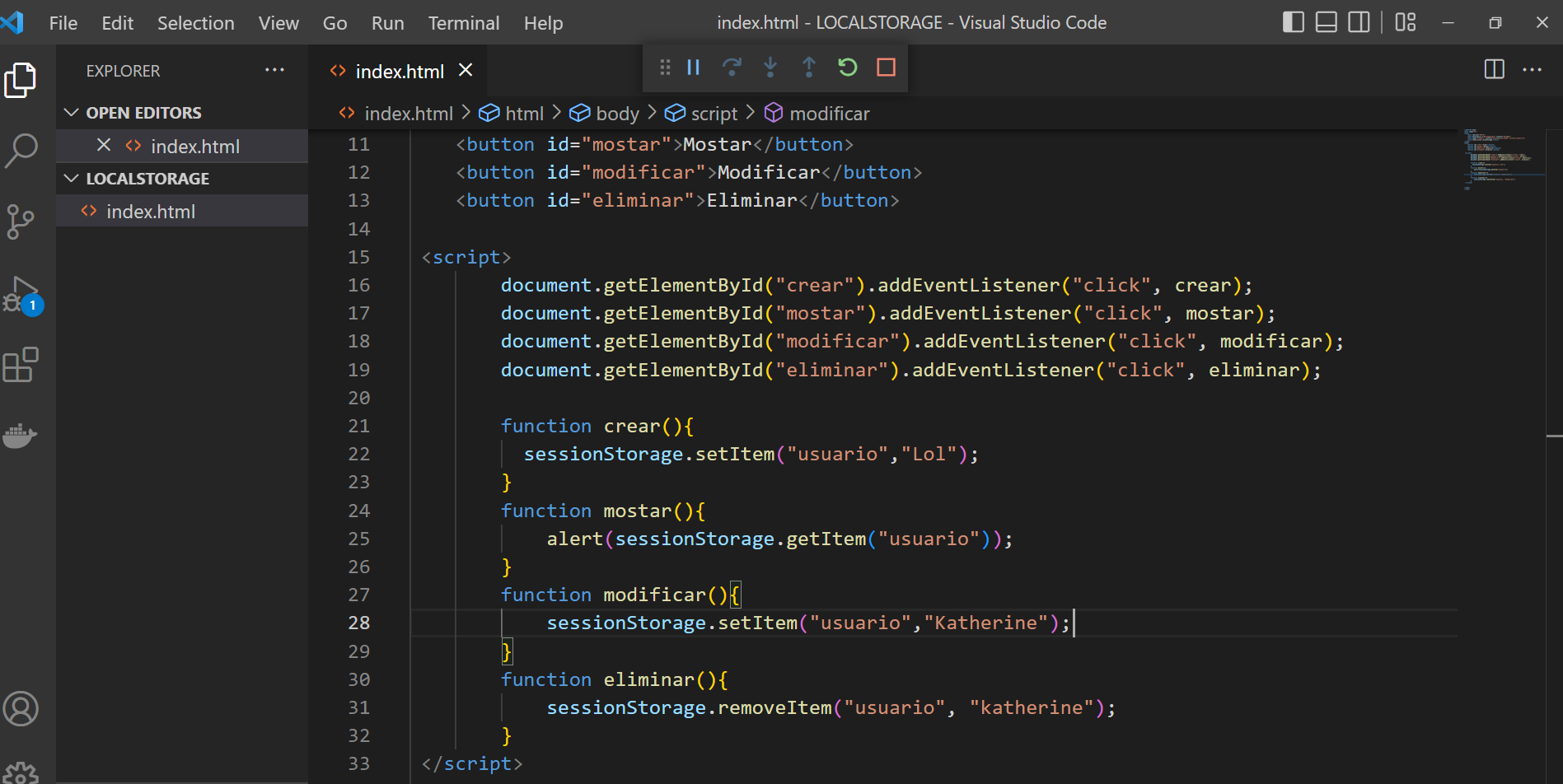
            console.log(user);

            console.log(carre);

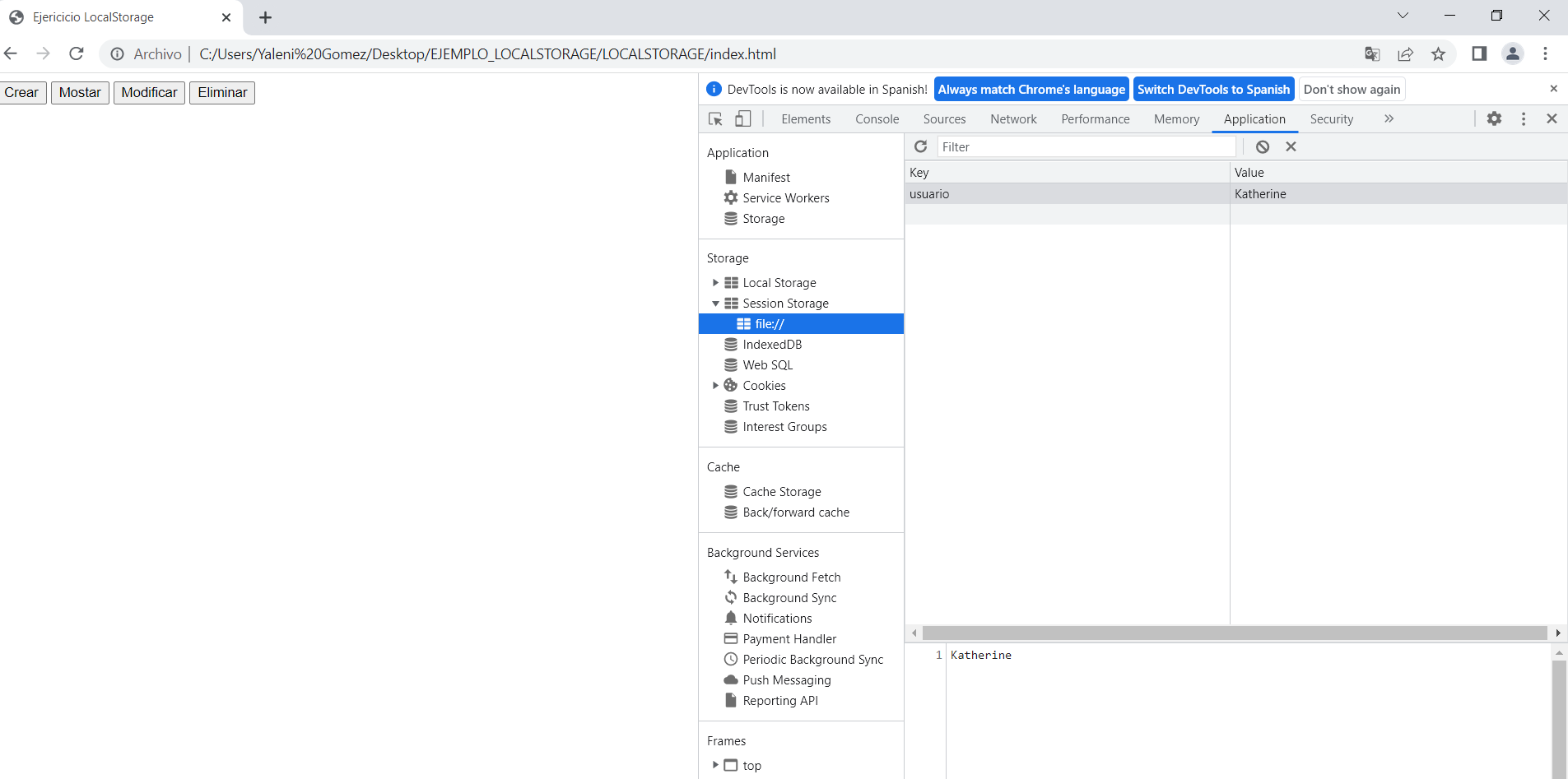
        }

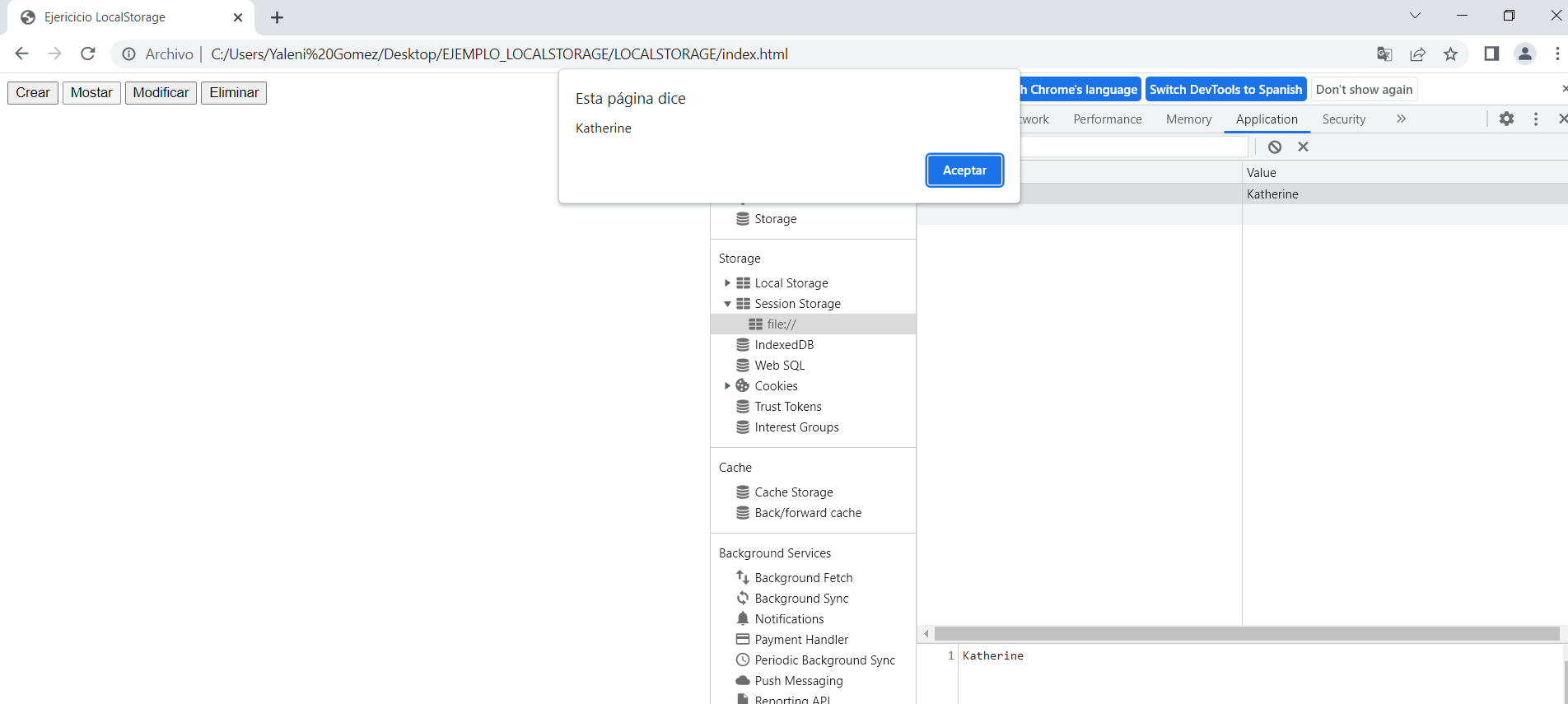


Ahora agregamos los botones **“modificar y eliminar”**

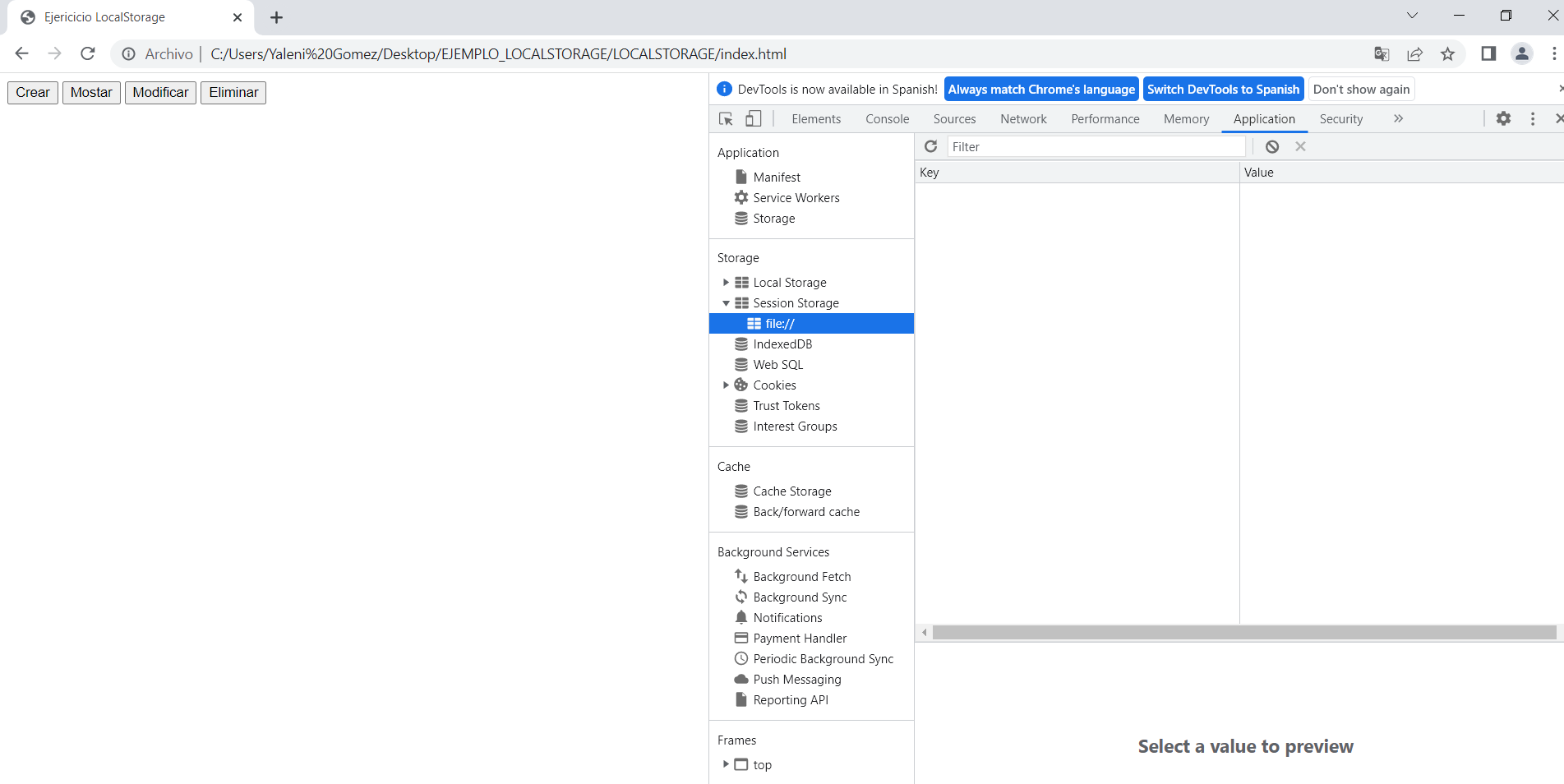


**Resultado de modificar y mostrando la información modificada:**

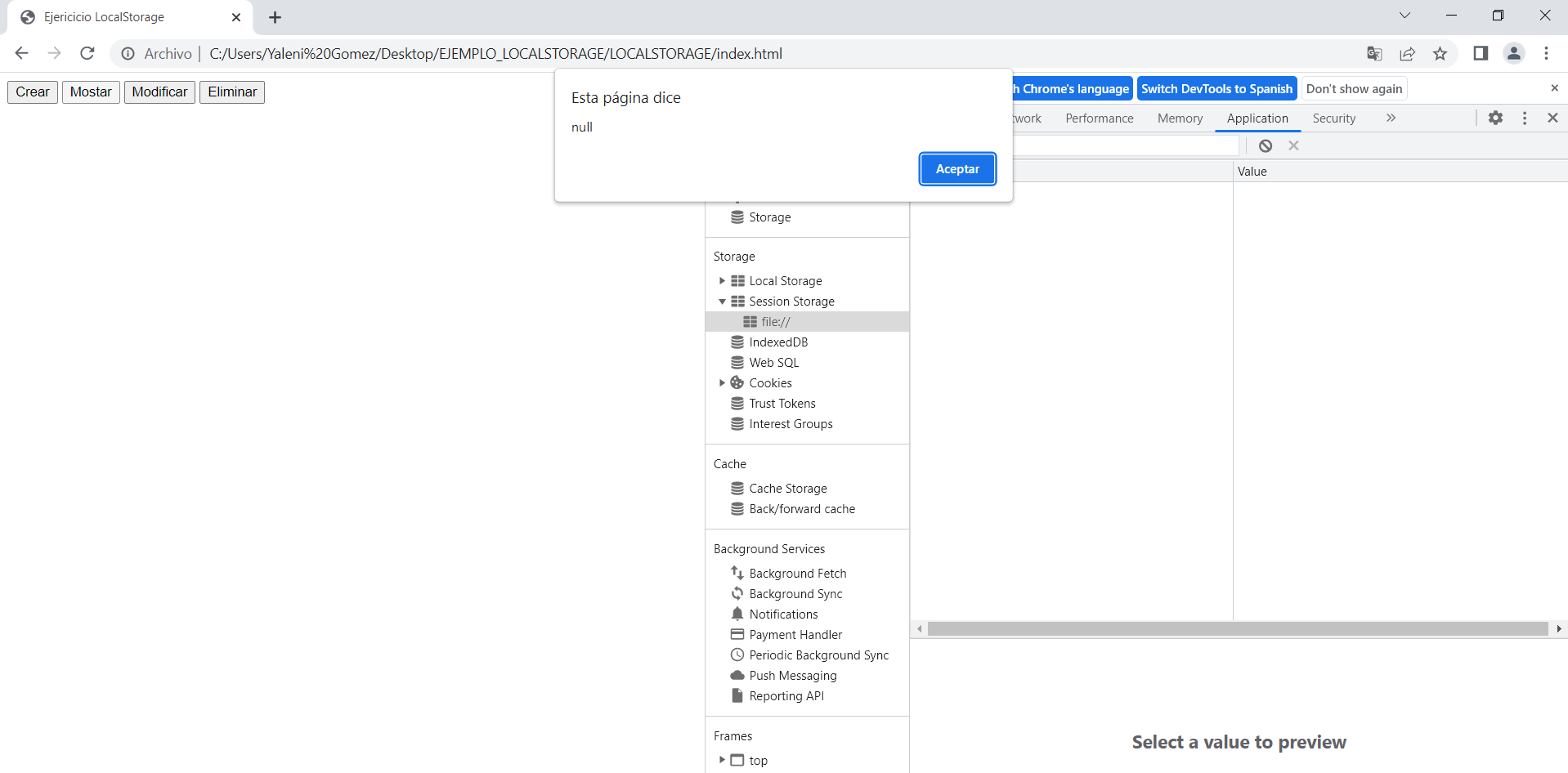




**Resultado de eliminar:**



**Mostrar la información:**



En resumen, si utilizamos sessionStorge la información estará disponible en el navegador que estamos utilizando ya que si abrimos otro navegador la información no estará disponible.

Ahora veremos el **LocalStorage.** A la siguiente línea de código solo modificaremosdonde dice **sessionStorage por**

**Antes**

function crear(){

          sessionStorage.setItem("usuario","Lol");

          sessionStorage.setItem("carrera","Ing en Informatica");

        }

**Despues**

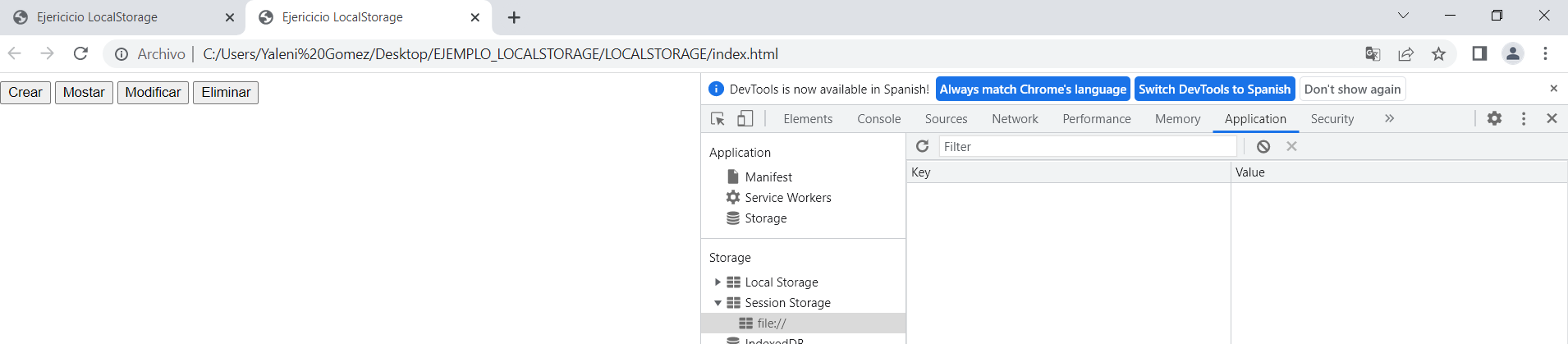
function crear(){

          localStorage.setItem("usuario","Lol");

          localStorage.setItem("carrera","Ing en Informatica");

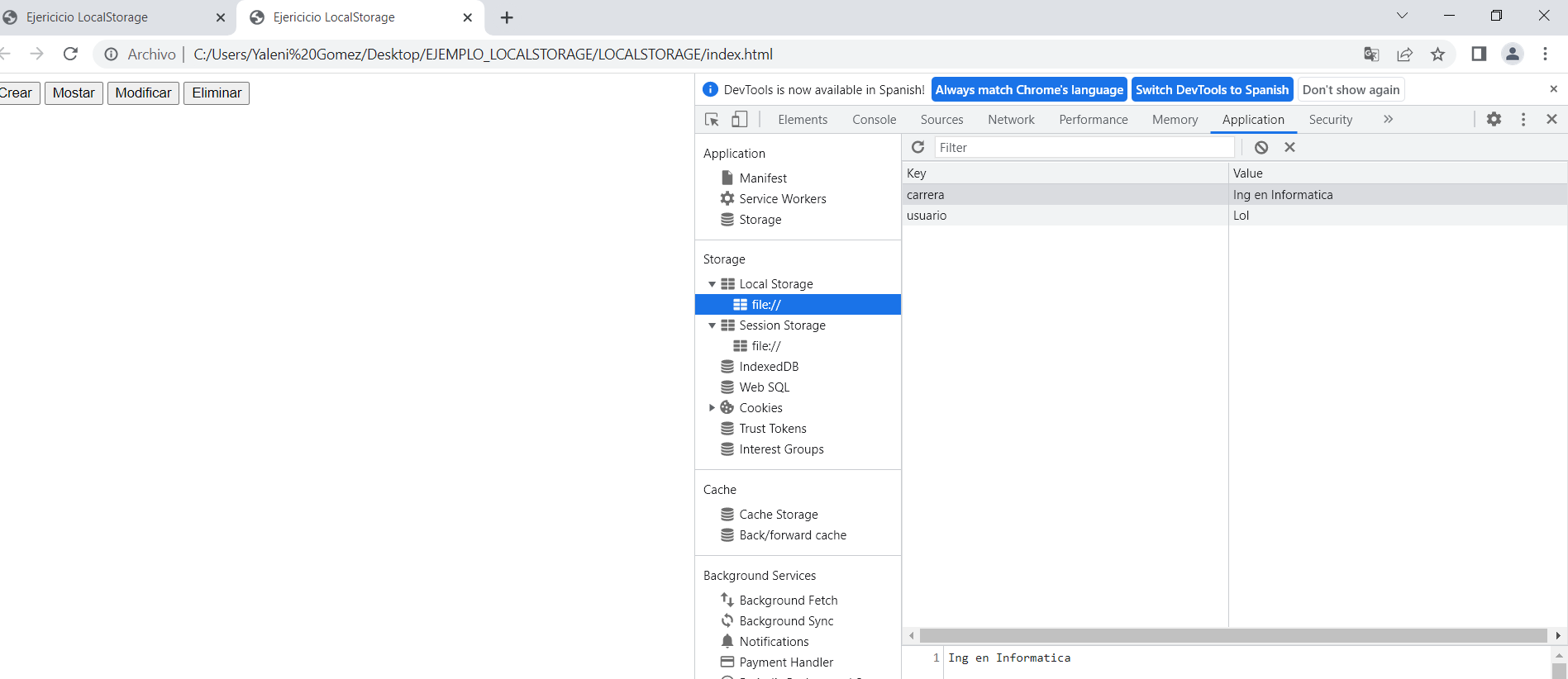
        }

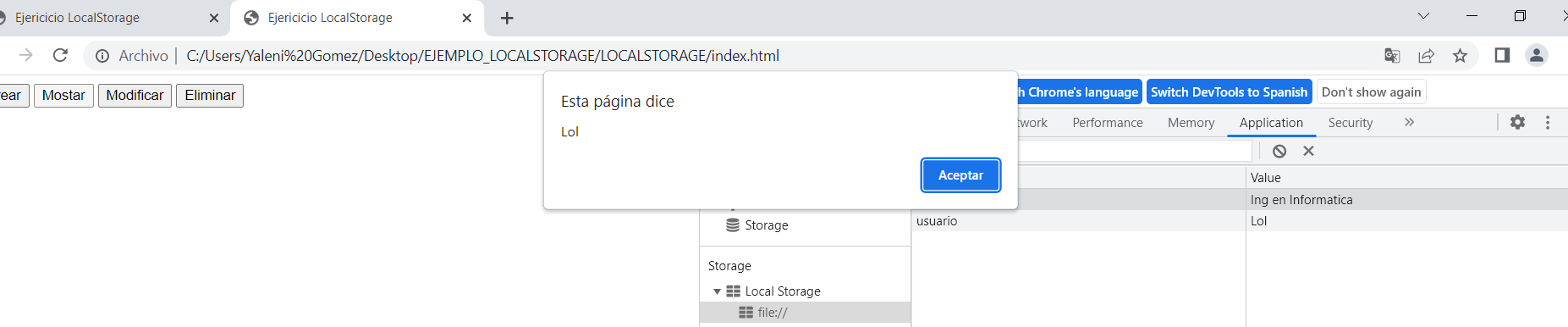
En la ubicación de sessionStorage ya no se ve reflejado la información

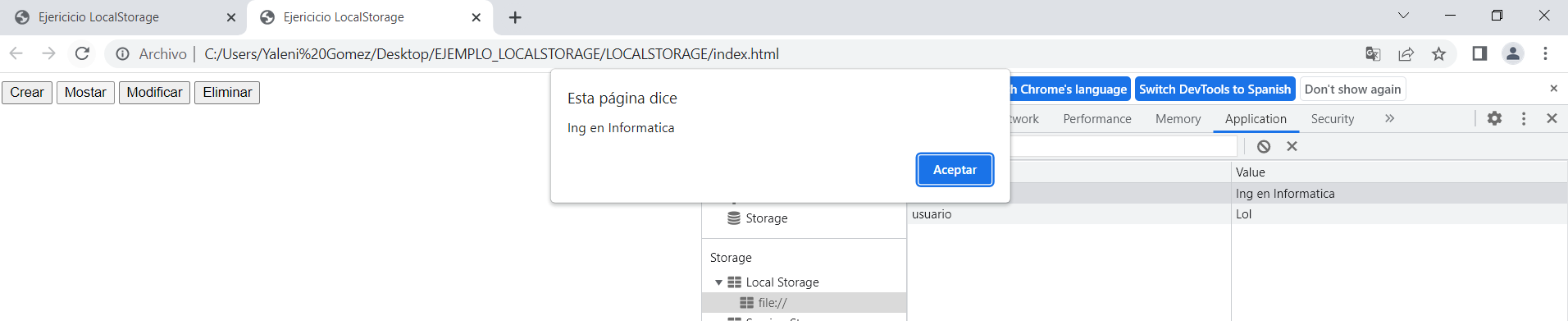


Pero si nos vamos a la ubicación localStorage el resultado es el siguiente:

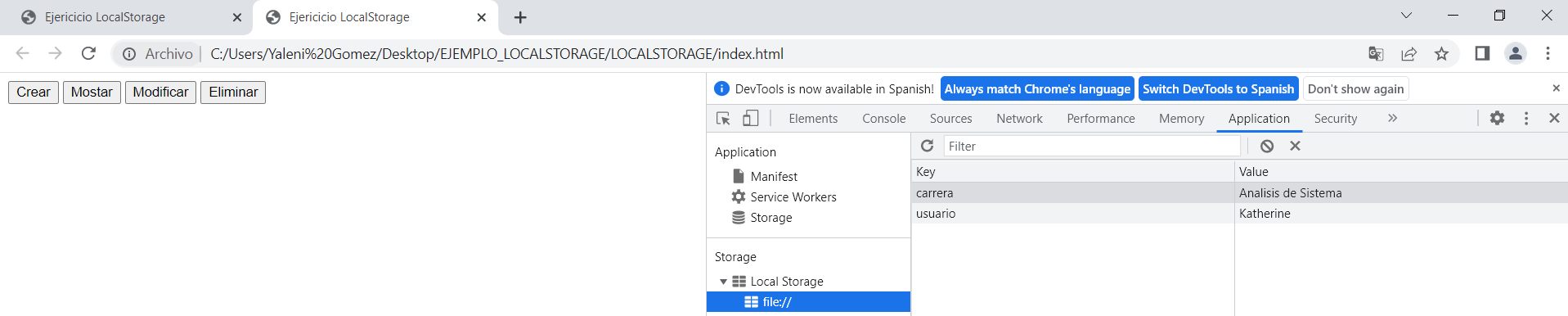
Creamos y mostramos información



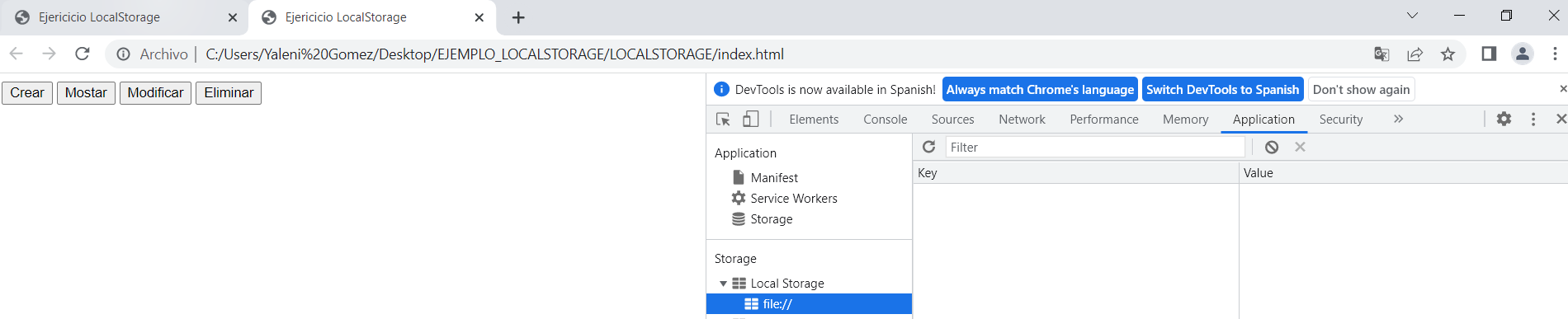


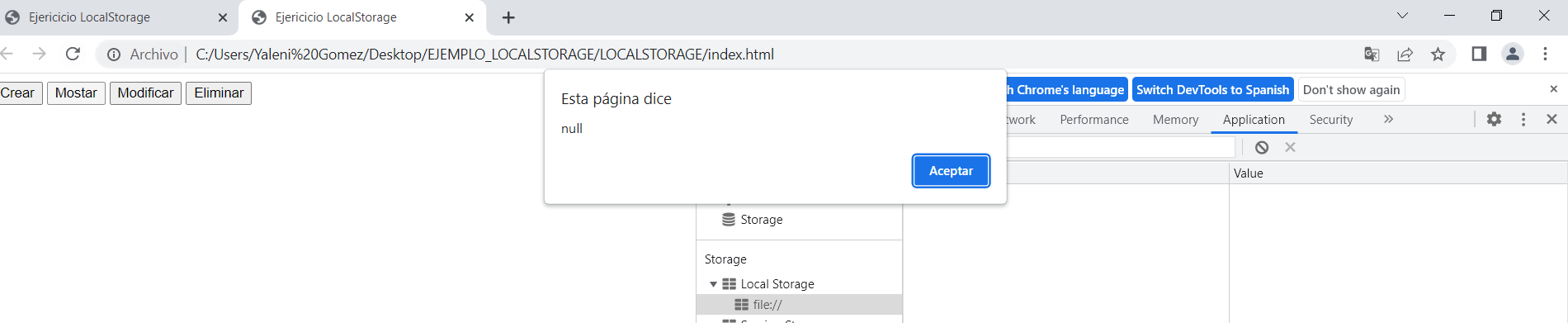


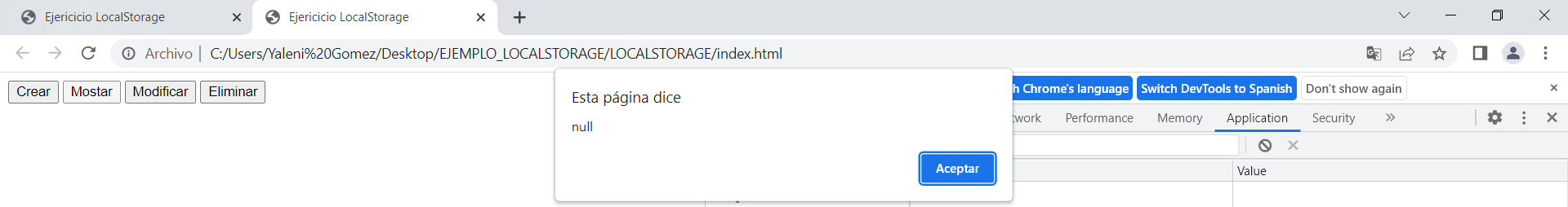
Ahora modificamos y eliminamos información:



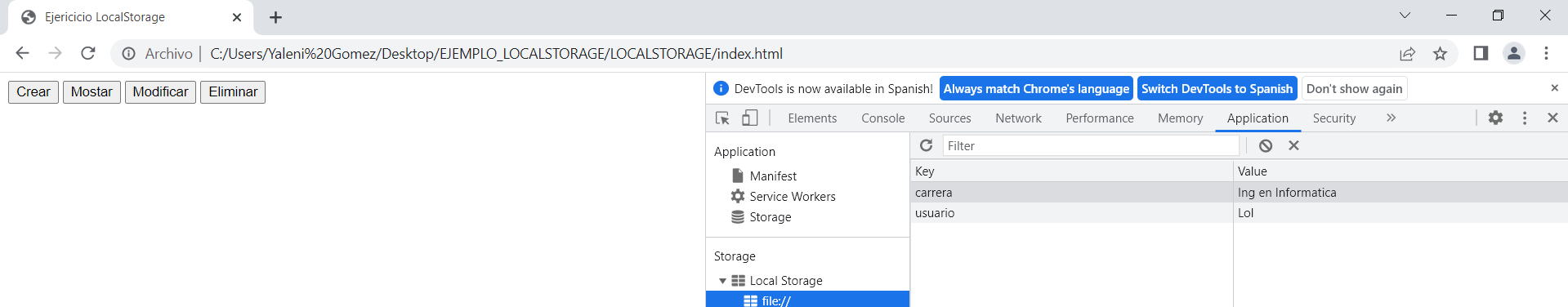
Ahora eliminamos información:

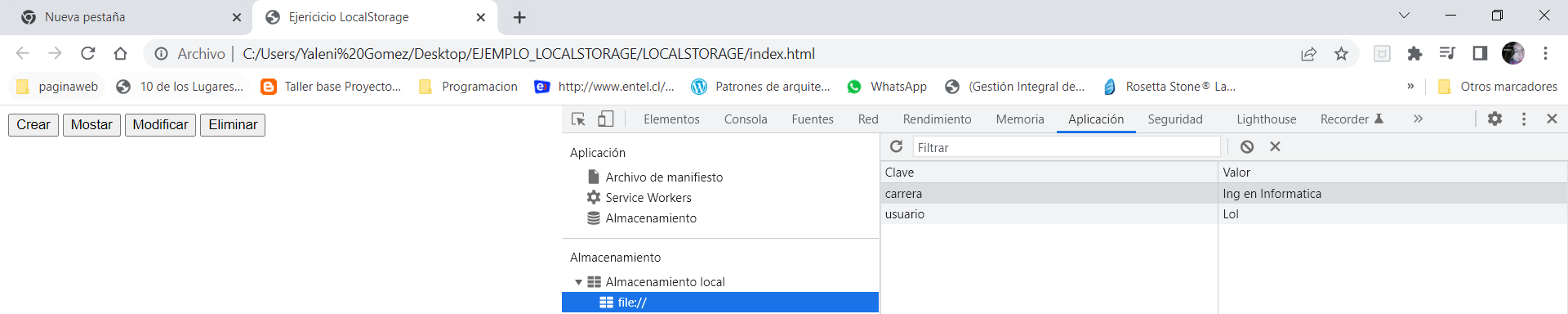






En resumen, si utilizamos **localStorage** la información está disponible en el navegador que estamos utilizando y si abrimos otro navegador la información sigue disponible.





Ejercicio1:

Se crea carpeta de trabajo

Crear 2 archivos

.- Index.js

.-app.js

**Escribiendo el código:**

**Archivo Index.js**

<!doctype html>

<html lang="en">

  <head>

    <meta charset="utf-8">

    <meta name="viewport" content="width=device-width, initial-scale=1">

    <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0-beta1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-0evHe/X+R7YkIZDRvuzKMRqM+OrBnVFBL6DOitfPri4tjfHxaWutUpFmBp4vmVor" crossorigin="anonymous">

  </head>

  <body>

    <div class="containes" mt-5>

            <h3>

                <span class="material-icons">

                    school

                </span>

                Cursos

            </h3>

            <form id="formulario">

                <input type="text" id="ramos" class="form-control my-3">

                <button type="submit" class="btn btn-primary">Guardar</button>

            </form>

            <div id="listaRamos" class="mt-3">

                <div class="alert alert-danger" role="alert">

                    <span class="material-icons mr-3" >

                        school

                    </span>

                    <span>Contenido de los ramos</span> -Estado

                    <span class="float-right"></span>

                        <span class="material-icons">

                           done

                        </span>

                        <span class="material-icons">

                            delete

                        </span>

                </div>

            </div>

    </div>

    <script src="app.js"></script>

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0-beta1/dist/js/bootstrap.bundle.min.js" integrity="sha384-pprn3073KE6tl6bjs2QrFaJGz5/SUsLqktiwsUTF55Jfv3qYSDhgCecCxMW52nD2" crossorigin="anonymous"></script>

    <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.5/dist/umd/popper.min.js" integrity="sha384-Xe+8cL9oJa6tN/veChSP7q+mnSPaj5Bcu9mPX5F5xIGE0DVittaqT5lorf0EI7Vk" crossorigin="anonymous"></script>

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0-beta1/dist/js/bootstrap.min.js" integrity="sha384-kjU+l4N0Yf4ZOJErLsIcvOU2qSb74wXpOhqTvwVx3OElZRweTnQ6d31fXEoRD1Jy" crossorigin="anonymous"></script>

</body>

</html>

**Archivo app.js**

//console.log('funcionando Ok.');

//const formularioUs=document.getElementById('formulario')

const formularioUs = document.querySelector('#formulario'); //esta opcion no solo puede seleccionar aid, tambien puede seleccionar clases.

const listaRamosUs = document.getElementById('listaRamos');

let arryClases =[]; //parte vacio el arreglo

//Creando una funcion:

const Creaitem =(curso)=>{

    let item={

        curso:curso,

        Estado:false

    }

    arryClases.push(item); //agrego el contenido que tiene el objeto item

    return item;

}

const SaveDB= () =>{

    localStorage.setItem('rutina', JSON.stringify(arryClases));

}

/\* let ramo1 = Creaitem('Programacion Front End');

let ramo2 = Creaitem('Base de Datos');

console.log(ramo1);

console.log(arryClases);  \*/  //Estoy imprimiendo por consola el contenido del arreglo

//EventListener

formularioUs.addEventListener('submit', (flecha) =>{

    flecha.preventDefault(); //previene de cualquier evento

    let ramosUs= document.querySelector('#ramos').value; //capturamos el valorque

                                                         //viene del input

    //console.log(ramosUs);

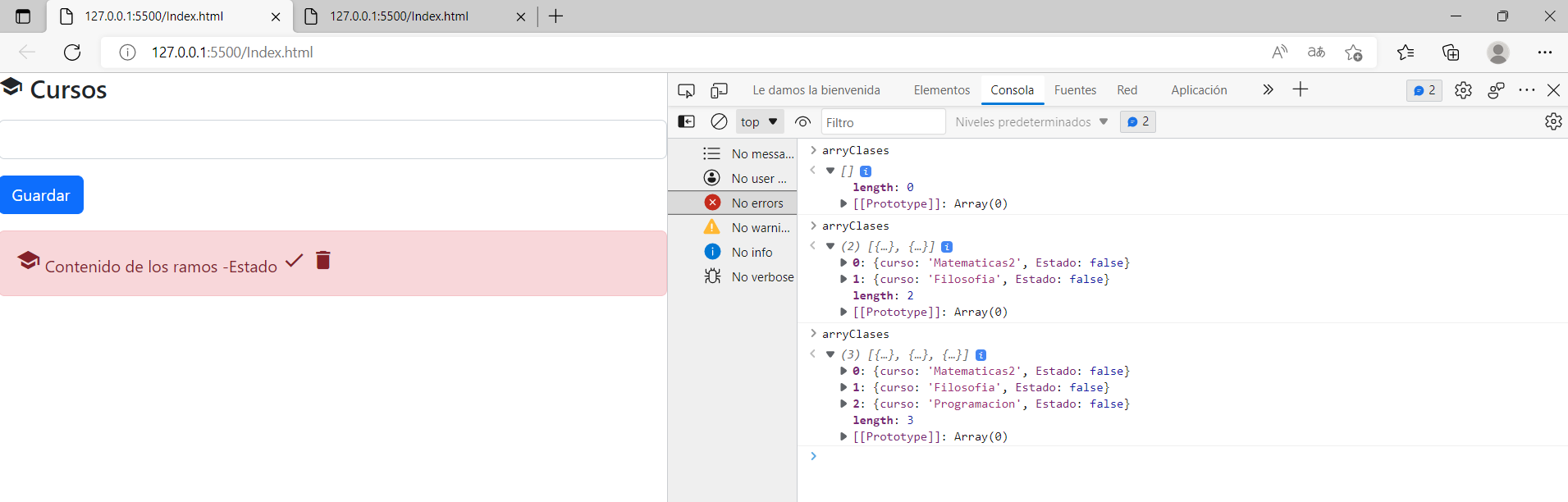
    Creaitem(ramosUs);

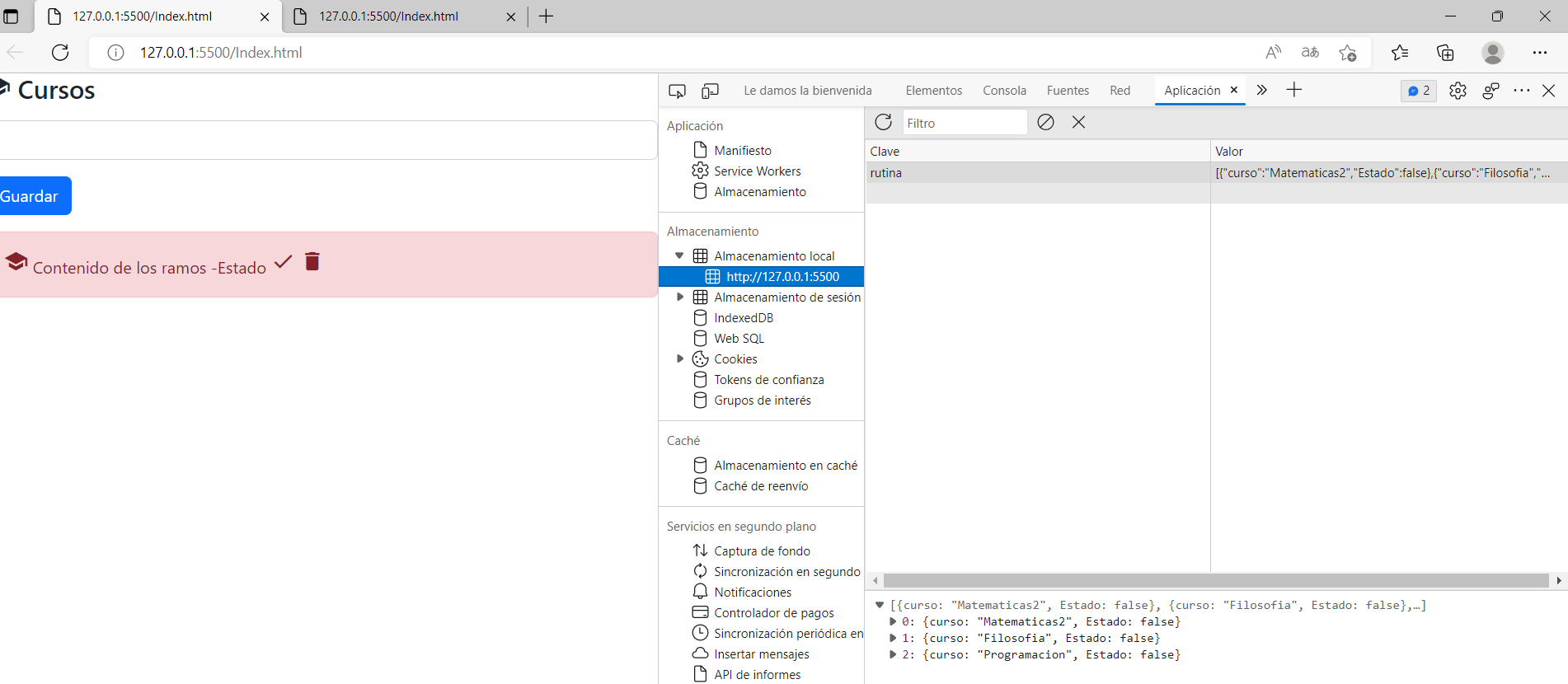
    SaveDB();

    formularioUs.reset();

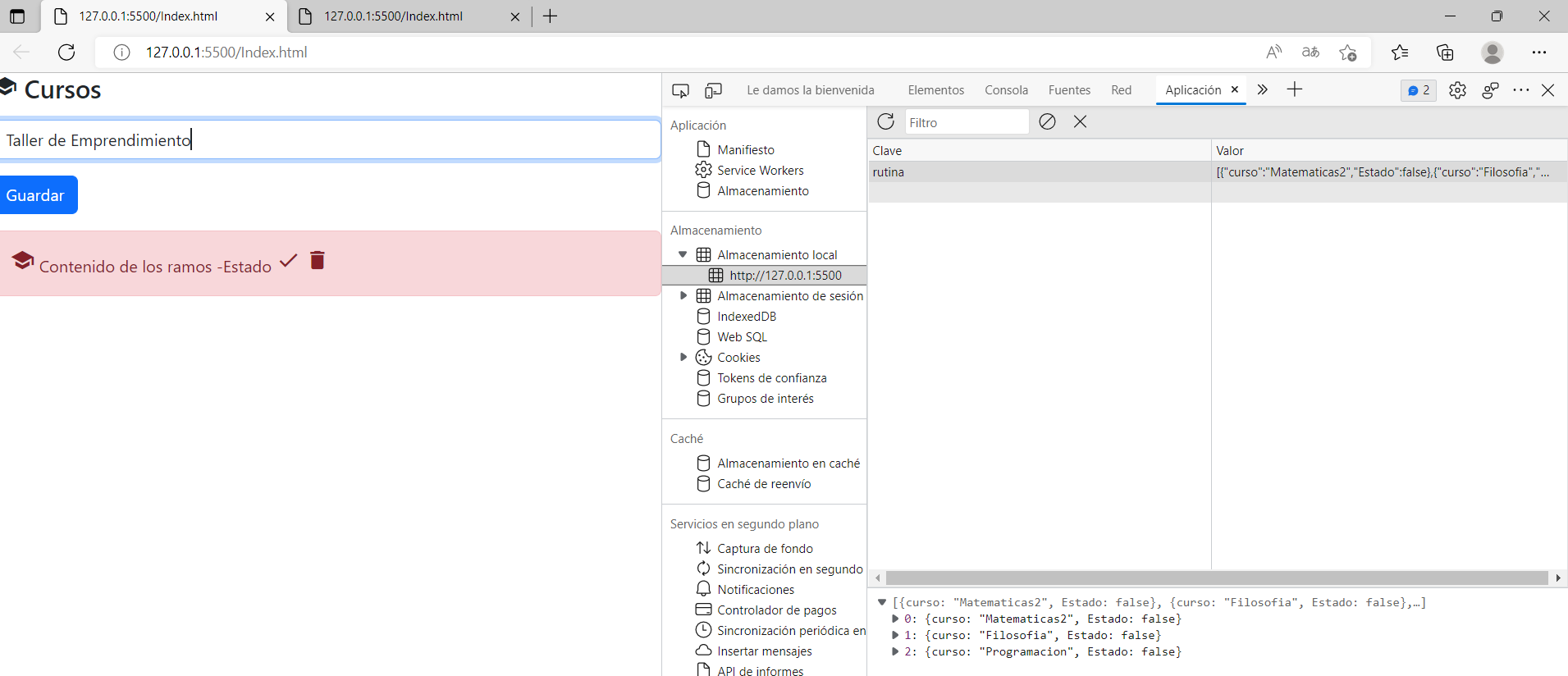
})

Resultado en browser:



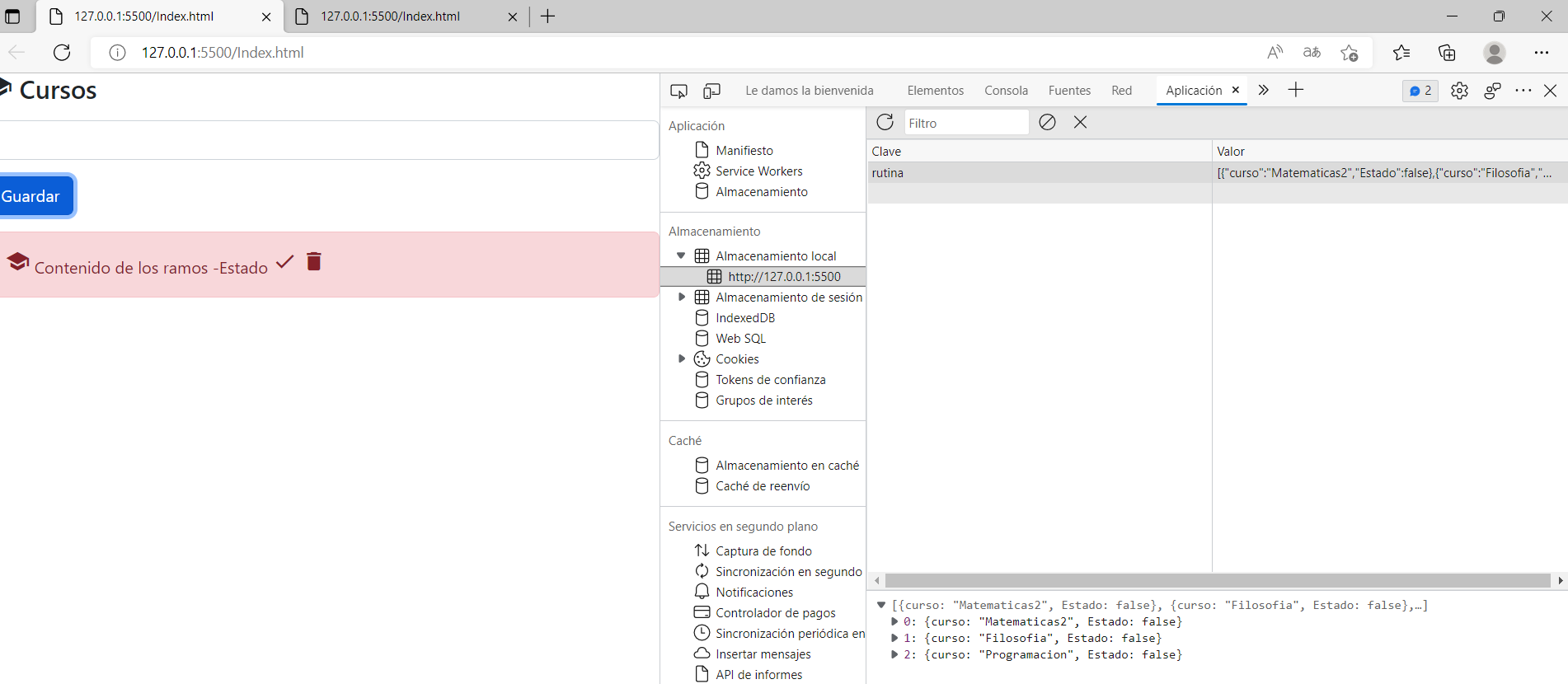


Ejemplo de agregar un registro:

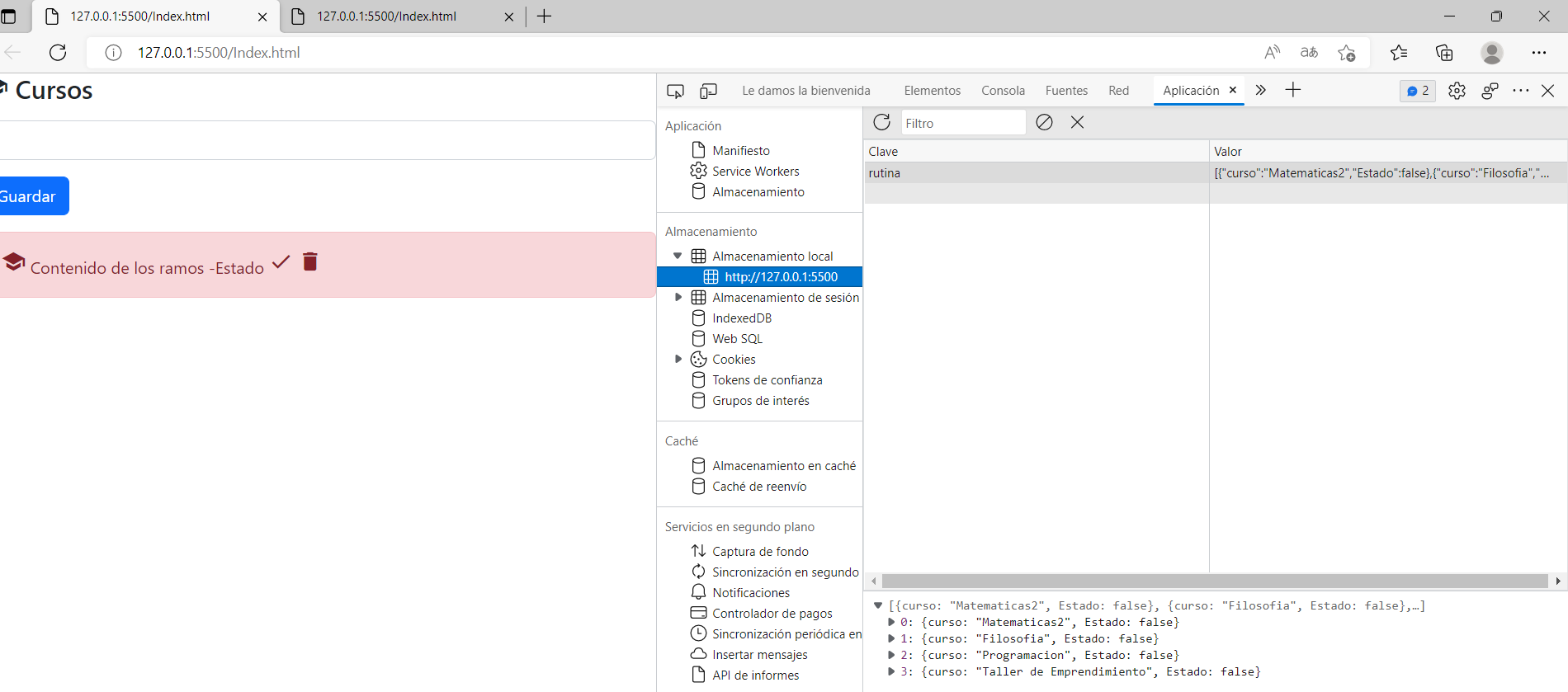


**Presionamos el boton Guardar:**

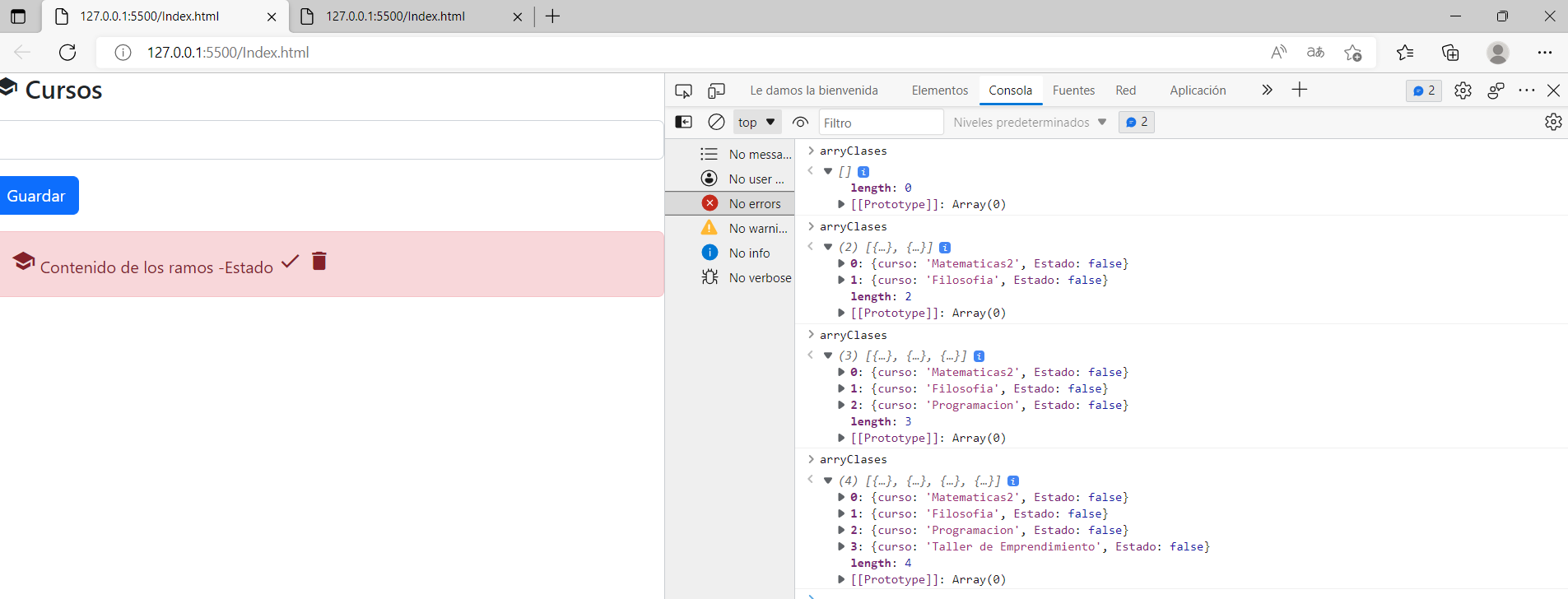
**Al guardar se limpia el formulario.**



Revisando si se agregó el registro:



Imprimiendo por consola:



Ahora mostraremos por pantalla la información que se encuentra en el localStorage:

//Funcion mostar la información que está en localStorage

const ready = ()=>{

    listaRamosUS.innerHTML ='';

    arryClases = JSON.parse(localStorage.getItem('rutina'));

    console.log('arryClases');

    if(arryClases === null){

        arryClases = [];

    }else{

        arryClases.forEach(element => {

            listaRamosUS.innerHTML +=`<div class="alert alert-danger" role="alert">

            <span class="material-icons mr-3" >

                school

            </span>

            <span>${element.curso}</span> -${element.Estado}

            <span class="float-right"></span>

                <span class="material-icons">

                   done

                </span>

                <span class="material-icons">

                    delete

                </span>

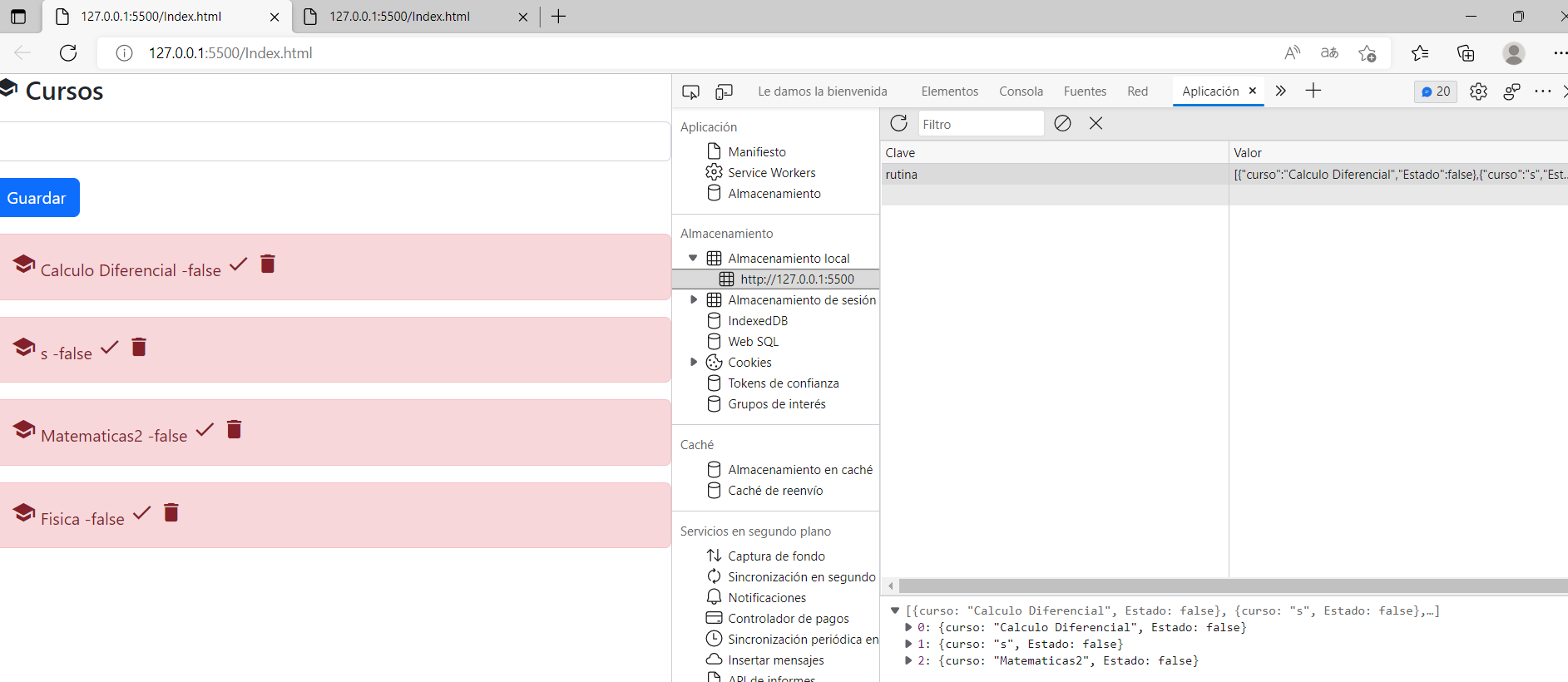
        </div>`

        });

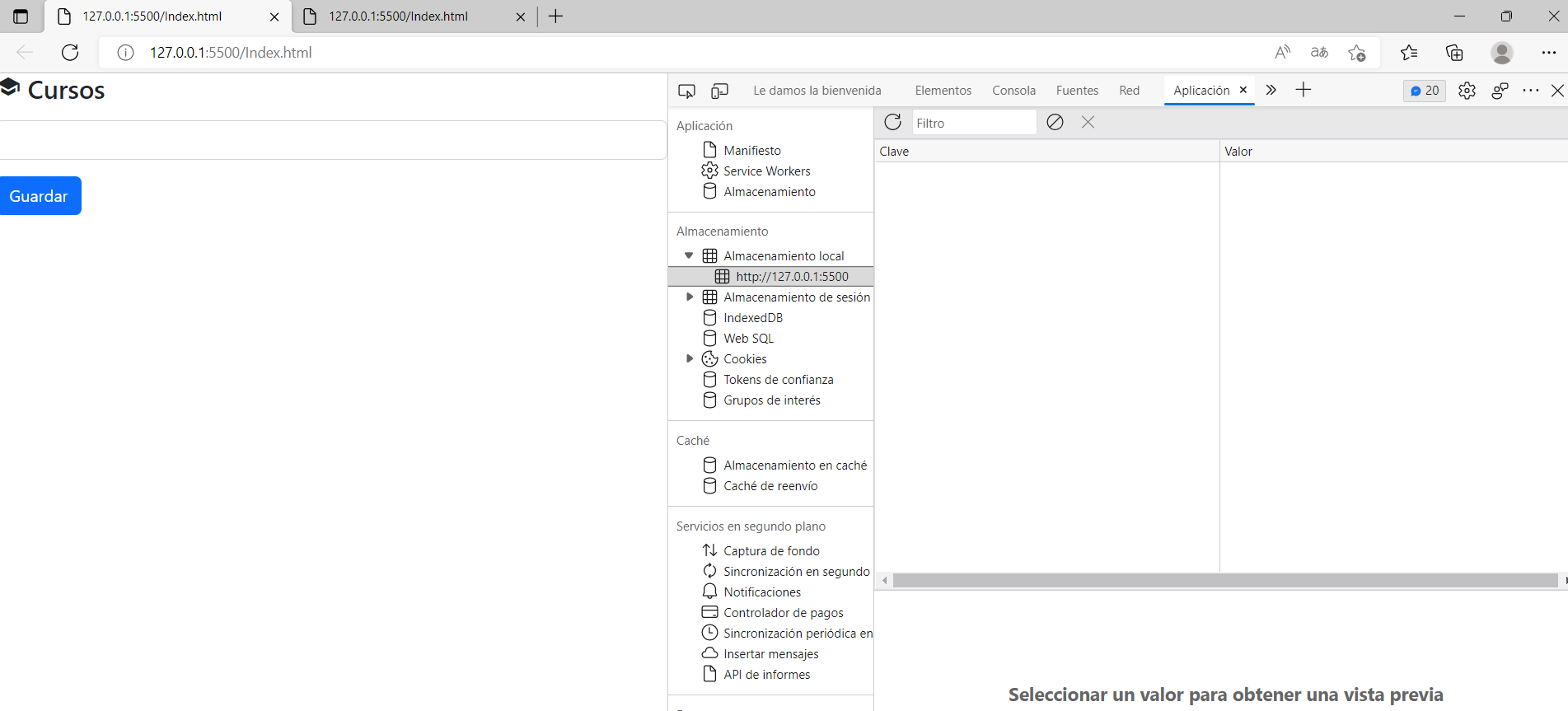
    }

}

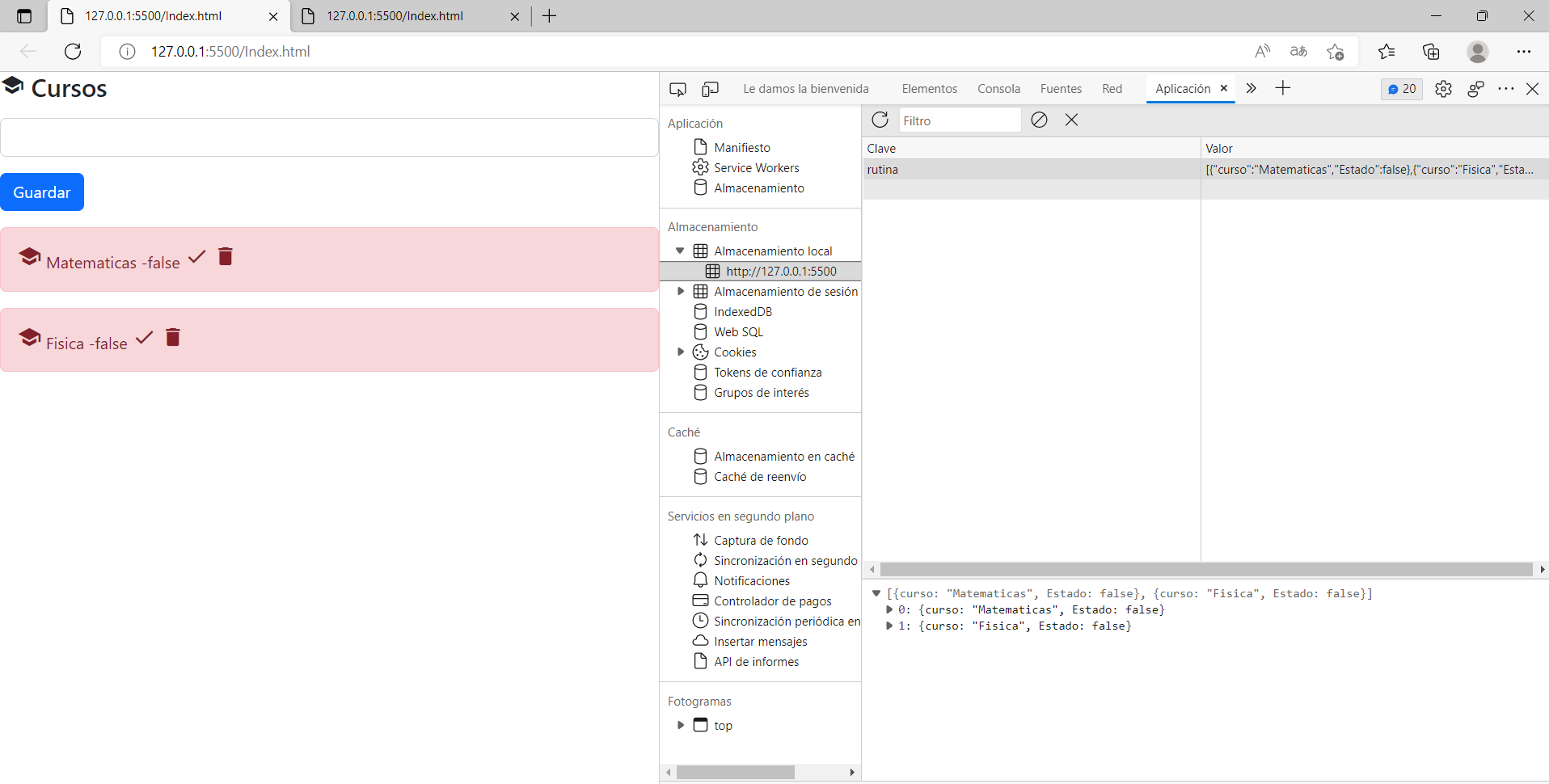
Resultado (lo que se refleja en browser):



Limpiamos el **localStorage**



Agregando información para corroborar que funciona:



**Referencias:** **https://www.acontracorrientech.com/local-storage-en-javascript-guia-completa/**