

“Curso de

HTML 5 y CSS 3”



Por: Ing. Ubaldo Acosta



Experiencia y Conocimiento para tu vida

www.GlobalMentoring.com.mx



Experiencia y Conocimiento para tu vida

Ejercicio: Manejo de Local Storage en HTML5



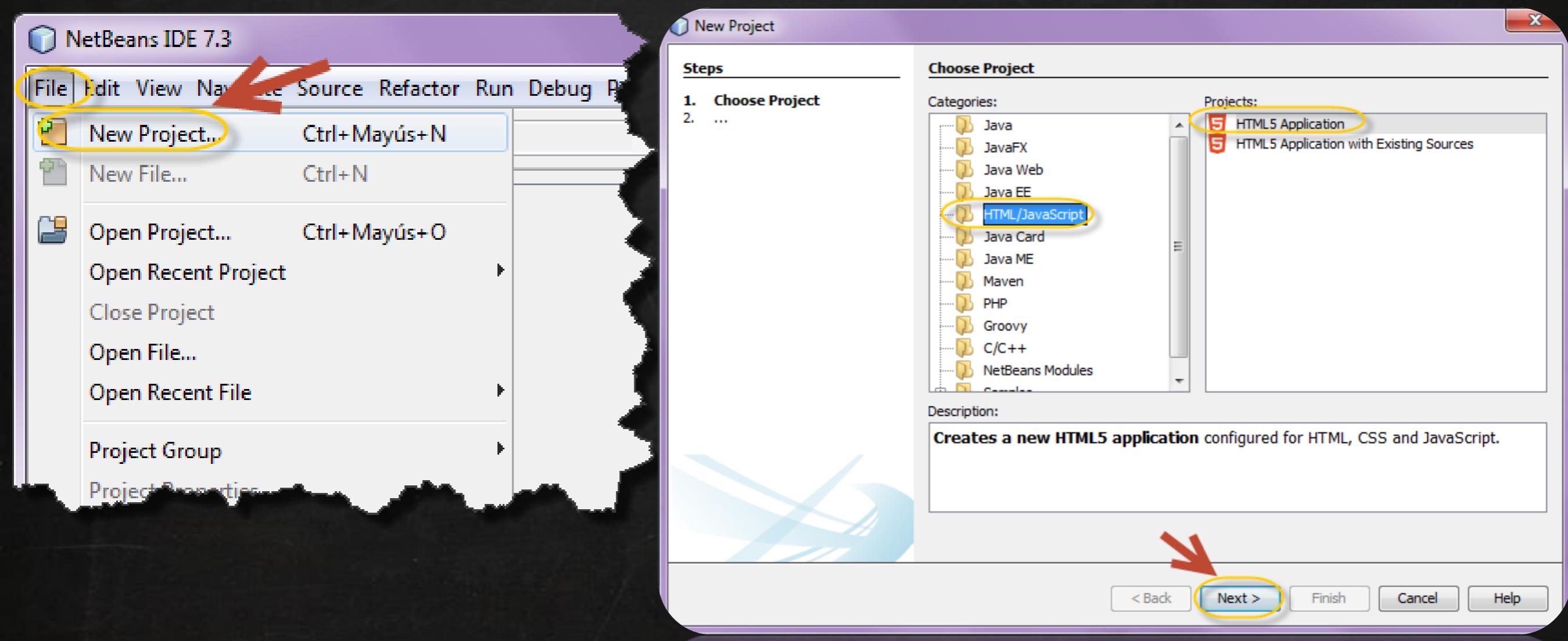
Objetivo del Ejercicio

- El objetivo del ejercicio es utilizar el API para el manejo de Local Storage en HTML5.
- Al finalizar deberemos observar la siguiente página:



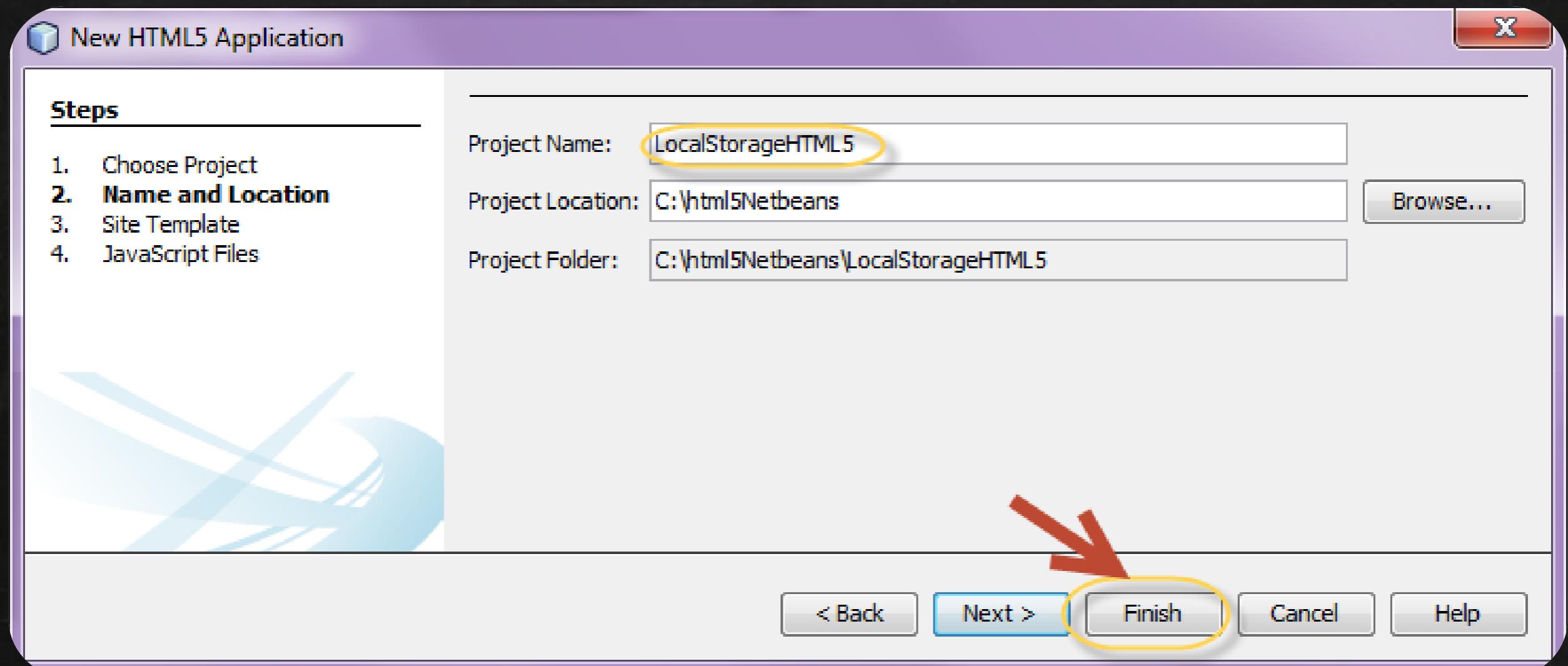
Paso 1. Crear proyecto LocalStorage

- Creamos el proyecto LocalStorageHTML5.



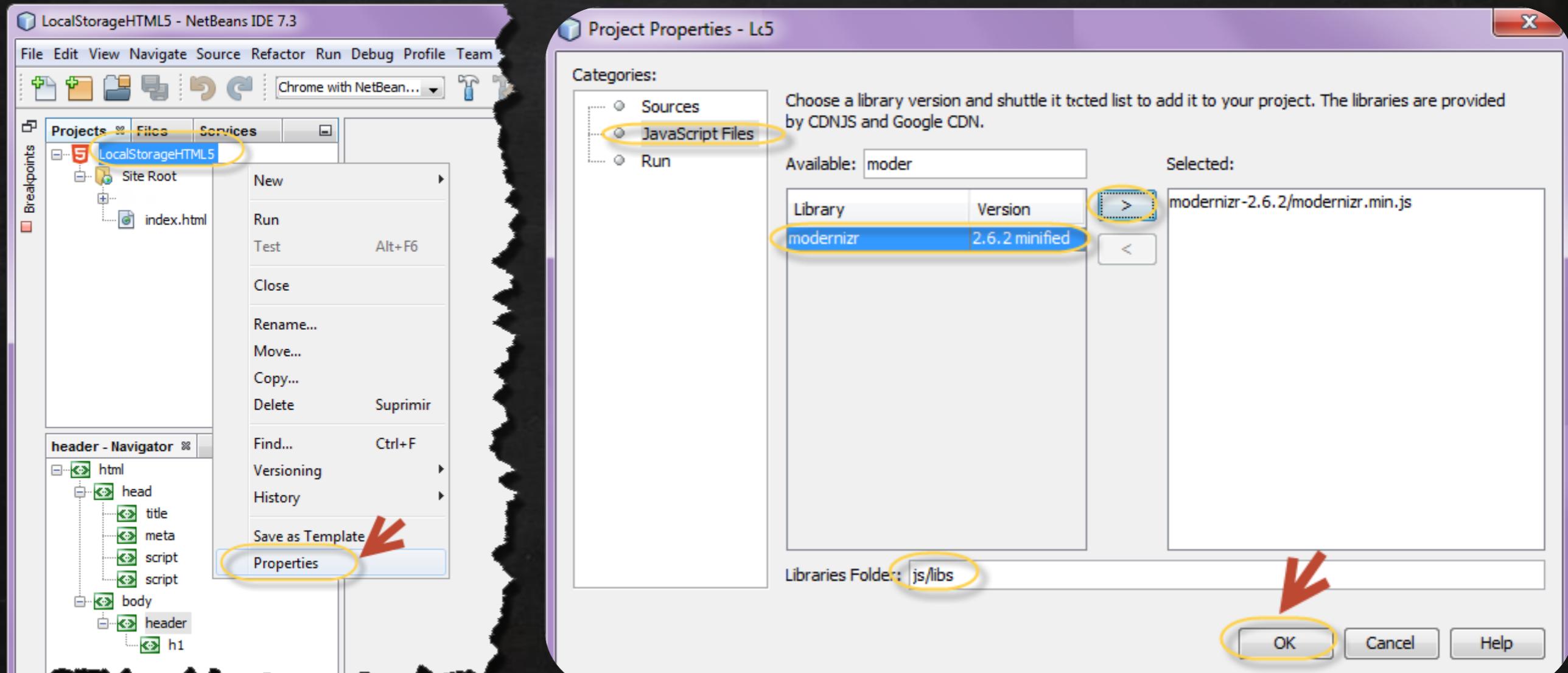
Paso 1. Crear proyecto LocalStorage (cont)

- El nombre del proyecto es LocalStorageHTML5.



Paso 2. Agregamos Modernizr

- Agregamos la librería de Modernizr a nuestro proyecto:



Paso 3. Agregamos el código

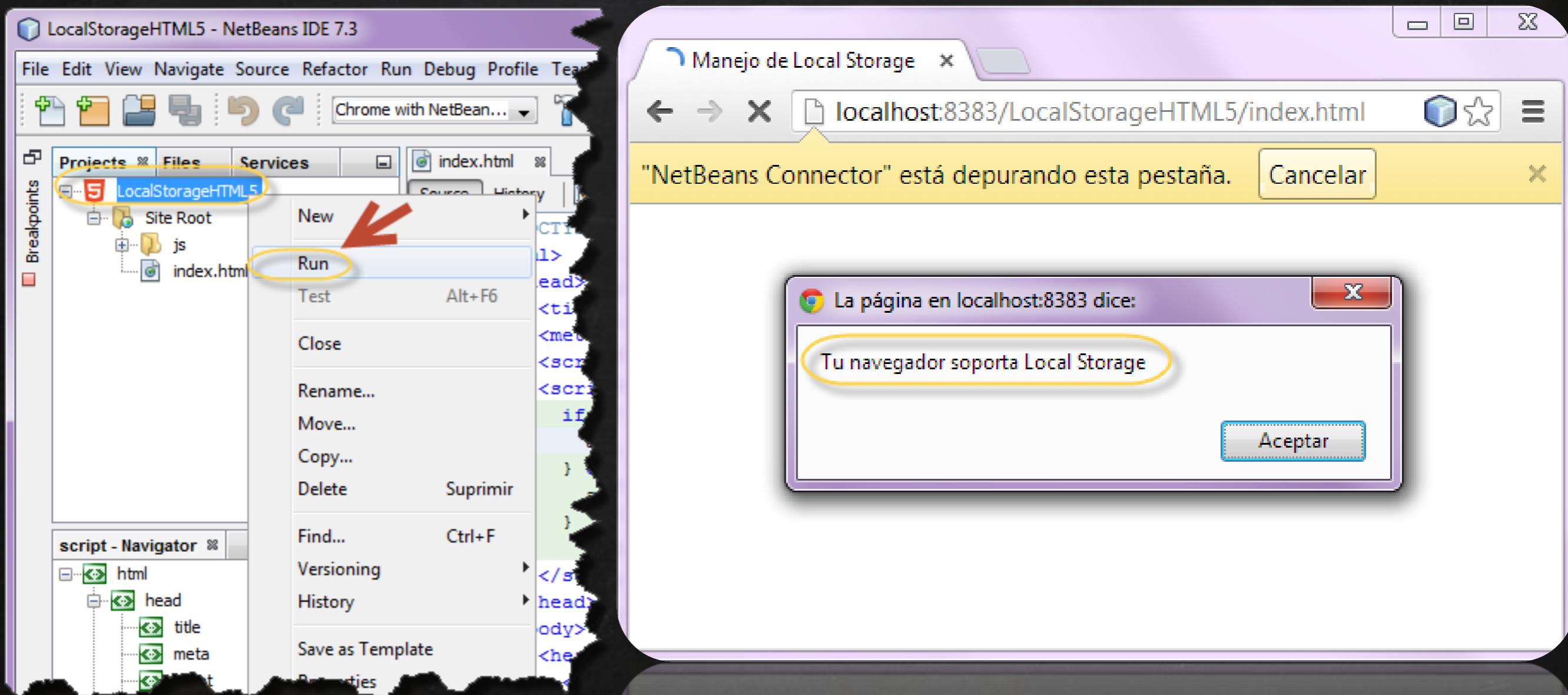
- Agregamos el código al archivo index.html:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Manejo de Local Storage</title>
    <meta charset="UTF-8">
    <script src="js/libs/modernizr-2.6.2/modernizr.min.js"></script>
    <script type="text/javascript">
      if (Modernizr.localstorage) {
        alert("Tu navegador soporta Local Storage");
      } else {
        alert("Lo lamentamos, tu navegador no soporta Local Storage");
      }

    </script>
  </head>
  <body>
    <header>
      <h1>Manejo de Local Storage</h1>
    </header>
  </body>
</html>
```

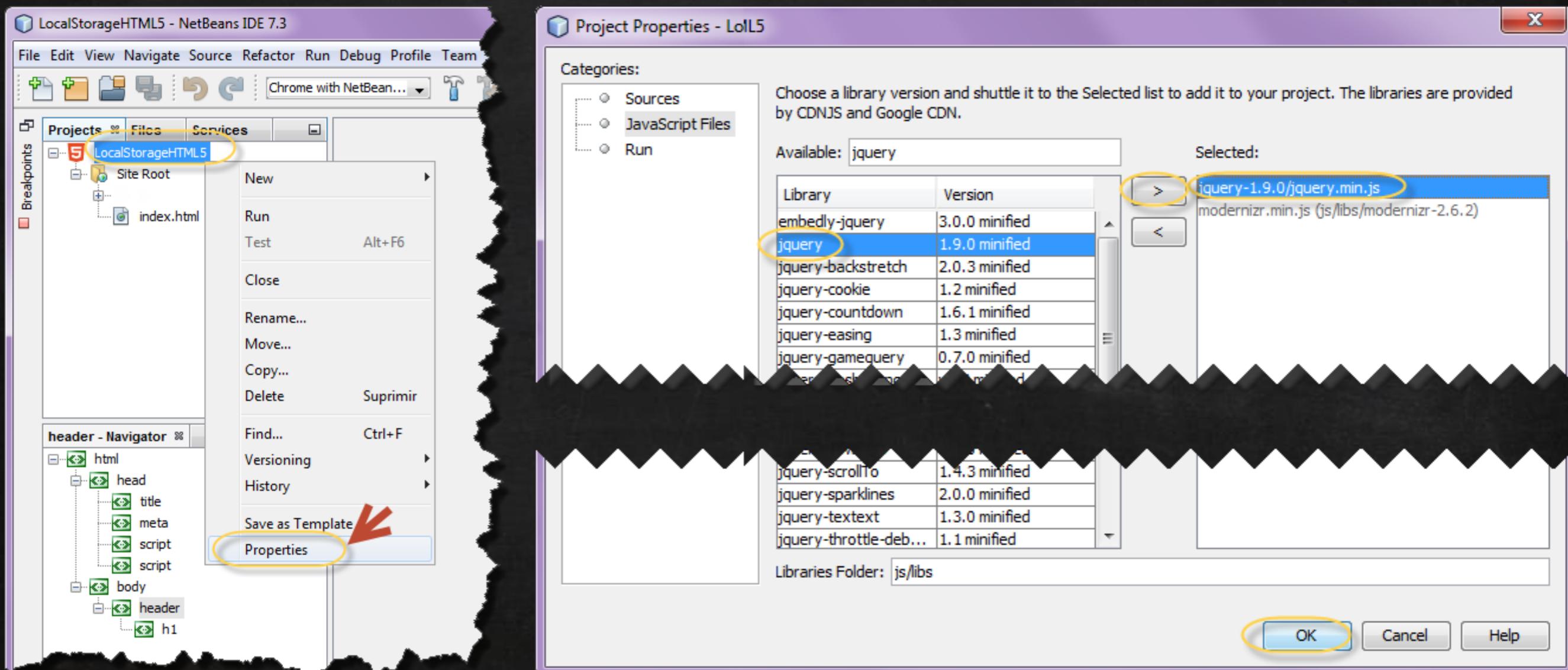
Paso 4. Ejecutamos el proyecto

- Verificamos que nuestro navegador soporte LocalStorage:



Paso 5. Agregamos JQuery

- Agregamos la librería de JQuery a nuestro proyecto:



Paso 6. Agregamos el código

- Agregamos el código al archivo index.html:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" />
    <title>Manejo de Local Storage</title>
    <script src="js/libs/jquery-1.9.0/jquery.min.js"></script>
    <script type="text/javascript">
      $(document).ready(function() {

        $("#lista").blur(function() {
          localStorage.setItem('listado', $("#lista").html());
          //regrescamos la lista
          location.reload();
        });

        if (localStorage.getItem('listado')) {
          $("#mensaje").html("<h3>Datos almacenados en Local Storage:</h3>");
          $("#lista").html(localStorage.getItem('listado'));
          $("#resultado").text(localStorage.getItem('listado'));
        } else {
          $("#mensaje").text("No hay datos en Local Storage.");
        }
      });
    </script>
  </head>
  <body>
    <h1>Manejo de Local Storage</h1>
    <input type="text" id="lista" placeholder="Ingresa los datos a almacenar" />
    <p>Mensaje: <span id="mensaje"></span></p>
    <p>Resultado: <span id="resultado"></span></p>
  </body>
</html>
```

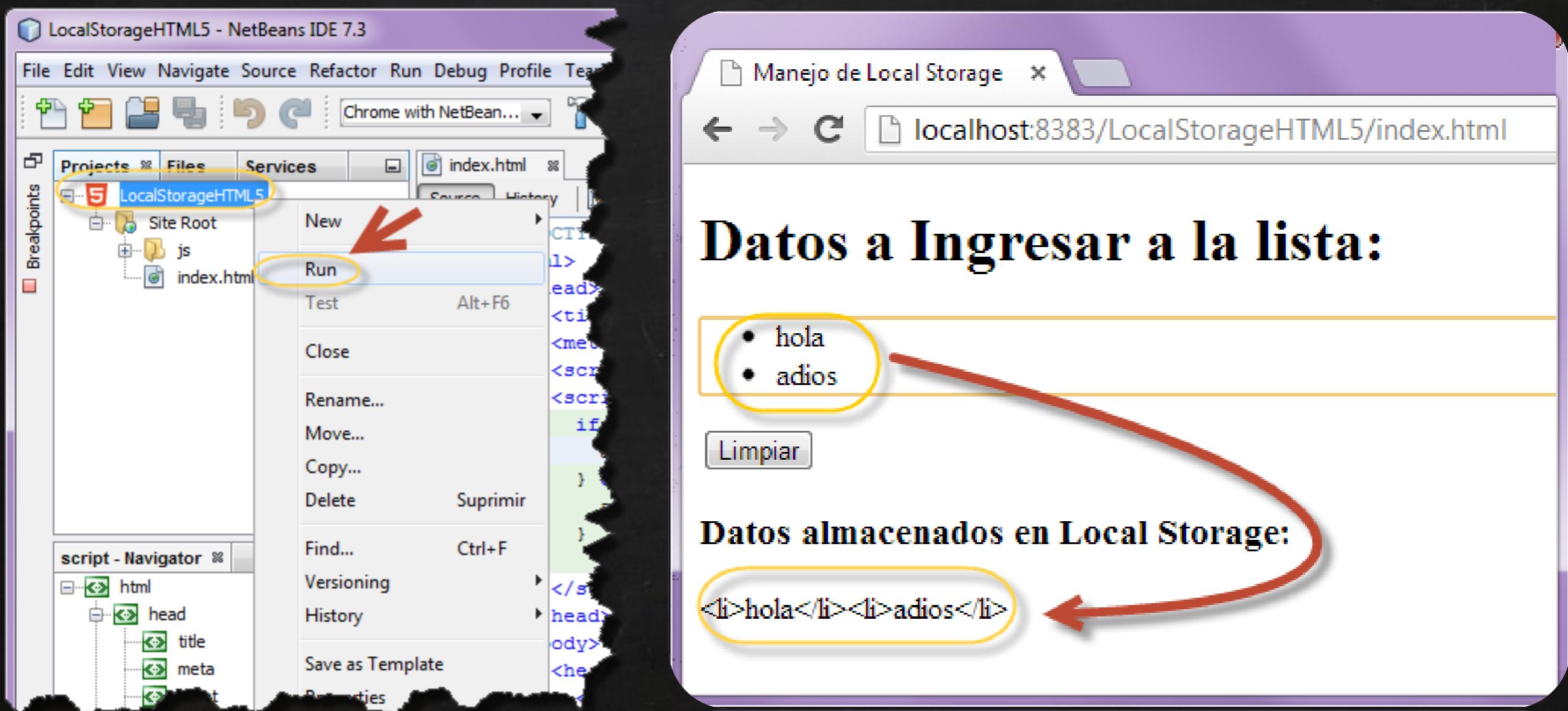
Paso 6. Agregamos el código (cont)

- **Agregamos el código al archivo index.html (cont):**

```
$("#limpiar").click(function() {  
    localStorage.removeItem('listado');  
    location.reload();  
});  
  
});  
</script>  
</head>  
<body>  
    <header>  
        <h1>Datos a Ingresar a la lista:</h1>  
    </header>  
    <div>  
        <ul id="Lista" contenteditable="true">  
            <li></li>  
        </ul>  
    </div>  
    <input type="button" id="Limpiar" value="Limpiar" />  
  
<div id="mensaje"></div>  
<div id="resultado"></div>  
</body>  
</html>
```

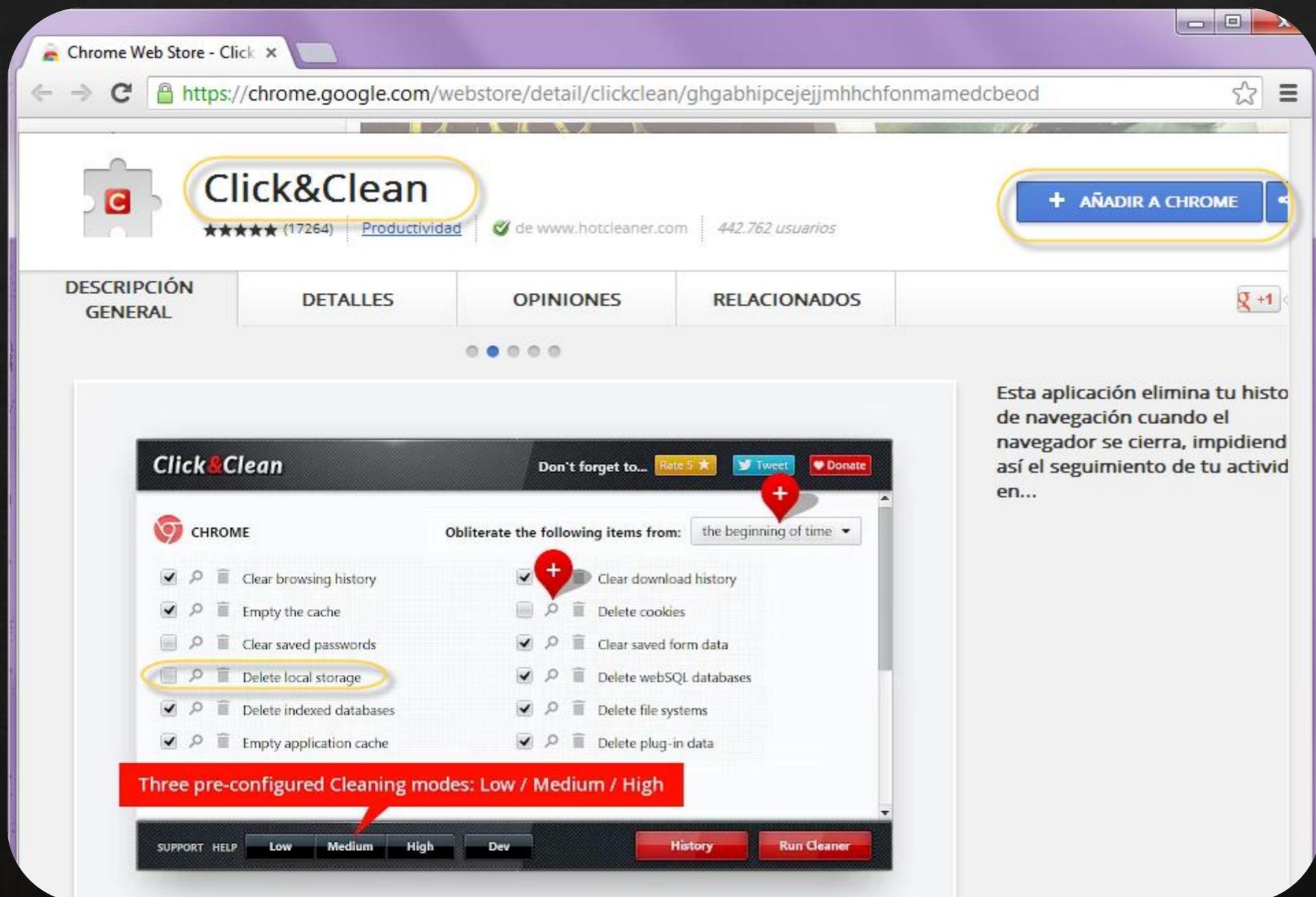
Paso 7. Ejecutamos el proyecto

- Agregamos datos y vemos que incluso cerrando el navegador los datos siguen mostrándose:



Paso 8. Borrar Local Storage

- Otra manera de borrar el Local Storage, además de los métodos ya vistos, es con un plug-in en el caso de Google Chrome, aunque esto es opcional.



Paso 9. Manejo de Eventos Local Storage

- Es posible avisar de un evento a otra ventana que tenga la misma información de Local Storage. A esto se le conoce como manejo de eventos de Local Storage.

```
window.addEventListener('storage', escucharEvento, false);
```

```
function escucharEvento(e) {  
    alert("Evento generado localStorage" + e.storageArea.listado);  
}
```

- El objeto e, es el que tiene la información del evento que se está generando, como puede ser: e.key, e.oldValue, e.newValue, e.url y el objeto storageArea, con el cual podemos recuperar el valor actual del key en interés. A continuación vamos a modificar el proyecto para que soporte el manejo de eventos de LocalStorage.

Paso 10. Agregamos el código

- **Modificamos el código del archivo index.html:**

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" />
    <title>Manejo de Local Storage</title>
    <script src="js/libs/jquery-1.9.0/jquery.min.js"></script>
    <script type="text/javascript">

      //Funcion de inicializacion
      $(document).ready(function() {

        $("#lista").blur(function() {
          localStorage.setItem('listado', $("#lista").html());
          //regrescamos la lista
          location.reload();
        });

        if (localStorage.getItem('listado')) {
          $("#mensaje").html("<h3>Datos almacenados en Local Storage:</h3>");
          $("#lista").html(localStorage.getItem('listado'));
          $("#resultado").text(localStorage.getItem('listado'));
        } else {
          $("#mensaje").html("<h3>No hay datos en Local Storage.</h3>");
        }
      });
    </script>
  </head>
  <body>
    <input type="text" id="lista" value="Alta 1" />
    <div id="resultado"></div>
    <div id="mensaje"></div>
  </body>
</html>
```

Paso 10. Agregamos el código (cont)

- **Modificamos el código del archivo index.html:**

```
//Limpiamos la lista
$("#limpiar").click(function() {
    localStorage.removeItem('listado');
    // limpia todas las variables de localstorage
    //localStorage.clear();
    location.reload();
});

//Registramos el evento escucha los cambios
//al modificar cualquier elemento
window.addEventListener('storage', escucharEvento, false);

function escucharEvento(e) {
    //alert("Evento generado localStorage" + e.storageArea.listado);
    location.reload();
}

});
</script>
</head>
```

Paso 10. Agregamos el código (cont)

- Modificamos el código del archivo `index.html`:

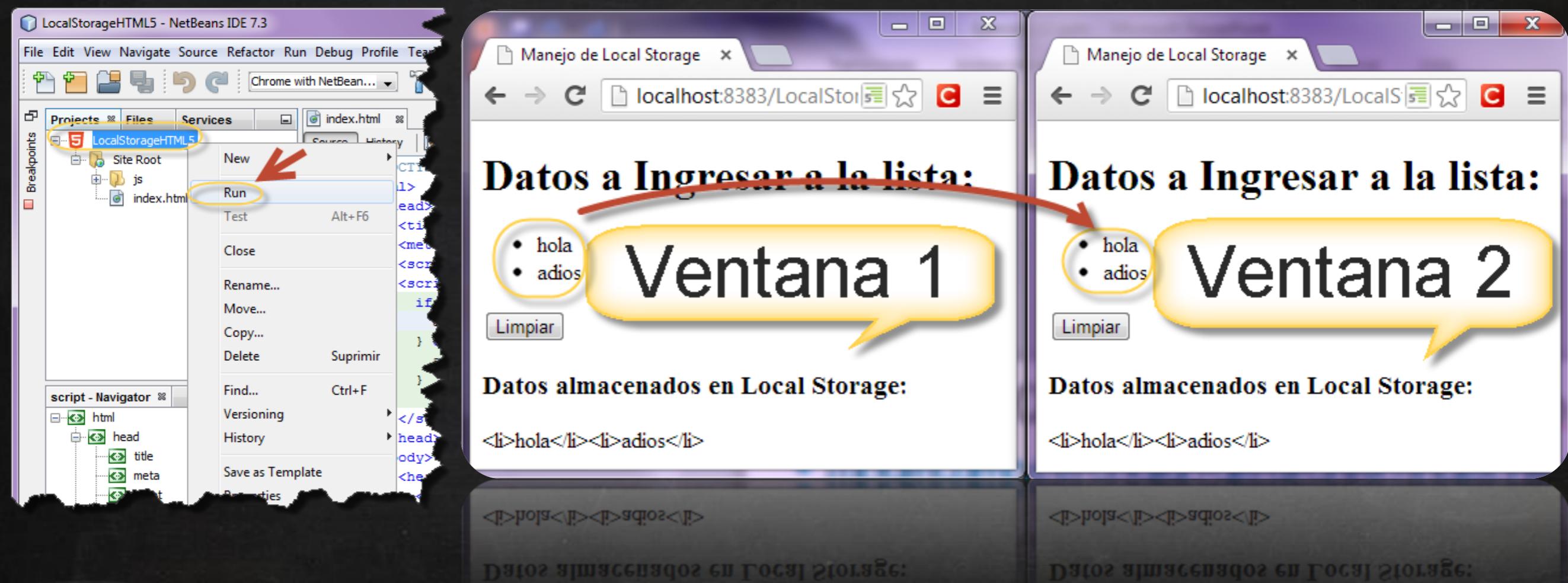
```
<body>
  <header>
    <h1>Datos a Ingresar a la lista:</h1>
  </header>
  <div>
    <ul id="Lista" contenteditable="true">
      <li></li>
    </ul>
  </div>
  <input type="button" id="Limpiar" value="Limpiar" />

  <div id="mensaje"></div>

  <div id="resultado"></div>
</body>
</html>
```

Paso 4. Ejecutamos el proyecto

- Abrimos 2 ventanas y verificamos que se replique la información debido al manejo de eventos:



Conclusión

- Con este ejercicio hemos visto cómo manejar el API de Local Storage, con la cual podemos sustituir las cookies persistentes. Esta información quedará almacenada en el navegador hasta que eliminemos esta información del mismo. No importa que cerremos el navegador.
- Además con el manejo de eventos, podemos reportar a todas las ventanas que manejen la misma llave de Local Storage que hay un cambio y por lo tanto sincronicen la información si así lo requiere la aplicación.

“Curso de

HTML 5 y CSS 3”



Por: Ing. Ubaldo Acosta



Experiencia y Conocimiento para tu vida

www.GlobalMentoring.com.mx