



UNIVERSIDAD NACIONAL AUTONOMA DE MÉXICO
FACULTAD DE ESTUDIOS SUPERIORES PLANTEL ARAGON



EXTRA. INGENIERIA DE SOFTWARE

INGENIERIA DE SOFTWARE

APP DE TAREAS

CARLOS NOE SALGADO FLORES

MARIANA VERDUZCO RODRIGUEZ

INTRODUCCION

Esta aplicación fue realizada con el lenguaje de programación de JavaScript además del lenguaje de enmarcado de HTML siendo que estos se pueden comportar como un sistema de aplicación multiplataforma sin la necesidad de instalar nada y a su vez nos ayuda en que no necesita tanto recursos para poder ser ejecutado más que un navegador e internet ya que para realizar un diseño más eficiente, responsivo y que su desarrollo sea más veloz se utilizó como diseño el framework Bootstrap.

Se realizo una aplicación la cual tiene como objetivo ser el registro de tareas de una persona para su vida cotidiana y que en ella podamos ver que orden llevan las tareas y que de alguna manera fuera lo suficientemente entendible para el usuario.

El tiempo de realización de este programa fue de 1 día ya que se tomó importancia el probar que la aplicación no tuviera errores básicos al pasar al diseño en celulares ya que al principio esto genero algunos conflictos ya que no se mostraban las tareas en el cambio de tamaño pudiéndose corregir en tiempo para que funcionara como se esperaba que lo hiciera además se buscó que no fuera una aplicación pesada para que cualquier dispositivo por más escaso en recursos no tardara en cargar los diseños.

La parte más complicada del proyecto fue poder pensar cual sería una idea que cumpliera con los requisitos que se piden que aun que no son muchos tenía que tener alguna estructura que pudiera funcionar en alguna caso específico y que funcionara como un sistema.



OBJETIVO DEL SISTEMA

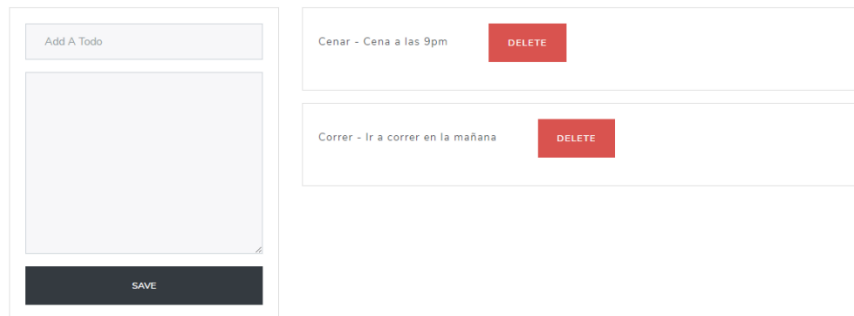
El objetivo de este sistema es que el usuario pueda organizar sus pendientes mediante una simple app la cual nos permite almacenar en el localStorage nuestras tareas sin necesidad de internet o alguna otra herramienta lo que lo hace mas factible para tener que organizarnos.

Cuantas veces no nos cuesta encontrar una app para organizarnos y que esta funcione mediante internet en esta sencilla pagina lo que realizamos es un todo en uno, donde podemos solo simplemente añadir una tarea con su descripción y posteriormente si ya no la necesitamos solamente le picamos al botón de borrar y eso sería todo la tarea que eliminada de nuestra base de datos

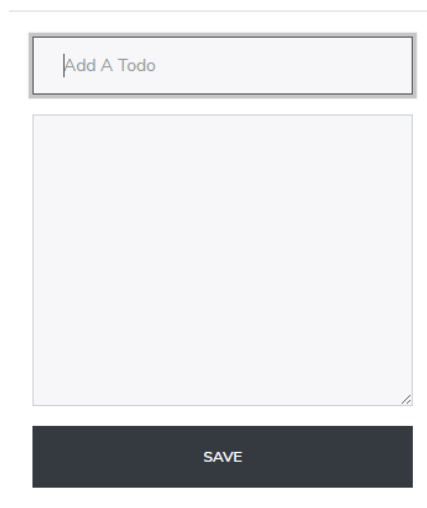
A screenshot of the task management app interface. On the left, there is a form to add a new task. It has a text input field with the placeholder 'Add A Todo', a large text area for description, and a 'SAVE' button at the bottom. On the right, there is a list of tasks. Each task is in a white box with a red 'DELETE' button. The first task is 'Cenar - Cena a las 9pm' and the second is 'Correr - Ir a correr en la mañana'.

DESCRIPCION GENERAL DEL SISTEMA

Al abrir la aplicación nos muestra directamente el programa sin la necesidad de tener que meternos en configuraciones y demás, es una sola página lo que lo hace mejor muy similar a lo que sería una landing page pero con la diferencia de que esta si tiene funcionalidades y no solo muestra información



Primero tenemos que pensar en qué tarea es la que queremos anotar



Una vez realizado esto, le damos en el botón de save, para guardar la tarea que queremos guardar

Add A Todo

SAVE

Extra de Ing. de Software - Tengo que hacer la documentación del extra de ingeniería o voy a reprobado la materia

DELETE

Bien, una vez entendido hasta este punto entendemos donde se nos guardan nuestras tareas y cómo funciona la aplicación, en realidad es bastante básico pero útil a la hora de añadir tareas.

Add A Todo

SAVE

Extra de Ing. de Software - Tengo que hacer la documentación del extra de ingeniería o voy a reprobado la materia

DELETE

Comer - Tengo que comer 3 veces al día y tomar agua

DELETE

Correr - Irme a correr temprano para poder tener buena condición

DELETE

Pagar la luz - Antes del 30 de agosto pagar el recibo o me quedo sin luz u.u

DELETE

Pero qué pasa si ya no queremos una tarea de nuestra lista, bueno sencillo para eso cree el botón de “delete” una vez que no queramos mas una tarea pinchamos el botón y listo desaparece la tarea. En este caso elimine “comer”.

Add A Todo

SAVE

Extra de Ing. de Software - Tengo que hacer la documentación del extra de ingeniería o voy a reprobado la materia

DELETE

Correr - Irme a correr temprano para poder tener buena condición

DELETE

Pagar la luz - Antes del 30 de agosto pagar el recibo o me quedo sin luz u.u

DELETE

Hasta aquí todo bien entendimos el funcionamiento de esta sencilla aplicación de tareas hablando de interfaz y a la vista del usuario. Sin embargo aun nos falta saber que hay detrás de esta app, así que veamos.

Explicare el código mediante el modelo vista controlador para que sea mas entendible y organizado.



Vista.

Tenemos nuestra vista simplemente en un html donde tenemos el formulario y el uso de Bootstrap el cual simplemente va a añadir estilos en el nav y los botones además de darle un toque responsivo a nuestra app.

```
index.html x JS app.js
Tareas-App > index.html > html > body
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8">
5     <title>Javascript App de Tareas</title>
6     <!-- BOOTSTRAP 4 -->
7     <link rel="stylesheet" href="https://bootswatch.com/4/lux/bootstrap.min.css">
8   </head>
9   <body>
10
11     <!-- NAVEGACION -->
12
13     <nav class="navbar navbar-dark bg-dark">
14       <div class="container">
15         <a class="navbar-brand" href="#">App de Tareas</a>
16       </div>
17     </nav>
18
19     <div class="container">
20       <div class="row my-5">
21         <div class="col-md-4">
22
23           <div class="card">
24             <div class="card-body">
25               <!-- FORMULARIO -->
26               <form id="formulario">
27                 <div class="form-group">
28                   <input type="text" id="title" placeholder="Add A Todo" class="form-control">
29                 </div>
30                 <div class="form-group">
31                   <textarea id="description" cols="30" rows="10" class="form-control"></textarea>
32                 </div>
33                 <button type="submit" class="btn btn-dark btn-block">Save</button>
34               </form>
35             </div>
36           </div>
37         </div>
38
39         <div class="col-md-8">
40           <div id="tasks"></div>
41         </div>
42       </div>
43     </div>
44
45     <!-- SCRIPTS -->
46     <script src="./app.js"></script>
47   </body>
48 </html>
```

Controlador.

Bien ahora nuestra lógica la cual viene dentro de nuestro js, tenemos 3 funciones las cuales son las importantes dentro de nuestra aplicación así que explicare para que funciona cada una de ellas.

```
function saveTask(e) {  
  let title = document.getElementById('title').value;  
  let description = document.getElementById('description').value;  
  console.log(description)  
  
  let task = {  
    title,  
    description  
  };  
  
  if(localStorage.getItem('tasks') === null) {  
    let tasks = [];  
    tasks.push(task);  
    localStorage.setItem('tasks', JSON.stringify(tasks));  
  } else {  
    let tasks = JSON.parse(localStorage.getItem('tasks'));  
    tasks.push(task);  
    localStorage.setItem('tasks', JSON.stringify(tasks));  
  }  
}
```

En primera instancia tenemos a saveTask, la cual se va a encargar de como lo dice la traducción guardar nuestras tareas, sin embargo ¿cómo lo hace? Bien mediante un objeto y un arreglo almacena las tareas que el usuario va a añadir, primero guarda el objeto y después lo convierte a string para añadirlo dentro del arreglo.


```
function getTasks() {
  let tasks = JSON.parse(localStorage.getItem('tasks'));
  let tasksView = document.getElementById('tasks');
  tasksView.innerHTML = '';
  for(let i = 0; i < tasks.length; i++) {
    let title = tasks[i].title;
    let description = tasks[i].description;

    tasksView.innerHTML += `<div class="card mb-3">
      <div class="card-body">
        <p>${title} - ${description}</p>
        <a href="#" onclick="deleteTask('${title}')" class="btn btn-danger ml-5">Delete</a>
      </div>
    </div>`;
  }
}

getTasks();
```

Ahora tenemos getTasks la cual nos va a permitir mandar llamar a la tarea que vayamos a añadir y posteriormente imprimirla en pantalla para eso tenemos el innerHTML, primero convierte el json en objeto nuevamente y después lo que hace es recorrer el contenido para posteriormente imprimirlo.

```
function deleteTask(title) {
  console.log(title)
  let tasks = JSON.parse(localStorage.getItem('tasks'));
  for(let i = 0; i < tasks.length; i++) {
    if(tasks[i].title == title) {
      tasks.splice(i, 1);
    }
  }
}
```

Por ultimo tenemos a deleteTask la cual como lo dice su nombre elimina las tareas que ya no vayamos a ocupar y es simple mediante la propiedad de splice nos permite eliminar el contenido que hay dentro de nuestro arreglo.

Modelo.

En este apartado final es donde se van a almacenar los datos de nuestra aplicación y de las cuales van a sufrir cambios mediante `document.getElementById` mandamos a llamar a nuestro input y lo guardamos como un valor para en un futuro asignarle un valor.

```
function saveTask(e) {  
  let title = document.getElementById('title').value;  
  let description = document.getElementById('description').value;  
  console.log(description)
```

HERRAMIENTAS DE ANALISIS

✓ Tabla de decisión

Palabra está vacía	Si	Si	No	No
Descripción está vacío	SI	No	No	SI
Pasa a añadir tarea			X	
Manda error	X	X		X

✓ Diagrama IPO

Entrada	Proceso	Salida
>Usuario escribe el titulo y posteriormente le añade una descripción a su tarea	>Se verifica que: -No sea un título vacío -Agregue una descripción el usuario	>Ya verificado que no es un título vacío y/o descripción entonces posteriormente la app pasa a imprimir lo que hay dentro del input

CONCLUSIONES

Fué un reto al principio poder desarrollar esta aplicación ya que como tal no tenía la idea de que presentar en esta extraordinario sin embargo una vez se me ocurrió fue divertido programarla ya que no solo use html y javascript si no que también use un framework de diseño como lo es Bootstrap y así poder manejar mas tecnologías además de que use el API de la LocalStorage para que el navegador una vez salga de la app y la vuelva a abrir los datos se puedan almacenarse sin ser eliminados cuando se cierre el navegador.

Aunque soy consciente que la app es bastante básica nos ayuda a entender el funcionamiento de un sistema y de una aplicación en un sitio web.