CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERÍAS (CUCEI)

DIVISIÓN DE ELECTRÓNICA Y COMPUTACIÓN

DEPARTAMENTO DE CIENCIAS COMPUTACIONALES

Carrera: Ingeniería en Computación

Nombre Materia: Seminario de Solución de Problemas de Sistemas Operativos

Profesor: Valdés López Julio Esteban

SECCIÓN: D08

Nombre alumno: López Arellano Ricardo David

CODIGO: 217136143

Tarea 1: Práctica 1

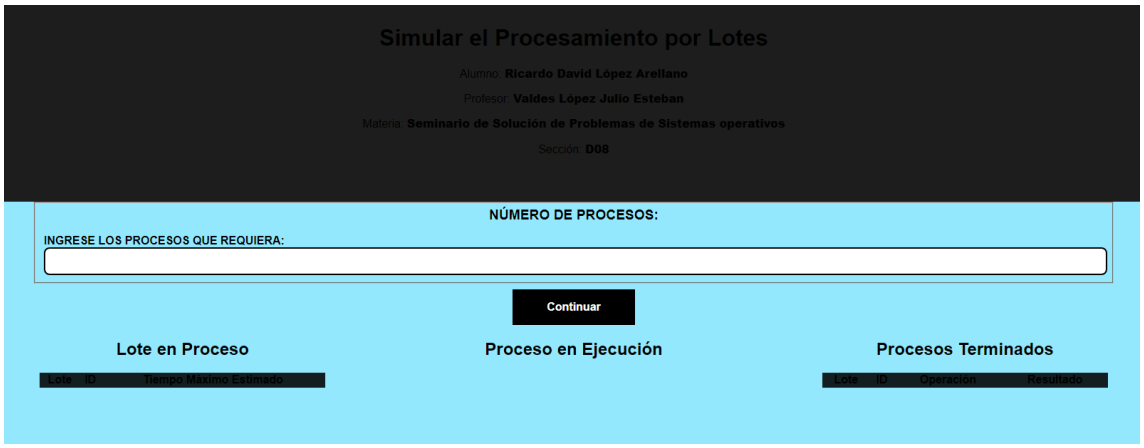
Fecha de entrega: 03/01/2023

CONTENIDO

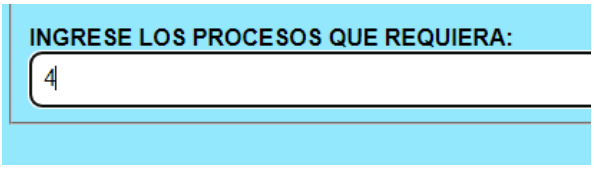
INTRODUCCIÓN:

Todos los algoritmos se pueden ejecutar como un proceso por lotes. Es decir, que se pueden ejecutar utilizando no sólo un único conjunto de insumos, sino varios de ellos y ejecutar el algoritmo tantas veces sea necesario. Esto es útil al procesar grandes cantidades de datos, ya que no es necesario poner en marcha el algoritmo muchas veces desde la caja de herramientas.

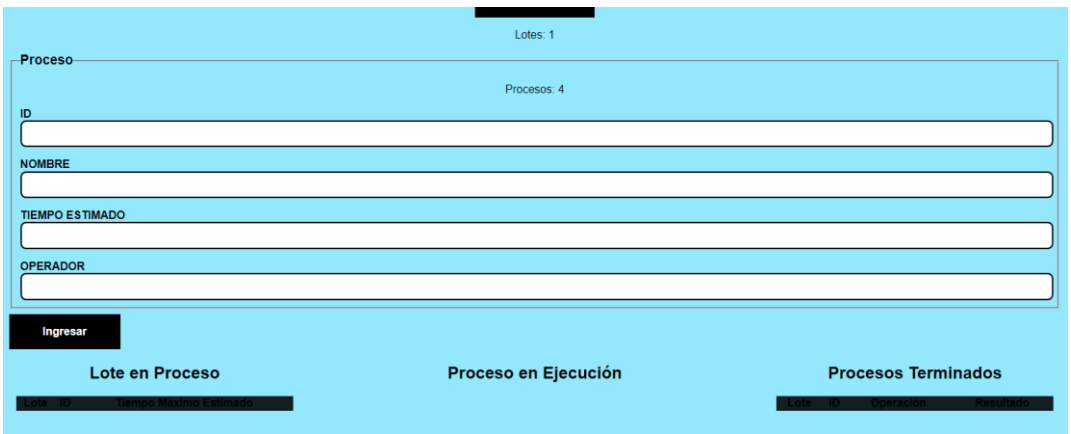
CAPTURAS DE PANTALLA:



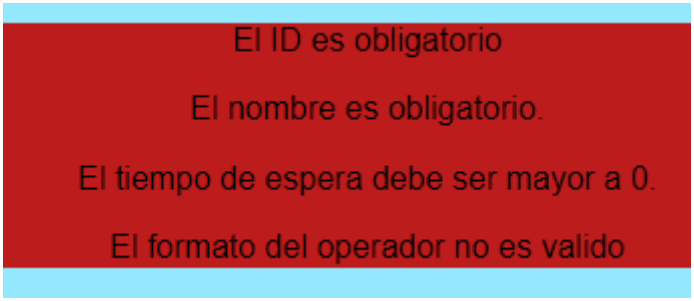
Esta en la pantalla principal de la práctica, aquí ingresamos los números de procesos que necesitamos.



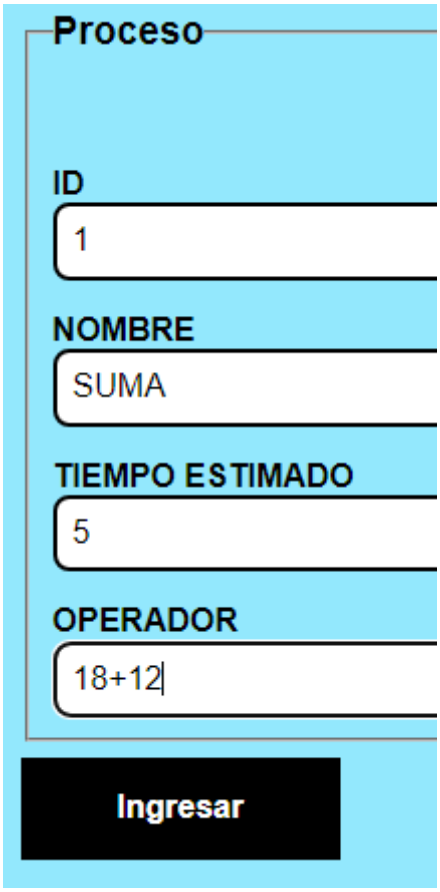
Por ejemplo ingresamos 4 procesos y presionamos continuar.

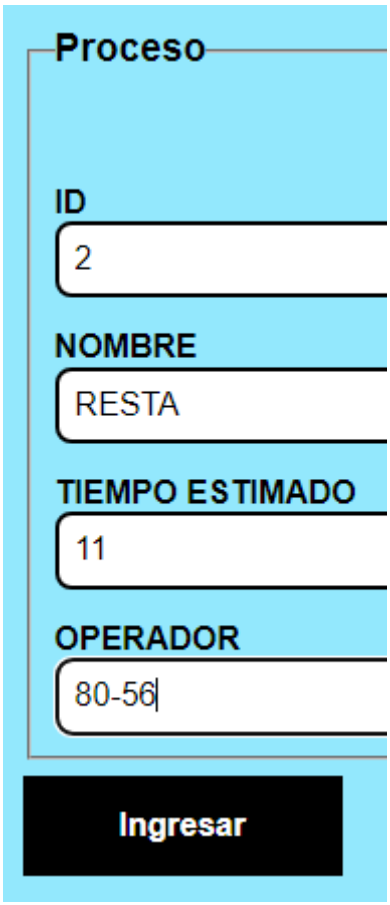


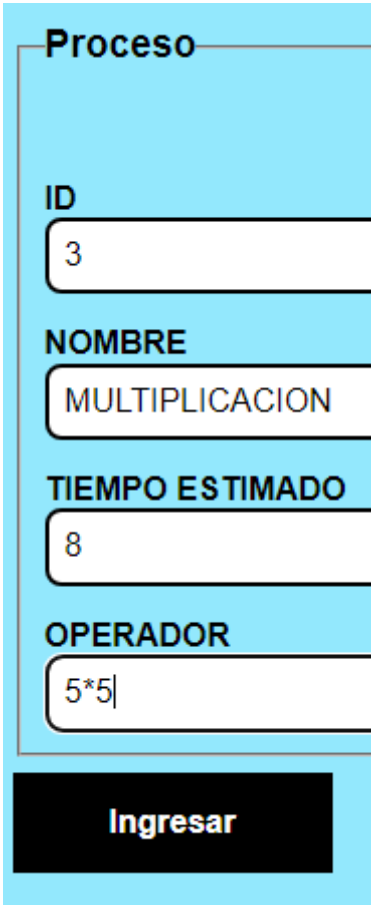
Nos mostrará la siguiente pantalla. Aquí ingresaremos cada uno de los datos requeridos, en dado caso que cometas un error te mostrará un aviso como los siguientes:

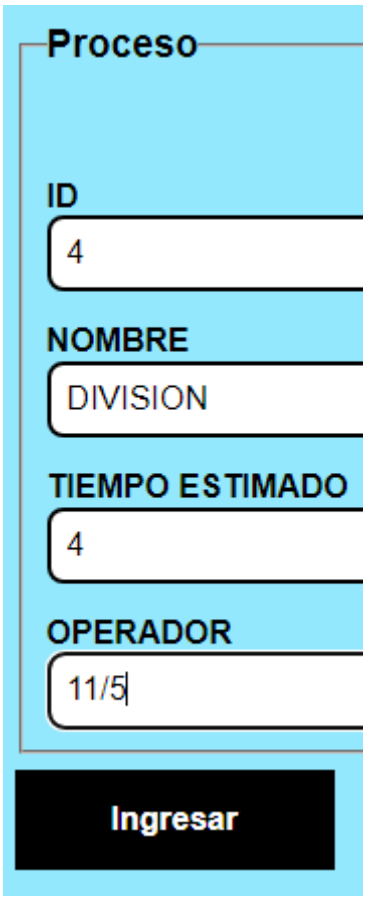


Ahora ingresamos los 4 procesos que pedimos anteriormente:

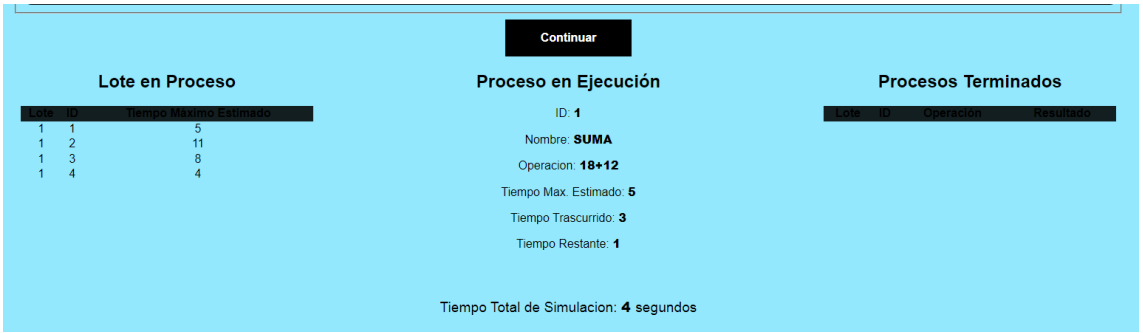
1. 

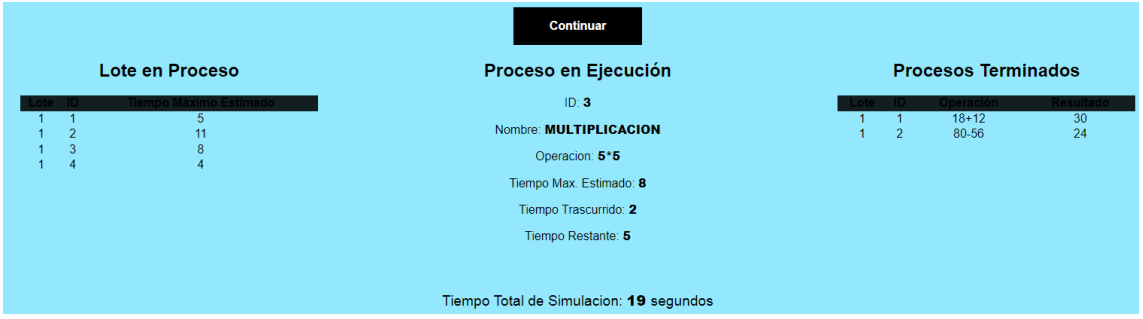
2. 
Proceso
ID
2
NOMBRE
RESTA
TIEMPO ESTIMADO
11
OPERADOR
80-56
Ingresar

3. 
Proceso
ID
3
NOMBRE
MULTIPLICACION
TIEMPO ESTIMADO
8
OPERADOR
5*5
Ingresar


Proceso
ID
4
NOMBRE
DIVISION
TIEMPO ESTIMADO
4
OPERADOR
11/5
Ingresar

4.


Continuar
Lote en Proceso
Lote ID Tiempo Máximo Estimado
1 1 5
1 2 11
1 3 8
1 4 4
Proceso en Ejecución
ID: 1
Nombre: SUMA
Operacion: 18+12
Tiempo Max. Estimado: 5
Tiempo Trascurrido: 3
Tiempo Restante: 1
Tiempo Total de Simulacion: 4 segundos
Procesos Terminados
Lote ID Operación Resultado


Continuar
Lote en Proceso
Lote ID Tiempo Máximo Estimado
1 1 5
1 2 11
1 3 8
1 4 4
Proceso en Ejecución
ID: 3
Nombre: MULTIPLICACION
Operacion: 5*5
Tiempo Max. Estimado: 8
Tiempo Trascurrido: 2
Tiempo Restante: 5
Tiempo Total de Simulacion: 19 segundos
Procesos Terminados
Lote ID Operación Resultado
1 1 18+12 30
1 2 80-56 24

Así se verán los resultados de un por uno y con el contador general hasta abajo.

CÓDIGO FUENTE:

Index.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <link rel="stylesheet" href="css/normalize.css">
    <link rel="stylesheet" href="css/styles.css">

    <title>Simular el Procesamiento por Lotes</title>

</head>
<body>
    <header>
        <h1>Simular el Procesamiento por Lotes</h1>
        <p>Alumno: <span>Ricardo David López Arellano</span></p>
        <p>Profesor: <span>Valdes López Julio Esteban</span></p>
        <p>Materia: <span>Seminario de Solución de Problemas de Sistemas
operativos</span></p>
        <p>Sección: <span>D08</span></p>
    </header>
    <div class="Nproceso">

        <form class="formulario" action="">
            <fieldset>
                <center><legend>NÚMERO DE PROCESOS:  </legend></center>
                <label for="procesos">Ingrese los procesos que requiera:
</label>
                <input type="num" id="procesos" required>
            </fieldset>
```

```
        </form>
        <center><input class='boton' type="submit" id="ingresar"
value="Continuar"></center>
        <div id="errores" class='alerta'>
        </div>
    </div>
    <div id="proceso" class="Nproceso">

    </div>
    <div class="contenedor">
        <div>
            <!--Lote en Proceso-->
            <h2>Lote en Proceso</h2>
            <table class="tabla">
                <thead>
                    <th>Lote</th>
                    <th>ID</th>
                    <th>Tiempo Máximo Estimado</th>
                </thead>
                <tbody id="loteProceso">

                </tbody>
            </table>
        </div>
        <!-- Proceso en Ejecucion -->
        <div >
            <h2>Proceso en Ejecución</h2>
            <div id="procesoEjecucion" class="proceso">

            </div>
        </div>
        <div>
            <h2>Procesos Terminados</h2>
            <table class="tabla">
                <thead>
                    <th>Lote</th>
                    <th>ID</th>
                    <th>Operación</th>
                    <th>Resultado</th>
                </thead>
                <tbody id="procesoTerminado">

                </tbody>
            </table>
        </div>
    </div>
    <footer>
        <p id="tiempoTotal" class="ttotal"></p>
    </footer>
```

```
    <script src="js/Proceso.js"> </script>
    <script src="js/index.js"> </script>
</body>
</html>
```

index.js:

```
//LOPEZ ARELLANO RICARDO DAVID

document.addEventListener('DOMContentLoaded', function(){
    addEventListener();

});
let lote = [];
let lotes= [];
let numeroLotes, numeroProcesos, id = 0, Nlote = 0;

function addEventListener(){
    const botonIngresar =  document.querySelector('#ingresar');
    botonIngresar.addEventListener('click', ObtenerNumeroProcesos);
}

function ObtenerNumeroProcesos(){
    const procesos = document.querySelector('#procesos');
    //Valida si es un numero mayor a 0
    if(validarNumero(procesos.value)){
       lote = [];
       lotes = [];
       numeroLotes =  redondear(procesos.value/5);
       numeroProcesos = procesos.value;
       menuProceso();
    }else{
        alert('Debe ser un numero mayor a 0');
    }
}

function menuProceso( id= '', nombre = '', tiempo = '', operador = ''){
    const divProceso = document.querySelector('#proceso');

    if(numeroProcesos > 0){
        divProceso.innerHTML = `
        <p>Lotes: ${numeroLotes}<p>
        <form class="formulario">
            <fieldset>
                <legend> Proceso </legend>
                <p> Procesos: ${numeroProcesos}</p>
```

```
                <label for="IdProceso">ID</label>
                <input type="num" id="IdProceso" value ="${id}">

                <label for="nombre">Nombre</label>
                <input type="text" id="nombre" value ="${nombre}">

                <label for="tiempo">Tiempo Estimado</label>
                <input type="num" id="tiempo" value ="${tiempo}">

                <label for="operador">Operador</label>
                <input type="text" id="operador"  value ="${operador}">
            </fieldset>
            <input class="boton" type="submit" id="nuevoproceso"
value="Ingresar">
        </form>
        `;

        const botonIngresar = document.querySelector('#nuevoproceso');

        botonIngresar.addEventListener('click', ()=>{
            ingresarProceso();
        });
    }else{
        divProceso.innerHTML = ``;
        ejecutar();

    }
}

async function ejecutar(){
    const tablaProceso = document.querySelector('#loteProceso');
    dividirLotes();
    Nlote = 0;
    //Lotes
    for(const lote of lotes){
        Nlote++;1
        actualizarTabla(lote);
        for(const proceso of lote){
            procesoEjecucion(proceso);
            let tr = proceso.tiempoEstimado;
            for(let i = 0; i<proceso.tiempoEstimado; i++){
                const tiempos = document.querySelectorAll('#tiempo');
                const tiempoTotal =
document.querySelector('#tiempoTotal');
                tiempos[0].innerHTML = `Tiempo Trascurrido:
<span>${i}</span>`;
                tiempos[1].innerHTML = `Tiempo Restante: <span>${--
tr}</span>`;
```

```
                tiempoTotal.innerHTML = `Tiempo Total de Simulacion:
<span> ${++numeroProcesos} </span>segundos`;
                await sleep(1000);

            }
            ProcesosTerminados(proceso);
        }
        tablaProceso.innerHTML = ``;
    }
}

function actualizarTabla(lote){
    const tablaProceso = document.querySelector('#loteProceso');
    for(const proceso of lote){
        row = createTr();
        row.innerHTML = `
            <td>${Nlote}</td>
            <td>${proceso.id}</td>
            <td>${proceso.tiempoEstimado}</td>
        `;
        tablaProceso.appendChild(row);
    }
}

function ProcesosTerminados(proceso){
    const tbody  = document.querySelector('#procesoTerminado');
    const row = createTr();
    row.innerHTML = `
        <td>${Nlote}</td>
        <td>${proceso.id}</td>
        <td>${proceso.operacion}</td>
        <td>${proceso.resultado}</td>
    `;
    tbody.appendChild(row);
}

function createTr(){
    const tr = document.createElement('TR');
    return tr;
}

function procesoEjecucion(proceso){
    divProceso = document.querySelector('#procesoEjecucion');
     divProceso.innerHTML = `
        <p>ID: <span>${proceso.id}</span></p>
        <p>Nombre: <span>${proceso.nombre}</span></p>
        <p>Operacion: <span>${proceso.operacion}</span></p>
        <p>Tiempo Max. Estimado:
<span>${proceso.tiempoEstimado}</span></p>
```

```
        <p id = "tiempo"></p>
        <p id = "tiempo"></p>
    `;
}

function ingresarProceso(){

    const id = document.querySelector('#IdProceso');
    const nombre = document.querySelector('#nombre');
    const tiempo = document.querySelector('#tiempo');
    const operador = document.querySelector('#operador');

    let errores = validar(id.value, nombre.value, tiempo.value,
operador.value);

    if(isEmpty(errores)){
        const proceso = new Proceso(id.value, nombre.value ,
tiempo.value, operador.value);
        lote.push(proceso);
        alert('Proceso Agregado');
        proceso.RealizarOperacion();
        --numeroProcesos
        menuProceso();
    }
    else{
        const div = document.querySelector('#errores');
        errores.forEach(error=> {
            console.log(error);
            const parrafo = document.createElement('P');
            parrafo.innerHTML = `${error}`;
            div.appendChild(parrafo);
        });
        menuProceso(id.value, nombre.value, tiempo.value,
operador.value);
    }
}

function dividirLotes(){
    for (let i = 0; i < lote.length; i+=5){
        let pedazo = lote.slice(i, i + 5);
        lotes.push(pedazo);
    }
}

function validarNumero(numero){
    return numero > 0 ? true : false;
}

function validarOperador( operacion ){
```

```
    const regex = new RegExp('^[0-9]+[+-/%\*x]{1,1}[0-9]+$');
    return regex.test(operacion); // true o false
}

function IsValidDivision(operacion){ //2/2
    const operador = operacion.match('[+-/%\*x]{1,1}')
    if( operador[0] ==='/' || operador [0]==='%' ){
        const operando = validarOperando(operacion, operador);
        if(operando === '0'){
           return false;
        }
    }
    return true
}


function validarOperando (operacion, operando){ // 2342342 / 0
    const operandos =  operacion.split(operando);
    return operandos[1];
}

function validarCadena(cadena){
    return cadena === '';
}

function validarID(id){
    for(const process of lote){
        if(process.id === id){
            return true;
        }
    }
    return false;
}

function validar(id, nombre, numero, operacion){
    const div = document.querySelector('#errores');
    div.innerHTML = ``;
    let errores = []

    if(validarCadena(id)){
        errores.push('El ID es obligatorio');
    }
    if(validarID(id)){
        errores.push('El ID: '+ id + ' ya fue registrado, ingrese uno
nuevo');
    }
    if(validarCadena(nombre)){
        errores.push('El nombre es obligatorio.');
    }
```

```
    if(!validarNumero(numero)){
        errores.push('El tiempo de espera debe ser mayor a 0.');
    }
    if(!validarOperador(operacion)){
        errores.push('El formato del operador no es valido');
    }else{
        if(!IsValidDivision(operacion)) {
            errores.push('El formato de Division o Residuo no es
valido.');
        }
    }

    return errores;
}

function isEmpty(array){
    return array.length === 0;
}

function redondear(lotes){
    if(lotes <=1){
        return 1;
    }
    if(lotes >1 && lotes <=2){
        return 2
    }
    if(lotes >2 && lotes <=3){
        return 3
    }
    if(lotes >3 && lotes <=4){
        return 4
    }
}

const sleep = (milliseconds) => {
    return new Promise(resolve => setTimeout(resolve, milliseconds))
}
```

Proceso.js:

```
//LOPEZ ARELLANO RICARDO DAVID

// Declaramos la clase
function Proceso(id, nombre, tiempoEstimado, operacion){
    //Operacion resultado operacion y resultado

    this.resultado;
    this.operador;
```

```
    this.operandos = [];

    //Atributos
    this.lote = lote;
    this.id = id;
    this.nombre = nombre;
    this.tiempoEstimado = tiempoEstimado;
    this.operacion = operacion;


    this.ToString = function(){
        return `Datos del Proceso: ${this.id}, ${this.nombre},
${this.tiempoEstimado}, ${this.operacion} = ${this.resultado}`;
    }

    this.ObtenerOperador = function(){
        this.operador = this.operacion.match('[+-/%\*x]{1,1}');
    }

    this.ObtenerOperandos = function(){
        const operando =  this.operacion.split(this.operador);
        for(let i=0 ; i<2; i++){
            this.operandos[i] = parseInt(operando[i]);
        }
    }

    this.RealizarOperacion =  function(){
        this.ObtenerOperador();
        this.ObtenerOperandos();

        switch(this.operador[0]){
            case '+':
                    this.resultado = this.operandos[0] +
this.operandos[1];
                break;
            case '-':
                    this.resultado = this.operandos[0] -
this.operandos[1];
                break;
            case '/':
                try{
                    this.resultado = this.operandos[0] /
this.operandos[1];
                }catch(error){
                    alert('Agregaste una division entre 0')
                    console.log(error);
                }
                break;
            case '*':
```

```
                    this.resultado = this.operandos[0] *
this.operandos[1];
                break;
            case 'x':
                    this.resultado = this.operandos[0] **
this.operandos[1];
                break;
            case '%':
                    this.resultado = this.operandos[0] %
this.operandos[1];
                break;
        }
    }
}
```

.css:

```
/*LOPEZ ARELLANO RICARDO DAVID*/

 html {
    line-height: 1.15; /* 1 */
    -webkit-text-size-adjust: 100%; /* 2 */
  }

  body {
    margin: 0;
  }

  main {
    display: block;
  }

    h1 {
    font-size: 2em;
    margin: 0.67em 0;
  }

  hr {
    box-sizing: content-box; /* 1 */
    height: 0; /* 1 */
    overflow: visible; /* 2 */
  }

    pre {
    font-family: monospace, monospace; /* 1 */
    font-size: 1em; /* 2 */
  }
```

```
a {
  background-color: transparent;
}

abbr[title] {
  border-bottom: none; /* 1 */
  text-decoration: underline; /* 2 */
  text-decoration: underline dotted; /* 2 */
}

b,
strong {
  font-weight: bolder;
}

code,
kbd,
samp {
  font-family: monospace, monospace; /* 1 */
  font-size: 1em; /* 2 */
}

small {
  font-size: 80%;
}

sub,
sup {
  font-size: 75%;
  line-height: 0;
  position: relative;
  vertical-align: baseline;
}

sub {
  bottom: -0.25em;
}

sup {
  top: -0.5em;
}

img {
  border-style: none;
}

button,
input,
optgroup,
```

```
select,
textarea {
  font-family: inherit; /* 1 */
  font-size: 100%; /* 1 */
  line-height: 1.15; /* 1 */
  margin: 0; /* 2 */
}

button,
input { /* 1 */
  overflow: visible;
}

button,
select { /* 1 */
  text-transform: none;
}

button,
[type="button"],
[type="reset"],
[type="submit"] {
  -webkit-appearance: button;
}

button::-moz-focus-inner,
[type="button"]::-moz-focus-inner,
[type="reset"]::-moz-focus-inner,
[type="submit"]::-moz-focus-inner {
  border-style: none;
  padding: 0;
}

button:-moz-focusring,
[type="button"]:-moz-focusring,
[type="reset"]:-moz-focusring,
[type="submit"]:-moz-focusring {
  outline: 1px dotted ButtonText;
}

fieldset {
  padding: 0.35em 0.75em 0.625em;
}

legend {
  box-sizing: border-box; /* 1 */
  color: inherit; /* 2 */
  display: table; /* 1 */
  max-width: 100%; /* 1 */
```

```
  padding: 0; /* 3 */
  white-space: normal; /* 1 */
}

progress {
  vertical-align: baseline;
}

textarea {
  overflow: auto;
}

[type="checkbox"],
[type="radio"] {
  box-sizing: border-box; /* 1 */
  padding: 0; /* 2 */
}

[type="number"]::-webkit-inner-spin-button,
[type="number"]::-webkit-outer-spin-button {
  height: auto;
}

[type="search"] {
  -webkit-appearance: textfield; /* 1 */
  outline-offset: -2px; /* 2 */
}

[type="search"]::-webkit-search-decoration {
  -webkit-appearance: none;
}

::-webkit-file-upload-button {
  -webkit-appearance: button; /* 1 */
  font: inherit; /* 2 */
}

details {
  display: block;
}

summary {
  display: list-item;
}

template {
  display: none;
}
```

```
  [hidden] {
    display: none;
  }
```

Styles.css:

```
/*LOPEZ ARELLANO RICARDO DAVID*/

html{
    line-height:1.15;
    -webkit-text-size-adjust:100%
}

header{
    background-color: rgb(29, 29, 29);
    padding: 1rem 0 3rem 0;
    text-align: center;
}

body{
    margin: 0;
    font-family: 'Lato', sans-serif;
    background-color: rgb(147, 232, 253);
}

h1, h2, p, span{
    text-align: center;
    color: rgb(0, 0, 0);
}

span{
    font-weight: 900;
}

.Nproceso{
    width: 95%;
    max-width: 120rem;
    margin: 0 auto;
}

.formulario{
    color: rgb(0, 0, 0);
}

.formulario legend{
    font-size: 1.2rem;
    font-weight: 700;
```

```css
}

.formulario label{
    margin-top: 1rem;
    font-weight: 700;
    text-transform: uppercase;
    display: block;
}

.formulario input:not([type="submit"]){
    padding: .5rem;
    display: block;
    width: 99%;
    background-color: white;
    border: 2px solid black;
    border-radius: .5rem;
}

.boton{
    background-color: rgb(0, 0, 0);
    color: white;
    text-decoration: none;
    font-weight: 700;
    padding: 1rem 3rem;
    text-align: center;
    margin-top: .5rem;
    display: inline-block;
    border: none;
    cursor: pointer;
}

.boton:hover{
    background-color: rgba(0, 0, 0, 0.87);
}

.contenedor{
    display:grid;
    gap: 3rem;
    grid-template-columns: repeat(3, 1fr);
}

.tabla {
    width: 80%;
    border-spacing: 0;
    margin: 0 auto;
    color: rgb(0, 0, 0);
}

.tabla thead{
```

```
    background-color: rgba(0, 0, 0, 0.87);
}

.tabla td{
    margin-left: 1rem;
    text-align: center;
}

.ttotal{

    margin-top: 2rem;
    font-size: 1.2rem;
    text-align: center;
}

.alerta{
    background-color: rgb(189, 28, 28);
}
footer{
    padding: 1rem 0 3rem 0;
    background-color: rgb(147, 232, 253);
}
```

CONCLUSIÓN:

En conclusión a esta práctica puedo decir que no fue tan complicada ya que ya tenia unos conocimientos anteriores de dicha práctica porque estoy repitiendo la materia, pero si no tuviera dichos conocimientos creo que si me hubiera resultado un poco más difícil.

Lo que podría ser lo más complicado de la práctica es hacer lo de los contadores ya que todo el tiempo este uno detrás del otro consecutivamente sin que paren, de ahí en más es como hacer una calculadora que te va midiendo el tiempo en que ejecuta sus acciones.