

Carrera:

Ingeniería en Software

Asignatura:

Lenguajes y autómatas

Proyecto 3. Máquina de Turing

Alumnos:

- Brandon Iván Márquez Morales
- Luis Gustavo Jaime Esquivel
- Brandon Javier Rodríguez Arellano

Profesor:

Dr. Sergio Valadez Godínez

Fecha de entrega: Jueves 08 de Diciembre del 2022.

Instrucciones:

A partir de un caso práctico de lenguajes y autómatas, elaborar un reporte que contenga:

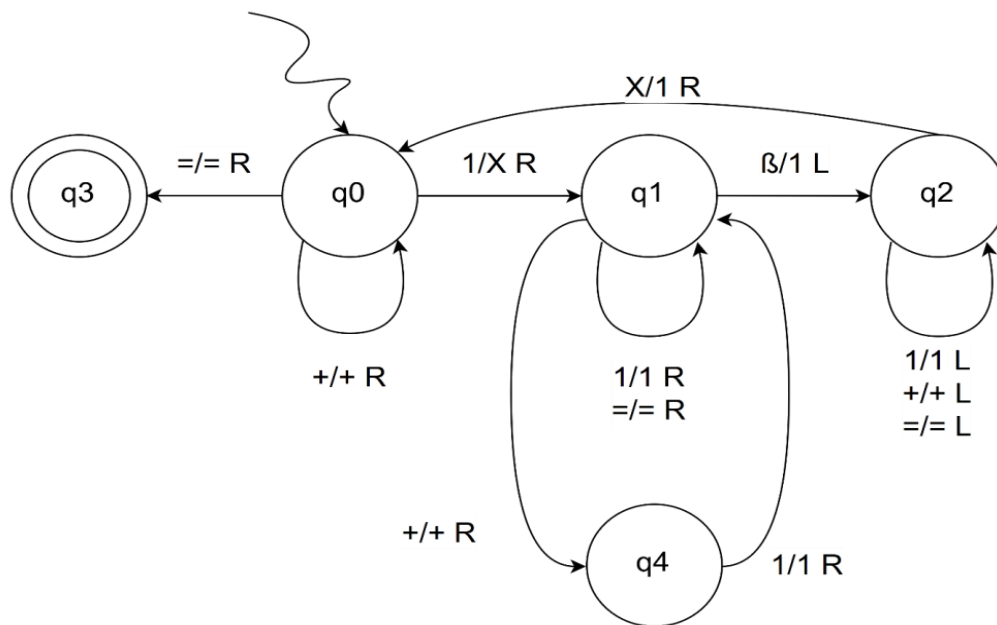
- Descripción del problema a resolver
- Diseño de la Máquina de Turing
- Implementación
- Resultados
- Conclusiones
- Referencias bibliográficas

a) Descripción del problema a resolver

En el presente documento se dará a conocer un simulador en una implementación de JavaScript en base a la máquina de Turing que permite visualizar el funcionamiento de la máquina, que trata de la suma de N números unarios.

b) Diseño de la Máquina de Turing

Para el desarrollo de este autómata se utilizó la herramienta de JFLAP donde se crearon 5 estados con sus respectivas transiciones y funciones: (gobetech, 2022)



(Rodger, 2022)

A continuación, se muestran las respectivas tablas de este autómata donde se muestra sus estados y los símbolos que se utilizaron para esta función de suma unaria:

M=	$(Q, \Sigma, \Gamma, \delta, q_0, \beta, F)$
Q=	$\{q_0, q_1, q_2, q_3, q_4\}$
Σ=	$\{1, +, =\}$
Γ=	$\{1, +, =, X, \beta\}$
F=	q_3

Estados	Símbolos				
	1	+	=	X	β
q0	q1 X R	q0 + R	q3 = R		
q1	q1 1 R	q4 + R	q1 = R		q2 1 L
q2	q2 1 L	q2 + L	q2 = L	q0 1 R	
*q3					
q4	q1 1 R				

c) Implementación

Para la implementación de este autómata como ya se mencionó este se desarrolló en el lenguaje de JavaScript. (Noah, 2022)

Principalmente se desarrolló lo que es el HTML que es la pantalla inicial donde interactúa el usuario:

```

SumadorUnario.html > html > body > div.sticky-top.bg-white > div.d-flex.justify-content-center.mt-1.mb-5
1  <html>
2  <head>
3    <meta charset="utf-8"/>
4    <title>Sumador Unario</title>
5    <script type="text/javascript" src="scripts/jquery.js"></script>
6    <script type="text/javascript" src="scripts/bootstrap.js"></script>
7    <script type="text/javascript" src="scripts/turing.js"></script>
8    <script type="text/javascript" src="scripts/ui.js"></script>
9    <script type="text/javascript" src="scripts/modal.js"></script>
10   <link rel="stylesheet" type="text/css" href="css/bootstrap.css">
11   <link rel="stylesheet" type="text/css" href="css/modal.css">
12   <link rel="stylesheet" type="text/css" href="css/fontawesome.css">
13 </head>
14 <body>
15
16   <div class="sticky-top bg-white">
17     <h1 class="text-center">
18       <strong id="etiqueta">Bienvenidos al sumador unario</strong>
19     </h1>
20     <div class="d-flex justify-content-center mt-1 mb-5">
21       <div class="row">
22         <div class="col">
23           <ul class="nav nav-tabs small" role="tablist">
24             <li></li>
25           </ul>
26         </div>
27       </div>
28     </div>
29     <center>
30       <div class="d-block">

```

```

<center>
  <div class="d-block">
    <div class="col-6 my-3">
      <div class="input-group">
        <input type="text" class="form-control" placeholder="Ingrese la cadena a evaluar ejemplo: '1+1='" id="cadena">
        <span class="input-group-btn">
          <button class="btn btn-info" type="button" id="cargar" data-toggle="popover1" title="Error de entrada" data-content="Error de entrada" data-bbox="198 201 548 219">Cargar</button>
          <button class="btn btn-secondary" id="info"><strong>?</strong></button>
        </span>
      </div>
    </div>
    <div class="col-3 my-4">
      <p class="lead">Contador: <span id="contador">0</span> | Estado: <span id="estado">0</span></p>

      <div class="slidecontainer">
        Velocidad:<br>
        <input type="range" min="0" max="1000" value="500" step="100" class="slider" id="rango"></p>
      </div>

      <div class="btn-group" role="group" aria-label="Funciones">
        <button class="btn btn-danger" id="evaluar" content="1">Evaluar</button>
        <button class="btn btn-warning" id="evaluar" content="1" onclick="Stop();">Detener</button>
      </div>
    </div>
  </div>
</center>
</div>
  
```

Después se tiene lo que es el código de la funcionalidad de esta máquina:

```

Ver  Ir  Ejecutar  Terminal  Ayuda  turing.js - Proyecto 3. Máquina de Turing - Visual
JS turing.js  JS ui.js  SumadorUnario.html  # bootstrap.css 4
scripts > JS turing.js > ...
1  ${(function() {
2
3    $('#cargar').click(function(){
4      id = ID;
5      if($('#cadena').val() == ""){
6        $(this).popover('show');
7        return false;
8      }else{
9        $(this).popover('hide');
10       LoadString(id)
11       Clear();
12       return true;
13     }
14   });
15
16   $('#evaluar').click(
17     function(){
18       id = ID;
19       time = 1000 - $('#rango').val();
20       Tick = setInterval(function(){Evaluate(id)}, time);
21     }
22   );
23
24   });
25 });
26
27
28 function AddBlank(table){
29   for(i = 0; i < 100; i++){
30     AddRow(BLANK, "", "", table);
  
```

```

> JS turing.js > ...
function AddBlank(table){
  for(i = 0; i < 100; i++){
    AddRow(BLANK, "", "", table);
  }
  return true;
}

function LoadString(table){
  var cadena = $('#cadena').val();
  ClearRows(table);
  AddRow(cadena.charAt(0), "active-row", "bg-info text-light", table);
  for(i = 1; i < cadena.length; i++){
    AddRow(cadena.charAt(i), "", "", table);
  }
  AddBlank(table);
  return true;
}

function EvaluateStep(symbol, id){
  Machines[id].Count++;
  symbols = Machines[id][Machines[id].State]

  if(symbols[symbol] == undefined) return { Error : true, Acceptable : Machines[id].Functions.Acceptable(Machines[i

  NextValues = symbols[symbol];
  Machines[id].State = NextValues[1];
  return { Error : false, Acceptable: Machines[id].Functions.Acceptable(Machines[id].State), Output : NextValues[0]
}

function Evaluate(id){
  chain = $("#tabla" + id + " td");
  i = Machines[id].i;

  if(!Machines[id].Functions.Acceptable(Machines[id].State) && chain.length > i && Machines[id].Count < 10000){
    time = 1000 - $("#rango").val();
    result = EvaluateStep(chain[i].textContent.trim(), id);
    if(result.Error){
      Update();
      Stop();
      $('#modalTitle').html('¡Error!');
      $('#modalText').html(result.Message);
      $('#myModal').modal('show');
    }else{
      if (i >= chain.length - 2) AddBlank(id);
      chain[i].textContent = (result.Output);
      i += result.Movement;
      Machines[id].i += result.Movement;
      chain[i].setAttribute('id', 'new-row');
      ChangeActiveRow(id, time);
      Update();
    }
  }
}

```

```

Machines[id].i += result.Movement;
chain[i].setAttribute('id', 'new-row');
ChangeActiveRow(id, time);
Update();
}

}else{
    if(Machines[id].Functions.Acceptable(Machines[id].State)){
        Update();
        Stop();
        $('#modalTitle').html('Cadena válida.');
```

\$('#modalText').html('La cadena ingresada es válida. Puede ver el resultado en la cinta.');

\$('#myModal').modal('show');

}

if(Machines[id].Count >= 10000){

Update();

Stop();

\$('#modalTitle').html('¡Error!');

\$('#modalText').html('La cadena ingresada ha generado muchas transiciones sin definir un resultado.');

\$('#myModal').modal('show');

}

}

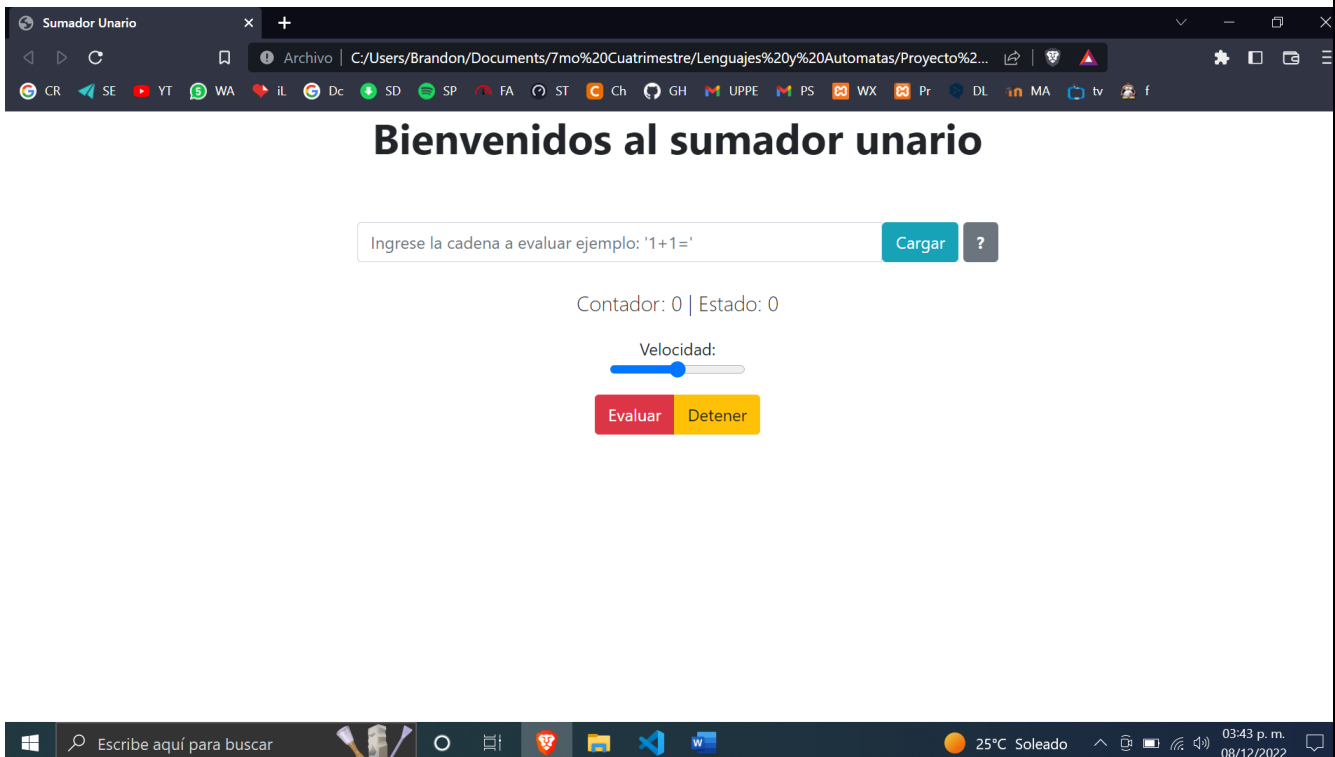
function Stop(){

clearInterval(Tick);

}

d) Resultados

Una vez estando en el archivo de HTML que es la siguiente pantalla:



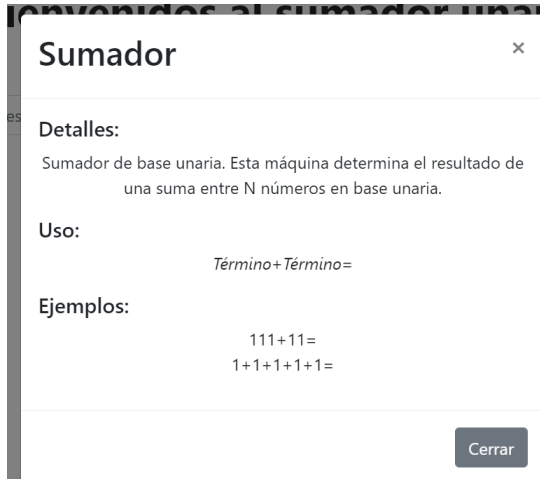
Debemos de ingresar una expresión el cuál sea válida para poder hacer la ejecución como se menciona en la caja de texto ejemplo (1+1=):

Ingrese la cadena a evaluar ejemplo: '1+1='

Cargar

?

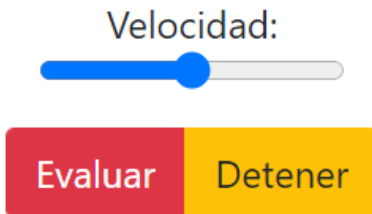
De igual manera al dar clic en el icono de signo de interrogación podemos ver la siguiente información:



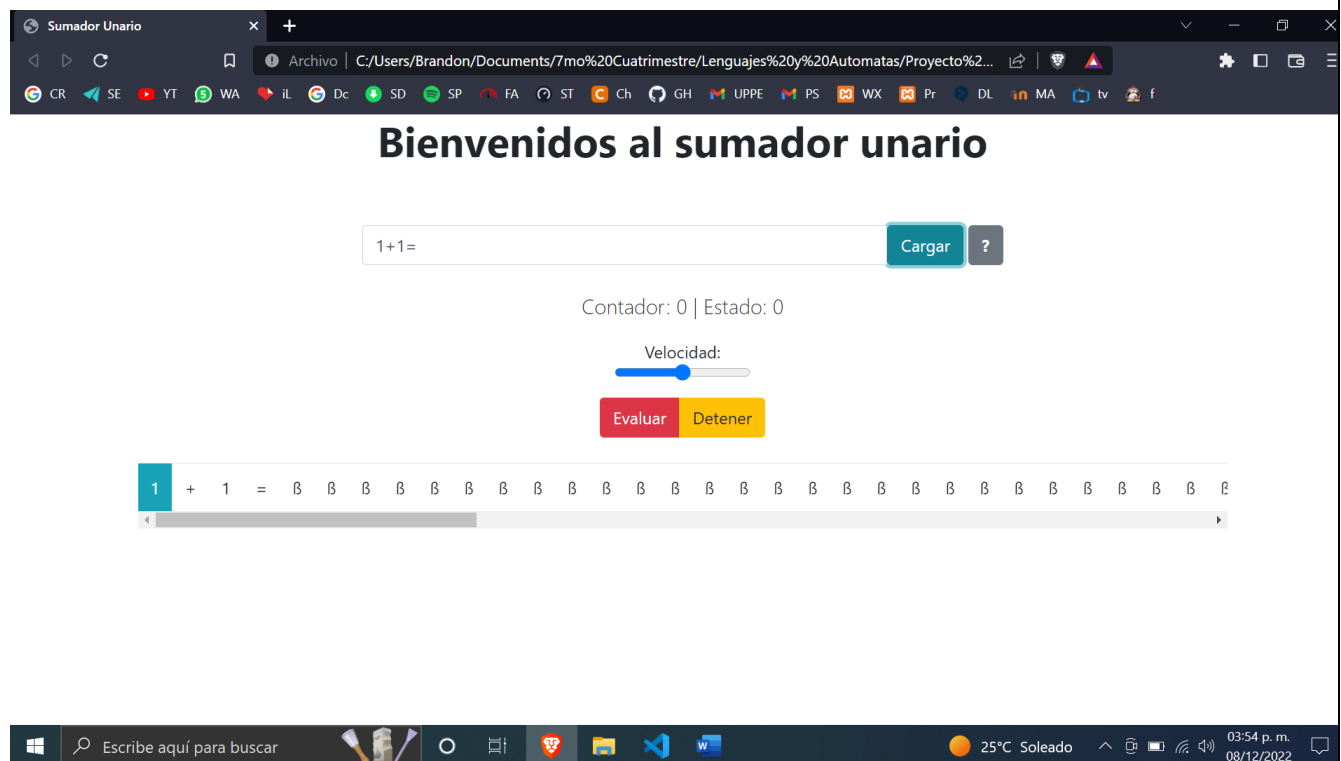
También contamos con un contador que indica el número de vueltas que van dando a través del recorrido, Además de un identificador de estados el cuál nos indica en que estado nos encontramos a través de cada transición:

Contador: 0 | Estado: 0

Si se ingresa una expresión larga el proceso puede ser tardado así que para ello se implementó un velocímetro para avanzar más rápido el proceso. Si estamos en ejecución y queremos aumentar a velocidad tenemos que tomar en cuenta que antes debemos de presionar el botón de detener, después aumentamos la velocidad que queramos y después volvemos a evaluar:



Ahora se mostrará un pequeño ejemplo para demostrar esta funcionalidad. Primero ingresamos la expresión y damos clic en cargar para que nos muestre la cinta de ejecución e ir viendo los tiempos:



[illegible][illegible]

X + 1 = β

X + 1 = β

X + 1 = 6

X + 1 = 1 β β β β β β β β β β β β β β β β β β β

[illegible]

Tiempo 7:

[illegible]

Tiempo 8:

X + 1 = 1 β

Tiempo 9:

$1 + 1 = 1$

Tiempo 10:

$$1 + \frac{1}{n} = \frac{n+1}{n}$$

Tiempo 11:

[illegible]

Tiempo 12:

$$1 + X = 1$$

Tiempo 13:

1 + X = 1 1 B

[illegible]

1 + 1 = 1 1 β

[illegible]

En conclusión, como equipo creemos fue gran importancia esta materia ya que los lenguajes y autómatas son de gran importancia en las actividades sociales, ya que todos estos tipos de autómatas se encuentran dentro de todo tipo de maquinaria y aplicaciones que conocemos hoy en día, donde estas tienen funcionalidades específicas con las que se realizan sus respectivas operaciones y actividades.

f) Referencias bibliográficas

REFERENCIAS

gobetech. (2022). *obetech*. Obtenido de obetech: <https://tech.gobetech.com/49408/como-disenar-una-maquina-de-turing-que-accepte-una-cadena-de-longitud-impar.html>

Noah, B. (2022). *Vintage Kitchen*. Obtenido de Vintage Kitchen: <https://vintage-kitchen.com/es/guide/what-is-unary-addition/>

Rodger, S. H. (2022). *jflap*. Obtenido de jflap: <https://www.jflap.org/jflaptmp/>