



TECNOLÓGICO  
NACIONAL DE MÉXICO



**TECNOLÓGICO NACIONAL DE MÉXICO  
INSTITUTO TECNOLÓGICO DE LA LAGUNA**



**Inteligencia Artificial**

**UNIDAD 4: Lógica difusa**

**Documentación del Proyecto de lógica difusa.**

**DOCENTE: LAMIA HAMDÁN M.**

NÚMERO DE CONTROL	NOMBRE
17130765	Enrique Antonio Belmarez Meraz
17130804	Luis Emmanuel Mendez Barrios
17130786	José Ángel García Arce
17130049	José Baltazar Martínez De La Rosa
17130028	Patricia García Almanza

**Torreón, Coahuila**

06/06/ 2021



## **TABLA DE CONTENIDO**

<b>INTRODUCCIÓN</b>	<b>3</b>
<b>METODOLOGÍA</b>	<b>3</b>
<b>DESARROLLO DEL TEMA</b>	<b>6</b>
<b>CONCLUSIONES</b>	<b>24</b>
<b>REFERENCIAS</b>	<b>25</b>



## 1. Introducción

La realización de este proyecto tiene como objetivo utilizar la lógica difusa para la detección del COVID en un sencillo formulario, observando cómo se trabajó con esta metodología ya que nos permite analizar la información del mundo real y así obteniendo como respuesta si toda información es verdadera o falsa.

## 2. Metodología

Para desarrollar este proyecto se trabajó mediante la metodología de lógica difusa: la lógica difusa se utiliza cuando la complejidad del proceso en cuestión es muy alta y no existen modelos matemáticos precisos, para procesos altamente no lineales y cuando se envuelven definiciones y conocimiento no estrictamente definido (impreciso o subjetivo).

Se basa en reglas heurísticas de la forma SI (antecedente) ENTONCES (consecuente), donde el antecedente y el consecuente son también conjuntos difusos, ya sea puros o resultado de operar con ellos. Sirvan como ejemplos de regla heurística para esta lógica (nótese la importancia de las palabras «muchísimo», «drásticamente», «un poco» y «levemente» para la lógica difusa):

- SI hace muchísimo frío. ENTONCES descendió drásticamente la temperatura.
- SI voy a llegar un poco tarde. ENTONCES aumento levemente la velocidad.



Imagen 1. Lógica difusa



TECNOLÓGICO  
NACIONAL DE MÉXICO



## Matlab

Combina un entorno de escritorio perfeccionado para el análisis iterativo y los procesos de diseño con un lenguaje de programación que expresa las matemáticas de matrices y arrays directamente.

Permiten ver cómo funcionan diferentes algoritmos con sus datos. Realice iteraciones hasta obtener los resultados deseados y, después, genere automáticamente un programa de MATLAB para reproducir o automatizar su trabajo.

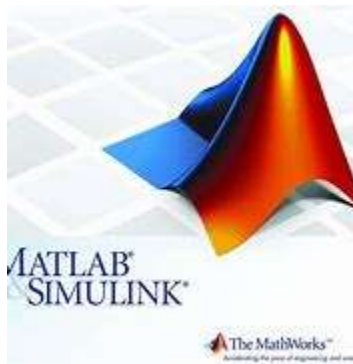


Imagen 2. Matlab



## JavaScript

Es un lenguaje de programación o de secuencias de comandos que te permite implementar funciones complejas en páginas web, cada vez que una página web hace algo más que sentarse allí y mostrar información estática para que la veas, muestra oportunas actualizaciones de contenido, mapas interactivos, animación de Gráficos 2D/3D, desplazamiento de máquinas reproductoras de vídeo, etc., puedes apostar que probablemente JavaScript está involucrado. Es la tercera capa del pastel de las tecnologías web estándar, dos de las cuales (HTML y CSS) se han cubierto con mucho más detalle en otras partes del Área de aprendizaje.

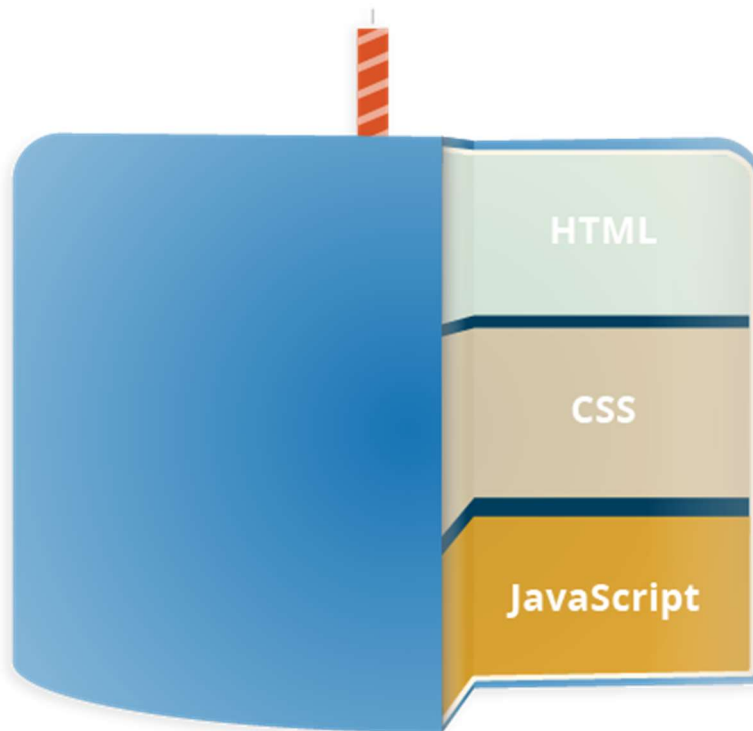


Imagen 3. JavaScript



### 3. Desarrollo del tema

Primeramente, ¿Qué es la lógica difusa?

Permite tomar decisiones más o menos intensas en función de grados intermedios de cumplimiento de una premisa; se adapta mejor al mundo real en el que vivimos, e incluso puede comprender y funcionar con nuestras expresiones, del tipo «hace mucho calor», «no es muy alto», «el ritmo del corazón está un poco acelerado», etc.

La clave de esta adaptación al lenguaje se basa en comprender los cuantificadores de cualidad para nuestras inferencias (en los ejemplos de arriba, «mucho», «muy» y «un poco»).

En la teoría de conjuntos difusos se definen también las operaciones de unión, intersección, diferencia, negación o complemento, y otras operaciones sobre conjuntos (véase también subconjunto difuso), en los que se basa esta lógica.

Para cada conjunto difuso, existe asociada una función de pertenencia para sus elementos, que indica en qué medida el elemento forma parte de ese conjunto difuso. Las formas de las funciones de pertenencia más típicas son trapezoidal, lineal y curva.

Se basa en reglas heurísticas de la forma SI (antecedente) ENTONCES (consecuente), donde el antecedente y el consecuente son también conjuntos difusos, ya sea puros o resultado de operar con ellos. Sirvan como ejemplos de regla heurística para esta lógica (nótese la importancia de las palabras «muchísimo», «drásticamente», «un poco» y «levemente» para la lógica difusa):

- SI hace muchísimo frío. ENTONCES descendió drásticamente la temperatura.
- SI voy a llegar un poco tarde. ENTONCES aumento levemente la velocidad.



Los métodos de inferencia para esta base de reglas deben ser sencillos, versátiles y eficientes. Los resultados de dichos métodos son un área final, fruto de un conjunto de áreas solapadas entre sí (cada área es resultado de una regla de inferencia). Para escoger una salida concreta a partir de tanta premisa difusa, el método más usado es el del centroide, en el que la salida final será el centro de gravedad del área total resultante.

La LDC es un modelo lógico multivalente que permite la modelación simultánea de los procesos deductivos y de toma de decisiones. El uso de la LDC en los modelos matemáticos permite utilizar conceptos relativos a la realidad siguiendo patrones de comportamiento similares al pensamiento humano. Las características más importantes de estos modelos son: La flexibilidad, la tolerancia con la imprecisión, la capacidad para moldear problemas no lineales y su fundamento en el lenguaje de sentido común. Bajo este fundamento se estudia específicamente cómo acondicionar el modelo sin condicionar la realidad.

La LDC utiliza la escala de la LD, la cual puede variar de 0 a 1 para medir el grado de verdad o falsedad de sus proposiciones, donde las proposiciones pueden expresarse mediante predicados. Un predicado es una función del universo  $X$  en el intervalo  $[0, 1]$ , y las operaciones de conjunción, disyunción, negación e implicación, se definen de modo que restringidas al dominio  $[0, 1]$  se obtenga la Lógica Booleana.



## Index.html

1.-Para iniciar con el formulario de hizo un html para la vista del mismo haciendo preguntas sobre el Covid y dando la opción de respuesta al usuario de Si o No.

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Logica Difusa</title>
</head>
<body>
  <h1>Logica Difusa</h1>
  <div>
    Temperatura <input type="text" placeholder="Ingresar Temperatura" name="temperatura" id="temperatura">
  </div>
  <br>
  <div>
    ¿Ardor en Garganta?
    <select name="selectgarganta" id="selectgarganta">
      <option value=0.3>NO</option>
      <option value=0.6>SI</option>
    </select>
  </div>
  <br>
  <div>
    ¿Perdida del Gusto?
    <select name="selectgusto" id="selectgusto" >
      <option value=0.3>NO</option>
      <option value=0.6>SI</option>
    </select>
  </div>
  <br>
  <div>
    ¿Perdida del Olfato?
    <select name="selectolfato" id="selectolfato">
```





2.- En esta parte se mandan a llamar los JavaScript en donde se encuentra la lógica difusa que serán parte de la vista del formulario.

```
<div>
  ¿Pérdida del Gusto?
  <select name="selectgusto" id="selectgusto" >
    <option value=0.3>NO</option>
    <option value=0.6>SI</option>
  </select>
</div>
<br>
<div>
  ¿Pérdida del Olfato?
  <select name="selectolfato" id="selectolfato">
    <option value=0.3>NO</option>
    <option value=0.6>SI</option>
  </select>
</div>
<br>
<input type="button" value="Enviar" id="enviar">
<div id="resultado">

</div>
<script src="logicadifusa.js" type="module"></script></script>
<script src="membresias.js" type="module"></script>
</body>
</html>
```

### logicadifusa.js

4. Este es el archivo js en donde se encuentra la lógica difusa, se declaran las constantes que obtendrán el elemento seleccionado, enviar la información y el resultado.

```
import * as f from './membresias.js'

const temperatura = document.getElementById('temperatura')
const olfato = document.getElementById('selectolfato')
const gusto = document.getElementById('selectgusto')
const garganta = document.getElementById('selectgarganta')
const boton = document.getElementById('enviar')
const h2resultado = document.getElementById('resultado')

//funciones con fuzzy
//temperatura
let fuzzTemperatura = t => {
  let tlow = f.funcionTrapezoidal(t,30,35,36,36.5)
  let tnormal = f.funcionTriangulo(t,36,36.7,37.5)
  let thigh = f.funcionTrapezoidal(t,37,37.5,41,50)
  //console.log(`fuzz temp = ${[tlow,tnormal,thigh]}`)
  return [tlow,tnormal,thigh]
}
//-----
//Olfato
let fuzzOlfato = olfato => {
  return f.funcionBool(olfato,.5)
}
//-----
//Gusto
let fuzzGusto = gusto => {
  return f.funcionBool(gusto , .5)
}
//-----
```



4.-Aquí se puede observar la tabla en donde se encuentra la matriz de relación que se elaboró para obtener el resultado del formulario para el usuario y mostrarlo después de hacer las comparaciones necesarias.

```
return [pasa , sospechoso , covid]
}
```

//creacion de la matrix de relacion (Step 3)

	baja_Temp	normal_Temp	alta_Temp	si_tiene_olfato	no_tiene_olfato
baja_Temp	null	null	null	puede_entrar	sospechoso_no_entra
normal_Temp	null	null	null	puede_entrar	sospechoso_no_entra
alta_Temp	null	null	null	sospechoso_no_entra	covid_no_entra
si_tiene_olfato	puede_entrar	puede_entrar	sospechoso_no_entra	null	
no_tiene_olfato	sospechoso_no_entra	sospechoso_no_entra	covid_no_entra	null	
si_tiene_gusto	puede_entrar	puede_entrar	sospechoso_no_entra	puede_entrar	sospechoso_no_entra
no_tiene_gusto	sospechoso_no_entra	sospechoso_no_entra	covid_no_entra	sospechoso_no_entra	covid_no_entra
no_tiene_ardor	puede_entrar	puede_entrar	sospechoso_no_entra	puede_entrar	sospechoso_no_entra
si_tiene_ardor	sospechoso_no_entra	sospechoso_no_entra	covid_no_entra	sospechoso_no_entra	covid_no_entra



5.- Aquí están las reglas de las que maneja la lógica difusa en donde si el usuario mostró algún síntoma entonces obtendrá un resultado ya sea bueno o malo dependiendo de las reglas que están escritas aquí.

```
//son 24 reglas
let reglas = (ardor , perdida_gusto , perdida_olfato , Temperatura) => {
  const limite = 0.5
  let rules = [0.1,0.1,0.1]
  let index = 0
  let baseR = (bardor , bperdida_gusto , bperdida_olfato , bTemperatura) => f.OperadorAND(f.OperadorAND(f.OperadorAND(bardor , bperdida_gusto , bperdida_olfato), bTemperatura), f.OperadorAND(bardor , bperdida_gusto , bperdida_olfato), bTemperatura)
  //reglas temp baja (>= limite = 1) (< limite = 0) temperatura[0] = baja temperatura[1] = alta temperatura
  //regla 1 pasa 000
  if((ardor < limite) && (perdida_gusto < limite) && (perdida_olfato < limite) && Temperatura[0] != 0){
    rules[0] = ( baseR(f.OperadorNOT(ardor), f.OperadorNOT(perdida_gusto), f.OperadorNOT(perdida_olfato), Temperatura[0] ) )
    index = 1
  }
  // regla 2 pasa 001
  if((ardor < limite) && (perdida_gusto < limite) && (perdida_olfato >= limite) && Temperatura[0] != 0){
    rules[0] = ( baseR(f.OperadorNOT(ardor), f.OperadorNOT(perdida_gusto), perdida_olfato, Temperatura[0] ) )
    index = 2
  }
  //regla 3 pasa 010
  if((ardor < limite) && (perdida_gusto >= limite) && (perdida_olfato < limite) && Temperatura[0] != 0){
    rules[0] = ( baseR(f.OperadorNOT(ardor), perdida_gusto, f.OperadorNOT(perdida_olfato), Temperatura[0] ) )
    index = 3
  }
  //regla 4 sospechoso 011
  if((ardor < limite) && (perdida_gusto >= limite) && (perdida_olfato >= limite) && Temperatura[0] != 0){
    rules[1] = ( baseR(f.OperadorNOT(ardor), perdida_gusto, perdida_olfato, Temperatura[0] ) )
    index = 4
  }
  //regla 5 pasa 100
  if((ardor >= limite) && (perdida_gusto < limite) && (perdida_olfato < limite) && Temperatura[0] != 0){
    rules[0] = ( baseR(ardor, f.OperadorNOT(perdida_gusto), f.OperadorNOT(perdida_olfato), Temperatura[0] ) )
    index = 5
  }
}
```

6.- Aquí siguen las reglas para poder definir el resultado que arrojará dependiendo de los síntomas seleccionados con las condiciones.

```
//regla 20 covid 011
if((ardor < limite) && (perdida_gusto >= limite) && (perdida_olfato >= limite) && Temperatura[2] != 0){
  rules[2] = ( baseR(f.OperadorNOT(ardor), perdida_gusto, perdida_olfato, Temperatura[2] ) )
  index = 20
}
//regla 21 sospechoso 100
if((ardor >= limite) && (perdida_gusto < limite) && (perdida_olfato < limite) && Temperatura[2] != 0){
  rules[1] = ( baseR(ardor, f.OperadorNOT(perdida_gusto), f.OperadorNOT(perdida_olfato), Temperatura[2] ) )
  index = 21
}
//regla 22 covid 101
if((ardor >= limite) && (perdida_gusto < limite) && (perdida_olfato >= limite) && Temperatura[2] != 0){
  rules[2] = ( baseR(ardor, f.OperadorNOT(perdida_gusto), perdida_olfato, Temperatura[2] ) )
  index = 22
}
//regla 23 covid 110
if((ardor >= limite) && (perdida_gusto >= limite) && (perdida_olfato < limite) && Temperatura[2] != 0){
  rules[2] = ( baseR(ardor, perdida_gusto, f.OperadorNOT(perdida_olfato), Temperatura[2] ) )
  index = 23
}
//regla 24 covid 111
if((ardor >= limite) && (perdida_gusto >= limite) && (perdida_olfato >= limite) && Temperatura[2] != 0){
  rules[2] = ( baseR(ardor, perdida_gusto, perdida_olfato, Temperatura[2] ) )
  index = 24
}

return [index,rules]
```



```
//funcion defuzzy centroide
let defuzzy = salida => {
  let anguloB = Math.atan(2/1) //radianes
  let anguloalfa = 1/2*Math.PI - anguloB //radianes
  let bmenorc1 = 4-salida[0]/Math.atan(anguloalfa)
  let bmenorc2 = 4-(2*salida[1]/Math.atan(anguloalfa))
  let bmenorc3 = 6-salida[2]/Math.atan(anguloalfa)
  let c1 = 1/2 * salida[0] * (bmenorc1+4)
  let c2 = 1/2 * salida[1] * (bmenorc2+4)
  let c3 = 1/2 * salida[2] * (bmenorc3+6)
  let x1 = 2
  let x2 = 5
  let x3 = 9

  return ((c1*x1)+(c2*x2)+(c3*x3))/((1/2)*salida[0]*(bmenorc1+4)+(1/2)*salida[1]*(bmenorc2+4)+(1/2)*salida[2]*(bmenorc3+6))
}

//funcion para convertir a grados
let Agrados = r => r*180/Math.PI

//funcion para convertir a radianes
let Aradianes = g => g*Math.PI/180
```

En esta parte encontramos el algoritmo de la defuzzificación en el cual decidimos utilizar el algoritmo del centroide, ya que consideramos que era el más práctico de aplicar.

Utilizándolo con la siguiente formula:

$$\frac{1}{2} * h * (base\ mayor + base\ menor) * centro\ geométrico / \frac{1}{2} * h * (base\ mayor + base\ menor + \frac{1}{2})$$

Obviamente la formula se va incrementar dependiendo del número de entradas que tengamos.

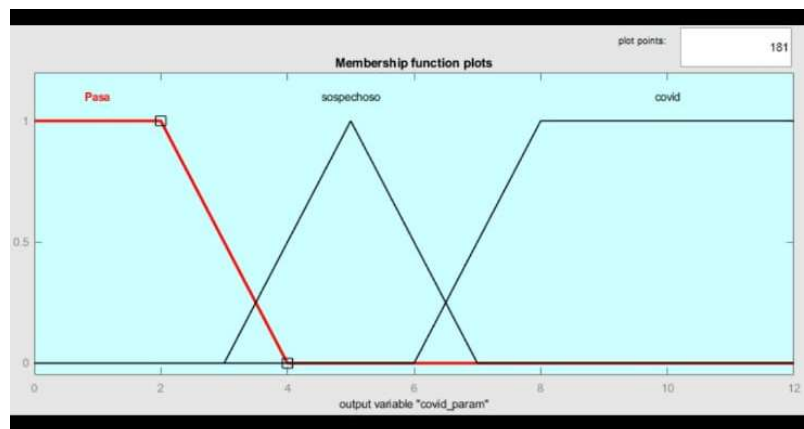


```
//funcion para convertir a grados
let Agrados = r => r*180/Math.PI

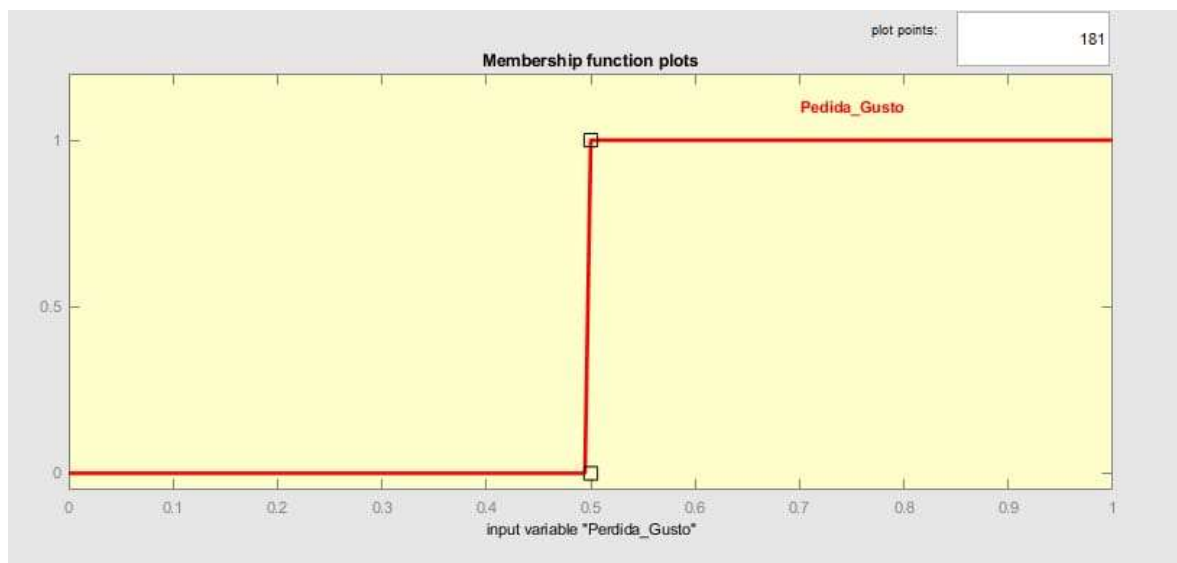
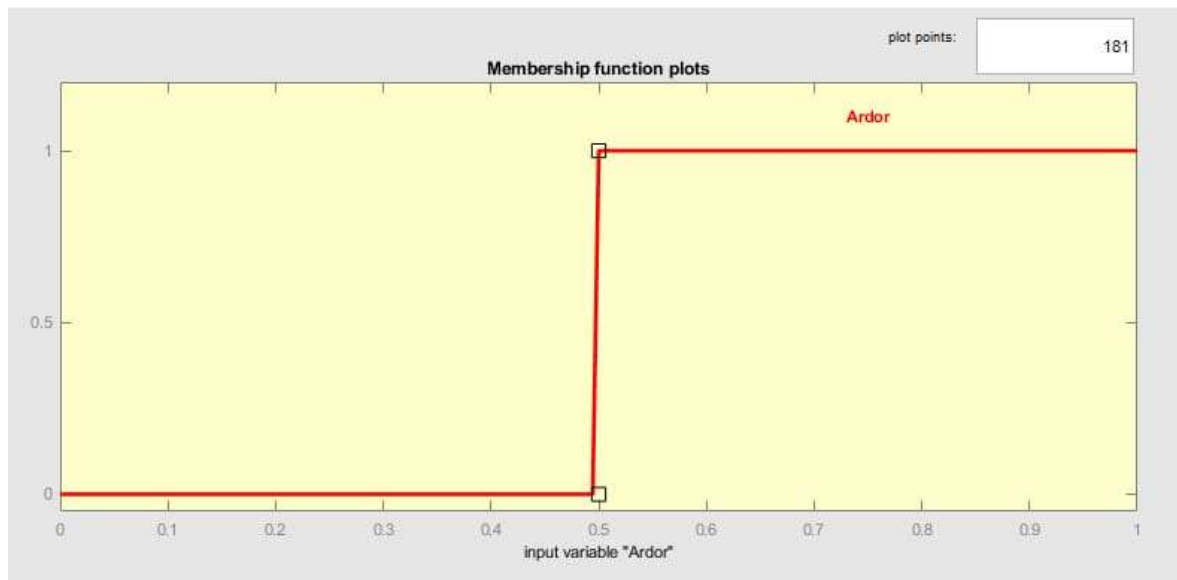
//funcion para convertir a radianes
let Aradianes = g => g*Math.PI/180

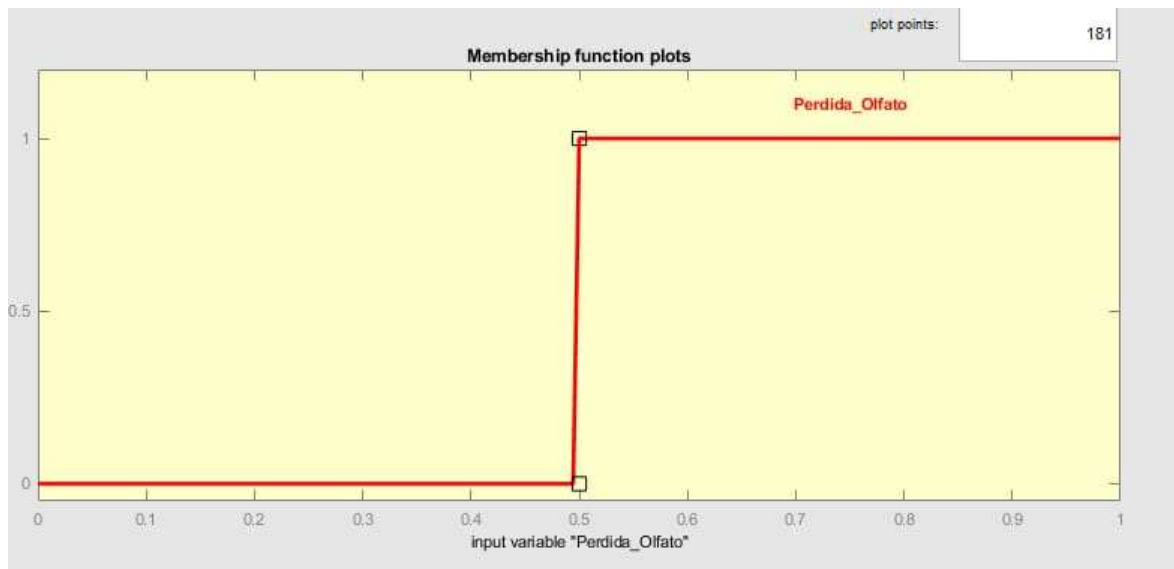
//boton enviar
boton.addEventListener("click", () => {
  //condicion para verificar que introducen una temperatura
  if(temperatura.value != ""){
    //imprimimos en consola el resultado de las
    let valores_fuzz = reglas(fuzzArdor(garganta.options[garganta.selectedIndex].value), fuzzGusto(gusto.options[gusto.selectedIndex].value))
    let valores_defuzz = defuzzy(valores_fuzz[1])
    let resultado = valores_defuzz > 0 && valores_defuzz <= 3.7 ? "Pasa" : valores_defuzz > 3.7 && valores_defuzz <= 6.7 ? "Sospechoso" : "Covid"
    console.log(`regla N° ${valores_fuzz[0]}`)
    //console.log(valores_defuzz)
    //console.log(`Su diagnostico es : ${resultado}`)
    h2resultado.innerHTML = `/*html*/<h2>Su diagnostico es : ${resultado}</h2>`
    //resultado == "pasa" ? " " : h2resultado.innerHTML += `/*html*/<p>No pasa</p>`
  }
  else{
    alert("Ingresa una temperatura ")
  }
})
})
```

7.- Por último, tenemos el evento del botón que al obtener los valores de la función fuzzy que se realizaron arriba mostrara el diagnostico basado en los condicionales y en las reglas declaradas.





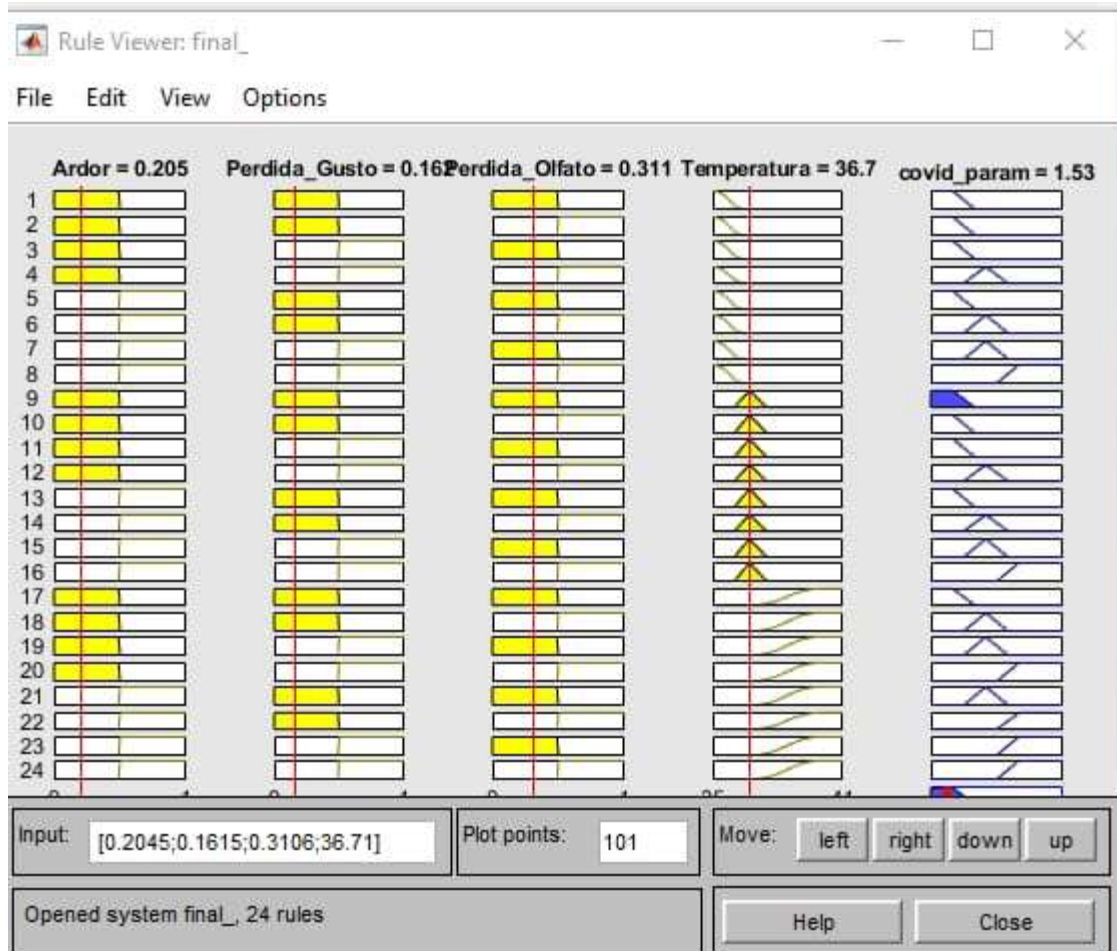




1. If (Ardor is not Ardor) and (Perdida\_Gusto is not Pedida\_Gusto) and (Perdida\_Olfato is not Perdida\_Olfato) and (Temperatura is Baja) then (covid\_param is Pasa) (1)
2. If (Ardor is not Ardor) and (Perdida\_Gusto is not Pedida\_Gusto) and (Perdida\_Olfato is Perdida\_Olfato) and (Temperatura is Baja) then (covid\_param is Pasa) (1)
3. If (Ardor is not Ardor) and (Perdida\_Gusto is Pedida\_Gusto) and (Perdida\_Olfato is not Perdida\_Olfato) and (Temperatura is Baja) then (covid\_param is Pasa) (1)
4. If (Ardor is not Ardor) and (Perdida\_Gusto is Pedida\_Gusto) and (Perdida\_Olfato is Perdida\_Olfato) and (Temperatura is Baja) then (covid\_param is sospechoso) (1)
5. If (Ardor is Ardor) and (Perdida\_Gusto is not Pedida\_Gusto) and (Perdida\_Olfato is not Perdida\_Olfato) and (Temperatura is Baja) then (covid\_param is Pasa) (1)
6. If (Ardor is Ardor) and (Perdida\_Gusto is not Pedida\_Gusto) and (Perdida\_Olfato is Perdida\_Olfato) and (Temperatura is Baja) then (covid\_param is sospechoso) (1)
7. If (Ardor is Ardor) and (Perdida\_Gusto is Pedida\_Gusto) and (Perdida\_Olfato is not Perdida\_Olfato) and (Temperatura is Baja) then (covid\_param is sospechoso) (1)
8. If (Ardor is Ardor) and (Perdida\_Gusto is Pedida\_Gusto) and (Perdida\_Olfato is Perdida\_Olfato) and (Temperatura is Baja) then (covid\_param is covid) (1)
9. If (Ardor is not Ardor) and (Perdida\_Gusto is not Pedida\_Gusto) and (Perdida\_Olfato is not Perdida\_Olfato) and (Temperatura is Normal) then (covid\_param is Pasa) (1)
10. If (Ardor is not Ardor) and (Perdida\_Gusto is not Pedida\_Gusto) and (Perdida\_Olfato is Perdida\_Olfato) and (Temperatura is Normal) then (covid\_param is Pasa) (1)
11. If (Ardor is not Ardor) and (Perdida\_Gusto is Pedida\_Gusto) and (Perdida\_Olfato is not Perdida\_Olfato) and (Temperatura is Normal) then (covid\_param is Pasa) (1)
12. If (Ardor is not Ardor) and (Perdida\_Gusto is Pedida\_Gusto) and (Perdida\_Olfato is Perdida\_Olfato) and (Temperatura is Normal) then (covid\_param is sospechoso) (1)
13. If (Ardor is Ardor) and (Perdida\_Gusto is not Pedida\_Gusto) and (Perdida\_Olfato is not Perdida\_Olfato) and (Temperatura is Normal) then (covid\_param is Pasa) (1)
14. If (Ardor is Ardor) and (Perdida\_Gusto is not Pedida\_Gusto) and (Perdida\_Olfato is Perdida\_Olfato) and (Temperatura is Normal) then (covid\_param is sospechoso) (1)
15. If (Ardor is Ardor) and (Perdida\_Gusto is Pedida\_Gusto) and (Perdida\_Olfato is not Perdida\_Olfato) and (Temperatura is Normal) then (covid\_param is sospechoso) (1)
16. If (Ardor is Ardor) and (Perdida\_Gusto is Pedida\_Gusto) and (Perdida\_Olfato is Perdida\_Olfato) and (Temperatura is Normal) then (covid\_param is covid) (1)
17. If (Ardor is not Ardor) and (Perdida\_Gusto is not Pedida\_Gusto) and (Perdida\_Olfato is not Perdida\_Olfato) and (Temperatura is Alta) then (covid\_param is Pasa) (1)
18. If (Ardor is not Ardor) and (Perdida\_Gusto is not Pedida\_Gusto) and (Perdida\_Olfato is Perdida\_Olfato) and (Temperatura is Alta) then (covid\_param is sospechoso) (1)
19. If (Ardor is not Ardor) and (Perdida\_Gusto is Pedida\_Gusto) and (Perdida\_Olfato is not Perdida\_Olfato) and (Temperatura is Alta) then (covid\_param is sospechoso) (1)
20. If (Ardor is not Ardor) and (Perdida\_Gusto is Pedida\_Gusto) and (Perdida\_Olfato is Perdida\_Olfato) and (Temperatura is Alta) then (covid\_param is covid) (1)
21. If (Ardor is Ardor) and (Perdida\_Gusto is not Pedida\_Gusto) and (Perdida\_Olfato is not Perdida\_Olfato) and (Temperatura is Alta) then (covid\_param is sospechoso) (1)
22. If (Ardor is Ardor) and (Perdida\_Gusto is not Pedida\_Gusto) and (Perdida\_Olfato is Perdida\_Olfato) and (Temperatura is Alta) then (covid\_param is covid) (1)
23. If (Ardor is Ardor) and (Perdida\_Gusto is Pedida\_Gusto) and (Perdida\_Olfato is not Perdida\_Olfato) and (Temperatura is Alta) then (covid\_param is covid) (1)
24. If (Ardor is Ardor) and (Perdida\_Gusto is Pedida\_Gusto) and (Perdida\_Olfato is Perdida\_Olfato) and (Temperatura is Alta) then (covid\_param is covid) (1)

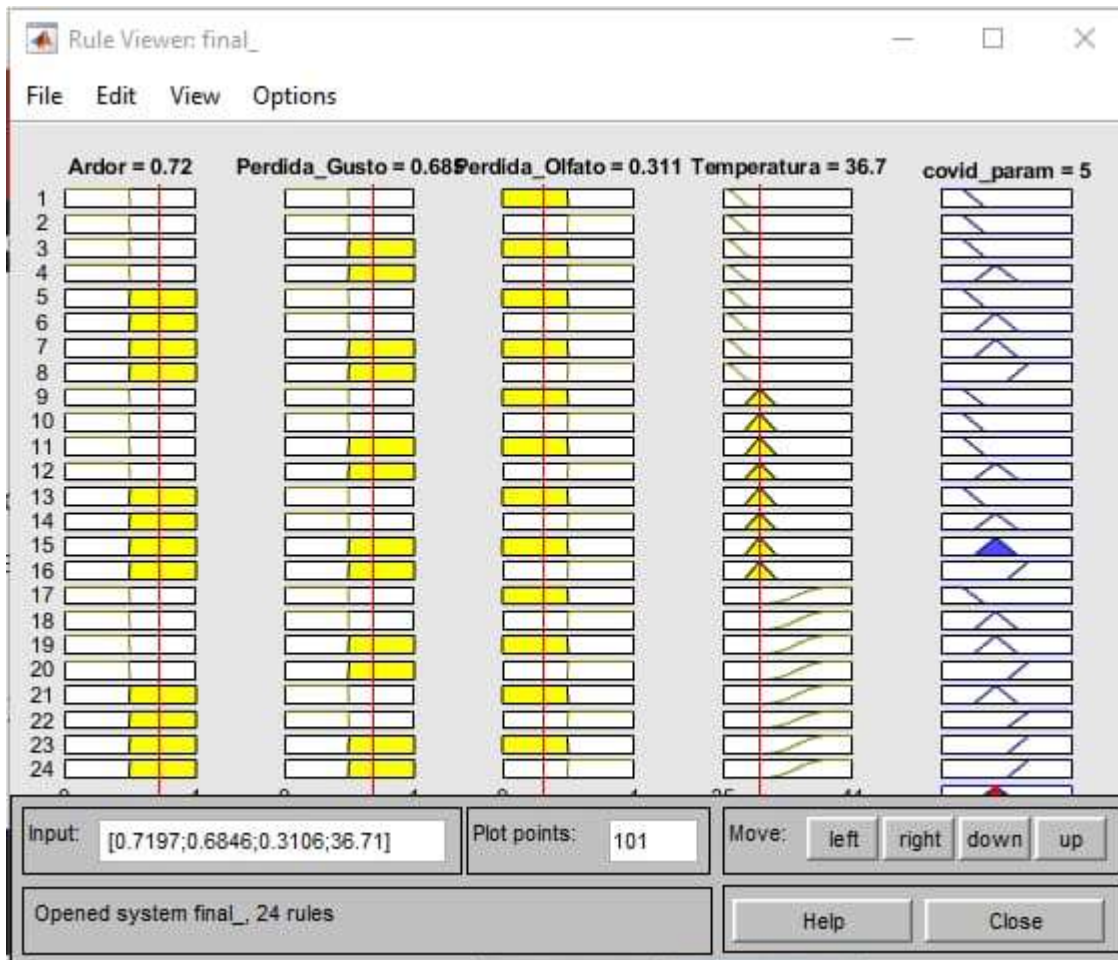


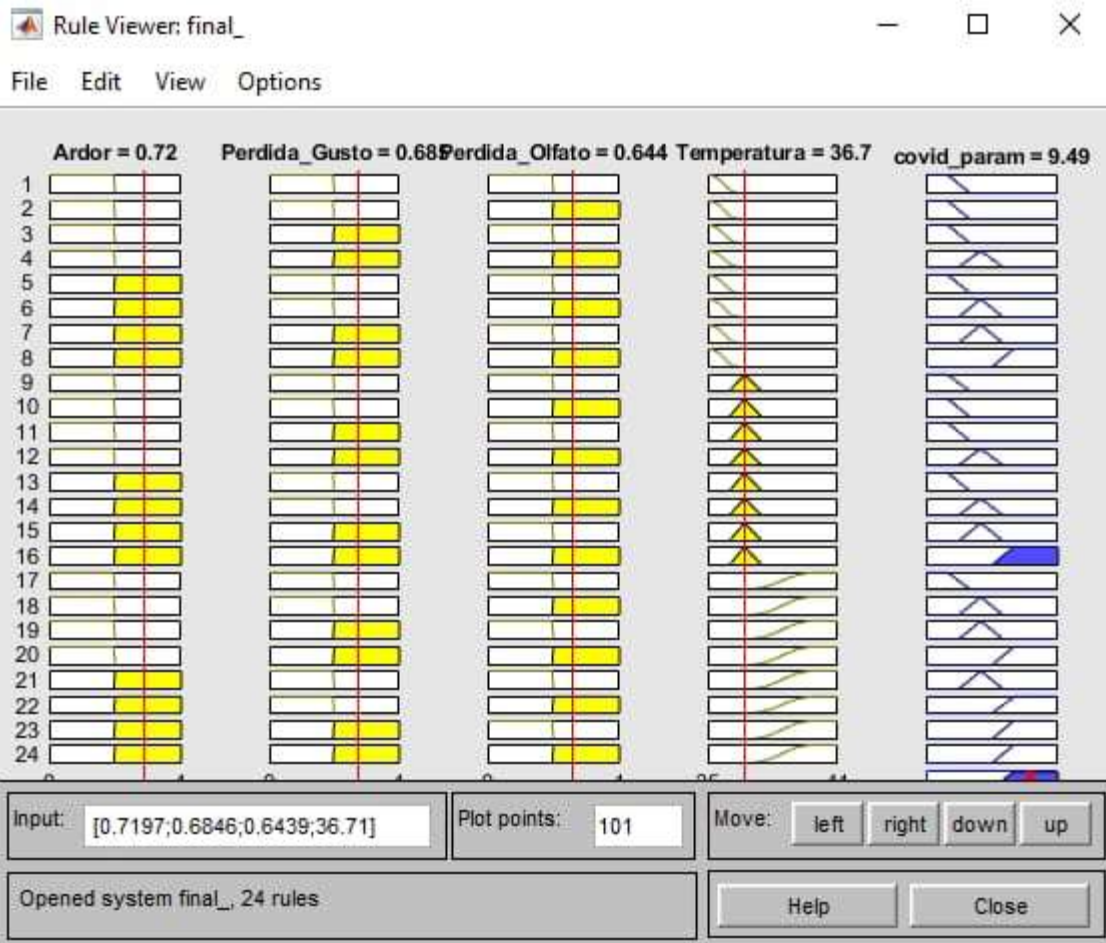
TECNOLÓGICO  
NACIONAL DE MÉXICO













## Membresías.js

```
JS membresias.js X
JS membresias.js > ...
1 //Membresias
2
3 //funcion Membresia booleana
4 let funcionBool = (x , x0) => x< x0 ? 0.0 : 1.0
5
6 let funcionInvBool = (x , x0) => x< x0 ? 1.0 : 0.0
7
8 //funcion Membresia de grado
9 let funcionGrado = (x , x0 , x1) => x <= x0 ? 0.0 : x>x0 && x<x1 ? (x/(x1-x0)) - (x0/(x1-x0)): 1.0
10
11 let funcionInvGrado = (x , x0 , x1) => x <= x0 ? 1.0 : x>x0 && x<x1 ? - (x/(x1-x0)) + (x0/(x1-x0)): 0.0
12
13 //funcion Membresia de Triangulo
14 let funcionTriangulo = (x , x0 , x1 , x2) => x <= x0 ? 0.0 : x>x0 && x<x1 ? ((x/(x1-x0)) - (x0/(x1-x0))) : x>x1 && x<x2 ?
15
16 let funcionInvTriangulo = (x , x0 , x1 , x2) => x <= x0 ? 1.0 : x>x0 && x<x1 ? - (x/(x1-x0)) + (x0/(x1-x0)) : x>x1 && x<x2
17
18
19 //funcion Membresia de Trapezoidal
20 let funcionTrapezoidal = (x , x0 , x1 , x2 , x3) => x <= x0 ? 0.0 : x>x0 && x<x1 ? (x/(x1-x0)) - (x0/(x1-x0)) : x>x1 && x<
21
22 let funcionInvTrapezoidal = (x , x0 , x1 , x2 , x3) => x <= x0 ? 1.0 : x>x0 && x<x1 ? -(x/(x1-x0)) + (x0/(x1-x0)) : x>x1 &
23
24
25 //-----
26 //Operadores
27
28 //operador AND
29 let OperadorAND = ( a , b ) => Math.min( a , b )
30
31 //operador OR
32 let OperadorOR = ( a , b ) => Math.max( a , b )
```

8.- Aquí se encuentran las funciones que se mencionaba también que manejaba la logia difusa, funciónBool, Grado, Triangulo, Trapezoidal dándole los valores en esta caso x, x0, x1 para la tabla de matriz que obtendrá los resultados dependiendo de lo ingresado por el usuario.

```
//-----
//Operadores

//operador AND
let OperadorAND = ( a , b ) => Math.min( a , b )

//operador OR
let OperadorOR = ( a , b ) => Math.max( a , b )

//operador NOT
let OperadorNOT = a => 1.0 - a

//-----

export{funcionBool , funcionInvBool , funcionGrado , funcionInvGrado , funcionTriangulo , funcionInvTriangulo , funcionTrap
```


9.- Por último, están los operadores AND, OR y NOT.



TECNOLÓGICO  
NACIONAL DE MÉXICO




Pruebas de ejecución algoritmo:

 Logica Difusa

[Home](#) [Documentación](#) [Acerca de](#)

## Logica Difusa

 34


¿Ardor en Garganta?  
NO

¿Pérdida del Gusto?  
NO

¿Pérdida del Olfato?  
NO

Enviar

Su diagnostico es: **Pasa**





TECNOLÓGICO  
NACIONAL DE MÉXICO



Logica Difusa



Home



Documentación



Acerca de

## Logica Difusa



40

¿Ardor en Garganta?

SI



¿Pérdida del Gusto?

SI



¿Pérdida del Olfato?

NO



Enviar

Su diagnostico es: Covid







40

¿Ardor en Garganta?  
SI



¿Pérdida del Gusto?  
NO

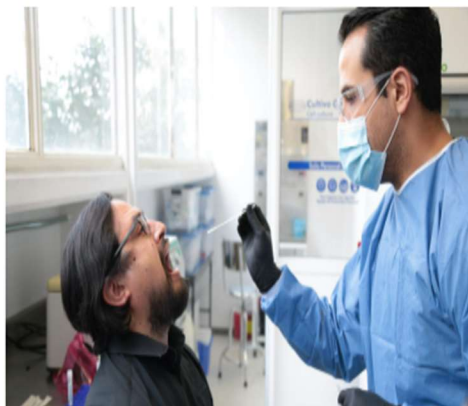


¿Pérdida del Olfato?  
NO



Enviar

Su diagnostico es: Sospechoso





## 5. Conclusiones

En conclusión, este proyecto utilizó esta metodología porque el tema es un problema de la vida real que todo el mundo está viviendo, entonces esta metodología ayuda a resolver este tipo de problemas y llegar a una conclusión pasando por un proceso deductivo y así poder tomar una decisión.





## 6. Referencias

- [1] "Lógica difusa [2021]", Wikipedia, 2021. Avialable: [Lógica difusa - Wikipedia, la enciclopedia libre](#) [Accessed: 06-Junio-2021].
- [2] "Matlab [2021]", MatlabWorks, 2021. Avialable: [MATLAB - El lenguaje del cálculo técnico - MATLAB & Simulink \(mathworks.com\)](#) [Accessed: 06-Junio-2021].
- [3] "JavaScript [2021]", MDN Web Docs, 2021. Avialable: [¿Qué es JavaScript? - Aprende sobre desarrollo web | MDN \(mozilla.org\)](#) [Accessed: 06-Junio-2021].