



Tecnólogo en Análisis y Desarrollo de  
Software

Ficha 2899747

**ARREGLOS**

**TEMA: ARREGLOS**

**INSTRUCTOR: ANDRES MORENO COLLASOS**

**TECNOLOGO: ADSO**

**FICHA: 2899747**

**NOMBRE: ANGEL FARID RIVERA SUAREZ**

**TI: 1075231948**

**SENA DE LA INDUSTRIA Y DE LA EMPRESA DE LOS SERVICIOS NEIVA/HUILA**



Tecnólogo en Análisis y Desarrollo de  
Software  
Ficha 2899747

## **ARREGLOS**

### Contenido

Ejercicios.....	3
-----------------	---



# Tecnólogo en Análisis y Desarrollo de Software

Ficha 2899747

## ARREGLOS

### Ejercicios

#### 1. Ejercicio practico

Nombre de la Arreglo:		Versión: 1.0
Descripción: Arreglo que almacena los primeros 5 números.		
numeros contar	Tipo de variable: Arreglo (array), Valores: Numérico (int)	

Código:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Practica</title>
</head>
<body>
  <script>
    let numeros=[];
    let contar;
    for(contar=1; contar<=5; contar++){
      numeros[contar]=contar;
    }
    console.log("los numero del 1 al 5:"+numeros )
  </script>
</body>
</html>
```

Prueba:

los numero del 1 al 5:,1,2,3,4,5

[practica.html:15](#)

#### Ciclo:

<b>Nombre: Ciclo for</b>	<b>Versión: 1.0</b>
<b>Descripción:</b> Ciclo que cuenta los numeros del 1 a 5	
<b>Variables:</b> numeros, contar	
<b>Código:</b> <pre>for(contar=1; contar&lt;=5; contar++){   numeros[contar]=contar; }</pre>	



# Tecnólogo en Análisis y Desarrollo de Software

Ficha 2899747

## ARREGLOS

### 2. Ejercicio Numero #1

Nombre de la Arreglo: par, impar, arreglo		Versión: 1.0
Descripción: arreglo: almacenaría los 10 primeros números par: Arreglo que solamente almacenaría los números pares del 1 al 10 impar: Arreglo que solamente almacenaría los números impares del 1 al 10		
arreglo, par impar	Tipo de variable: Arreglo (array), Valores: Numérico (int) Tipo de variable: Arreglo (array), Valores: Numérico (int) Tipo de variable: Arreglo (array), Valores: Numérico (int)	
Código:		
<pre>&lt;!DOCTYPE html&gt; &lt;html lang="en"&gt; &lt;head&gt;   &lt;meta charset="UTF-8"&gt;   &lt;meta name="viewport" content="width=device-width, initial-scale=1.0"&gt;   &lt;title&gt;Ejercicio 1&lt;/title&gt; &lt;/head&gt; &lt;body&gt;   &lt;script&gt;     let arreglo =[1,2,3,4,5,6,7,8,9,10];     let cantidad =arreglo.length;     let interar;     let par=[];     let impar=[];     for(interar=0;interar&lt;cantidad;interar++){       if(arreglo[interar]%2==0){         par.push(arreglo[interar]);       }else{         impar.push(arreglo[interar]);       }     }     console.log("Numeros de 1 al 10 "+ arreglo);     console.log("Estos valores son pares "+par);     console.log("Estos valores son impar "+impar);   &lt;/script&gt; &lt;/body&gt; &lt;/html&gt;</pre>		
Prueba:		
<div>Numeros de 1 al 10 1,2,3,4,5,6,7,8,9,10 <a href="#">index.html:23</a></div> <div>Estos valores son pares 2,4,6,8,10 <a href="#">index.html:24</a></div> <div>Estos valores son impar 1,3,5,7,9 <a href="#">index.html:25</a></div> <div>Live reload enabled. <a href="#">index.html:55</a></div>		

Ciclo:

<b>Nombre:</b> Ciclo for	<b>Versión:</b> 1.0
<b>Descripción:</b> Ciclo que genera los 10 primeros números y son almacenados en el arreglo 'numeros', después seleccionara los números pares e impares y serán almacenados en sus propios arreglos.	
<b>Variables:</b> par, impar, arreglo, cantidad.	
<b>Código:</b> <pre>for(interar=0;interar&lt;cantidad;interar++){   if(arreglo[interar]%2==0){     par.push(arreglo[interar]);   }else{     impar.push(arreglo[interar]);   } }</pre>	



# Tecnólogo en Análisis y Desarrollo de Software

## Ficha 2899747

### ARREGLOS

#### 3. Ejercicio Numero #2

Nombre del arreglo: arreglo, arreglo2, par, impar.		Versión: 1.0										
Descripción: arreglo: almacena la multiplicación del 5 hasta el 5. arreglo2: almacena la multiplicación del 9 hasta 5. par: almacena los numeros pares. impar: almacena los numero impares.												
arreglo, arreglo2, par, impar, num1, num2	Tipo de variable: Arreglo (array), Valores: Numérico (int) Tipo de variable: Arreglo (array), Valores: Numérico (int) Tipo de variable: Arreglo (array), Valores: Numérico (int) Tipo de variable: Numérico (int) Tipo de variable: Arreglo (array), Valores: Numérico (int) Tipo de variable: Numérico (int) Tipo de variable: Numérico (int)											
Código:												
<pre>&lt;!DOCTYPE html&gt; &lt;html lang="en"&gt; &lt;head&gt;   &lt;meta charset="UTF-8"&gt;   &lt;meta name="viewport" content="width=device-width, initial-scale=1.0"&gt;   &lt;title&gt;ejercicio 2&lt;/title&gt; &lt;/head&gt; &lt;body&gt;   &lt;script&gt;     let arreglo=[];     let arreglo2=[];     let par=[];     let impar=[];     let num1=1;     let num2=5;     i = 0;     e =0;     for (num1 ; num1 &lt;= num2; num1++) {       resul=num1*5; resul2=num1*9;       arreglo.push(resul)       arreglo2.push(resul2)       if (resul%2==0) {         i++         par =i;} else{           e++           impar=e;}       if (resul2%2==0) {         i++         par.i;} else{           e++           impar.e;         }     }      console.log("Multiplicacion de 5 hasta 5: "+ arreglo);     console.log("Multiplicacion de 9 hasta 5: "+ arreglo2);     console.log("Estos valores son pares: "+par);     console.log("Estos valores son impar: "+impar);   &lt;/script&gt; &lt;/body&gt; &lt;/html&gt;</pre>												
Prueba:												
<table><tr><td>Multiplicacion de 5 hasta 5: 5,10,15,20,25</td><td>index.html:34</td></tr><tr><td>Multiplicacion de 9 hasta 5: 9,18,27,36,45</td><td>index.html:35</td></tr><tr><td>Estos valores son pares: 3</td><td>index.html:36</td></tr><tr><td>Estos valores son impar: 5</td><td>index.html:37</td></tr><tr><td>Live reload enabled.</td><td>index.html:67</td></tr></table>			Multiplicacion de 5 hasta 5: 5,10,15,20,25	index.html:34	Multiplicacion de 9 hasta 5: 9,18,27,36,45	index.html:35	Estos valores son pares: 3	index.html:36	Estos valores son impar: 5	index.html:37	Live reload enabled.	index.html:67
Multiplicacion de 5 hasta 5: 5,10,15,20,25	index.html:34											
Multiplicacion de 9 hasta 5: 9,18,27,36,45	index.html:35											
Estos valores son pares: 3	index.html:36											
Estos valores son impar: 5	index.html:37											
Live reload enabled.	index.html:67											

Ciclo:

<b>Nombre:</b> Ciclo for	<b>Versión:</b> 1.0
<b>Descripción:</b> Ciclo que realiza la tabla del 5 y 9 condiciona que valores son pares e impares, para si poder insertarlas en sus propios arreglos.	
<b>Código:</b> <pre>for (num1 ; num1 &lt;= num2; num1++) {   resul=num1*5; resul2=num1*9;   arreglo.push(resul)   arreglo2.push(resul2)   if (resul%2==0) {     i++     par =i;} else{       e++       impar=e;}   if (resul2%2==0) {     i++     par.i;} else{       e++       impar.e;     } }</pre>	



# Tecnólogo en Análisis y Desarrollo de Software

## Ficha 2899747

### ARREGLOS

#### 4. Ejercicio Numero #3 Tabla del 5

Nombre de la Arreglo: par5,impar5,matrix5,matrix5 Versión: 1.0	
<b>Descripción:</b> <b>matrix5:</b> Arreglo que almacenaría los valores de la tabla del 5 <b>pares5:</b> Arreglo que solamente almacenaría los números pares de la tabla del 5 <b>impares5:</b> Arreglo que solamente almacenaría los números impares de la tabla del 5 <b>matrixX5:</b> Arreglo que permite hacer la forma "X" de la tabla del 5	
Num3 sumaimpar5 sumapar5	Tipo de variable: Arreglo (array), Valores: Numérico (int) Tipo de variable: Arreglo (array), Valores: Numérico (int) Tipo de variable: Arreglo (array), Valores: Numérico (int) Tipo de variable: Numérico (int) Tipo de variable: Arreglo (array), Valores: Numérico (int) Tipo de variable: Numérico (int) Tipo de variable: Numérico (int)

#### Código:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>ejercicio 3</title>
</head>
<body>
  <script>
    //tabla del 5
    let par5 = [];
    let impar5 = [];
    let sumaimpar5;
    let sumapar5;
    let num3=1;
    let matrix5 = [];

    let matrixX5 = "";
    for (let num1 = 0; num1 < 5; num1++) {
      matrix5[num1] = [];
      for (let num2 = 0; num2 < 5; num2++) {
        matrix5[num1][num2] = num3 * 5;
        num3++;
        if (matrix5[num1][num2] % 2 == 0) {
          par5.push(matrix5[num1][num2]);
        } else {
          impar5.push(matrix5[num1][num2]);
        }
        if (num1 === num2 || num1 + num2 === 4) {
          matrixX5 += matrix5[num1][num2] + "\t\t";
        } else {
          matrixX5 += "\t\t";
        }
      }
      matrixX5 += "\n";
    }

    sumapar5 = par5.reduce((sum, current) => sum + current, 0);
    sumaimpar5 = impar5.reduce((sum, current) => sum + current, 0);

    console.log(matrix5);
    console.log("valores pares " + par5);
    console.log("valores impares " + impar5);
    console.log("Suma de pares " + sumapar5);
    console.log("Suma de impares " + sumaimpar5);
    console.log(matrixX5);
```

#### Prueba:

```
index.html:41
(5) [Array(5), Array(5), Array(5), Array(5), Array(5)]
  0: (5) [5, 10, 15, 20, 25]
  1: (5) [30, 35, 40, 45, 50]
  2: (5) [55, 60, 65, 70, 75]
  3: (5) [80, 85, 90, 95, 100]
  4: (5) [105, 110, 115, 120, 125]
  length: 5
  [[Prototype]]: Array(0)
valores pares index.html:42
10,20,30,40,50,60,70,80,90,100,110,120
valores impares index.html:43
5,15,25,35,45,55,65,75,85,95,105,115,125
Suma de pares 780 index.html:44
Suma de impares 845 index.html:45
5 index.html:46
35 65 45 25
105 85 95 125
```



# Tecnólogo en Análisis y Desarrollo de Software

Ficha 2899747

## ARREGLOS

Ciclo:

<b>Nombre:</b> Ciclo for	<b>Versión:</b> 1.0
<b>Descripción:</b> Ciclo crea la matriz 5x5	
<b>Código:</b> <pre>let matrizx5 = ""; for (let num1 = 0; num1 &lt; 5; num1++) {   matriz5[num1] = [];   for (let num2 = 0; num2 &lt; 5; num2++) {</pre>	

Ciclo:

<b>Nombre:</b> Ciclo for	<b>Versión:</b> 1.0
<b>Descripción:</b> Formación de una "X" de la matriz de la tabla del 5 mediante una condición.	
<b>Código:</b> <pre>for (let num2 = 0; num2 &lt; 5; num2++) {   matriz5[num1][num2] = num3 * 5;   num3++;   if (matriz5[num1][num2] % 2 == 0) {     par5.push(matriz5[num1][num2]);   } else {     impar5.push(matriz5[num1][num2]);   }   if (num1 === num2    num1 + num2 === 4) {     matrizx5 += matriz5[num1][num2] + "\t\t";   } else {     matrizx5 += "\t\t";   } } matrizx5 += "\n";</pre>	

Ciclo:

<b>Nombre:</b> Ciclo for	<b>Versión:</b> 1.0
<b>Descripción:</b> sumaspar5= suma los numeros pares Sumasimpar5= suma los numeros impares	
<b>Código:</b> <pre>sumaspar5 = par5.reduce((sum, current) =&gt; sum + current, 0); sumasimpar5 = impar5.reduce((sum, current) =&gt; sum + current, 0);  console.log(matriz5); console.log("valores pares " + par5); console.log("valores impares " + impar5); console.log("Suma de pares " + sumaspar5); console.log("Suma de impares " + sumasimpar5); console.log(matrizx5);</pre>	



# Tecnólogo en Análisis y Desarrollo de Software

## Ficha 2899747

### ARREGLOS

Tabla del 9:

<b>Nombre de la Arreglo:</b> par9, impar9, matrizx9, matriz9		<b>Versión:</b> 1.0
<b>Descripción:</b> matriz9: Arreglo que almacenaría los valores de la tabla del 9 pares9: Arreglo que solamente almacenaría los números pares de la tabla del 9 impares9: Arreglo que solamente almacenaría los números impares de la tabla del 9 matrixX9: Arreglo que permite hacer la forma "X" de la tabla del 9		
num4 sumaimpar9 sumapar9	Tipo de variable: Arreglo (array), Valores: Numérico (int) Tipo de variable: Arreglo (array), Valores: Numérico (int) Tipo de variable: Arreglo (array), Valores: Numérico (int) Tipo de variable: Numérico (int) Tipo de variable: Arreglo (array), Valores: Numérico (int) Tipo de variable: Numérico (int) Tipo de variable: Numérico (int)	

#### Código:

```
//tabla del 9
let par9 = [];
let impar9 = [];
let sumaimpar9;
let num4=1;
let sumapar9;
let matriz9 = [];

let matrizx9 = "";
for (let num1 = 0; num1 < 5; num1++) {
  matriz9[num1] = [];
  for (let num2 = 0; num2 < 5; num2++) {
    matriz9[num1][num2] = num4 * 9;
    num4++;
    if (matriz9[num1][num2] % 2 == 0) {
      par9.push(matriz9[num1][num2]);
    } else {
      impar9.push(matriz9[num1][num2]);
    }
    if (num1 === num2 || num1 + num2 === 4) {
      matrizx9 += matriz9[num1][num2] + "\t\t";
    } else {
      matrizx9 += "\t\t";
    }
  }
  matrizx9 += "\n";
}

sumapar9 = par9.reduce((sum, current) => sum + current, 0);
sumaimpar9 = impar9.reduce((sum, current) => sum + current, 0);

console.log(matriz9);
console.log("valores pares " + par9);
console.log("valores impares " + impar9);
console.log("Suma de pares " + sumapar9);
console.log("Suma de impares " + sumaimpar9);
console.log(matrizx9);

</script>
</body>
</html>
```

#### Prueba:

```
index.html:80
(5) [Array(5), Array(5), Array(5), Array(5), Array(5)]
  0: (5) [9, 18, 27, 36, 45]
  1: (5) [54, 63, 72, 81, 90]
  2: (5) [99, 108, 117, 126, 135]
  3: (5) [144, 153, 162, 171, 180]
  4: (5) [189, 198, 207, 216, 225]
    length: 5
  [[Prototype]]: Array(0)
valores pares index.html:81
18,36,54,72,90,108,126,144,162,180,198,216
valores impares index.html:82
9,27,45,63,81,99,117,135,153,171,189,207,225
Suma de pares 1404 index.html:83
Suma de impares 1521 index.html:84
9 index.html:85
63 117 81 45
189 153 171 225
```





# Tecnólogo en Análisis y Desarrollo de Software

## Ficha 2899747

### ARREGLOS

Ciclo:

<b>Nombre:</b> Ciclo for	<b>Versión:</b> 1.0
<b>Descripción:</b> Ciclo crea la matriz 5x5 de la tabla del 9.	
<b>Código:</b> <pre>let matrizx9 = ""; for (let num1 = 0; num1 &lt; 5; num1++) {   matriz9[num1] = [];</pre>	

Ciclo:

<b>Nombre:</b> Ciclo for	<b>Versión:</b> 1.0
<b>Descripción:</b> Formación de una "X" de la matriz de la tabla del 9 mediante una condición.	
<b>Código:</b> <pre>for (let num2 = 0; num2 &lt; 5; num2++) {   matriz9[num1][num2] = num4 * 9;   num4++;   if (matriz9[num1][num2] % 2 == 0) {     par9.push(matriz9[num1][num2]);   } else {     impar9.push(matriz9[num1][num2]);   }   if (num1 === num2    num1 + num2 === 4) {     matrizx9 += matriz9[num1][num2] + "\t\t";   } else {     matrizx9 += "\t\t";   }   matrizx9 += "\n"; }</pre>	

Ciclo:

<b>Nombre:</b> Ciclo for	<b>Versión:</b> 1.0
<b>Descripción:</b> sumaspar9= suma los numeros pares Sumasimpar9= suma los numeros impares	
<b>Código:</b> <pre>sumaspar9 = par9.reduce((sum, current) =&gt; sum + current, 0); sumasimpar9 = impar9.reduce((sum, current) =&gt; sum + current, 0);  console.log(matriz9); console.log("valores pares " + par9); console.log("valores impares " + impar9); console.log("Suma de pares " + sumaspar9); console.log("Suma de impares " + sumasimpar9); console.log(matrizx9);</pre>	



# Tecnólogo en Análisis y Desarrollo de Software

## Ficha 2899747

### ARREGLOS

#### 5. Ejercicio Numero #4

<b>Nombre de la Arreglo:</b> bingo, bingoX1, bingoX2, bingoX3,		<b>Versión:</b> 1.0
<b>Descripción:</b> Crea una tabla del bingo.		
<b>número, interation1, interation2, acumular</b>	<b>Tipo de variable:</b> Arreglo (array), Valores: Numérico (int) <b>Tipo de variable:</b> Arreglo (array), Valores: Numérico (int) <b>Tipo de variable:</b> Arreglo (array), Valores: Numérico (int) <b>Tipo de variable:</b> Arreglo (array), Valores: Numérico (int) <b>Tipo de variable:</b> Arreglo (array), Valores: Numérico (int) <b>Tipo de variable:</b> Arreglo (array), Valores: Numérico (int) <b>Tipo de variable:</b> Arreglo (array), Valores: Numérico (int) <b>Tipo de variable:</b> Arreglo (array), Valores: Numérico (int) <b>Tipo de variable:</b> Arreglo (array), Valores: Numérico (int)	

#### Código:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Bingo</title>
</head>
<body>
  <script>
    let bingo=[]
    let bingoX1=[]
    let bingoX2=[]
    let bingoX3=[]
    let numero=1
    let interation1;
    let interation2;
    let acumulador=0;

    for(interation1=0; interation1<5; interation1++){
      let interno=[];
      for (interation2=0; interation2<5; interation2++){
        acumulador = acumulador + 1;
        interno[interation2] = acumulador*2;
      }
      bingo[interation1] = interno;

      console.log(bingo);
      console.log ("Letra B \n");
      for(interation2=0; interation2<5; interation2++){
        console.log(bingo[interation2][0])
      }
      console.log ("Letra I \n");
      for(interation2=0; interation2<5; interation2++){
        console.log(bingo[interation2][1])
      }
      console.log ("Letra N \n");
      console.log ("Letra N \n");
      for(interation2=0; interation2<5; interation2++){
        console.log(bingo[interation2][2])
      }
      console.log ("Letra G \n");
      for(interation2=0; interation2<5; interation2++){
        console.log(bingo[interation2][3])
      }
      console.log ("Letra O \n");
      for(interation2=0; interation2<5; interation2++){
        console.log(bingo[interation2][4])
      }

      for(let fila=0;fila<5;fila++){
        bingo[fila]=[]
        for(let columna=0;columna<5;columna++){
          bingo[fila][columna]=numero *2;
          numero++;
        }
      }
      for(let fila=0;fila<3;fila++){
        for(let columna=0;columna<3;columna++){
          if(fila==columna || fila+columna==2){
            bingoX1.push(bingo[fila][columna]);
          }
        }
      }
    }
  </script>
</body>
</html>
```



# Tecnólogo en Análisis y Desarrollo de Software

Ficha 2899747

## ARREGLOS

```
for(let fila=2;fila<5;fila++){
  for(let columna=0;columna<3;columna++){
    if(fila+columna==(columna+1)*2 || fila+columna==4){
      bingoX2.push(bingo[fila][columna]);
    }
  }
}

for(let fila=0; fila<3;fila++){
  for(let columna=2; columna<5;columna++){
    if(fila+columna==(fila+1)*2 || fila+columna==4){
      bingoX3.push(bingo[fila][columna]);
    }
  }
}

console.log("Matrices en x")
console.log("Matriz en x1")
console.log(bingoX1)
console.log("Matriz en x2")
console.log(bingoX2)
console.log("Matriz en x3")
console.log(bingoX3)
</script>
</body>
</html>
```

Prueba:

```
Array(5)
  0: (5) [2, 4, 6, 8, 10]
  1: (5) [12, 14, 16, 18, 20]
  2: (5) [22, 24, 26, 28, 30]
  3: (5) [32, 34, 36, 38, 40]
  4: (5) [42, 44, 46, 48, 50]
length: 5
```

```
Letra B index.html:29
2 index.html:31
12 index.html:31
22 index.html:31
32 index.html:31
42 index.html:31
Letra I index.html:34
4 index.html:37
14 index.html:37
24 index.html:37
34 index.html:37
44 index.html:37
Letra N index.html:39
6 index.html:42
16 index.html:42
26 index.html:42
36 index.html:42
```

```
Letra N index.html:39
6 index.html:42
16 index.html:42
26 index.html:42
36 index.html:42
46 index.html:42
Letra G index.html:44
8 index.html:47
18 index.html:47
28 index.html:47
38 index.html:47
48 index.html:47
Letra O index.html:50
10 index.html:53
20 index.html:53
30 index.html:53
40 index.html:53
50 index.html:53
Matrices en x index.html:89
```

```
Matrices en x index.html:89
Matriz en x1 index.html:90
Array(5)
  0: 2
  1: 6
  2: 14
  3: 22
  4: 26
length: 5
  [[Prototype]]: Array(0)
Matriz en x2 index.html:92
Array(5)
  0: 22
  1: 26
  2: 34
  3: 42
  4: 46
length: 5
  [[Prototype]]: Array(0)
```

```
Matriz en x3 index.html:94
Array(5)
  0: 6
  1: 10
  2: 18
  3: 26
  4: 30
length: 5
  [[Prototype]]: Array(0)
Live reload enabled. index.html:126
```



# Tecnólogo en Análisis y Desarrollo de Software

Ficha 2899747

## ARREGLOS

Ciclo:

<b>Nombre:</b> Ciclo for: variables: iteration1, interno, acumular, bingo	<b>Versión:</b> 1.0
<b>Descripción:</b> Ciclo que genera la tabla del bingo y mediante una condición, insertas las 5 columnas en sus propios arreglos.	
<b>Código:</b> <pre>for(interaction1=0; interaction1&lt;5; interaction1++){   let interno=[];   for (interaction2=0; interaction2&lt;5; interaction2++){     acumulador = acumulador + 1;     interno[interaction2] = acumulador*2;   }   bingo[interaction1] = interno; }  console.log(bingo);</pre>	

<b>Nombre:</b> Ciclo for	<b>Versión:</b> 1.0
<b>Descripción:</b> Ciclo almacena los numero de la segunda fila en la letra B	
<b>Código:</b> <pre>console.log(bingo); console.log ("Letra B \n"); for(interaction2=0; interaction2&lt;5; interaction2++){   console.log(bingo[interaction2][0]) }</pre>	

<b>Nombre:</b> Ciclo for	<b>Versión:</b> 1.0
<b>Descripción:</b> Ciclo almacena los numero de la segunda fila en la letra I	
<b>Código:</b> <pre>console.log ("Letra I \n");  for(interaction2=0; interaction2&lt;5; interaction2++){   console.log(bingo[interaction2][1]) }</pre>	



# Tecnólogo en Análisis y Desarrollo de Software

Ficha 2899747

## ARREGLOS

**Nombre:** Ciclo for: variables: iteration2, bingo | **Versión:** 1.0

**Descripción:** Ciclo almacena los numero de la primera fila en la letra N

**Código:**

```
console.log ("Letra N \n");  
  
for(iteration2=0; iteration2<5; iteration2++){  
    console.log(bingo[iteration2][2])  
}
```

**Nombre:** Ciclo for variables: iteration2, bingo | **Versión:** 1.0

**Descripción:** Ciclo almacena los numero de la segunda fila en la letra G

**Código:**

```
console.log ("Letra G \n");  
  
for(iteration2=0; iteration2<5; iteration2++){  
    console.log(bingo[iteration2][3])  
}
```

**Nombre:** Ciclo for | **Versión:** 1.0

**Descripción:** Ciclo almacena los numero de la segunda fila en la letra O

**Código:**

```
console.log ("Letra O \n");  
  
for(iteration2=0; iteration2<5; iteration2++){  
    console.log(bingo[iteration2][4])  
}
```

**Nombre:** Ciclo for: Variable: bingo, fila, columna, numero.

**Versión:** 1.0

**Descripción:** Ciclo que genera la forma de un "X1" mediante una condición, pero, estaría almacenada en un arreglo

**Código:**

```
for(let fila=0;fila<5;fila++){  
    bingo[fila]=[]  
    for(let columna=0;columna<5;columna++){  
        bingo[fila][columna]=numero *2;  
        numero++;  
    }  
}
```



# Tecnólogo en Análisis y Desarrollo de Software

Ficha 2899747

## ARREGLOS

**Nombre:** Ciclo for: **Variable:** bingoX2, fila, columna, numero.

**Versión:** 1.0

**Descripción:** Ciclo que genera la forma de un "X2" mediante una condición, pero, estaría almacenada en un arreglo

**Código:**

```
for(let fila=2;fila<5;fila++){  
  for(let columna=0;columna<3;columna++){  
    if(fila+columna==(columna+1)*2 || fila+columna==4){  
      bingoX2.push(bingo[fila][columna]);  
    }  
  }  
}
```

**Nombre:** Ciclo for Variable: bingoX3, fila, columna, numero.

**Versión:** 1.0

**Descripción:** Ciclo que genera la forma de un "X3" mediante una condición, pero, estaría almacenada en un arreglo

**Código:**

```
for(let fila=0; fila<3;fila++){  
  for(let columna=2; columna<5;columna++){  
    if(fila+columna==(fila+1)*2 || fila+columna==4){  
      bingoX3.push(bingo[fila][columna]);  
    }  
  }  
}
```



# Tecnólogo en Análisis y Desarrollo de Software

## Ficha 2899747

### ARREGLOS

#### 6. Ejercicio Numero #5

**Nombre de la Arreglo:** tienda y listaBusqueda

**Descripción:** Arreglo que almacena los primeros 5 números.

**tienda**  
**nuevoProducto**  
**listaBusqueda**

**Descripción:**

Tienda: Arreglo que almacena valores referentes a los productos de la canasta familiar.

listaBusqueda: Arreglo que busca un valor del anterior arreglo.

**Código:**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Fruver</title>
</head>
<body>
  <script>
    let tienda=[];
    let numeroRegistro;
    let interacion;
    let nuevoProducto;
    let listaBusqueda =[];

    tienda=[
      {producto:'Arroz', tipProducto:'Granos', tipoUnidad:'Gramos', cantidad:'1000', precio:'2450'},
      {producto:'Papa', tipProducto:'Fruver', tipoUnidad:'Gramos', cantidad:'500', precio:'1000'},
      {producto:'Mora', tipProducto:'Fruver', tipoUnidad:'Gramos', cantidad:'500', precio:'1500'},
      {producto:'Trucha', tipProducto:'Carne', tipoUnidad:'Gramos', cantidad:'1000', precio:'9000'},
      {producto:'Pollo completo', tipoProducto:'Carne', tipoUnidad:'Gramos', cantidad:'1000', precio:'4500'},
    ];

    nuevoProducto = {producto:'res', tipProducto:'Carne', tipUnidad:'gramos', cantidad: '1000', precio: '7500'};

    tienda.push(nuevoProducto)

    numeroRegistros = tienda.length;

    for(interacion=0; interacion<numeroRegistros; interacion++){
      //console.log(tienda[interacion]);
      if(tienda[interacion].tipProducto == 'Carne'){
        listaBusqueda.push(tienda[interacion]);
      } else{
        console.log(" ");
      }
    }

    console.log(listaBusqueda);
    console.log(tienda);

  </script>
</body>
</html>
```

**Prueba:**

```
▼ Array(2) ①
  ▼ 0:
    cantidad: "1000"
    precio: "9000"
    producto: "Trucha"
    tipProducto: "Carne"
    tipoUnidad: "Gramos"
    ▶ [[Prototype]]: Object
  ▶ 1: {producto: 'res', tipProducto: 'Carne', tipUnid
    length: 2
    ▶ [[Prototype]]: Array(0)
    index.html:40

▼ Array(6) ①
  ▼ 0:
    cantidad: "1000"
    precio: "2450"
    producto: "Arroz"
    tipProducto: "Granos"
    tipoUnidad: "Gramos"
    ▶ [[Prototype]]: Object
  ▶ 1: {producto: 'Papa', tipProducto: 'Fruver', tipoU
  ▶ 2: {producto: 'Mora', tipProducto: 'Fruver', tipoU
  ▶ 3: {producto: 'Trucha', tipProducto: 'Carne', tipo
  ▶ 4: {producto: 'Pollo completo', tipProducto: 'Car
  ▶ 5: {producto: 'res', tipProducto: 'Carne', tipUnid
    length: 6
    ▶ [[Prototype]]: Array(0)
    index.html:71

Live reload enabled.
```



# Tecnólogo en Análisis y Desarrollo de Software

Ficha 2899747

## ARREGLOS

Ciclo:

**Nombre:** Ciclo for

**Versión:** 1.0

**Descripción:** Ciclo que itera los valores del arreglo “tienda” y mediante una condición, imprima los valores que corresponden de tipo “Fruver”

**Código:**

```
numeroRegistros = tienda.length;

for(interacion=0; interacion<numeroRegistros; interacion++){
    //console.log(tienda[interacion]);
    if(tienda[interacion].tipProducto == 'Carne'){
        listaBusqueda.push(tienda[interacion]);
    } else{
        console.log(" ");
    }
}
```





# Tecnólogo en Análisis y Desarrollo de Software

## Ficha 2899747

### ARREGLOS

#### 7. Ejercicio Numero #6

**Nombre de la Arreglo: personas, nomina**

**Descripción:**

**personas:** Arreglo que almacenara los datos de las personas.

**nomina:** Arreglo que almacenara mediante sus funciones, la nómina de las personas.

<b>Personas</b>	Tipo de variable: Arreglo (array)
<b>Nomina</b>	Tipo de variable: Arreglo (array)
<b>salarioMin</b>	Tipo de variable: Numérico (int)
<b>numeroRegistros</b>	Tipo de variable: Numérico (int)
<b>mostrar</b>	Tipo de variable: Numérico (int)
<b>mostrarRetencion</b>	Tipo de variable: Numérico (int)
<b>paValorDia</b>	Tipo de variable: Numérico (int)
<b>paSalario</b>	Tipo de variable: Numérico (int)
<b>salud</b>	Tipo de variable: Numérico (int)
<b>iteración</b>	Tipo de variable: Numérico (int)
<b>salarioCalculado</b>	Tipo de variable: Numérico (int)
<b>trabajador</b>	Tipo de variable: Numérico (int)
<b>subTransCalculado</b>	Tipo de variable: Numérico (int)
<b>arlCalculado</b>	Tipo de variable: Numérico (int)
<b>pensionCalculado</b>	Tipo de variable: Numérico (int)
<b>deducibleCalculado</b>	Tipo de variable: Numérico (int)
<b>saludCalculado</b>	Tipo de variable: Numérico (int)
<b>retencionCalculado</b>	Tipo de variable: Numérico (int)
<b>totalCalculado</b>	Tipo de variable: Numérico (int)

**Código:**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Nomina</title>
</head>
<body>
  <script>
    let personas = [];
    let nomina = [];
    let salarioMin=16000000;
    let numeroRegistros;
    let mostrar;
    let mostrarRetencion;

    const funcionSalario = function(paValorDia, paDias) {
      return paDias * paValorDia;
    };

    const funcioSubtransporte = function(paSalario){
      if(paSalario<salarioMin*2){
        paSalario= paSalario+1200000
        mostrar= "Si se aplica el subsidio " + paSalario
      }else{
        paSalario= paSalario+0
        mostrar= "No se aplica el subsidio " + paSalario
      }
      return mostrar
    }

    const funcionSalud=function(paSalario){
      return paSalario * 0.008
    }

    const funcionPension=function(paSalario){
      return paSalario * 0.12
    }

    const funcionArl=function(paSalario){
      return paSalario * 0.05
    }

    const funcionDeducible = function(paSalario){
      return funcionSalud(paSalario) + funcionPension(paSalario) + funcionArl(paSalario)
    }
  </script>
</body>
</html>
```



# Tecnólogo en Análisis y Desarrollo de Software

## Ficha 2899747

### ARREGLOS

```
81 {id:1029854173, nombre:'Keiner', apellido:'Hernandez',cargo:'Director de Innovación',valorDia:110000,diasTrabajado:31 },
82 {id:4006524133, nombre:'Andres', apellido:'Horta',cargo:'Contador',valorDia:150000 ,diasTrabajado:31 },
83 {id:1272354717 , nombre:'Lorena', apellido:'Salazar',cargo:'Atención al cliente',valorDia:50000,diasTrabajado: 31},
84 {id:1013457570, nombre:'Pauila', apellido:'Suarez',cargo:'Atención al cliente',valorDia:50000,diasTrabajado:31 },
85 {id:1587231342 , nombre:'Ignacio', apellido:'Bermeo',cargo:'Director de Tecnología',valorDia:200000,diasTrabajado: 31}
86 ];
87
88
89 numeroRegistros = personas.length;
90
91 for (let iteracion = 0; iteracion < numeroRegistros; iteracion++) {
92   let trabajador = personas[iteracion];
93   let salarioCalculado = funcionSalario(trabajador.valorDia, trabajador.diasTrabajado);
94   let subTransCalculada=funcionSubtransporte(salarioCalculado);
95   let retencionCalculada=funcionRetener(salarioCalculado);
96   let saludCalculada = funcionSalud(salarioCalculado);
97   let pensionCalculada=funcionPension(salarioCalculado);
98   let arlCalculada= funcionArl(salarioCalculado);
99   let deducibleCalculada=funcionDeducible(salarioCalculado);
100   let totalCalculada=funcionTotal(salarioCalculado);
101
102   nomina.push({
103     id: trabajador.id,
104     nombre: trabajador.nombre,
105     apellido: trabajador.apellido,
106     cargo: trabajador.cargo,
107     salario: salarioCalculado,
108     subTransporte: subTransCalculada,
109     retencion: retencionCalculada,
110     salud: saludCalculada ,
111     pension: pensionCalculada,
112     arl:arlCalculada,
113     deducible: deducibleCalculada,
114     total:totalCalculada,
115   });
116 }
117
118
119
120 console.log(nomina);
121 </script>
122 </body>
123 </html>
```

```
const funcionArl=function(paSalario){
  return paSalario * 0.05
}

const funcionDeducible = function(paSalario){
  return funcionSalud(paSalario) + funcionPension(paSalario) + funcionArl(paSalario)
}

const funcionRetener = function(paSalario) {
  let mostrarRetencion;

  if (paSalario > salarioMin * 12) {
    paSalario = paSalario * 0.08;
    mostrarRetencion = "Retencion de 0.08 " + paSalario;
  } else if (paSalario > salarioMin * 8) {
    paSalario = paSalario * 0.04;
    mostrarRetencion = "Retencion de 0.04 " + paSalario;
  } else if (paSalario > salarioMin * 6) {
    paSalario = paSalario * 0.02;
    mostrarRetencion = "Retencion de 0.02 " + paSalario;
  } else {
    mostrarRetencion = "No aplica retención";
  }

  return mostrarRetencion;
};

const funcionTotal=function(paSalario){
  return paSalario - funcionDeducible(paSalario)
}

personas = [
  {id:1075231948 , nombre:'Sofia', apellido:'Plazas',cargo:'Aseadora',valorDia:43000,diasTrabajado:31 },
  {id:1074080923, nombre:'Gustavo', apellido:'Soto',cargo:'Director de Marketing',valorDia:120000,diasTrabajado: 31},
  {id:4077652312, nombre:'Wabel', apellido:'Rivera',cargo:'Directora de Recursos Humanos',valorDia:96000,diasTrabajado:31 },
  {id:1054236808, nombre:'Clara', apellido:'Lopez',cargo:'Aseadora',valorDia:43000,diasTrabajado:31 },
  {id:4124654334, nombre:'Costanza', apellido:'Ortiz',cargo:'Director comercial',valorDia:140000,diasTrabajado: 31 },
  {id:1029854173, nombre:'Keiner', apellido:'Hernandez',cargo:'Director de Innovación',valorDia:110000,diasTrabajado:31 },
  {id:4006524133, nombre:'Andres', apellido:'Horta',cargo:'Contador',valorDia:150000 ,diasTrabajado:31 },
  {id:1272354717 , nombre:'Lorena', apellido:'Salazar',cargo:'Atención al cliente',valorDia:50000,diasTrabajado: 31},

```



# Tecnólogo en Análisis y Desarrollo de Software

## Ficha 2899747

### ARREGLOS

Prueba:

```
ejercicio1.html:120
▼ Array(10) 1
  ► 0: {id: 1075231948, nombre: 'Sofia', apellido: 'Plaza'}
  ► 1: {id: 1074800923, nombre: 'Gustavo', apellido: 'Soto'}
  ► 2: {id: 4077652312, nombre: 'Mabel', apellido: 'Rivera'}
  ► 3: {id: 1054236808, nombre: 'Clara', apellido: 'Lopez'}
  ► 4: {id: 4124654334, nombre: 'Costanza', apellido: 'Ortiz'}
  ► 5: {id: 1029854173, nombre: 'Keiner', apellido: 'Hernandez'}
  ► 6: {id: 4006524133, nombre: 'Andres', apellido: 'Horta'}
  ► 7: {id: 1272354717, nombre: 'Lorena', apellido: 'Salazar'}
  ► 8: {id: 1013457578, nombre: 'Paubla', apellido: 'Suarez'}
  ► 9: {id: 1587231342, nombre: 'Ignacio', apellido: 'Bermeo'}
  length: 10
  ► [[Prototype]]: Array(0)
```

```
► 6: {id: 4006524133, nombre: 'Andres', apellido: 'Horta'}
▼ 7:
  apellido: "Salazar"
  arl: 77500
  cargo: "Atención al cliente"
  deducible: 275900
  id: 1272354717
  nombre: "Lorena"
  pension: 186000
  retencion: "No aplica retención"
  salario: 1550000
  salud: 12400
  subTransporte: "Si se aplica el subsidio 1670000"
  total: 1274100
```

- Registros

Nombre del arreglo: persona		Versión: 1.9
Descripción: Los 10 registros de las personas		
id	Tipo de propiedad: Numérico (int)	
nombre	Tipo de propiedad: String	
apellido	Tipo de propiedad: String	
cargo	Tipo de propiedad: String	
valorDia	Tipo de propiedad: Numérico (int)	
diasTrabajado	Tipo de propiedad: Numérico (int)	
Código:		
<pre>personas = [     {id:1075231948 , nombre:'Sofia', apellido:'Plazas',cargo:'Aseadora',valorDia:43000,diasTrabajado:31 },     {id:1074800923, nombre:'Gustavo', apellido:'Soto',cargo:'Director de Marketing',valorDia:120000,diasTrabajado: 31},     {id:4077652312, nombre:'Mabel', apellido:'Rivera',cargo:'Directora de Recursos Humanos',valorDia:96000,diasTrabajado:31 },     {id:1054236808, nombre:'Clara', apellido:'Lopez',cargo:'Aseadora',valorDia:43000,diasTrabajado:31 },     {id:4124654334, nombre:'Costanza', apellido:'Ortiz',cargo:'Director comercial',valorDia:140000,diasTrabajado: 31 },     {id:1029854173, nombre:'Keiner', apellido:'Hernandez',cargo:'Director de Innovación',valorDia:110000,diasTrabajado:31 },     {id:4006524133, nombre:'Andres', apellido:'Horta',cargo:'Contador',valorDia:150000 ,diasTrabajado:31 },     {id:1272354717 , nombre:'Lorena', apellido:'Salazar',cargo:'Atención al cliente',valorDia:50000,diasTrabajado: 31},     {id:1013457578, nombre:'Paubla', apellido:'Suarez',cargo:'Atención al cliente',valorDia:50000,diasTrabajado:31 },     {id:1587231342 , nombre:'Ignacio', apellido:'Bermeo',cargo:'Director de Tecnología',valorDia:200000,diasTrabajado: 31} ];</pre>		



# Tecnólogo en Análisis y Desarrollo de Software

Ficha 2899747

## ARREGLOS

Ciclo:

<b>Nombre de la función:</b> funcionTotal	<b>Versión:</b> 1.8
<b>Descripción:</b> Función que calcula el total	
<b>Código:</b> <pre>const funcionSalario = function(paValorDia, paDias) {   return paDias * paValorDia; };</pre>	

Ciclo:

<b>Nombre de la función:</b> funcionSubtransporte	<b>Versión:</b> 1.8
<b>Descripción:</b> Función que calcula el Transporte	
<b>Código:</b> <pre>const funcioSubtransporte = function(paSalario){   if(paSalario&lt;salarioMin*2){     paSalario= paSalario+120000     mostrar= "Si se aplica el subsidio " + paSalario   }else{     paSalario= paSalario+0     mostrar= "No se aplica el subsidio " + paSalario   }   return mostrar }</pre>	

Ciclo:

<b>Nombre de la función:</b> funcionSalud	<b>Versión:</b> 1.8
<b>Descripción:</b> Función que calcula la salud.	
<b>Código:</b> <pre>const funcionSalud=function(paSalario){   return paSalario * 0.008 }</pre>	



# Tecnólogo en Análisis y Desarrollo de Software

Ficha 2899747

## ARREGLOS

Función:

<b>Nombre de la función:</b> <b>funcionPension</b>	<b>Versión: 1.8</b>
<b>Descripción:</b> Función que calcula la pensión	
<b>Código:</b> <pre>const funcionPension=function(paSalario){     return paSalario * 0.12 }</pre>	

Función:

<b>Nombre de la función: funcionArl</b>	<b>Versión: 1.8</b>
<b>Descripción:</b> Función que calcula el arl	
<b>Código:</b> <pre>const funcionArl=function(paSalario){     return paSalario * 0.05 }</pre>	

Función:

<b>Nombre de la función:</b> <b>funcionTotal</b>	<b>Versión: 1.8</b>
<b>Descripción:</b> Función que calcula el total	
<b>Código:</b> <pre>const funcionDeducible = function(paSalario){     return funcionSalud(paSalario) + funcionPension(paSalario) + funcionArl(paSalario) }  const funcionTotal = function(paSalario) {</pre>	



# Tecnólogo en Análisis y Desarrollo de Software

Ficha 2899747

## ARREGLOS

Función:

<b>Nombre de la función:</b> funcionRetener	<b>Versión:</b> 1.8
<b>Descripción:</b> Función que calcula el total a Retener	
Código:	
<pre>const funcionRetener = function(paSalario) {   let mostrarRetencion;    if (paSalario &gt; salarioMin * 12) {     paSalario = paSalario * 0.08;     mostrarRetencion = "Retencion de 0.08 " + paSalario;   } else if (paSalario &gt; salarioMin * 8) {     paSalario = paSalario * 0.04;     mostrarRetencion = "Retencion de 0.04 " + paSalario;   } else if (paSalario &gt; salarioMin * 6) {     paSalario = paSalario * 0.02;     mostrarRetencion = "Retencion de 0.02 " + paSalario;   } else {     mostrarRetencion = "No aplica retención";   }    return mostrarRetencion; };</pre>	

Función:

<b>Nombre de la función:</b> funcionTotal	<b>Versión:</b> 1.8
<b>Descripción:</b> Función que calcula el total	
Código:	
<pre>const funcionTotal=function(paSalario){   return paSalario - funcionDeducible(paSalario) }</pre>	



# Tecnólogo en Análisis y Desarrollo de Software

Ficha 2899747

## ARREGLOS

Ciclo:

Ciclo:	Versión: 1.8
Descripción: Función que calcula el total a Retener	
Código:	
<pre>for (let iteracion = 0; iteracion &lt; numeroRegistros; iteracion++) {   let trabajador = personas[iteracion];   let salarioCalculado = funcionSalario(trabajador.valorDia, trabajador.diasTrabajado);   let subTransCalculada=funcioSubtransporte(salarioCalculado);   let retencionCalculada=funcionRetener(salarioCalculado);   let saludCalculada = funcionSalud(salarioCalculado);   let pensionCalculada=funcionPension(salarioCalculado);   let arlCalculada= funcionArl(salarioCalculado);   let deducibleCalculada=funcionDeducible(salarioCalculado);   let totalCalculada=funcionTotal(salarioCalculado);    nomina.push({     id: trabajador.id,     nombre: trabajador.nombre,     apellido: trabajador.apellido,     cargo: trabajador.cargo,     salario: salarioCalculado,     subTransporte: subTransCalculada,     retencion: retencionCalculada,     salud: saludCalculada ,     pension: pensionCalculada,     arl:arlCalculada,     deducible: deducibleCalculada,     total:totalCalculada,   }); }</pre>	