



Tecnólogo en Análisis y Desarrollo de  
Software  
Ficha

**Arreglos Js**

## **MANUAL FUNCIONES**

**RACINGER PRADA OLAYA**

**ANDRES MORENO**

**2024**



# Tecnólogo en Análisis y Desarrollo de Software

## Ficha

### Arreglos Js

**Nombre de la función:** holaMundo()

**Versión:** 1.0

**Descripción:**

Contar del 1 al 10

arreglo	Int
cantidadDatos	Int
numero	Int

**Código:**

```
0 let arreglo = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10,];
1   cantidadDatos = arreglo.length;
2   for(let numero = 0; numero<cantidadDatos;numero++){
3
4   }
5   console.log(arreglo);
```

```
▼ Array(10) 1
  0: 1
  1: 2
  2: 3
  3: 4
  4: 5
  5: 6
  6: 7
  7: 8
  8: 9
  9: 10
  length: 10
```

**Nombre de la función:** holaMundo()

**Versión:** 1.0

**Descripción:**

Pares e impares de la tabla del 5 y 9

matriz5	Int
pares5	Int
num1	Int
num2	Int
num3	Int
sumasPares5	Int
sumasImpares5	Int
matriz9	Int
pares9	Int



# Tecnólogo en Análisis y Desarrollo de Software

## Ficha

### Arreglos Js

multiplicador  
sumasPares9  
sumasImpares9

Int  
Int  
Int

Código:

```
*/
let matriz5 = [];
let pares5 = [];
let impares5 = [];
let num3 = 1;
let sumasPares5 = 0;
let sumasImpares5 = 0;

for (let num1 = 0; num1 < 5; num1++) {
  matriz5[num1] = [];
  for (let num2 = 0; num2 < 5; num2++) {
    matriz5[num1][num2] = num3 * 5;
    num3++;
    if(matriz5[num1][num2] % 2 == 0) {
      pares5.push(matriz5[num1][num2]);
    } else {
      impares5.push(matriz5[num1][num2]);
    }
  }
}

for (let i = 0; i < pares5.length; i++) {
  sumasPares5 += pares5[i];
}

for (let i = 0; i < impares5.length; i++) {
  sumasImpares5 += impares5[i];
}

console.log("Matriz 5:", matriz5);
console.log("Pares de la tabla del 5:", pares5);
console.log("Impares de la tabla del 5:", impares5);

let matriz9 = [];
let pares9 = [];
let impares9 = [];
let multiplicador = 1;
let sumasPares9 = 0;
let sumasImpares9 = 0;

for (let num1 = 0; num1 < 5; num1++) {
  matriz9[num1] = [];
  for (let num2 = 0; num2 < 5; num2++) {
    matriz9[num1][num2] = multiplicador * 9;
    multiplicador++;
    if(matriz9[num1][num2] % 2 == 0) {
      pares9.push(matriz9[num1][num2]);
    } else {
      impares9.push(matriz9[num1][num2]);
    }
  }
}

for (let i = 0; i < pares9.length; i++) {
  sumasPares9 += pares9[i];
}

for (let i = 0; i < impares9.length; i++) {
  sumasImpares9 += impares9[i];
}

console.log("Matriz 9:", matriz9);
console.log("Pares de la tabla del 9:", pares9);
console.log("Impares de la tabla del 9:", impares9);
```



# Tecnólogo en Análisis y Desarrollo de Software

## Ficha

### Arreglos Js

```
Matriz 5: <array(5) int>
  0: (5) [5, 10, 15, 20, 25]
  1: (5) [10, 15, 20, 25, 30]
  2: (5) [15, 20, 25, 30, 35]
  3: (5) [20, 25, 30, 35, 40]
  4: (5) [25, 30, 35, 40, 45]
  length: 5
  [[Prototype]]: <Array>

Pares de la tabla del 5: <array(12) int>
  0: 10
  1: 20
  2: 30
  3: 40
  4: 50
  5: 60
  6: 70
  7: 80
  8: 90
  9: 100
  10: 110
  11: 120
  length: 12
  [[Prototype]]: <Array>

Depares de la tabla del 5: <array(13) int>
  0: 5
  1: 10
  2: 15
  3: 20
  4: 25
  5: 30
  6: 35
  7: 40
  8: 45
  9: 50
  10: 55
  11: 60
  12: 65
  length: 13
  [[Prototype]]: <Array>

Matriz 5: <array(5) int>
  0: (5) [5, 10, 15, 20, 25]
  1: (5) [10, 15, 20, 25, 30]
  2: (5) [15, 20, 25, 30, 35]
  3: (5) [20, 25, 30, 35, 40]
  4: (5) [25, 30, 35, 40, 45]
  length: 5
  [[Prototype]]: <Array>

Pares de la tabla del 5: <array(12) int>
  0: 10
  1: 20
  2: 30
  3: 40
  4: 50
  5: 60
  6: 70
  7: 80
  8: 90
  9: 100
  10: 110
  11: 120
  length: 12
  [[Prototype]]: <Array>

Depares de la tabla del 5: <array(13) int>
  0: 5
  1: 10
  2: 15
  3: 20
  4: 25
  5: 30
  6: 35
  7: 40
  8: 45
  9: 50
  10: 55
  11: 60
  12: 65
  length: 13
  [[Prototype]]: <Array>
```

Nombre de la función: holaMundo()		Versión: 1.0
Descripción: Hacer una X		
matriz5	Int	
fila	Int	
columna	Int	
numero	Int	
matrixX	Int	
filaX	Int	
columnaX	Int	
Código:		



# Tecnólogo en Análisis y Desarrollo de Software

## Ficha

### Arreglos Js

```
let matriz5 = [];  
  
for (let fila = 0; fila < 5; fila++) {  
  matriz5[fila] = [];  
  for (let columna = 0; columna < 5; columna++) {  
    let numero = Math.floor(Math.random() * 100) + 1;  
    matriz5[fila][columna] = numero;  
  }  
}  
  
console.log("Matriz generada:");  
console.log(matriz5);  
  
let matrixX = "";  
  
for (let filaX = 0; filaX < 5; filaX++) {  
  for (let columnaX = 0; columnaX < 5; columnaX++) {  
    if (filaX === columnaX || filaX + columnaX === 4) {  
      matrixX += matriz5[filaX][columnaX] + "\t";  
    } else {  
      matrixX += " \t";  
    }  
  }  
  matrixX += "\n";  
}  
  
console.log("Matriz en forma de X:");  
console.log(matrixX);
```

Matriz generada:

[matris.js:16](#)

▼ Array(5) **i**

[matris.js:17](#)

▶ 0: (5) [56, 17, 96, 76, 80]  
▶ 1: (5) [69, 74, 46, 86, 4]  
▶ 2: (5) [33, 20, 61, 65, 98]  
▶ 3: (5) [59, 35, 71, 22, 76]  
▶ 4: (5) [32, 91, 35, 48, 91]  
length: 5  
▶ [[Prototype]]: Array(0)

Matriz en forma de X:

[matris.js:32](#)

56                    80  
  74            86  
      61  
  35            22  
32                    91

[matris.js:33](#)



# Tecnólogo en Análisis y Desarrollo de Software

## Ficha

### Arreglos Js

Nombre de la función: holaMundo()

Versión: 1.0

Descripción:

Tienda

tienda	Int
numero_registro	Int
iteración	Int
nuevoProducto	
listaBusqueda	

Código:

```
let tienda = [];  
let numero_registro;  
let iteracion;  
let nuevoProducto;  
let listaBusqueda=[];  
  
tienda = [  
  {producto:'Arroz', tipoProducto: 'granos', tipoUnidad: 'gramos', cantidad:1000, precio:2450},  
  {producto:'Papa', tipoProducto: 'fruver', tipoUnidad: 'gramos', cantidad:500, precio:1000},  
  {producto:'Mora', tipoProducto: 'fruver', tipoUnidad: 'gramos', cantidad:500, precio:1500},  
  {producto:'Trucha', tipoProducto: 'carnes', tipoUnidad: 'gramos', cantidad:1000, precio:9000},  
  {producto:'Pollo Entero', tipoProducto: 'carnes', tipoUnidad: 'gramos', cantidad:1000, precio:4500},  
];  
  
numero_registro = tienda.length;  
  
nuevoProducto = {producto:'Res', tipoProducto:'carnes', tipoUnidad:'gramos', cantidad:1000, precio:7500};  
  
tienda.push(nuevoProducto);  
  
for(iteracion=0; iteracion<numero_registro; iteracion++){  
  console.log(tienda[iteracion]);  
  if(tienda[iteracion].tipoProducto === 'fruver'){  
    listaBusqueda.push(tienda[iteracion]);  
  }  
  else{  
    console.log("??");  
  }  
}  
  
console.log(listaBusqueda)
```



# Tecnólogo en Análisis y Desarrollo de Software

## Ficha

### Arreglos Js

```
▼ {producto: 'Arroz', tipoProducto: 'granos', tipoUnidad: 'gramos', cantidad: 1000, precio: 2450} tienda.ts:22
  cantidad: 1000
  precio: 2450
  producto: "Arroz"
  tipoProducto: "granos"
  tipoUnidad: "gramos"
  ► [[Prototype]]: Object

?? tienda.ts:27
▼ {producto: 'Papa', tipoProducto: 'fruver', tipoUnidad: 'gramos', cantidad: 500, precio: 1000} tienda.ts:22
  cantidad: 500
  precio: 1000
  producto: "Papa"
  tipoProducto: "fruver"
  tipoUnidad: "gramos"
  ► [[Prototype]]: Object

▼ {producto: 'Mora', tipoProducto: 'fruver', tipoUnidad: 'gramos', cantidad: 500, precio: 1500} tienda.ts:22
  cantidad: 500
  precio: 1500
  producto: "Mora"
  tipoProducto: "fruver"
  tipoUnidad: "gramos"
  ► [[Prototype]]: Object

▼ {producto: 'Trucha', tipoProducto: 'carnes', tipoUnidad: 'gramos', cantidad: 1000, precio: 9000} tienda.ts:22
  cantidad: 1000
  precio: 9000
  producto: "Trucha"
  tipoProducto: "carnes"
  tipoUnidad: "gramos"
  ► [[Prototype]]: Object

?? tienda.ts:27
▼ {producto: 'Pollo Entero', tipoProducto: 'carnes', tipoUnidad: 'gramos', cantidad: 1000, precio: 4500} tienda.ts:22
  cantidad: 1000
  precio: 4500
  producto: "Pollo Entero"
  tipoProducto: "carnes"
  tipoUnidad: "gramos"
  ► [[Prototype]]: Object

?? tienda.ts:27
▼ (2) [-], [-] tienda.ts:31
  0: {producto: 'Papa', tipoProducto: 'fruver', tipoUnidad: 'gramos', cantidad: 500, precio: 1000}
  1: {producto: 'Mora', tipoProducto: 'fruver', tipoUnidad: 'gramos', cantidad: 500, precio: 1500}
  length: 2
  ► [[Prototype]]: Array(0)
```

Nombre de la función: holaMundo()		Versión: 1.0
Descripción: Bingo		
matriz	Int	
matriz1	Int	
matriz2	Int	
matriz3	Int	
numero	Int	
fila	Int	
columna	Int	
Código:		



# Tecnólogo en Análisis y Desarrollo de Software

## Ficha

### Arreglos Js

```
let matriz=[];
let matriz1=[];
let matriz2=[];
let matriz3=[];
let numero=1;

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

console.log("B"+"\\n"+"\\n");
for(let fila = 0; fila < 5; fila++) {
  console.log(matriz[fila][0]);
}

console.log("I"+"\\n"+"\\n");
for(let fila = 0; fila < 5; fila++) {
  console.log(matriz[fila][1]);
}

console.log("W"+"\\n"+"\\n");
for(let fila = 0; fila < 5; fila++) {
  console.log(matriz[fila][2]);
}

console.log("G"+"\\n"+"\\n");
for(let fila = 0; fila < 5; fila++) {
  console.log(matriz[fila][3]);
}

console.log("O"+"\\n"+"\\n");
for(let fila = 0; fila < 5; fila++) {
  console.log(matriz[fila][4]);
}

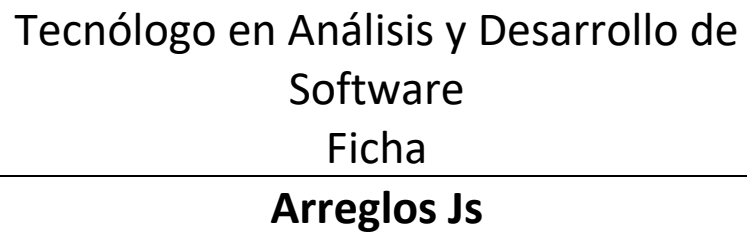
for(let fila=0;fila<3;fila++){
  for(let columna=0;columna<3;columna++){
    if(fila==columna || fila+columna==2){
      matriz1.push(matriz[fila][columna]);
    }
  }
}

for(let fila=2;fila<5;fila++){
  for(let columna=0;columna<3;columna++){
    if(fila+columna==(columna+1)*2 || fila+columna==4){
      matriz2.push(matriz[fila][columna]);
    }
  }
}

for(let fila=0; fila<3;fila++){
  for(let columna=2; columna<5;columna++){
    if(columna+fila==(fila+1)*2||fila+columna==4){
      matriz3.push(matriz[fila][columna]);
    }
  }
}

console.log(matriz)
console.log(matriz1)
console.log(matriz2)
console.log(matriz3)
```





8	<a href="#">Integer 2x 32</a>
16	<a href="#">Integer 2x 32</a>
32	<a href="#">Integer 2x 32</a>
64	<a href="#">Integer 2x 32</a>
128	<a href="#">Integer 2x 32</a>
256	<a href="#">Integer 2x 32</a>
512	<a href="#">Integer 2x 32</a>
1024	<a href="#">Integer 2x 32</a>
2048	<a href="#">Integer 2x 32</a>
4096	<a href="#">Integer 2x 32</a>
8192	<a href="#">Integer 2x 32</a>
16384	<a href="#">Integer 2x 32</a>
32768	<a href="#">Integer 2x 32</a>
65536	<a href="#">Integer 2x 32</a>
131072	<a href="#">Integer 2x 32</a>
262144	<a href="#">Integer 2x 32</a>
524288	<a href="#">Integer 2x 32</a>
1048576	<a href="#">Integer 2x 32</a>
2097152	<a href="#">Integer 2x 32</a>
4194304	<a href="#">Integer 2x 32</a>
8388608	<a href="#">Integer 2x 32</a>
16777216	<a href="#">Integer 2x 32</a>
33554432	<a href="#">Integer 2x 32</a>
67108864	<a href="#">Integer 2x 32</a>
134217728	<a href="#">Integer 2x 32</a>
268435456	<a href="#">Integer 2x 32</a>
536870912	<a href="#">Integer 2x 32</a>
1073741824	<a href="#">Integer 2x 32</a>
2147483648	<a href="#">Integer 2x 32</a>
4294967296	<a href="#">Integer 2x 32</a>
8589934592	<a href="#">Integer 2x 32</a>
17179869184	<a href="#">Integer 2x 32</a>
34359738368	<a href="#">Integer 2x 32</a>
68719476736	<a href="#">Integer 2x 32</a>
137438953472	<a href="#">Integer 2x 32</a>
274877906944	<a href="#">Integer 2x 32</a>
549755813888	<a href="#">Integer 2x 32</a>
1099511627776	<a href="#">Integer 2x 32</a>
2199023255552	<a href="#">Integer 2x 32</a>
4398046511104	<a href="#">Integer 2x 32</a>
8796093022208	<a href="#">Integer 2x 32</a>
17592186044416	<a href="#">Integer 2x 32</a>
35184372088832	<a href="#">Integer 2x 32</a>
70368744177664	<a href="#">Integer 2x 32</a>
140737488355328	<a href="#">Integer 2x 32</a>
281474976710656	<a href="#">Integer 2x 32</a>
562949953421312	<a href="#">Integer 2x 32</a>
1125899906842624	<a href="#">Integer 2x 32</a>
2251799813685248	<a href="#">Integer 2x 32</a>
4503599627370496	<a href="#">Integer 2x 32</a>
9007199254740992	<a href="#">Integer 2x 32</a>
18014398509481984	<a href="#">Integer 2x 32</a>
36028797018963968	<a href="#">Integer 2x 32</a>
72057594037927936	<a href="#">Integer 2x 32</a>
144115188075855872	<a href="#">Integer 2x 32</a>
288230376151711744	<a href="#">Integer 2x 32</a>
576460752303423488	<a href="#">Integer 2x 32</a>
1152921504606846976	<a href="#">Integer 2x 32</a>
2305843009213693952	<a href="#">Integer 2x 32</a>
4611686018427387904	<a href="#">Integer 2x 32</a>
9223372036854775808	<a href="#">Integer 2x 32</a>
18446744073709551616	<a href="#">Integer 2x 32</a>
36893488147419103232	<a href="#">Integer 2x 32</a>
73786976294838206464	<a href="#">Integer 2x 32</a>
147573952589676412928	<a href="#">Integer 2x 32</a>
295147905179352825856	<a href="#">Integer 2x 32</a>
590295810358705651712	<a href="#">Integer 2x 32</a>
1180591620717411303424	<a href="#">Integer 2x 32</a>
2361183241434822606848	<a href="#">Integer 2x 32</a>
4722366482869645213696	<a href="#">Integer 2x 32</a>
9444732965739290427392	<a href="#">Integer 2x 32</a>
18889465931478580854784	<a href="#">Integer 2x 32</a>
37778931862957161709568	<a href="#">Integer 2x 32</a>
75557863725914323419136	<a href="#">Integer 2x 32</a>
151115727451828646838272	<a href="#">Integer 2x 32</a>
302231454903657293676544	<a href="#">Integer 2x 32</a>
604462909807314587353088	<a href="#">Integer 2x 32</a>
1208925819614629174706176	<a href="#">Integer 2x 32</a>
2417851639229258349412352	<a href="#">Integer 2x 32</a>
4835703278458516698824704	<a href="#">Integer 2x 32</a>
9671406556917033397649408	<a href="#">Integer 2x 32</a>
19342813113834066795298816	<a href="#">Integer 2x 32</a>
38685626227668133590597632	<a href="#">Integer 2x 32</a>
77371252455336267181195264	<a href="#">Integer 2x 32</a>
154742504910672534362390528	<a href="#">Integer 2x 32</a>
309485009821345068724781056	<a href="#">Integer 2x 32</a>
618970019642690137449562112	<a href="#">Integer 2x 32</a>
1237940039285380274899124224	<a href="#">Integer 2x 32</a>
2475880078570760549798248448	<a href="#">Integer 2x 32</a>
4951760157141521099596496896	<a href="#">Integer 2x 32</a>
9903520314283042199192993792	<a href="#">Integer 2x 32</a>
19807040628566084398385987584	<a href="#">Integer 2x 32</a>
39614081257132168796771975168	<a href="#">Integer 2x 32</a>
79228162514264337593543950336	<a href="#">Integer </a>