



Tecnólogo en Análisis y Desarrollo de
Software
Ficha 2900177

ARRAYS

PRESENTADO POR:

KAROL NATALIA OSORIO POVEDA

INSTRUCTOR:

ANDRES MORENO

Neiva-Huila



Tecnólogo en Análisis y Desarrollo de Software

Ficha 2900177

ARRAYS

Nombre del Arrays: Factura

Versión: 1.0

Descripción:

El arreglo factura es una lista de objetos, donde cada objeto representa un producto comprado. Esta estructura facilita la visualización y gestión de los productos comprados y sus costos asociados.

valorTotalProducto Tipo de variable: number

iteracion Tipo de variable: number

Código:

```
let factura = [];  
let valorTotalProducto;  
let iteracion;  
let totalPago = [];  
  
factura=[  
  {codigo:1, nombreProducto:"Malteada", cantidad:2, valorUnidad:12000},  
  {codigo:2, nombreProducto:"Picada", cantidad:3, valorUnidad:25000},  
  {codigo:3, nombreProducto:"Hamburguesa Mixta", cantidad:4, valorUnidad:14000},  
  {codigo:4, nombreProducto:"Churrazco", cantidad:3, valorUnidad:25000},  
  {codigo:5, nombreProducto:"Gaseosa", cantidad:5, valorUnidad:5000},  
  {codigo:6, nombreProducto:"Limonada", cantidad:5, valorUnidad:6000},  
]  
  
console.log(factura);
```

```
▼ Array(6) 1  
  ▶ 0: {codigo: 1, nombreProducto: 'Malteada', cantidad: 2, valorUnidad: 12000}  
  ▶ 1: {codigo: 2, nombreProducto: 'Picada', cantidad: 3, valorUnidad: 25000}  
  ▶ 2: {codigo: 3, nombreProducto: 'Hamburguesa Mixta', cantidad: 4, valorUnidad: 14000}  
  ▶ 3: {codigo: 4, nombreProducto: 'Churrazco', cantidad: 3, valorUnidad: 25000}  
  ▶ 4: {codigo: 5, nombreProducto: 'Gaseosa', cantidad: 5, valorUnidad: 5000}  
  ▶ 5: {codigo: 6, nombreProducto: 'Limonada', cantidad: 5, valorUnidad: 6000}  
  length: 6  
  ▶ [[Prototype]]: Array(0)
```

Descripción:

Se calcula el valor total para el cuarto producto (Churrazco).

Código:

```
valorTotalProducto = factura[3].cantidad * factura[3].valorUnidad;  
console.log(factura[3].nombreProducto);  
console.log("Valor total: "+valorTotalProducto);
```

Churrazco

Valor total: 75000

Descripción:

Se itera sobre cada producto en el arreglo factura, calculando el total para cada uno y almacenando estos resultados en el arreglo totalPago:



Tecnólogo en Análisis y Desarrollo de Software

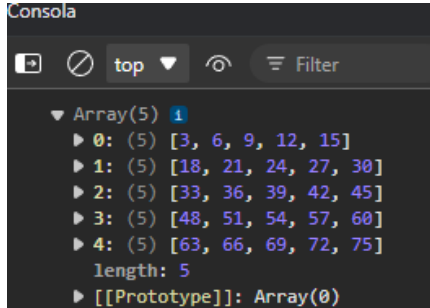
Ficha 2900177

ARRAYS

Código:

```
for (iteracion = 0; iteracion < factura.length; iteracion++) {  
    valorTotalProducto = factura[iteracion].cantidad * factura[iteracion].valorUnidad  
    totalPago.push({nombre: factura[iteracion].nombreProducto, cantidad: factura[iteracion].cantidad, valor: factura[iteracion].valorUnidad, total: valorTotalProducto})  
}  
console.log(totalPago);
```

```
▼ Array(6) 1  
▶ 0: {nombre: 'Malteada', cantidad: 2, valor: 12000, total: 24000}  
▶ 1: {nombre: 'Picada', cantidad: 3, valor: 25000, total: 75000}  
▶ 2: {nombre: 'Hamburguesa Mixta', cantidad: 4, valor: 14000, total: 56000}  
▶ 3: {nombre: 'Churrazco', cantidad: 3, valor: 25000, total: 75000}  
▶ 4: {nombre: 'Gaseosa', cantidad: 5, valor: 5000, total: 25000}  
▶ 5: {nombre: 'Limonada', cantidad: 5, valor: 6000, total: 30000}  
    length: 6  
    [[Prototype]]: Array(0)
```

Nombre del Arrays: Bingo[]		Versión: 1.0
Descripción: El código crea una matriz de 5x5 donde cada elemento es el contador multiplicado por 3.		
iteracion1	Tipo de variable: number	
iteracion2	Tipo de variable: number	
tabla	Tipo de variable: number	
contador	Tipo de variable: number	
Código:		
<pre>for(iteracion1=0; iteracion1<5; iteracion1++){ let interno=[]; for(iteracion2=0; iteracion2<5; iteracion2++){ contador =contador+1; tabla = contador*3; interno.push(tabla); } bingo.push(interno); } console.log(bingo);</pre>		<div>Consola</div> <div></div>

Nombre del Arrays: cantidadPares[], cantidadImpares[]	Versión: 1.0
Descripción:	



Tecnólogo en Análisis y Desarrollo de Software

Ficha 2900177

ARRAYS

Se recorren los elementos de la matriz bingo para separar los números pares e impares en los arreglos cantidadPares y cantidadImpares, respectivamente.

iteracion1 Tipo de variable: number

iteracion2 Tipo de variable: number

Código:

```
for(iteracion1=0; iteracion1<5; iteracion1++){
  for(iteracion2=0; iteracion2<5; iteracion2++){
    if(bingo[iteracion1][iteracion2]%2==0){
      cantidadPares.push(bingo[iteracion1][iteracion2]);
    }
    else{
      cantidadImpares.push(bingo[iteracion1][iteracion2]);
    }
  }
}
console.log("Pares: ",cantidadPares);
console.log("Impares: ",cantidadImpares);
```

Pares: ▼ Array(12) ⓘ

- 0: 6
- 1: 12
- 2: 18
- 3: 24
- 4: 30
- 5: 36
- 6: 42
- 7: 48
- 8: 54
- 9: 60
- 10: 66
- 11: 72

length: 12
▶ [[Prototype]]: Array(0)

Impares: ▼ Array(13) ⓘ

- 0: 3
- 1: 9
- 2: 15
- 3: 21
- 4: 27
- 5: 33
- 6: 39
- 7: 45
- 8: 51
- 9: 57
- 10: 63
- 11: 69
- 12: 75

length: 13
▶ [[Prototype]]: Array(0)

Nombre del Arrays: letraB[]

Versión: 1.0

Descripción:

Se extraen los números correspondientes a cada columna del bingo y se almacenan en los arreglos.

iteracion1 Tipo de variable: number

Código:

```
for(iteracion1=0; iteracion1<5; iteracion1++){
  letraB.push(bingo[iteracion1][0])
}
console.log("Letra B: "+letraB);
```

Letra B: 3,18,33,48,63

Nombre del Arrays: letal[]

Versión: 1.0

Descripción:

Se extraen los números correspondientes a cada columna del bingo y se almacenan en los arreglos.

iteracion1 Tipo de variable: number



Tecnólogo en Análisis y Desarrollo de Software

Ficha 2900177

ARRAYS

Código:

```
for(iteracion1=0; iteracion1<5; iteracion1++){  
    letraI.push(bingo[iteracion1][1])  
}  
console.log("Letra I: "+letraI);
```

Letra I: 6,21,36,51,66

Nombre del Arrays: letraN[]

Versión: 1.0

Descripción:

Se extraen los números correspondientes a cada columna del bingo y se almacenan en los arreglos.

iteracion1

Tipo de variable: number

Código:

```
for(iteracion1=0; iteracion1<5; iteracion1++){  
    letraN.push(bingo[iteracion1][2])  
}  
console.log("Letra N: "+letraN);
```

Letra N: 9,24,39,54,69

Nombre del Arrays: letraG[]

Versión: 1.0

Descripción:

Se extraen los números correspondientes a cada columna del bingo y se almacenan en los arreglos.

iteracion1

Tipo de variable: number

Código:

```
for(iteracion1=0; iteracion1<5; iteracion1++){  
    letraG.push(bingo[iteracion1][3])  
}  
console.log("Letra G: "+letraG);
```

Letra G: 12,27,42,57,72

Nombre del Arrays: letraO[]

Versión: 1.0

Descripción:



Tecnólogo en Análisis y Desarrollo de Software

Ficha 2900177

ARRAYS

Se extraen los números correspondientes a cada columna del bingo y se almacenan en los arreglos.

iteracion1	Tipo de variable: number
------------	--------------------------

Código:

```
for(iteracion1=0; iteracion1<5; iteracion1++){
  letra0.push(bingo[iteracion1][4])
}
console.log("Letra 0: "+letra0);
```

Letra 0: 15,30,45,60,75

Nombre del Arrays: equis[]	Versión: 1.0
-----------------------------------	--------------

Descripción:

Se extraen los números correspondientes para formar la primera X (6,24,42,12,36)

iteracion1	Tipo de variable: number
------------	--------------------------

iteracion2	Tipo de variable: number
------------	--------------------------

Código:

```
let equis = [];

for(iteracion1=0; iteracion1<3; iteracion1++){
  equis.push(bingo[iteracion1][iteracion1+1]);
}

for(iteracion2=0; iteracion2<3; iteracion2++){
  equis.push(bingo[iteracion2][4-(iteracion2+1)]);
}

console.log("X 1: "+equis);
```

X 1: 6,24,42,12,24,36

Nombre del Arrays: equis2[]	Versión: 1.0
------------------------------------	--------------

Descripción:

Se extraen los números correspondientes para formar la segunda X (33,51,69,39,63)

iteracion1	Tipo de variable: number
------------	--------------------------

iteracion2	Tipo de variable: number
------------	--------------------------

Código:

```
let equis2 = [];

for(iteracion1=0; iteracion1<3; iteracion1++){
  equis2.push(bingo[iteracion1+2][iteracion1]);
}

for(iteracion2=0; iteracion2<3; iteracion2++){
  equis2.push(bingo[iteracion2+2][4-(iteracion2+2)]);
}

console.log("X 2: "+equis2);
```

X 2: 33,51,69,39,51,63



Tecnólogo en Análisis y Desarrollo de Software

Ficha 2900177

ARRAYS

Nombre del Arrays: equis3[]		Versión: 1.0
Descripción: Se extraen los números correspondientes para formar la tercera X (39,57,75,45,69)		
iteracion1	Tipo de variable: number	
iteracion2	Tipo de variable: number	
Código:		
<pre>let equis3 =[]; for(iteracion1=0; iteracion1<3; iteracion1++){ equis3.push(bingo[iteracion1+2][iteracion1+2]); } for(iteracion2=0; iteracion2<3; iteracion2++){ equis3.push(bingo[iteracion2+2][4-iteracion2]); } console.log("X 3: "+equis3);</pre>		X 3: 39,57,75,45,57,69

Nombre del Arrays: xgrande[]		Versión: 1.0
Descripción:		
Se extraen los números correspondientes para formar la X Grande (3,21,39,57,75,15,27,51,63)		
iteracion1	Tipo de variable: number	
iteracion2	Tipo de variable: number	
Código:		
<pre>let xgrande = []; for(iteracion1=0; iteracion1<5; iteracion1++){ xgrande.push(bingo[iteracion1][iteracion1]); } for(iteracion2=0; iteracion2<5; iteracion2++){ xgrande.push(bingo[iteracion2][4-iteracion2]); } console.log("X GRANDE: "+xgrande);</pre>		X GRANDE: 3,21,39,57,75,15,27,39,51,63



Tecnólogo en Análisis y Desarrollo de Software

Ficha 2900177

ARRAYS

Nombre del Arrays: nomina[]	Versión: 1.0
Descripción: Colección de objetos que representan la información personal y laboral de varios empleados. Cada objeto dentro del arreglo contiene varios atributos que describen las características individuales de cada empleado.	
Código: <pre>nomina = [{id: 1075793094, nombres: "Karol Natalia", apellidos: "Osorio Poveda", edad: 17, estrato: 2, valorDia: 43330, diasTrabajados: 30}, {id: 1073723422, nombres: "Brayan Santiago", apellidos: "Guerrero Mendez", edad: 17, estrato: 6, valorDia: 43330, diasTrabajados: 30}, {id: 1075793093, nombres: "Camillo Andres", apellidos: "Losada Ramirez", edad: 17, estrato: 2, valorDia: 130000, diasTrabajados: 30 }, {id: 1074793094, nombres: "Ingrid Yulisa", apellidos: "Medina Esquivel", edad: 17, estrato: 3, valorDia: 218000, diasTrabajados: 30 }, {id: 1075723054, nombres: "Jesus Fernando", apellidos: "Carvajal Anacona", edad: 17, estrato: 1, valorDia: 304000, diasTrabajados: 30 }, {id: 1025694094, nombres: "Juan Manuel", apellidos: "Gutierrez", edad: 17, estrato: 6, valorDia: 304000, diasTrabajados: 30 }, {id: 1075219122, nombres: "Yerson Stiven", apellidos: "Cuellar Rubiano", edad: 17, estrato: 6, valorDia: 350000, diasTrabajados: 30 }, {id: 1076503347, nombres: "Mauricio", apellidos: "Noscue", edad: 17, estrato: 6, valorDia: 350000, diasTrabajados: 30 }, {id: 1076490275, nombres: "Maria Isabel", apellidos: "Puentes", edad: 17, estrato: 1, valorDia: 43330, diasTrabajados: 30 }, {id: 1075764001, nombres: "Jesus David", apellidos: "Fierro", edad: 17, estrato: 4, valorDia: 43330, diasTrabajados: 30 },] console.log(nomina);</pre> <pre>▼ Array(10) 0: {id: 1075793094, nombres: 'Karol Natalia', apellidos: 'Osorio Poveda', edad: 17, estrato: 2, ...} 1: {id: 1073723422, nombres: 'Brayan Santiago', apellidos: 'Guerrero Mendez', edad: 17, estrato: 6, ...} 2: {id: 1075793093, nombres: 'Camillo Andres', apellidos: 'Losada Ramirez', edad: 17, estrato: 2, ...} 3: {id: 1074793094, nombres: 'Ingrid Yulisa', apellidos: 'Medina Esquivel', edad: 17, estrato: 3, ...} 4: {id: 1075723054, nombres: 'Jesus Fernando', apellidos: 'Carvajal Anacona', edad: 17, estrato: 1, ...} 5: {id: 1025694094, nombres: 'Juan Manuel', apellidos: 'Gutierrez', edad: 17, estrato: 6, ...} 6: {id: 1075219122, nombres: 'Yerson Stiven', apellidos: 'Cuellar Rubiano', edad: 17, estrato: 6, ...} 7: {id: 1076503347, nombres: 'Mauricio', apellidos: 'Noscue', edad: 17, estrato: 6, ...} 8: {id: 1076490275, nombres: 'Maria Isabel', apellidos: 'Puentes', edad: 17, estrato: 1, ...} 9: {id: 1075764001, nombres: 'Jesus David', apellidos: 'Fierro', edad: 17, estrato: 4, ...} length: 10 [[Prototype]]: Array(0)</pre>	

Nombre del Arrays: totalPagos[]	Versión: 1.0
Descripción: Se calcula el pago total de cada persona de la nomina teniendo en cuenta lo siguiente: <ul style="list-style-type: none">• si la persona gana menos de dos salarios mínimos se sumara a su sueldo el subsidio de transporte, de lo contrario sumara 0.• Si la persona gana un salario Mínimo o menos y si su estrato es 1 o 2 se abonara a su sueldo 100.000, de lo contrario no.• Si la persona gana mas de 4 salarios mínimos se hará una retención del 0.03• Si la persona gana mas de 6 salarios mínimos se hará una retención del 0.04• Si la persona gana mas de 8 salarios mínimos y su estrato es 6 se hará una retención del 0.05 Para realizar esto, haremos uso de las siguientes funciones:	
Nombre de la Función: sueldo(pdiasTrab, pvalorDia)	Versión: 1.0
Descripción: Calcula el salario bruto de un empleado basado en los días trabajados y el valor diario de su salario.	
diasTrab	Tipo de variable: int
valorDia	Tipo de variable: int
salario	Tipo de variable: int



Tecnólogo en Análisis y Desarrollo de Software

Ficha 2900177

ARRAYS

Código:

```
function sueldo(pdiasTrab, pvalorDia){  
  let diasTrab = pdiasTrab;  
  let valorDia = pvalorDia;  
  let salario;  
  salario= diasTrab*valorDia;  
  return salario;  
}
```

Salario: 1299900

Nombre de la Función: salud(psalarario)

Versión: 1.0

Descripción:

Calcular la contribución a la salud basada en el salario del empleado.

salario	Tipo de variable: int
---------	-----------------------

sal	Tipo de variable: int
-----	-----------------------

Código:

```
function salud(psalarario){  
  let salario = psalarario;  
  let sal;  
  sal = salario * 0.12  
  return sal;  
}
```

Salud: 155988

Nombre de la Función: pension(psalarario)

Versión: 1.0

Descripción:

Calcular la contribución a la pensión basada en el salario del empleado.

salario	Tipo de variable: int
---------	-----------------------

pensi	Tipo de variable: int
-------	-----------------------

Código:

```
function pension(psalarario){  
  let salario = psalarario;  
  let pensi;  
  pensi = salario * 0.16  
  return pensi;  
}
```

Pension: 207984

Nombre de la Función: arl(psalarario)

Versión: 1.0

Descripción:

Calcular la contribución a la ARL basada en el salario del empleado.

salario	Tipo de variable: int
---------	-----------------------

arlP	Tipo de variable: float
------	-------------------------

Código:



Tecnólogo en Análisis y Desarrollo de Software

Ficha 2900177

ARRAYS

```
function arl(psalario){  
  let salario = psalario;  
  let arlP;  
  arlP = salario * 0.052  
  return arlP;  
}
```

Arl: 67594.8

Nombre de la Función: subTraspor(salario)

Versión: 1.0

Descripción:

Calcula el subsidio de transporte basado en el salario del empleado.

salarioM Tipo de variable: int

subTraspor Tipo de variable: int

Código:

```
function subTraspor(salario){  
  let salarioM= 1300000  
  let subTraspor;  
  
  if(salario < 2*salarioM){  
    subTraspor = 114000;  
  }else{  
    subTraspor = 0  
  }  
  return subTraspor;  
}
```

SubsidioTransporte: 114000

Nombre de la Función: abonos(salario, estrato)

Versión: 1.0

Descripción:

Calcula los abonos adicionales basados en el salario y el estrato socioeconómico del empleado.

estrato Tipo de variable: int

salarioM Tipo de variable: int

abono Tipo de variable: int

Código:

```
let estrato;  
  
function abonos(salario, estrato){  
  let salarioM= 1300000;  
  let abono;  
  
  if(salario <=salarioM && estrato ==1 ){  
    abono = 100000;  
  }else{  
    if(salario <=salarioM && estrato ==2 ){  
      abono = 100000;  
    }else{  
      abono = 0;  
    }  
  }  
  return abono;  
}
```

Abono: 100000



Tecnólogo en Análisis y Desarrollo de Software

Ficha 2900177

ARRAYS

Nombre de la Función: retencionn(salario, estrato)

Versión: 1.0

Descripción:

Calcular las retenciones aplicables al salario del empleado basadas en el nivel de salario y el estrato socioeconómico.

salario	Tipo de variable: int
retencion	Tipo de variable: int
estrato	Tipo de variable: int
salarioM	Tipo de variable: int

Código:

```
function retencionn(salario, estrato){
  let retencion;
  let salarioM = 1300000;

  if(salario>4*salarioM && salario<6*salarioM){
    retencion = salario * 0.03
  }else{
    if(salario>8*salarioM && estrato ==6){
      retencion =salario * 0.05
    }else{
      if(salario>6*salarioM){
        retencion = salario*0.04
      }else{
        retencion = 0
      }
    }
  }
  return retencion;
}
```

Retencion: 0

Array:

Estas funciones se utilizan en el bucle que recorre el arreglo nomina para calcular el salario neto y los distintos descuentos y subsidios aplicables a cada empleado, luego almacenar estos cálculos en el arreglo totalPagos.

Código:

```
for (iteracion = 0; iteracion < nomina.length; iteracion++) {
  let sueldo1 = sueldo[nomina[iteracion].diasTrabajados, nomina[iteracion].valorDia];
  let salud1 = salud(sueldo1);
  let pension1 = pension(sueldo1);
  let arl1 = arl(sueldo1);
  let retencion1 = retencionn(sueldo1, nomina[iteracion].estrato);
  let subsidio = subTraspor(sueldo1);
  let abono = abonos(sueldo1, nomina[iteracion].estrato);
  let descuento = salud1+pension1+arl1;
  let total = (sueldo1+subsidio+abono)-(retencion1+descuento);
  totalPagos.push({Id: nomina[iteracion].id, Nombres: nomina[iteracion].nombres, Apellidos: nomina[iteracion].apellidos, Edad: nomina[iteracion].edad, Estrato: nomina[iteracion].estrato, ValorDia: nomina[iteracion].valorDia, diasTrabajados: nomina[iteracion].diasTrabajados, Salario: sueldo1, Salud: salud1, Pension: pension1, Arl: arl1, Retencion: retencion1, SubsidioTransporte: subsidio, Abono: abono, Descuento: descuento, SueldoTotal: total});
}
console.table(totalPagos);
```

```
▼ Array(10)
  0: {Id: 1075793094, Nombres: 'Karol Natalia', Apellidos: 'Osorio Poveda', Edad: 17, Estrato: 2, ...}
  1: {Id: 1073723422, Nombres: 'Brayan Santiago', Apellidos: 'Guerrero Mendez', Edad: 17, Estrato: 6, ...}
  2: {Id: 1075793093, Nombres: 'Camilo Andres', Apellidos: 'Losada Ramirez', Edad: 17, Estrato: 2, ...}
  3: {Id: 1074793094, Nombres: 'Ingrid Yulisa', Apellidos: 'Medina Esquivel', Edad: 17, Estrato: 3, ...}
  4: {Id: 1075723054, Nombres: 'Jesús Fernando', Apellidos: 'Carvajal Anaconda', Edad: 17, Estrato: 1, ...}
  5: {Id: 1025694094, Nombres: 'Juan Manuel', Apellidos: 'Gutierrez', Edad: 17, Estrato: 6, ...}
  6: {Id: 1075219122, Nombres: 'Yerson Stiven', Apellidos: 'Cuellar Rubiano', Edad: 17, Estrato: 6, ...}
  7: {Id: 1076503347, Nombres: 'Mauricio', Apellidos: 'Noscue', Edad: 17, Estrato: 6, ...}
  8: {Id: 1076490275, Nombres: 'Maria Isabel', Apellidos: 'Puentes', Edad: 17, Estrato: 1, ...}
  9: {Id: 1075764001, Nombres: 'Jesus David', Apellidos: 'Fierro', Edad: 17, Estrato: 4, ...}
  length: 10
  [[Prototype]]: Array(0)
```

```
▼ 6:
  Abono: 0
  Apellidos: "Cuellar Rubiano"
  Arl: 546000
  Descuento: 3486000
  DiasTrabajados: 30
  Edad: 17
  Estrato: 6
  Id: 1075219122
  Nombres: "Yerson Stiven"
  Pension: 1680000
  Retencion: 525000
  Salario: 10500000
  Salud: 1260000
  SubsidioTransporte: 0
  SueldoTotal: 6489000
  ValorDia: 350000
  [[Prototype]]: Object
```

```
▼ 0:
  Abono: 100000
  Apellidos: "Osorio Poveda"
  Arl: 67594.8
  Descuento: 431566.8
  DiasTrabajados: 30
  Edad: 17
  Estrato: 2
  Id: 1075793094
  Nombres: "Karol Natalia"
  Pension: 207984
  Retencion: 0
  Salario: 1299900
  Salud: 155988
  SubsidioTransporte: 114000
  SueldoTotal: 1082333.2
  ValorDia: 43330
  [[Prototype]]: Object
```



Tecnólogo en Análisis y Desarrollo de
Software

Ficha 2900177

ARRAYS