



Tecnólogo en Análisis y Desarrollo de  
Software  
Ficha 2900177

**Funciones JS**

**PRESENTADO POR:**

**KAROL NATALIA OSORIO POVEDA**

**INSTRUCTOR:**

**ANDRES MORENO**

**Neiva-Huila**



# Tecnólogo en Análisis y Desarrollo de Software

## Ficha 2900177

### Funciones JS

1.

Nombre de la función: saludo(psaludo)		Versión: 1.0
Descripción: Función que saluda como parámetros		
saludar	Tipo de variable: Alfanumérico	
Código:		
<pre>function saludo(psaludo){ //como parametros   let saludar = psaludo;   return saludar; }</pre>		<pre>Parametros: Hola Mundo!</pre>

1.1

<b>Nombre de la función:</b> const saludoExp = function(psaludo)		<b>Versión: 2.0</b>
<b>Descripción:</b> Función que saluda como expresión		
saludar	Tipo de variable: alfanumérico	
Código:		
<pre>const saludoExp = function(psaludo){ //como Expresión   let saludar = psaludo;   return saludar; }</pre>		<pre>Como una Expresion: Hola Mundo!</pre>

2.

<b>Nombre de la función:</b> <b>suma(pnumUno, pnumDos)</b>		<b>Versión: 1.0</b>
<b>Descripción:</b> Función operaciones Aritméticas Como parámetro		
numUno	Tipo de variable: Int	
numDos	Tipo de variable: Int	
sumar	Tipo de variable: Int	



# Tecnólogo en Análisis y Desarrollo de Software

## Ficha 2900177

### Funciones JS

Código:

```
function suma(pnumUno, pnumDos){  
    let sumar;  
    let numUno = pnumUno;  
    let numDos = pnumDos;  
    sumar = numUno + numDos;  
    return sumar;  
}
```

<b>Nombre de la función:</b> <b>resta(pnumUno, pnumDos)</b>		<b>Versión: 1.0</b>
<b>Descripción:</b> Función operaciones Aritméticas Parámetros		
numUno	Tipo de variable: Int	
numDos	Tipo de variable: Int	
restar	Tipo de variable: Int	
<b>Código:</b>		<pre>function resta(pnumUno, pnumDos){     let restar;     let numUno = pnumUno;     let numDos = pnumDos;     restar = numUno - numDos;     return restar; }</pre>

<b>Nombre de la función:</b> <b>multiplicacion(pnumUno, pnumDos)</b>		<b>Versión: 1.0</b>
<b>Descripción:</b> Función operaciones Aritméticas Parámetros		
numUno	Tipo de variable: Int	
numDos	Tipo de variable: Int	
multiplicar	Tipo de variable: Int	



# Tecnólogo en Análisis y Desarrollo de Software

## Ficha 2900177

### Funciones JS

Código:

```
function multiplicacion(pnumUno, pnumDos){  
    let multiplicar;  
    let numUno = pnumUno;  
    let numDos = pnumDos;  
    multiplicar = numUno * numDos;  
    return multiplicar;  
}
```

<b>Nombre de la función:</b> <b>division(pnumUno, pnumDos)</b>		<b>Versión: 1.0</b>
<b>Descripción:</b> Función operaciones Aritméticas Parámetros		
numUno	Tipo de variable: Int	
numDos	Tipo de variable: Int	
dividir	Tipo de variable: Int	
Código:		
<pre>function division(pnumUno, pnumDos){     let dividir;     let numUno = pnumUno;     let numDos = pnumDos;     dividir = numUno / numDos;     return dividir; }</pre>		

<b>Nombre de la función:</b> <b>operaciones(poperador, pnumUno, pnumDos)</b>		<b>Versión: 1.0</b>
<b>Descripción:</b> Función operaciones Aritméticas con Parámetros		
operador	Tipo de variable: string	
numUno	Tipo de variable: Int	
numDos	Tipo de variable: Int	



# Tecnólogo en Análisis y Desarrollo de Software

Ficha 2900170177

## Funciones JS

Código:

```
function operaciones(poperador, pnumUno, pnumDos){
  let operador = poperador;
  let numUno = pnumUno;
  let numDos = pnumDos;
  if(operador=="suma"){
    return suma(numUno,numDos);
  }
  else if(operador=="resta"){
    return resta(numUno,numDos);
  }
  else if(operador=="multiplicacion"){
    return multiplicacion(numUno,numDos);
  }
  else if(operador=="division"){
    return division(numUno,numDos);
  }
  else{
    return "Error!!! no reconoce operador";
  }
}
```

```
Suma Uno: 5
Resta Uno: 1
Multiplicación Uno: 6
División Uno: 1.5
Suma Dos: 6
Multiplicación Dos: 9
```

2.1

Nombre de la función:

**const sumaExp = function(pnumUno, pnumDos)**

Versión: 2.0

Descripción:

Función operaciones Aritméticas como Expresión

numUno Tipo de variable: Int

numDos Tipo de variable: Int

sumar Tipo de variable: Int

Código:

```
const sumaExp = function(pnumUno, pnumDos){
  let sumar;
  let numUno = pnumUno;
  let numDos = pnumDos;
  sumar = numUno + numDos;
  return sumar;
}
```

Nombre de la función:

**const restaExp = function(pnumUno, pnumDos)**

Versión: 2.0

Descripción:

Función operaciones Aritméticas como Expresión

numUno Tipo de variable: Int

numDos Tipo de variable: Int

restar Tipo de variable: Int



# Tecnólogo en Análisis y Desarrollo de Software

## Ficha 2900177

### Funciones JS

Código:

```
const restaExp= function(pnumUno, pnumDos){  
    numUno = pnumUno;  
    numDos = pnumDos;  
    let restar;  
    restar = numUno - numDos;  
    return restar;  
}
```

<b>Nombre de la función:</b> const multiplicacionExp = function(pnumUno, pnumDos)		<b>Versión: 2.0</b>
<b>Descripción:</b> Función operaciones Aritméticas como Expresión		
numUno	Tipo de variable: Int	
numDos	Tipo de variable: Int	
multiplicar	Tipo de variable: Int	
<b>Código:</b> <pre>const multiplicacionExp = function(pnumUno, pnumDos){     let multiplicar;     let numUno = pnumUno;     let numDos = pnumDos;     multiplicar = numUno * numDos;     return multiplicar; }</pre>		

<b>Nombre de la función:</b> const divisionExp = function(pnumUno, pnumDos)		<b>Versión: 2.0</b>
<b>Descripción:</b> Función operaciones Aritméticas como Expresión		
numUno	Tipo de variable: Int	
numDos	Tipo de variable: Int	
dividir	Tipo de variable: float	
<b>Código:</b> <pre>const divisionExp = function(pnumUno, pnumDos){     let dividir;     let numUno = pnumUno;     let numDos = pnumDos;     dividir = numUno / numDos;     return dividir; }</pre>		



# Tecnólogo en Análisis y Desarrollo de Software

## Ficha 2900177

### Funciones JS

<b>Nombre de la función:</b> const operacionesExp = function(pnumUno, pnumDos)		<b>Versión: 2.0</b>
<b>Descripción:</b> Función operaciones Aritméticas como Expresión		
numUno	Tipo de variable: Int	
numDos	Tipo de variable: Int	
operador	Tipo de variable: string	
<b>Código:</b>		
<pre>const operacionesExp = function(poperador, pnumUno, pnumDos){   let operador = poperador;   let numUno = pnumUno;   let numDos = pnumDos;   if(operador=="sumaExp"){     return sumaExp(numUno,numDos);   }   else if(operador=="restaExp"){     return restaExp(numUno,numDos);   }   else if(operador=="multiplicacionExp"){     return multiplicacionExp(numUno,numDos);   }   else if(operador=="divisionExp"){     return divisionExp(numUno,numDos);   }   else{     return "Error!!! no reconoce operador";   } }</pre>		<div>Suma Uno Expresión: 5</div> <div>Resta Uno Expresión: 1</div> <div>multiplicacion Uno Expresión : 6</div> <div>Division Uno Expresión: 1.5</div> <div>Suma Dos Expresión: 6</div> <div>Multiplicacion Dos Expresión: 9</div>

```
Suma Uno Expresión: 5
Resta Uno Expresión: 1
multiplicacion Uno Expresión : 6
Division Uno Expresión: 1.5
Suma Dos Expresión: 6
Multiplicacion Dos Expresión: 9
```

3.

<b>Nombre de la función:</b> porcent(pnum)		<b>Versión: 1.0</b>
<b>Descripción:</b> Función que calcula el Porcentaje de un numero con Parámetros		
num	Tipo de variable: Int	
porcentaje	Tipo de variable: float	
Código:		
<pre>//con parametros  function porcent(pnum){   let num = pnum;   let porcentaje;   porcentaje = num / 100;   return porcentaje; }</pre>		<pre>Porcentaje de un número con Parametros: 0.5</pre>

```
Porcentaje de un número con Parametros: 0.5
```



# Tecnólogo en Análisis y Desarrollo de Software

## Ficha 2900177

### Funciones JS

3.1

<b>Nombre de la función:</b> const percentExp = function(pnum)		<b>Versión: 2.0</b>
<b>Descripción:</b> Función que calcula el Porcentaje de un numero como Expresión		
num	Tipo de variable: Int	
porcentaje	Tipo de variable: float	
<b>Código:</b>		
<pre>//como Expresión const percentExp = function(pnum){   let num = pnum;   let porcentaje;   porcentaje = num / 100;   return porcentaje; }</pre>		<b>Porcentaje de un número con Expresión: 0.5</b>

Porcentaje de un número con Expresión: 0.5

4.

<b>Nombre de la función:</b> <b>notas(pnota)</b>		<b>Versión: 1.0</b>
<b>Descripción:</b> Función que calcula el Promedio de tres notas con parámetros		
nota	Tipo de variable: float	
Código: <pre>function notas(pnota){     let nota = pnota;     return nota; }</pre>		

<b>Nombre de la función:</b> <b>prom(pnota1, pnota2, pnota3)</b>		<b>Versión: 1.0</b>
<b>Descripción:</b> Función que calcula el Promedio de tres notas con parámetros		
nota1	Tipo de variable: float	
nota2	Tipo de variable: float	
nota3	Tipo de variable: float	
promedio	Tipo de variable: float	
Código:		





# Tecnólogo en Análisis y Desarrollo de Software

## Ficha 2900177

### Funciones JS

```
function prom(pnota1, pnota2, pnota3){  
  let nota1 = pnota1  
  let nota2 = pnota2  
  let nota3 = pnota3  
  nota1 = notas(pnota1);  
  nota2 = notas(pnota2);  
  nota3 = notas(pnota3);  
  let promedio;  
  promedio = (nota1+nota2+nota3)/3  
  return promedio;  
}
```

```
Nota 1 Parametro: 4.7  
Nota 2 Parametro: 4.6  
Nota 3 Parametro: 2.6  
Promedio: 3.966666666666667
```

#### 4.1

<b>Nombre de la función:</b> <b>const notasExp = function(pnota)</b>		<b>Versión: 2.0</b>
<b>Descripción:</b> Función que calcula el Promedio de tres notas como Expresión		
nota	Tipo de variable: float	
Código:	<pre>const notasExp = function(pnota){   let nota = pnota;   return nota; }</pre>	

<b>Nombre de la función:</b> <b>promedioExp(pnota1, pnota2, pnota3)</b>		<b>Versión: 2.0</b>
<b>Descripción:</b> Función que calcula el Promedio de tres notas como Expresión		
nota1	Tipo de variable: float	
nota2	Tipo de variable: float	
nota3	Tipo de variable: float	
promedio	Tipo de variable: float	
Código:		



# Tecnólogo en Análisis y Desarrollo de Software

## Ficha 2900177

### Funciones JS

```
const promedioExp = function(pnota1, pnota2, pnota3){  
  let nota1 = pnota1;  
  let nota2 = pnota2;  
  let nota3 = pnota3;  
  nota1 = notasExp(pnota1);  
  nota2 = notasExp(pnota2);  
  nota3 = notasExp(pnota3);  
  let promedio;  
  promedio = (nota1+nota2+nota3)/3  
  return promedio;  
}
```

```
Nota 1 Exp: 2.8  
Nota 2 Exp: 2.6  
Nota 3 Exp: 4.6  
Promedio: 3.3333333333333335
```

5.

<b>Nombre de la función:</b> <b>porcNotas(pnota, pporc)</b>		<b>Versión: 1.0</b>
<b>Descripción:</b> Función que calcula la suma de los porcentajes de 3 notas con Parámetros		
nota	Tipo de variable: float	
porc	Tipo de variable: float	
result	Tipo de variable: float	
<b>Código:</b>		
<pre>//con parametros function porcNotas(pnota, pporc){   let nota = pnota;   let porc = pporc;   let result;   result = nota * porc;   return result; }</pre>		
<pre>Porcentaje de la nota 1: 1.08 Porcentaje de la nota 2: 1.17 Porcentaje de la nota 3: 1.16 Suma de los porcentajes: 3.41</pre>		

5.1

<b>Nombre de la función:</b> <b>const porcExp = function(pnota, pporc)</b>		<b>Versión: 2.0</b>
<b>Descripción:</b> Función que calcula la suma de los porcentajes de 3 notas como expresión		
nota	Tipo de variable: float	
porc	Tipo de variable: float	
result	Tipo de variable: float	



# Tecnólogo en Análisis y Desarrollo de Software

## Ficha 2900177

### Funciones JS

Código:

```
//como Expresion
const porcExp = function(pnota, pporc){
  let nota = pnota;
  let porc = pporc;
  let result;
  result = nota * porc;
  return result;
}
```

Porcentaje de la nota 1 Exp: 1.32

Porcentaje de la nota 2 Exp: 0.87

Porcentaje de la nota 3 Exp: 0.76

Suma de los porcentajes Exp: 2.95

6.

Nombre de la función:  
**areaCuadrado(plado)**

Versión: 1.0

Descripción:

Función que calcula las áreas de figuras geométricas con Parámetros

lado      Tipo de variable: Int

areaC      Tipo de variable: Int

Código:

```
function areaCuadrado(plado){
  let areaC;
  let lado = plado;
  areaC = lado*lado;
  return areaC;
}
```

Area del cuadrado Parametros: 81

Nombre de la función:

**areaRectangulo(pbaseRec, palturaRec)**

Versión: 1.0

Descripción:

Función que calcula las áreas de figuras geométricas con Parámetros

baseRec      Tipo de variable: Int

alturaRec      Tipo de variable: Int

areaR      Tipo de variable: Int

Código:

```
function areaRectangulo(pbaseRec, palturaRec){
  let areaR;
  let baseRec = pbaseRec;
  let alturaRec = palturaRec;
  areaR = baseRec*alturaRec;
  return areaR;
}
```

Area Rectangulo Parametros: 72

Nombre de la función:

Versión: 1.0



# Tecnólogo en Análisis y Desarrollo de Software

## Ficha 2900177

### Funciones JS

<b>areaTriangulo(pbaseTri, palturaTri)</b>	
<b>Descripción:</b> Función que calcula las áreas de figuras geométricas con Parámetros	
baseTri	Tipo de variable: Int
alturaTri	Tipo de variable: Int
areaT	Tipo de variable: Int
<b>Código:</b> <pre>function areaTriangulo(pbaseTri, palturaTri){   let areaT;   let baseTri = pbaseTri;   let alturaTri = palturaTri;   areaT = (baseTri*alturaTri)/2;   return areaT; }</pre>	
Area del Triangulo Parametros: 36	

6.1

<b>Nombre de la función:</b> const areaExpc = function(plado)		<b>Versión: 2.0</b>
<b>Descripción:</b> Función que calcula las áreas de figuras geométricas como expresión		
lado	Tipo de variable: Int	
areaC	Tipo de variable: Int	
<b>Código:</b> <pre>//como Expresión const areaExpc = function(plado){   let areaC;   let lado = plado;   areaC = lado*lado;   return areaC; }</pre>		
Area del cuadrado Exp: 81		

<b>Nombre de la función:</b> <b>const areaExpr = function(pbaseRec, palturaRec)</b>		<b>Versión: 2.0</b>
<b>Descripción:</b> Función que calcula las áreas de figuras geométricas como expresión		
baseRec	Tipo de variable: Int	
alturaRec	Tipo de variable: Int	
areaR	Tipo de variable: Int	



# Tecnólogo en Análisis y Desarrollo de Software

## Ficha 2900177

### Funciones JS

Código:

```
const areaExpr = function(pbaseRec, palturaRec){  
  let areaR;  
  let baseRec = pbaseRec;  
  let alturaRec = palturaRec;  
  areaR = baseRec*alturaRec;  
  return areaR;  
}
```

Area del Rectangulo Exp: 72

Nombre de la función:

`const areaExpt = function(pbaseTri, palturaTri)`

Versión: 2.0

Descripción:

Función que calcula las áreas de figuras geométricas como expresión

baseTri      Tipo de variable: Int

alturaTri    Tipo de variable: Int

areaT        Tipo de variable: Int

Código:

```
const areaExpt = function(pbaseTri, palturaTri){  
  let areaT;  
  let baseTri = pbaseTri;  
  let alturaTri = palturaTri;  
  areaT = (baseTri*alturaTri)/2;  
  return areaT;  
}
```

Area del Triangulo Exp: 36

7.

Nombre de la función:

`sueldo(pdiasTrab, pvalorDia)`

Versión: 1.0

Descripción:

Función que calcula el sueldo de una persona con Parámetros

diasTrab    Tipo de variable: Int

valorDia    Tipo de variable: Int

salario     Tipo de variable: Int



# Tecnólogo en Análisis y Desarrollo de Software

## Ficha 2900177

### Funciones JS

Código:

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

Nombre de la función:

salud(psalarario)

Versión: 1.0

Descripción:

Función que calcula el sueldo de una persona con Parámetros

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;  
}
```

Nombre de la función:

pension(psalarario)

Versión: 1.0

Descripción:

Función que calcula el sueldo de una persona con Parámetros

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;  
}
```



# Tecnólogo en Análisis y Desarrollo de Software

## Ficha 2900177

### Funciones JS

<b>Nombre de la función:</b> arl(psalarario)		<b>Versión: 1.0</b>
<b>Descripción:</b> Función que calcula el sueldo de una persona con Parámetros		
salario	Tipo de variable: Int	
arlP	Tipo de variable: float	
Código:		
<pre>function arl(psalarario){   let salario = psalarario;   let arlP;   arlP = salario * 0.052   return arlP; }</pre>		

<b>Nombre de la función:</b> <b>totalSueldo(psalario)</b>		<b>Versión: 1.0</b>
<b>Descripción:</b> Función que calcula el sueldo de una persona con Parámetros		
salario	Tipo de variable: Int	
saludF	Tipo de variable: Int	
pensionF	Tipo de variable: Int	
arlF	Tipo de variable: float	
descuento	Tipo de variable: float	
totalSalario	Tipo de variable: float	
<b>Código:</b>		
<pre>function totalSueldo(psalario){   let salario = psalario;   let saludF = salud(salario);   let pensionF = pension(salario);   let arlF = arl(salario);    let descuento = saludF + pensionF + arlF;    let totalSalario = salario - descuento;    return "Se aplica un descuento de: "+descuento + "\n"+ "El Total a pagar es: "+totalSalario }</pre>		
<div>Sueldo bruto: 1299900</div> <div>La salud es de: 155988</div> <div>La pension es de: 207984</div> <div>La arl es de: 67594.8</div> <div>Se aplica un descuento de: 431566.8</div> <div>El Total a pagar es: 868333.2</div>		



# Tecnólogo en Análisis y Desarrollo de Software

## Ficha 2900177

### Funciones JS

7.1

<b>Nombre de la función:</b> <code>const sueldoExp = function(pdiasTrab, pvalorDia)</code>		<b>Versión: 2.0</b>
<b>Descripción:</b> Función que calcula el sueldo de una persona como Expresión		
diasTrab	Tipo de variable: Int	
valorDia	Tipo de variable: Int	
salario	Tipo de variable: Int	
<b>Código:</b> <pre>//como Expresion const sueldoExp = function(pdiasTrab, pvalorDia){   let diasTrab = pdiasTrab;   let valorDia = pvalorDia;   let salario;   salario = diasTrab*valorDia;   return salario; }</pre>		

<b>Nombre de la función:</b> const saludExp = function(psalario)		<b>Versión: 2.0</b>
<b>Descripción:</b> Función que calcula el sueldo de una persona como Expresión		
salario	Tipo de variable: Int	
sal	Tipo de variable: Int	
Código:	<pre>const saludExp = function(psalario){     let salario = psalario;     let sal;     sal = salario * 0.12     return sal; }</pre>	

<b>Nombre de la función:</b> <b>const pensionExp = function(psalarario)</b>		<b>Versión: 2.0</b>
<b>Descripción:</b> Función que calcula el sueldo de una persona como Expresión		
salario	Tipo de variable: Int	
pensi	Tipo de variable: Int	





# Tecnólogo en Análisis y Desarrollo de Software

## Ficha 2900177

### Funciones JS

Código:

```
const pensionExp = function(psalario){  
  let salario = psalario;  
  let pensi;  
  pensi = salario * 0.16  
  return pensi;  
}
```

Nombre de la función:

**const arlExp = function(psalario)**

Versión: 2.0

Descripción:

Función que calcula el sueldo de una persona como Expresión

salario	Tipo de variable: Int
---------	-----------------------

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

Código:

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

Nombre de la función:

**const totalSueldoExp = function(psalario)**

Versión: 2.0

Descripción:

Función que calcula el sueldo de una persona como Expresión

salario	Tipo de variable: Int
---------	-----------------------

saludF	Tipo de variable: Int
--------	-----------------------

pensionF	Tipo de variable: Int
----------	-----------------------

arlF	Tipo de variable: float
------	-------------------------

descuento	Tipo de variable: float
-----------	-------------------------

totalSalario	Tipo de variable: float
--------------	-------------------------



# Tecnólogo en Análisis y Desarrollo de Software

## Ficha 2900177

### Funciones JS

Código:

```
const totalSueldoExp = function(psalario){
  let salario = psalario;
  let saludF = salud(salario);
  let pensionF = pension(salario);
  let arlF = arl(salario);

  let descuento = saludF + pensionF + arlF;

  let totalSalario = salario - descuento;

  return "Se aplica un descuento de: "+descuento + "\n"+ "El Total a pagar es: "+totalSalario
}
```

```
Sueldo bruto: 1299900
La salud es de: 155988
La pension es de: 207984
La arl es de: 67594.8
Se aplica un descuento de: 431566.8
El Total a pagar es: 868333.2
```

8.

Nombre de la función: nummayorP(pnumUno, pnumDos)	Versión: 1.0
Descripción: Función que calcula el numero mayor de dos numeros y si son iguales con Parámetros	
numUno	Tipo de variable: Int
numDos	Tipo de variable: Int
Código:	
<pre>//con parametros function nummayorP(pnumUno, pnumDos){   let numUno = pnumUno;   let numDos = pnumDos;    if(numUno!=numDos){     if(numUno&gt;numDos){       return "Numero 1 es mayor"     }else{       return "Numero 2 es mayor"     }   }else{     return "Los dos numeros son iguales"   } }</pre>	<pre>Numero 1: 6 Numero 2: 6 Los dos numeros son iguales</pre>



# Tecnólogo en Análisis y Desarrollo de Software

## Ficha 2900177

### Funciones JS

8.1

<b>Nombre de la función:</b> const nummayor= function(pnumUno, pnumDos)		<b>Versión: 2.0</b>
<b>Descripción:</b> Función que calcula el número mayor de dos numeros y si son iguales como Expresión		
numUno	Tipo de variable: Int	
numDos	Tipo de variable: Int	
Código:		

```
//como Expresion
const nummayor = function(pnumUno, pnumDos){
  let numUno = pnumUno;
  let numDos = pnumDos;

  if(numUno!=numDos){
    if(numUno>numDos){
      return "Numero 1 es mayor"
    }else{
      return "Numero 2 es mayor"
    }
  }else{
    return "Los dos numeros son iguales"
  }
}
```

Numero 1 Exp: 4

---

Numero 2 Exp: 5

---

Numero 2 es mayor

```
Numero 1 Exp: 4
Numero 2 Exp: 5
Numero 2 es mayor
```

9.

<b>Nombre de la función:</b> <code>edad(pfechaNac, pfechaActual)</code>		<b>Versión:</b> 1.0
<b>Descripción:</b> Función que calcula si una persona es mayor de edad con Parámetros		
fechaNac	Tipo de variable: date	
fechaActual	Tipo de variable: date	
edadd	Tipo de variable: date	

Código:

```
function edad(pfechaNac, pfechaActual){  
    let fechaNac = pfechaNac;  
    let fechaActual = pfechaActual;  
    let edadd  
    edadd= fechaActual-fechaNac;  
    return edadd;  
}
```



# Tecnólogo en Análisis y Desarrollo de Software

## Ficha 2900177

### Funciones JS

Nombre de la función: edadMayor(pedadd)		Versión: 1.0
Descripción: Función que calcula si una persona es mayor de edad		
edadd	Tipo de variable: date	
Código:		

```
function edadMayor(pedadd){  
  let edadd = pedadd  
  
  if(edadd>=18){  
    return "Es mayor de edad"  
  }else{  
    return "Es menor de edad"  
  }  
}
```

La edad de la persona es: 18  
Es mayor de edad

```
La edad de la persona es: 18  
Es mayor de edad
```

9.1

<b>Nombre de la función:</b> const edadExp= function(pfechaNac, pfechaActual)		<b>Versión: 2.0</b>
<b>Descripción:</b> Función que calcula si una persona es mayor de edad como Expresión		
fechaNac	Tipo de variable: date	
fechaActual	Tipo de variable: date	
edadd	Tipo de variable: date	
<b>Código:</b> <pre>const edadExp = function(pfechaNac, pfechaActual){   let fechaNac = pfechaNac;   let fechaActual = pfechaActual;   let edadd   edadd= fechaActual-fechaNac;   return edadd; }</pre>		



# Tecnólogo en Análisis y Desarrollo de Software

## Ficha 2900177

### Funciones JS

Nombre de la función: const mayorExp= function(pedadd)		Versión: 2.0
Descripción: Función que calcula si una persona es mayor de edad como Expresión		
edadd	Tipo de variable: date	
Código:		
<pre>const mayorExp = function(pedadd){     let edadd = pedadd;      if(edadd&gt;=18){         return "Es mayor de edad"     }else{         return "Es menor de edad"     } }</pre>		
<div>La edad de la persona Exp es: 16</div> <div>Es menor de edad</div>		

```
La edad de la persona Exp es: 16  
Es menor de edad
```

10.

<b>Nombre de la función:</b> num(pnumero)		<b>Versión:</b> 1.0
<b>Descripción:</b> Función que calcula el número mayor de tres numeros y si son iguales con Parámetros		
numero	Tipo de variable: int	
<b>Código:</b> <pre>function num(pnumero){     let numero = pnumero;     return numero }</pre>		

<b>Nombre de la función:</b> <code>numMayor(pnum1, pnum2, pnum3)</code>		<b>Versión:</b> 1.0
<b>Descripción:</b> Función que calcula el número mayor de tres numeros y si son iguales con Parámetros		
num1	Tipo de variable: int	
num2	Tipo de variable: int	
num3	Tipo de variable: int	



# Tecnólogo en Análisis y Desarrollo de Software

## Ficha 2900177

### Funciones JS

Código:

```
function numMayor(pnum1, pnum2, pnum3){
  let num1 = pnum1
  let num2 = pnum2
  let num3 = pnum3;

  if(num1==num2 && num1==num3 && num3==num2){
    return "los Tres numeros son iguales "
  }else{
    if(num1>num2 && num1>num3){
      return "El numero 1 es mayor"
    }else{
      if(num2>num1 && num2>num3){
        return "El numero 2 es mayor"
      }else{
        return "El numero 3 es mayor"
      }
    }
  }
}
```

```
Numero 1: 9
Numero 2: 4
Numero 3: 5
El numero 1 es mayor
```

10.1

Nombre de la función:

const numExp = function(pnumero)

Versión: 2.0

Descripción:

Función que calcula el número mayor de tres numeros y si son iguales como Expresión

numero

Tipo de variable: int

Código:

```
// como Expresion

const numExp = function(pnumero){
  let numero = pnumero;
  return numero
}
```

Nombre de la función:

const numMayorExp(pnum1, pnum2, pnum3)

Versión: 1.0

Descripción:

Función que calcula el número mayor de tres numeros y si son iguales como Expresión

num1

Tipo de variable: int

num2

Tipo de variable: int

num3

Tipo de variable: int



# Tecnólogo en Análisis y Desarrollo de Software

## Ficha 2900177

### Funciones JS

Código:

```
const numMayorExp = function(pnum1, pnum2, pnum3){
  let num1 = pnum1
  let num2 = pnum2
  let num3 = pnum3;

  if(num1==num2 && num1==num3 && num3==num2){
    return "los Tres numeros son iguales "
  }else{
    if(num1>num2 && num1>num3){
      return "El numero 1 es mayor"
    }else{
      if(num2>num1 && num2>num3){
        return "El numero 2 es mayor"
      }else{
        return "El numero 3 es mayor"
      }
    }
  }
}
```

```
Numero 4: 6
Numero 5: 6
Numero 6: 6
los Tres numeros son iguales
```

11.

Nombre de la función: areas(plado)		Versión: 1.0
Descripción: Función que calcula el área de tres cuadrados y si son iguales o cual es mayor con Parámetros		
lado	Tipo de variable: int	
area	Tipo de variable: Int	
Código:	<pre>//con parametros  function areas(plado){   let lado = plado;   let area;   area = lado*lado;   return area; }</pre>	

Nombre de la función: areaMayor(parea1, pareas2, pareas3)	Versión: 1.0
Descripción: Función que calcula el área de tres cuadrados y si son iguales o cual es mayor con Parámetros	



# Tecnólogo en Análisis y Desarrollo de Software

## Ficha 2900177

### Funciones JS

area1	Tipo de variable: int
area2	Tipo de variable: Int
area3	Tipo de variable: Int

Código:

```
function areaMayor(parea1, pareaa2, pareaa3){
  let area1 = pareaa1;
  let area2 = pareaa2;
  let area3 = pareaa3;

  if(area1==area2 && area1==area3 && area3==area2){
    return "las tres areas son iguales "
  }else{
    if(area1>area2 && area1>area3){
      return "El area 1 es mayor"
    }else{
      if(area2>area1 && area2>area3){
        return "El area 2 es mayor"
      }else{
        return "El area 3 es mayor"
      }
    }
  }
}
```

```
El area del 1 cuadrado como parametro es: 16
El area del 2 cuadrado como parametro es: 64
El area del 3 cuadrado como parametro es: 81
El area 3 es mayor
```

## 11.1

<b>Nombre de la función:</b> <code>const areasExp = function(plado)</code>	<b>Versión:</b> 2.0
<b>Descripción:</b> Función que calcula el área de tres cuadrados y si son iguales o cual es mayor como Expresión	
lado	Tipo de variable: int
area	Tipo de variable: Int
Código:	
<pre>const areasExp = function(plado){   let lado = plado;   let area;   area = lado*lado;   return area; }</pre>	

<b>Nombre de la función:</b> <code>areasMayorExp = function(parea1, pareaa2, pareaa3)</code>	<b>Versión:</b> 2.0
<b>Descripción:</b> Función que calcula el área de tres cuadrados y si son iguales o cual es mayor como Expresion	





# Tecnólogo en Análisis y Desarrollo de Software

Ficha 29001790177

## Funciones JS

area1	Tipo de variable: int
area2	Tipo de variable: Int
area3	Tipo de variable: Int

Código:

```
const areasMayorExp = function(parea1, pareaa2, pareaa3){
  let area1 = pareaa1;
  let area2 = pareaa2;
  let area3 = pareaa3;

  if(area1==area2 && area1==area3 && area3==area2){
    return "las tres areas son iguales "
  }else{
    if(area1>area2 && area1>area3){
      return "El area 1 es mayor"
    }else{
      if(area2>area1 && area2>area3){
        return "El area 2 es mayor"
      }else{
        return "El area 3 es mayor"
      }
    }
  }
}
```

El area del 1 cuadrado como Expresion es: 9  
El area del 2 cuadrado como Expresion es: 9  
El area del 3 cuadrado como Expresion es: 9  
las tres areas son iguales

12.

Nombre de la función:

edad(pfechaNac, pfechaActual)

Versión: 1.0

Descripción:

Función que calcula la edad de 3 personas y si el promedio de edades es mayor de edad Parametros

fechaNac Tipo de variable: date

fechaActual Tipo de variable: date

edadd Tipo de variable: date

Código:

```
//con parametros
let edadd
function edad(pfechaNac, pfechaActual){
  let fechaNac = pfechaNac;
  let fechaActual = pfechaActual;
  edadd= fechaActual-fechaNac;
  return edadd;
}
```

Nombre de la función:

mayorEdad(pedadd)

Versión: 1.0



# Tecnólogo en Análisis y Desarrollo de Software

## Ficha 2900177

### Funciones JS

#### Descripción:

Función que calcula la edad de 3 personas y si el promedio de edades es mayor de edad Parametros

edad	Tipo de variable: date
------	------------------------

Código:

```
function mayorEdad(edad){
  if(edad>=18){
    return "Es mayor de edad y su edad es: "+edad
  }else{
    return "Es menor de edad y su edad es: "+edad
  }
}
```

#### Nombre de la función:

**promedioEdades(pedad)**

**Versión: 1.0**

#### Descripción:

Función que calcula la edad de 3 personas y si el promedio de edades es mayor de edad Parametros

edad	Tipo de variable: float
------	-------------------------

Código:

```
function promedioEdades(pedad){
  edad = pedad
  if(edad>=18){
    return "El promedio de edades esta en la mayoría de edad ";
  }else{
    return " El promedio de edades no esta en la mayoría de edad ";
  }
}
```

```
La persona 1 Es mayor de edad y su edad es: 18
La persona 2 Es mayor de edad y su edad es: 20
La persona 3 Es mayor de edad y su edad es: 38
El promedio de edades es: 25.333333333333332
El promedio de edades esta en la mayoría de edad
```

## 12.1

#### Nombre de la función:

**edadExp(pfechaNac, pfechaActual)**

**Versión: 2.0**

#### Descripción:

Función que calcula la edad de 3 personas y si el promedio de edades es mayor de edad Expresion

fechaNac	Tipo de variable: date
----------	------------------------

fechaActual	Tipo de variable: date
-------------	------------------------

edad	Tipo de variable: date
------	------------------------



# Tecnólogo en Análisis y Desarrollo de Software

## Ficha 2900177

### Funciones JS

Código:

```
const edadExp = function(pfechaNac, pfechaActual){  
  let fechaNac = pfechaNac;  
  let fechaActual = pfechaActual;  
  edadd= fechaActual-fechaNac;  
  return edadd;  
}
```

**Nombre de la función:**

**mayorEdadExp(pedadd)**

**Versión: 2.0**

**Descripción:**

Función que calcula la edad de 3 personas y si el promedio de edades es mayor de edad Expresion

edadd

Tipo de variable: date

Código:

```
const mayorEdadExp = function(edadd){  
  if(edadd>=18){  
    return "Es mayor de edad y su edad es: "+edadd  
  }else{  
    return "Es menor de edad y su edad es: "+edadd  
  }  
}
```

**Nombre de la función:**

**promedioEdadExp(pedadd)**

**Versión: 2.0**

**Descripción:**

Función que calcula la edad de 3 personas y si el promedio de edades es mayor de edad Expresion

edadd

Tipo de variable: string



# Tecnólogo en Análisis y Desarrollo de Software

## Ficha 2900177

### Funciones JS

Código:

```
const promedioEdadExp = function(pedadd){  
  edadd = pedadd  
  if(edadd>=18){  
    return "El promedio de edades esta en la mayoria de edad ";  
  }else{  
    return " El promedio de edades no esta en la mayoria de edad ";  
  }  
}
```

```
La persona 4 Es mayor de edad y su edad es: 18  
La persona 5 Es menor de edad y su edad es: 15  
La persona 6 Es mayor de edad y su edad es: 23  
El promedio de edades es: 18.666666666666668  
El promedio de edades esta en la mayoria de edad
```

13.

<b>Nombre de la función:</b> sueldo(pdiasTrab, pvalorDia)		<b>Versión: 1.0</b>
<b>Descripción:</b> Función que calcula el sueldo de una persona con Parámetros		
diasTrab	Tipo de variable: Int	
valorDia	Tipo de variable: Int	
salario	Tipo de variable: Int	
<b>Código:</b>		
<pre>function sueldo(pdiasTrab, pvalorDia){   let diasTrab = pdiasTrab;   let valorDia = pvalorDia;   let salario;   salario= diasTrab*valorDia;   return salario; }</pre>		

<b>Nombre de la función:</b> salud(psalario)		<b>Versión: 1.0</b>
<b>Descripción:</b> Función que calcula el sueldo de una persona con Parámetros		
salario	Tipo de variable: Int	
sal	Tipo de variable: int	



# Tecnólogo en Análisis y Desarrollo de Software

## Ficha 2900177

### Funciones JS

Código:

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

<b>Nombre de la función:</b> pension(psalario)		<b>Versión: 1.0</b>
<b>Descripción:</b> Función que calcula el sueldo de una persona con Parámetros		
salario	Tipo de variable: Int	
pensi	Tipo de variable: int	
Código:		
<pre>function pension(psalario){     let salario = psalario;     let pensi;     pensi = salario * 0.16     return pensi; }</pre>		

<b>Nombre de la función:</b> arl(psalario)		<b>Versión: 1.0</b>
<b>Descripción:</b> Función que calcula el sueldo de una persona con Parámetros		
salario	Tipo de variable: Int	
arlP	Tipo de variable: float	
Código: <pre>function arl(psalario){     let salario = psalario;     let arlP;     arlP = salario * 0.052     return arlP; }</pre>		



# Tecnólogo en Análisis y Desarrollo de Software

## Ficha 2900177

### Funciones JS

<b>Nombre de la función:</b> <b>subTraspor(psalario)</b>		<b>Versión: 1.0</b>
<b>Descripción:</b> Función que calcula el sueldo de una persona con Parámetros		
salario	Tipo de variable: Int	
salarioM	Tipo de variable: Int	
subTrab	Tipo de variable: Int	
<b>Código:</b> <pre>function subTraspor(salario){     let salarioM= 1300000     let subTrab;      if(salario &lt; 2*salarioM){         subTrab = 114000;     }else{         subTrab = 0;     }      return subTrab; }</pre>		

<b>Nombre de la función:</b> retencion(psalario)		<b>Versión: 1.0</b>
<b>Descripción:</b> Función que calcula el sueldo de una persona con Parámetros		
salario	Tipo de variable: Int	
salarioM	Tipo de variable: Int	
retención	Tipo de variable: float	
<b>Código:</b> <pre>function retencionn(salario){     let retencion;     let salarioM;     if(salario&gt;4*salarioM){         retencion = salario * 0.04     }else{         retencion = 0     }     return retencion; }</pre>		

<b>Nombre de la función:</b> <b>totalSueldo(psalario)</b>		<b>Versión: 1.0</b>
<b>Descripción:</b> Función que calcula el sueldo de una persona con Parámetros		
salario	Tipo de variable: Int	



# Tecnólogo en Análisis y Desarrollo de Software

## Ficha 2900177

### Funciones JS

saludF	Tipo de variable: Int
pensionF	Tipo de variable: Int
arlF	Tipo de variable: float
retención	Tipo de variable: Int
subTrasporte	Tipo de variable: Int
descuento	Tipo de variable: float
totalSalario	Tipo de variable: float

Código:

```
function totalSueldo(psalario){
  let salario = psalario;
  let saludF = salud(salario);
  let pensionF = pension(salario);
  let arlF = arl(salario);

  let retencion = retencionn(salario);
  let subTrasporte = subTraspor(salario);
  let descuento = saludF + pensionF + arlF;

  let totalSalario = (salario + subTrasporte) - (retencion + descuento);

  return "Se aplica un descuento de: "+descuento + "\n" + "El Total a pagar es: "+totalSalario;
}
```

```
Sueldo bruto: 1299900
La salud es de: 155988
La pension es de: 207984
La arl es de: 67594.8
La retención es de: 0
El subsidio de transporte: 114000
Se aplica un descuento de: 431566.8
El Total a pagar es: 982333.2
```

## 13.1

<b>Nombre de la función:</b> const sueldoExp = function(pdiasTrab, pvalorDia)		<b>Versión: 2.0</b>
<b>Descripción:</b> Función que calcula el sueldo de una persona como Expresión		
diasTrab	Tipo de variable: Int	
valorDia	Tipo de variable: Int	
salario	Tipo de variable: Int	



# Tecnólogo en Análisis y Desarrollo de Software

## Ficha 2900177

### Funciones JS

Código:

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

Nombre de la función:

const saludExp = function(psalarario)

Versión: 2.0

Descripción:

Función que calcula el sueldo de una persona como Expresión

salario      Tipo de variable: Int

sal            Tipo de variable: int

Código:

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

Nombre de la función:

const pensionExp = function(psalarario)

Versión: 2.0

Descripción:

Función que calcula el sueldo de una persona como Expresión

salario      Tipo de variable: Int

pensi        Tipo de variable: int

Código:

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





# Tecnólogo en Análisis y Desarrollo de Software

## Ficha 2900177

### Funciones JS

Nombre de la función: const arlExp = function(psalario)		Versión: 2.0
Descripción: Función que calcula el sueldo de una persona como Expresión		
salario	Tipo de variable: Int	
arlP	Tipo de variable: float	
Código:		
<pre>const arlExp = function(psalario){   let salario = psalario;   let arlP;   arlP = salario * 0.052   return arlP; }</pre>		

<b>Nombre de la función:</b> const subExp = function(psalario)		<b>Versión: 2.0</b>
<b>Descripción:</b> Función que calcula el sueldo de una persona como Expresion		
salario	Tipo de variable: Int	
salarioM	Tipo de variable: Int	
subTrab	Tipo de variable: Int	
<b>Código:</b> <pre>const subExp = function(salario){   let salarioM;   let subTrab;   if(salario &lt; 2*salarioM){     subTrab = 114000;   }else{     subTrab = 0;   }    return subTrab; }</pre>		

<b>Nombre de la función:</b> <b>const retenExp = function(psalario)</b>		<b>Versión: 2.0</b>
<b>Descripción:</b> Función que calcula el sueldo de una persona como Expresion		
salario	Tipo de variable: Int	
salarioM	Tipo de variable: Int	
retencion	Tipo de variable: Int	



# Tecnólogo en Análisis y Desarrollo de Software

Ficha 29001900177

## Funciones JS

Código:

```
const retenExp= function(salario){  
    let retencion;  
    let salarioM;  
    if(salario>4*salarioM){  
        retencion = salario * 0.04  
    }else{  
        retencion = 0  
    }  
    return retencion;  
}
```

**Nombre de la función:**  
**totalSueldo(psalario)**

**Versión: 2.0**

**Descripción:**

Función que calcula el sueldo de una persona como Expresion

salario	Tipo de variable: Int
saludF	Tipo de variable: Int
pensionF	Tipo de variable: Int
arlF	Tipo de variable: int
retención	Tipo de variable: Int
subTrasporte	Tipo de variable: Int
descuento	Tipo de variable: int
totalSalario	Tipo de variable: int

Código:

```
function totalSueldoExp(psalario){  
    let salario = psalario;  
    let saludF = salud(salario);  
    let pensionF = pension(salario);  
    let arlF = arl(salario);  
  
    let retencion = retencionn(salario);  
    let subTrasporte = subTraspor(salario);  
    let descuento = saludF + pensionF + arlF;  
  
    let totalSalario = (salario + subTrasporte) - (retencion + descuento);  
  
    return totalSalario;  
}
```

```
Sueldo bruto: 1590000  
La salud es de: 190800  
La pension es de: 254400  
La arl es de: 82680  
La retención es de: 0  
El subsidio de transporte: 0  
Se aplica un descuento de: 527880  
El Total a pagar es: 1176120
```



# Tecnólogo en Análisis y Desarrollo de Software

## Ficha 2900177

### Funciones JS

14.

<b>Nombre de la función:</b> porcNotas(pnota, pporc)	<b>Versión: 1.0</b>
<b>Descripción:</b> Función que calcula 3 notas y realiza la suma de los porcentajes para definir la nota Parámetros	
nota	Tipo de variable: float
porc	Tipo de variable: float
result	Tipo de variable: float
<b>Código:</b> <pre>function porcNotas(pnota, pporc){   let nota = pnota;   let porc = pporc;   let result;   result = nota * porc;   return result; }</pre>	

<b>Nombre de la función:</b> suma(pnota1, pnota2, pnota3)	<b>Versión: 1.0</b>
<b>Descripción:</b> Función que calcula 3 notas y realiza la suma de los porcentajes para definir la nota Parametros	
nota1	Tipo de variable: float
nota2	Tipo de variable: float
nota3	Tipo de variable: float
sumar	Tipo de variable: string
<b>Código:</b> <pre>function suma(pnota1, pnota2, pnota3){   let nota1 = pnota1   let nota2 = pnota2   let nota3 = pnota3   let sumar;   sumar= nota1+nota2+nota3;   if(sumar&gt;4.5){     return "Es una nota superior";   }else{     if(sumar&lt;4.5 &amp;&amp; sumar&gt;3.5){       return "Es una nota buena";     }else{       if(sumar&lt;3.5 &amp;&amp; sumar&gt;3.0){         return "Es una nota media";       }else{         return "Es una nota mala";       }     }   } }</pre> <div>Porcentaje de la nota 1: 0.8800000000000001 Porcentaje de la nota 2: 1.015 Porcentaje de la nota 3: 1.305 la suma de los porcentajes Es una nota media</div>	



# Tecnólogo en Análisis y Desarrollo de Software

## Ficha 2900177

### Funciones JS

#### 14.1

Nombre de la función:

`const porcExp = function(pnota, pporc)`

Versión: 2.0

Descripción:

Función que calcula 3 notas y realiza la suma de los porcentajes para definir la nota Parámetros

nota Tipo de variable: float

porc Tipo de variable: float

result Tipo de variable: float

Código:

```
const porcExp = function(pnota, pporc){
  let nota = pnota;
  let porc = pporc;
  let result;
  result = nota * porc;
  return result;
}
```

Nombre de la función:

`const sumaExp= function(pnota1, pnota2, pnota3)`

Versión: 2.0

Descripción:

Función que calcula 3 notas y realiza la suma de los porcentajes para definir la nota

nota1 Tipo de variable: float

nota2 Tipo de variable: float

nota3 Tipo de variable: float

sumar Tipo de variable: string

Código:

```
const sumaExp = function(pnota1, pnota2, pnota3){
  let nota1 = pnota1
  let nota2 = pnota2
  let nota3 = pnota3
  let sumar;
  sumar= nota1+nota2+nota3;
  if(sumar>4.5){
    return "Es una nota superior";
  }else{
    if(sumar<4.5 && sumar>3.5){
      return "Es una nota buena";
    }else{
      if(sumar<3.5 && sumar>3.0){
        return "Es una nota media";
      }else{
        return "Es una nota mala";
      }
    }
  }
}
```

Porcentaje de la nota 4: 0.86

Porcentaje de la nota 5: 0.6649999999999999

Porcentaje de la nota 6: 0.63

la suma de los porcentajes Es una nota mala



# Tecnólogo en Análisis y Desarrollo de Software

## Ficha 2900177

### Funciones JS

15.

<b>Nombre de la función:</b> <b>contar(pnumero)</b>		<b>Versión: 1.0</b>
<b>Descripción:</b> Función que cuenta del 1 al 5 con while		
numero	Tipo de variable: Int	
contador	Tipo de variable: Int	
resultado	Tipo de variable: Int	
<b>Código:</b>		
<pre>// con parametros let numero function contar(pnumero){     numero = pnumero;     let contador = 0;     let resultado = "";      while(contador&lt;numero){         contador = contador+1;         resultado += contador + "\n";     }     return resultado; }</pre>		

<b>Nombre de la función:</b> <b>const contarExp = function(pnumero)</b>		<b>Versión: 2.0</b>
<b>Descripción:</b> Función que cuenta del 1 al 5 con while		
numero	Tipo de variable: Int	
contador	Tipo de variable: Int	
resultado	Tipo de variable: Int	



# Tecnólogo en Análisis y Desarrollo de Software

## Ficha 2900177

### Funciones JS

Código:

```
//Como expresion

const contarExp = function(pnumero){
  numero = pnumero;
  let contador = 0;
  let resultado = "";

  while(contador<numero){
    contador = contador+1;
    resultado += contador + "\n";
  }
  return resultado;
}
```

Con parametros:

1  
2  
3  
4  
5

Como Expresion:

1  
2  
3  
4  
5

15.1

Nombre de la función:  
**contar(pnumero)**

Versión: 1.0

**Descripción:**

Función que cuenta del 1 al 5 con ciclo for

contador	Tipo de variable: Int
----------	-----------------------

resultado	Tipo de variable: Int
-----------	-----------------------

numero	Tipo de variable: Int
--------	-----------------------

Código:

```
// con parametros

let numero
function contar(pnumero){
  numero = pnumero;
  let contador;
  let resultado = "";

  for(contador =1; numero>=contador; contador++){
    resultado += contador + "\n";
  }
  return resultado;
}
```

Nombre de la función:  
**const contarExp = function(pnumero)**

Versión: 2.0

**Descripción:**

Función que cuenta del 1 al 5 con ciclo for

contador	Tipo de variable: Int
----------	-----------------------



# Tecnólogo en Análisis y Desarrollo de Software

## Ficha 2900177

### Funciones JS

resultado	Tipo de variable: Int
numero	Tipo de variable: Int
Código:	<div><pre>//Como expresion  const contarExp = function(pnumero){   numero = pnumero;   let contador;   let resultado = "";    for(contador =1; numero&gt;=contador; contador++){     resultado += contador + "\n";   }   return resultado; }</pre></div> <div><p>Con parametros:</p><pre>1 2 3 4 5</pre><p>Como Expresion:</p><pre>1 2 3 4 5</pre></div>

16.

Nombre de la función: <b>factor(pnumero)</b>	Versión: 1.0
Descripción: Función que calcula el factorial de 5 con while	
numero	Tipo de variable: Int
contador	Tipo de variable: Int
factorial	Tipo de variable: Int
Código:	<pre>// con parametros  let numero function factor(pnumero){   numero = pnumero;   let contador;   let factorial;    contador = 0;   factorial = 1;    while(contador&lt;numero){     contador = contador+1;     factorial = factorial* contador + "\n";   }   return factorial; }</pre>



# Tecnólogo en Análisis y Desarrollo de Software

## Ficha 2900177

### Funciones JS

<b>Nombre de la función:</b> <b>const factorialExp = function(pnumero)</b>		<b>Versión: 2.0</b>
<b>Descripción:</b> Función que calcula el factorial de 5 con while		
numero	Tipo de variable: Int	
contador	Tipo de variable: Int	
factorial	Tipo de variable: Int	
<b>Código:</b> <div><pre>//Como expresion const factorialExp = function(pnumero){   numero = pnumero;   let contador;   let factorial;    contador = 0;   factorial = 1;    while(contador&lt;numero){     contador = contador+1;     factorial = factorial* contador + "\n";   }   return factorial; }</pre></div> <div><b>Con parametros:</b> 120 <b>Como Expresion:</b> 120</div>		

16.1

<b>Nombre de la función:</b> <b>factor(pnumero)</b>		<b>Versión: 1.0</b>
<b>Descripción:</b> Función que calcula el factorial de 5 con for		
numero	Tipo de variable: Int	
contador	Tipo de variable: Int	
factorial	Tipo de variable: Int	





# Tecnólogo en Análisis y Desarrollo de Software

Ficha 2900177

## Funciones JS

Código:

```
// con parametros

let numero
function factor(pnumero){
  numero = pnumero;
  let contador;
  let factorial =1;

  for(contador =1; numero>=contador; contador++){
    factorial = factorial* contador + "\n";
  }
  return factorial;
}
```

Nombre de la función:

**const factorialExp = function(pnumero)**

Versión: 2.0

Descripción:

Función que calcula el factorial de 5 con for

numero	Tipo de variable: Int
--------	-----------------------

contador	Tipo de variable: Int
----------	-----------------------

factorial	Tipo de variable: Int
-----------	-----------------------

Código:

```
//Como expresion

const factorialExp = function(pnumero){
  numero = pnumero;
  let contador;
  let factorial =1;

  for(contador =1; numero>=contador; contador++){
    factorial = factorial* contador + "\n";
  }
  return factorial;
}
```

Con parametros:

120

Como Expresion:

120



# Tecnólogo en Análisis y Desarrollo de Software

## Ficha 2900177

### Funciones JS

17.

<b>Nombre de la función:</b> <b>Tabla5(pnumero)</b>		<b>Versión: 1.0</b>
<b>Descripción:</b> Función tabla del 5 hasta el 5 con ciclo while		
numero	Tipo de variable: Int	
contador	Tipo de variable: Int	
multiplicador	Tipo de variable: Int	
resultado	Tipo de variable: Int	
<b>Código:</b>		
<pre>// con parametros  let numero function tabla5(pnumero){   numero = pnumero;   let contador = 0;   let multiplicador = 1;   let resultado = "";    while(contador&lt;numero){     contador = contador+1;     multiplicador = numero*contador;     resultado += (numero)+ "x" +(contador)+ "=" + (multiplicador +"\n");   }   return resultado; }</pre>		

<b>Nombre de la función:</b> <b>const tablaExp5 = function(pnumero)</b>		<b>Versión: 2.0</b>
<b>Descripción:</b> Función tabla del 5 hasta 5 con while		
numero	Tipo de variable: Int	
contador	Tipo de variable: Int	
multiplicador	Tipo de variable: Int	
resultado	Tipo de variable: Int	



# Tecnólogo en Análisis y Desarrollo de Software

Ficha 2900177-00177

## Funciones JS

Código:

```
//Como expresion
const tablaExp5= function(pnumero){
  numero = pnumero;
  let contador = 0;
  let multiplicador = 1;
  let resultado = "";

  while(contador<numero){
    contador = contador+1;
    multiplicador = numero*contador;
    resultado += (numero)+ "x" +(contador)+ "="+ (multiplicador +"\n");
  }
  return resultado;
}
```

Con parametros:

```
5x1=5
5x2=10
5x3=15
5x4=20
5x5=25
```

Como Expresion:

```
5x1=5
5x2=10
5x3=15
5x4=20
5x5=25
```

17.1

Nombre de la función:  
Tabla5(pnumero)

Versión: 1.0

Descripción:

Función tabla del 5 hasta el 5 con ciclo For

numero	Tipo de variable: Int
Contador	Tipo de variable: Int
multiplicador	Tipo de variable: Int
resultado	Tipo de variable: Int

Código:

```
// con parametros
let numero
function tabla5(pnumero){
  numero = pnumero;
  let contador;

  let multiplicador = 1;
  let resultado = "";

  for(contador =1; numero>=contador; contador++){
    multiplicador = numero*contador;
    resultado += (numero)+ "x" +(contador)+ "="+ (multiplicador +"\n");
  }
  return resultado;
}
```



# Tecnólogo en Análisis y Desarrollo de Software

## Ficha 2900177

### Funciones JS

Nombre de la función: const tablaExp5 = function(pnumero)		Versión: 2.0
Descripción: Función tabla del 5 hasta 5 con for		
numero	Tipo de variable: Int	
contador	Tipo de variable: Int	
multiplicador	Tipo de variable: Int	
resultado	Tipo de variable: Int	
Código:		
<pre>//Como expresion  const tablaExp5= function(pnumero){     numero = pnumero;     let contador;      let multiplicador = 1;     let resultado = "";      for(contador =1; numero&gt;=contador; contador++){         multiplicador = numero*contador;         resultado += (numero)+ "x" +(contador)+ "="+ (multiplicador +"\n");     }     return resultado; }</pre>		
<div>Con parametros:</div> <div>5x1=5 5x2=10 5x3=15 5x4=20 5x5=25</div> <div>Como Expresion:</div> <div>5x1=5 5x2=10 5x3=15 5x4=20 5x5=25</div>		

18.

<b>Nombre de la función:</b> <code>tabla9(ptabla)</code>		<b>Versión:</b> 1.0
<b>Descripción:</b> Función tabla del 9 hasta 5 con while como Parametros		
tabla	Tipo de variable: Int	
contador	Tipo de variable: Int	
multiplicador	Tipo de variable: Int	
resultado	Tipo de variable: Int	
msg	Tipo de variable: string	



# Tecnólogo en Análisis y Desarrollo de Software

## Ficha 2900177

### Funciones JS

Código:

```
// con parametros

function tabla9(ptabla){
  let tabla = ptabla;
  let contador = 0;
  let multiplicador = 5;

  let resultado;
  let msg = "";

  while(contador<multiplicador){
    contador = contador+1;
    resultado = tabla*contador;
    msg += (tabla)+ "x" +(contador)+ "="+ (resultado) +"\n";
    msg += parImpar(resultado);
  }

  return msg += "\n";
}
```

Nombre de la función:  
**parImpar(resultado)**

Versión: 1.0

Descripción:

Función tabla del 9 hasta 5 con while como parámetros

resultado	Tipo de variable: Int
msg	Tipo de variable: string

Código:

```
function parImpar(resultado){
  let msg = "";
  if(resultado%2==0){
    msg += "Par\n";
  }else{
    msg += "Impar\n";
  }
  return msg;
}
```

Con parametros:

```
9x1=9
Impar
9x2=18
Par
9x3=27
Impar
9x4=36
Par
9x5=45
Impar
```



# Tecnólogo en Análisis y Desarrollo de Software

## Ficha 2900177

### Funciones JS

<b>Nombre de la función:</b> <b>const tablaExp9 = function(ptabla)</b>		<b>Versión: 2.0</b>
<b>Descripción:</b> Función tabla del 9 hasta 5 con while como Expresión		
tabla	Tipo de variable: Int	
contador	Tipo de variable: Int	
multiplicador	Tipo de variable: Int	
resultado	Tipo de variable: Int	
msg	Tipo de variable: string	
<b>Código:</b> <pre>const tablaExp9= function(ptabla){     let tabla = ptabla;     let contador = 0;     let multiplicador = 5;      let resultado;     let msg = "";      while(contador&lt;multiplicador){         contador = contador+1;         resultado = tabla*contador;         msg += (tabla)+ "x" +(contador)+ "="+ (resultado) +"\n";         msg += parImpare(resultado);     }     return msg += "\n"; }</pre>		

<b>Nombre de la función:</b> <b>const parImpare = function(resultado)</b>		<b>Versión: 2.0</b>
<b>Descripción:</b> Función tabla del 9 hasta 5 con while como Expresión		
resultado	Tipo de variable: Int	
msg	Tipo de variable: string	



# Tecnólogo en Análisis y Desarrollo de Software

## Ficha 2900177

### Funciones JS

Código:

```
const parImpare = function(resultado){  
  let msg = "";  
  if(resultado%2==0){  
    msg += "Par\n";  
  }else{  
    msg += "Impar\n";  
  }  
  return msg;  
}
```

Como Expresion:

```
9x1=9  
Impar  
9x2=18  
Par  
9x3=27  
Impar  
9x4=36  
Par  
9x5=45  
Impar
```

18.1

Nombre de la función:  
tabla9(ptabla)

Versión: 1.0

Descripción:

Función tabla del 9 hasta 5 con for como Parametros

tabla	Tipo de variable: Int
contador	Tipo de variable: Int
multiplicador	Tipo de variable: Int
resultado	Tipo de variable: Int
msg	Tipo de variable: string



# Tecnólogo en Análisis y Desarrollo de Software

## Ficha 2900177

### Funciones JS

Código:

```
// con parametros
function tabla9(ptabla){
  let tabla = ptabla;
  let contador;
  let multiplicador = 5;
  let resultado;
  let msg = "";

  for(contador=1; multiplicador>=contador; contador++){
    resultado = tabla*contador;
    msg += (tabla)+ "x" +(contador)+ "="+ (resultado) +"\n";
    msg += parImpar(resultado);
  }
  return msg += "\n";
}
```

Nombre de la función:  
**parImpar(resultado)**

Versión: 1.0

Descripción:

Función tabla del 9 hasta 5 con while como parámetros

resultado	Tipo de variable: Int
-----------	-----------------------

msg	Tipo de variable: string
-----	--------------------------

Código:

```
function parImpar(resultado){
  let msg = "";
  if(resultado%2==0){
    msg += "Par\n";
  }else{
    msg += "Impar\n";
  }
  return msg;
}
```

Con parametros:

```
9x1=9
Impar
9x2=18
Par
9x3=27
Impar
9x4=36
Par
9x5=45
Impar
```





# Tecnólogo en Análisis y Desarrollo de Software

## Ficha 2900177

### Funciones JS

**Nombre de la función:**

**const tablaExp9 = function(ptabla)**

**Versión: 2.0**

**Descripción:**

Función tabla del 9 hasta 5 con for como Expresión

tabla	Tipo de variable: Int
-------	-----------------------

contador	Tipo de variable: Int
----------	-----------------------

multiplicador	Tipo de variable: Int
---------------	-----------------------

resultado	Tipo de variable: Int
-----------	-----------------------

msg	Tipo de variable: string
-----	--------------------------

**Código:**

```
const tablaExp9= function(ptabla){
  let tabla = ptabla;
  let contador;
  let multiplicador = 5;
  let resultado;
  let msg = "";

  for(contador=1; multiplicador>=contador; contador++){
    resultado = tabla*contador;
    msg += (tabla)+ "x" +(contador)+ "="+ (resultado) +"\n";
    msg += parImpare(resultado);
  }
  return msg += "\n";
}
```

**Nombre de la función:**

**const parImpare = function(resultado)**

**Versión: 2.0**

**Descripción:**

Función tabla del 9 hasta 5 con for como Expresión

resultado	Tipo de variable: Int
-----------	-----------------------

msg	Tipo de variable: string
-----	--------------------------



Tecnólogo en Análisis y Desarrollo de  
Software  
Ficha 2900177  
**Funciones JS**

Código:

```
const parImpare = function(resultado){  
  let msg = "";  
  if(resultado%2==0){  
    msg += "Par\n";  
  }else{  
    msg += "Impar\n";  
  }  
  return msg;  
}
```

Como Expresion:

9x1=9  
Impar  
9x2=18  
Par  
9x3=27  
Impar  
9x4=36  
Par  
9x5=45  
Impar

19.	Nombre de la función: Tablas(plimite)		Versión: 1.0
	Descripción: Función tabla del 1 hasta 5 while como Parámetros		
	limite	Tipo de variable: Int	
	tabla	Tipo de variable: Int	
	contador	Tipo de variable: Int	
	par	Tipo de variable: Int	
	impar	Tipo de variable: Int	
	resultado	Tipo de variable: Int	
	msg	Tipo de variable: string	



# Tecnólogo en Análisis y Desarrollo de Software

## Ficha 2900177

### Funciones JS

Código:

```
let limite;
let par = 0
let impar =0

function tablas(plimite){
    limite = plimite;
    let tabla;
    let contador;
    let resultado;
    let msg = "";
    tabla = 0;

    while(tabla<limite){
        contador = 0;
        tabla = tabla+1;

        while(contador<limite){
            contador = contador+1;
            resultado = tabla*contador;
            msg += (tabla)+ "x" +(contador)+ "="+ (resultado) +"\n";
            msg += parImpar(resultado);
        }
        msg += "\n";
    }
    return msg+ "Total de pares es: "+par+"\n"+
        "Total de impares es: "+impar+"\n";
}
```

<b>Nombre de la función:</b> parImpar(resultado)		<b>Versión: 1.0</b>
<b>Descripción:</b> Función tabla del 1 hasta 5 while como Parámetros		
par	Tipo de variable: Int	
impar	Tipo de variable: Int	
resultado	Tipo de variable: Int	



# Tecnólogo en Análisis y Desarrollo de Software

## Ficha 2900177

### Funciones JS

Código:

```
function parImpar(resultado){  
  if(resultado%2==0){  
    par = par+1  
    return "Buzz\n";  
  }else{  
    impar = impar+1  
    return "Bass\n";  
  }  
}
```

Con parametros:

1x1=1  
Bass  
1x2=2  
Buzz  
1x3=3  
Bass  
1x4=4  
Buzz  
1x5=5  
Bass

2x1=2  
Buzz  
2x2=4  
Buzz  
2x3=6  
Buzz  
2x4=8  
Buzz  
2x5=10  
Buzz

3x1=3  
Bass  
3x2=6  
Buzz  
3x3=9  
Bass  
3x4=12  
Buzz  
3x5=15  
Bass

4x1=4  
Buzz  
4x2=8  
Buzz  
4x3=12  
Buzz  
4x4=16  
Buzz  
4x5=20  
Buzz

5x1=5  
Bass  
5x2=10  
Buzz  
5x3=15  
Bass  
5x4=20  
Buzz  
5x5=25  
Bass

Total de pares es: 16  
Total de impares es: 9

<b>Nombre de la función:</b> conts TablasExp = function(plimite)		<b>Versión:</b> 2.0
<b>Descripción:</b> Función tabla del 1 hasta 5 while como Expresion		
limite	Tipo de variable: Int	
tabla	Tipo de variable: Int	
contador	Tipo de variable: Int	
par	Tipo de variable: Int	
impar	Tipo de variable: Int	



# Tecnólogo en Análisis y Desarrollo de Software

## Ficha 2900177

### Funciones JS

resultado	Tipo de variable: Int
msg	Tipo de variable: string
Código:	
<pre>const tablasExp= function(plimite){     limite = plimite;     let tabla;     let contador;     let resultado;     let msg = "";     tabla = 0;      while(tabla&lt;limite){         contador = 0;         tabla = tabla+1;          while(contador&lt;limite){             contador = contador+1;             resultado = tabla*contador;             msg += (tabla)+ "x" +(contador)+ "="+ (resultado) +"\n";             msg += parImpare(resultado);         }         msg += "\n";     }     return msg+ "Total de pares es: "+par+"\n"+         "Total de impares es: "+impar+"\n"; }</pre>	

Nombre de la función: <b>const parImpare = function(resultado)</b>	Versión: 2.0
Descripción: Función tabla del 1 hasta 5 while como Expresion	
par	Tipo de variable: Int
impar	Tipo de variable: Int
resultado	Tipo de variable: Int



# Tecnólogo en Análisis y Desarrollo de Software

## Ficha 2900177

### Funciones JS

Código:

```
const parImpare = function(resultado){  
  let par;  
  let impar;  
  if(resultado%2==0){  
    par = par+1  
    return "Buzz\n";  
  }else{  
    impar = impar+1  
    return "Bass\n";  
  }  
}
```

Como Expresion:

```
1x1=1  
Bass  
1x2=2  
Buzz  
1x3=3  
Bass  
1x4=4  
Buzz  
1x5=5  
Bass  
  
2x1=2  
Buzz  
2x2=4  
Buzz  
2x3=6  
Buzz  
2x4=8  
Buzz  
2x5=10  
Buzz  
  
3x1=3  
Bass  
3x2=6  
Buzz  
3x3=9  
Bass  
3x4=12  
Buzz  
3x5=15  
Bass
```

```
4x1=4  
Buzz  
4x2=8  
Buzz  
4x3=12  
Buzz  
4x4=16  
Buzz  
4x5=20  
Buzz  
  
5x1=5  
Bass  
5x2=10  
Buzz  
5x3=15  
Bass  
5x4=20  
Buzz  
5x5=25  
Bass
```

```
Total de pares es: 16  
Total de impares es: 9
```

19.1

<b>Nombre de la función:</b> <b>tablas(plimite)</b>		<b>Versión: 1.0</b>
<b>Descripción:</b> Función tabla del 1 hasta 5 for como Parámetros		
limite	Tipo de variable: Int	
tabla	Tipo de variable: Int	
contador	Tipo de variable: Int	
par	Tipo de variable: Int	
impar	Tipo de variable: Int	
resultado	Tipo de variable: Int	
msg	Tipo de variable: string	



# Tecnólogo en Análisis y Desarrollo de Software

## Ficha 2900177

### Funciones JS

Código:

```
// con parametros
let limite;
let par = 0;
let impar = 0;

function tablas(plimite){
    limite = plimite;
    let tabla;
    let contador;
    let resultado;
    let msg = "";

    for(tabla =1; limite>=tabla; tabla++){
        for(contador =1; limite>=contador; contador++){
            resultado = tabla*contador;
            msg += (tabla)+ "x" +(contador)+ "="+ (resultado) +"\n";
            msg += parImpar(resultado);
        }

        msg += "\n";
    }
    return msg+ "Total de pares es: "+par+"\n"+
        "Total de impares es: "+impar+"\n";
}
```

<b>Nombre de la función:</b> <b>parImpar(resultado)</b>		<b>Versión: 1.0</b>
<b>Descripción:</b> Función tabla del 1 hasta 5 while como Parámetros		
par	Tipo de variable: Int	
impar	Tipo de variable: Int	
resultado	Tipo de variable: Int	



# Tecnólogo en Análisis y Desarrollo de Software

## Ficha 2900177

### Funciones JS

Código:

```
function parImpar(resultado){  
  if(resultado%2==0){  
    par = par+1  
    return "Buzz\n";  
  }else{  
    impar = impar+1  
    return "Bass\n";  
  }  
}
```

Con parametros:

1x1=1  
Bass  
1x2=2  
Buzz  
1x3=3  
Bass  
1x4=4  
Buzz  
1x5=5  
Bass

2x1=2  
Buzz  
2x2=4  
Buzz  
2x3=6  
Buzz  
2x4=8  
Buzz  
2x5=10  
Buzz

3x1=3  
Bass  
3x2=6  
Buzz  
3x3=9  
Bass  
3x4=12  
Buzz  
3x5=15  
Bass

4x1=4  
Buzz  
4x2=8  
Buzz  
4x3=12  
Buzz  
4x4=16  
Buzz  
4x5=20  
Buzz

5x1=5  
Bass  
5x2=10  
Buzz  
5x3=15  
Bass  
5x4=20  
Buzz  
5x5=25  
Bass

Total de pares es: 16  
Total de impares es: 9

Nombre de la función: conts tablasExp = function(plimite)		Versión: 2.0
Descripción: Función tabla del 1 hasta 5 for como Expresion		
limite	Tipo de variable: Int	
tabla	Tipo de variable: Int	
contador	Tipo de variable: Int	
par	Tipo de variable: Int	
impar	Tipo de variable: Int	
resultado	Tipo de variable: Int	





# Tecnólogo en Análisis y Desarrollo de Software

## Ficha 2900177

### Funciones JS

msg	Tipo de variable: string
-----	--------------------------

Código:

```
const tablasExp= function(plimite){
  limite = plimite;
  let tabla;
  let contador;
  let resultado;
  let msg = "";

  for(tabla =1; limite>=tabla; tabla++){
    for(contador =1; limite>=contador; contador++){
      resultado = tabla*contador;
      msg += (tabla)+ "x" +(contador)+ "="+ (resultado) +"\n";
      msg += parImpare(resultado);
    }
    msg += "\n";
  }
  return msg+ "Total de pares es: "+par+"\n"+
    "Total de impares es: "+impar+"\n";
}
```

Nombre de la función:

**const parImpare = function(resultado)**

Versión: 2.0

Descripción:

Función tabla del 1 hasta 5 for como Expresion

par	Tipo de variable: Int
-----	-----------------------

impar	Tipo de variable: Int
-------	-----------------------

resultado	Tipo de variable: Int
-----------	-----------------------

Código:

```
const parImpare = function(resultado){
  let par;
  let impar;
  if(resultado%2==0){
    par = par+1
    return "Buzz\n";
  }else{
    impar = impar+1
    return "Bass\n";
  }
}
```



# Tecnólogo en Análisis y Desarrollo de Software

Ficha 2900177

## Funciones JS

Como Expresion:

1x1=1

Bass

1x2=2

Buzz

1x3=3

Bass

1x4=4

Buzz

1x5=5

Bass

2x1=2

Buzz

2x2=4

Buzz

2x3=6

Buzz

2x4=8

Buzz

2x5=10

Buzz

3x1=3

Bass

3x2=6

Buzz

3x3=9

Bass

3x4=12

Buzz

3x5=15

Bass

4x1=4

Buzz

4x2=8

Buzz

4x3=12

Buzz

4x4=16

Buzz

4x5=20

Buzz

5x1=5

Bass

5x2=10

Buzz

5x3=15

Bass

5x4=20

Buzz

5x5=25

Bass

Total de pares es: 16

Total de impares es: 9