



Tecnólogo en Análisis y Desarrollo de
Software
Ficha

Funciones JS

JUAN DAVID ARTUNDUAGA GÓMEZ

INSTRUCTOR: ANDRES MORENO COLLAZOS

SENA – CIES

2900177

2024



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

01 - SALUDO

Nombre de la función: saludo(psaludo)		Versión: 1.0
Descripción: Función que saluda		
saludar	Tipo de variable: Alfanumérico caracter	
Código: <pre>/** funtion como parametro */ function saludo(psaludo){ let saludar = psaludo; return saludar; } /** fin funtion como parametro */</pre>		

Nombre de la función: saludoExp = function(psaludo)		Versión: 2.0
Descripción: Función que saluda		
saludar	Tipo de variable: carácter	
Código: <pre>/** funtion como exprecion */ const saludoExp = function(psaludo){ let saludar = psaludo; return saludar; } /** fin funtion como exprecion */</pre>		
consola: <div><div>function parametro hola mundo</div><div>index.html:12</div><div>function expresion hola mundo</div><div>index.html:13</div></div>		



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

2 - suma

Nombre de la función: sumar(numero1 , numero2)		Versión: 1.0
Descripción: Función que suma dos números		
1 suma 2 numeroUno 3 numeroDos	Tipo de variable: Entero Entero Entero	
Código: <pre>/** funtion como parametro */ function sumar(numero1 , numero2){ let suma let numeroUno=numero1 let numeroDos=numero2 suma= numeroUno+numeroDos return suma</pre>		

Nombre de la función: sumarExp =function(numero1, numero2)		Versión: 2.0
Descripción: Función que suma dos números		
1 suma 2 numeroUno 3 numeroDos	Tipo de variable: Entero Entero Entero	
Código: <pre>/**funtion como expresion */ const sumarExp=function(numero1,numero2){ let suma let numeroUno=numero1 let numeroDos=numero2 suma=numeroUno+numeroDos return suma }</pre>		
Consola: <pre>suma como parametro 11 index.html:15 suma como expresion 11 index.html:16 suma como parametro 8 index.html:19 suma como expresion 7 index.html:20</pre>		



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

3 - operaciones

Nombre de la función: restar(num1,num2)		Versión: 1.0
Descripción: Función de operaciones básicas (sumar, restar, multiplicar, dividir)		
1 resta	Tipo de variable: numérica	
2 numeroUno	Entero	
3 numeroDos	Entero	
Código:		
<pre>let numeroUno; let numeroDos; /** funtion como parametro */ function restar(num1,num2){ let resta; numeroUno=num1; numeroDos=num2; resta = numeroUno - numeroDos; return resta; }</pre>		



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: multiplicar(num1,num2)		Versión: 1.0
Descripción: Función de operaciones básicas (sumar, restar, multiplicar, dividir)		
1 multiplicacion 2 numeroUno 3 numeroDos	Tipo de variable: Entero Entero Entero	
Código: <pre>function multiplicar(num1,num2){ let multiplicacion; numeroUno = num1; numeroDos = num2; multiplicacion = numeroUno * numeroDos; return multiplicacion; }</pre>		

Nombre de la función: dividir(num1,num2)		Versión: 1.0
Descripción: Función de operaciones básicas (sumar, restar, multiplicar, dividir)		
1 division	Tipo de variable: numérica	
2 numeroUno	Entero	
3 numeroDos	Entero	
Código:		
<pre>function dividir(num1,num2){ let division; numeroUno = num1; numeroDos = num2; division = numeroUno / numeroDos; return division; }</pre>		



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: restarExp = function(num1,num2)		Versión: 2.0
Descripción: Función de operaciones básicas (sumar, restar, multiplicar, dividir)		
1 resta 2 numeroUno 3 numeroDos	Tipo de variable: numérica Entero Entero Entero	
Código: <pre>/** funcion como expersion */ const restarExp=function(num1,num2){ let resta; numeroUno=num1; numeroDos=num2; resta = numeroUno - numeroDos; return resta; }</pre>		

Nombre de la función: multiplicarExp = function(num1,num2)		Versión: 2.0
Descripción: Función de operaciones básicas (sumar, restar, multiplicar, dividir)		
1 multiplicacion 2 numeroUno 3 numeroDos	Tipo de variable: Entero Entero Entero	
Código: <pre>const multiplicarExp=function(num1,num2){ let multiplicacion; numeroUno = num1; numeroDos = num2; multiplicacion = numeroUno * numeroDos; return multiplicacion; }</pre>		



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: dividirExp= function(num1,num2)

Versión: 2.0

Descripción:

Función de operaciones básicas (sumar, restar, multiplicar, dividir)

1 division
2 numeroUno
3 numeroDos

Tipo de variable:

Entero
Entero
Entero

Código:

```
const dividirExp=function(num1,num2){  
  let division;  
  numeroUno = num1;  
  numeroDos = num2;  
  division = numeroUno / numeroDos;  
  
  return division;  
}
```

Consola:

restar con parametro 0	index.html:11
multiplicar con parametro 4	index.html:12
dividir con parametro 1	index.html:13
-----	index.html:14
restar con expresiones 0	index.html:15
multiplicar con expresiones 4	index.html:16
dividir con expresiones 1	index.html:17



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

4 - porcentaje

Nombre de la función: porNota(pnota,pporce)		Versión: 1.0
Descripción: Halla el porcentaje de un número		
1 nota 2 porce 3 resultadoNota	Tipo de variable: Entero Float desimales	
Código: <pre>/** funtion como parametro */ function porNota(pnota,pporce){ let nota = pnota; let porce = pporce; let resultadoNot; resultadoNot=nota/porce; return resultadoNot; } /** fin funtion como parametro */</pre>		



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: `porNotaExp=function(pnota,pporce)`

Versión: 2.0

Descripción:

Halla el porcentaje de un número

1 nota
2 porce
3 resultadoNota

Tipo de variable:

Entero
Float
desimales

Código:

```
/** funtion como expresion */  
const porNotaExp=function(pnota,pporce){  
    let nota = pnota;  
    let porce = pporce;  
    let resultadoNot;  
    resultadoNot=nota/porce;  
  
    return resultadoNot;  
}  
/** fin funtion como expresion */
```

Consola:

funtion como parametro 0.5

[index.html:12](#)

funtion como expresion 0.4

[index.html:13](#)



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

5 – promedio de nota

Nombre de la función: proNota(nota1,nota2,nota3)		Versión: 1.0
Descripción: Imprime el promedio de tres notas		
1 suma 2 notaUno 3 notaDos 4 notaTres	Tipo de variable: Entero Entero Entero Entero	
Código:		
<pre>/** funtion como parametro */ function proNota(nota1,nota2,nota3){ let suma let notaUno= nota1 let notaDos=nota2 let notaTres=nota3 suma=(notaUno+notaDos+notaTres)/3 return suma } /** fin funtion como parametro */</pre>		



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: proNotaEsp=function(nota1,nota2,nota3,)		Versión: 2.0
Descripción: Imprime el promedio de tres notas		
1 suma 2 notaUno 3 notaDos 4 notaTres	Tipo de variable: Entero Entero Entero Entero	
Código: <pre>/** funtion como expresion */ const proNotaExp=function(nota1,nota2,nota3){ let suma let notaUno= nota1 let notaDos=nota2 let notaTres=nota3 suma=(notaUno+notaDos+notaTres)/3 return suma } /** fin funtion como expresion */</pre> <div><div>el promedio de las tre notas con parametro</div><div>3.1666666666666665</div><div>index.html:11</div></div> <div><div>_____</div><div>index.html:12</div></div> <div><div>el promedio de las tre notas con Expresion 2.2</div><div>index.html:13</div></div>		



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

6 – porcentaje

Nombre de la función: porNota(pnota,pporce)		Versión: 1.0
Descripción: Calcular el porcentaje ponderado de tres notas. Nota 1 = 30% Nota 2 = 30% Nota 3 = 40%		
1 nota 2 porcentaje 3 renota	Tipo de variable: Entero float desimales	
Código:		
<pre>/** funtion como parametro */ function porNota(pnota,pporce){ let nota=pnota; let porcentaje=pporce; let renota; renota = nota * porcentaje; return renota; } /** fin funtion como parametro */</pre>		



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: `porNotaExp=function(pnota,pporce)`

Versión: 2.0

Descripción:

Calcular el porcentaje ponderado de tres notas.

Nota 1 = 30% Nota 2 = 30% Nota 3 = 40%

1 nota
2 porcentaje
3 renota

Tipo de variable:

Entero

float

desimales

Código:

```
/** funtion como expresion */  
const porNotaExp=function(pnota,pporce){  
  let nota = pnota;  
  let porcentaje = pporce;  
  let renota;  
  renota = nota * porcentaje;  
  return renota  
}  
/** fin funtion como expresion */
```

el porcentaje de las notas con funtion como [index.html:19](#)
parametro

NOTA 1 =0.8999999999999999 [index.html:20](#)

NOTA 2 =1.5 [index.html:21](#)

NOTA 3 =1.8 [index.html:22](#)

suma de notas= 4.2 [index.html:23](#)

[index.html:24](#)

el porcentaje de las notas con funtion como [index.html:25](#)
expresion

NOTA 1 =0.69 [index.html:26](#)

NOTA 2 =1.5 [index.html:27](#)

NOTA 3 =1.4000000000000001 [index.html:28](#)

suma de notas= 3.59 [index.html:29](#)



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

7 – áreas de figuras geométricas

Nombre de la función: Acuadrado (plado, pcuadrado – pbase,paltura,prectangulo - pbase,paltura,ptriangulo)		Versión: 1.0
Descripción: Calcula el área de un cuadrado, rectángulo y triángulo		
1 cuadrado 2 lado	Tipo de variable: Entero Entero	
Código: <pre>/** funtion como parametro */ function Acuadrado(Plado,pcuadrado){ let cuadrado = pcuadrado; let lado = Plado cuadrado = lado * lado return cuadrado }</pre>		

Nombre de la función: Arectangulo (pbase,paltura,prectangulo)		Versión: 1.0
Descripción: Calcula el área de un cuadrado, rectángulo y triángulo		
1 rectangulo 2 base 3 altura	Tipo de variable: Entero Entero Entero	
Código: <pre>function Arectangulo(Pbase,paltura,prectangulo){ let rectangulo = prectangulo; let base = Pbase let altura = paltura rectangulo = base * altura return rectangulo }</pre>		



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: Atriangulo (pbase, paltura, ptriangulo)		Versión: 1.0
Descripción: Calcula el área de un cuadrado, rectángulo y triángulo		
1 triagunlo 2 base 3 altura	Tipo de variable: Entero Entero entero	
Código: <pre>function Atriangulo(Pbase,paltura,ptriangulo){ let triangulo = ptriangulo; let base = Pbase let altura = paltura triangulo = (base * altura)/2 return triangulo } /** fin funtion como parametro */</pre>		

Nombre de la función: AcuadradoExp = function(plado, pcuadrado)		Versión: 2.0
Descripción: Calcula el área de un cuadrado, rectángulo y triángulo		
1 cuadrado 2 lado	Tipo de variable: Entero Entero	
Código: <pre>/** funtion como expresion */ const AcuadradoExp = function(Plado,pcuadrado){ let cuadrado = pcuadrado; let lado = Plado cuadrado = lado * lado return cuadrado }</pre>		



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: ArectanguloExp = function(pbase,paltura,prectangulo)		Versión: 2.0
Descripción: Calcula el área de un cuadrado, rectángulo y triángulo		
1 rectangulo	Tipo de variable: Entero	
2 base	Entero	
3 altura	Entero	
Código:		
<pre>const ArectanguloExp = function (Pbase,paltura,prectangulo){ let rectangulo = prectangulo; let base = Pbase let altura = paltura rectangulo = base * altura return rectangulo }</pre>		



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: AtrianguloExp = function(pbase,paltura,ptriangulo)		Versión: 2.0																		
Descripción: Calcula el área de un cuadrado, rectángulo y triángulo																				
1 triangulo 2 base 3 altura	Tipo de variable: Entero Entero Entero																			
Código: <pre>const AtrianguloExp = function(Pbase,paltura,ptriangulo){ let triagulo = ptriangulo; let base = Pbase let altura = paltura triagulo = (base * altura)/2 return triagulo } /** fin funtion como expresion */</pre>																				
Consola: <table><tr><td>el area con las function como parametros</td><td>index.html:18</td></tr><tr><td>el area del cuadrado es 16</td><td>index.html:19</td></tr><tr><td>el area del rectangulo es 50</td><td>index.html:20</td></tr><tr><td>el area del triangulo es 25</td><td>index.html:21</td></tr><tr><td colspan="2"><hr/></td></tr><tr><td>el area con las function como exprecion</td><td>index.html:24</td></tr><tr><td>el area del cuadrado es 9</td><td>index.html:25</td></tr><tr><td>el area del rectangulo es 16</td><td>index.html:26</td></tr><tr><td>el area del triangulo es 36</td><td>index.html:27</td></tr></table>			el area con las function como parametros	index.html:18	el area del cuadrado es 16	index.html:19	el area del rectangulo es 50	index.html:20	el area del triangulo es 25	index.html:21	<hr/>		el area con las function como exprecion	index.html:24	el area del cuadrado es 9	index.html:25	el area del rectangulo es 16	index.html:26	el area del triangulo es 36	index.html:27
el area con las function como parametros	index.html:18																			
el area del cuadrado es 16	index.html:19																			
el area del rectangulo es 50	index.html:20																			
el area del triangulo es 25	index.html:21																			
<hr/>																				
el area con las function como exprecion	index.html:24																			
el area del cuadrado es 9	index.html:25																			
el area del rectangulo es 16	index.html:26																			
el area del triangulo es 36	index.html:27																			



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

8 – sueldo de persona

Nombre de la función: salario(pdiastra,pvalordias)		Versión: 1.0
Descripción: Calcular el sueldo de un trabajador con deducciones de pensión, salud y ARL. (pensión = 0.16, salud = 0.12, ARL = 0.052)		
1 psalario 2 diastra 3 valordias	Tipo de variable: Entero Entero Entero	
Código: <pre>function salario(pdiastra, pvalordias){ let psalario let diastra = pdiastra let valordias = pvalordias psalario = diastra * valordias return psalario }</pre>		

Nombre de la función: salud(pdiastra,pvalordias)		Versión: 1.0
Descripción: Calcular el sueldo de un trabajador con deducciones de pensión, salud y ARL. (pensión = 0.16, salud = 0.12, ARL = 0.052)		
1 salario	Tipo de variable: Entero	
Código: <pre>function salud(pdiastra, pvalordias){ let psalud psalud = salario(pdiastra, pvalordias)*0.12 return psalud }</pre>		

Nombre de la función: pension(pdiastra,pvalordias)		Versión: 1.0
---	--	---------------------



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Descripción:

Calcular el sueldo de un trabajador con deducciones de pensión, salud y ARL. (pensión = 0.16, salud = 0.12, ARL = 0.052)

1 pensionv

Tipo de variable:
Entero

Código:

```
function pension(pdiastra, pvalordias){  
    let pensionv  
    pensionv = salario(pdiastra, pvalordias)*0.16  
    return pensionv  
}
```

Nombre de la función: arl(pdiastra,pvalordias)

Versión: 1.0

Descripción:

Calcular el sueldo de un trabajador con deducciones de pensión, salud y ARL. (pensión = 0.16, salud = 0.12, ARL = 0.052)

1 arlv

Tipo de variable:
Entero

Código:

```
function arl(pdiastra, pvalordias){  
    let arlv  
    arlv = salario(pdiastra, pvalordias)*0.052  
    return arlv  
}
```

Nombre de la función: pensionTotal(pdiastra,pvalordias)

Versión: 1.0



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Descripción:

Calcular el sueldo de un trabajador con deducciones de pensión, salud y ARL. (pensión = 0.16, salud = 0.12, ARL = 0.052)

1 pensionT

Tipo de variable:

Entero

Código:

```
function pensionTotal(pdiastra,pvalordias){  
    let pensionT  
    let psalud = salud(pdiastra, pvalordias)  
    let ppension = pension(pdiastra, pvalordias)  
    let parl = arl(pdiastra, pvalordias)  
    let psalario = salario(pdiastra, pvalordias)  
    pensionT = psalario - (psalud + ppension + parl)  
    return pensionT  
}
```

Nombre de la función: salarioExp = function(pdiastra,pvalordias)

Versión: 2.0

Descripción:

Calcular el sueldo de un trabajador con deducciones de pensión, salud y ARL. (pensión = 0.16, salud = 0.12, ARL = 0.052)

1 psalario
2 diastra
3 valordias

Tipo de variable:

Entero
Entero
Entero

Código:

```
const salarioExp = function(pdiastra, pvalordias){  
    let psalario  
    let diastra = pdiastra  
    let valordias = pvalordias  
    psalario = diastra * valordias  
    return psalario  
}
```

Nombre de la función: pensionExp = function(pdiastra,pvalordias)

Versión: 2.0

Descripción:



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Calcular el sueldo de un trabajador con deducciones de pensión, salud y ARL. (pensión = 0.16, salud = 0.12, ARL = 0.052)

1 pensionv

Tipo de variable:
Entero

Código:

```
const pensionExp = function(pdiastra, pvalordias){  
  let pensionv  
  pensionv = salario(pdiastra, pvalordias)*0.16  
  return pensionv  
}
```

Nombre de la función: arlExp = function(pdiastra,pvalordias)

Versión: 2.0

Descripción:

Calcular el sueldo de un trabajador con deducciones de pensión, salud y ARL. (pensión = 0.16, salud = 0.12, ARL = 0.052)

1 arlv

Tipo de variable:
Entero

Código:

```
const arlExp = function(pdiastra, pvalordias){  
  let arlv  
  arlv = salario(pdiastra, pvalordias)*0.052  
  return arlv  
}
```

Nombre de la función: pensionTotalExp = function(pdiastra,pvalordias)

Versión: 2.0

Descripción:



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Calcular el sueldo de un trabajador con deducciones de pensión, salud y ARL. (pensión = 0.16, salud = 0.12, ARL = 0.052)

1 pensionT
2 psalud
3 ppension
4 parl
5 psalario

Tipo de variable:

Entero
Entero
Entero
Entero
Entero

Código:

```
const pensionTotalExp = function(pdiastra,pvalordias){  
  let pensionT  
  let psalud = saludExp(pdiastra, pvalordias)  
  let ppension = pensionExp(pdiastra, pvalordias)  
  let parl = arlExp(pdiastra, pvalordias)  
  let psalario = salarioExp(pdiastra, pvalordias)  
  pensionT = psalario - (psalud + ppension + parl)  
  return pensionT  
}
```

Consola:

```
function como parametro      index.html:13  
el salario de la persona es de: 1200000      index.html:14  
la salud de la persona es de: 144000      index.html:15  
la pension de la persona es de 192000      index.html:16  
el arl de la persona es de 62400      index.html:17  
el sueldo del trabajador es de: 801600      index.html:18  
function como expresion      index.html:19  
el salario de la persona es de: 1200000      index.html:20  
la salud de la persona es de: 144000      index.html:21  
la pension de la persona es de 192000      index.html:22  
el arl de la persona es de 62400      index.html:23  
el sueldo del trabajador es de: 801600      index.html:24
```



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: `mayor(pnumero1,pnumero2)`

Versión: 1.0

Descripción:

Imprimir el mayor de dos números

1 numero1

2 numero2

Tipo de variable:

Entero

Entero

Código:

```
/** funtion como parametro */  
function mayor(pnumero1,pnumero2){  
  let numero1=pnumero1  
  let numero2=pnumero2  
  if (numero1 > numero2){  
    return numero1  
  }  
  else{  
    return numero2  
  }  
}  
/** fin funtion como parametro */
```



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: mayorExp=function(pnumero1,pnumero2))

Versión: 2.0

Descripción:

Imprimir el mayor de dos números

1 numero1

Tipo de variable:

Entero

2 numero2

Entero

Código:

```
/** funtion como expresion */
const mayorExp=function(pnumero1,pnumero2){
  let numero1=pnumero1
  let numero2=pnumero2
  if (numero1 > numero2){
    return numero1
  }
  else{
    return numero2
  }
}
/** fin funtion como expresion */
```

Consola:

```
operacion con function como parametro      index.html:13
el numero mayor es: 5                       index.html:14
_____                                     index.html:15
operacion con function como expresion      index.html:16
el numero mayor es: 9                       index.html:17
```




Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: mayorEdad(pnacimiento,pactual)

Versión: 1.0

Descripción:

Calcular la edad de una persona e imprimir si es mayor de edad

1 edad
2 nacimiento
3 alctual

Tipo de variable:

Entero
Fecha
Fecha

Código:

```
/** funtion como parametro */  
function mayorEdad(pnacimiento,pactual){  
    let edad  
    let nacimiento=pnacimiento  
    let alctual=pactual  
    edad = alctual - nacimiento  
    if (edad > 17){  
        return "es mayor de edad " + "\n" + edad;  
    }  
    else{  
        return "es menor de edad " + "\n" + edad;  
    }  
}  
/** fin funtion como parametro */
```

Nombre de la función: mayorEdadExp = function(pnacimiento,pactual)

Versión: 2.0



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Descripción:

Calcular la edad de una persona e imprimir si es mayor de edad

1 edad
2 nacimiento
3 alctual

Tipo de variable:

Entero
Fecha
Fecha

Código:

```
/** funtion como expresion */  
const mayorEdadExp = function(pnacimiento,pactual){  
  let edad  
  let nacimiento=pnacimiento  
  let alctual=pactual  
  edad = alctual - nacimiento  
  if (edad > 17){  
    return "es mayor de edad " + "\n" + edad;  
  }  
  else{  
    return "es menor de edad " + "\n" + edad;  
  }  
}  
/** fin funtion como expresion */
```

Consola:

```
function como parametro  
es mayor de edad  
20
```

[index.html:13](#)

```
function como exprecion  
es menor de edad  
14
```

[index.html:14](#)



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: mayor(pnumero1,pnumero2)

Versión: 1.0

Descripción:

imprime el mayor de dos números validando que no sean iguales

1 numero1

Tipo de variable:

Entero

2 numero2

Entero

Código:

```
/** funtion como parametro */
function mayor(pnumero1,pnumero2){
  let numero1 = pnumero1
  let numero2 = pnumero2
  if (numero1 != numero2){
    if (numero1 > numero2){
      return "el numero 1 es mayor: " + "\n" + numero1
    }
    else {
      return "el numero 2 es mayor: " + "\n" + numero2
    }
  }
  else {
    return "los numeros son iguales "
  }
}
/** fin funtion como parametro */
```



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: mayorExp=function(pnumero1,pnumero2)		Versión: 2.0
Descripción: imprime el mayor de dos números validando que no sean iguales		
1 numero1 2 numero2	Tipo de variable: Entero Entero	
Código:		
<pre>/** funtion como expresion */ const mayorExp=function(pnumero1,pnumero2){ let numero1 = pnumero1 let numero2 = pnumero2 if (numero1 != numero2){ if (numero1 > numero2){ return "el numero 1 es mayor: " + "\n" + numero1 } else { return "el numero 2 es mayor: " + "\n" + numero2 } } else { return "los numeros son iguales " } } /** fin funtion como expresion */</pre>		
Consola:		
funtion como parametros los numeros son iguales		index.html:13
funtion como expresion el numero 2 es mayor: 5		index.html:14

12 - mayor de tres números



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: mayor(pnumero1,pnumero2,pnumero3)	Versión: 1.0
Descripción: Imprime el mayor de tres números con la misma validación	

1 numero1 2 numero2 3 numero3	Tipo de variable: Entero Entero Entero
-------------------------------------	---

Código:

```
/** funtion como parametro */  
function mayor(pnumero1,pnumero2,pnumero3){  
    let numero1 = pnumero1;  
    let numero2 = pnumero2;  
    let numero3 = pnumero3;  
    if(numero1 == numero2 && numero1 == numero3 && numero3 == numero2){  
        return "los 3 numeros son iguales" ;  
    }  
    else{  
        if(numero1 > numero2 && numero1 > numero3){  
            return "el numero mayor es el: " + "\n" + numero1;  
        }  
        else{  
            if (numero2 > numero1 && numero2 > numero3){  
                return "el numero mayor es el: " + "\n" + numero2;  
            }  
            else{  
                return "el numero mayor es el: " + "\n" + numero3;  
            }  
        }  
    }  
}  
/** fin funtion como parametro */
```



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: mayorExp = function(pnumero1,pnumero2,pnumero3)

Versión: 2.0

Descripción: Imprime el mayor de tres números con la misma validación

1 numero1	Tipo de variable:
2 numero2	Entero
3 numero3	Entero

Código:

```
/** funtion como expresion */
const mayorExp = function(pnumero1,pnumero2,pnumero3){
  let numero1 = pnumero1;
  let numero2 = pnumero2;
  let numero3 = pnumero3;
  if(numero1 == numero2 && numero1 == numero3 && numero3 == numero2){
    return "los 3 numeros son iguales" ;
  }
  else{
    if(numero1 > numero2 && numero1 > numero3){
      return "el numero mayor es el: " + "\n" + numero1;
    }
    else{
      if (numero2 > numero1 && numero2 > numero3){
        return "el numero mayor es el: " + "\n" + numero2;
      }
      else{
        return "el numero mayor es el: " + "\n" + numero3;
      }
    }
  }
}
/** fin funtion como expresion */
```

Consola:

```
fuction como parametro                                index.html:13
el numero mayor es el:
3

fuction como expresion                                index.html:14
el numero mayor es el:
9
```



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: Acuadrado(plado1, plado2,plado3)

Versión: 1.0

Descripción:

Calcular el área 3 de cuadrados e imprimir el mayor Calcula Área Tipo de variable: Alfanumérico

	Tipo de variable:
1 cuadrado1	Entero
2 cuadrado2	Entero
3 cuadrado3	Entero
4 lado1	Entero
5 lado2	Entero
6 lado3	Entero

Código:

```
/** funtion como parametro */
function Acuadrado(Plado1,plado2,plado3){
    let cuadrado1;
    let cuadrado2;
    let cuadrado3;
    let lado1 = Plado1
    let lado2 = plado2
    let lado3 = plado3
    cuadrado1 = lado1 * lado1;
    cuadrado2 = lado2 * lado2;
    cuadrado3 = lado3 * lado3
    if(cuadrado1 == cuadrado2 && cuadrado1 == cuadrado3 && cuadrado3 == cuadrado2){
        return "los 3 ncadrado son iguales" ;
    }
    else{
        if(cuadrado1 > cuadrado2 && cuadrado1 > cuadrado3){
            return "el area de la figura del cuadrado 1 es mayor: " + "\n" + cuadrado1;
        }
        else{
            if (cuadrado2 > cuadrado1 && cuadrado2 > cuadrado3){
                return "el area de la figura del cuadrado 2 es mayor: " + "\n" + cuadrado2;
            }
            else{
                return "el area de la figura del cuadrado 3 es mayor: " + "\n" + cuadrado3;
            }
        }
    }
}
/** fin funtion como parametro */
```



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: AcuadradoExp = function(plado1, plado2,plado3)

Versión: 2.0

Descripción:

Calcular el área 3 de cuadrados e imprimir el mayor Calcula Área Tipo de variable: Alfanumérico

1 cuadrado1
2 cuadrado2
3 cuadrado3
4 lado1
5 lado2
6 lado3

Tipo de variable:

Entero
Entero
Entero
Entero
Entero
Entero

Código:

```
/** funtion como expresion */
const AcuadradoExp = function(Plado1,plado2,plado3){
  let cuadrado1;
  let cuadrado2;
  let cuadrado3;
  let lado1 = Plado1
  let lado2 = plado2
  let lado3 = plado3
  cuadrado1 = lado1 * lado1;
  cuadrado2 = lado2 * lado2;
  cuadrado3 = lado3 * lado3
  if(cuadrado1 == cuadrado2 && cuadrado1 == cuadrado3 && cuadrado3 == cuadrado2){
    return "los 3 ncadrado son iguales" ;
  }
  else{
    if(cuadrado1 > cuadrado2 && cuadrado1 > cuadrado3){
      return "el area de la figura del cuadrado 1 es mayor: " + "\n" + cuadrado1;
    }
    else{
      if (cuadrado2 > cuadrado1 && cuadrado2 > cuadrado3){
        return "el area de la figura del cuadrado 2 es mayor: " + "\n" + cuadrado2;
      }
      else{
        return "el area de la figura del cuadrado 3 es mayor: " + "\n" + cuadrado3;
      }
    }
  }
}
/** fin funtion como expresion */
```

Consola:

```
fuction como parametro                                index.html:13
el area de la figura del cuadrado 3 es mayor:
25

fuction como expresion                                index.html:14
el area de la figura del cuadrado 1 es mayor:
81
```




Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: mayorEdad1(pnacimiento1,pactual)

Versión: 1.0

Descripción:

Calcular la edad de una persona e imprimir si es mayor o menor de edad y calcular el promedio de edades

1 persona1
2 nacimiento1
3 alctual

Tipo de variable:

Entero

Fecha

Fecha

Código:

```
/** funtion como parametro */  
let persona1  
let persona2  
let persona3  
  
function mayorEdad1(pnacimiento1,pactual){  
    let nacimiento1=pnacimiento1;  
    let alctual=pactual;  
    persona1 = alctual - nacimiento1  
  
    if (persona1 > 17){  
        return "es mayor de edad " + "\n" + persona1;  
    }  
    else{  
        return "es menor de edad " + "\n" + persona1;  
    }  
}
```



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: mayorEdad2(pnacimiento2,pactual)

Versión: 1.0

Descripción:

Calcular la edad de una persona e imprimir si es mayor o menor de edad y calcular el promedio de edades

1 persona2
2 nacimiento2
3 alctual

Tipo de variable:

Entero

Fecha

Fecha

Código:

```
}  
function mayorEdad2(pnacimiento2,pactual){  
  let nacimiento2=pnacimiento2;  
  let alctual=pactual;  
  persona2 = alctual - nacimiento2  
  if (persona2 > 17){  
    return "es mayor de edad " + "\n" + persona2;  
  }  
  else{  
    return "es menor de edad " + "\n" + persona2;  
  }  
}
```

Nombre de la función: mayorEdad3(pnacimiento3,pactual)

Versión: 1.0

Descripción:



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Calcular la edad de una persona e imprimir si es mayor o menor de edad y calcular el promedio de edades

1 persona3

2 nacimiento3

3 alctual

Tipo de variable:

Entero

Fecha

Fecha

Código:

```
function mayorEdad3(pnacimiento3,pactual){
    let nacimiento3=pnacimiento3;
    let alctual=pactual;
    persona3 = alctual - nacimiento3

    if (persona3 > 17){
        return "es mayor de edad " + "\n" + persona3;
    }
    else{
        return "es menor de edad " + "\n" + persona3;
    }
}
```

Nombre de la función: mayorEdadTotal(mayor1,mayor2,mayor3)

Versión: 1.0



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Descripción:

Calcular la edad de una persona e imprimir si es mayor o menor de edad y calcular el promedio de edades

1 promedio

Tipo de variable: alfanumérico

Entero

Código:

```
function mayorEdadTotal(mayor1,mayor2,mayor3){  
    let promedio = (mayor1 + mayor2 + mayor3)/3  
    if (promedio > 17){  
        return "el promedio es mayor de edad " + "\n" + promedio;  
    }  
    else{  
        return "el promedio es menor de edad " + "\n" + promedio;  
    }  
}  
  
/** fin funtion como parametro */
```

Nombre de la función: mayorEdad1Exp = function(pnacimiento1,pactual)

Versión: 2.0

Descripción:

Calcular la edad de una persona e imprimir si es mayor o menor de edad y calcular el promedio de edades



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

1 persona1
2 nacimiento1
3 alctual

Tipo de variable:

Entero

Fecha

Fecha

Código:

```
/** funtion como expresion */  
const mayorEdad1Exp=function(pnacimiento1,pactual){  
    let nacimiento1=pnacimiento1;  
    let alctual=pactual;  
    persona1 = alctual - nacimiento1  
  
    if (persona1 > 17){  
        return "es mayor de edad " + "\n" + persona1;  
    }  
    else{  
        return "es menor de edad " + "\n" + persona1;  
    }  
}
```

Nombre de la función: mayorEdad2Exp = function (pnacimiento2,pactual)

Versión: 2.0



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Descripción:

Calcular la edad de una persona e imprimir si es mayor o menor de edad y calcular el promedio de edades

1 persona2
2 nacimiento2
3 alctual

Tipo de variable:

Entero

Fecha

Fecha

Código:

```
const mayorEdad2Exp=function(pnacimiento2,pactual){  
    let nacimiento2=pnacimiento2;  
    let alctual=pactual;  
    persona2 = alctual - nacimiento2  
    if (persona2 > 17){  
        return "es mayor de edad " + "\n" + persona2;  
    }  
    else{  
        return "es menor de edad " + "\n" + persona2;  
    }  
}
```

Nombre de la función: mayorEdad3Exp = function (pnacimiento3,pactual)

Versión: 2.0

Descripción:

Calcular la edad de una persona e imprimir si es mayor o menor de edad y calcular el promedio de edades



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

1 persona3
2 nacimiento3
3 alctual

Tipo de variable:

Entero

Fecha

Fecha

Código:

```
const mayorEdad3Exp=function(pnacimiento3,pactual){  
    let nacimiento3=pnacimiento3;  
    let alctual=pactual;  
    persona3 = alctual - nacimiento3  
  
    if (persona3 > 17){  
        return "es mayor de edad " + "\n" + persona3;  
    }  
    else{  
        return "es menor de edad " + "\n" + persona3;  
    }  
}
```

Nombre de la función: mayorEdadTotalExp = function (mayor1,mayor2,mayor3)

Versión: 2.0



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Descripción:

Calcular la edad de una persona e imprimir si es mayor o menor de edad y calcular el promedio de edades

1 promedioExp

Tipo de variable:

Entero

Código:

```
const mayorEdadTotalExp = function(mayor1,mayor2,mayor3){  
    let promedioExp = (mayor1 + mayor2 + mayor3)/3  
    if (promedioExp > 17){  
        return "el promedio es mayor de edad " + "\n" + promedioExp;  
    }  
    else{  
        return "el promedio es menor de edad " + "\n" + promedioExp;  
    }  
}  
  
/** fin funtion como expresion */
```

Consola:

function como parametro	index.html:11
es mayor de edad	index.html:12
19	
es mayor de edad	index.html:13
18	
es menor de edad	index.html:14
17	
el promedio es mayor de edad	index.html:15
18	
<hr/>	
function como parametro	index.html:17
es menor de edad	index.html:18
16	
es menor de edad	index.html:19
15	
es mayor de edad	index.html:20
23	
el promedio es mayor de edad	index.html:21
18	



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: salario(pdiastra, pvalordias)		Versión: 1.0
Descripción: Calcular el sueldo de una persona todo incluido		
1 psalario	Tipo de variable: alfanumérico	
2 diastra	Entero	
3 valordias	Entero	
Código:		
<pre>function salario(pdiastra, pvalordias){ let psalario let diastra = pdiastra let valordias = pvalordias psalario = diastra * valordias return psalario }</pre>		

Nombre de la función: subsidiotras(pdiastra, pvalordias)		Versión: 1.0
Descripción: Calcular el sueldo de una persona todo incluido		
1 salarioM 2 psalario 3 psubsidiotras	Tipo de variable: alfanumérico Entero Entero Entero	
Código: <pre>function subsidiotras(pdiastra, pvalordias){ let salarioM = 1300000 let psalario = salario(pdiastra, pvalordias) let psubsidiotras if (psalario > salarioM *2){ psubsidiotras = 0 } else{ psubsidiotras = 114000 } return psubsidiotras }</pre>		



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: salud(pdiastra, pvalordias)		Versión: 1.0
Descripción: Calcular el sueldo de una persona todo incluido		
1 psalud	Tipo de variable: alfanumérico Entero	
Código: <pre>function salud(pdiastra, pvalordias){ let psalud psalud = salario(pdiastra, pvalordias)*0.12 return psalud }</pre>		

Nombre de la función: pension(pdiastra, pvalordias)		Versión: 1.0
Descripción: Calcular el sueldo de una persona todo incluido		
1 pensionv	Tipo de variable: alfanumérico Entero	
Código: <pre>function pension(pdiastra, pvalordias){ let pensionv pensionv = salario(pdiastra, pvalordias)*0.16 return pensionv }</pre>		



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: arl(pdiastra, pvalordias)		Versión: 1.0
Descripción: Calcular el sueldo de una persona todo incluido		
1 arlv	Tipo de variable: alfanumérico Entero	
Código: <div><pre>function arl(pdiastra, pvalordias){ let arlv arlv = salario(pdiastra, pvalordias)*0.052 return arlv }</pre></div>		

Nombre de la función: retencion(pdiastra, pvalordias)		Versión: 1.0
Descripción: Calcular el sueldo de una persona todo incluido		
1 salarioM 2 psalario 2 pretencion	Tipo de variable: alfanumérico Entero Entero Entero	
Código: <pre>function retencion(pdiastra, pvalordias){ let salarioM = 1300000 let psalario = salario(pdiastra, pvalordias) let pretencion if (psalario > salarioM *4){ pretencion = psalario * 0.04 } else{ pretencion = 0 } return pretencion }</pre>		

Nombre de la función: deducibles(pdiastra, pvalordias)	Versión: 1.0
--	--------------



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Descripción:

Calcular el sueldo de una persona todo incluido

	Tipo de variable: alfanumérico
1 pensionT	Entero
2 psalud	Entero
3 ppension	Entero
4 parl	Entero
5 psalario	Entero
6 psubsidiotras	Entero
7 pretencion	Entero

Código:

```
function deducibles(pdiastra,pvalordias){
    let pensionT
    let psalud = salud(pdiastra, pvalordias)
    let ppension = pension(pdiastra, pvalordias)
    let parl = arl(pdiastra, pvalordias)
    let psalario = salario(pdiastra, pvalordias)
    let psubsidiotras = subsidiotras(pdiastra, pvalordias)
    let pretencion = retencion(pdiastra, pvalordias)
    pensionT = psalario + psubsidiotras - (psalud + ppension + parl + pretencion)
    return pensionT
}
/** fin funtion como parametro */
```



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: salarioExo=function(pdiastra, pvalordias)		Versión: 2.0
Descripción: Calcular el sueldo de una persona todo incluido		
1 psalario	Tipo de variable: alfanumérico	
2 diastra	Entero	
3 valordias	Entero	
Código: <pre>const salarioExp = function(pdiastra, pvalordias){ let psalario let diastra = pdiastra let valordias = pvalordias psalario = diastra * valordias return psalario }</pre>		

Nombre de la función: subsidiotrasExp= function(pdiastra, pvalordias)		Versión: 2.0
Descripción: Calcular el sueldo de una persona todo incluido		
1 salarioM 2 psalario 3 psubsidiotras	Tipo de variable: alfanumérico Entero Entero Entero	
Código: <pre>const subsidiotrasExp= function(pdiastra, pvalordias){ let salarioM = 1300000 let psalario = salario(pdiastra, pvalordias) let psubsidiotras if (psalario > salarioM *2){ psubsidiotras = 0 } else{ psubsidiotras = 114000 } return psubsidiotras }</pre>		



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: saludExp = function(pdiastra, pvalordias)		Versión: 2.0
Descripción: Calcular el sueldo de una persona todo incluido		
1 psalud	Tipo de variable: alfanumérico Entero	
Código: <pre>const saludExp = function(pdiastra, pvalordias){ let psalud psalud = salario(pdiastra, pvalordias)*0.12 return psalud }</pre>		

Nombre de la función: pensionExp= function(pdiastra, pvalordias)		Versión: 2.0
Descripción: Calcular el sueldo de una persona todo incluido		
1 pensionv	Tipo de variable: alfanumérico Entero	
Código: <pre>const pensionExp = function(pdiastra, pvalordias){ let pensionv pensionv = salario(pdiastra, pvalordias)*0.16 return pensionv }</pre>		

Nombre de la función: arlExp = function(pdiastra, pvalordias)		Versión: 2.0
---	--	--------------



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Descripción:

Calcular el sueldo de una persona todo incluido

1 arlv

Tipo de variable: alfanumérico
Entero

Código:

```
const arlExp = function(pdiastra, pvalordias){  
  let arlv  
  arlv = salario(pdiastra, pvalordias)*0.052  
  return arlv  
}
```

Nombre de la función: retencionExp = function(pdiastra, pvalordias)

Versión: 2.0

Descripción:

Calcular el sueldo de una persona todo incluido

1 salarioM

Tipo de variable: alfanumérico
Entero

2 psalario

Entero

2 pretencion

Entero

Código:

```
const retencionExp = function(pdiastra, pvalordias){  
  let salarioM = 1300000  
  let psalario = salario(pdiastra, pvalordias)  
  let pretencion  
  if (psalario > salarioM *4){  
    pretencion = psalario * 0.04  
  }  
  else{  
    pretencion = 0  
  }  
  
  return pretencion  
}
```



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: deduciblesExp = function(pdiastra, pvalordias)

Versión: 2.0

Descripción:

Calcular el sueldo de una persona todo incluido

	Tipo de variable: alfanumérico
1 pensionT	Entero
2 psalud	Entero
3 ppension	Entero
4 parl	Entero
5 psalario	Entero
6 psubsidiotras	Entero
7 pretencion	Entero

Código:

```
const deduciblesExp = function(pdiastra,pvalordias){
    let pensionT
    let psalud = salud(pdiastra, pvalordias)
    let ppension = pension(pdiastra, pvalordias)
    let parl = arl(pdiastra, pvalordias)
    let psalario = salario(pdiastra, pvalordias)
    let psubsidiotras = subsidiotras(pdiastra, pvalordias)
    let pretencion = retencion(pdiastra, pvalordias)
    pensionT = psalario + psubsidiotras - (psalud + ppension + parl + pretencion)
    return pensionT
}
```

Consola:



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

function como parametro	index.html:13
el salario de la persona es de: 1800000	index.html:14
el subsidio de transporte es de: 114000	index.html:15
la salud de la persona es de: 216000	index.html:16
la pension de la persona es de 288000	index.html:17
el arl de la persona es de 93600	index.html:18
su retencion de su salario es de: 0	index.html:19
el sueldo del trabajador es de: 1316400	index.html:20
function como exprecion	index.html:22
el salario de la persona es de: 1800000	index.html:23
el subsidio de transporte es de: 114000	index.html:24
la salud de la persona es de: 216000	index.html:25
la pension de la persona es de 288000	index.html:26
el arl de la persona es de 93600	index.html:27
su retencion de su salario es de: 0	index.html:28
el sueldo del trabajador es de: 1316400	index.html:29



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

16 – promedio de 3 notas

Nombre de la función: notapor(pnota1,pnota2,pnota3)		Versión: 1.0
Descripción: Calcular el promedio de notas y demás		
1 nota1 2 nota2 3 nota3 4 promedio1 5 promedio2 6 promedio3 7 sumapromedio	Tipo de variable: Decimal Decimal Decimal Decimal Decimal Decimal Decimal	
Código:		
<pre>/** funtion como parametro */ function notapor(pnota1,pnota2,pnota3){ let nota1 = pnota1 let nota2 = pnota2 let nota3 = pnota3 let promedio1 = nota1 * 0.2 let promedio2 = nota2 * 0.35 let promedio3 = nota3 * 0.45 let sumapromedio = promedio1 + promedio2 + promedio3 if(sumapromedio > 4.5){ return "es una nota superior " + "\n" + sumapromedio } else{ if(sumapromedio <= 4.5 && sumapromedio > 3.5){ return "es una nota buena " + "\n" + sumapromedio } else{ if(sumapromedio <= 3.5 && sumapromedio >= 3.0){ return "es una nota media " + "\n" + sumapromedio } else{ return "es una nota mala " + "\n" + sumapromedio } } } } /** fin funtion como parametro */</pre>		



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: notaporExp = function(pnota1,pnota2,pnota3)

Versión: 2.0

Descripción:

Calcular el promedio de notas y demás

1 nota1

Tipo de variable:

Decimal

2 nota2

Decimal

3 nota3

Decimal

4 promedio1

Decimal

5 promedio2

Decimal

6 promedio3

Decimal

7 sumapromedio

Decimal

Código:

```
/** funcion como expresion */
const notaporExp=function(pnota1,pnota2,pnota3){
  let nota1 = pnota1
  let nota2 = pnota2
  let nota3 = pnota3
  let promedio1 = nota1 * 0.2
  let promedio2 = nota2 * 0.35
  let promedio3 = nota3 * 0.45
  let sumapromedio = promedio1 + promedio2 + promedio3

  if(sumapromedio > 4.5){
    return "es una nota superior " + "\n" + sumapromedio
  }
  else{
    if(sumapromedio <= 4.5 && sumapromedio > 3.5){
      return "es una nota buena " + "\n" + sumapromedio
    }
    else{
      if(sumapromedio <= 3.5 && sumapromedio >= 3.0){
        return "es una nota media " + "\n" + sumapromedio
      }
      else{
        return "es una nota mala " + "\n" + sumapromedio
      }
    }
  }
}

/** fin funcion como expresion */
```

Consola:

es una nota buena
4.425

[index.html:11](#)

es una nota media
3.25

[index.html:12](#)



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

17 – contar del 1 al 5

Nombre de la función: imprimir()		Versión: 1.0
Descripción: Imprime los números del 1 al 5		
1 contador	Tipo de variable:	
2 numero	numerico	
3 resultado	Entero	
Código:		
<pre>/** funtion como parametro */ function imprimir(pnumero){ let contador = 0 let numero = pnumero let resultado = "" while(contador < numero){ contador = contador + 1 resultado += `\${contador}\n` } return resultado }</pre>		



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

18 – factorial de 5

Nombre de la función: imprimir()		Versión: 2.0
Descripción: Imprimir factorial de 5		
1 numero 2 contador 3 factorial	Tipo de variable: Entero Numérico Entero	
Código: <pre>/** funtion como parametro */ function imprimir(pnumero){ let numero = pnumero let contador = 0 let factoreal = 1 while (contador < numero) { contador = contador + 1 factoreal = factoreal * contador } return "el factor de 5 es de: " + "\n" + factoreal } /** fin funtion como parametro */</pre>		



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: imprimirExp=function()

Versión: 2.0

Descripción:

Imprimir factorial de 5

1 numero
2 contador
3 factorial

Tipo de variable:

Entero

Número

Entero

Código:

```
/** funtion como expresion */  
const imprimirExp=function(pnumero){  
  let numero = pnumero  
  let contador = 0  
  let factoreal = 1  
  while (contador < numero) {  
    contador = contador + 1  
    factoreal = factoreal * contador  
  }  
  return "el factor de 5 es de: " + "\n" + factoreal  
}  
  
/** fin funtion como expresion */
```

Consola:

function como parametro	index.html:11
el factor de 5 es de: 120	index.html:12
function como expresion	index.html:13
el factor de 5 es de: 120	index.html:14



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

19 – tabla de multiplicar del 5

Nombre de la función: tabla()		Versión: 2.0
Descripción: Imprimir la tabla del 5		
1 numero 2 contador 3 multi 4 resultado	Tipo de variable: Entero Numérico Entero Entero	
Código:		
<pre>/** funtion como parametro */ function tabla(pnumero){ let numero = pnumero let contador = 0 let multi = 0 let resultado = "" while(contador < numero){ contador = contador + 1 multi = numero * contador resultado += `\${numero + " x " + contador + " = " + multi}\n` } return resultado } /** fin funtion como parametro */</pre>		



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: tablaExp = function()

Versión: 2.0

Descripción:

Imprimir la tabla del 5

1 numero	Tipo de variable:
2 contador	Entero
3 multi	Numérico
4 resultado	Entero

Código:

```
/** funtion como expresion */
const tablaExp = function(pnumero){
  let numero = pnumero
  let contador = 0
  let multi = 0
  let resultado = ""
  while(contador < numero){
    contador = contador + 1
    multi = numero * contador
    resultado += `${numero + " x " + contador + " = " + multi}\n`
  }
  return resultado
}

/** fin funtion como expresion */
```

Consola:

```
function con parametros                                index.html:11
5 x 1 = 5                                              index.html:12
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25

function con expresion                                index.html:13
5 x 1 = 5                                              index.html:14
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
```



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

20 – tabla de multiplicar del 9 par o impar

Nombre de la función: tabla(pnumero, pnumero2)		Versión: 1.0
Descripción: Imprimir la tabla del 9 con los números pares e impares		
1 pares 2 impares 3 numero 4 contador 5 multi 6 numero2 7 resultado	Tipo de variable: Entero Entero Entero Numérico Entero Entero Entero	
Código: <pre>/** funcion como parametro */ function tabla(pnumero, pnumero2){ let pares = 0; let impares = 0; let numero = pnumero let contador = 0 let multi = 0 let numero2 = pnumero2 let resultado = "" while(contador < numero){ contador = contador + 1 multi = numero2 * contador resultado += `\${numero2} x \${ contador} = \${multi} \${determinarTipo(multi)} \n` pares += espar2(multi); impares += espar2(multi) -1; } return resultado }</pre>		



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: determinarTipo(resultado)		Versión: 1.0
Descripción: Imprimir la tabla del 9 con los números pares e impares		
	Tipo de variable:	
Código: <pre>function determinarTipo (resultado){ if(resultado %2 == 0){ return "pares"; } else { return " impares "; } }</pre>		

Nombre de la función: espar2(pnumero)		Versión: 1.0
Descripción: Imprimir la tabla del 9 con los números pares e impares		
1 numer 2 resultado	Tipo de variable: Entero Entero	
Código: <pre>function espar2(pnumero){ let numer = pnumero; let resultado = 0; if(numer %2 == 0){ resultado = 1; } else{ resultado = 0; } return resultado; } /** fin funtion como parametro */</pre>		



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: `tablaExp = function(pnumero, pnumero2)`

Versión: 2.0

Descripción:

Imprimir la tabla del 9 con los números pares e impares

1 pares
2 impares
3 numero
4 contador
5 multi
6 numero2
7 resultado

Tipo de variable:

Entero
Entero
Entero
Numérico
Entero
Entero
Entero

Código:

```
/** funtion como parametro */  
function tabla(pnumero, pnumero2){  
    let pares = 0;  
    let impares = 0;  
    let numero = pnumero  
    let contador = 0  
    let multi = 0  
    let numero2 = pnumero2  
    let resultado = ""  
    while(contador < numero){  
        contador = contador + 1  
        multi = numero2 * contador  
        resultado += `${numero2} x ${ contador} = ${multi} ${determinarTipo(multi)} \n`  
        pares += espar2(multi);  
        impares += espar2(multi);  
    }  
    return resultado  
}
```



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: `determinarTipoExp = function(resultado)`

Versión: 2.0

Descripción:

Imprimir la tabla del 9 con los números pares e impares

Tipo de variable:

Código:

```
const determinarTipoExp = function (resultado){  
  if(resultado %2 == 0){  
    return "pares";  
  }  
  else {  
    return " impares ";  
  }  
}
```



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: `espar2Exp = function(pnumero)`

Versión: 2.0

Descripción:

Imprimir la tabla del 9 con los números pares e impares

1 numer
2 resultado

Tipo de variable:

Entero

Entero

Código:

```
const espar2Exp = function(pnumero){  
  let number = pnumero;  
  let resultado = 0;  
  if(number %2 == 0){  
    resultado = 1;  
  }  
  else{  
    resultado = 0;  
  }  
  return resultado;  
}  
  
/** fin funtion como expresion */
```

Consola:

function con pararmetros [index.html:13](#)

9 x 1 = 9 impares [index.html:14](#)

9 x 2 = 18 pares

9 x 3 = 27 impares

9 x 4 = 36 pares

9 x 5 = 45 impares

function con exprecion [index.html:15](#)

9 x 1 = 9 impares [index.html:16](#)

9 x 2 = 18 pares

9 x 3 = 27 impares

9 x 4 = 36 pares

9 x 5 = 45 impares



Tecnólogo en Análisis y Desarrollo de
Software
Ficha

Funciones JS

21 – contar del 1 al 5

Nombre de la función: imprimir(pnumero)		Versión: 2.0
Descripción: Imprimir los números del 1 al 5		
1 contador 2 numero 3 resultado	Tipo de variable: Numérico Entero Entero	
Código:		
<pre>/** funtion como parametro */ function imprimir(pnumero){ let contador let numero = pnumero let resultado = "" for(contador = 1;contador <= numero; contador ++){ resultado += `\${contador}\n` } return resultado } /** fin funtion como parametro */</pre>		



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: imprimirExp=function() Versión: 2.0

Descripción:

Imprimir los números del 1 al 5

1 contador	Tipo de variable:
2 numero	Entero
3 resultado	Entero

Código:

```
/** funtion como expresion */  
const imprimirExp=function(pnumero){  
  let contador  
  let numero = pnumero  
  let resultado = ""  
  for(contador = 1;contador <= numero; contador ++ ){  
    resultado += `${contador}\n`  
  }  
  return resultado  
}  
/** fin funtion como expresion */
```

Consola

```
funcion como parametros      index.html:12  
1                             index.html:13  
2  
3  
4  
5  
funcion como expresion      index.html:14  
1                             index.html:15  
2  
3  
4  
5  
Live reload enabled.         index.html:45
```

>



Tecnólogo en Análisis y Desarrollo de
Software
Ficha

Funciones JS

22 – factorial del 5

Nombre de la función: imprimir()		Versión: 2.0
Descripción: Imprimir la factorial del 5		
1 numer 2 factorial 3 contador 4 resultado	Tipo de variable: Entero Entero numérico Entero	
Código:		
<pre>/** funtion como parametro */ function imprimir(pnumero){ let numer = pnumero let factorial = 1 let contador let resultado = "" for(contador = 1;contador <= numer; contador++){ factorial = factorial * contador resultado += `\${factorial}\n` } return resultado } /** fin funtion como parametro */</pre>		



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: imprimirExp=function()

Versión: 2.0**Descripción:**

Imprimir la factorial del 5

```
1 numer
2 factorial
3 contador
4 resultado
```

Tipo de variable:

Entero

Entero

Numérico

Entero

Código:

```
/** funcion como expresion */
const imprimirExp=function(pnumero){
  let numer = pnumero
  let factorial = 1
  let contador
  let resultado = ""
  for(contador = 1;contador <= numer; contador++){
    factorial = factorial * contador
    resultado += `${factorial}\n`
  }
  return resultado
}

/** fin funcion como expresion */
```

Consola:

funcntion como parametro	index.html:12
1	index.html:13
2	
6	
24	
120	
funcntion como expresion	index.html:14
1	index.html:15
2	
6	
24	
120	



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

23 – tabla de multiplicar del 5

Nombre de la función: tabla(pnumero)		Versión: 1.0
Descripción: Imprimir la tabla del 5		
1 numero	Tipo de variable: Entero	
2 contador		
3 multi		
4 resultado	Entero	Entero
Código:		
<pre>/** funtion como parametro */ function tabla(pnumero){ let numero = pnumero let contador let multi let resultado = "" for(contador = 1; contador <= numero; contador++){ multi = numero * contador resultado += `\${numero + " x " + contador+ " = " +multi}\n` } return resultado } /** fin funtion como parametro */</pre>		



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: tablaExp = function()

Versión: 2.0

Descripción:

Imprimir la tabla del 5

1 numero
2 contador
3 multi
4 resultado

Tipo de variable:

Entero

Entero

Entero

Código:

```
/** funtion como expresion */  
const tablaExp = function(pnumero){  
  let numero = pnumero  
  let contador  
  let multi  
  let resultado = ""  
  for(contador = 1; contador <= numero; contador++){  
    multi = numero * contador  
    resultado += `${numero + " x " + contador+ " = " +multi}\n`  
  }  
  return resultado  
}  
/** fin funtion como expresion */
```

Consola:

```
function con parametros           index.html:12  
5 x 1 = 5                         index.html:13  
5 x 2 = 10  
5 x 3 = 15  
5 x 4 = 20  
5 x 5 = 25  
function con expresion           index.html:14  
5 x 1 = 5                         index.html:15  
5 x 2 = 10  
5 x 3 = 15  
5 x 4 = 20  
5 x 5 = 25
```



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

24 -tabla de multiplicar del 9 par o impar

Nombre de la función: tabla(pnumero, pnumero2)		Versión: 1.0
Descripción: Imprimir la tabla del 9 con los números pares e impares		
1 pares 2 impares 3 numero 4 contador 5 multi 6 numero2 7 resultado	Tipo de variable: Entero Entero Entero Numérico Entero Entero Entero	
Código:		
<pre>/** funtion como parametro */ function tabla(pnumero,pnumero2){ let pares = 0 ; let impares = 0; let numero = pnumero; let contador = 0; let multi = 0; let numero2 = pnumero2; let resultado = ""; for(contador = 1;contador <= numero; contador ++){ multi = numero2 * contador resultado += `\${numero2} x \${ contador} = \${multi} \${determinarTipo(multi)} \n` pares += espar2(multi); impares += espar2(multi); } return resultado }</pre>		



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: determinarTipo (resultado)		Versión: 1.0
Descripción: Imprimir la tabla del 9 con los números pares e impares		
	Tipo de variable: alfanumérico	
Código:		
<pre>function determinarTipo (resultado){ if(resultado %2 == 0){ return "pares"; } else { return " impares "; } }</pre>		

Nombre de la función: espar2(pnumero)		Versión: 1.0
Descripción: Imprimir la tabla del 9 con los números pares e impares		
1 numer	Tipo de variable: alfanumérico	
2 resultado	Entero	
Código:		
<pre>function espar2(pnumero){ let numer = pnumero; let resultado = 0; if(number %2 == 0){ resultado = 1; } else{ resultado = 0; } return resultado; } /** fin funtion como parametro */</pre>		



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: `tablaExp = function(pnumero, pnumero2)`

Versión: 2.0

Descripción:

Imprimir la tabla del 9 con los números pares e impares

	Tipo de variable:
1 pares	Entero
2 impares	Entero
3 numero	Entero
4 contador	Numérico
5 multi	Entero
6 numero2	Entero
7 resultado	Entero

Código:

```
/** funtion como expresion */
const tablaExp= function(){
    let pares = 0;
    let impares = 0 ;
    let numero = 5
    let contador
    let multi
    let numero2 = 9
    let resultado = ""
    for(contador = 1;contador <= numero; contador ++){
        multi = numero2 * contador
        resultado += `${numero2} x ${ contador} = ${multi} ${determinarTipoExp(multi)} \n`
        pares += espar2Exp(multi);
        impares += espar2Exp(multi);
    }

    return resultado
}
```



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: `determinarTipoExp = function (resultado)`

Versión: 2.0

Descripción:

Imprimir la tabla del 9 con los números pares e impares

Tipo de variable: alfanumérico

Código:

```
const determinarTipoExp = function (resultado){  
  if(resultado %2 == 0){  
    return "pares";  
  }  
  else {  
    return " impares ";  
  }  
}
```




Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: `espar2Exp = function(pnumeros)`

Versión: 2.0

Descripción:

Imprimir la tabla del 9 con los números pares e impares

1 numer

Tipo de variable: alfanumérico

Entero

2 resultado

Entero

Código:

```
const espar2Exp = function(pnumero){
  let numer = pnumero;
  let resultado = 0;
  if(numer %2 == 0){
    resultado = 1;
  }
  else{
    resultado = 0;
  }
  return resultado;
}

/** fin funtion como expresion */
```

Consola:

function como parametro

[index.html:13](#)

9 x 1 = 9 impares

[index.html:14](#)

9 x 2 = 18 pares

9 x 3 = 27 impares

9 x 4 = 36 pares

9 x 5 = 45 impares

function como expresion

[index.html:15](#)

9 x 1 = 9 impares

[index.html:16](#)

9 x 2 = 18 pares

9 x 3 = 27 impares

9 x 4 = 36 pares

9 x 5 = 45 impares



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

25 – tabla de multiplicar del 1 al 5

Nombre de la función: tabla(ptabla)		Versión: 1.0
Descripción: Imprimir las cinco primeras tablas, sacar cantidad de pares e impares e imprimir buzz para par y bass impar		
1 contador 2 multi 3 pares 4 impares 5 resultado 6 ntabla 7 resultado2	Tipo de variable: Numérico Entero Entero Entero Entero Entero Entero	
Código: <pre>/** funtion como parametro */ function tabla(ptabla) { let contador; let multi; let pares = 0; let impares = 0; let resultado; let ntabla = ptabla let resultado2 = ""; for (contador = 1; contador <= ntabla; contador++) { for (multi = 1; multi <= ntabla; multi++) { resultado = contador * multi; resultado2 += `\${contador} x \${multi} = \${resultado} \${determinarTipo(resultado)} \n`; pares += espar2(resultado); impares += espar2(resultado) -1; } } resultado2 += `el total de numeros pares es: \${pares}\n`; resultado2 += `el total de numeros impares es: \${impares * -1}\n`; return resultado2; }</pre>		



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: determinarTipo(resultado)

Versión: 1.0

Descripción:

Imprimir las cinco primeras tablas, sacar cantidad de pares e impares e imprimir buzz para par y bass impar

Tipo de variable:

Código:

```
function determinarTipo(resultado) {  
    if (resultado % 2 == 0) {  
        return "BUZZ";  
    } else {  
        return "BASS";  
    }  
}
```

Nombre de la función: espar2(pnumero)

Versión: 1.0

Descripción:

Imprimir las cinco primeras tablas, sacar cantidad de pares e impares e imprimir buzz para par y bass impar

Tipo de variable:

1 numero

Entero

2 resultado

Entero

Código:

```
function espar2(pnumero){  
    let number = pnumero;  
    let resultado = 0  
    if(number % 2 == 0){  
        resultado = 1;  
    }  
    else{  
        resultado = 0;  
    }  
    return resultado;  
}  
  
/** fin funtion como parametro */
```



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: tablaExp = function (ptabla)

Versión: 2.0

Descripción:

Imprimir las cinco primeras tablas, sacar cantidad de pares e impares e imprimir buzz para par y bass impar

1 contador

Tipo de variable:

Numérico

2 multi

Entero

3 pares

Entero

4 impares

Entero

5 resultado

Entero

6 ntabla

Entero

7 resultado2

Entero

Código:

```
/** funtion como expresion */
const tablaExp= function(ptabla){
  let contador
  let multi
  let pares = 0
  let impares = 0
  let resultado
  let resultado2 = ""
  for(contador = 1;contador <= ptabla; contador++){
    for(multi = 1; multi <= ptabla; multi++){
      resultado = contador * multi;
      resultado2 += `${contador} x ${multi} = ${resultado} ${determinarTipoExp(resultado)} \n`;
      pares += espar2Exp(resultado);
      impares += espar2Exp(resultado) -1;
    }
  }

  resultado2 += `el total de numeros pares es: ${pares}\n`;
  resultado2 += `el total de numeros impares es: ${impares * -1}\n`;
  return resultado2;
}
```



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: `determinarTipoExp = function (resultado)`

Versión: 2.0

Descripción:

Imprimir las cinco primeras tablas, sacar cantidad de pares e impares e imprimir buzz para par y bass impar

Tipo de variable:

Código:

```
const determinarTipoExp = function(resultado) {  
  if (resultado % 2 == 0) {  
    return "BUZZ";  
  } else {  
    return "BASS";  
  }  
}
```



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: espar2Exp = function(pnumero)

Versión: 2.0

Descripción:

Imprimir las cinco primeras tablas, sacar cantidad de pares e impares e imprimir buzz para par y bass impar

1 numero
2 resultado

Tipo de variable:

Entero

Entero

Código:

```
const espar2Exp = function(pnumero){  
    let numer = pnumero;  
    let resultado = 0  
    if(numer % 2 == 0){  
        resultado = 1;  
    }  
    else{  
        resultado = 0;  
    }  
    return resultado;  
}  
  
/** fin funtion como expresion */
```

Consola:



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

function como parametros

[index.html:12](#)

```
1 x 1 = 1 BASS
1 x 2 = 2 BUZZ
1 x 3 = 3 BASS
1 x 4 = 4 BUZZ
1 x 5 = 5 BASS
2 x 1 = 2 BUZZ
2 x 2 = 4 BUZZ
2 x 3 = 6 BUZZ
2 x 4 = 8 BUZZ
2 x 5 = 10 BUZZ
3 x 1 = 3 BASS
3 x 2 = 6 BUZZ
3 x 3 = 9 BASS
3 x 4 = 12 BUZZ
3 x 5 = 15 BASS
4 x 1 = 4 BUZZ
4 x 2 = 8 BUZZ
4 x 3 = 12 BUZZ
4 x 4 = 16 BUZZ
4 x 5 = 20 BUZZ
5 x 1 = 5 BASS
5 x 2 = 10 BUZZ
5 x 3 = 15 BASS
5 x 4 = 20 BUZZ
5 x 5 = 25 BASS
el total de numeros pares es: 16
el total de numeros impares es: 9
```

[index.html:13](#)

function como expresion

[index.html:14](#)

```
1 x 1 = 1 BASS
1 x 2 = 2 BUZZ
1 x 3 = 3 BASS
1 x 4 = 4 BUZZ
1 x 5 = 5 BASS
2 x 1 = 2 BUZZ
2 x 2 = 4 BUZZ
2 x 3 = 6 BUZZ
2 x 4 = 8 BUZZ
2 x 5 = 10 BUZZ
3 x 1 = 3 BASS
3 x 2 = 6 BUZZ
3 x 3 = 9 BASS
3 x 4 = 12 BUZZ
3 x 5 = 15 BASS
4 x 1 = 4 BUZZ
4 x 2 = 8 BUZZ
4 x 3 = 12 BUZZ
4 x 4 = 16 BUZZ
4 x 5 = 20 BUZZ
5 x 1 = 5 BASS
5 x 2 = 10 BUZZ
5 x 3 = 15 BASS
5 x 4 = 20 BUZZ
5 x 5 = 25 BASS
el total de numeros pares es: 16
el total de numeros impares es: 9
```

[index.html:15](#)



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

26 – tabla de multiplicar del 1 al 5

Nombre de la función: tabla(pnumero)		Versión: 1.0
Descripción: Imprimir las cinco primeras tablas, sacar cantidad de pares e impares e imprimir buzz para par y bass impar		
1 numero 2 contador 3 multi 4 pares 5 impares 6 resultado 7 ntabla 8 resultado2	Tipo de variable: Numérico Numérico Entero Entero Entero Entero Entero Entero	
Código: <pre>/** funtion como parametro */ function tabla(pnumero){ let numero = pnumero let contador = 0 let multi = 0 let pares = 0 let impares = 0 let resultado = 0 let resultado2 = "" while(contador < numero){ contador = contador + 1 multi = 0 while(multi < numero){ multi = multi + 1 resultado = contador * multi resultado2 += `\${contador} x \${multi} = \${resultado} \${determinarTipo(resultado)} \n` pares += espar2 (resultado) impares += espar2 (resultado) + -1 } } resultado2 += `los numero pares es: \${pares}\n`; resultado2 += `los numero impares es: \${impares * -1}\n`; return resultado2; }</pre>		



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: determinarTipo(resultado)		Versión: 1.0
Descripción:		
Imprimir las cinco primeras tablas, sacar cantidad de pares e impares e imprimir buzz para par y bass impar		
	Tipo de variable:	
Código:		
<pre>function determinarTipo(resultado) { if (resultado % 2 == 0) { return "BUZZ"; } else { return "BASS"; } }</pre>		

Nombre de la función: espar2(pnumero)		Versión: 1.0
Descripción: Imprimir las cinco primeras tablas, sacar cantidad de pares e impares e imprimir buzz para par y bass impar		
1 numero	Tipo de variable: Entero	
2 resultado	Entero	
Código: <pre>function espar2(pnumero){ let number = pnumero; let resultado = 0 if(number % 2 == 0){ resultado = 1; } else{ resultado = 0; } return resultado; } /** fin funtion como parametro */</pre>		



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: tablaExp = function(pnumero)

Versión: 2.0

Descripción:

Imprimir las cinco primeras tablas, sacar cantidad de pares e impares e imprimir buzz para par y bass impar

	Tipo de variable:
1 numero	Numérico
2 contador	Numérico
3 multi	Entero
4 pares	Entero
5 impares	Entero
6 resultado	Entero
7 ntabla	Entero
8 resultado2	Entero

Código:

```
/** funtion como expresion */
const tablaExp = function (pnumero){
  let numero = pnumero
  let contador = 0
  let multi = 0
  let pares = 0
  let impares = 0
  let resultado = 0
  let resultado2 = ""

  while(contador < numero){
    contador = contador + 1
    multi = 0
    while(multi < numero){
      multi = multi + 1
      resultado = contador * multi
      resultado2 += `${contador} x ${multi} = ${resultado} ${determinarTipoExp(resultado)} \n`
      pares += espar2Exp(resultado);
      impares += espar2Exp(resultado) -1;
    }
  }

  resultado2 += `los numero pares son: ${pares}\n`
  resultado2 += `los numero pares son: ${impares * -1}\n`
  return resultado2
}
```



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: determinarTipoExp(resultado)

Versión: 2.0

Descripción:

Imprimir las cinco primeras tablas, sacar cantidad de pares e impares e imprimir buzz para par y bass impar

Tipo de variable:

Código:

```
const determinarTipoExp = function(resultado) {  
  if (resultado % 2 == 0) {  
    return "BUZZ";  
  } else {  
    return "BASS";  
  }  
}
```



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: espar2Exp = function(pnumero)

Versión: 2.0

Descripción:

Imprimir las cinco primeras tablas, sacar cantidad de pares e impares e imprimir buzz para par y bass impar

1 numero
2 resultado

Tipo de variable:
Entero
Entero

Código:

```
const espar2Exp = function(pnumero){  
  let number = pnumero;  
  let resultado = 0  
  if(number % 2 == 0){  
    resultado = 1;  
  }  
  else{  
    resultado = 0;  
  }  
  return resultado;  
}  
  
/** fin funtion como expresion */
```

Consola:



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

function como parametro

[index.html:12](#)

```
1 x 1 = 1 BASS
1 x 2 = 2 BUZZ
1 x 3 = 3 BASS
1 x 4 = 4 BUZZ
1 x 5 = 5 BASS
2 x 1 = 2 BUZZ
2 x 2 = 4 BUZZ
2 x 3 = 6 BUZZ
2 x 4 = 8 BUZZ
2 x 5 = 10 BUZZ
3 x 1 = 3 BASS
3 x 2 = 6 BUZZ
3 x 3 = 9 BASS
3 x 4 = 12 BUZZ
3 x 5 = 15 BASS
4 x 1 = 4 BUZZ
4 x 2 = 8 BUZZ
4 x 3 = 12 BUZZ
4 x 4 = 16 BUZZ
4 x 5 = 20 BUZZ
5 x 1 = 5 BASS
5 x 2 = 10 BUZZ
5 x 3 = 15 BASS
5 x 4 = 20 BUZZ
5 x 5 = 25 BASS
los numero pares es: 16
los numero impares es: 9
```

[index.html:13](#)

function como expresion

[index.html:14](#)

```
1 x 1 = 1 BASS
1 x 2 = 2 BUZZ
1 x 3 = 3 BASS
1 x 4 = 4 BUZZ
1 x 5 = 5 BASS
2 x 1 = 2 BUZZ
2 x 2 = 4 BUZZ
2 x 3 = 6 BUZZ
2 x 4 = 8 BUZZ
2 x 5 = 10 BUZZ
3 x 1 = 3 BASS
3 x 2 = 6 BUZZ
3 x 3 = 9 BASS
3 x 4 = 12 BUZZ
3 x 5 = 15 BASS
4 x 1 = 4 BUZZ
4 x 2 = 8 BUZZ
4 x 3 = 12 BUZZ
4 x 4 = 16 BUZZ
4 x 5 = 20 BUZZ
5 x 1 = 5 BASS
5 x 2 = 10 BUZZ
5 x 3 = 15 BASS
5 x 4 = 20 BUZZ
5 x 5 = 25 BASS
los numero pares son: 16
los numero pares son: 9
```

[index.html:15](#)



Tecnólogo en Análisis y Desarrollo de
Software
Ficha

Funciones JS