



Tecnólogo en Análisis y Desarrollo de
Software
Ficha 2899747

Funciones JS

TEMA: FUNCIONES

INSTRUCTOR: ANDRES MORENO COLLASOS

TECNOLOGO: ADSO

NÚMERO DE FICHA: 2899747

NOMBRE: ANGEL FARID RIVERA SUAREZ

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



Tecnólogo en Análisis y Desarrollo de
Software
Ficha 2899747

Funciones JS

Contenido

FUNCIONES NORMALES.....	3
FUNCIONES CONDICIONALES.....	20
FUNCIONES CICLOS	39



Tecnólogo en Análisis y Desarrollo de Software

Ficha 2899747

Funciones JS

FUNCIONES NORMALES

1. Imprimir "Hola mundo".

Nombre de la función: saludo(psaludar)		Versión: 1.0
Descripción: Función que saluda		
Saludar	Tipo de variable: string	
Código:	<pre>//con parametros function saludo(psaludar){ let saludar = psaludar + " Con parametros" return(saludar); }</pre>	
Prueba:	<pre>hola bienvenido usuario Con parametros</pre> ejercicio1.html:13	

Nombre de la función: saludoexp (psaludo)		Versión: 2.0
Descripción: Función que saluda		
Saludar	Tipo de variable: String	
Código:	<pre>// con expresion const saludoexp= function(psaludo){ let saludar = psaludo + " Con expresion" return(saludar); }</pre>	
Prueba:	<pre>hola bienvenido usuario Con expresion</pre> ejercicio1.html:16	



Tecnólogo en Análisis y Desarrollo de Software

Ficha 2899747

Funciones JS

2. Imprimir la suma de dos números.

Nombre de la función: sumarNum (pnum1, pnum2, psuma) num1 num1 suma		Versión: 1.0
Descripción: Función que suma dos números		
num1 num1 suma	Tipo de variable: Numérico (int) Tipo de variable: Numérico (int) Tipo de variable: Numérico (int)	
Código: <pre>//Con parametros function sumarNum(pnum1,pnum2,psuma){ let num1 = pnum1; let num2 = pnum2; let suma = psuma; suma = num1 + num2 ; return(suma); }</pre>		
Prueba:		
La suma en Parametros 1566		ejercicio2.html:18

Nombre de la función: sumarNumexp, (pnum1, pnum2) psumar		Versión: 2.0
Descripción: Función que suma dos números		
Pasumar	Tipo de variable: Numérico (int)	
Código:		
<pre>// expresion const sumarNumsexp = function(pnum1, pnum2) { pnum1=pnum1 pnum2=pnum2 let psumar = pnum1 + pnum2; psumar = psumar; return psumar; }</pre>		
Prueba:		
La suma de los números en expresión es 86 ejercicio2.html:22		



Tecnólogo en Análisis y Desarrollo de Software

Ficha 2899747

Funciones JS

3. Imprimir el resultado de la multiplicación y división de dos números.

Nombre de la función: multiplicación (panum1, panum2)		Versión: 1.0
Descripción: Función que multiplica dos números		
Resultado	Tipo de variable: Numérico (int)	
Código:	<pre>// ----parametros---- //funcion de multiplicar function multiplicacion(panum1,panum2){ panum1=panum1 panum2=panum2 let resultado = panum1 * panum2 return resultado }</pre>	

Nombre de la función: division (panum1, panum2)		Versión: 1.1
Descripción: Función que divide dos números		
Resultado	Tipo de variable: Numérico (int)	
Código:	<pre>//funcion de dividir function division(panum1,panum2){ panum1=panum1 panum2=panum2 let resultado=panum1/panum2 return resultado }</pre>	

Nombre de la función: multiYdivi (panum1, panum2, signo)		Versión: 1.3
Descripción: Función que multiplica y a la vez divide		
Función	Tipo de variable: Numérico (int)	
Código:	<pre>//funcion de ambas en parametros function multiYdivi(panum1,panum2,signo){ panum1=panum1 panum2=panum2 signo=signo let funcion if(signo== "*"){ funcion = multiplicacion(panum1,panum2) }else{ if(signo == "/"){ funcion = division(panum1,panum2) }else{ funcion = "ERROR" } } return funcion }</pre>	

Prueba:

La multiplicación en parametros es 68

[ejercicio3.html:14](#)

La división en parametros es 6

[ejercicio3.html:15](#)



Tecnólogo en Análisis y Desarrollo de Software

Ficha 2899747

Funciones JS

Nombre de la función: multiplicacionExp	Versión: 2.0
Descripción: Función que multiplica dos números	
Resultado	Tipo de variable: Numérico (int)
Código:	<pre>// ----expresion---- //funcion de multiplicar const multiplicacionExp = function(panum1,panum2){ panum1=panum1 panum2=panum2 let resultado = panum1 * panum2 return resultado }</pre>

Nombre de la función: divisionExp	Versión: 2.1
Descripción: Función que multiplica dos números	
Resultado	Tipo de variable: Numérico (int)
Código:	<pre>//funcion de dividir const divisionExp = function(panum1,panum2){ panum1=panum1 panum2=panum2 let resultado = panum1 * panum2 return resultado }</pre>

Nombre de la función: multiYdiviExp	Versión: 2.3
Descripción: Función que multiplica y a la vez divide	
Función	Tipo de variable: Numérico (int)
Código:	<pre>//funcion de ambas en expresion const multiYdiviExp = function(panum1,panum2,signo){ panum1=panum1 panum2=panum2 signo=signo let funcion if(signo=="*"){ funcion = multiplicacion(panum1,panum2) }else{ if(signo == "/"){ funcion = division(panum1,panum2) }else{ funcion = "ERROR" } } return funcion }</pre>
Prueba:	
La multiplicación en expresión es 40	ejercicio3.html:19
La división en expresión es 1.6666666666666667	ejercicio3.html:20



Tecnólogo en Análisis y Desarrollo de Software

Ficha 2899747

Funciones JS

4. Imprimir el porcentaje de un número.

Nombre de la función: porcentajeNum (pnum, ptotal)		Versión: 1.0
Descripción: Función que permite sacar el porcentaje de un determinado numero		
Resultado	Tipo de variable: Numérico (int)	
Código:	<pre>function porcentajeNum(pnum,ptotal){ pnum = pnum ptotal = ptotal let resultado resultado = (pnum * ptotal) /100 return resultado }</pre>	
Prueba:		
El 15% de 200 corresponde a 30 calculada en parametros		ejerccio4.html:13

Nombre de la función: porcentajeNumExp		Versión: 2.0
Descripción: Función que permite sacar el porcentaje de un determinado numero		
Resultado	Tipo de variable: Numérico (int)	
Código:	<pre>//expresion const porcentajeNumExp = function(pnum,ptotal){ pnum = pnum ptotal = ptotal let resultado resultado = (pnum * ptotal) /100 return resultado }</pre>	
Prueba:		
El 20% de 450 corresponde a 180 calculada en expresion		ejercicio4.html:17



Tecnólogo en Análisis y Desarrollo de Software

Ficha 2899747

Funciones JS

5. Imprimir el promedio de tres notas.

Nombre de la función: promedioTresNotas (paNota1, paNota2, paNota3)		Versión: 1.0
Descripción: Función que permite sumar las tres notas		
Suma	Tipo de variable: Numérico (float)	
Código:	<pre>//con parentros function promedioTresNotas(paNota1,paNota2,paNota3){ paNot1=paNota1 paNot2=paNota2 paNot3=paNota3 let suma = paNota1 + paNota2 + paNota3 return suma }</pre>	

Nombre de la función: dividirNotas (paNota1, paNota2, paNota3)		Versión: 1.1
Descripción: Función que permite dividir el resultado de las notas		
Nota	Tipo de variable: Numérico (float)	
Código:	<pre>function dividirNotas(paNota1, paNota2, paNota3) { let nota = promedioTresNotas(paNota1, paNota2, paNota3) / 3; return nota; }</pre>	

Prueba:

La suma de las notas son: 9.6 y el promedio es 3.6
calculada en parametros

[ejercicio5.html:13](#)



Tecnólogo en Análisis y Desarrollo de Software

Ficha 2899747

Funciones JS

Nombre de la función: promedioTresNotasExp		Versión: 2.0
Descripción: Función que permite sumar las tres notas		
Suma	Tipo de variable: Numérico (float)	
Código:		
<pre>//con expresion const promedioTresNotasExp = function(paNot1,paNot2,paNot3){ paNot1=paNot1 paNot2=paNot2 paNot3=paNot3 let suma = paNot1 + paNot2 + paNot3 return suma }</pre>		

Nombre de la función: dividirNotasExp		Versión: 2.1
Descripción: Función que permite dividir el resultado de las notas		
Nota	Tipo de variable: Numérico (float)	
Código:		
<pre>const dividirNotasExp = function(paNot1, paNot2, paNot3){ let nota = promedioTresNotas(paNot1, paNot2, paNot3) / 3; return nota; }</pre>		
Prueba:		
<div>La suma de las notas son: 11.2 y el promedio es 3.4 calculada en expresión ejercicio5.html:16</div>		



Tecnólogo en Análisis y Desarrollo de Software

Ficha 2899747

Funciones JS

6. El porcentaje de 3 notas, la nota 1= 30%, la nota 2= 30% y la nota 3=40%. Sumar las tres notas.

Nombre de la función: porcentaje1 (not1) resul1		Versión: 1.0
Descripción: Función que calcula el 30% de la nota		
resul1	Tipo de variable: Numérico (float)	
Código:	<pre>// con parametros function porcentaje1(not1){ let resul1 not1 = not1 resul1 = not1 * 0.3 return resul1 }</pre>	

Nombre de la función: porcentaje2 (not2) resul2		Versión: 1.1
Descripción: Función que calcula el 30% de la nota		
resul2	Tipo de variable: Numérico (float)	
Código:	<pre>function porcentaje2(not2){ let resul2 not2= not2 resul2 = not2* 0.3 return resul2 }</pre>	

Nombre de la función: porcentaje3(not3) resul3		Versión: 1.2
Descripción: Función que calcula el 40% de la nota		
resul3	Tipo de variable: Numérico (float)	
Código:	<pre>function porcentaje3(not3){ let resul3 not3 = not3 resul3 = not3 * 0.4 return resul3 }</pre>	



Tecnólogo en Análisis y Desarrollo de Software

Ficha 2899747

Funciones JS

Nombre de la función: sumProm (not1, not2, Suma not3)

Versión: 1.3

Descripción: Función que suma el resultado de los porcentajes

Suma

Tipo de variable: Numérico (float)

Código:

```
function sumProm(not1,not2,not3){
  not1=not1
  not2=not2
  not3=not3
  let suma
  suma = porcentaje1(not1) + porcentaje2(not2) + porcentaje3(not3)
  return suma
}
```

Prueba:

La suma de los promedios con sus porcentajes es: 4.18 [ejercicio6.html:14](#)
calculada en parametros

Nombre de la función: porcentaje1Exp

Versión: 2.0

Descripción: Función que calcula el 30% de la nota

resul1

Tipo de variable: Numérico (float)

Código:

```
// expresion
const porcentaje1Exp = function(not1){
  let resul1
  not1 = not1
  resul1 = not1 * 0.3
  return resul1
}
```

Nombre de la función: porcentaje2Exp

Versión: 2.1

Descripción: Función que calcula el 30% de la nota

resul2

Tipo de variable: Numérico (float)

Código:

```
const porcentaje2Exp = function(not2){
  let resul2
  not2= not2
  resul2 = not2* 0.3
  return resul2
}
```



Tecnólogo en Análisis y Desarrollo de Software

Ficha 2899747

Funciones JS

Nombre de la función: porcentaje3Exp

Versión: 2.2

Descripción: Función que calcula el 40% de la nota

resul3

Tipo de variable: Numérico (float)

Código:

```
const porcentaje3Exp = function(not3){  
  let resul3  
  not3 = not3  
  resul3 = not3 * 0.4  
  return resul3  
}
```

Nombre de la función: sumDePromExp (not1, not2, not3)

Versión: 2.3

Descripción: Función que suma el resultado de los porcentajes

Suma

Tipo de variable: Numérico (float)

Código:

```
const sumPromExp = function(not1,not2,not3){  
  not1=not1  
  not2=not2  
  not3=not3  
  let suma  
  suma = porcentaje1Exp(not1) + porcentaje2Exp(not2) + porcentaje3Exp(not3)  
  return suma  
}
```

Prueba:

La suma de los promedios con sus porcentajes es: 2.38 [ejercicio6.html:18](#)
calculada en expresión



Tecnólogo en Análisis y Desarrollo de Software

Ficha 2899747

Funciones JS

7. Realizar un diagrama de flujo que imprima las áreas de las siguientes figuras geométricas, el cuadrado, el rectángulo y el triángulo.

Nombre de la función: areasCuadrado(palado1) palado1, palado2, areaCuadrado		Versión: 1.0
Descripción: Función que calcula el área de un cuadrado		
palado1	Tipo de variable: Numérico (int)	
palado2	Tipo de variable: Numérico (int)	
areaCuadrado	Tipo de variable: Alfanumérico	
Código: <pre>//en parametros let palado1; //globales let palado2; //globales function areasCuadrado(palado1) { palado1 = palado1 let areaCuadrado = palado1 * palado1; areaCuadrado= "El area del cuadrado en parametros es " + areaCuadrado return(areaCuadrado) }</pre>		
Prueba: <div>Cuadrado: El area del cuadrado en parametros es 36 ejercicio7.html:12</div>		

Nombre de la función: areasRectangulo (palado1, palado2)		Versión: 1.1
Descripción: Función que calcula el área de un rectángulo		
areaRectangulo	Tipo de variable: Alfanumérico	
Código:		
<pre>function areasRectangulo(palado1,palado2){ palado1 = palado1 palado2 = palado2 let areaRectangulo = palado1 * palado2 areaRectangulo = "El area del rectangulo en parametros es " + areaRectangulo return(areaRectangulo) }</pre>		
Prueba:		
Rectangulo: El area del rectangulo en parametros es 240 ejercicio7.html:13		



Tecnólogo en Análisis y Desarrollo de Software

Ficha 2899747

Funciones JS

Nombre de la función: areasTriagulo (palado1, palado2)		Versión: 1.2
Descripción: Función que calcula el área de un triangulo		
areaTriagulo	Tipo de variable: Alfanumérico	
Código:		
<pre>function areasTriagulo(palado1,palado2){ palado1 = palado1 palado2 = palado2 let areaTriangulo = (palado1*palado2)/2 areaTriangulo = "El area del triangulo en parametros es " + areaTriangulo return(areaTriangulo) }</pre>		
Prueba:		
Triangulo: El area del triangulo en parametros es 24 ejercicio7.html:14		

Nombre de la función: saberArea (palado1, palado2, paArea)		Versión: 1.3
Descripción: Función que permite calcular cualquiera de las tres figuras geométricas (Cuadrado, Rectángulo, Triangulo).		
Área	Tipo de variable: Alfanumérico	
Código:		
<pre>//la condicional en parametros function saberArea(palado1, palado2, paArea) { palado1 = palado1; palado2 = palado2; paArea = paArea; let area; if (paArea == "Cuadrado") { area = areasCuadrado(palado1); } else if (paArea == "Rectangulo") { area = areasRectangulo(palado1, palado2); } else if (paArea == "Triangulo") { area = areasTriagulo(palado1, palado2); } else { area = "Error, esta area geometrica no se encuentra"; } return area; }</pre>		
Prueba:		
Cuadrado: El area del cuadrado en parametros es 36		ejercicio7.html:12
Rectangulo: El area del rectangulo en parametros es 240		ejercicio7.html:13
Triangulo: El area del triangulo en parametros es 24		ejercicio7.html:14
Circulo Error, esta area geometrica no se encuentra		ejercicio7.html:15



Tecnólogo en Análisis y Desarrollo de Software

Ficha 2899747

Funciones JS

Nombre de la función: areasCuadradoConst

Versión: 2.0

Descripción: Función que calcula el área de un cuadrado

areaCuadrado

Tipo de variable: Numérico (int)

Código:

```
// expresion
const areasCuadradoConst= function(palado1){
  palado1 = palado1
  let areaCuadrado = palado1 * palado1;
  return("El area del cuadrado en expresion es " + areaCuadrado)
}
```

Prueba:

Cuadrado:El area del cuadrado en expresion es 81 [ejercicio7.html:19](#)

Nombre de la función: areasRectanguloConst

Versión: 2.1

Descripción: Función que calcula el área de un rectángulo

areaRectangulo

Tipo de variable: Numérico (int)

Código:

```
const areasRectanguloConst = function(palado1,palado2){
  palado1 = palado1
  palado2 = palado2
  let areaRectangulo = palado1 * palado2
  return("El area del rectangulo en expresion es " + areaRectangulo)
}
```

Prueba:

Rectangulo: El area del rectangulo en expresion es 300 [ejercicio7.html:20](#)



Tecnólogo en Análisis y Desarrollo de Software

Ficha 2899747

Funciones JS

Nombre de la función: areasTriaguloConst

Versión: 2.2

Descripción: Función que calcula el área de un triángulo

areaTriangulo

Tipo de variable: Numérico (int)

Código:

```
const areasTriaguloConst = function(palado1,palado2){
  palado1 = palado1
  palado2 = palado2
  let areaTriangulo = (palado1*palado2)/2
  return("El area del triangulo en expresion es " + areaTriangulo)
}
```

Prueba:

Triangulo: El area del triangulo en expresion es 10 [ejercicio7.html:21](#)

Nombre de la función: saberAreasConst

Versión: 2.3

Descripción: Función que permite calcular cualquiera de las tres figuras geométricas (Cuadrado, Rectángulo, Triángulo).

Área

Tipo de variable: Alfanumérico

Código:

```
//la condicional en expresion
const saberAreasConst = function (palado1, palado2, paArea){
  palado1 = palado1;
  palado2 = palado2;
  paArea = paArea;
  let area;

  if (paArea == "Cuadrado") {
    area = areasCuadradoConst(palado1);
  } else if (paArea == "Rectangulo") {
    area = areasRectanguloConst(palado1, palado2);
  } else if (paArea == "Triangulo") {
    area = areasTriaguloConst(palado1, palado2);
  } else {
    area = "Error, esta area geometrica no se encuentra";
  }
  return area;
}
```

Prueba:

Cuadrado:El area del cuadrado en expresion es 81 [ejercicio7.html:19](#)

Rectangulo: El area del rectangulo en expresion es 300 [ejercicio7.html:20](#)

Triangulo: El area del triangulo en expresion es 10 [ejercicio7.html:21](#)

Circulo Error, esta area geometrica no se encuentra [ejercicio7.html:22](#)



Tecnólogo en Análisis y Desarrollo de Software

Ficha 2899747

Funciones JS

8. Realizar un diagrama de flujo que imprima el pago total de una persona, sabiendo que el sueldo es igual a los días trabajados por el valor del día. Imprimir la salud, pensión y arl sabiendo que la suma de la salud, la pensión y al se descuentan del sueldo de la persona.

Nombre de la función: sueldoPersona (paDias, paValorDia)		Versión: 1.0
Descripción: Función que permite calcular el sueldo de la persona.		
sueldoPersona	Tipo de variable: Numérico (int)	
Código:		
<pre>//parametros function sueldoPersona(paDias,paValorDia){ paDias = paDias paValorDia = paValorDia let sueldoPersona = paDias * paValorDia return sueldoPersona }</pre>		
Prueba:		
El sueldo de la persona calculada en parametros es <u>ejercicio8.html:14</u> 1000000		

Nombre de la función: saludPersona (paDias, paValorDia)		Versión: 1.1
Descripción: Función que permite calcular la salud de la persona.		
sueldo salud	Tipo de variable: Numérico (int) Tipo de variable: Numérico (int)	
Código: <pre>function saludPersona(paDias,paValorDia){ paDias = paDias paValorDia = paValorDia let sueldo = sueldoPersona(paDias,paValorDia) let salud = sueldo * 0.12 return salud }</pre>		
Prueba: <div>La salud de la persona calculada en parametros es 120000 ejercicio8.html:15</div>		



Tecnólogo en Análisis y Desarrollo de Software

Ficha 2899747

Funciones JS

Nombre de la función: pensionPersona (paDias, paValorDia)		Versión: 1.2
Descripción: Función que permite calcular la pensión de la persona.		
sueldo pension	Tipo de variable: Numérico (int) Tipo de variable: Numérico (int)	
Código: <pre>function pensionPersona(paDias,paValorDia){ paDias = paDias paValorDia = paValorDia let sueldo = sueldoPersona(paDias,paValorDia) let pension = sueldo * 0.16 return pension }</pre>		
Prueba: La pensión de la persona calculada en parametros es <u>ejercicio8.html:16</u> 160000		

Nombre de la función: arlPersona (paDias, paValorDia)		Versión: 1.3
Descripción: Función que permite calcular el arl de la persona.		
sueldo arl	Tipo de variable: Numérico (int) Tipo de variable: Numérico (int)	
Código:	<pre>function arlPersona(paDias,paValorDia){ paDias = paDias paValorDia = paValorDia let sueldo = sueldoPersona(paDias,paValorDia) let arl = sueldo * 0.052 return arl }</pre>	
Prueba:		
<div>El arl de la persona calculada en parametros es 52000 ejercicio8.html:17</div>		



Tecnólogo en Análisis y Desarrollo de Software

Ficha 2899747

Funciones JS

Nombre de la función: deduciblePersona (paDias, paValorDia)

Versión: 1.4

Descripción: Función que permite calcular el deducible de la persona.

deducible

Tipo de variable: Numérico (int)

Código:

```
function deduciblePersona(paDias,paValorDia){
  paDias = paDias
  paValorDia = paValorDia
  let deducible = saludPersona(paDias,paValorDia) + pensionPersona(paDias,paValorDia) + arlPersona(paDias,paValorDia)
  return deducible
}
```

Prueba:

El de deducible de la persona calculada en parametros es [ejercicio8.html:18](#)
332000

Nombre de la función: totalSueldo (paDias, paValorDia)

Versión: 1.5

Descripción: Función que permite calcular el total de la persona.

Total

Tipo de variable: Numérico (int)

Código:

```
function totalSueldo(paDias,paValorDia){
  paDias = paDias
  paValorDia = paValorDia
  let total = sueldoPersona(paDias,paValorDia) - deduciblePersona(paDias,paValorDia)
  return total
}
```

Prueba:

El total es 668000



Tecnólogo en Análisis y Desarrollo de Software

Ficha 2899747

Funciones JS

Nombre de la función: sueldoPersonaExp		Versión: 2.0
Descripción: Función que permite calcular el sueldo de la persona.		
sueldo persona	Tipo de variable: Numérico (int)	
Código:	<pre>//expresion const sueldoPersonaExp = function(paDias,paValorDia){ paDias = paDias paValorDia = paValorDia let sueldoPersona = paDias * paValorDia return sueldoPersona }</pre>	
Prueba:	<p>El sueldo de la persona calculada en expresión es 750000 ejercicio8.html:23</p>	

Nombre de la función: saludPersonaExp		Versión: 2.1
Descripción: Función que permite calcular la salud de la persona.		
sueldo salud	Tipo de variable: Numérico (int) Tipo de variable: Numérico (int)	
Código:	<pre>const saludPersonaExp = function(paDias,paValorDia){ paDias = paDias paValorDia = paValorDia let sueldo = sueldoPersonaExp(paDias,paValorDia) let salud = sueldo * 0.12 return salud }</pre>	
Prueba:	<p>La salud de la persona calculada en expresión es 90000 ejercicio8.html:24</p>	

Nombre de la función: pensionPersonaExp		Versión: 2.2
Descripción: Función que permite calcular la pensión de la persona.		
sueldo pension	Tipo de variable: Numérico (int) Tipo de variable: Numérico (int)	
Código:	<pre>const pensionPersonaExp = function(paDias,paValorDia){ paDias = paDias paValorDia = paValorDia let sueldo = sueldoPersonaExp(paDias,paValorDia) let pension = sueldo * 0.16 return pension }</pre>	
Prueba:	<p>La pensión de la persona calculada en expresión es 120000 ejercicio8.html:25</p>	



Tecnólogo en Análisis y Desarrollo de Software

Ficha 2899747

Funciones JS

Nombre de la función: arlPersonaExp

Versión: 2.3

Descripción:

Función que permite calcular el arl de la persona.

sueldo
arl

Tipo de variable: Numérico (int)
Tipo de variable: Numérico (int)

Código:

```
const arlPersonaExp = function(paDias,paValorDia){  
  paDias = paDias  
  paValorDia = paValorDia  
  let sueldo = sueldoPersonaExp(paDias,paValorDia)  
  let arl = sueldo * 0.052  
  return arl  
}
```

Prueba:

El arl de la persona calculada en expresión es 39000 [ejercicio8.html:26](#)

Nombre de la función: deduciblePersonaExp

Versión: 2.4

Descripción: Función que permite calcular el deducible de la persona.

deducible

Tipo de variable: Numérico (int)

Código:

```
const deduciblePersonaExp = function(paDias,paValorDia){  
  paDias = paDias  
  paValorDia = paValorDia  
  let deducible = saludPersonaExp(paDias,paValorDia) + pensionPersonaExp(paDias,paValorDia) + arlPersonaExp(paDias,paValorDia)  
  return deducible  
}
```

Prueba:

El de deducible de la persona calculada en expresión es [ejercicio8.html:27](#)
249000



Tecnólogo en Análisis y Desarrollo de Software

Ficha 2899747

Funciones JS

Nombre de la función: totalSueldoExp

Versión: 2.5

Descripción: Función que permite calcular el sueldo total de la persona.

Total

Tipo de variable: Numérico (int)

Código:

```
const totalSueldoExp = function(paDias,paValorDia){
  paDias = paDias
  paValorDia = paValorDia
  let total = sueldoPersonaExp(paDias,paValorDia) - deduciblePersonaExp(paDias,paValorDia)
  return total
}
```

Prueba:

El total es 501000

[ejercicio8.html:28](#)



Tecnólogo en Análisis y Desarrollo de Software

Ficha 2899747

Funciones JS

CONDICIONALES

1. Imprimir si una persona es mayor o menor de edad.

Nombre de la función: saberEdad (paEdad)		Versión: 1.0
Descripción: Función que muestra si es mayor o menor de edad		
mostrar mayorEdad	Tipo de variable: string Tipo de variable: Numérico(int)	
Código:	<pre>//parametros let mostrar function saberEdad(paEdad){ paEdad = paEdad let mayorEdad = 17 if(paEdad>mayorEdad){ mostrar = "Eres mayor de edad en parametros" }else{ mostrar = "Eres menor de edad parametros " } return(mostrar) }</pre>	
Prueba:	<div>Eres menor de edad parametros ejercicio1.html:12</div>	

Nombre de la función: saberEdadExpre		Versión: 2.0
Descripción: Función que muestra si es mayor o menor de edad		
mostrar mayorEdad	Tipo de variable: string Tipo de variable: Numérico(int)	
Código:	<pre>//expresion const saberEdadExpre = function(paEdad){ paEdad = paEdad let mayorEdad = 17 if(paEdad>mayorEdad){ mostrar = "Eres mayor de edad en expresion " }else{ mostrar = "Eres menor de edad expresion " } return(mostrar) }</pre>	
Prueba:	<div>Eres mayor de edad en expresion ejercicio1.html:15</div>	



Tecnólogo en Análisis y Desarrollo de Software

Ficha 2899747

Funciones JS

2. Imprimir si es mayor o menor de edad solo con el año en que nació.

Nombre de la función: calcularEdad (paAnioNac)		Versión: 1.0
Descripción: Función que calcula la edad de una persona y saber si es mayor o menor de edad		
anioAct calcular mostrar	Tipo de variable: Numérico(int) Tipo de variable: Numérico(int) Tipo de variable: String	
Código: <pre>//con parametros let anioAct=2024 let calcular let mostrar function calcularEdad(paAnioNac){ paAnioNac = paAnioNac calcular= anioAct - paAnioNac if(calcular>17){ mostrar= "Es mayor de edad calculando tu edad en parametros" }else{ mostrar= "Es menor de edad calculando tu edad en parametros " } return(mostrar); }</pre>		
Prueba:		
Es menor de edad calculando tu edad en parametros ejercicio2.html:13		

Nombre de la función: calcularEdadExp		Versión: 2.0
Descripción: Función que calcula la edad de una persona y saber si es mayor o menor de edad		
anioAct calcular mostrar	Tipo de variable: Numérico(int) Tipo de variable: Numérico(int) Tipo de variable: String	
Código: <pre>//con expresion const calcularEdadExp = function(paAnioNac){ paAnioNac = paAnioNac calcular= anioAct - paAnioNac if(calcular>17){ mostrar= "Es mayor de edad calculando tu edad en expresion" }else{ mostrar= "Es menor de edad calculando tu edad en expresion " } return(mostrar); }</pre>		
Prueba: Es mayor de edad calculando tu edad en expresion ejercicio2.html:17		



Tecnólogo en Análisis y Desarrollo de Software

Ficha 2899747

Funciones JS

3. Imprimir el número mayor, menor o si son iguales.

Nombre de la función: diferenciarNumeros (paNum1, paNum2)		Versión: 1.0
Descripción: Función que muestra el número mayor, menor o si son iguales		
mostrar	Tipo de variable: String	
Código:	<pre>//parametros let mostrar function diferenciarNumeros(paNum1,paNum2){ paNum1=paNum1 paNum2=paNum2 mostrar = mostrar if(paNum1==paNum2){ mostrar = "Los dos números son iguales en parametros" }else{ if(paNum1>paNum2){ mostrar = "El número 1 es mayor en parametros" }else{ mostrar = "El número 2 es mayor en parametros" } } return(mostrar) }</pre>	
Prueba:	<div>El número 2 es mayor en parametros ejercicio3.html:12</div>	

Nombre de la función: diferenciarNumerosExp		Versión: 2.0
Descripción: Función que muestra el número mayor, menor o si son iguales		
Mostrar	Tipo de variable: String	
Código:	<pre>//expresion const diferenciarNumerosExp = function(paNum1,paNum2){ paNum1=paNum1 paNum2=paNum2 if(paNum1==paNum2){ mostrar = "Los dos números son iguales en expresion" }else{ if(paNum1>paNum2){ mostrar = "El número 1 es mayor en expresion" }else{ mostrar = "El número 2 es mayor en expresion" } } return(mostrar) }</pre>	
Prueba:	<div>El número 1 es mayor en expresion ejercicio3.html:15</div>	



Tecnólogo en Análisis y Desarrollo de Software

Ficha 2899747

Funciones JS

4. Calcular el área de 3 cuadros e imprimir si las áreas son iguales y/o cual es el mayor.

Nombre de la función: areasCuadrados(palado)	Versión: 1.0
Descripción: Función que calcula el área de un cuadrado	
Área	Tipo de variable: Numérico(int)
Código: <pre>//parametros function areasCuadrados(palado) { palado=palado let area = palado * palado; return area; }</pre>	

Nombre de la función: saberCuadrados (palado1, palado2, palado3)	Versión: 1.1
Descripción: Función que calcula tres áreas y determina si son iguales o cual es el mayor	
area1 area2 area3	Tipo de variable: Numérico(int) Tipo de variable: Numérico(int) Tipo de variable: Numérico(int)
Código:	<pre>function saberCuadrados(palado1, palado2, palado3) { palado1=palado1 palado2=palado2 palado3=palado3 let area1 = areasCuadrados(palado1); let area2 = areasCuadrados(palado2); let area3 = areasCuadrados(palado3); let mostrar; if (area1 == area2 && area1 == area3) { mostrar = "Las tres áreas son iguales"; } else { if (area1 > area2 && area1 > area3) { mostrar = "El área del cuadrado 1 es mayor"; } else { if (area2 > area1 && area2 > area3) { mostrar = "El área del cuadrado 2 es mayor"; } else { mostrar = "El área del cuadrado 3 es mayor"; } } } return mostrar; }</pre>
Prueba: <div>El área del cuadrado 1 es mayor calculada en parametros ejercicio4.html:12</div>	



Tecnólogo en Análisis y Desarrollo de Software

Ficha 2899747

Funciones JS

Nombre de la función: areasCuadradosExp

Versión: 2.0

Descripción: Función que calcula el área de un cuadrado

Area

Tipo de variable: Numérico(int)

Código:

```
//expresion
const areasCuadradosExp = function(palado){
  palado=palado
  let area = palado * palado;
  return area;
}
```

Nombre de la función: saberCuadradosExp

Versión: 2.1

Descripción: Función que calcula tres áreas y determina si son iguales o cual es el mayor

area1

area2

area3

Tipo de variable: Numérico(int)

Tipo de variable: Numérico(int)

Tipo de variable: Numérico(int)

Código:

```
const saberCuadradosExp = function(palado1, palado2, palado3){
  palado1=palado1
  palado2=palado2
  palado3=palado3
  let area1 = areasCuadrados(palado1);
  let area2 = areasCuadrados(palado2);
  let area3 = areasCuadrados(palado3);
  let mostrar;

  if (area1 == area2 && area1 == area3) {
    mostrar = "Las tres áreas son iguales";
  } else {
    if (area1 > area2 && area1 > area3) {
      mostrar = "El área del cuadrado 1 es mayor";
    } else {
      if (area2 > area1 && area2 > area3) {
        mostrar = "El área del cuadrado 2 es mayor";
      } else {
        mostrar = "El área del cuadrado 3 es mayor";
      }
    }
  }
  return mostrar;
}
```

Prueba:

El área del cuadrado 3 es mayor calculada en expresión [ejercicio4.html:15](#)



Tecnólogo en Análisis y Desarrollo de Software

Ficha 2899747

Funciones JS

5. Realizar un diagrama de flujo que calcule la edad de 3 personas e imprimir si cada persona es mayor de edad, calcular e imprimir en el mismo diagrama el promedio de edades de la persona cumple con el promedio de la mayoría de edad.

Nombre de la función: MayorDeEdad(pEdad)		Versión: 1.0
Descripción: Función que calcula la edad y condiciona si esta persona cumple con la mayoría de edad		
anioAct edad mostrar	Tipo de variable: Numérico(int) Tipo de variable: Numérico(int) Tipo de variable: string	
Código: <pre>//parametros function MayorDeEdad(pEdad){ pEdad=pEdad let anioAct = 2024 let edad = anioAct - pEdad let mostrar if(edad>17){ mostrar= " cumple con la mayoria de edad en parametros" }else{ mostrar= " no cumple con la mayoria de edad en parametros" } return mostrar }</pre>		

Nombre de la función: edadesPersonas (paAnio1, paAnio2, paAnio3,) edad1, edad2, edad3, promedio, aviso		Versión: 1.1
Descripción: Función que promedia la suma de las tres edades y condiciona si este promedio cumple con la mayoría de edad		
	Tipo de variable: string Tipo de variable: string Tipo de variable: string Tipo de variable: Numérico(int) Tipo de variable: string	
Código:	<pre>//parametros function MayorDeEdad(pEdad){ pEdad=pEdad let anioAct = 2024 let edad = anioAct - pEdad let mostrar if(edad>17){ mostrar= " cumple con la mayoría de edad en parametros" }else{ mostrar= " no cumple con la mayoría de edad en parametros" } return mostrar } function edadesPersonas(paAnio1,paAnio2,paAnio3){ paAnio1=paAnio1 paAnio2=paAnio2 paAnio3=paAnio3 let edad1 = "La primera persona" let edad2 = "La segunda persona" let edad3 = "La tercera persona" let promedio let aviso console.log(edad1 + MayorDeEdad(paAnio1)) console.log(edad2 + MayorDeEdad(paAnio2)) console.log(edad3 + MayorDeEdad(paAnio3)) promedio= (MayorDeEdad(paAnio1)+MayorDeEdad(paAnio2)+edad3 + MayorDeEdad(paAnio3))/3 if(promedio>17){ aviso = "El promedio de las edades si cumple la mayoría de edad en parametros" return aviso }else{ aviso = "El promedio de las edades no cumple la mayoría de edad en parametros" return aviso } }</pre>	

Prueba:

La primera persona no cumple con la mayoría de edad en [ejercicio5.js:26](#)
parametros
La segunda persona no cumple con la mayoría de edad en [ejercicio5.js:27](#)
parametros
La tercera persona no cumple con la mayoría de edad en [ejercicio5.js:28](#)
parametros
El promedio de las edades no cumple la mayoría de edad en [ejercicio5.html:12](#)
parametros



Tecnólogo en Análisis y Desarrollo de Software

Ficha 2899747

Funciones JS

Nombre de la función: MayorDeEdadExp

Versión: 2.0

Descripción: Función que calcula la edad y condiciona si esta persona cumple con la mayoría de edad

anioAct
edad
mostrar

Tipo de variable: Numérico(int)
Tipo de variable: Numérico(int)
Tipo de variable: string

Código:

```
//expresion
const MayorDeEdadExp = function(pEdad){
  pEdad=pEdad
  let anioAct = 2024
  let edad = anioAct - pEdad
  let mostrar

  if(edad>17){
    mostrar= " cumple con la mayoria de edad en expresión"
  }else{
    mostrar= " no cumple con la mayoria de edad en expresión"
  }
  return mostrar
}
```

Nombre de la función: edadesPersonasExp, edad1, edad2, edad3, promedio, aviso

Versión: 2.1

Descripción: Función que promedia la suma de las tres edades y condiciona si este promedio cumple con la mayoría de edad

Tipo de variable: string
Tipo de variable: string
Tipo de variable: string
Tipo de variable: Numérico(int)
Tipo de variable: string

Código:

```
//expresion
const MayorDeEdadExp = function(pEdad){
  pEdad=pEdad
  let anioAct = 2024
  let edad = anioAct - pEdad
  let mostrar

  if(edad>17){
    mostrar= " cumple con la mayoria de edad en expresión"
  }else{
    mostrar= " no cumple con la mayoria de edad en expresión"
  }
  return mostrar
}

const edadesPersonasExp = function(paAño1,paAño2,paAño3){
  paAño1=paAño1
  paAño2=paAño2
  paAño3=paAño3
  let edad1 = "La primera persona"
  let edad2 = "La segunda persona"
  let edad3 = "La tercera persona"
  let promedio
  let aviso

  console.log(edad1 + MayorDeEdadExp (paAño1))
  console.log(edad2 + MayorDeEdadExp (paAño2))
  console.log(edad3 + MayorDeEdadExp (paAño3))
  promedio= ( MayorDeEdadExp (paAño1)+ MayorDeEdadExp (paAño2)+edad3 + MayorDeEdadExp (paAño3))/3
  if(promedio>17){
    aviso = "El promedio de las edades si cumple la mayoria de edad en expresión"
    return aviso
  }else{
    aviso = "El promedio de las edades no cumple la mayoria de edad en expresión"
    return aviso
  }
}
```

Activar Windows

Prueba:

La primera persona cumple con la mayoria de edad en [ejercicio5.js:61](#)
expresión
La segunda persona no cumple con la mayoria de edad en [ejercicio5.js:62](#)
expresión
La tercera persona no cumple con la mayoria de edad en [ejercicio5.js:63](#)
expresión
El promedio de las edades no cumple la mayoria de edad en [ejercicio5.html:15](#)
expresión



Tecnólogo en Análisis y Desarrollo de Software

Ficha 2899747

Funciones JS

6. Realizar un diagrama de flujo que calcule el pago total del sueldo de una persona.

Nombre de la función: salarioPersonaYtransporte(pSalario)		Versión: 1.0
Descripción: Función que permite adicionar el subsidio de transporte al sueldo de la persona y condiciona a que este sea mayor a dos salarios mínimos.		
salarioMin	Tipo de variable: Numérico(int)	
subTrans	Tipo de variable: Numérico(int)	
Código: <pre>//parametros function salarioPersonaYtransporte(paSalario){ paSalario=paSalario let salarioMin = 1400000 let subTrans=162000 if(paSalario>salarioMin*2){ paSalario=paSalario + subTrans + " Con subsidio de transporte" }else{ paSalario=paSalario+ 0 + " sin subsidio de transporte" } return paSalario }</pre>		
Prueba: Salario de la persona: 5762000 Con subsidio de transporte En parametros ejercicio6.html:12		

Nombre de la función: salud(pSalario)		Versión: 1.1
Descripción: Función que calcula el seguro de la salud de la persona a base de su sueldo		
salario salud	Tipo de variable: Numérico(float) Tipo de variable: Numérico(int)	
Código:	<pre>function salud(pSalario){ let salario = parseFloat(salarioPersonaYtransporte(pSalario)); let salud = salario * 0.12; return salud; }</pre>	
Prueba:		
Salud: 691440 En parametros		ejercicio6.html:13



Tecnólogo en Análisis y Desarrollo de Software

Ficha 2899747

Funciones JS

Nombre de la función: pension(pSalario)		Versión: 1.2
Descripción: Función que calcula la pensión de la persona a base de su sueldo		
salario pension	Tipo de variable: Numérico(float) Tipo de variable: Numérico(int)	
Código:	<pre>function pension(pSalario){ let salario = parseFloat(salarioPersonaYtransporte(pSalario)); let pension = salario * 0.16; return pension; }</pre>	
Prueba:		
Pensión: 921920 En parametros		ejercicio6.html:14

Nombre de la función: arl(pSalario)		Versión: 1.3
Descripción: Función que calcula el ARL de la persona a base de su sueldo		
salario arl	Tipo de variable: Numérico(float) Tipo de variable: Numérico(int)	
Código: <pre>function arl(pSalario){ let salario = parseFloat(salarioPersonaYtransporte(pSalario)); let arl = salario * 0.052; return arl; }</pre>		
Prueba:		
ARL:299624		ejercicio6.html:15



Tecnólogo en Análisis y Desarrollo de Software

Ficha 2899747

Funciones JS

Nombre de la función: deducible(pSalario)		Versión: 1.4
Descripción: Función que suma los tres seguros		
deducible	Tipo de variable: Numérico(int)	
<pre>function deducible(pSalario){ let deducible = salud(pSalario) + pension(pSalario) + arl(pSalario) return deducible }</pre>		Código:
Prueba:		
Deducible: 1912984 En parametros ejercicio6.html:16		

Nombre de la función: retencion(pSalario)		Versión: 1.5
Descripción: Función que retiene el 4% del salario de la persona, si el sueldo es mayor a 4 salarios mínimos		
retención salarioMin	Tipo de variable: Numérico(float) Tipo de variable: Numérico(int)	
Código:		
<pre>function retencion(pSalario){ pSalario=pSalario let retencion = parseFloat(salarioPersonaYtransporte(pSalario)) let salarioMin = 1400000 if(retencion>salarioMin*4){ retencion=retencion * 0.04 + " Con retención" return retencion }else{ retencion="Sin retención" return retencion } }</pre>		
Prueba:		
Retencion: 230480 Con retención En parametros ejercicio6.html:17		



Tecnólogo en Análisis y Desarrollo de Software

Ficha 2899747

Funciones JS

Nombre de la función: totalPagar(paSalario)

Versión: 1.6

Descripción: Función que imprime el sueldo total de la persona

salario

Tipo de variable: Numérico(float)

deducciones

Tipo de variable: Numérico(int)

total

Tipo de variable: Numérico(int)

Código:

```
function totalPagar(pSalario){  
    let salario = parseFloat(salarioPersonaYtransporte(pSalario));  
    let deducciones = deducible(pSalario);  
    let total = salario - deducciones;  
    return total;  
}
```

Prueba:

Total: 3849016 En parametros

[ejercicio6.html:18](#)



Tecnólogo en Análisis y Desarrollo de Software

Ficha 2899747

Funciones JS

Nombre de la función: salarioPersonaYtransporteExp

Versión: 2.0

Descripción: Función que permite adicionar el subsidio de transporte al sueldo de la persona y condiciona a que este sea mayor a dos salarios mínimos.

salarioMin
subTrans

Tipo de variable: Numérico(int)
Tipo de variable: Numérico(int)

Código:

```
//expresion
const salarioPersonaYtransporteExp = function(paSalario){
  paSalario=paSalario
  let salarioMin = 1400000
  let subTrans=162000
  if(paSalario>salarioMin*2){
    paSalario=paSalario + subTrans + " Con subsidio de transporte"
  }else{
    paSalario=paSalario+ 0 + " sin subsidio de transporte"
  }
  return paSalario
}
```

Prueba:

Salario de la persona: 4762000 Con subsidio de
transporte En expresión

[ejercicio6.html:22](#)

Nombre de la función: saludExp

Versión: 2.1

Descripción:

Función que calcula el seguro de la salud de la persona a base de su sueldo

salario
salud

Tipo de variable: Numérico(float)
Tipo de variable: Numérico(int)

Código:

```
const saludExp = function(paSalario){
  let salario = parseFloat(salarioPersonaYtransporte(paSalario));
  let salud = salario * 0.12;
  return salud;
}
```

Prueba:

Salud: 571440 En expresión

[ejercicio6.html:23](#)



Tecnólogo en Análisis y Desarrollo de Software

Ficha 2899747

Funciones JS

Nombre de la función: pensionExp		Versión: 2.2
Descripción: Función que calcula la pensión de la persona a base de su sueldo		
salario pension	Tipo de variable: Numérico(float) Tipo de variable: Numérico(int)	
Código:	<pre>const pensionExp = function(paSalario){ let salario = parseFloat(salarioPersonaYtransporte(paSalario)); let pension = salario * 0.16; return pension; }</pre>	
Prueba:		
Pensión: 761920 En expresión		<u>ejercicio6.html:24</u>

Nombre de la función: arlExp		Versión: 2.3
Descripción: Función que calcula el ARL de la persona a base de su sueldo		
salario arl	Tipo de variable: Numérico(float) Tipo de variable: Numérico(int)	
Código:	<pre>const arlExp =function(paSalario){ let salario = parseFloat(salarioPersonaYtransporte(paSalario)); let arl = salario * 0.052; return arl; }</pre>	
Prueba:		
ARL:247624 En expresión		<u>ejercicio6.html:25</u>



Tecnólogo en Análisis y Desarrollo de Software

Ficha 2899747

Funciones JS

Nombre de la función: deducibleExp

Versión: 2.4

Descripción: Función que suma los tres seguros

deducible

Tipo de variable: Numérico(int)

Código:

```
const deducibleExp = function(paSalario){
  let deducible = salud(paSalario) + pension(paSalario) + arl(paSalario)
  return deducible
}
```

Prueba:

Deducible: 1580984 En expresión

[ejercicio6.html:26](#)

Nombre de la función: retencionExp

Versión: 2.5

Descripción: Función que retiene el 4% del salario de la persona, si el sueldo es mayor a 4 salarios mínimos

**retención
salarioMin**

Tipo de variable: Numérico(float)
Tipo de variable: Numérico(int)

Código:

```
const retencionExp = function(paSalario){
  paSalario=paSalario
  let retencion = parseFloat(salarioPersonaYtransporte(paSalario))
  let salarioMin = 1400000
  if(retencion>salarioMin*4){
    retencion=retencion * 0.04 + " Con retención"
    return retencion
  }else{
    retencion="Sin retención"
    return retencion
  }
}
```

Prueba:

Retencion: Sin retención En expresión

[ejercicio6.html:27](#)



Tecnólogo en Análisis y Desarrollo de Software

Ficha 2899747

Funciones JS

Nombre de la función: totalPagarExp

Versión: 2.6

Descripción:

Función que imprime el sueldo total de la persona

salario
deducciones
total

Tipo de variable: Numérico(float)
Tipo de variable: Numérico(int)
Tipo de variable: Numérico(int)

Código:

```
const totalPagarExp = function(paSalario){  
  let salario = parseFloat(salarioPersonaYtransporte(paSalario));  
  let deducciones = deducible(paSalario);  
  let total = salario - deducciones;  
  return total;  
}
```

Prueba:

Total: 3181016 En En expresión

[ejercicio6.html:28](#)



Tecnólogo en Análisis y Desarrollo de Software

Ficha 2899747

Funciones JS

7. Calcular tres notas e imprimir lo siguiente, el 20% de la nota 1, el 35% de la nota 2, el 45% de la nota 3.

Nombre de la función: nota1(panot)		Versión: 1.0
Descripción: Función que calcula el 20% de la nota		
resultado	Tipo de variable: Numérico(float)	
Código:	<pre>//parametros function nota1(panot){ panot=panot let resultado=panot*0.2 return resultado }</pre>	

Nombre de la función: nota2(panot)		Versión: 1.1
Descripción: Función que calcula el 35% de la nota		
resultado	Tipo de variable: Numérico(float)	
Código:	<pre>function nota2(panot){ panot=panot let resultado=panot*0.35 return resultado }</pre>	

Nombre de la función: nota3(panot)		Versión: 1.2
Descripción: Función que calcula el 45% de la nota		
resultado	Tipo de variable: Numérico(float)	
Código:	<pre>function nota3(panot){ panot=panot let resultado=panot*0.45 return resultado }</pre>	



Tecnólogo en Análisis y Desarrollo de Software

Ficha 2899747

Funciones JS

Nombre de la función: total (panot1, panot2, panot3)

Versión: 1.3

Descripción: Función que calcula la nota final

resultado1
resultado2
resultado3
mostrar

Tipo de variable: Numérico(float)
Tipo de variable: Numérico(float)
Tipo de variable: Numérico(float)
Tipo de variable: Numérico(float)

Código:

```
function total(panot1, panot2, panot3) {  
  let resultado1 = nota1(panot1);  
  let resultado2 = nota2(panot2);  
  let resultado3 = nota3(panot3);  
  let total = resultado1 + resultado2 + resultado3;  
  let mostrar;  
  
  if (total > 4.5) {  
    mostrar = "La nota es superior en parametros";  
  } else if (total > 3.5) {  
    mostrar = "La nota es buena en parametros";  
  } else if (total >= 3.0) {  
    mostrar = "La nota es media en parametros";  
  } else {  
    mostrar = "La nota es mala en parametros";  
  }  
  return mostrar;  
}
```

Prueba:

La nota es media en parametros

[ejercicio7.html:12](#)



Tecnólogo en Análisis y Desarrollo de Software

Ficha 2899747

Funciones JS

Nombre de la función: nota1Exp		Versión: 2.0
Descripción: Función que calcula el 20% de la nota		
resultado	Tipo de variable: Numérico(float)	
Código:	<pre>//expresion const nota1Exp = function(panot){ panot=panot let resultado=panot*0.2 return resultado }</pre>	

Nombre de la función: nota2Exp		Versión: 2.1
Descripción: Función que calcula el 35% de la nota		
Resultado	Tipo de variable: Numérico(float)	
Código:	<pre>const nota2Exp = function(panot){ panot=panot let resultado=panot*0.35 return resultado }</pre>	

Nombre de la función: nota3Exp		Versión: 2.2
Descripción: Función que calcula el 45% de la nota		
Resultado	Tipo de variable: Numérico(float)	
Código:	<pre>const nota3Exp = function(panot){ panot=panot let resultado=panot*0.45 return resultado }</pre>	



Tecnólogo en Análisis y Desarrollo de Software

Ficha 2899747

Funciones JS

Nombre de la función: totalExp

Versión: 2.3

Descripción: Función que calcula la nota final

resultado1

Tipo de variable: Numérico(float)

resultado2

Tipo de variable: Numérico(float)

resultado3

Tipo de variable: Numérico(float)

mostrar

Tipo de variable: Numérico(float)

Código:

```
const totalExp = function(panot1, panot2, panot3){  
  let resultado1 = nota1Exp(panot1);  
  let resultado2 = nota2Exp(panot2);  
  let resultado3 = nota3Exp(panot3);  
  let total = resultado1 + resultado2 + resultado3;  
  let mostrar;  
  
  if (total > 4.5) {  
    mostrar = "La nota es superior en expresión";  
  } else if (total > 3.5) {  
    mostrar = "La nota es buena en expresión";  
  } else if (total >= 3.0) {  
    mostrar = "La nota es media en expresión";  
  } else {  
    mostrar = "La nota es mala en expresión";  
  }  
  
  return mostrar;  
}
```

Prueba:

La nota es superior en expresión

[ejercicio7.html:15](#)



Tecnólogo en Análisis y Desarrollo de Software

Ficha 2899747

Funciones JS

FUNCIONES CICLOS

1. Contar los números del 1 al 5

- FOR

Nombre de la función: contar1al5For (paNumero, paContar)		Versión: 1.0
Descripción: Función que permite contar del 1 al 5		
resultado	Tipo de variable: String	
Código:	<pre>// Parametros let resultado = ""; function contar1al5For(paNumero, paContar) { paNumero=paNumero; paContar=paContar; if (paNumero === 1 && paContar === 5) { for (let i = paNumero; i <= paContar; i++) { resultado += i + " "; } return resultado; } else { return "No se puede realizar el ejercicio"; } }</pre>	
Prueba:	<p>Contar numero del 1 al 5 en for con parametros 1 2 3 4 5 ejercicio1.html:12</p>	

Nombre de la función: contar1al5ForExp		Versión: 2.0
Descripción: Función que permite contar del 1 al 5		
Resultado	Tipo de variable: String	
Código:	<pre>//expresion const contar1al5ForExp= function(paNumero, paContar) { paNumero=paNumero; paContar=paContar; resultado = "" if (paNumero === 1 && paContar === 5) { for (let i = paNumero; i <= paContar; i++) { resultado += i + " "; } return resultado; } else { return "No se puede realizar el ejercicio"; } }</pre>	
Prueba:	<p>Contar numero del 1 al 5 en for con expresion 1 2 3 4 5 ejercicio1.html:15</p>	



Tecnólogo en Análisis y Desarrollo de Software

Ficha 2899747

Funciones JS

- WHILE

Nombre de la función: contar1a5		Versión: 1.0
Descripción: Función que permite contar del 1 al 5		
Resultado	Tipo de variable: String	
Código:	<pre>// parametros let resultado = ""; function contar1a5(paContar, paNumero) { paContar = paContar; paNumero = paNumero; if (paContar == 1 && paNumero == 5) { while (paContar <= paNumero) { resultado += paContar + " "; paContar++; } return resultado; } else { return "No se puede realizar el ejercicio"; } }</pre>	
Prueba:	<div>While En parametros 1 2 3 4 5 ejercicio1.html:12</div>	

Nombre de la función: contar1a5Exp		Versión: 2.0
Descripción: Función que permite contar del 1 al 5		
resultado	Tipo de variable: String	
Código:	<pre>//expresion const contar1a5Exp = function(paContar, paNumero) { paContar = paContar; paNumero = paNumero; resultado = ""; if (paContar == 1 && paNumero == 5) { while (paContar <= paNumero) { resultado += paContar + " "; paContar++; } return resultado; } else { return "No se puede realizar el ejercicio"; } };</pre>	
Prueba:	<div>While En expresion 1 2 3 4 5 ejercicio1.html:13</div>	



Tecnólogo en Análisis y Desarrollo de Software

Ficha 2899747

Funciones JS

2. Contar del 1 al 5 e imprimir los números pares e impares

- FOR

Nombre de la función: <code>pareEimparFor (paNumero, paContar)</code>	Versión: 1.0
--	---------------------

Descripción: Función que permite saber cuáles números son impares y pares.

Resultado	Tipo de variable: String
------------------	--------------------------

Código:

```
// Parámetros
let resultado = "";
function pareEimparFor(paNumero, paContar){
  paNumero = paNumero;
  paContar = paContar;

  if(paNumero === 1 && paContar === 5){
    for(let numero = paNumero ; numero <= paContar ; numero++){
      if(numero%2 === 0){
        resultado += numero + " Es par en parametros For " + "\n";
      }else{
        resultado += numero + " Es impar en parametros For " + "\n";
      }
    }
    return(resultado);
  }else{
    return "No se puede realizar el ejercicio";
  }
}
```

Prueba:

```
1 Es impar en parametros For
2 Es par en parametros For
3 Es impar en parametros For
4 Es par en parametros For
5 Es impar en parametros For
```

[ejercicio2.html:12](#)

Nombre de la función: <code>pareEimparForExp</code>	Versión: 2.0
--	---------------------

Descripción: Función que permite saber cuáles números son impares y pares.

Resultado	Tipo de variable: String
------------------	--------------------------

Código:

```
//expresion
const pareEimparForExp= function(paNumero, paContar){
  paNumero = paNumero;
  paContar = paContar;
  resultado = ""
  if(paNumero === 1 && paContar === 5){
    for(let numero = paNumero ; numero <= paContar ; numero++){
      if(numero%2 === 0){
        resultado += numero + " Es par en expresion For " + "\n";
      }else{
        resultado += numero + " Es impar en expresion For " + "\n";
      }
    }
    return(resultado);
  }else{
    return "No se puede realizar el ejercicio";
  }
}
```

Prueba:



Tecnólogo en Análisis y Desarrollo de Software

Ficha 2899747

Funciones JS

```
1 Es impar en expresion For
2 Es par en expresion  For
3 Es impar en expresion For
4 Es par en expresion  For
5 Es impar en expresion For
```

[ejercicio2.html:15](#)



Tecnólogo en Análisis y Desarrollo de Software

Ficha 2899747

Funciones JS

- WHILE

Nombre de la función: paresImpares (paNumero, paContar)		Versión: 1.0
Descripción: Función que permite saber cuáles números son impares y pares.		
Resultado	Tipo de variable: String	
Código:	<pre>// Parámetros let resultado = ""; function paresImpares(paNumero, paContar) { paNumero = paNumero; paContar = paContar; if (paNumero == 1 && paContar == 5) { while (paNumero <= paContar) { if (paNumero % 2 == 0) { resultado += paNumero + " es par en parametros " + "\n"; } else { resultado += paNumero + " es impar en parametros " + "\n"; } paNumero++; } return resultado; } else { return "No se puede realizar el ejercicio"; } }</pre>	
Prueba:	<pre>1 es impar en parametros 2 es par en parametros 3 es impar en parametros 4 es par en parametros 5 es impar en parametros</pre> <p>ejercicio2.html:12</p>	

Nombre de la función: paresImparesExp		Versión: 2.0
Descripción: Función que permite saber cuáles números son impares y pares.		
Resultado	Tipo de variable: String	
Código:	<pre>//expresion const paresImparesExp = function(paNumero, paContar){ paNumero = paNumero; paContar = paContar; resultado = ""; if (paNumero == 1 && paContar == 5) { while (paNumero <= paContar) { if (paNumero % 2 == 0) { resultado += paNumero + " es par en expresion " + "\n"; } else { resultado += paNumero + " es impar en expresion " + "\n"; } paNumero++; } return resultado; } else { return "No se puede realizar el ejercicio"; } }</pre>	
Prueba:	<pre>1 es impar en expresion 2 es par en expresion 3 es impar en expresion 4 es par en expresion 5 es impar en expresion</pre> <p>ejercicio2.html:15</p>	



Tecnólogo en Análisis y Desarrollo de Software

Ficha 2899747

Funciones JS

3. Realizar la tabla de 5 que multiplique hasta 5 y debe imprimir los siguientes resultados:
1x5=5, 2x5=10, 3x5=, 4x5=15, 5x5=25.

- FOR

Nombre de la función: tabla5For (paNumero, paMulti)		Versión: 1.0
Descripción: Función que realiza la tabla del 5		
resultado mostrar	Tipo de variable: Numérico(int) Tipo de variable: String	
Código:	<pre>//parametros let resultado let mostrar function tabla5For(paNumero,paMulti){ paNumero=paNumero paMulti=paMulti mostrar = "" if(paNumero === 1 && paMulti ===5){ for(let numero = paNumero; numero <= paMulti ; numero++){ resultado = numero * paMulti; mostrar += numero + "X" + paMulti + "=" + resultado + " En parametros\n" } }else{ mostrar = "error" } return (mostrar) }</pre>	
Prueba:	<pre>1X5=5 En parametros 2X5=10 En parametros 3X5=15 En parametros 4X5=20 En parametros 5X5=25 En parametros</pre>	

Nombre de la función: tabla5ForExp		Versión: 2.0
Descripción: Función que realiza la tabla del 5		
resultado mostrar	Tipo de variable: Numérico(int) Tipo de variable: String	
Código:	<pre>//expresion const tabla5ForExp = function(paNumero,paMulti){ paNumero=paNumero paMulti=paMulti mostrar = "" if(paNumero === 1 && paMulti ===5){ for(let numero = paNumero; numero <= paMulti ; numero++){ resultado = numero * paMulti; mostrar += numero + "X" + paMulti + "=" + resultado + " En expresion\n" } }else{ mostrar = "error" } return (mostrar) }</pre>	

```
1X5=5 En expresion
2X5=10 En expresion
3X5=15 En expresion
4X5=20 En expresion
5X5=25 En expresion
```



Tecnólogo en Análisis y Desarrollo de Software

Ficha 2899747

Funciones JS

- WHILE

Nombre de la función: tabla5(paNumero, paMultiplicar)		Versión: 1.0
Descripción: Función que realiza la tabla del 5		
resultado mostrar	Tipo de variable: Numérico(int) Tipo de variable: String	
Código:		

```
// Parámetros
let resultado;
let mostrar;
function tabla5(paNumero, paMultiplicar){
    paNumero = paNumero;
    paMultiplicar = paMultiplicar;
    mostrar = "";

    if(paNumero === 1 && paMultiplicar === 5){
        while(paNumero <= paMultiplicar){
            resultado = paNumero * paMultiplicar;
            mostrar += paNumero + "X" + paMultiplicar + "=" + resultado + " En parametros\n";
            paNumero++;
        }
    } else{
        mostrar = "Error";
    }
    return mostrar;
}
```

1X5=5 En parametros
2X5=10 En parametros
3X5=15 En parametros
4X5=20 En parametros
5X5=25 En parametros

Nombre de la función: tabla5Exp		Versión: 2.0
Descripción: Función que realiza la tabla del 5		
resultado mostrar	Tipo de variable: Numérico(int) Tipo de variable: String	
Código:	<pre>//expresion const tabla5Exp = function(paNumero, paMultiplicar){ paNumero = paNumero; paMultiplicar = paMultiplicar; mostrar = ""; if(paNumero === 1 && paMultiplicar === 5){ while(paNumero <= paMultiplicar){ resultado = paNumero * paMultiplicar; mostrar += paNumero + "X" + paMultiplicar + "=" + resultado + " En expresion\n"; paNumero++; } } else{ mostrar = "Error"; } return mostrar; }</pre> <p>1X5=5 En expresion 2X5=10 En expresion 3X5=15 En expresion 4X5=20 En expresion 5X5=25 En expresion</p>	



Tecnólogo en Análisis y Desarrollo de Software

Ficha 2899747

Funciones JS

4. Realizar la tabla del 9 que multiplique hasta 5 y de los resultados que son pares e impares. (FOR).

- FOR

Nombre de la función: tablaDel9For (paNumero, paMulti)		Versión: 1.0
Descripción: Función que realiza la tabla del 9 y determina cuales son pares e impares		
resultado mostrar	Tipo de variable: Numérico(int) Tipo de variable: String	
Código: <pre>//parametros let resultado let mostrar function tablaDel9For(paNumero,paMulti){ paNumero = paNumero; paMulti = paMulti; mostrar = ""; if(paNumero === 1 && paMulti === 9){ for(let numero = paNumero ; numero <= 5 ; numero++){ resultado = numero * paMulti if(resultado%2 === 0){ mostrar += numero + "X" + paMulti + "=" + resultado + " Es par en parametros For \n" }else{ mostrar += numero + "X" + paMulti + "=" + resultado + " Es impar en parametros For \n" } } }else{ mostrar = "Error" } return(mostrar) }</pre>		

1X9=9 Es impar en parametros For

2X9=18 Es par en parametros For

3X9=27 Es impar en parametros For

4X9=36 Es par en parametros For

5X9=45 Es impar en parametros For

1X9=9 Es impar en parametros For
2X9=18 Es par en parametros For
3X9=27 Es impar en parametros For
4X9=36 Es par en parametros For
5X9=45 Es impar en parametros For

Nombre de la función: tablaDel9ForExp		Versión: 2.0
Descripción: Función que realiza la tabla del 9 y determina cuales son pares e impares		
resultado mostrar	Tipo de variable: Numérico(int) Tipo de variable: String	
Código: <pre>//expresion const tablaDel9ForExp = function(paNumero,paMulti){ paNumero = paNumero; paMulti = paMulti; mostrar = ""; if(paNumero === 1 && paMulti === 9){ for(let numero = paNumero ; numero <= 5 ; numero++){ resultado = numero * paMulti if(resultado%2 === 0){ mostrar += numero + "X" + paMulti + "=" + resultado + " Es par en expresion For \n" }else{ mostrar += numero + "X" + paMulti + "=" + resultado + " Es impar en expresion For \n" } } }else{ mostrar = "Error" } return(mostrar) }</pre>		

1X9=9 Es impar en expresion For
2X9=18 Es par en expresion For
3X9=27 Es impar en expresion For
4X9=36 Es par en expresion For
5X9=45 Es impar en expresion For

1X9=9 Es impar en expresion For
2X9=18 Es par en expresion For
3X9=27 Es impar en expresion For
4X9=36 Es par en expresion For
5X9=45 Es impar en expresion For



Tecnólogo en Análisis y Desarrollo de Software

Ficha 2899747

Funciones JS

- WHILE

Nombre de la función: tablaDel9(paNumero, paMulti)		Versión: 1.0
Descripción: Función que realiza la tabla del 9 y determina cuales son pares e impares		
resultado mostrar	Tipo de variable: Numérico(int) Tipo de variable: String	
Código:	<pre>//parametros let resultado let mostrar function tablaDel9(paNumero,paMulti){ paNumero = paNumero; paMulti = paMulti; mostrar = ""; if(paNumero === 1 && paMulti === 9){ while(paNumero<=5){ resultado = paNumero * paMulti if(resultado%2 === 0){ mostrar += paNumero + "X" + paMulti + "=" + resultado + " Es par en parametros \n" }else{ mostrar += paNumero + "X" + paMulti + "=" + resultado + " Es impar en parametros \n" } paNumero++ } }else{ mostrar = "Error" } return(mostrar) }</pre> <p>1X9=9 Es impar en parametros 2X9=18 Es par en parametros 3X9=27 Es impar en parametros 4X9=36 Es par en parametros 5X9=45 Es impar en parametros</p>	

Nombre de la función: tablaDel9Exp		Versión: 2.0
Descripción: Función que realiza la tabla del 9 y determina cuales son pares e impares		
resultado mostrar	Tipo de variable: Numérico(int) Tipo de variable: String	
Código:	<pre>//expresion const tablaDel9Exp = function(paNumero,paMulti){ paNumero = paNumero; paMulti = paMulti; mostrar = ""; if(paNumero === 1 && paMulti === 9){ while(paNumero<=5){ resultado = paNumero * paMulti if(resultado%2 === 0){ mostrar += paNumero + "X" + paMulti + "=" + resultado + " Es par en expresion \n" }else{ mostrar += paNumero + "X" + paMulti + "=" + resultado + " Es impar en expresion \n" } paNumero++ } }else{ mostrar = "Error" } return(mostrar) }</pre> <p>1X9=9 Es impar en expresion 2X9=18 Es par en expresion 3X9=27 Es impar en expresion 4X9=36 Es par en expresion 5X9=45 Es impar en expresion</p>	



Tecnólogo en Análisis y Desarrollo de Software

Ficha 2899747

Funciones JS

5. Realizar las tablas de multiplicar de 1 hasta el 5 y multiplique hasta cinco, debe imprimir lo siguiente: 1x1..... 5x5 (FOR).

- FOR

Nombre de la función: tablaDel1a5For (paNum, paLimite)

Versión: 1.0

Descripción: Función que permite realizar la tabla del 1 hasta el 5 y determina cuantos pares e impares hay

pares
impares
mostrar
resultado
numero

Tipo de variable: Numérico(int)
Tipo de variable: Numérico(int)
Tipo de variable: String
Tipo de variable: Numérico(int)
Tipo de variable: Numérico(int)

```
let pares = 0;
let impares = 0;
let mostrar = "";
let resultado;
function tablaDel1a5For(paNum, paLimite) {
    paNum=paNum
    paLimite=paLimite

    for (let numero = paNum; numero <= paLimite; numero++) {
        for (let multiplicador = 1; multiplicador <= 5; multiplicador++) {
            resultado = numero * multiplicador;
            if (resultado % 2 === 0) {
                mostrar += numero + " X " + multiplicador + " = " + resultado + " (Buzz en parametros For) \n";
                pares++;
            } else {
                mostrar += numero + " X " + multiplicador + " = " + resultado + " (Bass en parametros For) \n";
                impares++;
            }
        }
    }

    return mostrar + "\nCantidad de pares: " + pares + "\nCantidad de impares: " + impares;
}
```

```
1 X 1 = 1 (Bass en parametros For)
1 X 2 = 2 (Buzz en parametros For)
1 X 3 = 3 (Bass en parametros For)
1 X 4 = 4 (Buzz en parametros For)
1 X 5 = 5 (Bass en parametros For)
2 X 1 = 2 (Buzz en parametros For)
2 X 2 = 4 (Buzz en parametros For)
2 X 3 = 6 (Buzz en parametros For)
2 X 4 = 8 (Buzz en parametros For)
2 X 5 = 10 (Buzz en parametros For)
3 X 1 = 3 (Bass en parametros For)
3 X 2 = 6 (Buzz en parametros For)
3 X 3 = 9 (Bass en parametros For)
3 X 4 = 12 (Buzz en parametros For)
3 X 5 = 15 (Bass en parametros For)
4 X 1 = 4 (Buzz en parametros For)
4 X 2 = 8 (Buzz en parametros For)
4 X 3 = 12 (Buzz en parametros For)
4 X 4 = 16 (Buzz en parametros For)
4 X 5 = 20 (Buzz en parametros For)
5 X 1 = 5 (Bass en parametros For)
5 X 2 = 10 (Buzz en parametros For)
5 X 3 = 15 (Bass en parametros For)
5 X 4 = 20 (Buzz en parametros For)
5 X 5 = 25 (Bass en parametros For)
```

Cantidad de pares: 16
Cantidad de impares: 9



Tecnólogo en Análisis y Desarrollo de Software

Ficha 2899747

Funciones JS

Nombre de la función: tablaDel1a5ForExp

Versión: 2.0

Descripción: Función que permite realizar la tabla del 1 hasta el 5 y determina cuantos pares e impares hay

pares
impares
mostrar
resultado
numero

Tipo de variable: Numérico(int)
Tipo de variable: Numérico(int)
Tipo de variable: String
Tipo de variable: Numérico(int)
Tipo de variable: Numérico(int)

Código:

```
//expresion
const tablaDel1a5ForExp = function(paNum, paLimite){
  paNum=paNum
  paLimite=paLimite
  let pares = 0;
  let impares = 0;
  let mostrar = "";
  let resultado;

  for (let numero = paNum; numero <= paLimite; numero++) {
    for (let multiplicador = 1; multiplicador <= 5; multiplicador++) {
      resultado = numero * multiplicador;
      if (resultado % 2 === 0) {
        mostrar += numero + " X " + multiplicador + " = " + resultado + " (Buzz en expresion For) \n";
        pares++;
      } else {
        mostrar += numero + " X " + multiplicador + " = " + resultado + " (Bass en expresion For) \n";
        impares++;
      }
    }
  }
  return mostrar + "\nCantidad de pares: " + pares + "\nCantidad de impares: " + impares;
}
```

```
1 X 5 = 5 (Bass en expresion For)
2 X 1 = 2 (Buzz en expresion For)
2 X 2 = 4 (Buzz en expresion For)
2 X 3 = 6 (Buzz en expresion For)
2 X 4 = 8 (Buzz en expresion For)
2 X 5 = 10 (Buzz en expresion For)
3 X 1 = 3 (Bass en expresion For)
3 X 2 = 6 (Buzz en expresion For)
3 X 3 = 9 (Bass en expresion For)
3 X 4 = 12 (Buzz en expresion For)
3 X 5 = 15 (Bass en expresion For)
4 X 1 = 4 (Buzz en expresion For)
4 X 2 = 8 (Buzz en expresion For)
4 X 3 = 12 (Buzz en expresion For)
4 X 4 = 16 (Buzz en expresion For)
4 X 5 = 20 (Buzz en expresion For)
5 X 1 = 5 (Bass en expresion For)
5 X 2 = 10 (Buzz en expresion For)
5 X 3 = 15 (Bass en expresion For)
5 X 4 = 20 (Buzz en expresion For)
5 X 5 = 25 (Bass en expresion For)
```

Cantidad de pares: 16
Cantidad de impares: 9



Tecnólogo en Análisis y Desarrollo de Software

Ficha 2899747

Funciones JS

- WHILE

Nombre de la función: tablaDel1a5(paTabla,paLimite)

Versión: 1.0

Descripción: Función que permite realizar la tabla del 1 hasta el 5 y determina cuantos pares e impares hay

pares
impares
mostrar
resultado

Tipo de variable: Numérico(int)
Tipo de variable: Numérico(int)
Tipo de variable: String
Tipo de variable: Numérico(int)

```
// Parámetros
let pares = 0;
let impares = 0;
let mostrar = "";
let resultado;

function tablaDel1a5(paTabla,paLimite) {
  paTabla = paTabla;
  paLimite=paLimite
  while (paTabla <= paLimite) {
    let multiplicador = 1;
    while (multiplicador <= 5) {
      resultado = paTabla * multiplicador;

      if (resultado % 2 === 0) {
        mostrar += paTabla + "X" + multiplicador + "=" + resultado + " (Buzz en parametros)\n";
        pares++;
      } else {
        mostrar += paTabla + "X" + multiplicador + "=" + resultado + " (Bass en parametros)\n";
        impares++;
      }

      multiplicador++;
    }

    paTabla++;
  }

  return mostrar + "Cantidad de pares: " + pares + "\n" + "Cantidad de impares:" + impares;
}
```

Prueba:

```
1X1=1 (Bass en parametros)
1X2=2 (Buzz en parametros)
1X3=3 (Bass en parametros)
1X4=4 (Buzz en parametros)
1X5=5 (Bass en parametros)
2X1=2 (Buzz en parametros)
2X2=4 (Buzz en parametros)
2X3=6 (Buzz en parametros)
2X4=8 (Buzz en parametros)
2X5=10 (Buzz en parametros)
3X1=3 (Bass en parametros)
3X2=6 (Buzz en parametros)
3X3=9 (Bass en parametros)
3X4=12 (Buzz en parametros)
3X5=15 (Bass en parametros)
4X1=4 (Buzz en parametros)
4X2=8 (Buzz en parametros)
4X3=12 (Buzz en parametros)
4X4=16 (Buzz en parametros)
4X5=20 (Buzz en parametros)
5X1=5 (Bass en parametros)
5X2=10 (Buzz en parametros)
5X3=15 (Bass en parametros)
5X4=20 (Buzz en parametros)
5X5=25 (Bass en parametros)
Cantidad de pares: 16
Cantidad de impares:9
```



Tecnólogo en Análisis y Desarrollo de Software

Ficha 2899747

Funciones JS

Nombre de la función: tablaDel1al5Exp

Versión: 2.0

Descripción: Función que permite realizar la tabla del 1 hasta el 5 y determina cuantos pares e impares hay

pares
impares
mostrar
resultado

Tipo de variable: Numérico(int)
Tipo de variable: Numérico(int)
Tipo de variable: String
Tipo de variable: Numérico(int)

```
//expresion
const tablaDel1al5Exp= function(paTabla, paLimite) {
  let mostrar = "";
  let pares = 0;
  let impares = 0;

  while (paTabla <= paLimite) {
    let multiplicador = 1;
    while (multiplicador <= 5) {
      let resultado = paTabla * multiplicador;

      if (resultado % 2 === 0) {
        mostrar += paTabla + "X" + multiplicador + "=" + resultado + " (Buzz en expresion)\n";
        pares++;
      } else {
        mostrar += paTabla + "X" + multiplicador + "=" + resultado + " (Bass en expresion)\n";
        impares++;
      }

      multiplicador++;
    }

    paTabla++;
  }

  return mostrar + "Cantidad de pares: " + pares + "\n" + "Cantidad de impares: " + impares;
}
```

Prueba:

```
1X4=4 (Buzz en expresion)
1X5=5 (Bass en expresion)
2X1=2 (Buzz en expresion)
2X2=4 (Buzz en expresion)
2X3=6 (Buzz en expresion)
2X4=8 (Buzz en expresion)
2X5=10 (Buzz en expresion)
3X1=3 (Bass en expresion)
3X2=6 (Buzz en expresion)
3X3=9 (Bass en expresion)
3X4=12 (Buzz en expresion)
3X5=15 (Bass en expresion)
4X1=4 (Buzz en expresion)
4X2=8 (Buzz en expresion)
4X3=12 (Buzz en expresion)
4X4=16 (Buzz en expresion)
4X5=20 (Buzz en expresion)
5X1=5 (Bass en expresion)
5X2=10 (Buzz en expresion)
5X3=15 (Bass en expresion)
5X4=20 (Buzz en expresion)
5X5=25 (Bass en expresion)
Cantidad de pares: 16
Cantidad de impares: 9
```