



Tecnólogo en Análisis y Desarrollo de
Software
Ficha

Funciones JS

ESTEBAN PALOMAR MURCIA

**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: 2.0
Descripción: Función que saluda		
saludar	Tipo de variable: Alfanumérico	
Código:	<pre>//como Parametros function saludo(psaludo){ let saludar = psaludo; return saludar; }</pre>	

Nombre de la función: const saludoExp = function(psaludo)		Versión: 3.0
Descripción: Función que saluda		
saludar	Tipo de variable: Alfanumérica	
Código:	<pre>//como expresión const saludoExp = function(psaludo){ let saludar = psaludo; return saludar; }</pre>	

CONSOLA:

```
parametros: hola mundo      index.html:13
expresión: hola mundo       index.html:14
Live reload enabled.        index.html:44
```

>



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

02 - Suma

Nombre de la función: suma(pnumUno, pnumDos)		Versión: 2.0
Descripción: Función que suma dos números		
numUno	Tipo de variable: int	
numDos	Tipo de variable: int	
sumar	Tipo de variable: int	
Código:	<pre>let numeroUno; let numeroDos; //como parametro function suma(pnumeroUno, pnumeroDos){ numUno=pnumeroUno; numDos=pnumeroDos; let sumar; sumar= numUno+numDos; return sumar }</pre>	



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: sumaExp = function(pnumUno, pnumDos)

Versión: 3.0

Descripción:

Función que suma dos números

numUno

Tipo de variable: int

numDos

Tipo de variable: int

sumar

Tipo de variable: int

Código:

```
//como expresión
const sumaExp = function(pnumeroUno, pnumeroDos){
  let sumar;
  numeroUno = pnumeroUno;
  numeroDos = pnumeroDos;
  sumar = numeroUno+numeroDos;
  return sumar;
}
```

CONSOLA

```
La suma con parametros es: 13 index.html:11
La suma con expresión es: 8 index.html:12
Live reload enabled. index.html:42
>
```



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

03 - Operaciones

Nombre de la función: function suma(pnumeroUno, pnumeroDos)		Versión: 2.0
Descripción: imprimir la suma, resta, multiplicación y división		
numUno	Tipo de variable: int	
numDos	Tipo de variable: int	
sumar	Tipo de variable: int	
Código:		
<pre>//con parametro function suma(pnumeroUno, pnumeroDos){ numeroUno=pnumeroUno; numeroDos=pnumeroDos; let sumar; sumar= numeroUno+numeroDos; return sumar; }</pre>		



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: function resta(pnumeroUno, pnumeroDos)

Versión: 2.0

Descripción:

imprimir la suma, resta, multiplicación y división

numUno

Tipo de variable: int

numDos

Tipo de variable: int

restar

Tipo de variable: int

Código:

```
function resta(pnumeroUno, pnumeroDos){  
    let restar;  
    numeroUno = pnumeroUno;  
    numeroDos = pnumeroDos;  
    restar = numeroUno-numeroDos;  
    return restar;  
}
```

Nombre de la función: function multiplicacion(pnumeroUno, pnumeroDos)

Versión: 2.0

Descripción:

imprimir la suma, resta, multiplicación y división

numUno

Tipo de variable: int

numDos

Tipo de variable: int

multiplicar

Tipo de variable: int

Código:

```
function multiplicacion(pnumeroUno, pnumeroDos){  
    numeroUno=pnumeroUno;  
    numeroDos=pnumeroDos;  
    let multiplicar;  
    multiplicar= numeroUno*numeroDos;  
    return multiplicar;  
}
```



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: function division(pnumeroUno, pnumeroDos)

Versión: 2.0

Descripción:

imprimir la suma, resta, multiplicación y división

numUno	Tipo de variable: int
numDos	Tipo de variable: int
dividir	Tipo de variable: float

Código:

```
function division(pnumeroUno, pnumeroDos){  
    numeroUno=pnumeroUno;  
    numeroDos=pnumeroDos;  
    let dividir;  
    dividir= numeroUno/numeroDos;  
    return dividir;  
}
```

Nombre de la función: function operaciones(poperador,pnumeroUno, pnumeroDos)

Versión: 2.0

Descripción:

imprimir la suma, resta, multiplicación y división

numUno	Tipo de variable: int
numDos	Tipo de variable: int
operador	Tipo de variable: string

Código:

```
function operaciones(poperador,pnumeroUno, pnumeroDos){  
    let operador = poperador;  
    numeroUno = pnumeroUno;  
    numeroDos = pnumeroDos;  
    if(operador=="suma"){  
        return suma(numeroUno,numeroDos);  
    }  
    else if(operador=="resta"){  
        return resta(numeroUno,numeroDos);  
    }  
    else if(operador=="multiplicacion"){  
        return multiplicacion(numeroUno,numeroDos);  
    }  
    else if(operador=="division"){  
        return division(numeroUno,numeroDos);  
    }  
    else{  
        return "Error!!! no reconoce operador";  
    }  
}
```



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: const sumaExp = function(pnumeroUno, pnumeroDos)	Versión: 3.0
---	---------------------

Descripción:

Función que hace operaciones básicas (sumar - restar – multiplicar – dividir)

numUno	Tipo de variable: int
numDos	Tipo de variable: int
sumarExp	Tipo de variable: int

Código:

```
//con expresión
const sumaExp = function(pnumeroUno, pnumeroDos){
  let sumar;
  numeroUno = pnumeroUno;
  numeroDos = pnumeroDos;
  sumar = numeroUno+numeroDos;
  return sumar;
}
```

Nombre de la función: const restaExp = function(pnumeroUno, pnumeroDos)	Versión: 3.0
--	---------------------

Descripción:

Función que hace operaciones básicas (sumar - restar – multiplicar – dividir)

numUno	Tipo de variable: int
numDos	Tipo de variable: int
restar	Tipo de variable: int

Código:

```
const restaExp = function(pnumeroUno, pnumeroDos){
  let restar;
  numeroUno = pnumeroUno;
  numeroDos = pnumeroDos;
  restar = numeroUno-numeroDos;
  return restar;
}
```




Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: const multiplicacionExp = function(pnumeroUno, pnumeroDos)		Versión: 3.0
Descripción: Función que hace operaciones básicas (sumar - restar – multiplicar – dividir)		
numUno	Tipo de variable: int	
numDos	Tipo de variable: int	
multiplicar	Tipo de variable: int	
Código: <pre>const multiplicacionExp = function(pnumeroUno, pnumeroDos){ let multiplicar; numeroUno = pnumeroUno; numeroDos = pnumeroDos; multiplicar = numeroUno*numeroDos; return multiplicar; }</pre>		
Nombre de la función: const divisionExp = function(pnumeroUno, pnumeroDos)		Versión: 3.0
Descripción: Función que hace operaciones básicas (sumar - restar – multiplicar – dividir)		
numUno	Tipo de variable: int	
numDos	Tipo de variable: int	
dividir	Tipo de variable: float	
Código: <pre>const divisionExp = function(pnumeroUno, pnumeroDos){ let dividir; numeroUno = pnumeroUno; numeroDos = pnumeroDos; dividir = numeroUno/numeroDos; return dividir; }</pre>		



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: const operadorExp = function(poperador,pnumeroUno, pnumeroDos) **Versión:** 3.0

Descripción:

Función que hace operaciones básicas (sumar - restar – multiplicar – dividir)

numUno	Tipo de variable: int
numDos	Tipo de variable: int
operador	Tipo de variable: sting

Código:

```
const operacionesExp = function(poperador,pnumeroUno, pnumeroDos){
  let operador = poperador;
  numeroUno = pnumeroUno;
  numeroDos = pnumeroDos;
  if(operador=="suma"){
    return sumaExp(numeroUno,numeroDos);
  }
  else if(operador=="resta"){
    return restaExp(numeroUno,numeroDos);
  }
  else if(operador=="multiplicacion"){
    return multiplicacionExp(numeroUno,numeroDos);
  }
  else if(operador=="division"){
    return divisionExp(numeroUno,numeroDos);
  }
  else{
    return "Error!!! no reconoce operador";
  }
}
```

CONSOLA

```
CON PARAMETROS                                     index.html:11
                                                    index.html:12
suma con parametros: 8
resta con parametros: 10                           index.html:13
Multiplicación con parametros: 15                  index.html:14
División con parametros: 5                         index.html:15
                                                    index.html:16
CON EXPRESIÓN                                       index.html:17
suma con Expresión es: 107
resta con Expresión es: 32                         index.html:18
Multiplicación con Expresión es: 979               index.html:19
División con Expresión es: 9                      index.html:20
Live reload enabled.                               index.html:51
```

> |



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

04 - Porcentaje

Nombre de la función: function porNumero(pNumero)		Versión: 2.0
Descripción: Halla el porcentaje de un número		
numero	Tipo de variable: float	
rnumero	Tipo de variable: float	
Código:		
<pre>//con parametro function porNumero(pnumero){ let numero = pnumero; let rnumero; rnumero= numero/100; return rnumero; }</pre>		

Nombre de la función: const porNumeroExp = function(pnumero)		Versión: 3.0
Descripción: Halla el porcentaje de un número		
Numero	Tipo de variable: float	
rnumero	Tipo de variable: float	
Código:		
<pre>//con expresión const porNumeroExp = function(pnumero){ let numero = pnumero; let rnumero; rnumero= numero/100; return rnumero; }</pre>		



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

CONSOLE:

```
CON PARAMETROS index.html:15  
el porcentaje del numero es: 0.69% index.html:16  
index.html:17  
CON EXPRESIÓN  
El porcentaje del numero con expresión es: 0.11% index.html:18  
Live reload enabled. index.html:48
```

>



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

05 – Promedio de Notas

Nombre de la función: function proNota(pnota1, pnota2, pnota3)		Versión: 2.0
Descripción: Imprime el promedio de tres notas		
nota1	Tipo de variable: float	
nota2	Tipo de variable: float	
nota3	Tipo de variable: float	
promedio	Tipo de variable: float	
Código:		
<pre>//con parametro function proNota(pnota1, pnota2, pnota3){ let nota1 = pnota1; let nota2 = pnota2; let nota3 = pnota3; let promedio; promedio= (nota1+nota2+nota3)/3; return promedio; }</pre>		

Nombre de la función: const proNotaExp = function(pnota1, pnota2, pnota3)		Versión: 3.0
Descripción: Imprime el promedio de tres notas		
Promedio	Tipo de variable: float	
nota1	Tipo de variable: float	
nota2	Tipo de variable: float	
nota3	Tipo de variable: float	
Código:		
<pre>//con expresión const proNotaExp = function(pnota1, pnota2, pnota3){ let nota1 = pnota1; let nota2 = pnota2; let nota3 = pnota3; let promedio; promedio= (nota1+nota2+nota3)/3; return promedio; }</pre>		



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

CONSOLA

CON PARAMETROS [index.html:12](#)

[index.html:13](#)
El promedio de las 3 notas con parametros es: 3.3333333333333335

[index.html:14](#)
CON EXPRESIÓN

El promedio de las 3 notas con expresión es: 3.1 [index.html:15](#)

Live reload enabled. [index.html:45](#)

> |



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

06 – Promedio de Notas 2

Nombre de la función: function porNota(pnota, pporce)		Versión: 2.0
Descripción: Calcular el porcentaje ponderado de tres notas. Nota 1 = 30% Nota 2 = 30% Nota 3 = 40%		
nota	Tipo de variable: float	
porce	Tipo de variable: float	
rnota	Tipo de variable: float	
Código:		
<pre>//con parametro function porNota(pnota, pporce){ let nota = pnota; let porce = pporce; let rnota; rnota= nota*porce; return rnota; }</pre>		

Nombre de la función: const porNotaExp = function(pnota, pporce)		Versión: 3.0
Descripción: Calcular el porcentaje ponderado de tres notas. Nota 1 = 30% Nota 2 = 30% Nota 3 = 40%		
Proce	Tipo de variable: int	
Nota	Tipo de variable: float	
rnota	Tipo de variable: float	
Código:		
<pre>//con expresión const porNotaExp = function(pnota, pporce){ let nota = pnota; let porce = pporce; let rnota; rnota= nota*porce; return rnota; }</pre>		



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

CONSOLE:

CON PARAMETROS [index.html:12](#)

[index.html:13](#)

El promedio de las 3 notas con parametros es: 3.3333333333333335

[index.html:14](#)

CON EXPRESIÓN

El promedio de las 3 notas con expresión es: 3.1 [index.html:15](#)

Live reload enabled. [index.html:45](#)

>



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

07 – Área de Figuras Geométricas

Nombre de la función: function areaCua(plado1,plado2)		Versión: 2.0
Descripción: Calcula el área de un cuadrado, rectángulo y triángulo		
lado1	Tipo de variable: int	
lado2	Tipo de variable: int	
area	Tipo de variable: int	
Código: <pre>//con parametro function AreaCua(plado1,plado2){ let lado1= plado1; let lado2= plado2; let area; area= lado1*lado2; return area; }</pre>		
Nombre de la función: function areaTri(pbase,paltura)		Versión: 2.0
Descripción: Calcula el área de un cuadrado, rectángulo y triángulo		
base	Tipo de variable: int	
altura	Tipo de variable: int	
area	Tipo de variable: float	
Código: <pre>function AreaTri(pbase,paltura){ let base= pbase; let altura= paltura; let area; area= (base*altura)/2; return area; }</pre>		



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: `function areaRec(pbase,paltura)`

Versión: 2.0

Descripción:

Calcula el área de un cuadrado, rectángulo y triángulo

base

Tipo de variable: int

altura

Tipo de variable: int

area

Tipo de variable: int

Código:

```
function AreaRec(pbase,paltura){  
    let base= pbase;  
    let altura= paltura;  
    let area;  
    area= base*altura;  
    return area;  
}
```

Nombre de la función: `const areaCuaExp = function(plado1,plado2)`

Versión: 3.0

Descripción:

Calcula el área de un cuadrado, rectángulo y triángulo

lado1

Tipo de variable: int

lado2

Tipo de variable: int

area

Tipo de variable: int

Código:

```
//con expresión  
const AreaCuaExp = function(plado1,plado2){  
    let lado1= plado1;  
    let lado2= plado2;  
    let area;  
    area= lado1*lado2;  
    return area;  
}
```



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: const <u>areaTriExp</u> = function(pbase,paltura)		Versión: 3.0
Descripción: Calcula el área de un cuadrado, rectángulo y triángulo		
base	Tipo de variable: int	
altura	Tipo de variable: int	
area	Tipo de variable: float	
Código: <pre>const AreaTriExp = function(pbase,paltura){ let base= pbase; let altura= paltura; let area; area= (base*altura)/2; return area; }</pre>		
Nombre de la función: const <u>areaRecExp</u> = function(pbase,paltura)		Versión: 3.0
Descripción: Calcula el área de un cuadrado, rectángulo y triángulo		
base	Tipo de variable: int	
altura	Tipo de variable: int	
area	Tipo de variable: int	
Código: <pre>const AreaRecExp = function(pbase,paltura){ let base= pbase; let altura= paltura; let area; area= base*altura; return area; }</pre>		



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

CONSOLA:

```
CON PARAMETROS                                index.html:12  
el area del cuadrado con parametro es: 16      index.html:13  
El area del triangulo con parametro es: 100    index.html:14  
elarea del rectangulo con parametro es: 450    index.html:15  
                                                index.html:16  
CON EXPRESIÓN  
el area del cuadrado con expresión es: 256     index.html:17  
El area del triangulo con expresión es: 1350   index.html:18  
elarea del rectangulo con expresión es: 45     index.html:19  
Live reload enabled.                           index.html:49
```

> |



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

08 – Sueldo de Persona

Nombre de la función: function sueldo(pdiasT,pvalorD)		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)		
diasT	Tipo de variable: int	
valorD	Tipo de variable: int	
sueldo	Tipo de variable: int	
Código:		
<pre>//con parametro function sueldo(pdiasT,pvalorD){ let diasT = pdiasT; let valorD = pvalorD; let sueldo; sueldo = valorD*diasT; return sueldo; }</pre>		
Nombre de la función: function pension(pdiasT,pvalorD)		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)		
pension	Tipo de variable: int	
Código:		
<pre>function pension(pdiasT,pvalorD){ let pension; pension = sueldo(pdiasT,pvalorD)*0.16 return pension }</pre>		



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: function salud(pdiasT,pvalorD)

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)

salud

Tipo de variable: int

Código:

```
function salud(pdiasT,pvalorD){  
    let salud;  
    salud= sueldo(pdiasT,pvalorD)*0.12;  
    return salud  
}
```

Nombre de la función: function arl(pdiasT,pvalorD)

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)

arl

Tipo de variable: int

Código:

```
function arl(pdiasT,pvalorD){  
    let arl;  
    arl = sueldo(pdiasT,pvalorD)*0.052  
    return arl  
}
```



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: function sueldoTotal(pdiasT,pvalorD)

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)

sueldoInicial	Tipo de variable: int
pensionTotal	Tipo de variable: int
saludTotal	Tipo de variable: int
arlTotal	Tipo de variable: int
deducibles	Tipo de variable: int
sueldoFinal	Tipo de variable: int

Código:

```
function sueldoTotal(pdiasT,pvalorD){  
    let sueldoInicial = sueldo(pdiasT,pvalorD);  
    let pensionTotal = pension(pdiasT,pvalorD);  
    let arlTotal = arl(pdiasT,pvalorD);  
    let saludTotal = salud(pdiasT,pvalorD);  
    let deducibles = pensionTotal+arlTotal+saludTotal;  
    let sueldoFinal = sueldoInicial-deducibles;  
  
    return sueldoFinal;  
}
```



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: const sueldoExp = function(pdiasT,pvalorD)		Versión: 3.0
Descripción: Calcular el sueldo de un trabajador con deducciones de pensión, salud y ARL. (pensión = 0.16, salud = 0.12, ARL = 0.052)		
diasT	Tipo de variable: int	
valorD	Tipo de variable: int	
sueldo	Tipo de variable: int	
Código:		
<pre>//con Expresión const sueldoExp= function(pdiasT,pvalorD){ let diasT = pdiasT; let valorD = pvalorD; let sueldo; sueldo = valorD*diasT; return sueldo; }</pre>		

Nombre de la función: const pensionExp = function(pdiasT,pvalorD)		Versión: 3.0
Descripción: Calcular el sueldo de un trabajador con deducciones de pensión, salud y ARL. (pensión = 0.16, salud = 0.12, ARL = 0.052)		
pension	Tipo de variable: int	
Código:		

```
const pensionExp = function(pdiasT,pvalorD){
  let pension;
  pension = sueldo(pdiasT,pvalorD)*0.16
  return pension
}
```




Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: `const saludExp = function(pdiasT,pvalorD)`

Versión: 3.0

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

salud

Tipo de variable: int

Código:

```
const saludExp = function(pdiasT,pvalorD){  
    let salud;  
    salud= sueldo(pdiasT,pvalorD)*0.12;  
    return salud  
}
```

Nombre de la función: `const arlExp = function(pdiasT,pvalorD)`

Versión: 3.0

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

arl

Tipo de variable: int

Código:

```
const arlExp = function(pdiasT,pvalorD){  
    let arl;  
    arl = sueldo(pdiasT,pvalorD)*0.052  
    return arl  
}
```



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: const sueldoTotalExp = function(pdiasT,pvalorD)		Versión: 3.0
Descripción: Calcular el sueldo de un trabajador con deducciones de pensión, salud y ARL. (pensión = 0.16, salud = 0.12, ARL = 0.052)		
sueldoInicial	Tipo de variable: int	
pensionTotal	Tipo de variable: int	
saludTotal	Tipo de variable: int	
arlTotal	Tipo de variable: int	
deducibles	Tipo de variable: int	
sueldoFinal	Tipo de variable: int	
Código:		
<pre>const sueldoTotalExp = function(pdiasT,pvalorD){ let sueldoInicial = sueldo(pdiasT,pvalorD); let pensionTotal = pension(pdiasT,pvalorD); let arlTotal = arl(pdiasT,pvalorD); let saludTotal = salud(pdiasT,pvalorD); let deducibles = pensionTotal+arlTotal+saludTotal; let sueldoFinal = sueldoInicial-deducibles; return sueldoFinal; }</pre>		
CONSOLA:		
<div>CON PARAMETROS</div> <div><div>El sueldo inicial de la persona es: 4800000</div><div>Se le descuenta la pensión que es: 768000</div><div>Se le descuenta la salud que es: 576000</div><div>Se le descuenta el ARL que es: 249600</div><div>El sueldo total de la persona termina siendo de: 3206400</div></div> <div>CON EXPRESIÓN</div> <div><div>El sueldo inicial de la persona es: 1800000</div><div>Se le descuenta la pensión que es: 288000</div><div>Se le descuenta la salud que es: 216000</div><div>Se le descuenta el ARL que es: 93600</div><div>El sueldo total de la persona termina siendo de: 1202400</div></div>		



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

CONDICIONALES

09 – Numero Mayor

Nombre de la función: function mayorQue(pnumero1, pnumero2)		Versión: 2.0
Descripción: Imprimir el mayor de dos números		
numero1	Tipo de variable: int	
numero2	Tipo de variable: int	
Código:		
<pre>//con parametro function mayorQue(pnumero1,pnumero2){ let numero1=pnumero1; let numero2=pnumero2; let numeroMayor; if(numero1>numero2){ numeroMayor=numero1; return numeroMayor; }else{ numeroMayor=numero2; return numeroMayor; } }</pre>		

Nombre de la función: const mayorQueExp = function(pnumero1,numero2)		Versión: 3.0
Descripción: Imprimir el mayor de dos números		
numero1	Tipo de variable: int	
numero2	Tipo de variable: int	
Código:		
<pre>//con expresión const mayorQueExp = function(pnumero1,pnumero2){ let numero1=pnumero1; let numero2=pnumero2; let numeroMayor; if(numero1>numero2){ numeroMayor=numero1; return numeroMayor; }else{ numeroMayor=numero2; return numeroMayor; } }</pre>		
CONSOLE:		
<div>CON PARAMETROS index.html:12</div> <div>El numero mayor de los dos es: 22 index.html:13</div> <div>CON EXPRESIÓN index.html:14</div> <div>El numero mayor de los dos es: 66 index.html:15</div> <div>Live reload enabled. index.html:45</div> <div>></div>		



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

10 – Calcular Edad

Nombre de la función: function calcularEdad(pnacimiento)		Versión: 2.0
Descripción: Calcular la edad de una persona e imprimir si es mayor de edad		
Anacimiento edad	Tipo de variable: Alfanumérico Tipo de variable: Alfanumérico	
Código: <pre>//con parametro function calcularEdad(pnacimiento){ let Anacimiento=pnacimiento; let edad; edad= 2024-Anacimiento; if(edad>17){ return "Mayor de edad con "+edad; }else{ return "Menor de edad con "+edad; } }</pre>		

Nombre de la función: const calcularEdadExp = function(pnacimiento)		Versión: 3.0
Descripción: Calcular la edad de una persona e imprimir si es mayor de edad		
Anacimiento edad	Tipo de variable: Alfanumérica Tipo de variable: Alfanumérico	
Código: <pre>//con expresión const calcularEdadExp = function(pnacimiento){ let Anacimiento=pnacimiento; let edad; edad= 2024-Anacimiento; if(edad>17){ return "Mayor de edad con "+edad; }else{ return "Menor de edad con "+edad; } }</pre>		
CONSOLE: <div><div>CON PARAMETROSindex.html:12</div><div>La edad calculada de la persona nos dice que es: Mayor de edad con 19index.html:13</div><div>CON EXPRESIÓNindex.html:14</div><div>La edad calculada de la persona nos dice que es: Menor de edad con 14index.html:15</div><div>Live reload enabled.index.html:46</div><div>> </div></div>		



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

11 – Mayor de Dos Números

Nombre de la función: function mayorQue(pnumero1, pnumero2)		Versión: 2.0
Descripción: Imprime el mayor de dos números validando que no sean iguales		
numero1	Tipo de variable: int	
numero2	Tipo de variable: int	
numeroMayor	Tipo de variable: int	
Código:		
<pre>//con parametro function mayorQue(pnumero1,pnumero2){ let numero1=pnumero1; let numero2=pnumero2; let numeroMayor; if(numero1==numero2){ return "Los numeros "+numero1+" y "+numero2+" son iguales" }else{ if(numero1>numero2){ numeroMayor=numero1; return numeroMayor+" es el numero mayor"; }else{ numeroMayor=numero2; return numeroMayor+" es el numero mayor"; } } }</pre>		



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: `const mayorQueExp = function(pNum1, pNum2)`

Versión: 3.0

Descripción:

Imprime el mayor de dos números validando que no sean iguales

numero1

Tipo de variable: int

numero2

Tipo de variable: int

numeroMayor

Tipo de variable: int

Código:

```
//con expresión
const mayorQueExp = function(pnumero1,pnumero2){
  let numero1=pnumero1;
  let numero2=pnumero2;
  let numeroMayor;
  if(numero1==numero2){
    return "Los numeros "+numero1+" y "+numero2+" son iguales"
  }else{
    if(numero1>numero2){
      numeroMayor=numero1;
      return numeroMayor+" es el numero mayor";
    }else{
      numeroMayor=numero2;
      return numeroMayor+" es el numero mayor";
    }
  }
}
```

CONSOLA:

CON PARAMETROS

[index.html:12](#)

Los numeros 11 y 11 son iguales

[index.html:13](#)

CON EXPRESIÓN

[index.html:14](#)

16 es el numero mayor

[index.html:15](#)

Live reload enabled.

[index.html:45](#)

>



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

12- Mayor de Tres Números

Nombre de la función: function mayorQue(pnumero1,pnumero2,pnumero3){		Versión: 2.0
Descripción: Imprime el mayor de tres números con la misma validación		
numero1	Tipo de variable: int	
numero2	Tipo de variable: int	
numero3	Tipo de variable: int	
numeroMayor	Tipo de variable: int	
Código:		
<pre>//con parametro function mayorQue(pnumero1,pnumero2,pnumero3){ let numero1=pnumero1; let numero2=pnumero2; let numero3=pnumero3; let numeroMayor; if(numero1>numero2&numero1>numero3){ numeroMayor=numero1; return numeroMayor+" es el numero mayor"; } else{ if(numero2>numero1&numero2>numero3){ numeroMayor=numero2; return numeroMayor+" es el numero mayor"; } else{ numeroMayor=numero3; return numeroMayor+" es el numero mayor"; } } }</pre>		



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: <code>const mayorQueExp = function(pnumero1,pnumero2,pnumero3)</code>	Versión: 3.0
--	---------------------

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

numero1	Tipo de variable: int
numero2	Tipo de variable: int
numero3	Tipo de variable: int
numeroMayor	Tipo de variable: int

Código:

```
//con expresión
const mayorQueExp = function(pnumero1,pnumero2,pnumero3){
  let numero1=pnumero1;
  let numero2=pnumero2;
  let numero3=pnumero3;
  let numeroMayor;
  if(numero1>numero2&numero1>numero3){
    numeroMayor=numero1;
    return numeroMayor+" es el numero mayor";
  }
  else{
    if(numero2>numero1&numero2>numero3){
      numeroMayor=numero2;
      return numeroMayor+" es el numero mayor";
    }
    else{
      numeroMayor=numero3;
      return numeroMayor+" es el numero mayor";
    }
  }
}
```

CONSOLA:

```
CON PARAMETROS                                     index.html:12
66 es el numero mayor                               index.html:13
                                                    index.html:14
CON EXPRESIÓN
4 es el numero mayor                                index.html:15
Live reload enabled.                                index.html:45
> |
```




Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

13 - Área de Tres Cuadrados

Nombre de la función: function areaCua(pladoCua1,pladoCua2,pladoCua3)		Versión: 2.0
Descripción: Calcular el área 3 de cuadrados e imprimir el mayor		
ladoCuadrado1	Tipo de variable: int	
ladoCuadrado2	Tipo de variable: int	
ladoCuadrado3	Tipo de variable: int	
areaCua1	Tipo de variable: int	
areaCua2	Tipo de variable: int	
areaCua3	Tipo de variable: int	
Código:		
<pre>//con parametro function areaCua(pladoCua1,pladoCua2,pladoCua3){ let ladoCuadrado1= pladoCua1; let ladoCuadrado2= pladoCua2; let ladoCuadrado3= pladoCua3; let areaCua1; let areaCua2; let areaCua3; let areaMayor; areaCua1= ladoCuadrado1*ladoCuadrado1; areaCua2= ladoCuadrado2*ladoCuadrado2; areaCua3= ladoCuadrado3*ladoCuadrado3; if(areaCua1>areaCua2&areaCua1>areaCua3){ areaMayor=areaCua1; return areaMayor+" es el area mayor"; } else{ if(areaCua2>areaCua1&areaCua2>areaCua3){ areaMayor=areaCua2; return areaMayor+" es el area mayor"; } else{ areaMayor=areaCua3; return areaMayor+" es el area mayor"; } } }</pre>		



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: `const areaCuaExp = function(pladoCua1,pladoCua2,pladocua3)` Versión: 3.0

Descripción:

Calcular el área 3 de cuadrados e imprimir el mayor.

ladoCuadrado1	Tipo de variable: int
ladoCuadrado2	Tipo de variable: int
ladoCuadrado3	Tipo de variable: int
areaCua1	Tipo de variable: int
areaCua2	Tipo de variable: int
areaCua3	Tipo de variable: in

Código:

```
//con expresión
const areaCuaExp = function(pladoCua1,pladoCua2,pladoCua3){
  let ladoCuadrado1= pladoCua1;
  let ladoCuadrado2= pladoCua2;
  let ladoCuadrado3= pladoCua3;
  let areaCua1;
  let areaCua2;
  let areaCua3;
  let areaMayor;
  areaCua1= ladoCuadrado1*ladoCuadrado1;
  areaCua2= ladoCuadrado2*ladoCuadrado2;
  areaCua3= ladoCuadrado3*ladoCuadrado3;
  if(areaCua1>areaCua2&areaCua1>areaCua3){
    areaMayor=areaCua1;
    return areaMayor+" es el area mayor";
  }
  else{
    if(areaCua2>areaCua1&areaCua2>areaCua3){
      areaMayor=areaCua2;
      return areaMayor+" es el area mayor";
    }
    else{
      areaMayor=areaCua3;
      return areaMayor+" es el area mayor";
    }
  }
}
```

CONSOLE:

CON PARAMETROS	index.html:12
625 es el area mayor	index.html:13
CON EXPRESIÓN	index.html:14
196 es el area mayor	index.html:15
Live reload enabled.	index.html:45
>	



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

14 – Mayor de Tres Edades

Nombre de la función: function calcularEdad1(pnacimiento)

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

Anacimiento
edaduno

Tipo de variable: int
Tipo de variable: int

Código:

```
function calcularEdad1(pnacimiento){  
    let Anacimiento=pnacimiento;  
    edadUno= 2024-Anacimiento;  
    if(edadUno>17){  
        return "La primera persona Es Mayor de edad con "+edadUno;  
    }else{  
        return "La primera persona Es Menor de edad con "+edadUno;  
    }  
}
```

Nombre de la función: function calcularEdad2(pnacimiento)

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

Anacimiento
edadDos

Tipo de variable: int
Tipo de variable: int

Código:

```
function calcularEdad2(pnacimiento){  
    let Anacimiento=pnacimiento;  
    edadDos= 2024-Anacimiento;  
    if(edadDos>17){  
        return "La segunda persona Es Mayor de edad con "+edadDos;  
    }else{  
        return "La segunda persona Es Menor de edad con "+edadDos;  
    }  
}
```



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: function calcularEdad3(pnacimiento)		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		
Anacimiento	Tipo de variable: int	
edadTres	Tipo de variable: int	
Código:		
<pre>function calcularEdad3(pnacimiento){ let Anacimiento=pnacimiento; edadTres= 2024-Anacimiento; if(edadTres>17){ return "La tercera persona Es Mayor de edad con "+edadTres; }else{ return "La tercera persona Es Menor de edad con "+edadTres; } }</pre>		

Nombre de la función: function promedioEdad()		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		
promedio	Tipo de variable: Float	
Código:		
<pre>function promedioEdad(){ let promedio = (edadUno+edadDos+edadTres)/3; if(promedio>17){ return "El promedio de las edades cumplen la mayoria de edad con: "+promedio+" años"; }else{ return "El promedio de las edades no cumplen con la mayoria de edad con: "+promedio+" años"; } }</pre>		



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: const calcularEdad1Exp = Function(pnacimiento)		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		
Anacimiento	Tipo de variable: int	
edadUno	Tipo de variable: int	
Código:		
<pre>//con expresión const calcularEdad1Exp = function(pnacimiento){ let Anacimiento=pnacimiento; edadUno= 2024-Anacimiento; if(edadUno>17){ return "La primera persona Es Mayor de edad con "+edadUno; }else{ return "La primera persona Es Menor de edad con "+edadUno; } }</pre>		

Nombre de la función: const calcularEdad2Exp = Function(pnacimiento)		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		
Anacimiento	Tipo de variable: int	
edadDos	Tipo de variable: int	
Código:		
<pre>const calcularEdad2Exp = function(pnacimiento){ let Anacimiento=pnacimiento; edadDos= 2024-Anacimiento; if(edadDos>17){ return "La segunda persona Es Mayor de edad con "+edadDos; }else{ return "La segunda persona Es Menor de edad con "+edadDos; } }</pre>		



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: const calcularEdad3Exp = Function(pnacimiento)

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

Anacimiento
edadTres

Tipo de variable: int
Tipo de variable: int

Código:

```
const calcularEdad3Exp = function(pnacimiento){
  let Anacimiento=pnacimiento;
  edadTres= 2024-Anacimiento;
  if(edadTres>17){
    return "La tercera persona Es Mayor de edad con "+edadTres;
  }else{
    return "La tercera persona Es Menor de edad con "+edadTres;
  }
}
```

Nombre de la función: const promedioEdadExp = Function()

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

promedio

Tipo de variable: Float

Código:

```
const promedioEdadExp = function(){
  let promedio = (edadUno+edadDos+edadTres)/3;
  if(promedio>17){
    return "El promedio de las edades cumplen la mayoría de edad con: "+promedio+" años";
  }else{
    return "El promedio de las edades no cumplen con la mayoría de edad con: "+promedio+" años";
  }
}
```

CONSOLE:

```
CON PARAMETROS                                index.html:14
La primera persona Es Mayor de edad con 22 años  index.html:15
La segunda persona Es Menor de edad con 16 años  index.html:16
La tercera persona Es Menor de edad con 9 años   index.html:17
El promedio de las edades no cumplen con la mayoría de edad con: 15.666666666666666 años index.html:18
index.html:19
CON EXPRESIÓN
La primera persona Es Mayor de edad con 100 años index.html:21
La segunda persona Es Mayor de edad con 56 años  index.html:22
La tercera persona Es Mayor de edad con 26 años   index.html:23
El promedio de las edades cumplen la mayoría de edad con: 60.666666666666664 años  index.html:24
Live reload enabled.                             index.html:54
>
```



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

15 – Sueldo Persona 2

Nombre de la función: function sueldo(pdiasT,pvalorD)		Versión: 2.0
Descripción: Calcular el sueldo de una persona todo incluido		
diasT	Tipo de variable: int	
valorD	Tipo de variable: int	
Código:	<pre>//con parametro function sueldo(pdiasT,pvalorD){ let diasT = pdiasT; let valorD = pvalorD; let sueldo; sueldo = valorD*diasT; return sueldo; }</pre>	

Nombre de la función: function salud(pdiasT,pvalorD)		Versión: 2.0
Descripción: Calcular el sueldo de una persona todo incluido		
salud	Tipo de variable: int	
Código: <div><pre>function salud(pdiasT,pvalorD){ let salud; salud= sueldo(pdiasT,pvalorD)*0.12; return salud }</pre></div>		



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: function pension(pdiasT,pvalorD)

Versión: 2.0

Descripción:

Calcular el sueldo de una persona todo incluido

pension

Tipo de variable: int

Código:

```
function pension(pdiasT,pvalorD){  
    let pension;  
    pension = sueldo(pdiasT,pvalorD)*0.16  
    return pension  
}
```

Nombre de la función: function arl(pdiasT,pvalorD)

Versión: 2.0

Descripción:

Calcular el sueldo de una persona todo incluido

arl

Tipo de variable: int

Código:

```
function arl(pdiasT,pvalorD){  
    let arl;  
    arl = sueldo(pdiasT,pvalorD)*0.052  
    return arl  
}
```




Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: function trasnporte(pdiasT,pvalorD,psalarioM)		Versión: 2.0
Descripción: Calcular el sueldo de una persona todo incluido		
trasporteT	Tipo de variable: int	
salarioMinimo	Tipo de variable: int	
Código: <pre>function transporte(pdiasT,pvalorD,psalarioM){ let transporteT; let salarioMinimo = psalarioM; let sueldoCal = sueldo(pdiasT,pvalorD) if(sueldoCal<=salarioMinimo){ transporteT= 114000; return transporteT; }else{ transporteT= 0; return transporteT; } }</pre>		

Nombre de la función: function retencion(pdiasT,pvalorD)		Versión: 2.0
Descripción: Calcular el sueldo de una persona todo incluido		
retencionT	Tipo de variable: int	
salarioMinimo	Tipo de variable: int	
Código: <pre>function retencion(pdiasT,pvalorD,psalarioM){ let retencionT; let salarioMinimo = psalarioM; let sueldoCal = sueldo(pdiasT,pvalorD); if (sueldoCal >= salarioMinimo * 4) { retencionT = sueldoCal * 0.04; } else { retencionT = 0; } return retencionT; }</pre>		



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: function sueldoTotal(pdiasT,pvalorD,psalarioM)		Versión: 2.0
Descripción: Calcular el sueldo de una persona todo incluido		
sueldoFinal	Tipo de variable: string	
Código:		
<pre>function sueldoTotal(pdiasT,pvalorD,psalarioM){ let sueldoInicial = sueldo(pdiasT,pvalorD); let subsidioTrasn = transporte(pdiasT,pvalorD,psalarioM); let pensionTotal = pension(pdiasT,pvalorD); let arlTotal = arl(pdiasT,pvalorD); let saludTotal = salud(pdiasT,pvalorD); let retencionToltal = retencion(pdiasT,pvalorD,psalarioM); let deducibles = pensionTotal+arlTotal+saludTotal+retencionToltal; let sueldoFinal = (sueldoInicial+subsidioTrasn)-deducibles; return sueldoFinal; }</pre>		



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: const sueldoExp = function(pdiasT,pvalorD)

Versión: 3.0

Descripción:

Calcular el sueldo de una persona todo incluido

diasT	Tipo de variable: int
valorD	Tipo de variable: int
sueldo	Tipo de variable: int

Código:

```
//con expresión
const sueldoExp = function(pdiasT,pvalorD){
  let diasT = pdiasT;
  let valorD = pvalorD;
  let sueldo;
  sueldo = valorD*diasT;
  return sueldo;
}
```

Nombre de la función: const sueldoExp = function(pdiasT,pvalorD)

Versión: 3.0

Descripción:

Calcular el sueldo de una persona todo incluido

salud	Tipo de variable: int
-------	-----------------------

Código:

```
const saludExp = function(pdiasT,pvalorD){
  let salud;
  salud= sueldo(pdiasT,pvalorD)*0.12;
  return salud
}
```



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: const pensionExp = function(pdiasT,pvalorD)		Versión: 3.0
Descripción: Calcular el sueldo de una persona todo incluido		
pension	Tipo de variable: int	
Código: <pre>const pensionExp = function(pdiasT,pvalorD){ let pension; pension = sueldo(pdiasT,pvalorD)*0.16 return pension }</pre>		

Nombre de la función: const arlExp = function(pdiasT,pvalorD)		Versión: 3.0
Descripción: Calcular el sueldo de una persona todo incluido		
arl	Tipo de variable: int	
Código: <pre>const arlExp = function(pdiasT,pvalorD){ let arl; arl = sueldo(pdiasT,pvalorD)*0.052 return arl }</pre>		



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: `const transporteExp = function(pdiasT,pvalorD,psalarioM)`

Versión: 3.0

Descripción:

Calcular el sueldo de una persona todo incluido

transporteT

Tipo de variable: int

salarioMinimo

Tipo de variable: int

Código:

```
const transporteExp = function(pdiasT,pvalorD,psalarioM){
    let transporteT;
    let salarioMinimo = psalarioM;
    let sueldoCal = sueldo(pdiasT,pvalorD)
    if(sueldoCal<=salarioMinimo){
        transporteT= 114000;
        return transporteT;
    }else{
        transporteT= 0;
        return transporteT;
    }
}
```

Nombre de la función: `const retencionExp = function(pdiasT,pvalorD,psalarioM)`

Versión: 3.0

Descripción:

Calcular el sueldo de una persona todo incluido

retencionT

Tipo de variable: string

salarioMinimo

Tipo de variable: int

Código:

```
const retencionExp = function(pdiasT,pvalorD,psalarioM){
    let retencionT;
    let salarioMinimo = psalarioM;
    let sueldoCal = sueldo(pdiasT,pvalorD);
    if (sueldoCal >= salarioMinimo * 4) {
        retencionT = sueldoCal * 0.04;
    } else {
        retencionT = 0;
    }
    return retencionT;
}
```



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: `const sueldoTotalExp = function(pdiasT,pvalorD,psalarioM)`

Versión: 3.0

Descripción:

Calcular el sueldo de una persona todo incluido

sueldoFinal

Tipo de variable: string

Código:

```
const sueldoTotalExp = function(pdiasT,pvalorD,psalarioM){
    let sueldoInicial = sueldo(pdiasT,pvalorD);
    let subsidioTrasn = transporte(pdiasT,pvalorD,psalarioM);
    let pensionTotal = pension(pdiasT,pvalorD);
    let arlTotal = arl(pdiasT,pvalorD);
    let saludTotal = salud(pdiasT,pvalorD);
    let retencionToltal = retencion(pdiasT,pvalorD,psalarioM);
    let deducibles = pensionTotal+arlTotal+saludTotal+retencionToltal;
    let sueldoFinal = (sueldoInicial+subsidioTrasn)-deducibles;

    return sueldoFinal;
}
```

CONSOLA:

```
CON PARAMETROS                                index.html:15
el sueldo inicial del trabajador es de: $600000 index.html:16
el monto que se le descontó por salud fue: $72000 index.html:17
el monto que se le descontó por pensión fue: $96000 index.html:18
el ARL fue de: $31200                          index.html:19
el sudcío de transporte que le corresponde es de: $114000 index.html:20
la retención del sueldo es de: $0              index.html:21
el total del deducible fue de : $199200        index.html:22
el sueldo total del empleado es de: $400800    index.html:23
                                                index.html:26
CON EXPRESIÓN
el sueldo inicial del trabajador es de: $9000000 index.html:27
el monto que se le descontó por salud fue: $1080000 index.html:28
el monto que se le descontó por pensión fue: $1440000 index.html:29
el ARL fue de: $468000                        index.html:30
el sudcío de transporte que le corresponde es de: $0 index.html:31
la retención del sueldo es de: $360000        index.html:32
el total del deducible fue de : $199200        index.html:33
el sueldo total del empleado es de: $400800    index.html:34
Live reload enabled.                          index.html:64
> |
```



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

16 - Promedio de Notas 3

Nombre de la función porcentajeNotas(pnota1,pnota2,pnota3)		Versión: 2.0
Descripción: Calcular el promedio de notas y demás		
nota1	Tipo de variable: float	
nota2	Tipo de variable: float	
nota3	Tipo de variable: float	
sumaPorcentaje	Tipo de variable: float	
Código:		
<pre>//con parametro function porcentajeNotas(pnota1,pnota2,pnota3){ let nota1 = pnota1; let nota2 = pnota2; let nota3 = pnota3; let sumaPorcentaje; porcentaje1= nota1*0.21; porcentaje2= nota2*0.35; porcentaje3= nota3*0.45; sumaPorcentaje = porcentaje1+porcentaje2+porcentaje3; if(sumaPorcentaje>4.5){ return sumaPorcentaje+" es una exelente nota."; }else{ if(sumaPorcentaje<=4.5 & sumaPorcentaje>=3.5){ return sumaPorcentaje+" es una buena nota."; }else{ if(sumaPorcentaje<=3.5 & sumaPorcentaje>=3.0){ return sumaPorcentaje+" es una mediocre nota.";; }else{ return sumaPorcentaje+" es una mala nota."; } } } }</pre>		



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: `const porcentajeNotasExp = function(pnota1,pnota2,pnota3)`

Versión: 3.0

Descripción:

Calcular el promedio de notas y demás

nota1	Tipo de variable: float
nota2	Tipo de variable: float
nota3	Tipo de variable: float
sumaPorcentaje	Tipo de variable: float

Código:

```
//con expresión
const porcentajeNotasExp = function(pnota1,pnota2,pnota3){
  let nota1 = pnota1;
  let nota2 = pnota2;
  let nota3 = pnota3;
  let sumaPor;
  porcentaje1= nota1*0.21;
  porcentaje2= nota2*0.35;
  porcentaje3= nota3*0.45;
  sumaPorcentaje = porcentaje1+porcentaje2+porcentaje3;
  if(sumaPorcentaje>4.5){
    return sumaPorcentaje+" es una exelente nota.";
  }else{
    if(sumaPorcentaje<=4.5 & sumaPorcentaje>=3.5){
      return sumaPorcentaje+" es una buena nota.";
    }else{
      if(sumaPorcentaje<=3.5 & sumaPorcentaje>=3.0){
        return sumaPorcentaje+" es una mediocre nota.";
      }else{
        return sumaPorcentaje+" es una mala nota.";
      }
    }
  }
}
```

CONSOLA:

CON PARAMETROS

[index.html:12](#)

4.855 es una exelente nota.

[index.html:13](#)

CON EXPRESIÓN

[index.html:14](#)

2.8179999999999996 es una mala nota.

[index.html:15](#)

Live reload enabled.

[index.html:45](#)

> |



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

CICLO WHILE

17 – Contar al 5

Nombre de la función: function contar(pnumero)		Versión: 2.0
Descripción: Imprime los números del 1 al 5		
contador	Tipo de variable: Alfanumérico	
resultado	Tipo de variable: string	
Código:		

```
//con parametro
function contar(numero) {
  let contador = 0;
  let resultado = ""
  while (contador < numero) {
    contador = contador + 1;
    resultado += `${contador}\n`
  }
  return resultado;
}
```

Nombre de la función: const contarExp = function(numero)		Versión: 3.0
Descripción: Imprimir los números del 1 al 5		
contador	Tipo de variable: Alfanumérica	
resultado	Tipo de variable: string	
Código:		
<pre>//con expresión const contarExp = function(numero) { let contador = 0; let resultado = "" while (contador < numero) { contador = contador + 1; resultado += `\${contador}\n` } return resultado; }</pre>		
CONSOLA:		
<pre>CON PARAMETROS 1 2 3 4 5 CON EXPRESIÓN 1 2 3 4 5 ></pre>		



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

18 – Factorial de 5

Nombre de la función: function factorial(numero)		Versión: 2.0
Descripción: Imprimir la factorial de 5		
factorial	Tipo de variable: int	
contador	Tipo de variable: int	
Código:		
<pre>//con parametro function factorial(numero) { let factorial = 1; let contador = 0; while (contador < numero) { contador = contador + 1; factorial = factorial * contador; } return factorial; }</pre>		

Nombre de la función: const factorialExp = function(numero)		Versión: 3.0
Descripción: Imprimir la factorial de 5		
factorial	Tipo de variable: Alfanumérica	
contador	Tipo de variable: int	
Código: <pre>//con expresión const factorialExp = function(numero) { let factorial = 1; let contador = 0; while (contador < numero) { contador = contador + 1; factorial = factorial * contador; } return factorial; }</pre>		
CONSOLA: <pre>CON PARAMETROS 120 CON EXPRESIÓN 120 Live reload enabled. > </pre>		



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

19 – Tabla del 5

Nombre de la función: function tabla5(numero)		Versión: 2.0
Descripción: Imprime la tabla del 5		
contador	Tipo de variable: int	
resultado	Tipo de variable: int	
multiplicar	Tipo de variable: int	
Código:		
<pre>//con parametro function tabla5(numero) { let contador = 0; let multiplicar = 0; let resultado = "" while (contador < 5) { contador = contador + 1; multiplicar = numero * contador; resultado += `\${numero} x \${contador} = \${multiplicar}\n`; } return resultado; }</pre>		

Nombre de la función: const tabla5Expresion = function(pNumero){		Versión: 3.0
Descripción: Imprime la tabla del 5		
contador	Tipo de variable: int	
resultado	Tipo de variable: int	
multiplicar	Tipo de variable: int	
Código:		
<pre>//con expresión const tabla5Exp = function(numero) { let contador = 0; let multiplicar = 0; let resultado = "" while (contador < 5) { contador = contador + 1; multiplicar = numero * contador; resultado += `\${numero} x \${contador} = \${multiplicar}\n`; } return resultado; }</pre>		
CONSOLE:		
<div>CON PARAMETROS</div> <div>5 x 1 = 5 5 x 2 = 10 5 x 3 = 15 5 x 4 = 20 5 x 5 = 25</div> <div>CON EXPRESIÓN</div> <div>5 x 1 = 5 5 x 2 = 10 5 x 3 = 15 5 x 4 = 20 5 x 5 = 25</div> <div>></div>		



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

20 – Tabla del 9 par e impar

Nombre de la función: function tabla9(numero)		Versión: 2.0
Descripción: Imprimir la tabla del 9 con los números pares e impares		
contador multiplicar resultado	Tipo de variable: int Tipo de variable: int Tipo de variable: string	
Código:	<pre>//con parametro function tabla9(numero) { let contador = 0; let multiplicar = 0; let resultado = ""; while (contador < 5) { contador = contador + 1; multiplicar = numero * contador; if (multiplicar % 2 === 0) { resultado += `\${numero} x \${contador} = \${multiplicar} es par.` + "\n"; } else { resultado += `\${numero} x \${contador} = \${multiplicar} es impar.` + "\n"; } } return resultado; }</pre>	

Nombre de la función: const tablaExpresion = function(pNumero)		Versión: 3.0
Descripción: Imprimir la tabla del 9 con los números pares e impares		
Tabla del Multiplicar	Tipo de variable: Alfanumérica	
Código:	<pre>//con expresión const tabla9Exp = function(numero) { let contador = 0; let multiplicar = 0; let resultado = ""; while (contador < 5) { contador = contador + 1; multiplicar = numero * contador; if (multiplicar % 2 === 0) { resultado += `\${numero} x \${contador} = \${multiplicar} es par.` + "\n"; } else { resultado += `\${numero} x \${contador} = \${multiplicar} es impar.` + "\n"; } } return resultado; }</pre>	
CONSOLA:		
<div>CON PARAMETROS index.html:12</div> <div>9 x 1 = 9 es impar. index.html:14</div> <div>9 x 2 = 18 es par.</div> <div>9 x 3 = 27 es impar.</div> <div>9 x 4 = 36 es par.</div> <div>9 x 5 = 45 es impar.</div> <div>CON EXPRESIÓN index.html:16</div> <div>9 x 1 = 9 es impar. index.html:17</div> <div>9 x 2 = 18 es par.</div> <div>9 x 3 = 27 es impar.</div> <div>9 x 4 = 36 es par.</div> <div>9 x 5 = 45 es impar.</div>		



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

21 – Tablas del 5 con ciclo FOR

Nombre de la función: function tabla5(numero)		Versión: 2.0
Descripción: imprimir la tabla del 5 hasta el 5 con ciclo FOR.		
multiplicar	Tipo de variable: int	
contador	Tipo de variable: int	
resultado	Tipo de variable: string	
Código:		
<pre>//con parametro function tabla5(numero) { let resultado = "" for (let contador = 1; contador <= 5; contador++) { let multiplicar = numero * contador; resultado += `\${numero + " x " + contador + " = " + multiplicar}\n` } return resultado; }</pre>		



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: `const tabla5Exp = function(pnumero)`

Versión: 3.0

Descripción:

imprimir la tabla del 5 hasta el 5 con ciclo FOR.

multiplicar

Tipo de variable: int

contador

Tipo de variable: int

resultado

Tipo de variable: string

Código:

```
//con expresión
const tabla5Exp = function(numero) {
  let resultado = ""
  for (let contador = 1; contador <= 5; contador++) {
    let multiplicar = numero * contador;

    resultado += `${numero} x ${contador} = ${multiplicar}\n`
  }
  return resultado;
}
```

CONSOLA:

CON PARAMETROS

[index.html:12](#)

```
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
```

[index.html:14](#)

CON EXPRESIÓN

[index.html:16](#)

```
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
```

[index.html:17](#)

>



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

CICLO FOR

22 – Tabla del 9

Nombre de la función: function tabla9(numero)		Versión: 2.0
Descripción: imprimir la tabla del 9 hasta el 5 con ciclo FOR y sacar los numero pares e impares		
Contador	Tipo de variable: Alfanumérico	
Código:		
<pre>//con parametro function tabla9(numero) { let resultado = "" for (let contador = 1; contador <= 5; contador++) { let multiplicar = numero * contador; if (multiplicar % 2 === 0) { resultado += `\${numero} x \${contador} = \${multiplicar} es par.` + "\n" } else { resultado += `\${numero} x \${contador} = \${multiplicar} es impar.` + "\n" } } return resultado; }</pre>		

Nombre de la función: const tabla9Exp = function(numero)		Versión: 3.0
Descripción: imprimir la tabla del 9 hasta el 5 con ciclo FOR y sacar los numero pares e impares		
Contador	Tipo de variable: Alfanumérica	
Código:	<pre>//con expresión const tabla9Exp = function(numero) { let resultado = "" for (let contador = 1; contador <= 5; contador++) { let multiplicar = numero * contador; if (multiplicar % 2 === 0) { resultado += `\${numero} x \${contador} = \${multiplicar} es par.` + "\n" } else { resultado += `\${numero} x \${contador} = \${multiplicar} es impar.` + "\n" } } return resultado; }</pre>	
CONSOLA:		
<pre>CON PARAMETROS 9 x 1 = 9 es impar. 9 x 2 = 18 es par. 9 x 3 = 27 es impar. 9 x 4 = 36 es par. 9 x 5 = 45 es impar. CON EXPRESIÓN 9 x 1 = 9 es impar. 9 x 2 = 18 es par. 9 x 3 = 27 es impar. 9 x 4 = 36 es par. 9 x 5 = 45 es impar.</pre>		



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

23 – Números de 1-5

Nombre de la función: function contar(pnumero)		Versión: 2.0
Descripción: los numeros deo 1 al 5 con ciclo FOR		
resultado contador	Tipo de variable: string tipo de variable: int	
Código: <pre>// Función con parámetro function contar(numero) { let resultado = "" for (let contador = 1; contador <= numero; contador++) { resultado += `\${contador}\n` } return resultado; }</pre>		

Nombre de la función: const contarExp = function(numero){		Versión: 3.0
Descripción: los numeros deo 1 al 5 con ciclo FOR		
contador resultado	Tipo de variable: int Tipo de variable: string	
Código:	<pre>// Función con expresión const contarExp = function(numero) { let resultado = "" for (let contador = 1; contador <= numero; contador++) { resultado += `\${contador}\n` } return resultado; };</pre>	
CONSOLA:	<div>CON PARAMETROS</div> <div>1 2 3 4 5</div> <div>CON EXPRESIÓN</div> <div>1 2 3 4 5</div> <div>></div>	



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

24 – Tabla del 5

Nombre de la función: factorial(numero)		Versión: 2.0
Descripción: calcular el factorial de 5 con ciclo for.		
factorial	Tipo de variable: int	
contador	Tipo de variable: int	
Código:		
<pre>// Función con parámetro function factorial(numero) { let factorial = 1; for (let contador = 1; contador <= numero; contador++) { factorial *= contador; } return factorial; }</pre>		

Nombre de la función: const factorialExp = function(numero)		Versión: 3.0
Descripción: calcular el facrorial de 5 con ciclo for.		
factorial	Tipo de variable: int	
contador	Tipo de variable: int	
Código: <pre>// Función con expresión const factorialExp = function(numero) { let factorial = 1; for (let contador = 1; contador <= numero; contador++) { factorial *= contador; } return factorial; };</pre>		
CONSOLA: <pre>CON PARAMETROS 120 CON EXPRESIÓN 120 Live reload enabled. > </pre>		



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

25 – Tabla del 1 hasta el 5 con While

Nombre de la función: tablas(tabla)

Versión: 2.0

Descripción:

tabla del 1 al 5 hasta el 5 e imprimir los pares e impares con ciclo while

contador

Tipo de variable: int

par

Tipo de variable: int

impar

Tipo de variable: int

multiplicar

Tipo de variable: int

res

Tipo de variable: alfanumerico

resultado

Tipo de variable: string

Código:

```
//con parametro
function tablas(tabla) {
    let contador = 1;
    let par = 0;
    let impar = 0;
    let resultado = "";

    while (contador <= tabla) {
        let multiplicar = 1;
        while (multiplicar <= 5) {
            let res = contador * multiplicar;

            if (res % 2 === 0) {
                par++;
                resultado += `${contador} x ${multiplicar} = ${res} Buzz.\n`;
            } else {
                impar++;
                resultado += `${contador} x ${multiplicar} = ${res} Bass.\n`;
            }

            multiplicar++;
        }
        contador++;
    }

    return `${resultado}\nLa cantidad de pares es: ${par}\nLa cantidad de impares es: ${impar}\n`;
}
```



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: `const tablasExp = function(tabla)`

Versión: 3.0

Descripción:

tabla del 1 al 5 hasta el 5 e imprimir los pares e impares con ciclo while

contador

Tipo de variable: int

par

Tipo de variable: int

impar

Tipo de variable: int

multiplicar

Tipo de variable: int

res

Tipo de variable: alfanumérico

resultado

Tipo de variable: string

Código:

```
//con expresión
const tablasExp = function(tabla) {
  let contador = 1;
  let par = 0;
  let impar = 0;
  let resultado = "";

  while (contador <= tabla) {
    let multiplicar = 1;
    while (multiplicar <= 5) {
      let res = contador * multiplicar;

      if (res % 2 === 0) {
        par++;
        resultado += `${contador} x ${multiplicar} = ${res} Buzz.\n`;
      } else {
        impar++;
        resultado += `${contador} x ${multiplicar} = ${res} Bass.\n`;
      }

      multiplicar++;
    }
    contador++;
  }

  return `${resultado}\nLa cantidad de pares es: ${par}\nLa cantidad de impares es: ${impar}\n`;
}
```

CONSOLA:

CON EXPRESIÓN

```
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.
```

```
La cantidad de pares es: 16
La cantidad de impares es: 9
```

> |

CON PARAMETROS

```
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.
```

```
La cantidad de pares es: 16
La cantidad de impares es: 9
```



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

26 – Tablas del 1 al 5 con FOR

Nombre de la función: tablas(tabla)

Versión: 2.0

Descripción:

tabla del 1 al 5 hasta el 5 e imprimir los pares e impares con ciclo FOR.

contador	Tipo de variable: int
par	Tipo de variable: int
impar	Tipo de variable: int
multiplicar	Tipo de variable: int
res	Tipo de variable: alfanumerico
resultado	Tipo de variable: string

Código:

```
//con parametro
function tablas(tabla) {
  let contador = 1;
  let par = 0;
  let impar = 0;
  let resultado = "";

  while (contador <= tabla) {
    let multiplicar = 1;
    while (multiplicar <= 5) {
      let res = contador * multiplicar;

      if (res % 2 === 0) {
        par++;
        resultado += `${contador} x ${multiplicar} = ${res} Buzz.\n`;
      } else {
        impar++;
        resultado += `${contador} x ${multiplicar} = ${res} Bass.\n`;
      }

      multiplicar++;
    }
    contador++;
  }

  return `${resultado}\nLa cantidad de pares es: ${par}\nLa cantidad de impares es: ${impar}\n`;
}
```



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: `const tablasExp = function(tabla)`

Versión: 3.0

Descripción:

tabla del 1 al 5 hasta el 5 e imprimir los pares e impares con ciclo FOR

contador	Tipo de variable: int
par	Tipo de variable: int
impar	Tipo de variable: int
multiplicar	Tipo de variable: int
res	Tipo de variable: alfanumerico
resultado	Tipo de variable: string

Código:

```
//con Expresión
const tablasExp = function(tabla) {
  let contador = 1;
  let par = 0;
  let impar = 0;
  let resultado = "";

  while (contador <= tabla) {
    let multiplicar = 1;
    while (multiplicar <= 5) {
      let res = contador * multiplicar;

      if (res % 2 === 0) {
        par++;
        resultado += `${contador} x ${multiplicar} = ${res} Buzz.\n`;
      } else {
        impar++;
        resultado += `${contador} x ${multiplicar} = ${res} Bass.\n`;
      }

      multiplicar++;
    }
    contador++;
  }

  return `${resultado}\nLa cantidad de pares es: ${par}\nLa cantidad de impares es: ${impar}\n`;
}
```

CONSOLA:

CON EXPRESIÓN

```
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.
```

La cantidad de pares es: 16
La cantidad de impares es: 9

> |

CON EXPRESIÓN

```
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.
```

La cantidad de pares es: 16
La cantidad de impares es: 9

> |