



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

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

Nombre de la función: holaMundoParametros(psaludo)		Versión: 2.0
Descripción: Función que saluda		
saludo	Tipo de variable: String	
Código: <pre>function holaMundoParametros(psaludo){ let saludo=psaludo+" Ok"; return saludo; }</pre>		
parametro: hola mundo		index.html:13
como una expresion: hola mundo		index.html:14



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función:
suma de dos números()

Versión: 1.0

Descripción:

Función suma dos números

sumar

Tipo de variable: float

Código:

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

Nombre de la función:

suma de dos números parámetro()

Versión: 2.0

Descripción:

Función que suma dos números

sumar

Tipo de variable: float

Código:

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

suma uno con parametro: 14

[index.html:11](#)

suma uno con expresion: 9

[index.html:12](#)



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: resultado resta, multiplicación y división parámetro()

Versión: 1.0

Descripción:

Función que suma, resta, multiplique y divide

Suma,
Resta,
multiplica,
divide

Tipo de variable: float

Código:

```
let numUno;  
let numDos;  
  
const suma = function(pnumeroUno, pnumeroDos){  
    let sumar;  
    numUno = pnumeroUno;  
    numDos = pnumeroDos;  
    sumar = numUno + numDos;  
    return sumar;  
}  
  
const resta = function(pnumeroUno, pnumeroDos){  
    let restar;  
    numUno = pnumeroUno;  
    numDos = pnumeroDos;  
    restar = numUno - numDos;  
    return restar;  
}  
  
const multiplicacion = function(pnumeroUno, pnumeroDos){  
    let multiplicar;  
    numUno = pnumeroUno;  
    numDos = pnumeroDos;  
    multiplicar = numUno * numDos;  
    return multiplicar;  
}  
  
const division = function(pnumeroUno, pnumeroDos){  
    let dividir;  
    numUno = pnumeroUno;  
    numDos = pnumeroDos;  
    dividir = numUno / numDos;  
    return dividir;  
}
```



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: resultado resta, multiplicación y división()

Versión: 2.0

Descripción:

Función que suma, resta, multiplica y divide

Suma,
Resta,
multiplica,
divide

Tipo de variable: float

Código:

```
function suma(pnumeroUno, pnumeroDos){
    let sumar;
    numUno = pnumeroUno;
    numDos = pnumeroDos;
    sumar = numUno + numDos;
    return sumar;
}
function resta(pnumeroUno, pnumeroDos){
    let restar;
    numUno = pnumeroUno;
    numDos = pnumeroDos;
    restar = numUno - numDos;
    return restar;
}
function multiplicacion (pnumeroUno, pnumeroDos){
    let multiplicar;
    numUno = pnumeroUno;
    numDos = pnumeroDos;
    multiplicar = numUno * numDos;
    return multiplicar;
}
function division(pnumeroUno, pnumeroDos){
    let dividir;
    numUno = pnumeroUno;
    numDos = pnumeroDos;
    dividir = numUno / numDos;
    return dividir;
}
```

suma uno: 8	index.html:11
Resta uno: -2	index.html:12
Multiplicacion uno: 35	index.html:13
suma Dos: 18	index.html:14
Division uno: 5	index.html:15
suma Tres: 13	index.html:16
Multiplicacion Dos: 160	index.html:17
suma cuatro: Error!!! no reconoce operador	index.html:18



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: porcentaje de un numero parametro()		Versión: 1.0
Descripción: Función porcentaje		
porcientos	Tipo de variable: float	
Código: <pre>let numUno; const porcientoExp = function(pnumeroUno,){ let porcientos; numUno = pnumeroUno; porcientos = numUno / 100; return porcientos; }</pre>		

Nombre de la función: porcentaje de un número ()		Versión: 2.0
Descripción: Función porcentaje		
porcientos	Tipo de variable: float	
Código: <pre>function porciento(pnumeroUno,) { let porcientos; numUno = pnumeroUno; porcientos = numUno / 100; return porcientos; }</pre>		
Porcentaje uno con parametro: 0.05		index.html:11
Porcentaje uno con exprecion: 0.08		index.html:12



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función:

promedio de 3 notas Parámetros()

Versión: 1.0

Descripción:

Función promediar 3 notas

promedio

Tipo de variable: float

Código:

```
let notaUno;  
let notaDos;  
let notaTres;  
  
const promediosExp = function(pnnotaUno, pnnotaDos, pnnotaTres){  
  let promedio;  
  notaUno = pnnotaUno;  
  notaDos = pnnotaDos;  
  notaTres = pnnotaTres;  
  promedio = (notaUno + notaDos + notaTres)/3;  
  return promedio;  
}
```

Nombre de la función:

promedio de 3 notas ()

Versión: 2.0

Descripción:

Función promediar 3 notas

promedio

Tipo de variable: float

Código:

```
function promedios(pnnotaUno, pnnotaDos, pnnotaTres){  
  let promedio;  
  notaUno = pnnotaUno;  
  notaDos = pnnotaDos;  
  notaTres = pnnotaTres;  
  promedio = (notaUno + notaDos + notaTres)/3;  
  return promedio;  
}
```

Promedios uno con parametro: 6

[index.html:11](#)

Promedios uno con exprecion: 7

[index.html:12](#)



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: porcentaje de 3 notas parametro()		Versión: 1.0
Descripción: Función que promediar 3 notas		
suma	Tipo de variable: float	
Código:	<pre>let notaUno; let notaDos; let notaTres; const porcientExp = function(pnnotaUno,pnnotaDos,pnnotaTres){ let suma; notaUno = pnnotaUno * 30/100; notaDos = pnnotaDos * 30/100; notaTres = pnnotaTres * 30/100; suma = notaUno + notaDos + notaTres; return suma; }</pre>	

Nombre de la función: porcentaje de 3 notas ()		Versión: 2.0
Descripción: Función que promediar 3 notas		
suma	Tipo de variable: float	
Código:		
<pre>function porciento(pnnotaUno,pnnotaDos,pnnotaTres){ let suma; notaUno = pnnotaUno * 30/100; notaDos = pnnotaDos * 30/100; notaTres = pnnotaTres * 30/100; suma = notaUno + notaDos + notaTres; return suma; }</pre>		
Promedios uno con expresion: 6.3		index.html:11
Promedios uno con parametro: 5.4		index.html:12



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función:

las áreas geométricas parámetro()

Versión: 1.0

Descripción:

Función que de el resultado de las fórmulas de las áreas

Cuadrado,
rectángulo,
triángulo

Tipo de variable: float

Código:

```
let lado;  
let baseUno;  
let alturaUno;  
let baseDos;  
let alturalDos;  
  
const cuadradoExp = function(pnlado, pnnlado){  
    let cuadrados;  
    lado = pnlado;  
    lado = pnnlado;  
    cuadrados = pnlado * pnnlado;  
    return cuadrados;  
}  
  
const rectanguloExp = function(pnbaseUno, pnalturaUno){  
    let rectangulos;  
    alturaUno = pnalturaUno;  
    baseUno = pnbaseUno;  
    rectangulos = pnbaseUno * pnalturaUno;  
    return rectangulos;  
}  
  
const trianguloExp = function(pnbaseDos, pnalturaDos){  
    let triangulos;  
    alturalDos = pnalturaDos;  
    baseDos = pnbaseDos;  
    triangulos = (pnbaseDos * pnalturaDos)/2;  
    return triangulos;  
}
```




Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función:
las áreas geométricas ()

Versión: 2.0

Descripción:

Función que devuelve el resultado de las fórmulas de las áreas

Cuadrado,
rectángulo,
triángulo

Tipo de variable: float

Código:

```
function cuadrado(pnlado, pnnlado){
    let cuadrados;
    lado = pnlado;
    lado = pnnlado;
    cuadrados = pnlado * pnnlado;
    return cuadrados;
}

function rectangulo(pnbaseUno, pnalturaUno){
    let rectangulos;
    alturaUno = pnalturaUno;
    baseUno = pnbaseUno;
    rectangulos = pnbaseUno * pnalturaUno;
    return rectangulos;
}

function triangulo(pnbaseDos, pnalturaDos){
    let triangulos;
    alturalDos = pnalturaDos;
    baseDos = pnbaseDos;
    triangulos = (pnbaseDos * pnalturaDos)/2;
    return triangulos;
}
```

Area del cuadrado parametro: 25

[index.html:11](#)

Area del rectangulo parametro: 80

[index.html:12](#)

Area del triangulo parametro: 15

[index.html:13](#)

Area del cuadrado: 4

[index.html:14](#)

Area del rectangulo: 40

[index.html:15](#)

Area del triangulo: 22.5

[index.html:16](#)



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función:

pago total de una persona parámetro()

Versión: 1.0

Descripción:

Función que de el pago total de una persona

Pago total de una persona

Tipo de variable: float

Código:

```
function suledo(pdiaT, pvalorD) {  
    diaT = pdiaT;  
    valorD = pvalorD;  
    pago = diaT * valorD;  
    return pago;  
}  
  
function saludp(ppsalud) {  
    porcentaje = ppsalud;  
    operacion = pago * porcentaje;  
    return operacion;  
}  
  
function pensionp(ppesion) {  
    porcentaje = ppesion;  
    operacion = pago * ppesion;  
    return operacion;  
}  
  
function arlp(pparl) {  
    porcentaje = pparl;  
    operacion = pago * porcentaje;  
    return operacion;  
}  
  
function descuentos(psalud, ppesnsion, parl) {  
    let pensionD;  
    let arlD;  
    let saludD;  
    saludD = psalud;  
    pensionD = ppesnsion;  
    arlD = parl;  
  
    operacion = saludD + pensionD + arlD;  
    return operacion;  
}
```



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función:
pago total de una persona()

Versión: 2.0

Descripción:

Función que de el pago total de una persona

Pago total de una persona

Tipo de variable: float

Código:

```
const sueldoExp = function (pdiaT, pvalorD) {  
    diaT = pdiaT;  
    valorD = pvalorD;  
    pago = diaT * valorD;  
    return pago;  
}  
  
const saludpExp = function (ppsalud) {  
    porcentaje = ppsalud;  
    operacion = pago * porcentaje;  
    return operacion;  
}  
  
const pensionpExp = function (ppesion) {  
    porcentaje = ppesion;  
    operacion = pago * ppesion;  
    return operacion;  
}  
  
const arlpExp = function (pparl) {  
    porcentaje = pparl;  
    operacion = pago * porcentaje;  
    return operacion;  
}  
  
const descuentosExp = function(psalud, ppesnsion, parl) {  
    let pensionD;  
    let arlD;  
    let saludD;  
    saludD = psalud;  
    pensionD = ppesnsion;  
    arlD = parl;  
  
    operacion = saludD + pensionD + arlD;  
    return operacion;  
}
```

Su pago es: 400.
El porcentaje de la salud es48%.
El porcentaje de la pensión es64%.
el porcentaje de ARL es20.8%.
el total a pagar con el descuentos será:132.8.
el Sueldo Total a pagar es: 267.2

[index.html:24](#)



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: el numero mayor de 2 numero parámetro()		Versión: 1.0
Descripción: Función que imprima el numero mayor de 2 números		
2 números	Tipo de variable: int – int	
Código:	<pre>let numeroUno; let numeroDos; const mayorExp = function(pnumeroUno, pnumeroDos){ numeroUno = pnumeroUno; numeroDos = pnumeroDos; if(numeroUno > numeroDos){ return "El numero uno es mayor "+numeroUno } else{ return "El numero dos es mayor " + numeroDos } }</pre>	

Nombre de la función: el número mayor de 2 numero ()		Versión: 2.0
Descripción: Función que imprima el número mayor de 2 números		
2 números	Tipo de variable: int – int	
Código:	<pre>function mayor(pnumeroUno, pnumeroDos) { numeroUno = pnumeroUno; numeroDos = pnumeroDos; if(numeroUno> numeroDos){ return "El numero uno es mayor "+numeroUno } else{ return "El numero dos es mayor " + numeroDos } }</pre>	
numero uno con expresion: El numero uno es mayor 5		index.html:11
numero dos con expresion: El numero dos es mayor 15		index.html:12
numero uno : El numero dos es mayor 15		index.html:13
numero dos: El numero dos es mayor 15		index.html:14



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función:

calcular la edad de una persona parámetro ()

Versión: 1.0

Descripción:

Función que calcule la edad de una persona

edad

Tipo de variable: String

Código:

```
let edad
let fechaNacimiento;
let fechaActual;
const edadesExp = function (pfechaNacimiento,pfechaActual){

    fechaNacimiento = pfechaNacimiento;
    fechaActual = pfechaActual
    edad = fechaActual - fechaNacimiento;
    if(edad < 18 ){
        return "es menor de edad" + edad;
    }
    else {
        return "es mayor de edad" + edad;
    }
}
```

Nombre de la función:

calcular la edad de una persona ()

Versión: 2.0

Descripción:

Función que calcule la edad de una persona

edad

Tipo de variable: String

Código:

```
function edades(pfechaNacimiento,pfechaActual){

    fechaNacimiento = pfechaNacimiento;
    fechaActual = pfechaActual
    edad = fechaActual - fechaNacimiento;
    if(edad < 18 ){
        return "es menor de edad" + edad;
    }
    else {
        return "es mayor de edad" + edad;
    }
}
```

edad Como Expresión:
es mayor de edad24

[index.html:12](#)



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: imprimir el número mayor parametro()		Versión: 1.0
Descripción: Función que calcule el numero mayor		
Dos numeros	Tipo de variable: numerico	
Código:	<pre>let numeroUno let numeroDos const mayorExp= function (pnumeroUno,pnumeroDos){ numeroUno = pnumeroUno; numeroDos = pnumeroDos; if(numeroUno == numeroDos){ return "Los número son iguales"; }else if(numeroUno > numeroDos){ return `El número uno es mayor \${numeroUno}`; }else{ return `El número dos es mayor \${numeroDos}`; } }</pre>	

Nombre de la función: imprimir el número mayor()		Versión: 2.0
Descripción: Función que calcule el numero mayor		
Dos numeros	Tipo de variable: numerico	
Código:	<pre>function mayorr (pnumeroUno,pnumeroDos){ numeroUno = pnumeroUno; numeroDos = pnumeroDos; if(numeroUno == numeroDos){ return "Los número son iguales"; }else if(numeroUno > numeroDos){ return `El número uno es mayor \${numeroUno}`; }else{ return `El número dos es mayor \${numeroDos}`; } }</pre>	
Como Expresión: El número uno es mayor 10		index.html:11
Como parametro: El número dos es mayor 8		index.html:12



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: el número mayor de 3 números parámetro()	Versión: 1.0
--	---------------------

Descripción: Función que calcule el numero mayor de 3 numeros

tres numeros	Tipo de variable: numerico
--------------	----------------------------

Código:	<pre>function mayor (numero){ let numeroP = numero; let mayorr; mayorr =numeroP * numeroP; return mayorr; } let numUno; let numDos; let numTres; function validar (numUno,numDos,numTres) { numUno = numUno; numDos = numDos; numTres = numTres; if(numUno == numDos && numUno == numTres && numDos == numTres) { return "Los tres numeros son iguales"; }else if (numUno > numDos && numUno > numTres) { return "El número uno es mayor que número dos y que número tres: El número uno es \${numUno}, El número dos es \${numDos}, El número tres es \${numTres}"; } else if(numDos > numUno && numDos > numTres){ return "El número dos es mayor que número uno y que número tres: El número uno es \${numUno}, El número dos es \${numDos}, El número tres es \${numTres}"; }else{ return "El número tres es mayor que número uno y que número tres: El número uno es \${numUno}, El número dos es \${numDos}, El número tres es \${numTres}"; } }</pre>
---------	--

Nombre de la función: el número mayor de 3 números ()	Versión: 2.0
---	---------------------

Descripción: Función que calcule el numero mayor de 3 numeros

tres números	Tipo de variable: numerico
--------------	----------------------------

Código:	<pre>const mayorExp = function (numero){ let numeroP = numero; let mayorr; mayorr =numeroP * numeroP; return mayorr; } let nuUno; let nuDos; let nuTres; const validarExp = function (numUno,numDos,numTres) { nuUno = numUno; nuDos = numDos; nuTres = numTres; if(numUno == numDos && numUno == numTres && numDos == numTres) { return "Los tres numeros son iguales"; }else if (numUno > numDos && numUno > numTres) { return "El número uno es mayor que número dos y que número tres: El número uno es \${numUno}, El número dos es \${numDos}, El número tres es \${numTres}"; } else if(numDos > numUno && numDos > numTres){ return "El número dos es mayor que número uno y que número tres: El número uno es \${numUno}, El número dos es \${numDos}, El número tres es \${numTres}"; }else{ return "El número tres es mayor que número uno y que número tres: El número uno es \${numUno}, El número dos es \${numDos}, El número tres es \${numTres}"; } }</pre>
numero mayor uno: 25	index.html:19
numero mayor Dos: 9	index.html:20
numero mayor Tres: 16	index.html:21
El número uno es mayor que número dos y que número tres: El número uno es 25, El número dos es 9, El número tres es 16	index.html:23



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: Calcular las áreas de 3 cuadrados y si son iguales parámetro()	Versión: 1.0
---	---------------------

Descripción:

Función que calcular las áreas de 3 cuadrados

3 cuadrados

Tipo de variable: numerico

Código:

```
let ladoUno;
let ladoDos;
let ladoTres;
const areasExp = function (pladoUno,pladoDos,pladoTres){
    ladoUno = pladoUno;
    ladoDos = pladoDos;
    ladoTres = pladoTres;
    let areaUno;
    let areaDos;
    let areaTres;
    areaUno = ladoUno * ladoUno;
    areaDos = ladoDos * ladoDos;
    areaTres = ladoTres * ladoTres;

    if (areaUno == areaDos && areaUno == areaTres && areaDos == areaTres ){
        return `El area del primer cuadrado es mayor.\nEl area cuadrado uno es ${areaUno},\nEl area cuadrado dos es ${areaDos},\nEl area cuadrado tres es ${areaTres}`
    }else if(areaUno > areaDos && areaUno > areaTres){
        return `El area del segundo cuadrado es mayor.\nEl area cuadrado uno es ${areaUno},\nEl area cuadrado dos es ${areaDos},\nEl area cuadrado tres es ${areaTres}`
    }else if(areaDos > areaUno && areaDos > areaTres){
        return `El area del segundo cuadrado es mayor.\nEl area cuadrado uno es ${areaUno},\nEl area cuadrado dos es ${areaDos},\nEl area cuadrado tres es ${areaTres}`
    }else{
        return `El area del tercer cuadrado es mayor.\nEl area cuadrado uno es ${areaUno},\nEl area cuadrado dos es ${areaDos},\nEl area cuadrado tres es ${areaTres}`
    }
}
```




Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función:

Calcular las áreas de 3 cuadrados y si son iguales ()

Versión: 2.0

Descripción:

Función que calcular las áreas de 3 cuadrados

3 cuadrados

Tipo de variable: numerico

Código:

```
function areas (pladoUno,pladoDos,pladoTres){
  ladoUno = pladoUno;
  ladoDos = pladoDos;
  ladoTres = pladoTres;

  let areaUno;
  let areaDos;
  let areaTres;

  areaUno = ladoUno * ladoUno;
  areaDos = ladoDos * ladoDos;
  areaTres = ladoTres * ladoTres;

  if (areaUno == areaDos && areaUno == areaTres && areaDos == areaTres ){
    return `El area del primer cuadrado es mayor.\nEl area cuadrado uno es ${areaUno},\nEl area cuadrado dos es ${areaDos},\nEl area cuadrado tres es ${areaTres}`
  }else if(areaUno > areaUno && areaUno > areaTres){
    return `El area del segundo cuadrado es mayor.\nEl area cuadrado uno es ${areaUno},\nEl area cuadrado dos es ${areaDos},\nEl area cuadrado tres es ${areaTres}`
  }else if(areaDos > areaUno && areaDos > areaTres){
    return `El area del segundo cuadrado es mayor.\nEl area cuadrado uno es ${areaUno},\nEl area cuadrado dos es ${areaDos},\nEl area cuadrado tres es ${areaTres}`
  }else{
    return `El area del tercer cuadrado es mayor.\nEl area cuadrado uno es ${areaUno},\nEl area cuadrado dos es ${areaDos},\nEl area cuadrado tres es ${areaTres}`
  }
}
```

Como Expresión:

[index.html:15](#)

El area del segundo cuadrado es mayor.

El area cuadrado uno es 25,

El area cuadrado dos es 100,

El area cuadrado tres es 36

Como parametro:

[index.html:16](#)

El area del tercer cuadrado es mayor.

El area cuadrado uno es 16,

El area cuadrado dos es 225,

El area cuadrado tres es 1225



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función:

Calcular la edad de 3 personas parámetro()

Versión: 1.0

Descripción:

Función que calcular la edad de 3 personas

3 personas

Tipo de variable: numérico

Código:

```
const edadesExp = function(pfechaAc, pfechaNac) {  
  
    fechaAc = pfechaAc;  
    fechaNac = pfechaNac;  
  
    let diferencia;  
    let edad;  
  
    diferencia = fechaAc - fechaNac;  
    edad = Math.floor(diferencia / (1000 * 60 * 60 * 24 * 365.25));  
    return edad;  
}  
const validarExp = function(pedadUno, pedadDos, pedadTres){  
  
    edadUno = pedadUno;  
    edadDos = pedadDos;  
    edadTres = pedadTres  
  
    if(edadUno >=18){  
        resultado += "La edad es: "+edadUno+" años, Es mayor de edad.\n";  
    }else{  
        resultado += "La edad es: "+edadUno+" años, Es menor de edad.\n";  
    }  
    if(edadDos >=18){  
        resultado += "La edad es: "+edadDos+" años, Es mayor de edad.\n";  
    }else{  
        resultado += "La edad es: "+edadDos+" años, Es menor de edad.\n";  
    }  
    if(edadTres >=18){  
        resultado += "La edad es: "+edadTres+" años, Es mayor de edad.\n";  
    }else{  
        resultado += "La edad es: "+edadTres+" años, Es menor de edad. ";  
    }  
    return resultado;  
}
```



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función:

Calcular la edad de 3 personas ()

Versión: 1.0

Descripción:

Función que calcular la edad de 3 personas

3 personas

Tipo de variable: numerico

Código:

```
let fechaAc;  
let fechaNac;  
function edades(pfFechaAc, pfFechaNac) {  
    fechaAc = pfFechaAc;  
    fechaNac = pfFechaNac;  
  
    let diferencia;  
    let edad;  
  
    diferencia = fechaAc - fechaNac;  
    edad = Math.floor(diferencia / (1000 * 60 * 60 * 24 * 365.25));  
    return edad;  
}  
  
let edadUno;  
let edadDos;  
let edadTres;  
let resultado = "";  
function validar(pedadUno, pedadDos, pedadTres) {  
    edadUno = pedadUno;  
    edadDos = pedadDos;  
    edadTres = pedadTres  
  
    if(edadUno >= 18){  
        resultado += "La edad es: "+edadUno+" años, Es mayor de edad.\n";  
    }else{  
        resultado += "La edad es: "+edadUno+" años, Es menor de edad.\n";  
    }  
    if(edadDos >= 18){  
        resultado += "La edad es: "+edadDos+" años, Es mayor de edad.\n";  
    }else{  
        resultado += "La edad es: "+edadDos+" años, Es menor de edad.\n";  
    }  
    if(edadTres >= 18){  
        resultado += "La edad es: "+edadTres+" años, Es mayor de edad.\n";  
    }else{  
        resultado += "La edad es: "+edadTres+" años, Es menor de edad. ";  
    }  
    return resultado;  
}
```

La edad de la primera persona es: 19

[index.html:20](#)

La edad de la segunda persona es: 44

[index.html:21](#)

La edad de la tercera persona es: 17

[index.html:22](#)

La edad es: 19 años, Es mayor de edad.

[index.html:24](#)

La edad es: 44 años, Es mayor de edad.

La edad es: 17 años, Es menor de edad.



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función:

Calcular el pago total del sueldo de una persona parámetro()

Versión: 1.0

Descripción:

Función que calcule el pago total del sueldo de una persona

1 persona

Tipo de variable: float

Código:

```
const sueldoExp = function(pdiasT,pvalorD){
    diasT = pdiasT;
    valorD = pvalorD;
    salario = diasT * valorD;
    return salario;
}

const saludExp = function(psalarario,ppSalud){
    salario = psalarario;
    pSalud = ppSalud;
    operaciones = salario * pSalud;
    return operaciones;
}

const pensionExp = function(psalarario,ppPension){
    salario = psalarario;
    pPension = ppPension;
    operaciones = salario * ppPension;
    return operaciones;
}

const arlExp = function(psalarario,ppArl){
    salario = psalarario;
    pArl = ppArl;
    operaciones = salario * ppArl;
    return operaciones;
}
```

```
const validacionExp = function(psalarario,psalarioM,ptrasnporte,pretencion){
    salario = psalarario;
    salariorM = psalarioM;
    trasnporteP = ptrasnporte;
    retencionP = pretencion;
    if(salarario <= salariorM * 2){
        trasnporteP = ptrasnporte;
    }else{
        trasnporteP = 0
    }
    if(salarario > salariorM * 4){
        retencionP = salario * pretencion;
    }else{
        retencionP = 0;
    }
    return {
        trasnporteP,
        retencionP
    };
}
```



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función:

Calcular el pago total del sueldo de una persona()

Versión: 2.0

Descripción:

Función que calcule el pago total del sueldo de una persona

1 persona

Tipo de variable: float

Código:

```
let diasT;  
let valorD;  
let operaciones;  
let salario;  
let pSalud;  
let pPension;  
let pArl;  
let salarioM;  
let transporteP;  
let retencionP;  
function sueldo(pdiasT,pvalorD){  
    diasT = pdiasT;  
    valorD = pvalorD;  
    salario = diasT * valorD;  
    return salario;  
}  
function salud(psalarario,ppSalud){  
    salario = psalarario;  
    pSalud = ppSalud;  
    operaciones = salario * pSalud;  
    return operaciones;  
}  
function pension(psalarario,ppPension){  
    salario = psalarario;  
    pPension = ppPension;  
    operaciones = salario * ppPension;  
    return operaciones;  
}
```

```
function arl(psalarario,ppArl){  
    salario = psalarario;  
    pArl = ppArl;  
    operaciones = salario * pArl;  
    return operaciones;  
}  
function validacion(psalarario,psalararioM,ptrasnporte,pretencion){  
    salario = psalarario;  
    salarioM = psalararioM;  
    trasnporteP = ptrasnporte;  
    retencionP = pretencion;  
    if(salario <= salarioM * 2){  
        trasnporteP = ptrasnporte;  
    }else{  
        trasnporteP = 0  
    }  
    if(salario > salarioM * 4){  
        retencionP = salario * pretencion;  
    }else{  
        retencionP = 0;  
    }  
    return {  
        trasnporteP,  
        retencionP  
    };  
}
```

pago 6000000	index.html:21
salud 720000	index.html:24
pensión 960000	index.html:27
arl 312000	index.html:30
retencion 240000	index.html:33
trasporte 0	index.html:34
descuento 2232000	index.html:37
total 3768000	index.html:40



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: Calcular tres notas e imprimir parametro ()		Versión: 1.0
Descripción: Función que calcule la s3 notas e imprimir		
3 notas	Tipo de variable: float	
Código:	<pre>const notasExp = function(pnota, pporcentaje) { nota = pnota; porcentaje = pporcentaje; operacion = nota * porcentaje; return operacion; } const validarExp = function(psuma) { suma = psuma if (suma > 4.5) { return "La calificación es Superior " + suma; } else if (suma <= 4.5 && suma > 3.5) { return "La calificación es Buena " + suma; } else if (suma >= 3 && suma <= 3.5) { return "La calificación es Media " + suma; } else { return "la calificación es Mala" + suma; } }</pre>	

Nombre de la función: Calcular tres notas e imprimir ()		Versión: 2.0
Descripción: Función que calcule la 3 notas e imprimir		
3 notas	Tipo de variable: float	
Código:	<pre>let nota; let porcentaje; let operacion; let suma; function notas(pnota, pporcentaje) { nota = pnota; porcentaje = pporcentaje; operacion = nota * porcentaje; return operacion; } function validar(psuma) { suma = psuma if (suma > 4.5) { return "La calificación es Superior " + suma; } else if (suma <= 4.5 && suma > 3.5) { return "La calificación es Buena " + suma; } else if (suma >= 3 && suma <= 3.5) { return "La calificación es Media " + suma; } else { return "la calificación es Mala" + suma; } }</pre> <p>La nota uno es 2, que equivale al 0.4% del examen. index.html:21 La nota dos es 3, que equivale al 1.0499999999999998% del examen. La nota tres es 2.5, que equivale al 1.125% del examen. la calificación es Mala2.5749999999999997 index.html:27</p>	



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: Cuente e imprima los numeros del 1 al 5 parametro ()		Versión: 1.0
Descripción: Contar e imprimir numero del 1 al 5		
numero	Tipo de variable: float	
Código:	<pre>let numero; let contador; const contarExp = function (pnumero,pcontador) { numero = pnumero; contador = pcontador; contador = 0; let resultado = ""; while (contador < numero) { contador = contador + 1; resultado += contador + "\n"; } return resultado; }</pre>	

Nombre de la función: Cuente e imprima los numeros del 1 al 5 ()		Versión: 2.0
Descripción: Contar e imprimir numero del 1 al 5		
numero	Tipo de variable: float	
Código:	<pre>function contar (pnumero,pcontador) { numero = pnumero; contador = pcontador; contador = 0; let resultado = ""; while (contador < numero) { contador = contador + 1; resultado += contador + "\n"; } return resultado; }</pre> <div><p>Como Expresión: index.html:12</p><pre>1 2 3 4 5</pre><p>Como parámetro: index.html:13</p><pre>1 2 3 4 5</pre></div>	



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función:
imprimir factorial de 5 parámetro()

Versión: 1.0

Descripción:

Imprimir factorial de 5

numero

Tipo de variable: float

Código:

```
let numero;  
let contador;  
let factorial;  
const factorrExp = function (pnumero,pcontador,pfactorial){  
    numero = pnumero;  
    contador = pcontador;  
    factorial = pfactorial;  
  
    contador = 0;  
    factorial = 1;  
  
    while(contador < numero){  
        contador = contador + 1;  
        factorial = factorial * contador;  
    }  
    return factorial;  
}
```

Nombre de la función:
imprimir factorial de 5 ()

Versión: 2.0

Descripción:

Imprimir factorial de 5

numero

Tipo de variable: float

Código:

```
function factor (pnumero,pcontador,pfactorial){  
    numero = pnumero;  
    contador = pcontador;  
    factorial = pfactorial;  
  
    contador = 0;  
    factorial = 1;  
    while(contador < numero){  
        contador = contador + 1;  
        factorial = factorial * contador;  
    }  
    return factorial;  
}
```

Como Expresión:
120

[index.html:12](#)

Como parámetro:
120

[index.html:13](#)



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: Realizar la tabla de multiplicar del 5 parametro()		Versión: 1.0
Descripción: Realizar la tabla de multiplicar del 5		
numero	Tipo de variable: float	
Código:	<pre>let multiplicar; let contador; let numero; const tablaExp = function (pmultiplicar,){ multiplicar = pmultiplicar; let contador = 0; let numero = 5; let resultado; let inf = ""; while (contador < numero) { contador = contador +1; resultado = multiplicar * contador; inf += `\${multiplicar} x \${contador} = \${resultado}\n` } return inf; }</pre>	
Nombre de la función: Realizar la tabla de multiplicar del 5()		Versión: 2.0
Descripción: Realizar la tabla de multiplicar del 5		
numero	Tipo de variable: float	
Código:	<pre>function tabla(pmultiplicar,){ multiplicar = pmultiplicar; let contador = 0; let numero = 5; let resultado; let inf = ""; while (contador < numero) { contador = contador +1; resultado = multiplicar * contador; inf += `\${multiplicar} x \${contador} = \${resultado}\n` } return inf; }</pre> <div><p>Como Expresión: index.html:12</p><p>5 x 1 = 5 5 x 2 = 10 5 x 3 = 15 5 x 4 = 20 5 x 5 = 25</p><p>Como parametro: index.html:13</p><p>5 x 1 = 5 5 x 2 = 10 5 x 3 = 15 5 x 4 = 20 5 x 5 = 25</p></div>	



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: Realizar la tabla del 9 que multiplique hasta 5 parametro()	Versión: 1.0
Descripción: Realizar la tabla del 9 que multiplique hasta 5 si son pares e impares	
numero	Tipo de variable: float
Código:	<pre>let tablasMultiplicar; const tablaExp = function (ptablasMultiplicar){ tablasMultiplicar = ptablasMultiplicar; let numero; let contador; let resultados=""; let resultado; numero = 5; contador = 0; resultados = ""; while (contador < numero) { contador = contador +1 resultado = tablasMultiplicar * contador; resultados += `\${tablasMultiplicar} x \${contador} = \${resultado} `; if(resultado %2==0){ resultados += "par\n"; }else{ resultados += "impar\n"; } } return resultados; }</pre>

Nombre de la función: Realizar la tabla del 9 que multiplique hasta 5()	Versión: 2.0
Descripción: Realizar la tabla del 9 que multiplique hasta 5 si son pares e impares	
numero	Tipo de variable: float
Código:	<div><pre>function tabla (ptablasMultiplicar){ tablasMultiplicar = ptablasMultiplicar; let numero; let contador; let resultados=""; let resultado; numero = 5; contador = 0; resultados = ""; while (contador < numero) { contador = contador +1 resultado = tablasMultiplicar * contador; resultados += `\${tablasMultiplicar} x \${contador} = \${resultado} `; if(resultado %2==0){ resultados += "par\n"; }else{ resultados += "impar\n"; } } return resultados; }</pre></div> <div><p>Como Expresión: index.html:12</p><p>9 x 1 = 9 impar 9 x 2 = 18 par 9 x 3 = 27 impar 9 x 4 = 36 par 9 x 5 = 45 impar</p><p>Como parametro: index.html:13</p><p>9 x 1 = 9 impar 9 x 2 = 18 par 9 x 3 = 27 impar 9 x 4 = 36 par 9 x 5 = 45 impar</p></div>



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: Imrpimir la factorial de 1 al 5 con for parametro()		Versión: 1.0
Descripción: Contar de 1 al 5 con for		
numero	Tipo de variable: float	
Código:	<pre>let numeros; const contarExp = function (pnumero){ numeros = pnumero; let contador; let numero = 5; let resultado = ""; for(contador = 1; numero >= contador; contador ++){ resultado += contador+".\n"; } return resultado; }</pre>	

Nombre de la función: Imrpimir la factorial del 1 al 5 con for ()		Versión: 2.0
Descripción: Contar de 1 al 5 con for		
numero	Tipo de variable: float	
Código:	<pre>function contar(pnumero){ numeros = pnumero; let contador; let numero = 5; let resultado = ""; for(contador = 1; numero >= contador; contador ++){ resultado += contador+".\\n"; } return resultado; }</pre>	
Como Expresión:		index.html:12
1.		
2.		
3.		
4.		
5.		
Como parametro:		index.html:13
1.		
2.		
3.		
4.		
5.		



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: Imrpimir la factorial de 5 con for parametro()		Versión: 1.0
Descripción: Factorial del 5 con for		
numero	Tipo de variable: float	
Código:	<pre>let numero; const factorExp = function (pnumero){ numero = pnumero; let factorial = 1; let contador; for(contador = 1; numero >= contador; contador++){ factorial = factorial * contador; } return factorial; }</pre>	

Nombre de la función: Imrpimir la factorial de 5 con for ()		Versión: 2.0
Descripción: Factorial del 5 con for		
numero	Tipo de variable: float	
Código:	<pre>function factor(pnumero){ numero = pnumero; let factorial = 1; let contador; for(contador = 1; numero >= contador; contador++){ factorial = factorial * contador; } return factorial; }</pre>	
Como Expresión:	index.html:12	
120		
Como parametro:	index.html:13	
120		



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función:
tabla del 5 que multiplique hasta el 5 con for
parametro()

Versión: 1.0

Descripción:
Tabla del 5 que multiplique hasta el 5 con for

numero Tipo de variable: float

Código:

```
let multiplicar;  
const tablaExp = function (pmultiplicar,){  
    multiplicar = pmultiplicar;  
    let contador = 0;  
    let numero = 5;  
    let resultado;  
    let inf = "";  
  
    for(contador=1; numero >= contador; contador ++ ){  
        resultado = multiplicar * contador;  
  
        inf += `${multiplicar} x ${contador} = ${resultado}\n`;  
    }  
    return inf;  
}
```

Nombre de la función:
tabla del 5 que multiplique hasta el 5 con for ()

Versión: 2.0

Descripción:
Tabla del 5 que multiplique hasta el 5 con for

numero Tipo de variable: float

Código:

```
function tabla(pmultiplicar,){  
    multiplicar = pmultiplicar;  
    let contador = 0;  
    let numero = 5;  
    let resultado;  
    let inf = "";  
  
    for(contador=1; numero >= contador; contador ++ ){  
        resultado = multiplicar * contador;  
  
        inf += `${multiplicar} x ${contador} = ${resultado}\n`;  
    }  
    return inf;  
}
```

Como Expresión:

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

[index.html:12](#)

Como parametro:

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

[index.html:13](#)



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función:

tabla del 9 que multiplique hasta el 5 si son pares o impares con for parametro()

Versión: 1.0

Descripción:

Tabla del 9 que multiplique hasta el 5 si son pares o impares con for

numero

Tipo de variable: float

Código:

```
let tablasMultiplicar;  
const tablaExp = function (ptablasMultiplicar){  
    tablasMultiplicar = ptablasMultiplicar;  
    let numero;  
    let contador;  
    let resultados="";  
    let resultado;  
    numero = 5;  
    contador = 0;  
    resultados = "";  
  
    for(contador=1; numero >= contador; contador ++ ){  
        resultado = tablasMultiplicar * contador;  
  
        resultados += `${tablasMultiplicar} x ${contador} = ${resultado}  
        if(resultado %2==0){  
            resultados += "par\n";  
        }else{  
            resultados += "impar\n"  
        }  
    }  
    return resultados;  
}
```



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función:

tabla del 9 que multiplique hasta el 5 si son pares o impares con for ()

Versión: 2.0

Descripción:

Tabla del 9 que multiplique hasta el 5 si son pares o impares con for

numero

Tipo de variable: float

Código:

```
function tabla(ptablasMultiplicar){
    tablasMultiplicar = ptablasMultiplicar;
    let numero;
    let contador;
    let resultados="";
    let resultado;

    numero = 5;
    contador = 0;
    resultados = "";

    for(contador=1; numero >= contador; contador ++ ){
        resultado = tablasMultiplicar * contador;

        resultados += `${tablasMultiplicar} x ${contador} = ${resultado} `
        if(resultado %2==0){
            resultados += "par\n";
        }else{
            resultados += "impar\n"
        }
    }
    return resultados;
}
```

Como Expresión:

[index.html:12](#)

9 x 1 = 9 impar
9 x 2 = 18 par
9 x 3 = 27 impar
9 x 4 = 36 par
9 x 5 = 45 impar

Como parametro:

[index.html:13](#)

9 x 1 = 9 impar
9 x 2 = 18 par
9 x 3 = 27 impar
9 x 4 = 36 par
9 x 5 = 45 impar



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función:
tablas de multiplicar de 1 hasta el 5 y multiplique hasta el 5 si son pares o impares con for paramatro()

Versión: 1.0

Descripción:
tablas de multiplicar de 1 hasta el 5 y multiplique hasta el 5 si son pares o impares con for

numero

Tipo de variable: float

Código:

```
let numero;
const tablaExp =function(pnumeros){
    numero = pnumeros;
    let tabla;
    let contador;
    let par = 0;
    let impar = 0;
    let resultado;
    let mensaje = "";
    for(tabla = 1; numero >= tabla; tabla++){
        for(contador=1; numero >= contador; contador ++ ){
            resultado = tabla * contador;
            mensaje += `\n ${tabla} x ${contador} = ${resultado}`;

            if(resultado % 2==0){
                par = par + 1;
                mensaje += " Buzz\n"
            }else{
                impar = impar + 1;
                mensaje += " Bass\n"
            }
        }
        mensaje += "\n";
    }
    return mensaje + "total de pares: "+par+"\n"+
        "total de impares "+impar;
}
```




Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función:

tablas de multiplicar de 1 hasta el 5 y multiplique hasta el 5 si son pares o impares con for()

Versión: 2.0

Descripción:

tablas de multiplicar de 1 hasta el 5 y multiplique hasta el 5 si son pares o impares con for

numero

Tipo de variable: float

Código:

```
function tabla(pnumeros){
  numero = pnumeros;
  let tabla;
  let contador;
  let par = 0;
  let impar = 0;
  let resultado;
  let mensaje = "";

  for(tabla = 1; numero >= tabla; tabla++){
    for(contador=1; numero >= contador; contador ++ ){
      resultado = tabla * contador;
      mensaje += `\n ${tabla} x ${contador} = ${resultado}`;
      if(resultado % 2==0){
        par = par + 1;
        mensaje += " Buzz\n"
      }else{
        impar = impar + 1;
        mensaje += " Bass\n"
      }
    }
    mensaje += "\n";
  }
  return mensaje + "total de pares: "+par+"\n"+
    "total de impares "+impar;
}
```

Como Expresion
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

total de pares: 16
total de impares 9

Como parametro
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

total de pares: 16
total de impares 9



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: tablas de multiplicar de 1 hasta el 5 y multiplique hasta el 5 si son pares o impares con while paramatro()		Versión: 1.0
Descripción: tablas de multiplicar de 1 hasta el 5 y multiplique hasta el 5 si son pares o impares con while		
numero	Tipo de variable: float	

Código:

```
let numero;
const tablasExp = function (pnumero){
    numero = pnumero;
    let tabla;
    let contador;
    let par;
    let impar;
    let resultado;
    let mensaje= "";
    tabla = 0;
    par = 0;
    impar = 0;
    while (tabla < numero) {
        contador = 0;
        tabla = tabla + 1;
        while (contador < numero) {
            contador = contador + 1;
            resultado = tabla * contador;
            mensaje += `\n ${tabla} x ${contador} = ${resultado}`;
            if (resultado % 2 == 0) {
                par = par + 1;
                mensaje += " Buzz\n";
            } else {
                impar = impar + 1;
                mensaje += " Bass \n";
            }
        }
        mensaje += "\n"
    }
    return mensaje +"total de pares: "+par+"\n"+
        "total de impares "+impar;
}
```



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función:

tablas de multiplicar de 1 hasta el 5 y multiplique hasta el 5 si son pares o impares con while()

Versión: 2.0

Descripción:

tablas de multiplicar de 1 hasta el 5 y multiplique hasta el 5 si son pares o impares con while

numero

Tipo de variable: float

Código:

```
function tablas(pnumero){
    numero = pnumero;
    let tabla;
    let contador;
    let par;
    let impar;
    let resultado;
    let mensaje= "";
    tabla = 0;
    par = 0;
    impar = 0;
    while (tabla < numero) {
        contador = 0;
        tabla = tabla + 1;
        while (contador < numero) {
            contador = contador + 1;
            resultado = tabla * contador;
            mensaje += ` \n ${tabla} x ${contador} = ${resultado}`;
            if (resultado % 2 == 0) {
                par = par + 1;
                mensaje += " Buzz\n";
            } else {
                impar = impar + 1;
                mensaje += " Bass \n";
            }
        }
        mensaje += "\n"
    }
    return mensaje +"total de pares: "+par+"\n"+
        "total de impares "+impar;
}
```

Como Expresion
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

total de pares: 16

total de impares 9

Como parametro
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

total de pares: 16

total de impares 9



Tecnólogo en Análisis y Desarrollo de
Software
Ficha

Funciones JS