

Tema:
Manual de Funciones

Instructor:
Andrés Moreno Collazos

Aprendiz:

Ejercicios de JS normales

Nombre de la función: saludo (<i>psaludar</i>)		Versión: 1.0
Descripción: Imprimir “Hola Mundo” dentro de una variable.		
saludar	Tipo de variable: saludar (string)	
Código: <pre>function saludo(psaludar){ let saludar = psaludar; return saludar + " Parametros "; }</pre>		

Nombre de la función: saludoExp(psaludar)		Versión: 1.1
Descripción: Imprimir "Hola Mundo" dentro de una variable.		
saludo	Tipo de variable: saludar (string)	
Código: <pre>const saludoExp=function(psaludar){ let saludar = psaludar; return saludar + " Expresion "; }</pre>		

2.

Nombre de la función: suma(<i>pnumeroUno</i> , <i>pnumeroDos</i>)		Versión: 2.0
Descripción: Imprimir la suma de dos números.		
numeroUno numeroDos sumar	Tipo de variable: numeroUno(int) numeroDos(int) sumar(int)	
Código: <pre>function suma(pnumeroUno, pnumeroDos){ let numeroUno = pnumeroUno;</pre>		

```

    let numeroDos = pnumeroDos;
    let sumar;
    sumar = numeroUno + numeroDos
    return sumar;
}

```

Nombre de la función: sumarExp		Versión: 2.1
Descripción: Imprimir la suma de dos números.		
sumar	Tipo de variable: numeroUno(int) numeroDos(int) sumar(int)	
Código: <pre>const sumaExp = function(pnumeroUno, pnumeroDos){ let numeroUno = pnumeroUno; let numeroDos = pnumeroDos; let sumar; sumar = numeroUno + numeroDos; return sumar; }</pre>		

Nombre de la función: operacion(<i>poperador</i> , <i>pnumeroUno</i> , <i>pnumeroDos</i>), multiplicacion(<i>pnumeroUno</i> , <i>pnumeroDos</i>), division(<i>pnumeroUno</i> , <i>pnumeroDos</i>), resta(<i>pnumeroUno</i> , <i>pnumeroDos</i>)		Versión: 3.0
Descripción: imprimir la resta, la multiplicacion y la division de dos numeros.		
operador numeroUno numeroDos restar multiplicar divisor operacionResultado	Tipo de variable: operador (string) numeroUno(int) numeroDos(int) restar(int) multiplicar(int) divisor(float) operacionResultado(Alfanumerico)	

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

    restar = pnumeroUno - pnumeroDos;
    return restar;
}

function multiplicacion(pnumeroUno, pnumeroDos){
    let multiplicar;
    numeroUno = pnumeroUno;
    numeroDos = pnumeroDos;

    multiplicar = pnumeroUno * pnumeroDos;
    return multiplicar;
}

function division(pnumeroUno, pnumeroDos){
    let divisor;
    numeroUno = pnumeroUno;
    numeroDos = pnumeroDos;

    divisor = pnumeroUno / pnumeroDos;
    return divisor;
}

function operacion(poperador, pnumeroUno, pnumeroDos){
    let operador = poperador;
    numeroUno = pnumeroUno;
    numeroDos = pnumeroDos;
    if (operador == "resta") {
        operacionResultado = resta(pnumeroUno, pnumeroDos);
    }else if (operador == "multiplicacion") {
        operacionResultado = multiplicacion(pnumeroUno, pnumeroDos);
    }else if (operador == "division") {
        operacionResultado = division(pnumeroUno, pnumeroDos);
    }else{
        operacionResultado = "Error!!! no se puede operar";
    }
    return operacionResultado;
}
```

Nombre de la función: operacionExp, restaExp, multiplicacionExp, divisionExp	Versión: 3.1
Descripción: imprimir la resta, la multiplicacion y la division de dos numeros.	
operador numeroUno numeroDo restar multiplicar divisor operacionResultado	Tipo de variable: operador (string) pnumeroUno(int) pnumeroDos(int) restar(int) multiplicar(int) divisor(float) operacionResultado(Alfanumerico)
<pre> const restaExp=function(pnumeroUno, pnumeroDos){ let restar; numeroUno = pnumeroUno; numeroDos = pnumeroDos; restar = pnumeroUno - pnumeroDos; return restar; } const multiplicacionExp=function(pnumeroUno, pnumeroDos){ let multiplicar; numeroUno = pnumeroUno; numeroDos = pnumeroDos; multiplicar = pnumeroUno * pnumeroDos; return multiplicar; } const divisionExp=function(pnumeroUno, pnumeroDos){ let divisor; numeroUno = pnumeroUno; numeroDos = pnumeroDos; divisor = pnumeroUno / pnumeroDos; return divisor; } const operacionExp = function(poperador, pnumeroUno, pnumeroDos){ let operador = poperador; numeroUno = pnumeroUno; numeroDos = pnumeroDos; if (operador == "resta") { operacionResultado = restaExp(pnumeroUno, pnumeroDos); }else if (operador == "multiplicacion") { operacionResultado = multiplicacionExp(pnumeroUno, pnumeroDos); } } </pre>	

```

    }else if (operador == "division") {
        operacionResultado = divisionExp(pnumeroUno, pnumeroDos);
    }else{
        operacionResultado = "Error!!! no se puede operar";
    }
    return operacionResultado;
}

```

Nombre de la función: porcentaje(pnum1)		Versión: 4.0
Descripción: Imprimir el porcentaje de un número.		
porcen num	Tipo de variable: porcen(float) num1(int)	
Código: <pre>function porcentaje(pnum1){ let porcen; let num1 = pnum1; porcen = num1 / 100; return porcen; }</pre>		

Nombre de la función: porcentajeExp		Versión: 4.1
Descripción: Imprimir el porcentaje de un número.		
porcen num	Tipo de variable: porcen (float) num1 (int)	
Código: <pre>const porcentajeExp = function (pnum1){ let porcen;</pre>		

```

let num1 = pnum1;

porcen = num1 / 100;
return porcen;
}

```

Nombre de la función: promedio(pnota1,p nota2,p nota3)		Versión: 5.0
Descripción: Imprimir el promedio de 3 notas.		
nota1 nota2 nota3 prom	Tipo de variable: nota1(int) nota2(int) noat3(int) prom(float)	
Código: <pre>function promedio(pnota1, pnota2, pnota3){ let prom; let nota1 = pnota1; let nota2 = pnota2; let nota3 = pnota3; prom = (nota1+nota2+nota3)/3; return prom; }</pre>		

Nombre de la función: promediosExp		Versión: 5.1
Descripción: Imprimir el promedio de 3 notas.		
nota1 nota2 nota3 prom	Tipo de variable: nota1(int) nota2(int) noat3(int) prom(float)	

Código:	<pre>const promedioExp = function(pnota1, pnota2, pnota3){ let prom; let nota1 = pnota1; let nota2 = pnota2; let nota3 = pnota3; prom = (nota1+nota2+nota3)/3; return prom; }</pre>

Nombre de la función: suma(pnota,pporcenNota)	Versión: 6.0
Descripción: a. El porcentaje de 3 notas, la nota 1 =30%, la nota2=30% y la nota3 =40%. b. Sumar el resultado de los porcentajes de las notas.	
suma nota1 nota2 nota3 porcenNota suma2 nota4 nota5 nota6	Tipo de variable: suma(float) not1(int) not2(int) not3(int) porcenNota(float) suma2float) nota4(int) nota5(int) nota6(int)
Código:	<pre>function sumar(pnota, pporcenNota){ let suma; let nota = pnota; let porcenNota = pporcenNota; suma = nota * porcenNota return suma; }</pre>


```

nota1 = sumar( 5, 0.3);
nota2 = sumar( 3.5, 0.3);
nota3 = sumar( 4, 0.4);

```

```

let suma = nota1+nota2+nota3

```

Nombre de la función: sumarExp

Versión: 6.1

Descripción:

- El porcentaje de 3 notas, la nota 1 =30%, la nota2=30% y la nota3 =40%.
- Sumar el resultado de los porcentajes de las notas.

suma	Tipo de variable:
nota1	suma(float)
nota2	not1(int)
nota3	not2(int)
porcenNota	not3(int)
suma2	porcenNota(float)
nota4	suma2float)
nota5	nota4(int)
nota6	nota5(int)
	nota6(int)

Código:

```

const sumarExp = function( pnota, pporcenNota){
    let suma2;
    let nota = pnota;
    let porcenNota = pporcenNota;

    suma2 = nota * porcenNota
    return suma2;
}

```

```

nota4 = sumar( 5, 0.3);
nota5 = sumar( 3.5, 0.3);
nota6 = sumar( 4, 0.4);

```

```

let suma2 = nota4+nota5+nota6

```

7.

Nombre de la función: áreas(operador,pbase,paltura,plado)		Versión: 7.0
Descripción: realizar un diagrama de flujo que imprima las áreas de las figuras geométricas. Teniendo en cuenta: Cuadrado = lado x lado. Rectángulo= base x altura Triangulo = (base x altura) /2		
areas operador base altura lado	Tipo de variable: areas(Alfanumerica) operador(Alfanumerica) base(int) altura(int) lado(int)	
<pre>function areas(operador, pbase, paltura, plado){ let area if (operador=="rectangulo") { area = pbase*paltura }else if (operador=="triangulo"){ area = (pbase*paltura)/2 } else if (operador=="cuadrado"){ area = plado*plado }else{ area ="Error!!! no se puede operar" } return area }</pre>		

Nombre de la función: areasExp(operador,pbase,paltura,plado)		Versión: 7.1
Descripción: realizar un diagrama de flujo que imprima las áreas de las figuras geométricas. Teniendo en cuenta: Cuadrado = lado x lado. Rectángulo= base x altura Triangulo = (base x altura) /2		
areas operador base altura lado	Tipo de variable: areas(Alfanumerica) operador(Alfanumerica) base(int) altura(int)	

	lado(int)
<pre>const areasExp = function(operador, pbase, paltura, plado){ let area if (operador=="rectangulo") { area = pbase*paltura }else if (operador=="triangulo"){ area = (pbase*paltura)/2 } else if (operador=="cuadrado"){ area = plado*plado }else{ area ="Error!!! no se puede operar" } return area }</pre>	

Nombre de la función: pagoPersona(valorDia,diasTrabajados)		Versión: 8.0
Descripción: un diagrama de flujo que imprima el pago total de una persona, sabiendo que el sueldo es igual a los días por el valor de los días, imprimir salud, pensión, y arl sabiendo que la suma de la salud, pensión y arl se descuentan del sueldo de la persona.		
sueldo arl salud pensión descuentos pagoTotal	Tipo de variable: sueldo(int) arl(int) salud(int) pensión(int) descuentos(int) pagoTotal(int)	
<pre>function pagoPersona(valorDia, diasTrabajados) { let sueldo</pre>		

```

let arl
let salud
let pension
let descuentos
let pagoTotal

sueldo = valorDia* diasTrabajados
salud= sueldo * 0.12
pension = sueldo *0.16
arl =sueldo * 0.052
descuentos = salud+ pension+ arl
pagoTotal= sueldo - descuentos

return [pagoTotal+" \n"+"El sueldo de la persona es: "+sueldo+" \n"+
"EL descuento de salud es :"+salud+" \n"+"EL descuento de pension
es :"+pension+" \n"+
"EL descuento de arl es :"+arl+" \n"+"EL total de descuento
es:"+descuentos]

}

```

Nombre de la función: pagoPersona(valorDia,diasTrabajados)	Versión: 8.1
Descripción: un diagrama de flujo que imprima el pago total de una persona, sabiendo que el sueldo es igual a los días por el valor de los días, imprimir salud, pensión, y arl sabiendo que la suma de la salud, pensión y arl se descuentan del sueldo de la persona.	
sueldo arl salud pensión descuentos pagoTotal	Tipo de variable: sueldo(int) arl(int) salud(int) pensión(int) descuentos(int) pagoTotal(int)
<pre> const pagoPersonaExp = function(valorDia, diasTrabajados) { let sueldo let arl let salud let pension let descuentos let pagoTotal </pre>	

```

    sueldo = valorDia* diasTrabajados
    salud= sueldo * 0.12
    pension = sueldo *0.16
    arl =sueldo * 0.052
    descuentos = salud+ pension+ arl
    pagoTotal= sueldo - descuentos

    return [pagoTotal+"\n"+"El sueldo de la persona es: "+sueldo+"\n"+
            "EL descuento de salud es :"+salud+"\n"+"EL descuento de pension
es :"+pension+"\n"+
            "EL descuento de arl es :"+arl+"\n"+"EL total de descuento
es:"+descuentos]

}

```

Nombre de la función: edades(pedad)		Versión: 9.0
Descripción: Imprimir si es mayor o menor de edad.		
edad	Tipo de variable: edad(int)	
Código: <pre>function edades(pedad){ let edad = pedad; if (edad>=18){ edad = "Es mayor de edad " }else{ edad = "Es menor de edad " } return edad; }</pre>		

Nombre de la función: edadesExp(pedad)		Versión: 9.1
Descripción: Imprimir si es mayor o menor de edad.		

edad	Tipo de variable: edad(int)
Código: <pre>const edadesExp = function (pedad){ let edad = pedad; if (edad >= 18){ edad = "Es mayor de edad " }else{ edad = "Es menor de edad " } return edad; }</pre>	

Nombre de la función: edades(p <i>anioNac</i> , <i>panioAct</i>)		Versión: 10.0
Descripción: Calcular la edad de una persona, imprimirla edad y si es mayor o menor de edad.		
edad	Tipo de variable: edad(int) panioAct(int) panioNac(int)	
Código: <pre>function edades(panioNac, panioAct){ edad = panioAct - panioNac; if (edad>17){ edad = edad + " Es mayor de edad " }else{ edad = edad + " Es menor de edad " } return edad; }</pre>		

Nombre de la función: edadesExp(p <i>anioNac</i> , <i>panioAct</i>)		Versión: 10.1
Descripción: Calcular la edad de una persona, imprimirla edad y si es mayor o menor de edad.		
edad	Tipo de variable: edad(int) panioAct(int)	

	panioNac(int)
Código: <pre>const edadesExp = function(panioNac, panioAct){; edad = panioAct - panioNac; if (edad>17){ edad = edad + " Es mayor de edad " }else{ edad = edad + " Es menor de edad " } return edad; }</pre>	

Nombre de la función: numeros(pnumUno,pnumDos)		Versión: 11.0
Descripción: Imprimir el número mayor, el menor o si son iguales los dos.		
numero numUno numDos	Tipo de variable: numDos(Int) numUno(Int) numero(string)	
Código: <pre>function numeros(pnumUno, pnumDos){ let numero; let numUno = pnumUno; let numDos = pnumDos; if (numUno==numDos){ numero = "Son iguales" } else if (numUno>numDos){ numero = "Numero 1 es mayor" } else{ numero = "Numero 2 es mayor" } return numero; }</pre>		

Nombre de la función: numerosExp(pnumUno,pnumDos)		Versión: 11.1
Descripción: Imprimir el número mayor, el menor o si son iguales los dos.		
numero	Tipo de variable:	

numUno numDos	numDos(Int) numUno(Int) numero(string)
<pre>const numerosExp = function(pnumUno, pnumDos){ let numero; let numUno = pnumUno; let numDos = pnumDos; if (numUno==numDos){ numero = "Son iguales" } else if (numUno>numDos){ numero = "Numero 1 es mayor" } else{ numero = "Numero 2 es mayor" } return numero; }</pre>	

Nombre de la función: areaMayor(pladCuad1,pladCuad2,pladCuad3)		Versión: 12.0
Descripción: Calcular el área de 3 cuadrado e imprimir si las áreas son iguales o mayor.		
pladCuad1 pladCuad2 pladCuad3 cuad1 cuad2 cuad3	Tipo de variable: pladCuad1(int) pladCuad2 (int) pladCuad3(int) cuad1(int) cuad2(int) cuad3(int)	
<pre>function areaMayor(pladCuad1, pladCuad2, pladCuad3){ let cuad1 = pladCuad1; let cuad2 = pladCuad2; let cuad3 = pladCuad3; cuad1=pladCuad1* pladCuad1 cuad2=pladCuad2* pladCuad2 cuad3=pladCuad3* pladCuad3</pre>		


```

    if (cuad1==cuad2 && cuad1==cuad3){
        return ("Las areas de los cuadrados son iguales")
    }
    else{
        if (cuad1>cuad2 && cuad1>cuad3){
            return ("El area del cuadrado uno es mayor")
        }
        else{
            if (cuad2>cuad3 && cuad2>cuad1){
                return("El area del cuadrado dos es mayor")
            }
            else{
                return("El area del cuadrado 3 es mayor")
            }
        }
    }
}

```

Nombre de la función: areaMayor(pladCuad1,pladCuad2,pladCuad3)	Versión: 12.1
Descripción: Calcular el área de 3 cuadrado e imprimir si las áreas son iguales o mayor.	
pladCuad1 pladCuad2 pladCuad3 cuad1 cuad2 cuad3	Tipo de variable: pladCuad1(int) pladCuad2 (int) pladCuad3(int) cuad1(int) cuad2(int) cuad3(int)
<pre> const areaMayorExp = function(pladCuad1, pladCuad2, pladCuad3){ let cuad1 = pladCuad1; let cuad2 = pladCuad2; let cuad3 = pladCuad3; cuad1=pladCuad1* pladCuad1 cuad2=pladCuad2* pladCuad2 cuad3=pladCuad3* pladCuad3 if (cuad1==cuad2 && cuad1==cuad3){ return ("Las areas de los cuadrados son iguales") } </pre>	

```

else{
    if (cuad1>cuad2 && cuad1>cuad3){
        return ("El area del cuadrado 1 es mayor")
    }
    else{
        if (cuad2>cuad3 && cuad2>cuad1){
            return("El area del cuadrado dos es mayor")
        }
        else{
            return("El area del cuadrado 3 es mayor")
        }
    }
}
}
}
}

```

<p>Nombre de la función: mayor(anionac1, anionac2, anionac3), anios(anionac1), anios2(anionac2), anios3(anionac3)</p>	<p>Versión: 13.0</p>
<p>Descripción: Realizar un diagrama de flujo que calcule la edad de 3 personas e imprimir si cada persona es mayor de edad, calcular e imprimir el mismo diagrama de promedio de las 3 edades y si el promedio de edades si es mayor de edad.</p>	
<p>aninac1 aninac2 aninac3 edad edad2 edad3 eda1 eda2 eda3 promedio prom</p>	<p>Tipo de variable: aninac1(int) aninac2(int) aninac3(int) edad1(int) edad2(int) edad3(int) eda1(string) eda2(string) eda3(string) promedio(int) prom(string)</p>
<pre> let edad1; let edad2; let edad3; let eda1; let eda2; let eda3; function mayor(anionac1, anionac2, anionac3) { let aninac1=anionac1; </pre>	

```

    let aninac2=anionac2;
    let aninac3=anionac3;
    let edad1;
    let edad2;
    let edad3;
    let promedio=anios(anionac1)+anios2(anionac2)+anios3(anionac3);
    let prom
    if (promedio>17) {
        prom="El promedio cumple con la mayoría de edad";
    } else {
        prom= "El promedio no cumple con la mayoría de edad";
    }
    return prom
}
function anios(anionac1,) {
    let aninac1=anionac1;
    let edad1 = 2024- aninac1;
    if (edad1>17) {
        eda1 ="Es mayor de edad";
    } else{
        eda1="Es menor de edad";
    }
    console.log(eda1);
    return edad1
}
function anios2(anionac2,) {
    let aninac2=anionac2;
    let edad2 = 2024- aninac2;
    if (edad2>17) {
        eda2 ="Es mayor de edad";
    } else{
        eda2="Es menor de edad";
    }
    console.log(eda2);
    return edad2
}
function anios3(anionac3,) {
    let aninac3=anionac3;
    let edad3 = 2024- aninac3;
    if (edad3>17) {
        eda3 ="Es mayor de edad";
    } else{
        eda3="Es menor de edad";
    }
}

```

```

    console.log(eda3);
    return edad3
}

```

Nombre de la función: mayor2,
anios4, anios5, anios6

Versión: 13.1

Descripción:

Realizar un diagrama de flujo que calcule la edad de 3 personas e imprimir si cada persona es mayor de edad, calcular e imprimir el mismo diagrama de promedio de las 3 edades y si el promedio de edades si es mayor de edad.

aninac1	Tipo de variable:
aninac2	aninac4(int)
aninac3	aninac5(int)
edad	aninac6(int)
edad2	edad4(int)
edad3	edad5(int)
eda1	edad6(int)
eda2	eda4(string)
eda3	eda5(string)
promedio	eda6(string)
prom	promedio(int)
	prom(string)

Código:

```

const mayor2=function( anionac1, anionac2, anionac3) {
    let aninac1=anionac1;
    let aninac2=anionac2;
    let aninac3=anionac3;
    let edad1;
    let edad2;
    let edad3;
    let promedio=anios4( anionac1 )+anios5( anionac2 )+anios6( anionac3);
    let prom
    if (promedio>17) {
        prom="El promedio cumple con la mayoría de edad";
    } else {
        prom= "El promedio no comple con la mayoría de edad";
    }
    return prom
}

const anios4=function( anionac4,) {
    let aninac4=anionac4;
    let edad4 = 2024- aninac4;
    let eda4;
    if (edad4>17) {

```

```

        eda4 ="Es mayor de edad";
    } else{
        eda4="Es menor de edad";
    }
    console.log(eda4);
    return edad4;
}

const anios5=function(anionac5,) {
    let aninac5=anionac5;
    let edad5 = 2024- aninac5;
    let eda5;
    if (edad5>17) {
        eda5 ="Es mayor de edad";
    } else{
        eda5="Es menor de edad";
    }
    console.log(eda5);
    return edad5
}

const anios6=function(anionac6,) {
    let aninac6=anionac6;
    let edad6 = 2024- aninac6;
    let eda6;
    if (edad6>17) {
        eda6 ="Es mayor de edad";
    } else{
        eda6="Es menor de edad";
    }
    console.log(eda6);
    return edad6;
}

```

Nombre de la función:

salarioExp(pDiasTrabajados,pValorDia)
subTransExp(pDiasTrabajados,pValorDia)
saludExp(pDiasTrabajados,pValorDia)
pensionExp(pDiasTrabajados,pValorDia)
arlExp(pDiasTrabajados,pValorDia)
retencionExp(pDiasTrabajados,pValorDia)
deduccionesExp(pDiasTrabajados,pValorDia)

Versión: 14.0

pagoFinalExp(pDiasTrabajados,pValorDia)	
<p>Descripción:</p> <p>Calcule el pago total del sueldo de una persona donde calcular e imprima lo siguiente:</p> <ol style="list-style-type: none"> salario de la persona. si la persona gana más de dos salarios mínimos se suma a su sueldo, el subsidio de transporte de lo contrario sumara. Calcular la salud, pensión y arl sabiendo que: salud = salario x 0,12 pensión= salario x 0,16 arl = salario x 0,052 Si la persona gana mas de 4 salarios mínimos debe hacer una retención del 0,04 de un salario. sumar los deducibles que son salud, pensión y arl, y restar el deducible al salario de la persona. calcular e imprimir el total de pagar 	
<div>diasTrabajados</div> <div>valorDia</div> <div>salarioPersona</div> <div>salarioMinimo</div> <div>salarioTrans</div> <div>subTransporte</div> <div>pagoSalud</div> <div>pagoPension</div> <div>pagoArl</div> <div>pagoDeducibl</div> <div>pagoFinal</div>	<div>Tipo de variable:</div> <div>diasTrabajados(int)</div> <div>valorDia(int)</div> <div>salarioPersona(int)</div> <div>salarioMinimo(int)</div> <div>salarioTrans(int)</div> <div>subTransporte(int)</div> <div>pagoSalud(float)</div> <div>pagoPension(float)</div> <div>pagoArl(float)</div> <div>pagoDeducible(int)</div> <div>pagoFinal(int)</div>
<pre>function salarioExp (pDiasTrabajados, pValorDia) { let salarioPersona salarioPersona = pDiasTrabajados * pValorDia return salarioPersona } function subTransExp (pDiasTrabajados, pValorDia) { let salarioMinimo = 1600000 let salarioTrans = salarioExp (pDiasTrabajados, pValorDia) let subTransporte if (salarioTrans <= 2 * salarioMinimo) { subTransporte = 162000 } else{ subTransporte = 0 } }</pre>	

```
    return subTransporte
}
```

```
function saludExp (pDiasTrabajados, pValorDia){
    let pagoSalud
    pagoSalud = salarioExp(pDiasTrabajados, pValorDia) * 0.12

    return pagoSalud
}
```

```
function pensionExp(pDiasTrabajados, pValorDia){
    let pagoPension
    pagoPension =salarioExp(pDiasTrabajados, pValorDia) * 0.16

    return pagoPension
}
```

```
function arlExp(pDiasTrabajados, pValorDia){
    let pagoArl
    pagoArl = salarioExp(pDiasTrabajados, pValorDia) * 0.052

    return pagoArl
}
```

```
function retencionExp(pDiasTrabajados, pValorDia){
    let reten
    let salarioMinimo = 1600000
    reten = salarioExp(pDiasTrabajados, pValorDia)

    if (reten>4*salarioMinimo) {
        descuento = reten * 0.04
    }
    else{
        descuento = 0
    }

    return descuento
}
```

```
function deduccionesExp (pDiasTrabajados, pValorDia){
    let descuentos
    descuentos = saludExp(pDiasTrabajados, pValorDia) +
pensionExp(pDiasTrabajados, pValorDia) + arlExp(pDiasTrabajados, pValorDia)
```

```
+ retencionExp( pDiasTrabajados, pValorDia)
```

```
    return descuentos  
}
```

```
function pagoFinalExp( pDiasTrabajados, pValorDia){
```

```
    let pagoTotal
```

```
    pagoTotal= salarioExp( pDiasTrabajados, pValorDia) -
```

```
    deducionesExp( pDiasTrabajados, pValorDia)
```

```
    return pagoTotal
```

```
}
```

Nombre de la función:

salarioExp(pDiasTrabajados,pValorDia)
subTransExp(pDiasTrabajados,pValorDia)
saludExp(pDiasTrabajados,pValorDia)
pensionExp(pDiasTrabajados,pValorDia)
arlExp(pDiasTrabajados,pValorDia)
retencionExp(pDiasTrabajados,pValorDia)
deducionesExp(pDiasTrabajados,pValorDia
)
pagoFinalExp(pDiasTrabajados,pValorDia)

Versión: 14.1

Descripción:

Calcule el pago total del sueldo de una persona donde calcular e imprima lo siguiente:

- salario de la persona.
- si la persona gana más de dos salarios mínimos se suma a su sueldo, el subsidio de transporte de lo contrario sumara.
- Calcular la salud, pensión y arl sabiendo que:
 $\text{salud} = \text{salario} \times 0,12$
 $\text{pensión} = \text{salario} \times 0,16$
 $\text{arl} = \text{salario} \times 0,052$
- Si la persona gana mas de 4 salarios mínimos debe hacer una retención del 0,04 de un salario.
- sumar los deducibles que son salud, pensión y arl, y restar el deducible al salario de la persona.
- calcular e imprimir el total de pagar

diasTrabajados
valorDia
salarioPersona
salarioMinimo
salarioTrans
subTransporte

Tipo de variable:
diasTrabajados(int)
valorDia(int)
salarioPersona(int)
salarioMinimo(int)
salarioTrans(int)

pagoSalud	subTransporte(int)
pagoPension	pagoSalud(float)
pagoArl	pagoPension(float)
pagoDeducibl	pagoArl(float)
pagoFinal	pagoDeducible(int)
	pagoFinal(int)

```
const salud = function (pDiasTrabajados, pValorDia){  
  let pagoSalud  
  pagoSalud = salario(pDiasTrabajados, pValorDia) * 0.12  
  
  return pagoSalud  
}  
  
const pension = function(pDiasTrabajados, pValorDia){  
  let pagoPension  
  pagoPension =salario(pDiasTrabajados, pValorDia) * 0.16  
  
  return pagoPension  
}  
  
const arl = function(pDiasTrabajados, pValorDia){  
  let pagoArl  
  pagoArl = salario(pDiasTrabajados, pValorDia) * 0.052  
  
  return pagoArl  
}  
  
const retencion = function(pDiasTrabajados, pValorDia){  
  let reten  
  let salarioMinimo = 1600000  
  reten = salario(pDiasTrabajados, pValorDia)  
  
  if (reten>4*salarioMinimo) {  
    descuento = reten * 0.04  
  }  
  else{  
    descuento = 0  
  }  
  
  return descuento  
}  
  
const deducciones = function (pDiasTrabajados, pValorDia){  
  let descuentos
```

```

    descuentos = salud( pDiasTrabajados, pValorDia) +
pension( pDiasTrabajados, pValorDia) + arl( pDiasTrabajados, pValorDia) +
retencion( pDiasTrabajados, pValorDia)

    return descuentos
}

const pagoFinal = function( pDiasTrabajados, pValorDia){
    let pagoTotal
    pagoTotal= salario( pDiasTrabajados, pValorDia) -
deducciones( pDiasTrabajados, pValorDia)

    return pagoTotal
}

function salarioExp ( pDiasTrabajados, pValorDia) {
    let salarioPersona
    salarioPersona = pDiasTrabajados * pValorDia

    return salarioPersona
}

function subTransExp ( pDiasTrabajados, pValorDia) {
    let salarioMinimo = 1600000
    let salarioTrans = salarioExp ( pDiasTrabajados, pValorDia)
    let subTransporte

    if ( salarioTrans<=2*salarioMinimo) {
        subTransporte = 162000
    }
    else{
        subTransporte = 0
    }

    return subTransporte
}

function saludExp ( pDiasTrabajados, pValorDia){
    let pagoSalud
    pagoSalud = salarioExp( pDiasTrabajados, pValorDia) * 0.12

    return pagoSalud
}

```

```
function pensionExp( pDiasTrabajados, pValorDia){  
    let pagoPension  
    pagoPension =salarioExp( pDiasTrabajados, pValorDia) * 0.16  
  
    return pagoPension  
}
```

```
function arlExp( pDiasTrabajados, pValorDia){  
    let pagoArl  
    pagoArl = salarioExp( pDiasTrabajados, pValorDia) * 0.052  
  
    return pagoArl  
}
```

```
function retencionExp( pDiasTrabajados, pValorDia){  
    let reten  
    let salarioMinimo = 1600000  
    reten = salarioExp( pDiasTrabajados, pValorDia)  
  
    if (reten>4*salarioMinimo) {  
        descuento = reten * 0.04  
    }  
    else{  
        descuento = 0  
    }  
  
    return descuento  
}
```

```
function deduccionesExp ( pDiasTrabajados, pValorDia){  
    let descuentos  
    descuentos = saludExp( pDiasTrabajados, pValorDia) +  
pensionExp( pDiasTrabajados, pValorDia) + arlExp( pDiasTrabajados, pValorDia)  
+ retencionExp( pDiasTrabajados, pValorDia)  
  
    return descuentos  
}
```

```
function pagoFinalExp( pDiasTrabajados, pValorDia){  
    let pagoTotal  
    pagoTotal= salarioExp( pDiasTrabajados, pValorDia) -  
deduccionesExp( pDiasTrabajados, pValorDia)  
  
    return pagoTotal  
}
```

```
}
```

Nombre de la función: nota1(nota), nota2(nota), nota3(nota), suma(sumaNotas)		Versión: 15.0
Descripción: Calcular tres notas e imprimir lo siguiente a. El 20 por ciento de la nota1 b. El 35 por ciento de la nota2 c. El 45 por ciento de la nota3 d. Sumar los porcentajes de las tres notas e imprimir los siguiente Si la suma del porcentaje es mayor a 4.5 es una nota superior Si la suma del porcentaje esta entre 4.5 y 3.5 la nota es buena Si la suma del porcentaje esta entre 3.5 y 3 la nota es media Si la suma de porcentaje es menor de 3 es una nota mala.		
calificacion porcenNota sumaNotas nota	Tipo de variable: califiaccion(int) porcenNota(float) suma(float) nota(Alfanumerico)	
<pre>function nota1(nota) { let calificacion calificacion = nota porceNota= calificacion * 0.2 return porceNota } function nota2(nota) { let calificacion calificacion = nota porceNota= calificacion * 0.35 return porceNota } function nota3(nota) { let calificacion calificacion = nota</pre>		

```

    porceNota= calificacion * 0.45

    return porceNota
}

function suma(sumaNotas) {
    let calificacion
    calificacion= sumaNotas

    if (calificacion>=4.5) {
        nota = "Es una nota superior: "
    }
    else{
        if (calificacion<=4.5 & calificacion>=3.5) {
            nota = "La nota es buena: "
        }
        else{
            if (calificacion<=3.5 & calificacion>=3){
                nota = "La nota es media: "
            }
            else{
                nota = "Es una nota mala: "
            }
        }
    }
    return nota
}

```

Nombre de la función: nota1(nota), nota2(nota), nota3(nota), suma(sumaNotas)	Versión: 15.1
Descripción: Calcular tres notas e imprimir lo siguiente <ol style="list-style-type: none"> El 20 por ciento de la nota1 El 35 por ciento de la nota2 El 45 por ciento de la nota3 Sumar los porcentajes de las tres notas e imprimir los siguiente Si la suma del porcentaje es mayor a 4.5 es una nota superior Si la suma del porcentaje esta entre 4.5 y 3.5 la nota es buena	

Si la suma del porcentaje esta entre 3.5 y 3 la nota es media
Si la suma de porcentaje es menor de 3 es una nota mala.

calificacion porcenNota sumaNotas nota	Tipo de variable: califiaccion(int) porcenNota(float) suma(float) nota(Alfanumerico)
---	--

```
const nota4 = function(nota) {  
  let calificacion  
  calificacion = nota
```

```
  porceNota= calificacion * 0.2  
  
  return porceNota  
}
```

```
const nota5 = function(nota) {  
  let calificacion  
  calificacion = nota
```

```
  porceNota= calificacion * 0.35  
  
  return porceNota  
}
```

```
const nota6 = function(nota) {  
  let calificacion  
  calificacion = nota
```

```
  porceNota= calificacion * 0.45  
  
  return porceNota  
}
```

```
const suma2 = function(sumaNotas) {  
  let calificacion  
  calificacion= sumaNotas
```

```
  if (calificacion>=4.5) {  
    nota = "Es una nota superior: "  
  }
```

```
  else{  
    if (calificacion<=4.5 & calificacion>=3.5) {
```

```

        nota = "La nota es buena: "
    }
    else{
        if ( calificacion<=3.5 & calificacion>=3){
            nota = "La nota es media: "
        }
        else{
            nota = "Es una nota mala: "
        }
    }
}
return nota
}

```

Nombre de la función: contar(pnumero)		Versión: 16.0
Descripción: Imprimir los números de 1 a 5.		
contar numero	Tipo de variable: contar(int) numero(int)	
Código: <pre>function contar(pnumero) { let numero = pnumero let contador = 1; let resultadoContar=""; while(contador<=numero){ resultadoContar+=contador; contador=contador+1 } return resultadoContar; }</pre>		

Nombre de la función: contarExp(pnumero)	Versión: 16.1
--	---------------

Descripción: Imprimir los números de 1 a 5.	
<div> <div>contar</div> <div>numero</div> </div>	<div>Tipo de variable:</div> <div> <div>contar(int)</div> <div>numero(int)</div> </div>
<pre>const contarExp = function(pnumero) { let numero = pnumero let contador = 1; let resultadoContar=""; while(contador<=numero){ resultadoContar+=contador; contador=contador+1 } return resultadoContar; }</pre>	

Nombre de la función: contar(pnumero)		Versión: 17.0
Descripción: Imprimir los números de 1 a 5.		
<div> <div>contar</div> <div>numero</div> </div>	<div>Tipo de variable:</div> <div> <div>contar(int)</div> <div>numero(int)</div> </div>	
<div>Código:</div> <pre>function contar(pnumero) { let contador let numero=pnumero let resultadoContar = "" for(contador=1; contador<=numero; contador++){ resultadoContar+=contador } return resultadoContar }</pre>		

Nombre de la función: contarExp(pnumero)	Versión: 17.1
--	---------------

Descripción: Imprimir los números de 1 a 5.	
<div> <div>contar</div> <div>numero</div> </div>	<div>Tipo de variable:</div> <div> <div>contar(int)</div> <div>numero(int)</div> </div>
<pre>const contarExp=function(pnumero) { let contador let numero=pnumero let resultadoContar = "" for(contador=1; contador<=numero; contador++){ resultadoContar+=contador } return resultadoContar }</pre>	

Nombre de la función: conta(<i>contando</i>)		Versión: 18.0
Descripción: Contar de 1 a 5 e imprimir pares e impares.		
<div> <div>contar</div> <div>numero</div> </div>	<div>Tipo de variable:</div> <div> <div>contar(int)</div> <div>numero(int)</div> </div>	
<div>Código:</div> <pre>function conta(contando) { let contar=contando; let numero=4; while (contar<=numero) { contar++ if (contar%2==0) { console.log(contar+" Es par"); } else { console.log(contar+" Es impar"); } } return "Fin" }</pre>		

--

Nombre de la función: conta(<i>contando</i>)	Versión: 18.1
Descripción: Contar de 1 a 5 e imprimir pares e impares.	
contar numero	Tipo de variable: contar(int) numero(int)
<pre>const contaExp = function(contando) { let contar=contando; let numero=4; while (contar<=numero) { contar++; if (contar%2==0) { console.log(contar+" Es par"); } else { console.log(contar+" Es impar"); } } return "Fin" }</pre>	

Nombre de la función: conta(<i>contando</i>)	Versión: 19.0
Descripción: Contar de 1 a 5 e imprimir pares e impares.	
contar numero	Tipo de variable: contar(int) numero(int)
<pre>function conta(contando){ let contar=contando; let numero=5; for (contar=1; contar<=numero; contar++) { if (contar%2==0) { console.log(contar+" Es par"); } else { console.log(contar+" Es impar"); } } return "Fin" }</pre>	

```
}
```

Nombre de la función: conta(contando)		Versión: 19.1
Descripción: Contar de 1 a 5 e imprimir pares e impares.		
contar numero	Tipo de variable: contar(int) numero(int)	
Código: <pre>const contaExp = function(contando){ let contar=contando; let numero=5; for (contar=1; contar<=numero; contar++) { if (contar%2==0) { console.log(contar+" Es par"); } else { console.log(contar+" Es impar"); } } return "Fin" }</pre>		

Nombre de la función: tabla(pmultiplicar,pnumero)		Versión: 20.0
Descripción: Realizar la tabla del 5 que multiplique hasta el 5 y debe imprimir los siguientes resultados: 5 x 1 = 5 5 x 2 = 10 5 x 3 = 15 5 x 4 = 20 5 x 5 = 25		
multiplicar numero result	Tipo de variable: multiplicar(int) numero (int) result(int)	
Código: function tabla(pmultiplicar, pnumero){ let multiplicar = pmultiplicar		

```

let numero = pnumero
let result

while(multiplicar<numero){
    multiplicar = multiplicar + 1
    result = multiplicar * numero
    console.log(numero + "x" + multiplicar + "=" + result);
}
return "Final"
}

```

Nombre de la función: tabla(pmultiplicar,pnumero)		Versión: 20.1
Descripción: Realizar la tabla del 5 que multiplique hasta el 5 y debe imprimir los siguientes resultados: 5 x 1 = 5 5 x 2 = 10 5 x 3 = 15 5 x 4 = 20 5 x 5 = 25		
multiplicar numero result	Tipo de variable: multiplicar(int) numero (int) result(int)	
Código: <pre>const tablaExp = function(pmultiplicar, pnumero){ let multiplicar = pmultiplicar let numero = pnumero let result while(multiplicar<numero){ multiplicar = multiplicar + 1 result = multiplicar * numero console.log(numero +"x"+ multiplicar +"="+ result); } return "Final" }</pre>		

Nombre de la función: tabla(pmultiplicar,pnumero)		Versión: 21.0
Descripción: Realizar la tabla del 5 que multiplique hasta el 5 y debe imprimir los siguientes resultados: 5 x 1 = 5 5 x 2 = 10 5 x 3 = 15 5 x 4 = 20 5 x 5 = 25		
multiplicar numero result		Tipo de variable: multiplicar(int) numero (int) result(int)
Código: <pre>function tabla(pmultiplicar, pnumero){ let multiplicar = pmultiplicar let numero = pnumero let result for(multiplicar = 1; multiplicar<=numero; multiplicar++){ result = multiplicar * numero console.log(numero + "x" + multiplicar + "=" + result) } return "Final" }</pre>		

Nombre de la función: tabla(pmultiplicar,pnumero)		Versión: 21.1
Descripción: Realizar la tabla del 5 que multiplique hasta el 5 y debe imprimir los siguientes resultados: 5 x 1 = 5 5 x 2 = 10 5 x 3 = 15 5 x 4 = 20 5 x 5 = 25		
multiplicar numero	Tipo de variable: multiplicar(int)	

result	numero (int) result(int)
Código: <pre>const tablaExp = function(pmultiplicar, pnumero){ let multiplicar = pmultiplicar let numero = pnumero let result for(multiplicar = 1; multiplicar<=numero; multiplicar++){ result = multiplicar * numero console.log(numero + "x" + multiplicar + "=" + result) } return "Final" }</pre>	

Nombre de la función: tabla(pmultiplicar,pnumero) Versión: 22.0	
Descripción: Realizar la tabla del 5 que multiplique hasta 5 y den los resultados que son pares e impares.	
multiplicar numero result	Tipo de variable: multiplicar(int) numero (int) result(int)
Código: <pre>function tabla(pmultiplicar, pnumero){ let multiplicar = pmultiplicar let numero = pnumero let result while(multiplicar<numero){ multiplicar ++ result = multiplicar * numero console.log(numero + "x" + multiplicar + "=" + result) if (result%2==0){ console.log("E Par") } else{ console.log("Es Impar") } } return "Fin" }</pre>	

Nombre de la función: tabla(pmultiplicar,pnumero) Versión: 22.1	
Descripción: Realizar la tabla del 5 que multiplique hasta 5 y den los resultados que son pares e impares.	
multiplicar numero result	Tipo de variable: multiplicar(int) numero (int) result(int)
<pre>const tablaExp = function(pmultiplicar, pnumero){ let multiplicar = pmultiplicar let numero = pnumero let result while(multiplicar<numero){ multiplicar ++ result = multiplicar * numero console.log(numero + "X" + multiplicar + "=" + result) if (result%2==0){ console.log("Es Par") } else{ console.log("Es Impar") } } return "Fin" }</pre>	

Nombre de la función: tabla(pmultiplicar,pnumero) Versión: 23.0	
Descripción: Realizar la tabla del 5 que multiplique hasta 5 y den los resultados que son pares e impares.	
multiplicar numero result	Tipo de variable: multiplicar(int) numero (int) result(int)
Código: <pre>function tabla(pmultiplicar, pnumero){ let multiplicar = pmultiplicar let numero = pnumero let result</pre>	

```

for(multiplicar= 1; multiplicar <=numero; multiplicar++){
    result = multiplicar * numero
    console.log(numero + "x" + multiplicar + "=" + result)
    if (result%2==0){
        console.log("Es Par")
    }
    else{
        console.log("Es Impar")
    }
}
return "Fin"
}

```

Nombre de la función: tabla(pmultiplicar,pnumero)	Versión: 23.1
---	---------------

Descripción:

Realizar la tabla del 5 que multiplique hasta 5 y den los resultados que son pares e impares.

multiplicar	Tipo de variable:
numero	multiplicar(int)
result	numero (int)
	result(int)

```

const tablaExp = function(pmultiplicar, pnumero){
    let multiplicar = pmultiplicar
    let numero = pnumero
    let result

    for(multiplicar= 1; multiplicar <=numero; multiplicar++){
        result = multiplicar * numero
        console.log(numero + "x" + multiplicar + "=" + result)
        if (result%2==0){
            console.log("Es Par")
        }
        else{
            console.log("Es Impar")
        }
    }
    return "Fin"
}

```


Nombre de la función: tablas9(ptabla1,ptabla2,plimit1,plimit2,ptablaPar,ptablaImpar)	Versión: 24.0
Descripción: Realizar tablas de multiplicar de 1 a 5 & multiplique hasta 5, debe imprimir los siguientes resultados a. 1X1=1 1x2=2 5X5=25 b. Cuando los resultados sean pares debe imprimir buzz Cuando sean impares debe imprimir bass. C. Debe imprimir Cuantos números sean pares e impares hay en los resultados de las multiplicaciones.	
tabla1 tabla2 limit1 limit2 tablaPar tablaImpar result	Tipo de variable: tabla1(int) tabla2(int) limit1(int) limit2(int) tablaPar(int) tablaImpar(int) result(int)
<pre>function tablas9(ptabla1, ptabla2, plimit1, plimit2, ptablaPar, ptablaImpar){ let tabla1 = ptabla1 let tabla2 = ptabla2 let limit1 = plimit1 let limit2 = plimit2 let tablaPar = ptablaPar let tablaImpar = ptablaImpar tabla1 = 0 while(tabla1<limit1){ tabla1 ++ tabla2 = 0 while(tabla2<limit2){ tabla2++ result = tabla1 * tabla2 console.log(tabla1 + "x" + tabla2 + "=" + result) if (result%2==0){ console.log("Buzz") } } } }</pre>	

```

        tablaPar = tablaPar + 1
    }
    else{
        console.log("Bass")
        tablalmpar = tablalmpar + 1
    }
}
}
}
return "Pares: " + tablaPar + " Impar: " + tablalmpar + "\n" + "Final"
}

```

Nombre de la función: tablas9Exp(ptabla1,ptabla2,plimit1,plimit2,ptablaPar,ptablalmpar)		Versión: 24.1
Descripción: Realizar tablas de multiplicar de 1 a 5 & multiplique hasta 5, debe imprimir los siguientes resultados a. 1X1=1 1x2=2 5X5=25 b. Cuando los resultados sean pares debe imprimir buzz Cuando sean impares debe imprimir bass. C. Debe imprimir Cuantos números sean pares e impares hay en los resultados de las multiplicaciones.		
tabla1 tabla2 limit1 limit2 tablaPar tablalmpar result	Tipo de variable: tabla1(int) tabla2(int) limit1(int) limit2(int) tablaPar(int) tablalmpar(int) result(int)	
<pre>const tablas9Exp = function(ptabla1, ptabla2, plimit1, plimit2, ptablaPar, ptablalmpar){ let tabla1 = ptabla1 let tabla2 = ptabla2 let limit1 = plimit1 let limit2 = plimit2 let tablaPar = ptablaPar let tablalmpar = ptablalmpar tabla1 = 0</pre>		

```

while( tabla1<limit1){
    tabla1 ++
    tabla2 = 0
    while( tabla2<limit2){
        tabla2++
        result = tabla1 * tabla2

        console.log( tabla1 + "x" + tabla2 + "=" + result)

        if ( result%2==0){
            console.log( "Buzz")
            tablaPar = tablaPar + 1
        }
        else{
            console.log( "Bass")
            tablaImpar = tablaImpar + 1
        }
    }
}

return "Pares: " + tablaPar + " Impar: " + tablaImpar + "\n" + "Final"
}

```

OBJ

Nombre de la función: tablas9(ptabla1,ptabla2,plimit1,plimit2,ptablaPar,ptablaImpar)	Versión: 25.0
Descripción: Realizar tablas de multiplicar de 1 a 5 & multiplique hasta 5, debe imprimir los siguientes resultados a. 1X1=1 1x2=2 5X5=25	

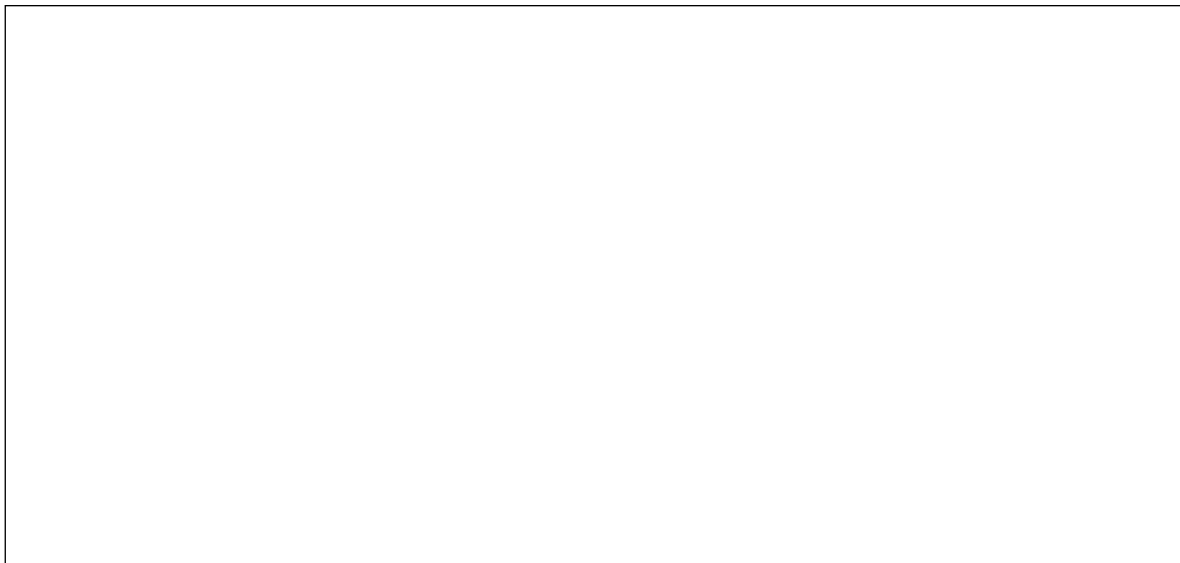
b. Cuando los resultados sean pares debe imprimir buzz Cuando sean impares debe imprimir bass.

C. Debe imprimir Cuantos números sean pares e impares hay en los resultados de las multiplicaciones.

tabla1	Tipo de variable:
tabla2	tabla1(int)
limit1	tabla2(int)
limit2	limit1(int)
tablaPar	limit2(int)
tablaImpar	tablaPar(int)
result	tablaImpar(int)
	result(int)

Código:

```
function ejercici5(contador1) {  
    let par=0;  
    let impar=0;  
    let contar1=contador1;  
    let contar2;  
  
    for (contar1; contar1 <= 5; contar1++) {  
        for (let contar2 = 1; contar2 <= 5; contar2++) {  
  
            let resultado = contar1 * contar2;  
  
            console.log(contar1+"X"+contar2+"="+resultado);  
  
            if (resultado%2==0) {  
                console.log("Buzz");  
                par++;  
            } else{  
                console.log("Bass");  
                impar++;  
            }  
        }  
    }  
  
    console.log("cantidad de pares "+par);  
    console.log("cantidad de impares "+impar);  
    return "Finalizacion"  
}
```



Nombre de la función: tablas9(ptabla1,ptabla2,plimit1,plimit2,ptablaPar,ptablaImpar)		Versión: 25.1
Descripción: Realizar tablas de multiplicar de 1 a 5 & multiplique hasta 5, debe imprimir los siguientes resultados a. 1X1=1 1x2=2 5X5=25 b. Cuando los resultados sean pares debe imprimir buzz Cuando sean impares debe imprimir bass. C. Debe imprimir Cuantos números sean pares e impares hay en los resultados de las multiplicaciones.		
tabla1 tabla2 limit1 limit2 tablaPar tablaImpar result	Tipo de variable: tabla1(int) tabla2(int) limit1(int) limit2(int) tablaPar(int) tablaImpar(int) result(int)	
Código: <pre>const ejercici52=function (contador1) { let par=0; let impar=0; let contar1=contador1; let contar2; for (contar1; contar1 <= 5; contar1++) {</pre>		

```
for (let contar2 = 1; contar2 <= 5; contar2++) {  
  
    let resultado = contar1 * contar2;  
  
    console.log(contar1+"X"+contar2+"="+resultado);  
  
    if (resultado%2==0) {  
        console.log("Buzz");  
        par++;  
    }  
    else{  
        console.log("Bass");  
        impar++;  
    }  
}  
  
console.log("cantidad de pares "+par);  
console.log("cantidad de impares "+impar);  
return "Finalizacion"  
}
```

