



PROGRAMA:

TECNOLOGO EN ANALISIS Y
DESARROLLO DE SOFTWARE

FICHA TECNICA 2900177

Presentado por:

Camilo Andrés Losada
Ramírez

Instructor:

ANDRES MORENO

Neiva Huila



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: saludo()		Versión: 1.0
Descripción: Función que permite realizar un saludo		
rsaludo	tipo de variable: string	
saludo	Tipo de variable: alfanumérico	

Código:

```
function saludo(){
    let saludo = document.getElementById('txtNumeroUno').value;
    let rsaludo

    if(saludo){
        rsaludo = `<strong> mensaje : </strong> ${saludo} <br>`

        document.getElementById('resultados').innerHTML = saludo;
    }
    else{
        document.getElementById('resultados').innerHTML = `digita algun mensaje `;
    }
    return false;
}
```

```
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <link rel="stylesheet" href="bootstrap/css/bootstrap.css">
  <script src="bootstrap/js/bootstrap.js"></script>
  <link rel="stylesheet" href="css/estilos.css">
  <link rel="stylesheet" href="fontawesome/css/all.css">
</head>
<body>
  <div class="container">
    <div class="col-10 row forma">
      <div class="formulario col-6">
        <form class="row g-3 needs-validation" novalidate onsubmit="return saludo();">
          <div class="col-md-8">
            <label for="txtNumeroUno" class="form-label">mensaje</label>
            <input type="text" class="form-control" id="txtNumeroUno" required>
            <div class="valid-feedback">
            </div>
          </div>
          <div class="col-12">
            <button class="btn btn-info" type="submit">calcular</button>
          </div>
        </form>
      </div>
      <div class="respuesta col-4">
        <div id="resultados"></div>
      </div>
    </div>
  </div>

  <script src="bootstrap/js/bootstrap.bundle.js"></script>
  <script src="js/saludo.js"></script>
  <script src="js/javascript.js"></script>
</body>
</html>
```

mensaje

wuenas

✓

calcular

wuenas

```


// Example starter JavaScript for disabling form submissions if there are invalid fields
(function () {
    'use strict'

    // Fetch all the forms we want to apply custom Bootstrap validation styles to
    var forms = document.querySelectorAll('.needs-validation')

    // Loop over them and prevent submission
    Array.prototype.slice.call(forms)
        .forEach(function (form) {
            form.addEventListener('submit', function (event) {
                if (!form.checkValidity()) {
                    event.preventDefault()
                    event.stopPropagation()
                }

                form.classList.add('was-validated')
            }, false)
        })
})();

```

	Tecnólogo en Análisis y Desarrollo de Software Ficha
	Funciones JS

Nombre de la función: operaciones()		Versión: 1.0
Descripción: Función que permite realizar sumar, resta , división y multiplicacion		
NumeroUno	Tipo de variable: int	
NumeroDos	Tipo de variable: int	
sumar	Tipo de variable: int	
restar	Tipo de variable: int	
multiplicar	Tipo de variable: int	

dividir	Tipo de variable: int
rsuma	Tipo de variable: int
rresta	Tipo de variable: int
rmultiplicacion	Tipo de variable: int
rdvision	Tipo de variable: int

```
function operaciones(){
  let NumeroUno = parseInt(document.getElementById('txtNumeroUno').value);
  let NumeroDos = parseInt(document.getElementById('txtNumeroDos').value);
  let sumar;
  let restar;
  let multiplicar;
  let dividir;
  let rsuma;
  let rresta;
  let rmultiplicacion;
  let rdvision;

  sumar = NumeroUno + NumeroDos;
  restar = NumeroUno - NumeroDos;
  multiplicar = NumeroUno * NumeroDos;
  dividir = NumeroUno / NumeroDos;

  if(rsuma && rresta && rdvision && rdvision){
    rsuma = `suma de dos numeros ${sumar} <br>`;
    rresta = `resta de dos numeros ${restar} <br>`;
    rmultiplicacion = `multiplicacion de dos numeros ${multiplicar} <br>`;
    rdvision = `division de dos numeros ${dividir} <br>`;
    document.getElementById('resultados').innerHTML = rsuma + rresta + rmultiplicacion + rdvision;
  }
  else{
    document.getElementById('resultados').innerHTML = `<strong>debes digitar algun numero</strong>`;
  }
  return false;
}
```

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <link rel="stylesheet" href="bootstrap/css/bootstrap.css">
  <script src="bootstrap/js/bootstrap.js"></script>
  <link rel="stylesheet" href="css/estilos.css">
  <link rel="stylesheet" href="Fontawesome/css/all.css">
</head>

<body>
  <div class="container">
    <div class="col-10 row forma">
      <div class="Formulario col-6">
        <form class="row g-3 needs-validation" novalidate onsubmit="return operaciones();">
          <div class="col-md-6">
            <label for="txtNumeroUno" class="form-label">Numero Uno</label>
            <input type="number" class="form-control" id="txtNumeroUno" required>
            <div class="valid-feedback">
            </div>
          </div>
          <div class="col-md-6">
            <label for="txtNumeroDos" class="form-label">Numero Dos</label>
            <input type="number" class="form-control" id="txtNumeroDos" required>
            <div class="valid-feedback">
            </div>
          </div>
          <div class="col-12">
            <button class="btn btn-info" type="submit">calcular</button>
          </div>
        </form>
      </div>
      <div class="respuesta col-4">
        <div id="resultados"></div>
      </div>
    </div>
  </div>

  <script src="bootstrap/js/bootstrap.bundle.js"></script>
  <script src="js/operaciones.js"></script>
  <script src="js/javascript.js"></script>
</body>
```

```
// Example starter JavaScript for disabling form submissions if there are invalid fields
(function () {
    'use strict'

    // Fetch all the forms we want to apply custom Bootstrap validation styles to
    var forms = document.querySelectorAll('.needs-validation')

    // Loop over them and prevent submission
    Array.prototype.slice.call(forms)
        .forEach(function (form) {
            form.addEventListener('submit', function (event) {
                if (!form.checkValidity()) {
                    event.preventDefault()
                    event.stopPropagation()
                }

                form.classList.add('was-validated')
            }, false)
        })
})();
```

Numero Uno

Numero Dos

calcular

suma de dos numeros 10
resta de dos numeros 2
multiplicacion de dos numeros 24
division de dos numeros 1.5



Tecnólogo en Análisis y Desarrollo de
Software

Ficha

Funciones JS

Nombre de la función: porcentaje()		Versión: 1.0
Descripción: Función que permite pasar cualquier numero a porcentaje		
NumeroUno	Tipo de variable: int	
porcentaje	Tipo de variable: float	
rporcentaje	Tipo de variable: float	

```

function porcentaje(){
  let NumeroUno = parseInt(document.getElementById('txtNumeroUno').value);
  let porcentaje;
  let rpocentaje

  porcentaje = NumeroUno / 100;

  if(porcentaje ){
    rpocentaje = `<strong>porcentaje : ${porcentaje}</strong>`

    document.getElementById('resultados').innerHTML = rpocentaje;
  }
  else{
    document.getElementById('resultados').innerHTML = `<strong>debes digitar algun numero</strong>`
  }
  return false;
}

```

```

<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <link rel="stylesheet" href="bootstrap/css/bootstrap.css">
  <script src="bootstrap/js/bootstrap.js"></script>
  <link rel="stylesheet" href="css/estilos.css">
  <link rel="stylesheet" href="fontawesome/css/all.css">
</head>

<body>
  <div class="container">
    <div class="col-10 row forma">
      <div class="formulario col-6">
        <form class="row g-3 needs-validation" novalidate onsubmit="return operaciones();">
          <div class="col-md-6">
            <label for="txtNumeroUno" class="form-label">Numero Uno</label>
            <input type="number" class="form-control" id="txtNumeroUno" required>
            <div class="valid-feedback">
            </div>
          </div>
          <div class="col-md-6">
            <label for="txtNumeroDos" class="form-label">Numero Dos</label>
            <input type="number" class="form-control" id="txtNumeroDos" required>
            <div class="valid-feedback">
            </div>
          </div>
          <div class="col-12">
            <button class="btn btn-info" type="submit">calcular</button>
          </div>
        </form>
      </div>
      <div class="respuesta col-4">
        <div id="resultados"></div>
      </div>
    </div>
  </div>

  <script src="bootstrap/js/bootstrap.bundle.js"></script>
  <script src="js/operaciones.js"></script>
  <script src="js/javascript.js"></script>
</body>

```

```

// Example starter JavaScript for disabling form submissions if there are invalid fields
(function () {
  'use strict'

  // Fetch all the forms we want to apply custom Bootstrap validation styles to
  var forms = document.querySelectorAll('.needs-validation')

  // Loop over them and prevent submission
  Array.prototype.slice.call(forms)
    .forEach(function (form) {
      form.addEventListener('submit', function (event) {
        if (!form.checkValidity()) {
          event.preventDefault()
          event.stopPropagation()
        }

        form.classList.add('was-validated')
      }, false)
    })
})()

```

Numero Uno

6

✓

Numero Dos

4

✓


calcular

suma de dos numeros 10

resta de dos numeros 2

multiplicacion de dos numeros 24

division de dos numeros 1.5

	Tecnólogo en Análisis y Desarrollo de Software Ficha
	Funciones JS

Nombre de la función: porcentaje()		Versión: 1.0
Descripción: Función que permite calcular las notas de un estudiante		
Nota1	Tipo de variable: string	
Nota2	Tipo de variable: string	
Nota3	Tipo de variable: string	
promedio	Tipo de variable: float	
rpromedio	Tipo de variable: float	

```

function porcentaje(){
  let nota1 = parseInt(document.getElementById("txtNumeroUno").value)
  let nota2 = parseInt(document.getElementById("txtNumeroDos").value)
  let nota3 = parseInt(document.getElementById("txtNumeroTres").value)
  let promedio;
  let rpromedio
  promedio = (nota1 + nota2 + nota3) / 3
  if(promedio){
    rpromedio = `<strong>nota final : ${promedio}</strong>`
    document.getElementById('resultados').innerHTML = rpromedio;
  }
  else{
    document.getElementById('resultados').innerHTML = "completa todos los campos ";
  }
  return false
}

```

```

</head>
<body>
  <div class="container">
    <div class="col-10 row forma">
      <div class="formulario col-6">
        <form class="row g-3 needs-validation" novalidate onsubmit="return porcentaje();">
          <div class="col-md-6">
            <label for="txtNumeroUno" class="form-label">Primera Nota </label>
            <input type="number" class="form-control" id="txtNumeroUno" required>
            <div class="valid-feedback">
            </div>
          </div>
          <div class="col-md-6">
            <label for="txtNumeroDos" class="form-label">segunda Nota </label>
            <input type="number" class="form-control" id="txtNumeroDos" required>
            <div class="valid-feedback">
            </div>
          </div>
          <div class="col-md-6">
            <label for="txtNumeroTres" class="form-label">tercer nota </label>
            <input type="number" class="form-control" id="txtNumeroTres" required>
            <div class="valid-feedback">
            </div>
          </div>
          <div class="col-12">
            <button class="btn btn-info" type="submit">calcular</button>
          </div>
        </form>
      </div>
      <div class="respuesta col-4">
        <div id="resultados"></div>
      </div>
    </div>
  </div>
</body>
</html>
<script src="bootstrap/js/bootstrap.bundle.js"></script>
<script src="js/porcentaje.js"></script>
<script src="js/javascript.js"></script>

```

```

// Example starter JavaScript for disabling form submissions if there are invalid fields
(function () {
  'use strict'

  // Fetch all the forms we want to apply custom Bootstrap validation styles to
  var forms = document.querySelectorAll('.needs-validation')

  // Loop over them and prevent submission
  Array.prototype.slice.call(forms)
    .forEach(function (form) {
      form.addEventListener('submit', function (event) {
        if (!form.checkValidity()) {
          event.preventDefault()
          event.stopPropagation()
        }

        form.classList.add('was-validated')
      }, false)
    })
})();

```

Primera Nota



segunda Nota



tercer nota



calcular

nota final : 3.3333333333333335



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: porcentaje()		Versión: 1.0
Descripción: Función que permite calcular el promedio y la nota final de un estudiante		
Nota1	Tipo de variable: string	
Nota2	Tipo de variable: string	
Nota3	Tipo de variable: string	
notafinal	Tipo de variable: float	
rnotafinal	Tipo de variable: string	
PromedioNota1	Tipo de variable: float	
PromedioNota2	Tipo de variable: float	
PromedioNota3	Tipo de variable: float	
Rpromedio1	Tipo de variable: string	
Rpromedio2	Tipo de variable: string	
Rpromedio3	Tipo de variable: string	

```

function porcentaje(){
  let nota1 = parseFloat(document.getElementById("txtNumeroUno").value)
  let nota2 = parseFloat(document.getElementById("txtNumeroDos").value)
  let nota3 = parseFloat(document.getElementById("txtNumeroTres").value)
  let notaFinal;
  let rnotaFinal;
  let promedioNota1;
  let promedioNota2;
  let promedioNota3;
  let rpromedio1
  let rpromedio2
  let rpromedio3
  promedioNota1 = nota1 * 0.3
  promedioNota2 = nota2 * 0.3
  promedioNota3 = nota3 * 0.4
  notaFinal = promedioNota1 + promedioNota2 + promedioNota3

  if(promedioNota1 && promedioNota2 && promedioNota3){
    rpromedio1 = `<strong>primera Nota : ${promedioNota1}</strong> <br>`
    rpromedio2 = `<strong>segunda Nota : ${promedioNota2}</strong> <br>`
    rpromedio3 = `<strong>tercera Nota : ${promedioNota3}</strong> <br>`
    rnotaFinal = `<strong>Nota final : ${notaFinal}</strong>`
    document.getElementById('resultados').innerHTML = rpromedio1 + rpromedio2 + rpromedio3 + rnotaFinal;
  }
  else{
    document.getElementById('resultados').innerHTML = " <strong> rellena las casillas </strong>";
  }
  return false
}

```

```

body>
  <div class="container">
    <div class="col-10 row forma">
      <div class="formulario col-6">
        <form class="row g-3 needs-validation" novalidate onsubmit="return porcentaje();">
          <div class="col-md-6">
            <label for="txtNumeroUno" class="form-label">Primera Nota</label>
            <input type="number" class="form-control" id="txtNumeroUno" required>
            <div class="valid-feedback">
            </div>
          </div>
          <div class="col-md-6">
            <label for="txtNumeroDos" class="form-label">segunda Nota </label>
            <input type="number" class="form-control" id="txtNumeroDos" required>
            <div class="valid-feedback">
            </div>
          </div>
          <div class="col-md-6">
            <label for="txtNumeroTres" class="form-label">tercer nota </label>
            <input type="number" class="form-control" id="txtNumeroTres" required>
            <div class="valid-feedback">
            </div>
          </div>
          <div class="col-12">
            <button class="btn btn-info" type="submit">calcular</button>
          </div>
        </form>
      </div>
      <div class="respuesta col-4">
        <div id="resultados"></div>
      </div>
    </div>
  </div>

```

```

// Example starter JavaScript for disabling form submissions if there are invalid fields
(function () {
  'use strict'

  // Fetch all the forms we want to apply custom Bootstrap validation styles to
  var forms = document.querySelectorAll('.needs-validation')

  // Loop over them and prevent submission
  Array.prototype.slice.call(forms)
    .forEach(function (form) {
      form.addEventListener('submit', function (event) {
        if (!form.checkValidity()) {
          event.preventDefault()
          event.stopPropagation()
        }

        form.classList.add('was-validated')
      }, false)
    })
})();

```

Primera Nota

5.0 ✓

segunda Nota


3.0 ✓

tercer nota

2.0 ✓

calcular

primera Nota : 1.5
segunda Nota : 0.8999999999999999
tercera Nota : 0.8
Nota final : 3.2

	Tecnólogo en Análisis y Desarrollo de Software Ficha
	Funciones JS

Nombre de la función: porcentaje()		Versión: 1.0
Descripción: Función que permite calcular el área de un triángulo un cuadrado y un rectángulo		
cuadrado	Tipo de variable: string	
RectanguloB	Tipo de variable: string	
RectanguloA	Tipo de variable: string	
trianguloB	Tipo de variable: string	
trianguloA	Tipo de variable: string	
AreaC	Tipo de variable: float	
AreaR	Tipo de variable: float	
AreaT	Tipo de variable: float	
rAreaC	Tipo de variable: string	
rAreaR	Tipo de variable: string	

```

function porcentaje(){
  let Cuadrado = parseFloat(document.getElementById("txtCuadradoLado").value)
  let rectanguloB = parseFloat(document.getElementById("txtRectanguloBase").value)
  let rectanguloA = parseFloat(document.getElementById("txtRectanguloAltura").value)
  let trianguloB = parseFloat(document.getElementById("txtTrianguloBase").value)
  let trianguloA = parseFloat(document.getElementById("txtTrianguloAltura").value)
  let AreaC
  let AreaR
  let AreaT
  let rAreaC
  let rAreaR
  let rAreaT

  AreaC = Cuadrado * Cuadrado
  AreaR = rectanguloB * rectanguloA
  AreaT = (trianguloB * trianguloA) / 2

  if(AreaC && AreaR && AreaT){
    rAreaC = `<strong>Area del Cuadrado : ${AreaC}</strong> <br>`
    rAreaR = `<strong>Area del Rectangulo : ${AreaR}</strong> <br>`
    rAreaT = `<strong>Area del Triangulo : ${AreaT}</strong> <br>`

    document.getElementById('resultados').innerHTML = rAreaC + rAreaR + rAreaT
  }else{
    document.getElementById('resultados').innerHTML = `<strong>por favor rellena todos los campos</strong>`
  }
  return false
}

<div class="container">
  <div class="col-10 row forma">
    <div class="formulario col-6">
      <form class="row g-2 needs-validation" novalidate onsubmit="return porcentaje();">
        <div class="col-md-5">
          <h5>Area Cuadrado</h5>
          <label for="txtCuadradoLado" class="form-label">Lado</label>
          <input type="number" class="form-control" id="txtCuadradoLado" required>
          <div class="valid-feedback">
            </div>
        </div>
        <h5>Area Rectangulo</h5>
        <div class="col-md-4">
          <label for="txtRectanguloBase" class="form-label">Base</label>
          <input type="number" class="form-control" id="txtRectanguloBase" required>
          <div class="valid-feedback">
            </div>
        </div>
        <div class="col-md-4">
          <label for="txtRectanguloAltura" class="form-label">Altura</label>
          <input type="number" class="form-control" id="txtRectanguloAltura" required>
          <div class="valid-feedback">
            </div>
        </div>
        <h5>Area Triangulo</h5>
        <div class="col-md-4">
          <label for="txtTrianguloBase" class="form-label">Base</label>
          <input type="number" class="form-control" id="txtTrianguloBase" required>
          <div class="valid-feedback">
            </div>
        </div>
        <div class="col-md-4">
          <label for="txtTrianguloAltura" class="form-label">Altura</label>
          <input type="number" class="form-control" id="txtTrianguloAltura" required>
          <div class="valid-feedback">
            </div>
        </div>
        <div class="col-12">
          <button class="btn btn-info" type="submit">calcular</button>
        </div>
      </form>
    </div>
  </div>

```

```

// Example starter JavaScript for disabling form submissions if there are invalid fields
(function () {
  'use strict'

  // Fetch all the forms we want to apply custom Bootstrap validation styles to
  var forms = document.querySelectorAll('.needs-validation')

  // Loop over them and prevent submission
  Array.prototype.slice.call(forms)
    .forEach(function (form) {
      form.addEventListener('submit', function (event) {
        if (!form.checkValidity()) {
          event.preventDefault()
          event.stopPropagation()
        }

        form.classList.add('was-validated')
      }, false)
    })
})()

```

Area Cuadrado

Lado

 ✓

Area Rectangulo

Base Altura

 ✓ ✓

Area Triangulo


Base Altura

 ✓ ✓

Area del Cuadrado : 9

Area del Rectangulo : 28

Area del Triangulo : 9

	Tecnólogo en Análisis y Desarrollo de Software Ficha
	Funciones JS

Nombre de la función: operaciones()		Versión: 1.0
Descripción: Función que permite calcular el la nomina de una persona		
DiasTrabajados	Tipo de variable: string	
ValorDia	Tipo de variable: string	
sueldo	Tipo de variable: string	
pension	Tipo de variable: string	
arl	Tipo de variable: string	
salud	Tipo de variable: float	
rsueldo	Tipo de variable: float	
rpension	Tipo de variable: float	

rarl	Tipo de variable: string
rsalud	Tipo de variable: string
total	Tipo de variable: string
rtotal	Tipo de variable: string

```
function operaciones(){
  let DiasTrabajados = parseInt(document.getElementById('txtNumeroUno').value);
  let valorDia = parseInt(document.getElementById('txtNumeroDos').value);
  let sueldo;
  let pension;
  let arl;
  let salud;
  let rsueldo;
  let rpension;
  let rarl;
  let rsalud;
  let total;
  let rtotal;

  sueldo = parseInt(DiasTrabajados * valorDia);
  arl = parseInt(sueldo * 0.052);
  salud = parseInt(sueldo * 0.12);
  pension = parseInt(sueldo * 0.16);
  total = sueldo - (arl + salud + pension) ;

  if(sueldo && arl && salud && pension && total){
    rsueldo = `<strong> sueldo de la persona : </strong> ${sueldo} <br>`
    rpension = `<strong> retencion del arl : </strong> ${arl} <br>`
    rarl = `<strong> retencion de salud : </strong> ${salud} <br>`
    rsalud = `<strong> retencion de pension : </strong> ${pension} <br>`
    rtotal = `<strong> sueldo total de la persona : </strong> ${total}`

    document.getElementById('resultados').innerHTML = rsueldo + rpension + rarl + rsalud + rtotal;
  }else{
    document.getElementById('resultados').innerHTML = 'Rellena todas las carpetas'
  }
  return false;
}
```

```
<div class="container">
  <div class="col-10 row forma">
    <div class="formulario col-6">
      <form class="row g-3 needs-validation" novalidate onsubmit="return operaciones();">
        <div class="col-md-6">
          <label for="txtNumeroUno" class="form-label">Dias trabajados</label>
          <input type="number" class="form-control" id="txtNumeroUno" required>
          <div class="valid-feedback">
            Looks good!
          </div>
        </div>
        <div class="col-md-6">
          <label for="txtNumeroDos" class="form-label">valor dia </label>
          <input type="number" class="form-control" id="txtNumeroDos" required>
          <div class="valid-feedback">
            Looks good!
          </div>
        </div>
        <div class="col-12">
          <button class="btn btn-info" type="submit">calcular</button>
        </div>
      </form>
    </div>
    <div class="respuesta col-4">
      <div id="resultados"></div>
    </div>
  </div>
</div>
```

```
// Example starter JavaScript for disabling form submissions if there are invalid fields
(function () {
  'use strict'

  // Fetch all the forms we want to apply custom Bootstrap validation styles to
  var forms = document.querySelectorAll('.needs-validation')

  // Loop over them and prevent submission
  Array.prototype.slice.call(forms)
    .forEach(function (form) {
      form.addEventListener('submit', function (event) {
        if (!form.checkValidity()) {
          event.preventDefault()
          event.stopPropagation()
        }

        form.classList.add('was-validated')
      }, false)
    })
})()
```

Días trabajados

30



Looks good!

calcular

valor día

45000



Looks good!

suelo de la persona : 1350000

retencion del arl : 70200

retencion de salud : 162000

retencion de pension : 216000

suelo total de la persona : 901800



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: DosNumeros()		Versión: 1.0
Descripción:		
Función que permite saber cual de los dos Numeros es Mayor		
NumeroUno	Tipo de variable: string	
NumerosDos	Tipo de variable: string	
Respuesta	Tipo de variable: string	

```
function DosNumeros(){
  let NumeroUno = parseInt(document.getElementById('txtNumeroUno').value);
  let NumeroDos = parseInt(document.getElementById('txtNumeroDos').value);
  let Respuesta

  if(NumeroUno > NumeroDos){
    Respuesta = `<strong> El Numero uno ${NumeroUno} es mayor al Numero Dos ${NumeroDos} </strong> <br>`
    document.getElementById('resultados').innerHTML = Respuesta;
  }else{
    Respuesta = `<strong> El Numero Dos ${NumeroDos} es mayor al Numero Uno ${NumeroUno} </strong> <br>`
    document.getElementById('resultados').innerHTML = Respuesta;
    if(NumeroUno && NumeroDos){

    }else{
      document.getElementById('resultados').innerHTML = ` <strong> Rellena todas las casillas </strong>`;
    }
  }
  return false
}
```

```
<div class="container">
  <div class="col-10 row forma">
    <div class="formulario col-6">
      <form class="row g-3 needs-validation" novalidate onsubmit="return DosNumeros();">
        <div class="col-md-6">
          <label for="txtNumeroUno" class="form-label">Numero Uno</label>
          <input type="number" class="form-control" id="txtNumeroUno" required>
          <div class="valid-feedback">
            </div>
        </div>
        <div class="col-md-6">
          <label for="txtNumeroDos" class="form-label">Numero Dos</label>
          <input type="number" class="form-control" id="txtNumeroDos" required>
          <div class="valid-feedback">
            </div>
        </div>
        <div class="col-12">
          <button class="btn btn-info" type="submit">calcular</button>
        </div>
      </form>
    </div>
    <div class="respuesta col-4">
      <div id="resultados"></div>
    </div>
  </div>
</div>
```

```
// Example starter JavaScript for disabling form submissions if there are invalid fields
(function () {
  'use strict'

  // Fetch all the forms we want to apply custom Bootstrap validation styles to
  var forms = document.querySelectorAll('.needs-validation')

  // Loop over them and prevent submission
  Array.prototype.slice.call(forms)
    .forEach(function (form) {
      form.addEventListener('submit', function (event) {
        if (!form.checkValidity()) {
          event.preventDefault()
          event.stopPropagation()
        }
        form.classList.add('was-validated')
      }, false)
    })
})();
```

Numero Uno

6



Numero Dos

4



calcular

El Numero uno 6 es mayor al Numero Dos 4



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: calcularEdad()		Versión: 1.0
Descripción: Función que permite calcular la edad de una persona		
fechaNacimiento	Tipo de variable: string	
fechaActual	Tipo de variable: string	
diferencia	Tipo de variable: string	
edad	Tipo de variable: float	
resultado	Tipo de variable: string	

```
function calcularEdad() {  
  let fechaNacimiento = new Date(document.getElementById('txtedad').value);  
  let fechaActual = new Date();  
  
  let diferencia = fechaActual - fechaNacimiento;  
  
  let edad = Math.floor(diferencia / (1000 * 60 * 60 * 24 * 365.25));  
  
  let resultado = '<strong>La edad de la persona es ${edad} años.</strong>';  
  document.getElementById('resultados').innerHTML = resultado;  
  
  return false;  
}
```

```
<body>  
  <div class="container">  
    <div class="col-10 row forma">  
      <div class="formulario col-6">  
        <form class="row g-3 needs-validation" novalidate onsubmit="return calcularEdad();">  
          <div class="col-md-6">  
            <label for="txtedad" class="form-label">Selecciona el Día , Mes y Año </label>  
            <input type="date" class="form-control" id="txtedad" required>  
            <div class="valid-feedback">  
            </div>  
          </div>  
          <div class="col-12">  
            <button class="btn btn-info" type="submit">calcular</button>  
          </div>  
        </form>  
      </div>  
  
      <div class="respuesta col-4">  
        <div id="resultados"></div>  
      </div>  
    </div>  
  </div>
```

```
// Example starter JavaScript for disabling form submissions if there are invalid fields
(function () {
  'use strict'

  // Fetch all the forms we want to apply custom Bootstrap validation styles to
  var forms = document.querySelectorAll('.needs-validation')

  // Loop over them and prevent submission
  Array.prototype.slice.call(forms)
    .forEach(function (form) {
      form.addEventListener('submit', function (event) {
        if (!form.checkValidity()) {
          event.preventDefault()
          event.stopPropagation()
        }

        form.classList.add('was-validated')
      }, false)
    })
})()
```

Selecciona el Día , Mes y Año

03/07/2007



calcular

La edad de la persona es 17 años.



Tecnólogo en Análisis y Desarrollo de
Software
Ficha

Funciones JS

Nombre de la función: MayorDeTres()		Versión: 1.0
Descripción: Función que permite saber cual de los 3 numeros es mayor o si los 3 son iguales		
NumeroUno	Tipo de variable: string	
NumeroDos	Tipo de variable: string	
NumeroTres	Tipo de variable: string	
respuesta	Tipo de variable: string	

```

function MayorDeTres(){
  let NumeroUno = parseInt(document.getElementById('txtNumeroUno').value);
  let NumeroDos = parseInt(document.getElementById('txtNumeroDos').value);
  let NumeroTres = parseInt(document.getElementById('txtNumeroTres').value);
  let Respuesta

  if(NumeroUno == NumeroDos && NumeroUno == NumeroTres){
    Respuesta = '<strong> Los 3 Numeros son iguales </strong> <br>'
    document.getElementById('resultados').innerHTML = Respuesta;
  }else if(NumeroUno > NumeroDos && NumeroUno > NumeroTres ){
    Respuesta = '<strong> El Numero Uno ${NumeroUno} es el numero mayor </strong> <br>'
    document.getElementById('resultados').innerHTML = Respuesta;
  }else if(NumeroDos > NumeroUno && NumeroDos > NumeroTres){
    Respuesta = '<strong> El Numero Dos ${NumeroDos} es el numero mayor </strong> <br>'
    document.getElementById('resultados').innerHTML = Respuesta;
  }else if(NumeroTres > NumeroUno && NumeroTres > NumeroDos){
    Respuesta = '<strong> El Numero Tres ${NumeroTres} es el numero mayor </strong> <br>'
    document.getElementById('resultados').innerHTML = Respuesta;
  }else if(NumeroUno && NumeroDos && NumeroTres){
  }else{
    document.getElementById('resultados').innerHTML = ' <strong> Rellena todas las casillas </strong>';
  }
  return false
}

```

```

<div class="container">
  <div class="col-10 row forma">
    <div class="formulario col-6">
      <form class="row g-3 needs-validation" novalidate onsubmit="return MayorDeTres();">
        <div class="col-md-6">
          <label for="txtNumeroUno" class="form-label">Numero Uno</label>
          <input type="number" class="form-control" id="txtNumeroUno" required>
          <div class="valid-feedback">
            </div>
        </div>
        <div class="col-md-6">
          <label for="txtNumeroDos" class="form-label">Numero Dos</label>
          <input type="number" class="form-control" id="txtNumeroDos" required>
          <div class="valid-feedback">
            </div>
        </div>
        <div class="col-md-6">
          <label for="txtNumeroTres" class="form-label">Numero Tres</label>
          <input type="number" class="form-control" id="txtNumeroTres" required>
          <div class="valid-feedback">
            </div>
        </div>
        <div class="col-12">
          <button class="btn btn-info" type="submit">calcular</button>
        </div>
      </form>
    </div>

    <div class="respuesta col-4">
      <div id="resultados"></div>
    </div>
  </div>

```

```

// Example starter JavaScript for disabling form submissions if there are invalid fields
(function () {
  'use strict'


  // Fetch all the forms we want to apply custom Bootstrap validation styles to
  var forms = document.querySelectorAll('.needs-validation')

  // Loop over them and prevent submission
  Array.prototype.slice.call(forms)
    .forEach(function (form) {
      form.addEventListener('submit', function (event) {
        if (!form.checkValidity()) {
          event.preventDefault()
          event.stopPropagation()
        }

        form.classList.add('was-validated')
      }, false)
    })
})();

```

<p>Numero Uno</p> <input type="text" value="6"/> ✓	<p>Numero Dos</p> <input type="text" value="5"/> ✓	<p>El Numero Tres 29 es el numero mayor</p>
<p>Numero Tres</p> <input type="text" value="29"/> ✓		
<p><input type="button" value="calcular"/></p>		

	<p>Tecnólogo en Análisis y Desarrollo de Software</p> <p>Ficha</p>
	<p>Funciones JS</p>

Nombre de la función: MayorDos()		Versión: 1.0
Descripción: Función que permite saber cual de los 2 números es el mayor o si los 2 son iguales		
NumeroUno	Tipo de variable: string	
NumeroDos	Tipo de variable: string	
respuesta	Tipo de variable: string	

```

function MayorDos(){
  let NumeroUno = parseInt(document.getElementById('txtNumeroUno').value);
  let NumeroDos = parseInt(document.getElementById('txtNumeroDos').value);
  let Respuesta

  if(NumeroUno == NumeroDos && NumeroDos == NumeroUno){
    Respuesta = `<strong> Los 2 Numeros son iguales </strong> <br>`
    document.getElementById('resultados').innerHTML = Respuesta;
  }else if(NumeroUno > NumeroDos){
    Respuesta = `<strong> El Numero Uno ${NumeroUno} es el numero mayor </strong> <br>`
    document.getElementById('resultados').innerHTML = Respuesta;
  }else{
    Respuesta = `<strong> El Numero Dos ${NumeroDos} es el numero mayor </strong> <br>`
    document.getElementById('resultados').innerHTML = Respuesta;
  }if(NumeroUno && NumeroDos){

  }else{
    document.getElementById('resultados').innerHTML = ` <strong> Rellena todas las casillas </strong>`;
  }
  return false
}

```

```

<body>
  <div class="container">
    <div class="col-10 row forma">
      <div class="formulario col-6">
        <form class="row g-3 needs-validation" novalidate onsubmit="return MayorDos();">
          <div class="col-md-6">
            <label for="txtNumeroUno" class="form-label">Numero Uno</label>
            <input type="number" class="form-control" id="txtNumeroUno" required>
            <div class="valid-feedback">
            </div>
          </div>
          <div class="col-md-6">
            <label for="txtNumeroDos" class="form-label">Numero Dos</label>
            <input type="number" class="form-control" id="txtNumeroDos" required>
            <div class="valid-feedback">
            </div>
          </div>
          <div class="col-12">
            <button class="btn btn-info" type="submit">calcular</button>
          </div>
        </form>
      </div>
      <div class="respuesta col-4">
        <div id="resultados"></div>
      </div>
    </div>
  </div>

```

```

// Example starter JavaScript for disabling form submissions if there are invalid fields
(function () {
  'use strict'

  // Fetch all the forms we want to apply custom Bootstrap validation styles to
  var forms = document.querySelectorAll('.needs-validation')

  // Loop over them and prevent submission
  Array.prototype.slice.call(forms)
    .forEach(function (form) {
      form.addEventListener('submit', function (event) {
        if (!form.checkValidity()) {
          event.preventDefault()
          event.stopPropagation()
        }

        form.classList.add('was-validated')
      }, false)
    })
})()


```

Numero Uno

Numero Dos

calcular

Los 2 Numeros son iguales



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: Areas()		Versión: 1.0
Descripción: Función que permite calcular el área de un Triangulo		
LadoUno	Tipo de variable: string	

LadoDos	Tipo de variable: string
LadoTres	Tipo de variable: string
respuesta	Tipo de variable: string
AreaUno	Tipo de variable: float
AreaDos	Tipo de variable: float
AreaTres	Tipo de variable: float

```
function Areas(){
    let LadoUno = parseInt(document.getElementById('txtNumeroUno').value);
    let LadoDos = parseInt(document.getElementById('txtNumeroDos').value);
    let LadoTres = parseInt(document.getElementById('txtNumeroTres').value);
    let Respuesta
    let AreaUno
    let AreaDos
    let AreaTres

    AreaUno = LadoUno * LadoUno
    AreaDos = LadoDos * LadoDos
    AreaTres = LadoTres * LadoTres

    if (AreaUno == AreaDos && AreaUno == AreaTres && AreaDos == LadoTres){
        Respuesta = '<strong> las Areas son iguales </strong> <br>'
        document.getElementById('resultados').innerHTML = Respuesta;
    } else if (AreaUno > AreaDos && AreaUno > AreaTres) {
        Respuesta = '<strong> El Area Numero Uno ${AreaUno} es mayor </strong> <br>'
        document.getElementById('resultados').innerHTML = Respuesta;
    } else if (AreaDos > AreaUno && AreaDos > AreaTres) {
        Respuesta = '<strong> El Area Numero Dos ${AreaDos} es mayor </strong> <br>'
        document.getElementById('resultados').innerHTML = Respuesta;
    } else {
        Respuesta = '<strong> El Area Numero Tres ${AreaTres} es mayor </strong> <br>'
        document.getElementById('resultados').innerHTML = Respuesta;
    } if (AreaUno && AreaDos && AreaTres){
    } else {
        document.getElementById('resultados').innerHTML = '<strong> Rellena todas las casillas </strong>';
    }
    return false
}
```

```
<div class="container">
    <div class="col-10 row forma">
        <div class="formulario col-6">
            <form class="row g-3 needs-validation" novalidate onsubmit="return Areas();">
                <div class="col-md-6">
                    <label for="txtNumeroUno" class="form-label">Lado Uno</label>
                    <input type="number" class="form-control" id="txtNumeroUno" required>
                    <div class="valid-feedback">
                    </div>
                </div>
                <div class="col-md-6">
                    <label for="txtNumeroDos" class="form-label">Lado Dos</label>
                    <input type="number" class="form-control" id="txtNumeroDos" required>
                    <div class="valid-feedback">
                    </div>
                </div>
                <div class="col-md-6">
                    <label for="txtNumeroTres" class="form-label">Lado Tres</label>
                    <input type="number" class="form-control" id="txtNumeroTres" required>
                    <div class="valid-feedback">
                    </div>
                </div>
                <div class="col-12">
                    <button class="btn btn-info" type="submit">calcular</button>
                </div>
            </form>
        </div>
        <div class="respuesta col-4">
            <div id="resultados"></div>
        </div>
    </div>
</div>
```

```
// Example starter JavaScript for disabling form submissions if there are invalid fields
(function () {
    'use strict'

    // Fetch all the forms we want to apply custom Bootstrap validation styles to
    var forms = document.querySelectorAll('.needs-validation')

    // Loop over them and prevent submission
    Array.prototype.slice.call(forms)
        .forEach(function (form) {
            form.addEventListener('submit', function (event) {
                if (!form.checkValidity()) {
                    event.preventDefault()
                    event.stopPropagation()
                }

                form.classList.add('was-validated')
            }, false)
        })
})()
```

Lado Uno



Lado Dos



Lado Tres



calcular

El Area Numero Dos 1764 es mayor



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: CalcularEdad()		Versión: 1.0
Descripción: Función que permite calcular la edad de una persona		
fechaNacimientoPerUno	Tipo de variable: string	
fechaNacimientoPerDos	Tipo de variable: string	
fechaNacimientoPerTres	Tipo de variable: string	
respuesta	Tipo de variable: string	
PersonasUno	Tipo de variable: string	
PersonasDos	Tipo de variable: string	

PersonasTres	Tipo de variable: string
promedio	Tipo de variable: float

```
function calcularEdad() {
  let fechaNacimientoPerUno = new Date(document.getElementById('txtedad').value);
  let fechaNacimientoPerDos = new Date(document.getElementById('txtedadDos').value);
  let fechaNacimientoPerTres = new Date(document.getElementById('txtedadTres').value);
  let respuesta = "";

  let PersonaUno = nacimiento(fechaNacimientoPerUno);
  let PersonaDos = nacimiento(fechaNacimientoPerDos);
  let PersonaTres = nacimiento(fechaNacimientoPerTres);
  let promedio;

  promedio = (PersonaUno + PersonaDos + PersonaTres) / 3

  respuesta += validar(PersonaUno)
  respuesta += validar(PersonaDos)
  respuesta += validar(PersonaTres)

  document.getElementById("resultados").innerHTML = `
  la edad de la persona Uno es de ${PersonaUno} <br>
  la edad de la persona Dos es de ${PersonaDos} <br>
  la edad de la persona Tres es de ${PersonaTres} <br>
  el promedio de las edades es de : ${promedio} <br>
  ${respuesta}`

  return false;
}
```

```
<div class="container">
  <div class="col-10 row forma">
    <div class="formulario col-6">
      <form class="row g-3 needs-validation" novalidate onsubmit="return calcularEdad();">
        <div class="col-md-6">
          <label for="txtedad" class="form-label">Personas Uno</label>
          <input type="date" class="form-control" id="txtedad" required>
          <div class="valid-feedback">
            </div>
        </div>
        <div class="col-md-6">
          <label for="txtedadDos" class="form-label">Personas Dos</label>
          <input type="date" class="form-control" id="txtedadDos" required>
          <div class="valid-feedback">
            </div>
        </div>
        <div class="col-md-6">
          <label for="txtedadTres" class="form-label">Personas Tres</label>
          <input type="date" class="form-control" id="txtedadTres" required>
          <div class="valid-feedback">
            </div>
        </div>
        <div class="col-12">
          <button class="btn btn-info" type="submit">calcular</button>
        </div>
      </form>
    </div>
    <div class="respuesta col-4">
      <div id="resultados"></div>
    </div>
  </div>
</div>
```



```
// Example starter JavaScript for disabling form submissions if there are invalid fields
(function () {
  'use strict'

  // Fetch all the forms we want to apply custom Bootstrap validation styles to
  var forms = document.querySelectorAll('.needs-validation')



  // Loop over them and prevent submission
  Array.prototype.slice.call(forms)
    .forEach(function (form) {
      form.addEventListener('submit', function (event) {
        if (!form.checkValidity()) {
          event.preventDefault()
          event.stopPropagation()
        }

        form.classList.add('was-validated')
      }, false)
    })
})()
```




Personas Uno

21/07/2017  

Personas Dos

21/08/2008  

Personas Tres

03/07/2008  

calcular

la edad de la persona Uno es de 6

la edad de la persona Dos es de 15


la edad de la persona Tres es de 16

el promedio de las edades es de : 12.333333333333334

la persona tiene : 6 , eres Menor de edad

la persona tiene : 15 , eres Menor de edad

la persona tiene : 16 , eres Menor de edad

	Tecnólogo en Análisis y Desarrollo de Software Ficha
	Funciones JS

Nombre de la función: Nacimiento()		Versión: 1.0
Descripción: Función que calcula la edad exacta de una prsona		
fechaAc	Tipo de variable: float	
Respuesta	Tipo de variable: float	
edad	Tipo de variable: float	
<pre>function nacimiento(fechaNac){ let fechaAc = new Date(); let Respuesta; let edad; Respuesta = fechaAc - fechaNac edad = Math.floor(Respuesta / (1000 * 60 * 60 * 24 * 365.25)); return edad }</pre>		



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: validar(edad)

Versión: 1.0

Descripción:

Función valida que la persona tiene 18 años

edad

Tipo de variable: int

```
function validar(edad) {  
  if (edad >= 18) {  
    return `la persona tiene : ${edad} , eres Mayor de edad <br>`;  
  } else {  
    return `la persona tiene : ${edad} , eres Menor de edad <br>`;  
  }  
}
```



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: Areas()

Versión: 1.0

Descripción:

Función que permite calcular el área de un Triangulo

DiasTrabajados

Tipo de variable: int

ValorDia

Tipo de variable: int

SalarioP

Tipo de variable: string

parl

Tipo de variable: string

psalud

Tipo de variable: int

ppension

Tipo de variable: int

psubtrasporte

Tipo de variable: int

rsubtrasporte

Tipo de variable: int

retencion	Tipo de variable: int
pagototal	Tipo de variable: int
rRetencion	Tipo de variable: int
rsalid	Tipo de variable: int
rpesion	Tipo de variable: int
rsalud	Tipo de variable: int
rarl	Tipo de variable: int
rtotal	Tipo de variable: int

```
function salario(){
  let DiasTrabajados = parseInt(document.getElementById('txtNumeroUno').value);
  let valorDia = parseInt(document.getElementById('txtNumeroDos').value);
  let salarioP
  let Par1
  let Psalud
  let Ppension
  let rsubtrasporte
  let Psubtrasporte
  let retencion
  let pagoTotal
  let rRetencion
  let rsueldo
  let rpension
  let rsalud
  let rarl
  let rtotal

  salarioP = sueldo(DiasTrabajados,valorDia);
  Par1 = arl(salarioP);
  Psalud = salud(salarioP);
  Ppension = pension(salarioP);
  Psubtrasporte = subtras(salarioP);
  retencion = reten(salarioP)
  pagoTotal = pagoT(salarioP)

  rsueldo = `<strong> sueldo de la persona : </strong> ${salarioP} <br>`
  rpension = `<strong> retencion del arl : </strong> ${Par1} <br>`
  rarl = `<strong> retencion de salud : </strong> ${Psalud} <br>`
  rsalud = `<strong> retencion de pension : </strong> ${Ppension} <br>`
  rsubtrasporte = `<strong> subtrasporte : </strong> ${Psubtrasporte} <br>`
  rRetencion = `<strong> retencion del sueldo : </strong> ${retencion} <br>`
  rtotal = `<strong> sueldo total de la persona : </strong> ${pagoTotal}`

  document.getElementById('resultados').innerHTML = rsueldo + rpension + rarl + rsalud + rsubtrasporte + rRetencion + rtotal;

  return false;
}
```

```
<div class="col-10 row forma">
  <div class="formulario col-6">
    <form class="row g-3 needs-validation" novalidate onsubmit="return salario();" >
      <div class="col-md-6">
        <label for="txtNumeroUno" class="form-label">Dias trabajados</label>
        <input type="number" class="form-control" id="txtNumeroUno" required>
        <div class="valid-feedback">
          Looks good!
        </div>
      </div>
      <div class="col-md-6">
        <label for="txtNumeroDos" class="form-label">valor dia </label>
        <input type="number" class="form-control" id="txtNumeroDos" required>
        <div class="valid-feedback">
          Looks good!
        </div>
      </div>
      <div class="col-12">
        <button class="btn btn-info" type="submit">calcular</button>
      </div>
    </form>
  </div>
  <div class="respuesta col-4">
    <div id="resultados"></div>
  </div>
</div>
```

```
// Example starter JavaScript for disabling form submissions if there are invalid fields
(function () {
    'use strict'

    // Fetch all the forms we want to apply custom Bootstrap validation styles to
    var forms = document.querySelectorAll('.needs-validation')

    // Loop over them and prevent submission
    Array.prototype.slice.call(forms)
        .forEach(function (form) {
            form.addEventListener('submit', function (event) {
                if (!form.checkValidity()) {
                    event.preventDefault()
                    event.stopPropagation()
                }

                form.classList.add('was-validated')
            }, false)
        })
})()
```

Días trabajados

30



Looks good!

calcular

valor día

60000



Looks good!

suelto de la persona : 1800000
retencion del arl : 93600
retencion de salud : 216000
retencion de pension : 288000
subtrasporte : 114000
retencion del sueldo : 0
suelto total de la persona : 1316400



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: sueldo(pdiasT , pvalorD)		Versión: 1.0
Descripción: Función que calcula el sueldo de una persona		
diasT	Tipo de variable: int	
valorD	Tipo de variable: int	
pago	Tipo de variable: int	
<pre>function sueldo(pdiasT, pvalorD) { let diaT; //Dias Trabajados let valorD; //Valor por día let pago; diaT = pdiasT; valorD = pvalorD; pago = diaT * valorD; return pago; }</pre>		



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: salud(pago)

Versión: 1.0

Descripción:

Función que calcula cuando dinero debera dar para la salud

salud

Tipo de variable: int

```
function salud(pago) {  
  let saludP = pago * 0.12;  
  return saludP;  
}
```



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: pension(pago)

Versión: 1.0

Descripción:

Función que calcula cuando dinero debera dar para la pension

pensionP

Tipo de variable: int

```
function pension(pago) {  
  let pensionP = pago * 0.16;  
  return pensionP;  
}
```



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: arl(pago)

Versión: 1.0

Descripción:

Función que calcula cuando dinero debera dar para la arl

arlP

Tipo de variable: int

```
function arl(pago) {  
  let arlP = pago * 0.052;  
  return arlP;  
}
```



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: subtras(pago)

Versión: 1.0

Descripción:

Función que calcula si la persona recibe o no recibe subtrasporte

salarioM

Tipo de variable: int

trans

Tipo de variable: int

```
function subTras(pago) {  
  let salarioM = 1300000;  
  let trans;  
  if (pago <= salarioM * 2) {  
    trans = 114000;  
  } else {  
    trans = 0;  
  }  
  return trans;  
}
```



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: reten(pago)		Versión: 1.0
Descripción: Función que calcula si a la persona se le deberá retener una parte del sueldo		
salarioM	Tipo de variable: int	
retencion	Tipo de variable: int	
	<pre>function reten(pago) { let salarioM = 1300000; let retencion; if (pago > salarioM * 4) { retencion = pago * 0.04; } else { retencion = 0; } return retencion; }</pre>	




Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: pagoT(pago)		Versión: 1.0
Descripción: Función que calcula el todo el dinero que deberá recibir la persona		
salario	Tipo de variable: int	
saludE	Tipo de variable: int	
pensionE	Tipo de variable: int	
arIE	Tipo de variable: int	
retencion	Tipo de variable: int	
subtrasporte	Tipo de variable: int	
descuento	Tipo de variable: int	

totalSalario	Tipo de variable: int
<pre>function pagoT(pago) { let salario = pago; let saludE = salud(pago); let pensionE = pension(pago); let arlE = arl(pago); let retencion = reten(pago); let subTrasporte = subTras(pago); let descuento = saludE + pensionE + arlE; let totalSalario = salario + subTrasporte - (retencion + descuento); return totalSalario; }</pre>	

	Tecnólogo en Análisis y Desarrollo de Software Ficha
	Funciones JS

Nombre de la función: porcentaje()		Versión: 1.0
Descripción: Función que permite calcular la nota final de una persona		
nota1	Tipo de variable: int	
nota2	Tipo de variable: int	
nota3	Tipo de variable: string	
notaFinal	Tipo de variable: string	
rnotaFinal	Tipo de variable: int	
PromedioNota1	Tipo de variable: int	
PromedioNota2	Tipo de variable: int	
PromedioNota3	Tipo de variable: int	
Rpromedio1	Tipo de variable: int	
Rpromedio2	Tipo de variable: int	
Rpromedio3	Tipo de variable: int	
validarNotas	Tipo de variable: int	


```

function porcentaje(){
  let nota1 = parseFloat(document.getElementById("txtNumeroUno").value)
  let nota2 = parseFloat(document.getElementById("txtNumeroDos").value)
  let nota3 = parseFloat(document.getElementById("txtNumeroTres").value)
  let notaFinal;
  let rnotaFinal;
  let promedioNota1;
  let promedioNota2;
  let promedioNota3;
  let rpromedio1
  let rpromedio2
  let rpromedio3
  let validarNotas

  promedioNota1 = nota1 * 0.2
  promedioNota2 = nota2 * 0.35
  promedioNota3 = nota3 * 0.45
  notaFinal = promedioNota1 + promedioNota2 + promedioNota3
  validarNotas = validar(notaFinal)

  if(promedioNota1 && promedioNota2 && promedioNota3){
    rpromedio1 = `<strong>primera Primera Nota : ${promedioNota1}</strong> <br>`
    rpromedio2 = `<strong>segunda Segunda Nota : ${promedioNota2}</strong> <br>`
    rpromedio3 = `<strong>tercera Tercera Nota : ${promedioNota3}</strong> <br>`
    rnotaFinal = `<strong>Nota final : ${notaFinal} <br> ${validarNotas}</strong>`

    document.getElementById('resultados').innerHTML = rpromedio1 + rpromedio2 + rpromedio3 + rnotaFinal;
  }
  else{
    document.getElementById('resultados').innerHTML = "<strong> rellena las casillas </strong>";
  }
  return false
}

<div class="container">
  <div class="col-10 row forma">
    <div class="Formulario col-6">

      <form class="row g-3 needs-validation" novalidate onsubmit="return porcentaje();">
        <div class="col-md-6">
          <label for="txtNumeroUno" class="form-label">Primera Nota</label>
          <input type="number" class="form-control" id="txtNumeroUno" required>
          <div class="valid-feedback">
          </div>
        </div>
        <div class="col-md-6">
          <label for="txtNumeroDos" class="form-label">segunda Nota </label>
          <input type="number" class="form-control" id="txtNumeroDos" required>
          <div class="valid-feedback">
          </div>
        </div>
        <div class="col-md-6">
          <label for="txtNumeroTres" class="form-label">tercer nota </label>
          <input type="number" class="form-control" id="txtNumeroTres" required>
          <div class="valid-feedback">
          </div>
        </div>
        <div class="col-12">
          <button class="btn btn-info" type="submit">calcular</button>
        </div>
      </form>
    </div>
  </div>
</div>

// Example starter JavaScript for disabling form submissions if there are invalid fields
(function () {
  'use strict'


  // Fetch all the forms we want to apply custom Bootstrap validation styles to
  var forms = document.querySelectorAll('.needs-validation')

  // Loop over them and prevent submission
  Array.prototype.slice.call(forms)
    .forEach(function (form) {
      form.addEventListener('submit', function (event) {
        if (!form.checkValidity()) {
          event.preventDefault()
          event.stopPropagation()
        }

        form.classList.add('was-validated')
      }, false)
    })
})();

```

<div> <div>Primera Nota</div> <div> <input type="text" value="4.0"/> ✓ </div> </div> <div> <div>segunda Nota</div> <div> <input type="text" value="1.0"/> ✓ </div> </div> <div> <div>tercer nota</div> <div> <input type="text" value="5.0"/> ✓ </div> </div> <div> <div>calcular</div> </div>	<div> <div>primera Primera Nota : 0.8</div> <div>segunda Segunda Nota : 0.35</div> <div>tercera Tercera Nota : 2.25</div> <div>Nota final : 3.4</div> <div>La calificacion es Media 3.4</div> </div>
---	--

	Tecnólogo en Análisis y Desarrollo de Software Ficha
	Funciones JS

Nombre de la función: validar(psuma)		Versión: 1.0
Descripción: Función valida si la nota es superior, buena, media o mala		
suma	Tipo de variable: int	
<pre>function validar(psuma) { let suma; 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>		



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS


Nombre de la función: contadorNumeros()		Versión: 1.0
Descripción: Función contar del 1 hasta el número que desees		
NumeroUno	Tipo de variable: int	
contador	Tipo de variable: int	
<pre>function contadorNumeros(){ let NumeroUno = parseInt(document.getElementById('txtNumeroUno').value); let contador if(NumeroUno){ contador = contar(NumeroUno) document.getElementById('resultados').innerHTML = `Contador : \${contador}`; } else{ document.getElementById('resultados').innerHTML = `debes digitar algun numero` } return false; }</pre> <pre><div class="container"> <div class="col-10 row forma"> <div class="formulario col-6"> <form class="row g-3 needs-validation" novalidate onsubmit="return contadorNumeros();"> <div class="col-md-6"> <label for="txtNumeroUno" class="form-label">digita un numero</label> <input type="number" class="form-control" id="txtNumeroUno" required> <div class="valid-feedback"> </div> </div> <div class="col-12"> <button class="btn btn-info" type="submit">calcular</button> </div> </form> </div> <div class="respuesta col-4"> <div id="resultados"></div> </div> </div> </div></pre> <pre>// Example starter JavaScript for disabling form submissions if there are invalid fields (function () { 'use strict' // Fetch all the forms we want to apply custom Bootstrap validation styles to var forms = document.querySelectorAll('.needs-validation') // Loop over them and prevent submission Array.prototype.slice.call(forms) .forEach(function (form) { form.addEventListener('submit', function (event) { if (!form.checkValidity()) { event.preventDefault() event.stopPropagation() } form.classList.add('was-validated') }, false) }) })();</pre>		

digita un numero


✓

calcular

Contador : 1 2 3 4 5 6 7 8 9 10 11 12
13 14 15 16 17 18 19 20 21 22 23 24
25 26 27 28 29 30

	Tecnólogo en Análisis y Desarrollo de Software Ficha
	Funciones JS

Nombre de la función: contar(pnumero)		Versión: 1.0
Descripción: Función esta es la función que permite contar hasta que numero que tu quieras		
numero	Tipo de variable: int	
contador	Tipo de variable: int	
resultado	Tipo de variable: int	
<pre>function contar(pnumero){ let numero numero = pnumero; let contador = 0; let resultado = ""; while (contador < numero) { contador = contador + 1; resultado += contador + "\n"; } return resultado; }</pre>		

	Tecnólogo en Análisis y Desarrollo de Software Ficha
	Funciones JS

Nombre de la función: factorialNum()	Versión: 1.0
Descripción: Función que permite sacar el factorail de cada numero	

NumeroUno	Tipo de variable: int
factorial	Tipo de variable: int

```
function FactorialNum(){
    let NumeroUno = parseInt(document.getElementById('txtNumeroUno').value);
    let factorial

    if(NumeroUno ){
        factorial = factor(NumeroUno)

        document.getElementById('resultados').innerHTML = `<strong>Factorial : ${factorial}</strong>`;
    }
    else{
        document.getElementById('resultados').innerHTML = `<strong>debes digitar algun numero</strong>`
    }
    return false;
}
```

```
<div class="container">
  <div class="col-10 row forma">
    <div class="formulario col-6">
      <form class="row g-3 needs-validation" novalidate onsubmit="return FactorialNum();">
        <div class="col-md-6">
          <label for="txtNumeroUno" class="form-label">digita un numero</label>
          <input type="number" class="form-control" id="txtNumeroUno" required>
          <div class="valid-feedback">
          </div>
        </div>
        <div class="col-12">
          <button class="btn btn-info" type="submit">calcular</button>
        </div>
      </form>
    </div>
```

```
// Example starter JavaScript for disabling form submissions if there are invalid fields
(function () {
  'use strict'

  // Fetch all the forms we want to apply custom Bootstrap validation styles to
  var forms = document.querySelectorAll('.needs-validation')

  // Loop over them and prevent submission
  Array.prototype.slice.call(forms)
    .forEach(function (form) {
      form.addEventListener('submit', function (event) {
        if (!form.checkValidity()) {
          event.preventDefault()
          event.stopPropagation()
        }

        form.classList.add('was-validated')
      }, false)
    })
})()
```

digita un numero

calcular

Factorial : 120



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: factor(pnumero)		Versión: 1.0
Descripción: Función esta es la función que permite sacar el factorial de cualquier numero		
numero	Tipo de variable: int	
contador	Tipo de variable: int	
factorial	Tipo de variable: int	
<pre>let numero; function factor(pnumero){ numero = pnumero; let factorial; let contador; contador = 0; factorial = 1; while(contador < numero){ contador = contador+1; factorial =factorial*contador; } return factorial; }</pre>		



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: tablaExp()		Versión: 1.0
Descripción: Función permite realizar la tabla de multiplicar que tu quieras		
tablaMul	Tipo de variable: int	
LimiteMul	Tipo de variable: int	
tablaMultiplicar	Tipo de variable: string	

```

function tablaExp(){
    let tablaMul = parseInt(document.getElementById('tabla').value);
    let limiteMul = parseInt(document.getElementById('Limite').value);
    let tablaMultiplicar

    if(tablaMul && limiteMul ){
        tablaMultiplicar = tabla(tablaMul , limiteMul)

        document.getElementById('resultados').innerHTML = `<strong> ${tablaMultiplicar}</strong>`;
    }
    else{
        document.getElementById('resultados').innerHTML = `<strong>debes digitar algun numero</strong>`;
    }
    return false;
}

<div class="formulario col-6">

    <form class="row g-3 needs-validation" novalidate onsubmit="return tablaExp();">
        <div class="col-md-6">
            <label for="tabla" class="form-label">digita La tabla de multiplicar</label>
            <input type="number" class="form-control" id="tabla" required>
            <div class="valid-feedback">
            </div>
        </div>
        <div class="col-md-6">
            <label for="Limite" class="form-label">Limite de la tabla de multiplicar</label>
            <input type="number" class="form-control" id="Limite" required>
            <div class="valid-feedback">
            </div>
        </div>
        <div class="col-12">
            <button class="btn btn-info" type="submit">calcular</button>
        </div>
    </form>
</div>

// Example starter JavaScript for disabling form submissions if there are invalid fields
(function () {
    'use strict'

    // Fetch all the forms we want to apply custom Bootstrap validation styles to
    var forms = document.querySelectorAll('.needs-validation')

    // Loop over them and prevent submission
    Array.prototype.slice.call(forms)
        .forEach(function (form) {
            form.addEventListener('submit', function (event) {
                if (!form.checkValidity()) {
                    event.preventDefault()
                    event.stopPropagation()
                }

                form.classList.add('was-validated')
            }, false)
        })
    })()

```

digita La tabla de multiplicar

4



Limite de la tabla de multiplicar

7



calcular

$$4 \times 1 = 4$$

$$4 \times 2 = 8$$

$$4 \times 3 = 12$$

$$4 \times 4 = 16$$

$$4 \times 5 = 20$$

$$4 \times 6 = 24$$

$$4 \times 7 = 28$$



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: tabla(ptablas , plimite)		Versión: 1.0
Descripción: Función esta es la función que permite sacar el factorial de cualquier numero		
tablas	Tipo de variable: int	
contador	Tipo de variable: int	
factorial	Tipo de variable: int	
resultado	Tipo de variable: int	
msg	Tipo de variable: string	
limite	Tipo de variable: int	

```
let tablas;  
let limite;  
  
function tabla(ptablas,plimite){  
  
    tablas = ptablas;  
    limite = plimite;  
  
    let contador = 0;  
    let resultado;  
    let msg = "";  
  
    while (contador < limite) {  
        contador = contador +1;  
        resultado = tablas * contador;  
  
        msg += `${tablas} x ${contador} = ${resultado} <br>`  
    }  
  
    return msg;  
}
```




Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: tablaExp()		Versión: 1.0
Descripción: Función permite realizar la tabla de multiplicar que tu quieras, aparte te dice cuales son pares e impares		
tablaMul	Tipo de variable: int	
LimiteMul	Tipo de variable: int	
tablaMultiplicar	Tipo de variable: string	

```
function tablaExp(){
  let tablaMul = parseInt(document.getElementById('tabla').value);
  let limiteMul = parseInt(document.getElementById('Limite').value);
  let tablaMultiplicar

  if(tablaMul && limiteMul ){
    tablaMultiplicar = tabla(tablaMul , limiteMul)

    document.getElementById('resultados').innerHTML = `<strong> ${tablaMultiplicar}</strong>`;
  }
  else{
    document.getElementById('resultados').innerHTML = `<strong>debes digitar algun numero</strong>`
  }
  return false;
}
```

```
<div class="container">
  <div class="col-10 row forma">
    <div class="formulario col-6">

      <form class="row g-3 needs-validation" novalidate onsubmit="return tablaExp();">
        <div class="col-md-6">
          <label for="tabla" class="form-label">digita la tabla de multiplicar</label>
          <input type="number" class="form-control" id="tabla" required>
          <div class="valid-feedback">
        </div>
        </div>
        <div class="col-md-6">
          <label for="limite" class="form-label">limite de la tabla de multiplicar</label>
          <input type="number" class="form-control" id="limite" required>
          <div class="valid-feedback">
        </div>
        </div>
        <div class="col-12">
          <button class="btn btn-info" type="submit">calcular</button>
        </div>
      </form>
    </div>

    <div class="respuesta col-4">
      <div id="resultados"></div>
    </div>
  </div>
</div>
```

```
// Example starter JavaScript for disabling form submissions if there are invalid fields
(function () {
  'use strict'

  // Fetch all the forms we want to apply custom Bootstrap validation styles to
  var forms = document.querySelectorAll('.needs-validation')

  // Loop over them and prevent submission
  Array.prototype.slice.call(forms)
    .forEach(function (form) {
      form.addEventListener('submit', function (event) {
        if (!form.checkValidity()) {
          event.preventDefault()
          event.stopPropagation()
        }

        form.classList.add('was-validated')
      }, false)
    })
})();
```

digita la tabla de multiplicar limite de la tabla de multiplicar





calcular

2 x 1 = 2

par

2 x 2 = 4

par

2 x 3 = 6

par

2 x 4 = 8

par

2 x 5 = 10

par



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: tabla(ptablas, plimite)		Versión: 1.0
Descripción: Función esta es la función que permite realizar las multiplicaciones de cualquier numero		
tablas	Tipo de variable: int	
contador	Tipo de variable: int	
resultado	Tipo de variable: int	
msg	Tipo de variable: string	
limite	Tipo de variable: int	

```
let resultado;  
  
function tabla(ptablas,plimite){  
    let tablas;  
    let limite  
    tablas = ptablas;  
    limite = plimite  
    let contador;  
    let msg="";  
    contador = 0;  
    msg = "";  
  
    while (contador<limite) {  
        contador = contador +1  
        resultado = tablas * contador;  
  
        msg += `${tablas} x ${contador} = ${resultado} <br> `;  
        msg += validar(resultado)  
    }  
  
    return msg;  
}
```



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: validar(resultado)

Versión: 1.0

Descripción:

Función esta es la función que permite calcular si los numeros son pares o impares

resultado

Tipo de variable: int

```
function validar(resultado){  
    if(resultado %2==0){  
        return 'par <br>';  
    }else{  
        return 'impar <br>'  
    }  
}
```



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: contadorNumeros()

Versión: 1.0

Descripción:

Función permite realizar el conteo de cualquier numero

NumeroUno

Tipo de variable: int

contadorNum

Tipo de variable: string

```
function contadorNumeros(){
  let NumeroUno = parseInt(document.getElementById('contador').value);
  let contadorNum

  if(NumeroUno ){
    contadorNum = contar(NumeroUno)

    document.getElementById('resultados').innerHTML = `<strong> ${contadorNum}</strong>`;
  }
  else{
    document.getElementById('resultados').innerHTML = `<strong>debes digitar algun numero</strong>`
  }
  return false;
}
```

```
<div class="col-12 row-forma">
  <div class="formulario col-6">

    <form class="row g-3 needs-validation" novalidate onsubmit="return contadorNumeros();">
      <div class="col-md-6">
        <label for="contador" class="form-label">Digita algun Numero</label>
        <input type="number" class="form-control" id="contador" required>
        <div class="valid-feedback">
        </div>
      </div>
      <div class="col-12">
        <button class="btn btn-info" type="submit">calcular</button>
      </div>
    </form>
  </div>

  <div class="respuesta col-4">
    <div id="resultados"></div>
  </div>
</div>
```

```
// Example starter JavaScript for disabling form submissions if there are invalid fields
(function () {
  'use strict'

  // Fetch all the forms we want to apply custom Bootstrap validation styles to
  var forms = document.querySelectorAll('.needs-validation')

  // Loop over them and prevent submission
  Array.prototype.slice.call(forms)
    .forEach(function (form) {
      form.addEventListener('submit', function (event) {
        if (!form.checkValidity()) {
          event.preventDefault()
          event.stopPropagation()
        }

        form.classList.add('was-validated')
      }, false)
    })
})();
```

Digita algun Numero

calcular

1
2
3
4
5
6
7
8
9
10



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: contar(pnumero)		Versión: 1.0
Descripción: Función esta es la función que permite realizar el conteo a los numeros		
numero	Tipo de variable: int	
contador	Tipo de variable: int	
resultado	Tipo de variable: string	
	<pre>let numero; function contar(pnumero){ numero = pnumero; let contador; let resultado = ""; for(contador = 1; numero >= contador; contador++){ resultado += contador+ "
"; } return resultado; }</pre>	



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: FactorialNum()		Versión: 1.0
Descripción: Función permite mostrar y sacar el factorial de cualquier Numero		
FactorialNumero	Tipo de variable: int	
FactorialNumber	Tipo de variable: int	

```
function FactorialNum(){
  let FactorialNumero = parseInt(document.getElementById('Factorial').value);
  let FactorialNumber

  if(FactorialNumero ){
    FactorialNumber = factor(FactorialNumero)

    document.getElementById('resultados').innerHTML = `<strong> ${FactorialNumber}</strong>`;
  }
  else{
    document.getElementById('resultados').innerHTML = `<strong>debes digitar algun numero</strong>`
  }
  return false;
}
```

```
<div class="container">
  <div class="col-10 row forma">
    <div class="formulario col-6">
      <form class="row g-3 needs-validation" novalidate onsubmit="return FactorialNum();">
        <div class="col-md-6">
          <label for="Factorial" class="form-label">Digita algun Numero</label>
          <input type="number" class="form-control" id="Factorial" required>
          <div class="valid-feedback">
          </div>
        </div>
        <div class="col-12">
          <button class="btn btn-info" type="submit">calcular</button>
        </div>
      </form>
    </div>
    <div class="respuesta col-4">
      <div id="resultados"></div>
    </div>
  </div>
</div>
```

```
// Example starter JavaScript for disabling form submissions if there are invalid fields
(function () {
  'use strict'

  // Fetch all the forms we want to apply custom Bootstrap validation styles to
  var forms = document.querySelectorAll('.needs-validation')

  // Loop over them and prevent submission
  Array.prototype.slice.call(forms)
    .forEach(function (form) {
      form.addEventListener('submit', function (event) {
        if (!form.checkValidity()) {
          event.preventDefault()
          event.stopPropagation()
        }

        form.classList.add('was-validated')
      }, false)
    })
})();
```

Digita algun Numero

calcular

120



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: factor(pnumero)		Versión: 1.0
Descripción: Esta es la función que saca el factorial		
numero	Tipo de variable: int	
factorial	Tipo de variable: int	
resultado	Tipo de variable: int	
	<pre>let numero; function factor(pnumero){ numero = pnumero; let factorial = 1; let contador; for(contador = 1; numero >= contador; contador++){ factorial = factorial * contador; } return factorial; }</pre>	



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: TablaFor()		Versión: 1.0
Descripción: Función permite mostrar cualquier tabla de multiplicar que quieras		
tablaMul	Tipo de variable: int	
LimiteMul	Tipo de variable: int	
tablaMultiplicar	Tipo de variable: int	


```

function TablaFor(){
  let tablaMul = parseInt(document.getElementById('tabla').value);
  let limiteMul = parseInt(document.getElementById('Limite').value);
  let tablaMultiplicar

  if(tablaMul && limiteMul ){
    tablaMultiplicar = tablas(tablaMul , limiteMul)

    document.getElementById('resultados').innerHTML = `<strong> ${tablaMultiplicar}</strong>`;
  }
  else{
    document.getElementById('resultados').innerHTML = `<strong> debes digitar algun numero</strong>`
  }
  return false;
}

<div class="container">
  <div class="col-10 row forma">
    <div class="formulario col-6">

      <form class="row g-3 needs-validation" novalidate onsubmit="return TablaFor();">
        <div class="col-md-6">
          <label for="tabla" class="form-label">digita La tabla de multiplicar</label>
          <input type="number" class="form-control" id="tabla" required>
          <div class="valid-feedback">
            </div>
        </div>
        <div class="col-md-6">
          <label for="Limite" class="form-label">limite de la tabla de multiplicar</label>
          <input type="number" class="form-control" id="Limite" required>
          <div class="valid-feedback">
            </div>
        </div>
        <div class="col-12">
          <button class="btn btn-info" type="submit">calcular</button>
        </div>
      </form>
    </div>

    // Example starter JavaScript for disabling form submissions if there are invalid fields
    (function () {
      'use strict'

      // Fetch all the forms we want to apply custom Bootstrap validation styles to
      var forms = document.querySelectorAll('.needs-validation')

      // Loop over them and prevent submission
      Array.prototype.slice.call(forms)
        .forEach(function (form) {
          form.addEventListener('submit', function (event) {
            if (!form.checkValidity()) {
              event.preventDefault()
              event.stopPropagation()
            }

            form.classList.add('was-validated')
          }, false)
        })
    })()
  
```

digita La tabla de multiplicar



Limite de la tabla de multiplicar



calcular

$$4 \times 1 = 4$$

$$4 \times 2 = 8$$

$$4 \times 3 = 12$$


$$4 \times 4 = 16$$

$$4 \times 5 = 20$$

$$4 \times 6 = 24$$

$$4 \times 7 = 28$$

Nombre de la función: tablas(pnumero)		Versión: 1.0
Descripción: Esta es la función que se encarga del proceso de la multiplicacion		
tablaMulti	Tipo de variable: int	
limite	Tipo de variable: int	
contador	Tipo de variable: int	
resultado	Tipo de variable: int	
msg	Tipo de variable: string	
<pre>let tablaMulti; let limite; function tablas(ptablas,plimite){ tablaMulti = ptablas; limite = plimite; let contador; let resultado; let msg = ""; for(contador=1; limite >= contador; contador ++){ resultado = tablaMulti * contador; msg += `\${tablaMulti} x \${contador} = \${resultado}
`; } return msg; }</pre>		

	Tecnólogo en Análisis y Desarrollo de Software Ficha
	Funciones JS

Nombre de la función: TablaFor()		Versión: 1.0
Descripción: Función permite mostrar cualquier tabla de multiplicar que quieras y aparte te dice si es numero es par o impar		
tablaMul	Tipo de variable: int	
LimiteMul	Tipo de variable: int	
tablaMultiplicar	Tipo de variable: int	

```

function TablaFor(){
  let tablaMul = parseInt(document.getElementById('tabla').value);
  let LimiteMul = parseInt(document.getElementById('Limite').value);
  let tablaMultiplicar

  if(tablaMul && LimiteMul ){
    tablaMultiplicar = tablas(tablaMul , LimiteMul)

    document.getElementById('resultados').innerHTML = `<strong> ${tablaMultiplicar}</strong>`;
  }
  else{
    document.getElementById('resultados').innerHTML = `<strong> debes digitar algun numero</strong>`
  }
  return false;
}

<body>
  <div class="container">
    <div class="col-10 row forma">
      <div class="formulario col-6">
        <form class="row g-3 needs-validation" novalidate onsubmit="return TablaFor();">
          <div class="col-md-6">
            <label for="tabla" class="form-label">digita la tabla de multiplicar</label>
            <input type="number" class="form-control" id="tabla" required>
            <div class="valid-feedback">
            </div>
          </div>
          <div class="col-md-6">
            <label for="limite" class="form-label">Limite de la tabla de multiplicar</label>
            <input type="number" class="form-control" id="Limite" required>
            <div class="valid-feedback">
            </div>
          </div>
        </form>
        <div class="col-12">
          <button class="btn btn-info" type="submit">calcular</button>
        </div>
      </div>
      <div class="respuesta col-4">
        <div id="resultados"></div>
      </div>
    </div>
  </div>

```

```

// Example starter JavaScript for disabling form submissions if there are invalid fields
(function () {
  'use strict'

  // Fetch all the forms we want to apply custom Bootstrap validation styles to
  var forms = document.querySelectorAll('.needs-validation')

  // Loop over them and prevent submission
  Array.prototype.slice.call(forms)
    .forEach(function (form) {
      form.addEventListener('submit', function (event) {
        if (!form.checkValidity()) {
          event.preventDefault()
          event.stopPropagation()
        }

        form.classList.add('was-validated')
      }, false)
    })
})();

```

digita la tabla de multiplicar



Limite de la tabla de multiplicar



calcular

4 x 1 = 4 par

4 x 2 = 8 par

4 x 3 = 12 par

Nombre de la función: tablas(ptablas , plimite)		Versión: 1.0
Descripción: Esta es la función que se encarga del proceso de la multiplicacion		
tablaMulti	Tipo de variable: int	
limite	Tipo de variable: int	
contador	Tipo de variable: int	
resultado	Tipo de variable: int	
msg	Tipo de variable: string	
<pre>let tablaMulti; let limite; function tablas(ptablas,plimite){ tablaMulti = ptablas; limite = plimite; let contador; let resultado; let msg = ""; for(contador=1; limite >= contador; contador ++){ resultado = tablaMulti * contador; msg += `\${tablaMulti} x \${contador} = \${resultado} `; msg+= validar(resultado); } return msg; }</pre>		

Nombre de la función: tablas(resultado)		Versión: 1.0
Descripción: Esta es la función que se encarga del proceso de validar si es par o impar		
resultado	Tipo de variable: int	
<pre>function validar(resultado){ if(resultado%2==0){ return "par
"; }else{ return"impar
"; } }</pre>		



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: TablaFor()		Versión: 1.0
Descripción: Función permite mostrar cualquier tabla de multiplicar que quieras y aparte te dice si es numero es par o impar		
tablaMul	Tipo de variable: int	
LimiteMul	Tipo de variable: int	
tablaMultiplicar	Tipo de variable: int	
finish	Tipo de variable: int	

```
function TablaFor(){
    let tablaMul = parseInt(document.getElementById('tabla').value);
    let limiteMul = parseInt(document.getElementById('Limite').value);
    let finish = parseInt(document.getElementById('fin').value);
    let tablaMultiplicar

    if(tablaMul && limiteMul && finish){
        tablaMultiplicar = tablas(tablaMul , limiteMul, finish)

        document.getElementById('resultados').innerHTML = `<strong> ${tablaMultiplicar}</strong>`;
    }
    else{
        document.getElementById('resultados').innerHTML = `<strong> debes digitar algun numero</strong>`;
    }
    return false;
}
```

```
<body>
<div class="container">
<div class="col-10 row forma">
<div class="formulario col-6">

    <form class="row g-3 needs-validation" novalidate onsubmit="return TablaFor();">
        <div class="col-md-6">
            <label for="tabla" class="form-label">Inicio de la tabla</label>
            <input type="number" class="form-control" id="tabla" required>
            <div class="valid-feedback">
            </div>
        </div>
        <div class="col-md-6">
            <label for="Limite" class="form-label">tabla de Multiplicar que desea</label>
            <input type="number" class="form-control" id="Limite" required>
            <div class="valid-feedback">
            </div>
        </div>
        <div class="col-md-6">
            <label for="Fin" class="form-label">Hasta que Numero Desea</label>
            <input type="number" class="form-control" id="fin" required>
            <div class="valid-feedback">
            </div>
        </div>
        <div class="col-12">
            <button class="btn btn-info" type="submit">calcular</button>
        </div>
    </form>
</div>

<div class="respuesta col-4">
<div id="resultados"></div>
</div>
</div>
```

```
// Example starter JavaScript for disabling form submissions if there are invalid fields
(function () {
    'use strict'

    // Fetch all the forms we want to apply custom Bootstrap validation styles to
    var forms = document.querySelectorAll('.needs-validation')

    // Loop over them and prevent submission
    Array.prototype.slice.call(forms)
        .forEach(function (form) {
            form.addEventListener('submit', function (event) {
                if (!form.checkValidity()) {
                    event.preventDefault()
                    event.stopPropagation()
                }

                form.classList.add('was-validated')
            }, false)
        })
})();
```

Inicio de la tabla

tabla de Multiplicar que desea

Hasta que Numero Desea

calcular

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
 total de pares: 23
 total de impares 12

Nombre de la función: tablas(ptablas , plimite, plimiteT)		Versión: 1.0
Descripción: Esta es la función que se encarga del proceso de la multiplicacion		
limites	Tipo de variable: int	
limiteT	Tipo de variable: int	
contador	Tipo de variable: int	
tabla	Tipo de variable: int	
msg	Tipo de variable: string	
<pre>function tablas(ptabla, plimite,plimiteT) { let limite; let limiteT; let tabla; let contador; let msg = ""; tabla = ptabla; limite = plimite; limiteT = plimiteT for (tabla = 1; limite >= tabla; tabla++) { for (contador = 1; limiteT >= contador; contador++) { resultado = tabla * contador; msg += `
 \${tabla} x \${contador} = \${resultado}`; msg += validar(resultado); } msg += "
"; } return msg + "total de pares: " + par + "
" + "total de impares " + impar; }</pre>		

Nombre de la función: validar(resultado)		Versión: 1.0
Descripción: Esta es la función que se encarga del proceso de validar si es par o impar		
resultado	Tipo de variable: int	
par	Tipo de variable: int	
impar	Tipo de variable: int	
<pre>function validar(resultado){ if (resultado % 2 == 0){ par ++ return " Buzz"; }else{ impar++ return " Bass"; } }</pre>		



Tecnólogo en Análisis y Desarrollo de Software

Ficha

Funciones JS

Nombre de la función: TablaFor()		Versión: 1.0
Descripción: Función permite mostrar cualquier tabla de multiplicar que quieras y aparte te dice si es numero es par o impar		
tablaMul	Tipo de variable: int	
LimiteMul	Tipo de variable: int	
tablaMultiplicar	Tipo de variable: int	
finish	Tipo de variable: int	


```
function TablaWhile(){
    let tablaMul = parseInt(document.getElementById('tabla').value);
    let limiteMul = parseInt(document.getElementById('limite').value);
    let finish = parseInt(document.getElementById('fin').value);
    let tablaMultiplicar

    if(tablaMul && limiteMul && finish ){
        tablaMultiplicar = tablas(tablaMul , limiteMul, finish)

        document.getElementById('resultados').innerHTML = `<strong> ${tablaMultiplicar}</strong>`;
    }
    else{
        document.getElementById('resultados').innerHTML = `<strong> debes digitar algun numero</strong>`
    }
    return false;
}

```

```
<div class="col-12 row form">
  <div class="formulario col-6">

    <form class="row g-3 needs-validation" novalidate onsubmit="return TablaWhile();">
      <div class="col-md-6">
        <label for="tabla" class="form-label">Inicio de la tabla</label>
        <input type="number" class="form-control" id="tabla" required>
        <div class="valid-feedback">
        </div>
      </div>
      <div class="col-md-6">
        <label for="limite" class="form-label">tabla de Multiplicar que desea</label>
        <input type="number" class="form-control" id="limite" required>
        <div class="valid-feedback">
        </div>
      </div>
      <div class="col-md-6">
        <label for="fin" class="form-label">Hasta que Numero Desea</label>
        <input type="number" class="form-control" id="fin" required>
        <div class="valid-feedback">
        </div>
      </div>
      <div class="col-12">
        <button class="btn btn-info" type="submit">calcular</button>
      </div>
    </form>
  </div>

  <div class="respuesta col-4">
    <div id="resultados"></div>
  </div>

```

```
// Example starter JavaScript for disabling form submissions if there are invalid fields
(function () {
    'use strict'

    // Fetch all the forms we want to apply custom Bootstrap validation styles to
    var forms = document.querySelectorAll('.needs-validation')

    // Loop over them and prevent submission
    Array.prototype.slice.call(forms)
        .forEach(function (form) {
            form.addEventListener('submit', function (event) {
                if (!form.checkValidity()) {
                    event.preventDefault()
                    event.stopPropagation()
                }

                form.classList.add('was-validated')
            }, false)
        })
})();

```

Inicio de la tabla

tabla de Multiplicar que desea

Hasta que Numero Desea

calcular

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
 total de pares: 23
 total de impares 12

Nombre de la función: tablas(ptablas , plimite, plimiteT)		Versión: 1.0
Descripción: Esta es la función que se encarga del proceso de la multiplicacion		
limites	Tipo de variable: int	
limiteT	Tipo de variable: int	
contador	Tipo de variable: int	
tabla	Tipo de variable: int	
msg	Tipo de variable: string	
resultado	Tipo de variable: int	
	<pre>function tablas(ptabla, plimite, plimiteT) { let tabla; let limite; let limiteT; let contador; let resultado; let msg = ""; tabla = ptabla; limite = plimite; limiteT = plimiteT; while (tabla < limite) { contador = 0; tabla = tabla + 1; while (contador < limiteT) { contador = contador + 1; resultado = tabla * contador; msg += `
 \${tabla} x \${contador} = \${resultado}`; msg += validar(resultado); } msg += "
"; } return (msg + "
" + "total de pares: " + par + "
" + "total de impares " + impar); }</pre>	

Nombre de la función: validar(resultado)		Versión: 1.0
Descripción: Esta es la función que se encarga del proceso de validar si es par o impar		
resultado	Tipo de variable: int	
par	Tipo de variable: int	
impar	Tipo de variable: int	

```
function validar(resultado) {  
  if (resultado % 2 == 0) {  
    par = par + 1;  
    return " Buzz";  
  } else {  
    impar = impar + 1;  
    return " Bass";  
  }  
}
```