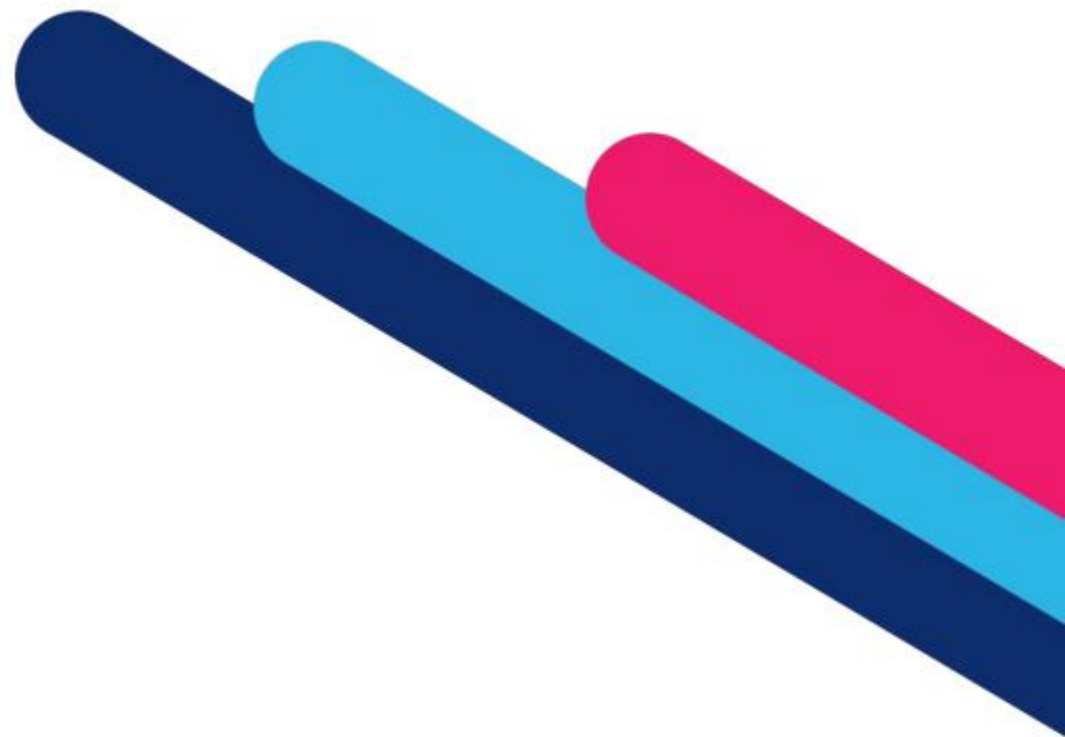




Lenguajes para el desarrollo y soporte de Sistemas

Mg. Remigio Huarcaya Almeyda

SESIÓN 10



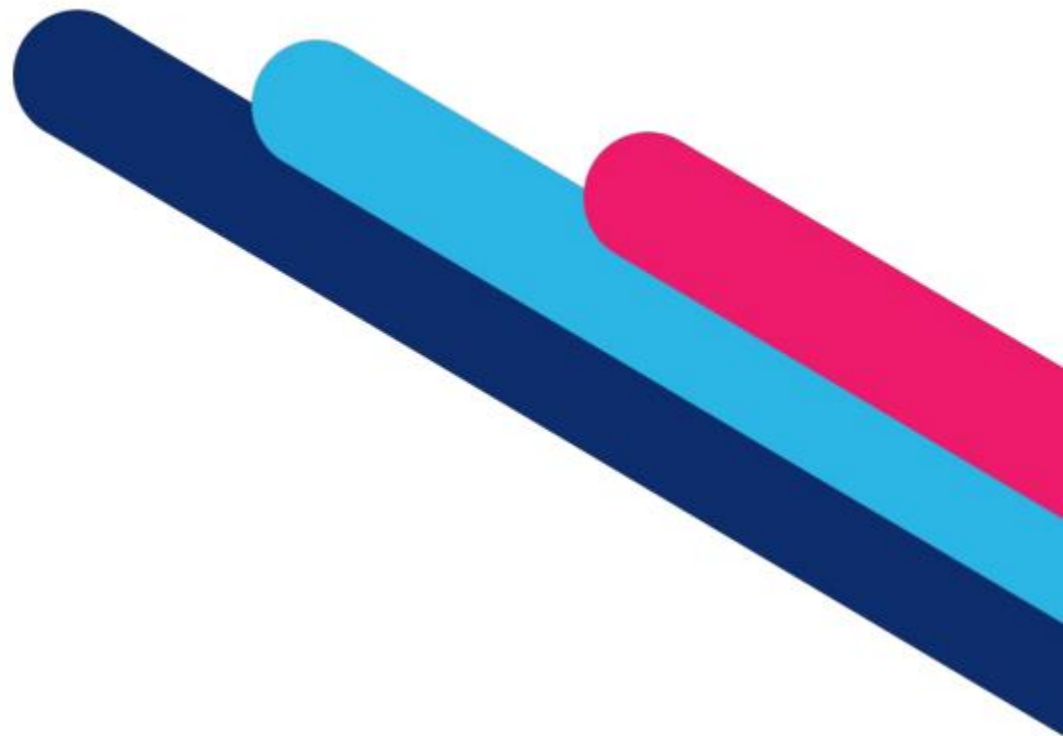


Objetivo

Conocer la estructura **while** del lenguaje JavaScript para crear página web dinámicas

Actividades

2. Análisis de la estructura **while** de javascript
3. Ejercicios de la estructura while
4. Subir al repositorio github





DEMOSTRACIÓN

Contexto del Desarrollo web

Lenguajes para el desarrollo y soporte de Sistemas

Mg. Remigio Huarcaya Almeyda

SESIÓN 10



JavaScript

Lenguajes para el desarrollo y soporte de Sistemas

JavaScript



JavaScript es un **lenguaje de programación** interpretado, de alto nivel y dinámico que se utiliza principalmente para el desarrollo web.

Originalmente fue creado por **Brendan Eich** en 1995 mientras trabajaba en Netscape Communications Corporation.

JavaScript permite a los desarrolladores agregar interactividad y efectos dinámicos a las páginas web

JavaScript - Eventos

Lenguajes para el desarrollo y soporte de Sistemas

JavaScript



Un **evento** es una **acción** o suceso que ocurre en el navegador o en la página web, que puede ser detectado y manejado mediante código.

Los eventos pueden ser generados por acciones del usuario, como hacer **click** en un botón, mover el ratón o presionar una tecla.



JavaScript

Lenguajes para el desarrollo y soporte de Sistemas

Usos



Eventos

Evento	Descripción	Elementos para los que está definido
<code>onblur</code>	Un elemento pierde el foco	<code><button></code> , <code><input></code> , <code><label></code> , <code><select></code> , <code><textarea></code> , <code><body></code>
<code>onchange</code>	Un elemento ha sido modificado	<code><input></code> , <code><select></code> , <code><textarea></code>
<code>onclick</code>	Pulsar y soltar el ratón	Todos los elementos
<code>ondblclick</code>	Pulsar dos veces seguidas con el ratón	Todos los elementos
<code>onfocus</code>	Un elemento obtiene el foco	<code><button></code> , <code><input></code> , <code><label></code> , <code><select></code> , <code><textarea></code> , <code><body></code>
<code>onkeydown</code>	Pulsar una tecla y no soltarla	Elementos de formulario y <code><body></code>
<code>onkeypress</code>	Pulsar una tecla	Elementos de formulario y <code><body></code>
<code>onkeyup</code>	Soltar una tecla pulsada	Elementos de formulario y <code><body></code>
<code>onload</code>	Página cargada completamente	<code><body></code>
<code>onmousedown</code>	Pulsar un botón del ratón y no soltarlo	Todos los elementos
<code>onmousemove</code>	Mover el ratón	Todos los elementos
<code>onmouseout</code>	El ratón "sale" del elemento	Todos los elementos
<code>onmouseover</code>	El ratón "entra" en el elemento	Todos los elementos
<code>onmouseup</code>	Soltar el botón del ratón	Todos los elementos
<code>onreset</code>	Inicializar el formulario	<code><form></code>
<code>onresize</code>	Modificar el tamaño de la ventana	<code><body></code>

JavaScript - Operadores

Lenguajes para el desarrollo y soporte de Sistemas

Operadores lógicos

Operador	Significado
&&	AND (Y lógico)
	OR (O lógico)
!	NOT (NO lógico)

Operadores aritméticos

Operador	Significado
+	Suma
-	Resta
*	Multiplicación
/	Dividir
%	Resto de la división
++	Incremento
--	Decremento

Operadores de comparación

Operador	Significado
==	Igual
!=	Distinto
>=	Mayor o igual
<=	Menor o igual
>	Mayor
<	Menor

Operadores de asignación

Operador	Ejemplo
+=	Suma y asignación
-=	Resta y asignación
*=	Producto y asignación
/=	División y asignación
%=	Resto y asignación

JavaScript – Estructura condicional if

Lenguajes para el desarrollo y soporte de Sistemas

IF básico

```
var variablemostrarMensaje = true;

if (mostrarMensaje){
    alert("Hola Mundo")
}
```

IF - else

```
var edad = 18;

if(condiciónedad >= 18){
    alert("Eres mayor de edad");
} else {
    alert("Todavía eres menor de edad");
}
```

IF anidado

```
let edad = 18;

if(edad < 12){
    alert("Eres pequeño");
} else if ( edad < 19){
    alert("Eres un adolescente");
} else if (edad < 35){
    alert("Aún eres joven")
}
else {
    alert("Piensa en cuidarte")
}
```

JavaScript



JavaScript – Estructura condicional if

Lenguajes para el desarrollo y soporte de Sistemas

IF con operadores ternario

```
let result = (age >= 18) ? "Adult" : "Minor";
```

IF con operadores lógico

```
if (condition1 && condition2) {  
    // código a ejecutar si ambas condiciones son verdaderas  
}  
  
if (condition1 || condition2) {  
    // código a ejecutar si al menos una de las condiciones es verdadera  
}  
  
if (!condition) {  
    // código a ejecutar si la condición es falsa  
}
```

JavaScript



JavaScript – Estructura selectiva : switch

Lenguajes para el desarrollo y soporte de Sistemas

JavaScript



```
Palabra reservada → switch (expresión) {  
    Palabra reservada → case x:   
        //Bloque de código  
    Palabra reservada → break;  
    case y:   
        //Bloque de código  
        break;  
    default:   
        //Bloque de código  
        break;  
}
```

Expresión a evaluar

Si la expresión es igual a **x**, se ejecutará este bloque de código

Si la expresión es igual a **y**, se ejecutará este bloque de código

Si no entramos en ningún caso, se ejecutará este bloque de código

JavaScript – function

Lenguajes para el desarrollo y soporte de Sistemas

Una función es un bloque de código reusable que realiza una tarea específica o calcula un valor.

Las funciones permiten organizar el código de manera modular, haciéndolo más legible, mantenible y reusable

```
function nombreDeLaFunción(parametro1, parametro2){  
    //Código a ejecutar  
}
```

```
//Primero declaramos la función  
function saludo(){  
    alert("Hola y Bienvenido");  
}  
  
//llamada a la función  
saludo();
```

```
function saludar(a,b){  
    return a + b  
}  
  
console.log(sumar(3,4));
```

JavaScript – Estructura condicional if

Lenguajes para el desarrollo y soporte de Sistemas

```
1  body{
2    background-color: #f5deb3;
3  }
4  h1{
5    text-align: center;
6    color: #0000ff;
7  }
8  h2{
9    text-align: center;
10   color: #4169e1;
11 }
12 button{
13   height: 35px;
14   background-color: #6495ed;
15   color: #ffffff;
16   border-radius: 10px;
17 }
18 button:hover{
19   background-color: #000000;
20   color: #008000;
21 }
```

```
<script>
  function calcular(){
    var xres = "";
    var xnota = document.getElementById('txtNota').value;
    if ( parseFloat(xnota) >= 11){
      xres = "Aprobado";
      document.getElementById('txtRes').style.color= "blue";
    }else{
      xres = "Desaprobado";
      document.getElementById('txtRes').style.color= "red";
    }
    document.getElementById('txtRes').value = xres;
  }
  function limpiar(){
    document.getElementById('txtNota').value = "";
    document.getElementById('txtRes').value = "";
    document.getElementById('txtNota').focus(); //Focalizar
  }
</script>
```

JavaScript – aplicación de eventos

Lenguajes para el desarrollo y soporte de Sistemas

```
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>CERTUS | Javascript</title>
7      <script src="../js/s30.js"></script>
8      <link rel="stylesheet" href="../css/s30.css">
9  </head>
10 <body>
11     <h1>JAVASCRIPT</h1>
12     <h2>Evento click</h2>
13     <hr>
14     <div style="margin-bottom: 20px; margin-top: 20px;">
15         <input type="text" name="" id="txtEmpresa" placeholder="ingresa empresa" >
16     </div>
17     <div>
18         <button onclick="saludar()" >Saludo</button>
19         <button onmouseover = "mensaje()" >Frase del día</button>
20         <button onclick="mostrar()" >Empresa</button>
21     </div>
22 </body>
23 </html>
```

A green rectangular box with the word "evento" inside is positioned over the "Saludar()" button in the code. A small arrow points from the box to the button, indicating that an event is being triggered on the button click.

JavaScript - eventos

Lenguajes para el desarrollo y soporte de Sistemas

JavaScript



Tengo el focus, puedes escribir aquí

- *onfocus* Un elemento obtiene el focus
- *onblur* Un elemento pierde el focus

Código usado en el ejemplo anterior:

```
<input type="text" value=""  
onfocus="value='Tengo el focus';style.backgroundColor='yellow';"  
onblur="value='Perdi el focus';style.backgroundColor='#ddd';">
```

CERTUS

Es utilizado el siguiente código:

```
<input type="text" value="onchange" onchange="style.backgroundColor='red';">
```

JavaScript - for

Lenguajes para el desarrollo y soporte de Sistemas



El ciclo for en JavaScript es una estructura que repite un bloque de código según tres expresiones definidas al momento de su declaración.

Estas expresiones controlan la **inicialización**, la **condición** de continuación y el paso de **iteración** del ciclo, ofreciendo un control preciso sobre la ejecución repetida del código.

```
for (expression 1; expression 2; expression 3) {  
    // Bloque de código  
}
```

```
<script>  
    let lenguajes = ['Java', 'C#', 'PHP'];  
    for (let i = 0; i <= 3; i++) {  
        alert(lenguajes[i]);  
    }  
</script>
```



JavaScript - for

Lenguajes para el desarrollo y soporte de Sistemas

```
<script>
    document.write("LENGUAJE DE PROGRAMACIÓN" + "<br>");

    let lenguajes = ['Java', 'C#', 'PHP', 'Go', 'Kotlin'];
    for (let i = 0; i < lenguajes.length; i++) {
        document.write(lenguajes[i]+"<br>");
    }
</script>
```

```
<script>
    //Crear un arreglo
    let perifericos = new Array();
    //Agregar elementos al arreglo
    perifericos[0] = 'Impresora';
    perifericos[1] = 'Scanner';
    perifericos[2] = 'Mouse';

    // Mostrar el arreglo
    let k = 0
    for (k = 0; k <= 2; k++){
        alert(perifericos[k]);
    }
</script>
```




JavaScript - for

Lenguajes para el desarrollo y soporte de Sistemas

```
<script>
    document.write("NÚMERO CONSECUTIVOS" + "<br>");

    for (let i = 1; i <= 10; i++) {
        document.write(i + "<br>");
    }
</script>
```

```
<script>
    document.write("NÚMERO PARES" + "<br>");

    for (let i = 1; i <= 10; i++) {
        document.write(i*2 + "<br>");
    }
</script>
```



JavaScript - while

Lenguajes para el desarrollo y soporte de Sistemas

La estructura **while** es un bucle que permite repetir un bloque de código mientras se cumpla una **condición** específica.

La sintaxis básica del bucle while es la siguiente:

```
while (condición) {  
    // Código javascript a ejecutar  
}
```



JavaScript - while

Lenguajes para el desarrollo y soporte de Sistemas

Ejercicio 01

```
<body>
  <h2>Lista de números consecutivos</h2>
  <hr>
  <script>
    let x = 1;
    while (x <= 10){
      document.write(x + "<br>");
      x = x + 1;
    }
  </script>
</body>
```

Ejercicio 02

```
<body>
  <h2>Suma e imprimirlos pares</h2>
  <script>
    let x = 1;
    let xsuma = 0;
    while (x <= 10){
      if (x % 2 == 0){
        document.write(x + "<br>");
        xsuma = xsuma + x;
      }
      x = x + 1;
    }
    document.write("La suma de pares es : " + xsuma)
  </script>
</body>
```



JavaScript - while

Lenguajes para el desarrollo y soporte de Sistemas

Ejercicio 03

```
<body>
  <h2>Tabla de multiplicar</h2>
  <script>
    let num = parseInt(prompt("Ingrese número"));
    let xn = 1;
    while(xn<=12){
      document.write(num + " x " + xn + " = " + num*xn + "<br>")
      xn=xn+1
    }
  </script>
</body>
```



JavaScript - while

Lenguajes para el desarrollo y soporte de Sistemas

Ejercicio 04

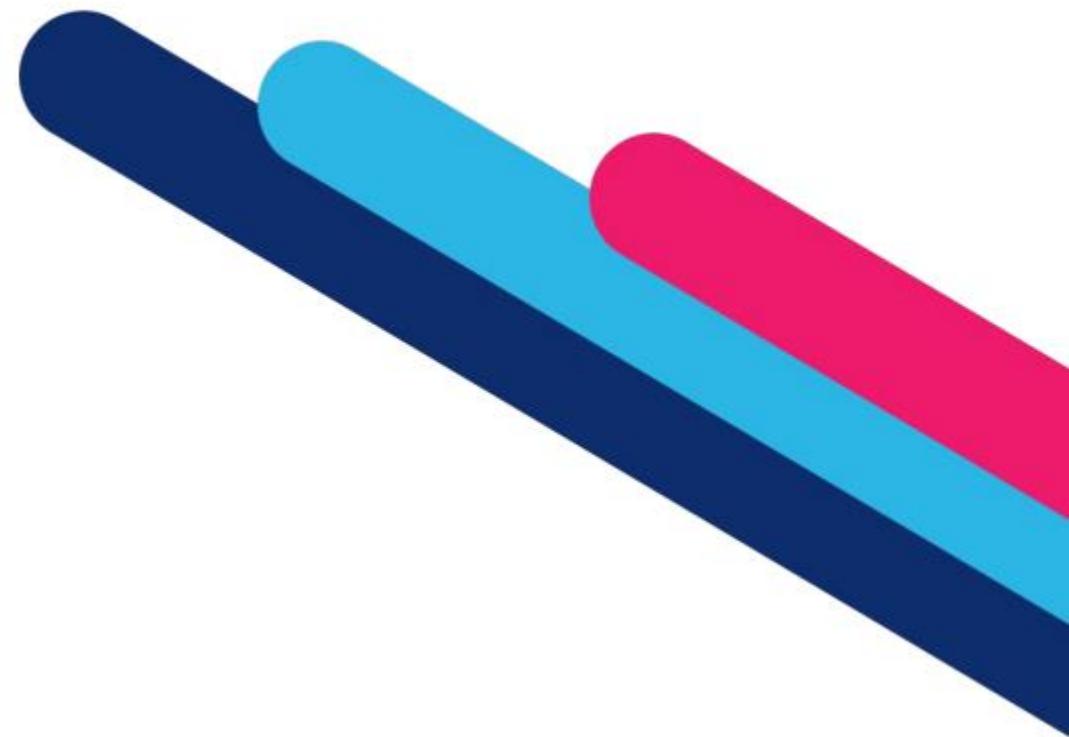
```
<body>
  <h2>Hallar el menor 4 números positivos ingresado</h2>
  <script>
    let xmenor = parseInt(prompt("Ingrese número : "));
    for (let k=1; k<4;k++){
      let num = parseInt(prompt("Ingrese número : "));
      if (num < xmenor){
        xmenor = num;
      }
    }
    document.write("El menor número es : " + xmenor);

    //-----
    // Crear la misma solucion usando al estructura while
    //-----
  </script>
</body>
```



RECURSOS

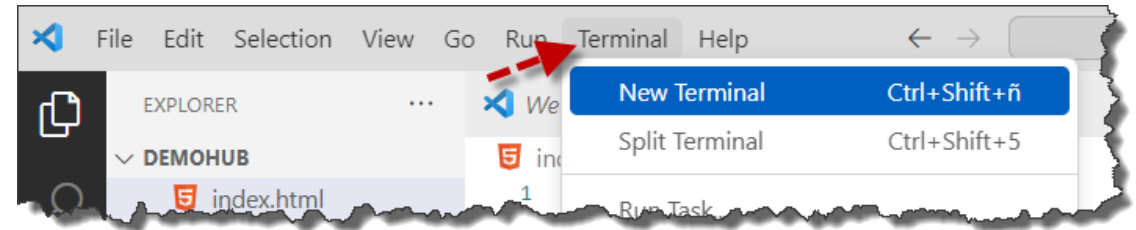
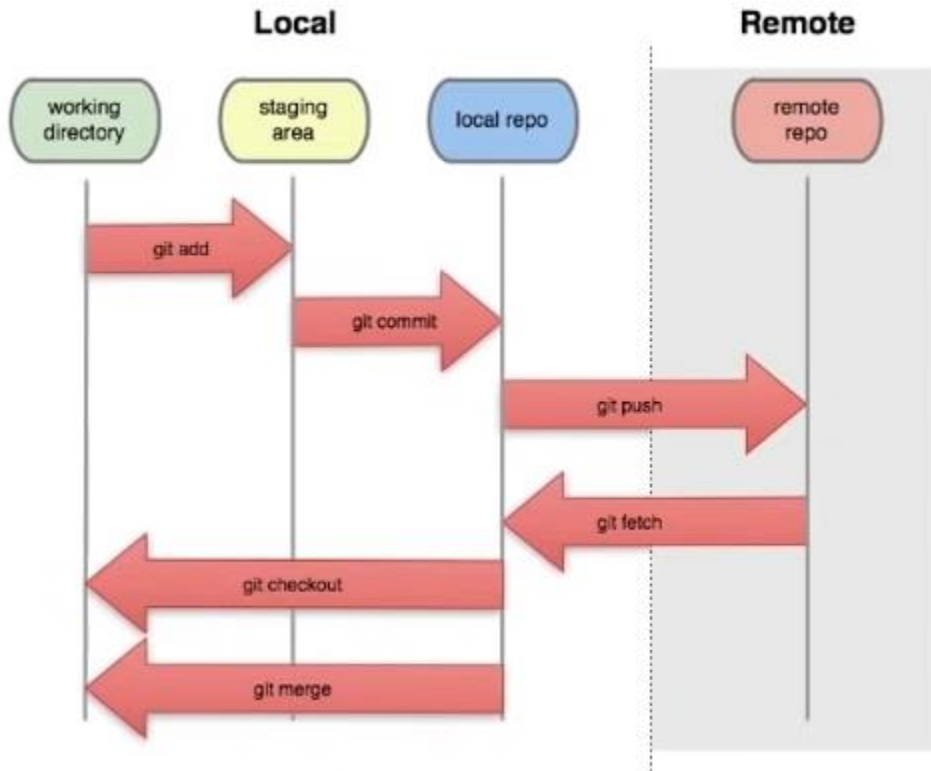
Lenguajes para el desarrollo y soporte de Sistemas



Mg. Remigio Huarcaya Almeyda

SESIÓN 10

Flujo de trabajo de git



Principales comandos git

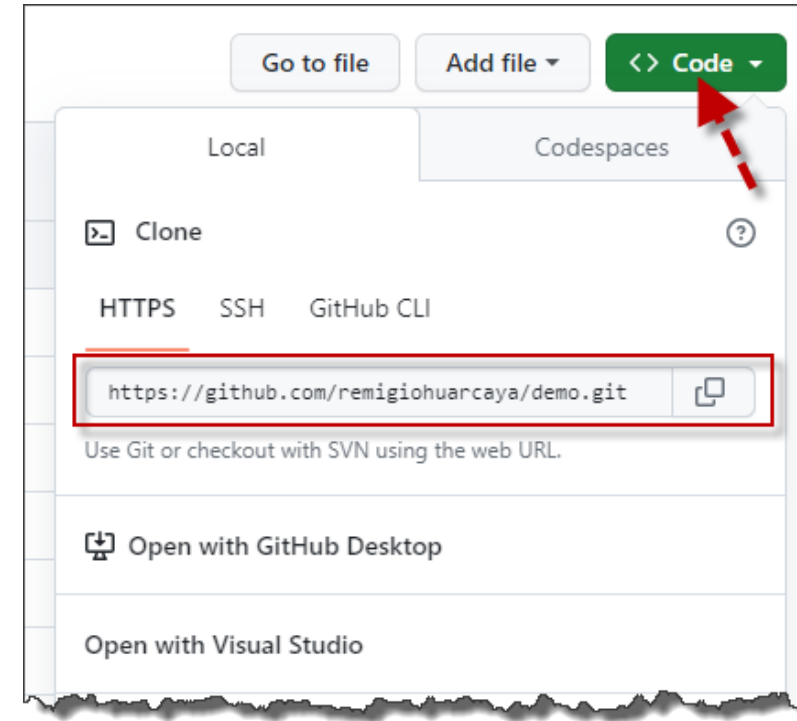
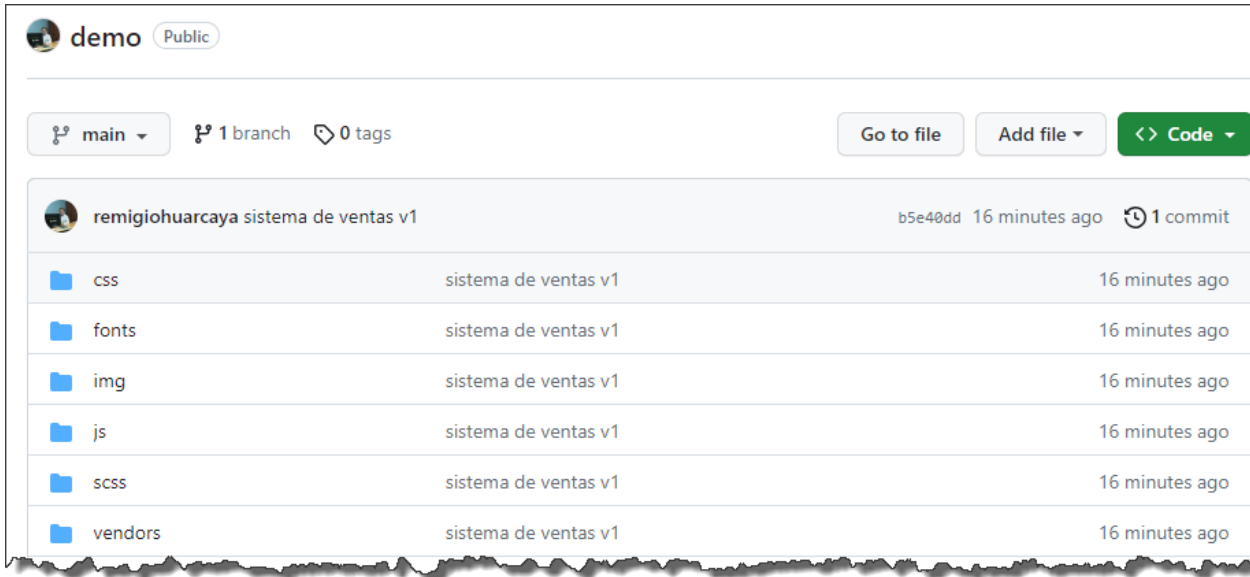
- git init
- git add .
- git log
- git commit -m "mensaje de cambio"
- git push

HTML – Sistema web

DEMOSTRACIÓN

Actividad

Clonar un sistema que se encuentra en el repositorio demo



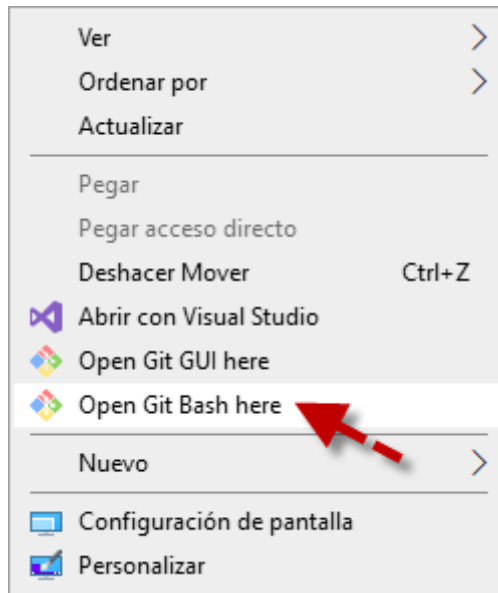
HTML – Sistema web

DEMOSTRACIÓN

Actividad

Clonar un sistema que se encuentra en el repositorio demo

- 01** Hacer click derecho en el lugar que desea clonar y
Luego seleccione **Open Git Bash here**



- 02** Ejecutar **git clone <url de github>**

```
MINGW64:/c/Users/Remigio/Desktop/demo

Remigio@DESKTOP-4C2NKUR MINGW64 ~/Desktop
$ git clone https://github.com/remigiohuarcaya/demo.git
Cloning into 'demo'...
remote: Enumerating objects: 247, done.
remote: Counting objects: 100% (247/247), done.
remote: Compressing objects: 100% (216/216), done.
remote: Total 247 (delta 28), reused 247 (delta 28), pack-reused 0
Receiving objects: 100% (247/247), 7.54 MiB | 410.00 KiB/s, done.
Resolving deltas: 100% (28/28), done.

Remigio@DESKTOP-4C2NKUR MINGW64 ~/Desktop
$ cd demo

Remigio@DESKTOP-4C2NKUR MINGW64 ~/Desktop/demo (main)
$ dir
blog.html      contact_process.php  index.html      single-product.html
cart.html      css                  js              tracking.html
category.html  elements.html        prepros-6.config vendors
checkout.html  fonts                scss
contact.html   img                  single-blog.html
```