

CLASE NRO. 06 – ESTRUCTURAS REPETITIVAS

Tipo de Clase: Teórico Práctica

Duración: 120 minutos

Objetivo:

Al finalizar este capítulo, el alumno debe dominar y tener los conceptos de las tres estructuras repetitivas básicas, estructurales y fundamentales en programación y especialmente en JavaScript. Estas instrucciones son ciclo for, ciclo while y do while. Debe entender y comprender en que momentos se utilizan (ejemplos básicos donde vemos estos temas y logramos comprender la implementación de estas herramientas), las formas de salir de los bucles iterativos como así también iremos incorporando los temas previos vistos con ejercicios de lógica como ser:

- ✓ Declaración e inicialización de variables
- ✓ Estructuras condicionales if/else
- ✓ Estructuras condicionales anidadas
- ✓ Contadores
- ✓ Acumuladores
- ✓ Cálculo de porcentajes sobre el total
- ✓ Salir de estructuras repetitivas con break
- ✓ Saltar una iteración en estructuras repetitivas con continue
- ✓ Estructuras repetitivas
 - o Ciclo for
 - o Ciclo while
 - o Ciclo do while

En JavaScript, las estructuras repetitivas permiten ejecutar un bloque de código varias veces según una condición. Las estructuras repetitivas básicas que veremos son: for, while y do...while.

Ciclo for

El ciclo for se utiliza cuando se conoce de antemano cuántas veces se debe ejecutar un bloque de código. Su sintaxis básica es:

```
for (inicialización; condición; actualización)
{
    // código a ejecutar
}
```

DIPLOMATURA EN DISEÑO WEB FULL STACK CON JAVASCRIPT

MÓDULO 02 - JAVASCRIPT

inicialización: Se ejecuta una vez al comienzo del bucle. Aquí generalmente se inicializa una variable de control.

condición: Se evalúa antes de cada iteración. Si es true, se ejecuta el bloque de código; si es false, el bucle termina.

actualización: Se ejecuta al final de cada iteración. Aquí generalmente se actualiza la variable de control.

Ejemplo:

```
for (let i = 0; i < 5; i++)  
{  
    console.log(i);  
}
```

Ciclo while

El ciclo while se utiliza cuando no se sabe cuántas veces se debe ejecutar el bloque de código y se depende de una condición que puede cambiar durante la ejecución. Su sintaxis básica es:

```
while (condición)  
{  
    // código a ejecutar  
}
```

DIPLOMATURA EN DISEÑO WEB FULL STACK CON JAVASCRIPT

MÓDULO 02 - JAVASCRIPT

condición: Se evalúa antes de cada iteración. Si es true, se ejecuta el bloque de código; si es false, el bucle termina.

Ejemplo:

```
let i = 0;

while (i < 5)

{

    console.log(i);

    i++;

}
```

Ciclo do...while

El ciclo do...while es similar al ciclo while, pero con la diferencia de que el bloque de código se ejecuta al menos una vez antes de evaluar la condición. Su sintaxis básica es:

```
do

{

    // código a ejecutar

} while (condición);
```

condición: Se evalúa después de cada iteración. Si es true, el bloque de código se ejecuta nuevamente; si es false, el bucle termina.

Ejemplo:

```
let i = 0;

do {

    console.log(i);

    i++;

} while (i < 5);
```

DIPLOMATURA EN DISEÑO WEB FULL STACK CON JAVASCRIPT

MÓDULO 02 - JAVASCRIPT

Comparación de las Estructuras Repetitivas

for: Se usa cuando se conoce el número de iteraciones. Es útil para iterar sobre rangos de números o colecciones conocidas.

while: Se usa cuando no se conoce el número de iteraciones y depende de una condición externa. Es más flexible, pero puede ser propenso a bucles infinitos si no se maneja correctamente la condición.

do...while: Se usa cuando se quiere asegurar que el bloque de código se ejecute al menos una vez. Es útil cuando la condición depende de operaciones que se realizan dentro del bucle.

Consideraciones

Bucle Infinito: Un bucle que nunca termina se llama bucle infinito. Es crucial asegurarse de que la condición de salida eventualmente se cumplirá para evitar que el programa se bloquee.

Ejemplo de bucle infinito (evitar):

```
while (true) {  
    console.log("Esto se ejecutará infinitamente");  
}
```

Interrupción de Bucles: La declaración `break` se puede usar para salir de un bucle prematuramente.

Ejemplo con `break`:

```
for (let i = 0; i < 10; i++) {  
    if (i === 5) {  
        break;  
    }  
    console.log(i); // Imprimirá del 0 al 4  
}
```

Continuar en Bucles: La declaración `continue` se usa para saltar a la siguiente iteración del bucle.

DIPLOMATURA EN DISEÑO WEB FULL STACK CON JAVASCRIPT

MÓDULO 02 - JAVASCRIPT

Ejemplo con continue:

```
for (let i = 0; i < 5; i++) {  
    if (i === 3) {  
        continue;  
    }  
    console.log(i); // Imprimirá 0, 1, 2, 4 (se saltará el 3)  
}
```

Conclusión

Las estructuras repetitivas for, while y do...while son fundamentales en JavaScript para ejecutar bloques de código repetidamente bajo diferentes condiciones. Comprender cuándo y cómo usar cada tipo de bucle es crucial para escribir código eficiente y efectivo.

Ejercicio propuesto Nro. 06: (estructuras repetitivas)

Realizar un proceso que permita cargar (tipo caja de supermercado) los productos comprados por los clientes de un pequeño comercio, donde el cajero irá introduciendo el precio de los productos y al final de ello visualizar el Total. Realizar este proceso con una estructura repetitiva para 8 productos.

Solución:

```
/* EJERCICIO NRO. 06  
Realizar un proceso que permita cargar (tipo caja de supermercado)  
los productos comprados por los clientes de un pequeño comercio,  
donde el cajero irá introduciendo el precio de los productos  
y al final de ello visualizar el Total. Realizar este proceso  
con una estructura repetitiva para 8 productos.  
*/  
  
{  
  
    /*  
    total de factura es una variable  
    donde iré acumulando los precios de los  
    productos que voy registrando, debe estar  
    fuera del ciclo repetitivo porque debe  
    ser global al cuerpo del ciclo.  
    */  
}
```

DIPLOMATURA EN DISEÑO WEB FULL STACK CON JAVASCRIPT

MÓDULO 02 - JAVASCRIPT

```
let totalFactura = 0;

for(let i = 1;i <= 8;i=i+1)
{
    /* declaro una variable y la inicializo en 0,
    en esta variable iré solicitando que me ingrese
    el precio de los productos
    Esta variable se declarará e inicializará en cada
    paso del ciclo. en cada iteración.
    */

    let precioProducto = 0;

    /* solicito al usuario ingresar el precio del producto y lo guardo
    justamente en una variable que se llama precioProducto */

    precioProducto = Number(prompt(`Ingrese el Precio del Producto
    ${i}`));

    /* antes de comenzar la siguiente iteración
    la guardo y acumulo en la variable totalFactura */

    totalFactura = totalFactura + precioProducto;

    /* visualizo el producto registrado */
    console.log(`Producto Registrado:= ${i} valor ${precioProducto}`);

}

/* Muestro el total de los Productos */
console.log(`El Total de la Factura:= ${totalFactura}`);
}
```

DIPLOMATURA EN DISEÑO WEB FULL STACK CON JAVASCRIPT

MÓDULO 02 - JAVASCRIPT

Ejercicio propuesto Nro. 07: (estructura repetitiva while)

Realizar un programa en JavaScript que permita ingresar las notas de los trabajos finales de los alumnos de la diplomatura en "Desarrollo Web Full Stack con JavaScript" para ello se establecen las siguientes condiciones.

- No está establecido la cantidad de trabajos finales que se evaluarán
- Este será el cuadro con el que se analizará y asignará la clasificación de los mismos.
 - o Si la nota ≥ 0 y ≤ 4 serán trabajos desaprobados
 - o Si la nota > 4 y ≤ 7 serán trabajos aprobados
 - o Si la nota > 7 y < 10 serán trabajos muy buenos
 - o Si la nota = 10 serán trabajos excelentes
- Contemplar que el operador podría ingresar notas incorrectas, es decir podría poner una nota menor a cero o mayor a 10 con lo que sería claramente un error. Contemplar la cantidad de veces que se equivoca.
- Siempre preguntar si desea continuar cargando notas ¿?.

Solución:

```
let continua = "SI";
let contadorNotas = 0;
let contadorErrores = 0;
let contadorDesaprobados = 0;
let contadorAprobados = 0;
let contadorMuyBuenos = 0;
let contadorExcelentes = 0;

while (continua === "SI")
{
    let notaAlumno = Number (prompt("Ingrese la nota del alumno")) ;

    contadorNotas = contadorNotas + 1;

    if((notaAlumno < 0) || (notaAlumno > 10))
    {
        contadorErrores = contadorErrores + 1;
    }
    else
    {
        if((notaAlumno >= 0) && (notaAlumno <=4))
        {
            contadorDesaprobados = contadorDesaprobados + 1;
        }
        if((notaAlumno >4) && (notaAlumno <=7))
```

DIPLOMATURA EN DISEÑO WEB FULL STACK CON JAVASCRIPT

MÓDULO 02 - JAVASCRIPT

```
{
    contadorAprobados = contadorAprobados + 1;
}
if((notaAlumno >7) && (notaAlumno <10))
{
    contadorMuyBuenos = contadorMuyBuenos + 1;
}
if(notaAlumno ===10)
{
    contadorExcelentes = contadorExcelentes + 1;
}
}
continua = prompt("continua con otro alumno ? (SI/NO)");
}

console.log(`La Cantidad de Desaprobados fueron
${contadorDesaprobados}`);
console.log(`La Cantidad de Aprobados fueron ${contadorAprobados}`);
console.log(`La Cantidad de Muy Buenos fueron ${contadorMuyBuenos}`);
console.log(`La Cantidad de Excelentes fueron ${contadorExcelentes}`);
```


DIPLOMATURA EN DISEÑO WEB FULL STACK CON JAVASCRIPT

MÓDULO 02 - JAVASCRIPT

Ejercicio propuesto Nro. 08: (estructuras repetitivas) – ciclo for

Realizar un programa en JavaScript que permita ingresar la edad de todos los estudiantes de la diplomatura de la “Diplomatura Universitaria en Desarrollo Web Full Stack con JavaScript” y obtener a partir de esos datos los siguientes datos.

Cantidades:

Cantidad de alumnos mayores a 50 años.

Cantidad de alumnos cuya edad sea menor o igual a 50 años.

Porcentajes:

Porcentaje de alumnos mayores a 50 años sobre el Total.

Porcentaje de alumnos menores o iguales a 50 años sobre el Total

Solución:

```
/*
Ejercicio propuesto Nro. 08: (estructuras repetitivas)
Realizar un programa en JavaScript que permita ingresar la
edad de todos los estudiantes de la diplomatura de la
“Diplomatura Universitaria en Desarrollo Web Full Stack con JavaScript”
y obtener a partir de esos datos los siguientes datos.

Cantidades:
    Cantidad de alumnos mayores a 50 años.
    Cantidad de alumnos cuya edad sea menor o igual a 50 años.
    Cantidad total de alumnos.
Porcentajes:
    Porcentaje de alumnos mayores a 50 años sobre el Total.
    Porcentaje de alumnos menores o iguales a 50 años sobre el Total

Considere como error aquellos que están por debajo de los 18 años
y por arriba de los 100.
*/

{

    /* declaro las variables principales */

    let cantidadErrores = 0; // para cantidad de errores
    let cantidadMayores50 = 0; // para mayores a 50 años
    let cantidadMenores50 = 0; // para menores a 50 años
    let cantidadTotal = 0; // cantidad total
```

DIPLOMATURA EN DISEÑO WEB FULL STACK CON JAVASCRIPT

MÓDULO 02 - JAVASCRIPT

```
for(let i = 1;i <= 5;i = i + 1)
{
    let edadAlumno = Number(prompt(`Por favor Ingrese la Edad del alumno
nro. ${i}`));

    console.log(`Edad leida del alumno ${i} - ${edadAlumno}`);

    if((edadAlumno < 18) || (edadAlumno >= 100))
    {
        console.log("la edad registrada corresponde a un error");
        cantidadErrores = cantidadErrores + 1;
    }
    else
    {
        if((edadAlumno >= 18) && (edadAlumno <= 50))
        {
            console.log("edad registrada entre 18 y 50 años");
            cantidadMenores50 = cantidadMenores50 + 1;
        }
        else
        {
            console.log("edad registrada > 50 y < 100 años");
            cantidadMayores50 = cantidadMayores50 + 1;
        }
    }
}

/* Muestro los Totales */
console.log(`1 - Cantidad de Errores son : ${cantidadErrores}`);
console.log(`2 - Cantidad de Mayores a 50 años ${cantidadMayores50}`);
console.log(`3 - Cantidad de Menores o iguales a 50 años
${cantidadMenores50}`);

/* obtengo la cantidad total */
cantidadTotal = cantidadErrores + cantidadMayores50 + cantidadMenores50;

/* declaro variables y las inicializo, alli
voy a guardar los porcentajes sobre el total */

let porcentajeErrores = 0;
let porcentajeMayores50 = 0;
let porcentajeMenores50 = 0;

/* si hubo al menos 1 error, obtengo el porcentaje */
if(cantidadErrores > 0)
{
```

DIPLOMATURA EN DISEÑO WEB FULL STACK CON JAVASCRIPT

MÓDULO 02 - JAVASCRIPT

```
    porcentajeErrores = (cantidadErrores/cantidadTotal)*100;
  }

  /* si hubo al menos 1 mayor a 50 años obtengo el porcentaje */
  if(cantidadMayores50 > 0)
  {
    porcentajeMayores50 = (cantidadMayores50/cantidadTotal)*100;
  }

  /* si hubo al menos 1 (una persona) entre 18 y 50 años obtengo el
promedio */
  if(cantidadMenores50 > 0)
  {
    porcentajeMenores50 = (cantidadMenores50/cantidadTotal)*100;
  }

  console.log(`El porcentaje de errores es de ${porcentajeErrores} %`);
  console.log(`El porcentaje de mayores a 50 años es de
${porcentajeMayores50} %`);
  console.log(`El porcentaje de menores o iguales a 50 años es de
${porcentajeMenores50} %`);
}
```

DIPLOMATURA EN DISEÑO WEB FULL STACK CON JAVASCRIPT

MÓDULO 02 - JAVASCRIPT

Ejercicio propuesto Nro. 09: (estructuras repetitivas)

Una Frigorífico posee una cinta transportadora y clasificadora de huevos para consumo humano. La cinta no tan solo los transporta sino también los clasifica según su peso. es decir, al final de la cinta existe una balanza electrónica de alta precisión que evalúa su peso y los clasifica.

- a) XL, super grandes: peso \geq 73 gramos.
- b) L, grandes: peso \geq 63 gramos y $<$ 73 gramos.
- c) M, medianos: peso \geq 53 gramos y $<$ 63 gramos.

Determinar lo siguiente:

- 1) Cantidad total de Huevos (Todas las categorías)
- 2) Cantidad total de Huevos XL
- 3) Cantidad total de Huevos L
- 4) Cantidad total de Huevos M
- 5) Cantidad total de Huevos descartados
- 6) Determinar el % de Huevos XL sobre el Total
- 7) Determinar el % de Huevos L sobre el Total
- 8) Determinar el % de Huevos M sobre el Total

Solución:

```
{

  let cantidadXL = 0;
  let cantidadL = 0;
  let cantidadM = 0;
  let cantidadDescarte = 0;
  let cantidadTotal = 0;

  let porcentajeXL = 0;
  let porcentajeL = 0;
  let porcentajeM = 0;
  let porcentajeDescarte = 0;

  for(let i = 1;i <=5;i++)
  {
    /* Dentro del Ciclo debería ir registrando el peso de cada huevo */

    /* declaro e inicializo la variable donde
    leeré el peso de cada huevo */
```

DIPLOMATURA EN DISEÑO WEB FULL STACK CON JAVASCRIPT

MÓDULO 02 - JAVASCRIPT

```
let pesoDeHuevo = 0;

/* leo desde el teclado el Peso de cada
huevo y lo convierto a número */
pesoDeHuevo = Number(prompt(`Ingrese el peso del huevo ${i}`));

/* muestro el peso del huevo registrado*/
console.log(`Huevo Registrado Nro. ${i} con peso ${pesoDeHuevo}`);

/*-----
comenzamos a analizar en que categoría
cae en función de su peso
-----*/

/* Si el peso del huevo es mayor o igual que 73 gramos es XL */
if(pesoDeHuevo >= 73)
{
    //cuento la cantidad de XL en su contador //
    cantidadXL = cantidadXL + 1;
    console.log(`Cantidad de XL := ${cantidadXL}`);
}
else
{
    /* si el peso es mayor o igual a 63 gramos y menor
    o a 73 es L*/
    if((pesoDeHuevo >= 63) && (pesoDeHuevo < 73))
    {
        cantidadL = cantidadL + 1;
        console.log(`cantidad de L := ${cantidadL}`)
    }
    else
    {
        /* Si el peso es mayor a 53 y menor a 63 es M */
        if((pesoDeHuevo >= 53) && (pesoDeHuevo < 63))
        {
            cantidadM = cantidadM + 1;
            console.log(`cantidad de M:= ${cantidadM}`);
        }
        else
        { // los que no ingresan a ninguna categoría son
descartes //
            cantidadDescarte = cantidadDescarte + 1;
            console.log(`cantidad de descartes:
${cantidadDescarte}`);
        }
    }
}
}
```

DIPLOMATURA EN DISEÑO WEB FULL STACK CON JAVASCRIPT

MÓDULO 02 - JAVASCRIPT

```
}

/* obtengo la cantidad total de huevos
sumando las cantidades parciales
de cada categoria */

cantidadTotal = cantidadXL + cantidadL + cantidadM + cantidadDescarte;

/* muestro las cantidades de cada huevo */
console.log(`total de huevos XL: ${cantidadXL}`);
console.log(`total de huevos L:  ${cantidadL}`);
console.log(`total de huevos M:  ${cantidadM}`);
console.log(`cantidad de descartes: ${cantidadDescarte}`);

/* obtengo el porcentaje de huevos XL sobre el total */
if(cantidadXL > 0)
{
    porcentajeXL = (cantidadXL/cantidadTotal)*100;
}

/* obtengo el porcentaje de huevos L sobre el total */
if(cantidadL > 0)
{
    porcentajeL = (cantidadL/cantidadTotal)*100;
}

/* obtengo el porcentaje de huevos M sobre el total */
if(cantidadM > 0)
{
    porcentajeM = (cantidadM/cantidadTotal)*100;
}

/* obtengo el porcentaje de huevos M sobre el total */
if(cantidadDescarte > 0)
{
    porcentajeDescarte = (cantidadDescarte/cantidadTotal)*100;
}

/* visualizo los porcentajes */
console.log(`-----`);
console.log(`el porcentaje de XL ${porcentajeXL} %`);
console.log(`el porcentaje de L  ${porcentajeL} %`);
console.log(`el porcentaje de M  ${porcentajeM} %`);
console.log(`el porcentaje de Descartes ${porcentajeDescarte} %`);
}
```

DIPLOMATURA EN DISEÑO WEB FULL STACK CON JAVASCRIPT

MÓDULO 02 - JAVASCRIPT

LINK PROYECTO:

<https://drive.google.com/file/d/1fXQiMRMpxccf6gYhCrmZnP8YpurixOVD/view?usp=sharing>