

DESARROLLO WEB EN ENTORNO SERVIDOR

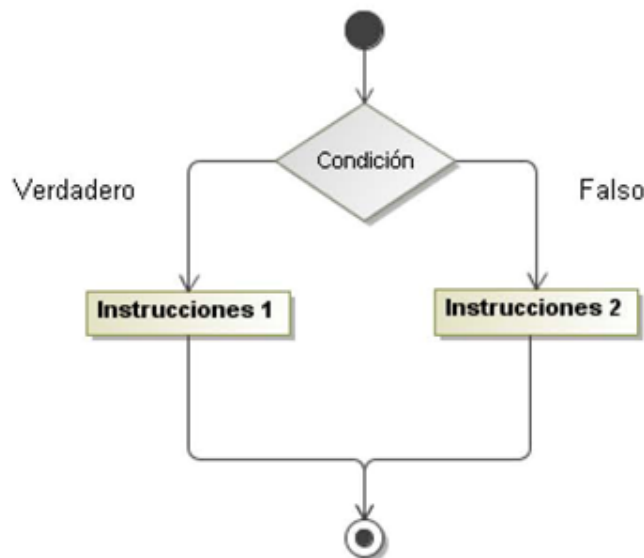
CAPÍTULO 4: **Programación en servidor** **Sentencias de control**

Sentencias condicionales

- Son estructuras de control que permiten decidir el flujo de ejecución de un programa, es decir, el orden en el que las instrucciones de un programa se van ejecutar.
- Tipos:
 - Sentencias if.
 - Sentencias switch.

Sentencias If

- Este tipo de estructuras de control condicionales definen dos flujos de ejecución dependiendo de si se cumple o no la condición establecida por el programador



Sentencias If

- La sintaxis es la siguiente:

- Sentencia If.

- **PHP y JSP:**

```
if (condición){  
    instrucciones;  
}
```

- **ASP:**

```
if (condición) then  
    instrucciones  
end if
```

Sentencias If

- Sentencia If-else.

- **PHP y JSP:**

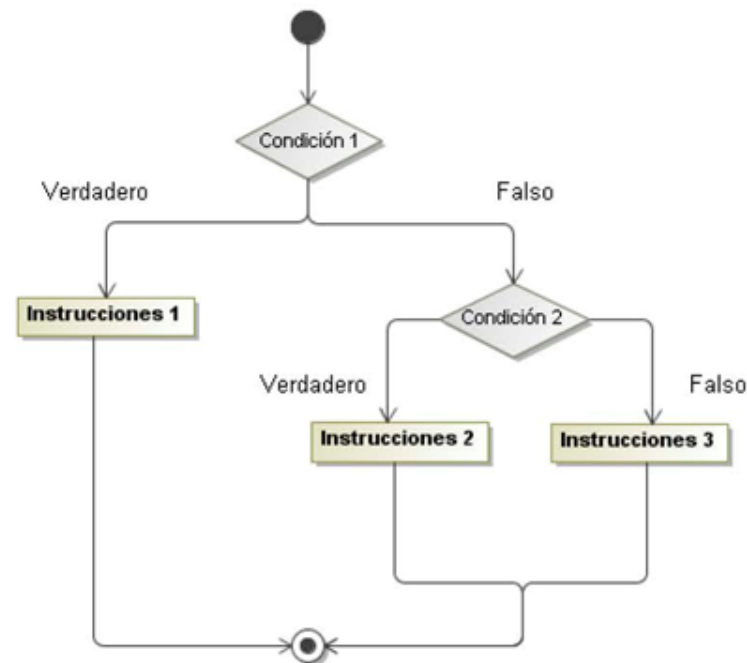
```
if (condición){  
    instrucciones1;  
}else{  
    instrucciones2;  
}
```

- **ASP:**

```
if (condición) then  
    instrucciones1  
else  
    instrucciones2  
end if
```

Sentencias If

- Los if anidados permiten evaluar varias condiciones previas antes de ejecutar las instrucciones correspondientes.



Sentencias If

- La sintaxis es la siguiente:

- **PHP:**

```
if (condición1){  
    instrucciones1;  
}elseif (condición2){  
    instrucciones2;  
}else{  
    instrucciones3;  
}
```

- **PHP y JSP:**

```
if (condición1){  
    instrucciones1;  
}else if (condición2){  
    instrucciones2;  
}else{  
    instrucciones3;  
}
```


Sentencias If

○ ASP:

- ```
if (condición1) then
 instrucciones1
elseif (condición2) then
 instrucciones2
else
 instrucciones3
end if
```
- ```
if (condición1) then
    instrucciones1
else
    if (condición2) then
        instrucciones2
    else
        instrucciones3
    end if
end if
```


Sentencias If - Ejemplo

Reproduce el siguiente archivo
if else.php

```
<?php
    $galletas = true;
    if($galletas == true) {
        echo 'Hay galletas';
    } else {
        echo 'No hay galletas';
    }
?>
```

Reproduce el siguiente archivo:
if else if.php

```
<?php
    $test = 33;
    if ($test > 40) {
        echo "Sí, $test es mayor que 40.";
    } else if ($test > 35) {
        echo "Sí, $test es mayor que 35.";
    } else if ($test > 30) {
        echo "Sí, $test es mayor que 30.";
    } else {
        echo "No, $test es menor que 40, 35 y 30.";
    }
?>
```

Sentencias Switch

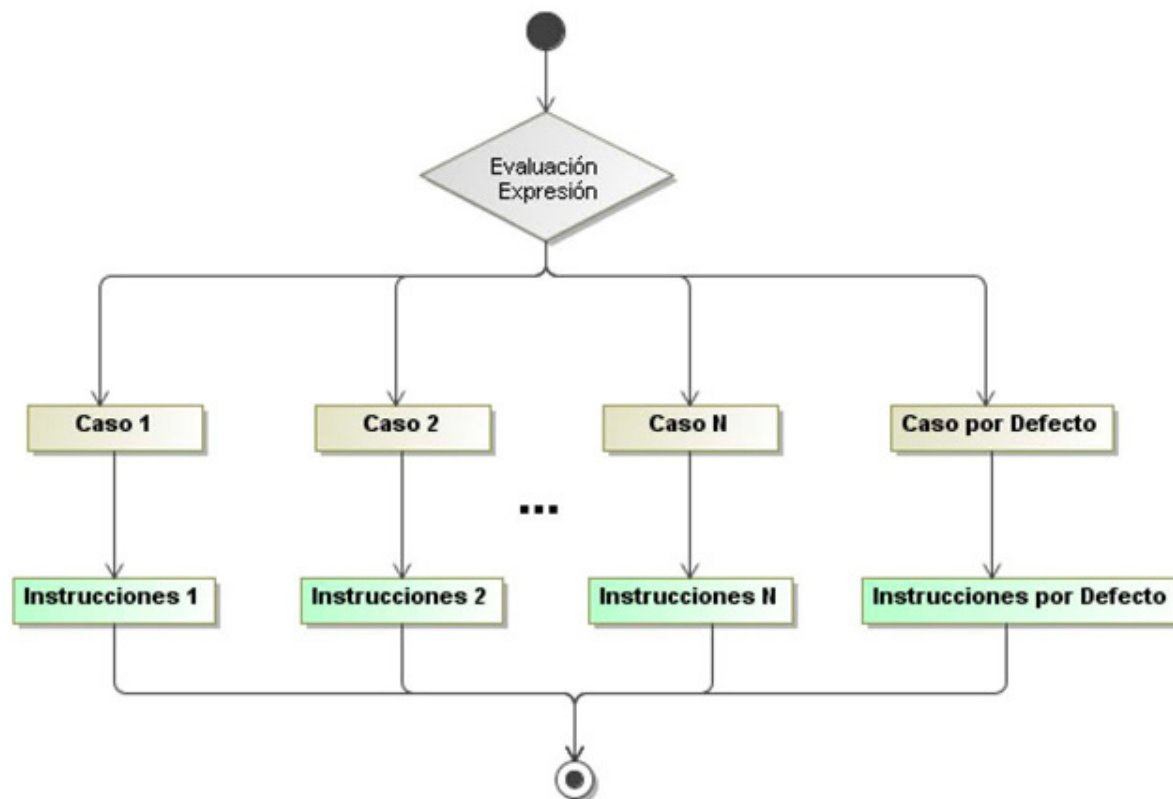
- Estas sentencias se usan cuando dependiendo del valor que toma una variable o expresión, se necesita que se ejecute un conjunto de instrucciones distintas para cada uno de los valores que pueda tomar.
- Suelen emplearse cuando los valores están definidos en un numero concreto de casos.

Sentencias Switch

- Comportamiento:
 1. Se calcula el valor de la expresión.
 2. Se compara dicho valor con cada uno de los casos.
 - 1) Si coincide con un caso se ejecutan las instrucciones contenidas dentro del mismo.
 - 2) Si no coincide con ningún caso se ejecutan las instrucciones definidas en el caso por defecto si es que está definido (el caso por defecto es opcional).

Sentencias Switch

- Comportamiento (continuación):



Sentencias Switch

- Sintaxis:
 - **PHP y JSP:**

```
switch (expresión){  
    case valor1: instrucciones1; break;  
    case valor2: instrucciones2; break;  
    ...  
    case valorN: instruccionesN; break;  
    [default: instruccionesN+1;]  
}
```

Sentencias Switch

- Sintaxis (continuación):

- **ASP:**

```
select case (expresión)
  case valor1
    instrucciones1
  case valor2
    instrucciones2
  ...
  case valorN
    instruccionesN
  [case else
    instruccionesN+1]
end select
```


Sentencias Switch – Ejemplo php

Reproduce el siguiente archivo - [Switch.php](#)

```
<?php
    $mes='febrero'
    switch ($mes) {
    case 'enero':
        echo ("El mes es enero");
    break;
    case 'febrero': echo ("El mes es febrero"); break;
    case 'octubre': echo ("El mes es octubre");
    break;
    default:
        echo ("El mes no es enero, febrero ni octubre");
    break;
    }
?>
```


Bucles

- Este tipo de sentencias se utilizan para ejecutar de forma reiterativa una instrucción o grupo de instrucciones.
- Tipos:
 - While o Do While... Loop.
 - Do -While o Do Loop... While.
 - Do Until... Loop.
 - Do Loop... Until.
 - For o For... Next.
 - Foreach.

While o Do While...Loop

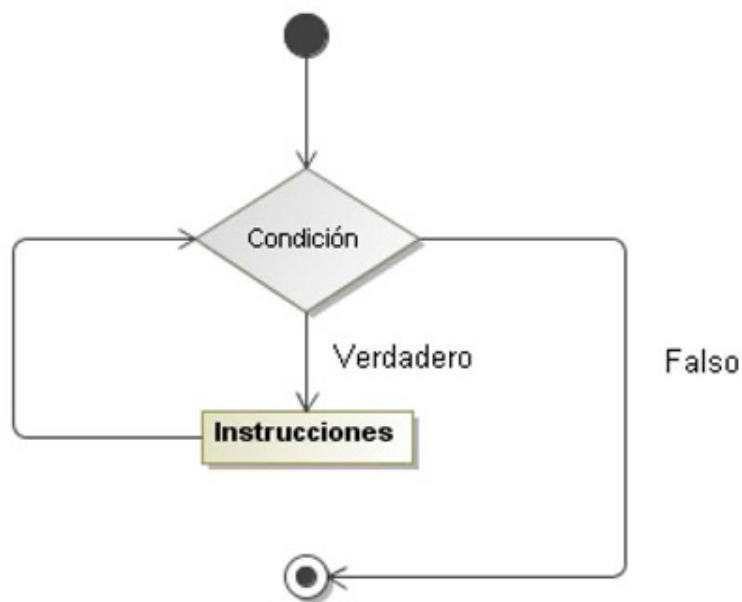
- Este tipo de estructuras permiten ejecutar un número indeterminado de veces una instrucción o grupo de instrucciones, mientras se cumpla la condición.

While o Do While...Loop

- Comportamiento:
 - En cada iteración del bucle se evalúa la condición y si esta es verdadera pasan a ejecutarse las instrucciones contenidas en el cuerpo del bucle. El bucle termina cuando el resultado de evaluar la condición es falso, es decir, cuando la condición ha dejado de cumplirse.

While o Do While...Loop

- Comportamiento (continuación):



While o Do While...Loop

■ Sintaxis:

○ PHP y JSP:

```
while (condición){  
    instrucciones;  
}
```

○ ASP:

- do while (condición)
 instrucciones
loop
- while (condición)
 instrucciones
wend

While o Do While...Loop

- Sintaxis:

- ASP ofrece dos formas distintas de definir este bucle:
 - 1) La primera es la más utilizada y se refiere a la forma nueva de definir este bucle.
 - 2) La segunda es un vestigio de los inicios de Basic. Actualmente los intérpretes soportan también la segunda forma de definir este bucle, para los programadores reticentes a utilizar la nueva forma, pero es posible que en el futuro los intérpretes dejen de contemplarlo.

Sentencias While – Ejemplo php

Reproduce el siguiente archivo - [While.php](#)

<?php

```
$galletas = 0;  
while ($galletas < 10){  
    echo "$galletas, ";  
    $galletas++;  
}
```

?>

Salida: 0,1,2,3,4,5,6,7,8,9

Do-While o Do Loop...While

- Este bucle se ejecuta un número indeterminado de veces hasta que el resultado de evaluar la condición es falsa.
- Características:
 - Siempre se ejecuta al menos una vez se cumpla o no la condición.
 - la evaluación de la condición se realiza al final de cada iteración y no al principio.

Do-While o Do Loop... While

- Comportamiento:



Do-While o Do Loop...While

- Sintaxis:

- **PHP y JSP:**

```
do{  
    instrucciones;  
}while (condición);
```

- **ASP:**

```
do  
    instrucciones  
loop while (condición)
```

Sentencia Do While – Ejemplo php

Reproduce el siguiente archivo – [Do While.php](#)

<?php

```
$galletas = 0;  
do{  
    echo "$galletas, ";  
    $galletas++;  
}  
while ($galletas < 10);
```

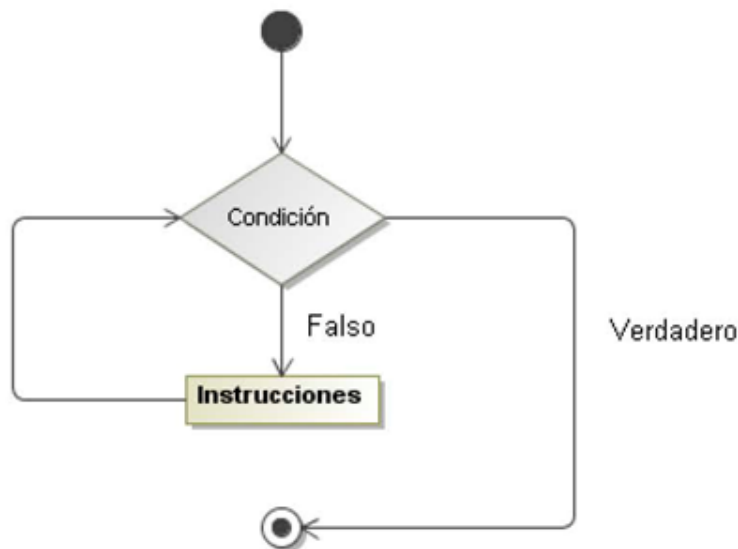
?>

Salida: 0,1,2,3,4,5,6,7,8,9

Bucle Do Until...Loop (ASP)

■ Comportamiento:

- La condición se evalúa al principio de cada iteración.
 - Si la condición no se cumple se ejecuta las instrucciones que contiene.
 - Si la condición se cumple (es verdadera) el bucle finaliza.



Bucle Do Until...Loop

- Sintaxis:

- **ASP:**

```
do until (condición)
    instrucciones
loop
```

Bucle Do Loop...Until

- Comportamiento:
 - Se ejecutan las instrucciones hasta que la condición se cumple.
 - La condición se evalúa al final de cada iteración del bucle, lo que significa que este bucle se ejecuta al menos una vez.



Bucle Do Loop...Until

- Sintaxis:

- **ASP:**

```
do
    instrucciones
loop until (condición)
```

Bucle For o For...Next

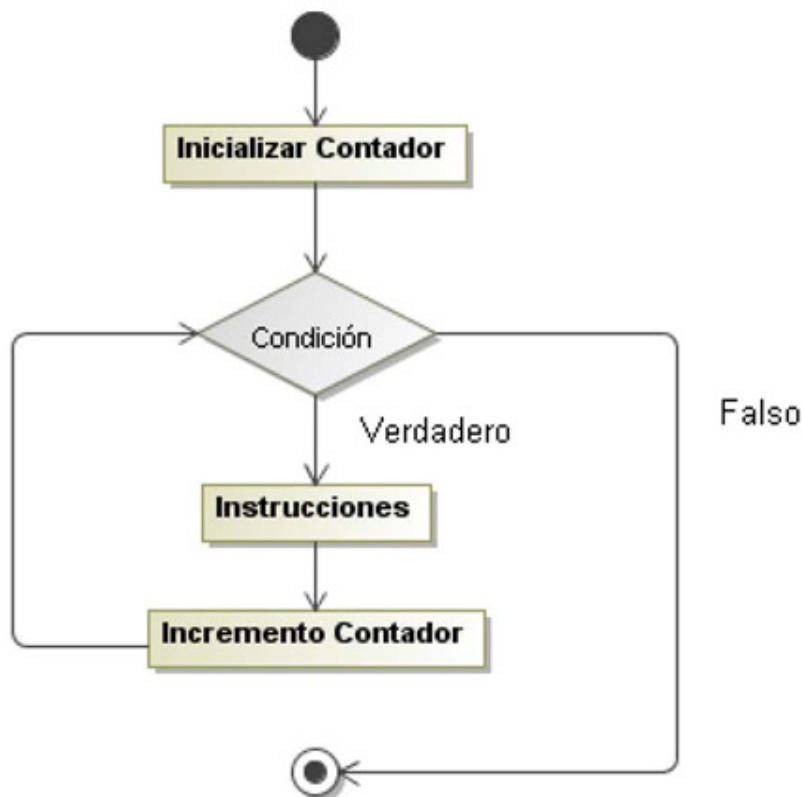
- A diferencia del resto de bucles este tipo de bucles se ejecuta un número determinado de veces. Al igual que el resto de bucles permite ejecutar un conjunto de instrucciones de forma repetitiva.

Bucle For o For...Next

- Comportamiento:
 1. Se inicializa el contador, que controla el número de veces que se ejecuta el bucle.
 2. Se evalúa la condición :
 - Si es verdadera se ejecuta el contenido del bucle y se actualiza el contador.
 - Si es falsa, el bucle finaliza siguiendo con la ejecución de las instrucciones siguientes al bucle.

Bucle For o For...Next

- Comportamiento (continuación):



Bucle For o For...Next

- Sintaxis:

- **PHP y JSP:**

```
for([contador=valorInicial];[condición];[incremento]){  
    instrucciones;  
}
```

- **ASP:**

```
for contador=valorInicial to valorFinal [step incremento]  
    instrucciones  
next
```

Sentencia For – Ejemplo php

Reproduce el siguiente archivo – **for.php**

```
<?php
    for ($galletas = 0; $galletas < 10; $galletas++){
        echo "$galletas, ";
    }
?>
```

Salida: 0,1,2,3,4,5,6,7,8,9

Podríamos variar el incremento (step=paso), cambiando el operador ++ por, ejemplo, +=5

Sentencia Break

- Se utiliza para interrumpir la ejecución de un bucle.
- Sintaxis:
 - **ASP:**
 - For...Next: exit for
 - Do While...Loop, Do Until...Loop, Do...Loop While o Do...Loop Until: exit do
 - While...Wend: exit while

Sentencia Break

- Sintaxis (continuación):

- **PHP:** `break [número];`
- **JSP:** `break [etiqueta];`

- Ejemplo (JSP):

```
etiqueta1:
while(condición1){
    instrucciones;
    if(condición2){
        break etiqueta1;
    }
}
```

Sentencia Continue

- Esta sentencia se utiliza para saltar el resto de la iteración de un bucle y continuar con la evaluación de la condición y la siguiente iteración.
- Sintaxis:
 - **PHP:** `continue [número];`
 - **JSP:** `continue [etiqueta];`

Ejemplo de utilización Break / Continue

```
<?php
$i = 10;
while (--$i)
{
    if ($i == 8)
    {
        continue;
    }
    if ($i == 5)
    {
        break;
    }
    echo $i . "\n" }
?>
```

Salida del Script:

9
7
6

Bucle Foreach

- Proporciona una forma sencilla de iterar sobre los elementos de un array o matrices.
- Sintaxis:
 - **PHP:**
 - ```
foreach(variableArray as variableValor){
 instrucciones;
}
```
    - ```
foreach(variableArray as variableIndice =>  
variableValor){  
    instrucciones;  
}
```

Bucle Foreach

- Sintaxis (continuación):

- **ASP:**

```
for each variableValor in variableArray
    instrucciones
next
```

Sentencia For each– Ejemplo php

Reproduce el siguiente archivo – **foreach.php**

```
<?php
    $estacion[0] = "Primavera";
    $estacion[1] = "Verano";
    $estacion[2] = "Otoño";
    $estacion[3] = "Invierno";
    foreach($estacion as $valor) {
        echo "$valor <br/>";
    }
?>
```


Gracias por tu interés

Fin de la presentación