



Instructor

Diego Fernando Calderón Silva
Correo: dfcalderon@sena.edu.co



Introducción a PHP

¿Qué es PHP?

PHP, es un lenguaje de programación de código abierto que permite el desarrollo de aplicaciones WEB o aplicaciones WEB dinámicas, el cual es apto para incrustar en lenguaje HTML

Tomado de:
<https://www.php.net/manual/es/index.php>



2. Estructuras condicionales



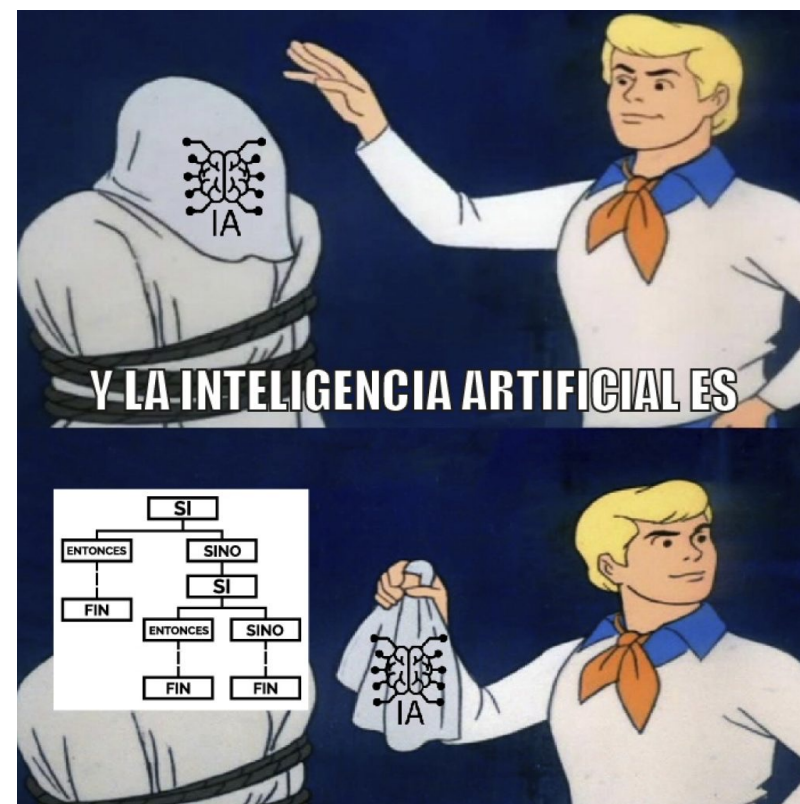
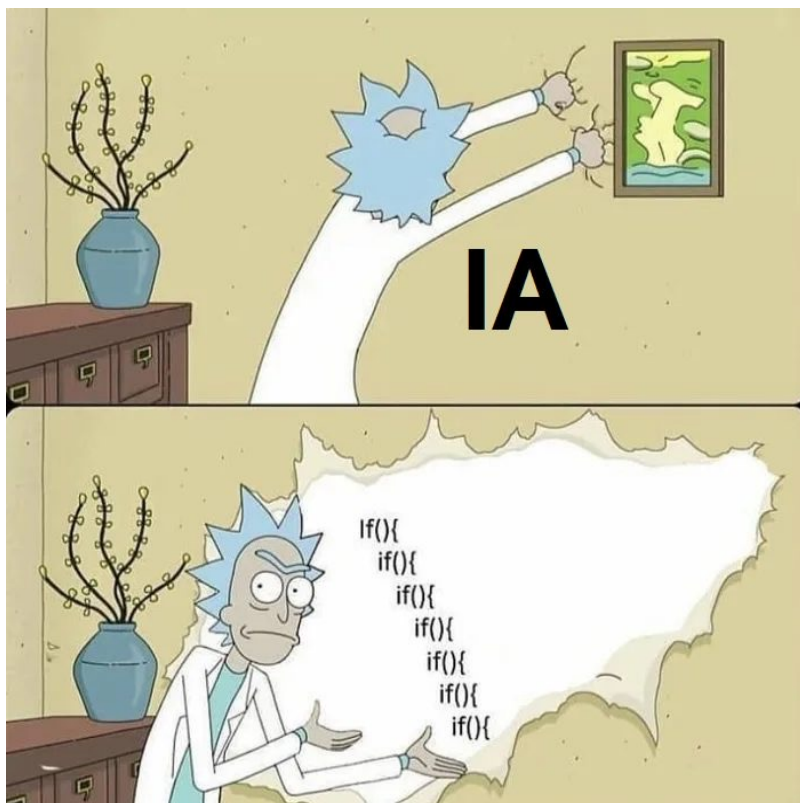
- Operador ternario

Ejercicio:

Hacer un programa que calcule el total a pagar por la compra de camisas. Si se compran 3 camisas o mas se debe aplicar un descuento del 20% sobre el total de la compra y si son menos de 3 camisas un descuento del 10%

2. Estructuras condicionales

- Estructura condicional anidada



2. Estructuras condicionales



- Estructura condicional anidada

```
if (condicion) {  
    if(condicion){  
        Bloque de código a ejecutar si la condición es TRUE  
    }else{  
        Bloque de código a ejecutar si la condición es FALSE  
    }  
} elseif (condicion) {  
    Bloque de código a ejecutar si la condición es TRUE  
} else {  
    Bloque de código a ejecutar si la condición es FALSE  
}
```

2. Estructuras condicionales



- Estructura condicional anidada

- Ejercicio:

Pida a un usuario la edad y genero, para que el sistema le indique si ya esta en la edad para pensionarse. Tenga en cuenta que un Hombre se puede pensionar a los 60años o mas, mientras que las mujeres lo pueden hacer después de los 54 años.

2. Estructuras condicionales



- Estructura condicional SWITCH

- La sentencia switch en PHP es una estructura de control que permite ejecutar diferentes bloques de código dependiendo del valor de una expresión o variable. La sintaxis de la sentencia switch consiste en una expresión que se evalúa y se compara con diferentes casos utilizando la palabra clave “**case**”, seguida de un bloque de código que se ejecutará si el valor de la expresión coincide con el caso. Si ninguno de los casos coincide con el valor de la expresión, se puede incluir un bloque de código “**default**” que se ejecutará en su lugar. La sentencia switch es útil cuando se tienen múltiples opciones de ejecución basadas en el valor de una variable, lo que simplifica el código en lugar de tener una serie de sentencias if encadenadas.

2. Estructuras condicionales



- Estructura condicional SWITCH

```
<?php
$opcion = 2;

switch ($opcion) {
    case 1:
        echo "Has seleccionado la opción 1";
        break;
    case 2:
        echo "Has seleccionado la opción 2";
        break;
    case 3:
        echo "Has seleccionado la opción 3";
        break;
    default:
        echo "Opción inválida";
        break;
}
?>
```

2. Estructuras condicionales



- Estructura condicional SWITCH

- Ejercicio:

Diseñe un software que escriba los nombre del día de la semana en función de una variable llamada “día”.

Los días de la semana son 7, por lo que, el rango de los valores de “dia” debe ser de 1 a 7 y dado el caso que “dia” tome un valor fuera de rango se deberá producir un mensaje de alerta!!!

2. Estructuras condicionales



- Estructura de selección múltiple
MATCH

- Esta estructura se introdujo en PHP 8, es similar a la estructura SWITCH pero acá las operaciones tienen en cuenta el tipo de dato a iterar.
- Permite doble comparación y distingue entre tipos de datos
- Permite hacer condicionales.

2. Estructuras condicionales



- Estructura de selección múltiple
MATCH

```
$opcion = 2;

switch ($opcion) {
    case 1:
        echo "Seleccionaste la opción 1";
        break;
    case 2:
        echo "Seleccionaste la opción 2";
        break;
    case 3:
        echo "Seleccionaste la opción 3";
        break;
    default:
        echo "Opción inválida";
        break;
}
```

```
$color = 'rojo';

$action = match ($color) {
    'rojo' => 'Detenerse',
    'amarillo' => 'Precaución',
    'verde' => 'Avanzar',
    default => 'Color desconocido'
};

echo $action;
```

2. Estructuras condicionales



- Estructura de selección múltiple
MATCH

```
<?php

$a="7";

$x=10;
$y=9;
$z=7;

$resultado = match($a){
    $x => "Valor igual a X",
    $y => "Valor igual a Y",
    $z => "Valor igual a Z",
    default => "No coincide con ninguna variable"
};

echo $resultado;
```


2. Estructuras condicionales



- Estructura de selección múltiple
MATCH

```
<?php
$edad = 18;

$resultado = match(true){
    $edad >= 60 => "Eres de la tercer edad",
    $edad >= 30 => "Eres adulto",
    $edad >= 10 => "Eres un niño"
};

echo $resultado;
```

3. Ciclos y bucles



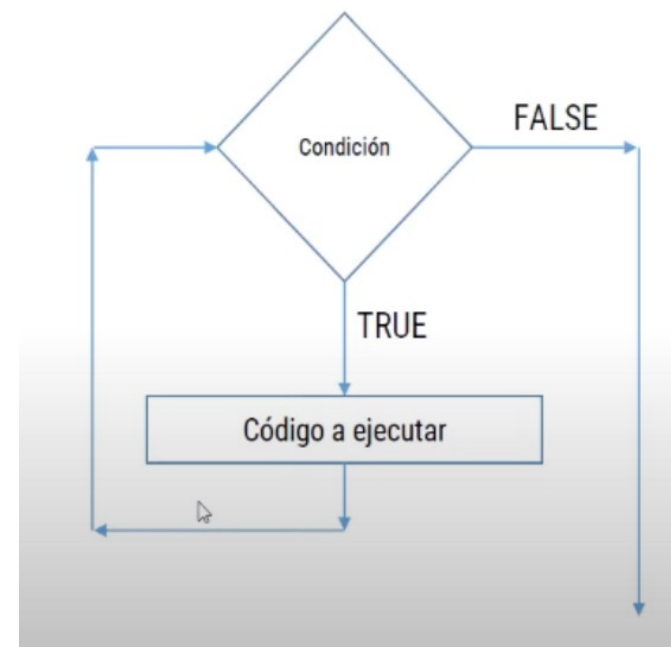
- Estructura repetitivas

- **WHILE**

While (mientras), ejecuta una porción de código designado mientras la condición que se le indica en su constructor sea **VERDADERA**. Una vez se sale del ciclo, es por que la condición deja de cumplirse.

```
While(Condicion){  
Código a ejecutar;  
}
```

```
$num = 1;  
  
while ($num <= 10) {  
    echo $num . "<br>";  
    $num++;  
}
```



3. Ciclos y bucles



- Estructura repetitivas

- **WHILE**

Ejercicio:

1. Diseñe un código que imprima los números de -10 a 20 usando incrementos y decrementos.
2. Diseñe un código que imprima la tabla de multiplicar de un numero entre el 1 y 12.
3. Diseñe un código que muestre los números pares entre 0 y 100.

3. Ciclos y bucles



- Estructura repetitivas

- **DO - WHILE**

Do - while (hacer mientras mientras), es similar a la estructura while, excepto que la expresión verdadera es verificada al final de cada iteración en lugar de al principio.

```
do {  
Código a ejecutar;  
}while(Condicion)
```

```
$num = 1;  
  
do {  
    echo $num . "<br>";  
    $num++;  
} while ($num <= 10);
```

3. Ciclos y bucles



- Estructura repetitivas

- **DO-WHILE**

Ejercicio:

1. Diseñe un código que imprima los números de -10 a 20 usando incrementos y decrementos.
2. Diseñe un código que imprima la tabla de multiplicar de un numero entre el 1 y 12.
3. Diseñe un código que muestre los números impares entre 0 y 100.

3. Ciclos y bucles



- Estructura repetitivas

- **FOR(para)**

La estructura repetitiva “for” se utiliza cuando se tienen definidas la cantidad de iteraciones a realizar

```
for($var; Condicion; (incre | decre) ){
```

```
    Código a ejecutar;
```

```
}
```

3. Ciclos y bucles



- Estructura repetitivas

- **FOR**

Ejercicios

1. Diseñe un código que muestre los numero del -10 al 10.
2. Escriba un código que muestre la suma de los primeros 10 números enteros.
3. Diseñe un código que genere un ciclo for anidado para imprimir las tablas de multiplicar del 1 al 10.

```
1 2 3 4 5 6 7 8 9 10
2 4 6 8 10 12 14 16 18 20
3 6 9 12 15 18 21 24 27 30
4 8 12 16 20 24 28 32 36 40
5 10 15 20 25 30 35 40 45 50
6 12 18 24 30 36 42 48 54 60
7 14 21 28 35 42 49 56 63 70
8 16 24 32 40 48 56 64 72 80
9 18 27 36 45 54 63 72 81 90
10 20 30 40 50 60 70 80 90 100
```

3. Ciclos y bucles



- Estructura repetitivas

- FOR

```
echo "<table>";

for ($i = 1; $i <= 10; $i++) {
    echo "<tr>";

    for ($j = 1; $j <= 10; $j++) {
        $mult = $i * $j;
        echo "<td>" . $mult . "</td>";
    }

    echo "</tr>";
}

echo "</table>";
```

3. Ciclos y bucles



- Estructura repetitivas

- **FOREACH**

Este ciclo proporciona un modo bastante sencillo de iterar sobre arrays. Se debe tener en cuenta que **foreach solo funciona sobre arrays** y de intentar usarlo sobre otro tipo de variable emitirá error. Para esta estructura existen dos sintaxis.

```
foreach($array as $valor){  
    $valor tendrá en cada iteración  
    un valor del array  
}
```

```
foreach($array as $clave => $valor){  
    $clave tendrá en cada iteración  
    una clave del array  
  
    $valor tendrá en cada iteración  
    un valor del array  
}
```

3. Ciclos y bucles



- Estructura repetitivas

- **FOREACH**

```
$array = array('uno', 'dos', 'tres');  
  
foreach ($array as $valor) {  
    echo $valor;  
}
```

```
<?php  
  
$frutas = ["Mora", "Naranja", "Fresa", "Maracuya", "Mandarina"];  
  
$computadores = [  
    "marca" => "Acer",  
    "Modelo" => "a2338",  
    "Color" => "Rojo",  
    "Ram" => "256GB"  
];  
  
foreach($computadores as $clave => $com){  
    echo "clave=".$clave." valor=".$com."<br>";  
}
```


3. Ciclos y bucles



- ## Desarrolle

- **1. Ejercicio de suma y promedio:** Escribe un programa que solicite al usuario ingresar una lista de números y luego calcule la suma y el promedio de esos números.
- **2. Ejercicio de números pares:** Escribe un programa que muestre todos los números pares del 1 al 100.
- **3. Ejercicio de factorial:** Escribe un programa que calcule el factorial de un número ingresado por el usuario.

3. Ciclos y bucles



- ## Desarrolle

- **4. Ejercicio de palíndromos:** Escribe un programa que verifique si una palabra ingresada por el usuario es un palíndromo (se lee igual de izquierda a derecha y de derecha a izquierda).
- **5. Ejercicio de primos:** Escribe un programa que determine si un número ingresado por el usuario es primo o no.
- **6. Ejercicio de reverso:** Escribe un programa que invierta un número ingresado por el usuario. Por ejemplo, si el usuario ingresa "12345", el programa debe mostrar "54321".
- **7. Ejercicio de búsqueda:** Escribe un programa que busque un elemento específico en una lista ingresada por el usuario y muestre su posición.

3. Ciclos y bucles



- ## Desarrolle

- **8. Ejercicio de números perfectos:** Escribe un programa que encuentre todos los números perfectos en un rango específico. Un número perfecto es aquel cuya suma de sus divisores propios (excluyendo al número mismo) es igual al número.
- **9. Ejercicio de Fibonacci:** Escribe un programa que genere la secuencia de Fibonacci hasta un número ingresado por el usuario.
- **10. Ejercicio de ordenamiento:** Escribe un programa que ordene una lista de números ingresada por el usuario de menor a mayor utilizando el algoritmo de ordenamiento de burbuja.

3. Ciclos y bucles



- ¿Cómo salir o detener un ciclo?

¿?



G R A C I A S

Línea de atención al ciudadano: 01 8000 910270
Línea de atención al empresario: 01 8000 910682



www.sena.edu.co