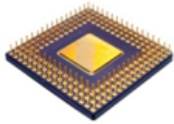


Estructuras

Una estructura establecer una relación y orden entre distintas partes.

Estructuras control

Es la que establece el **orden de ejecución** de las instrucciones:



secuencial
alternativa
repetitiva

Estructuras de datos



son formas de organizar datos en una computadora para que información pueda ser utilizada de forma eficiente:

Arreglos --->vemos en IP

Arboles
Diccionarios
Pilas
Colas
....

VARIABLE

Variable de tipo de dato simple: Es un contenedor que almacena un dato (string, float, int, boolean). Se le define un nombre, etiqueta o identificador.

Variable de tipo de dato Arreglo (array) o arreglo: es una colección de **datos del mismo tipo**. Se le define un nombre, etiqueta o identificador.

Tipos de arreglos que vamos a estudiar:

- a) **arreglos indexados**
- b) **arreglos asociativos**
- c) arreglos multidimensionales

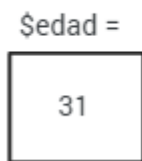
¿Que vamos a estudiar de cada tipo de arreglo?

- i) cómo se representa en “papel”.
- ii) qué caracteriza a cada tipo de arreglo? → TIPO DE DATO DEL INDICE!!!!!!! (depende si el índice es de tipo de dato entero o string).
- iii) función predefinida COUNT!
- iv) Cómo se inicializa con el arreglo vacío (arreglo sin elementos)
cómo se inicializa el arreglo con datos (arreglo con elementos)
- v) cómo se agregar valores al arreglo
- vi) cómo acceder a un elemento del arreglo
- vii) cómo recorrer los elementos del arreglo.

a) ARREGLOS INDEXADOS

Es un arreglo que utiliza índice numéricos (int) para acceder a los valores almacenados en el arreglo.

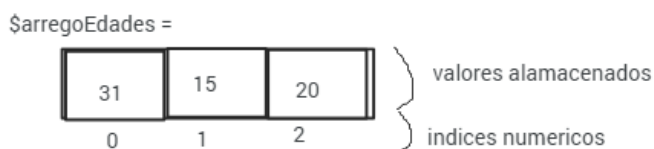
```
//int $edad  
$edad = 31;
```



```
//array $arregloEdades
```

```
$arregloEdades = array( 31, 15, 20 ) ;
```

```
$arregloEdades = [ 31, 15, 20 ] ; //a partir versión php 5.4 //arreglo indexado de tipo de dato entero.
```



```
$arreglosProfesores = [“Majo”, “David”, “Kari”, “Agustin” ]; //arreglo indexado de tipo de dato string.
```



```
$arregloEdades = [ ] ; //crear un arreglo sin elementos, vacio
$arregloEdades = array( ); //crear un arreglo sin elementos, vacio
```

```
$arregloEdades[0] = 31;
$arregloEdades[1] = 20;
$arregloEdades[2] = 4;
$arregloEdades[3] = 15;
//int $j
$j = 4;
$arregloEdades[ $j ] = 35;
```

FUNCION PREDEFINIDA count!

```
/**
 * obtiene la cantidad de elementos de un arreglo
 * @param array $arreglo
 * @return int
 */
function count( $arreglo )
```

```
echo "la cantidad de profesores es ". count($arregloProfesores); //la cantidad de
profesores es 4;
```

```
$arregloTemperaturas = [ ] ;
echo "La cantid de temperaturas es:". count($arregloTemperaturas); // 0
```

```
$arregloProfesores = [ "Majo", "David", "Kari", "Agustin" ]; //arreglo indexado de tipo de dato
string.
```

\$arregloProfesores =

majo	David	Kari	Agustin
0	1	2	3

```
echo "El profesor de la posicion 1 es: ". $arregloProfesores[1];
for( $i = 0; $i < count($arregloProfesores); $i++){
    echo $arregloProfesores[ $i ]. "\n";
}
```

```
<?php
//array $arregloProfesores //arreglo indexado de string.
```

```

$arregloProfesores = ["Majo", "David", "Kari", "Agustin"];

echo "El profesor de la posicion 1 es: " . $arregloProfesores[1] .
"\n\n";

for ($i = 0; $i < count($arregloProfesores); $i++) {
    echo $arregloProfesores[$i] . "\n";
}

```

```

echo "Cuantos profes hay?";
$cantProfes = trim(fgets(STDIN));

for ($i = 0; $i < $cantProfes; $i++) {
    echo "ingresa el nombre:";
    $arregloProfesores[$i] = trim(fgets(STDIN));
}

echo "IMPRIMO LOS PROFES: ";
$acumuladorString = "";
for ($i = 0; $i < count($arregloProfesores); $i++) {
    $acumuladorString = $acumuladorString . "-" .
$arregloProfesores[$i];
}

echo $acumuladorString;

```

```

<?php
/* Ciclo interactivo para ingresar nombres y edades de profes
en arreglos indexados paralelos.*/
//array $arregloProfesores //arreglo indexado de string.
//string $seguir, int $i

$arregloProfesores = [];
$arregloEdades = [];
$i = 0;

echo "Desea agregar un profe? s/n :";
$seguir = trim(fgets(STDIN));

while ($seguir == "s") {
    echo "Ingrese nombre: ";
    $arregloProfesores[$i] = trim(fgets(STDIN));
}

```

```

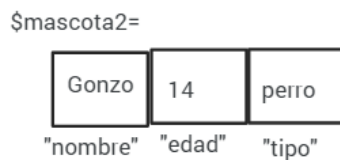
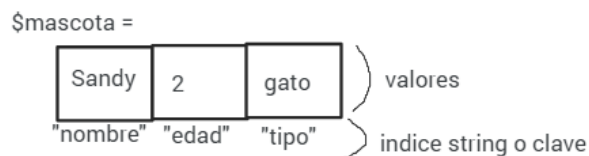
    echo "Ingrese edad:";
    $arregloEdades[$i] = trim(fgets(STDIN));
    $i++;
    echo "Desea agregar otro profe? s/n :";
    $seguir = trim(fgets(STDIN));
}

echo "QUE PROFES INGRESE?\n";
for ($i = 0; $i < count($arregloProfesores); $i++) {
    echo "nombre: " . $arregloProfesores[$i] . " edad: " .
$arregloEdades[$i] . "\n";
}

```

b) ARREGLOS ASOCIATIVOS

Es un arreglo que utiliza índice STRING (o clave) para acceder a los valores almacenados en el arreglo.



```

$mascota = array (
    "nombre" => "Sandy",
    "edad"   => 2 ,
    "tipo"   => "gato"
);

```

```

$mascota2 = array (
    "nombre" => "Gonzo",
    "edad"   => 14 ,
    "tipo"   => "perro"
);

```

//A PARTIR DE PHP 5.4

```

$mascota = [

```

```
        "nombre" => "Sandy",
        "edad"    => 2 ,
        "tipo"    => "gato"
    ];
```

```
$mascota2 = [
    "nombre" => "Gonzo",
    "edad"    => 14 ,
    "tipo"    => "perro"
];
```

```
<?php
//creando arreglos asociativos
//array $mascota, $mascota2 //arreglos asociativos
$mascota = [
    "nombre" => "Sandy",
    "edad"    => 2,
    "tipo"    => "gato"
];

$mascota2 = [
    "nombre" => "Gonzo",
    "edad"    => 14,
    "tipo"    => "perro"
];

print_r($mascota);
print_r($mascota2);

echo "La cantidad de elementos de mascota es: ", count($mascota)."\n";
echo "La cantidad de elementos de mascota2 es: ",
count($mascota2)."\n";
```

//OTRA FORMA DE CREAR EL ARREGLO

```
$mascota = [ ] ; //arreglo vacio
```

```
//string $valor
$valor = "tipo";
```

```
$mascota["nombre"] = "Cuca";
$mascota["edad"] = 34;
$mascota[$valor] = "cucaracha";
```

```

<?php
//creando arreglos asociativos
//array $mascota //arreglos asociativos
$mascota = [
    "nombre" => "Sandy",
    "edad"    => 2,
    "tipo"    => "gato"
];

foreach ($mascota as $key => $value) {
    echo $key . " es " . $value . "\n";
}

```

```

<?php
//creando arreglos asociativos
//array $mascota //arreglos asociativos
$mascota = [
    "nombre" => "Sandy",
    "edad"    => 2,
    "tipo"    => "gato"
];

echo "la edad es: " . $mascota["edad"];

$mascota["edad"] = 5; //reemplazo el valor de la edad.

if ($mascota["edad"] % 2 == 0) {
    echo "\n La edad es par!" . $mascota["edad"];
} else {
    echo "\n La edad es impar!" . $mascota["edad"];
}

```

c) ARREGLOS MULTIDIMENSIONALES (Bidimensionales)

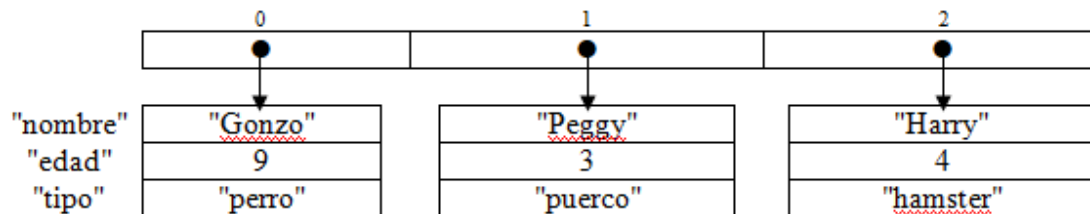
Son arreglos (indexados o asociativos) que almacenan otras estructuras de datos:

- . **Arreglo indexado que en cada posición almacena un arreglo asociativo** (registro de una base de datos).

- . Arreglo indexado que en cada posición almacena un arreglo indexado (matrices)

- . Arreglo asociativo que almacena indexados.
- . Arreglo asociativo que almacena asociativos.

`$misMascotas =`



¿Cómo accedo al nombre de la primer mascota?
`echo $misMascotas[0]["nombres"];`

`$misMacotas = [];`

`$misMascotas[0] = [`
 `"nombre" => "Gonzo",`
 `"edad" => 9,`
 `"tipo" => "perro",`
`];`

`$misMascotas[1] = [`
 `"nombre" => "Peggy",`
 `"edad" => 3,`
 `"tipo" => "puerco",`
`];`

`$misMascotas[2] = [`
 `"nombre" => "HARry",`
 `"edad" => 4,`
 `"tipo" => "Hamster",`
`];`

`$misMascotas[3] = [`
 `"nombre" => "Cuca",`
 `"edad" => 30,`
 `"tipo" => "Cucaracha",`
`];`

```
<?php
$misMascotas = [ ];

$misMascotas[0] = [
    "nombre" => "Gonzo",
```



```
        "edad" => 9,
        "tipo" => "perro",
    ];
    $misMascotas[1] = [
        "nombre" => "Peggy",
        "edad" => 3,
        "tipo" => "puerco",
    ];

    $misMascotas[2] = [
        "nombre" => "HArry",
        "edad" => 4,
        "tipo" => "Hamster",
    ];

    $misMascotas[3] = [
        "nombre" => "Cuca",
        "edad" => 30,
        "tipo" => "Cucaracha",
    ];

    //print_r($misMascotas);
    //
    //¿Cómo accedo al nombre de la primer mascota?
    echo $misMascotas[0]["nombre"] . "\n";

    foreach ($misMascotas as $key => $value) {
        echo "key: " . $key . "\n";
        echo "value: " . $value["nombre"] . "\n";
        //print_r($value);
    }
```