



# Estructuras de Datos

Introducción a la Programación  
Facultad de Informática  
Univ.Nac. del Comahue

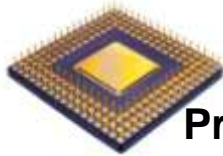


# Temario

- Arreglos
  - Arreglos unidimensionales o indexados
  - Arreglos Asociativos
  - Arreglo Multidimensionales
- Recorridos de arreglos
  - Recorrido Exhaustivo
  - Recorrido Parcial
- Métodos de ordenamiento PHP
- String



# Estructuras de control



Procesador

**Controlan la secuencia o flujo de ejecución de las instrucciones de un programa**

Secuencial  
Alternativa  
Repetitiva

≠

# Estructuras de datos



Memoria

**son formas de organizar datos en una computadora para que la información pueda ser utilizada de manera eficiente.**

Arreglos ✖  
Arboles  
Diccionarios  
Pilas  
Colas...

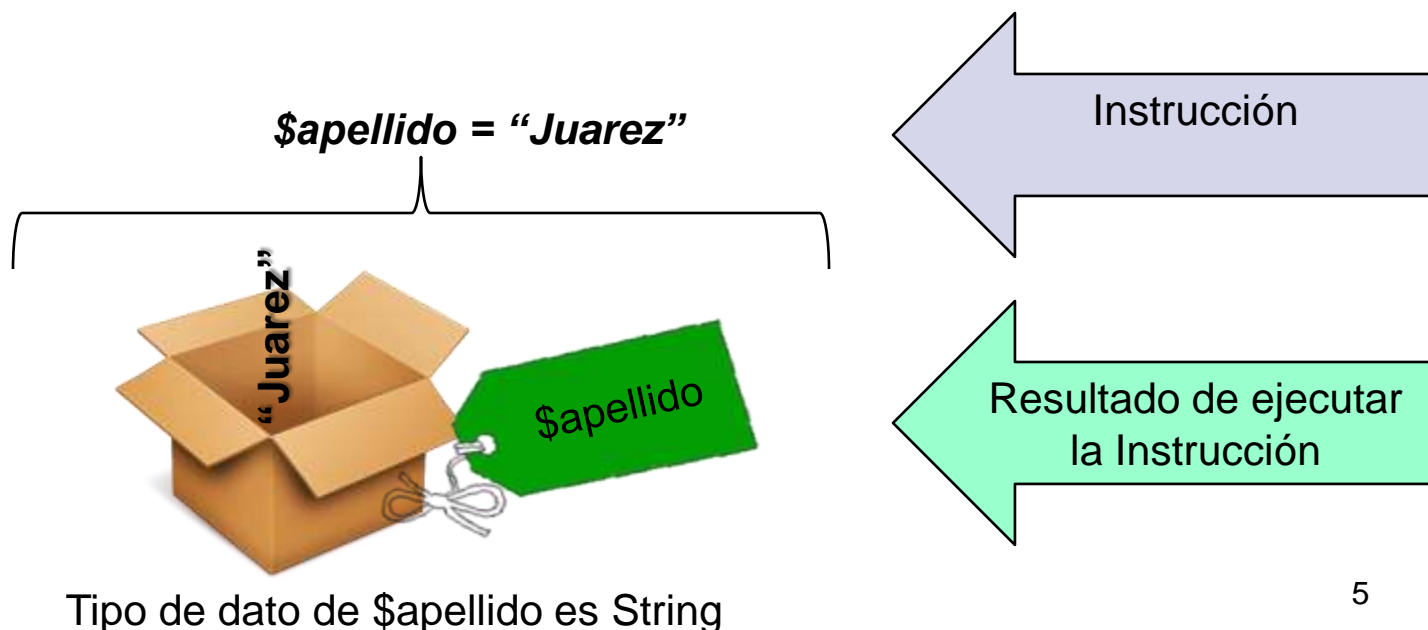


# ¿Qué es una variable?



Es un **contenedor en memoria** cuyo valor puede **variar** durante la ejecución de un programa

Para **acceder** a una variable utilizamos un **NOMBRE o ETIQUETA**  
Contendrá valores de un determinado **TIPO DE DATO**.  
Almacena, guarda o contiene un **VALOR**





## Variable de tipo simple



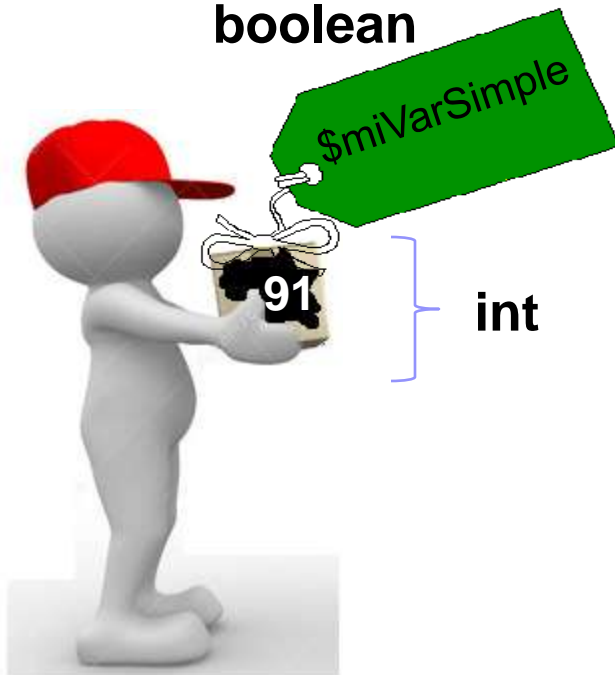
## Variable de tipo Arreglo





# Variable de tipo simple

int  
string  
float  
boolean



Nombre o etiqueta usado para identificar una ubicación de memoria en donde puede almacenarse **un dato**

# Variable de tipo Arreglo

(o directamente Arreglo)

\$miVarArreglo



Los arreglos son estructuras de datos que nos permiten manipular una **colección de datos**

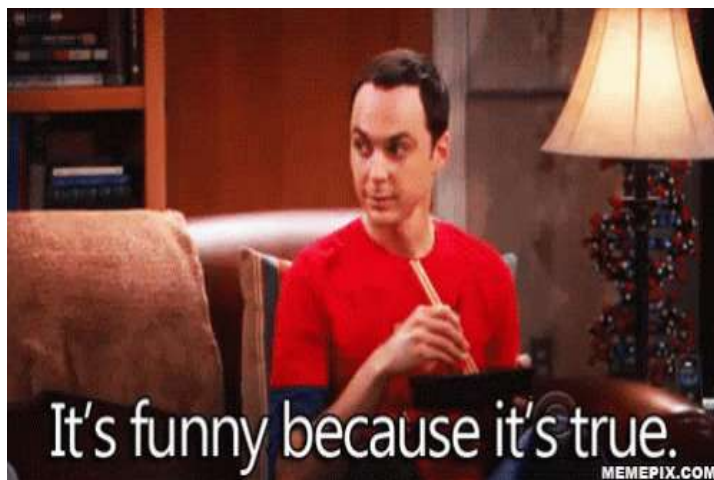


# Arreglos

- Los arreglos son estructuras de datos que nos permiten manipular una **colección** de datos del mismo tipo.



La realidad es que **depende** cómo lo implemente el lenguaje de programación



En PHP un arreglo puede almacenar datos de distinto tipo. Los datos que guardemos en un arreglo, dependerán del problema que debamos resolver.

Me la paso resolviendo problemas





# Arreglos

- Los arreglos son estructuras de datos que nos permiten manipular una **colección** de datos ~~del mismo tipo.~~  
(Podemos escuchar hablar de Colecciones en lugar de Arreglos)



## Representar un arreglo para:

- visualizar fácilmente la estructura y los datos que almacena
- poder compartirla con otros compañeros,
- poder ejecutar trazas: analizar cómo van cambiando los datos del arreglos a medida que ejecutamos un programa

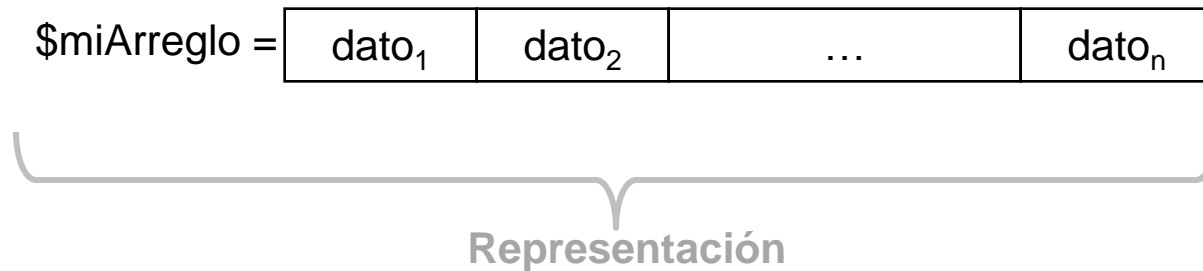






# Arreglos

Arreglo de N elementos:



El tipo de dato de ***dato<sub>1</sub>***, ***dato<sub>2</sub>***, ..., ***dato<sub>n</sub>*** puede ser:  
enteros, float, booleanos, string,  
otras estructuras de datos



# Arreglos

Ejemplos de **representaciones** de arreglos de datos:

Arreglo de 3 elementos de tipo entero:

\$misNumeros = 

45	-5	187
----	----	-----

Arreglo de 5 elementos de tipo String:

\$misNombres = 

"lucia"	"carlos"	"euge"	"mario"	"marta"
---------	----------	--------	---------	---------

Arreglo de 6 elementos de tipo boolean:

\$mayoresDeEdad = 

false	true	true	false	true	true
-------	------	------	-------	------	------

Arreglo de 3 elementos de tipo Vehículos:

\$misVehiculos = 

		
---	---	--

Arreglo de 4 elementos de tipo Clientes:

\$misClientes = 

							
---	--	---	--	--	--	---	--

Programación  
Orientada a Objetos  
(materia de la  
Tec.Un.WEB)



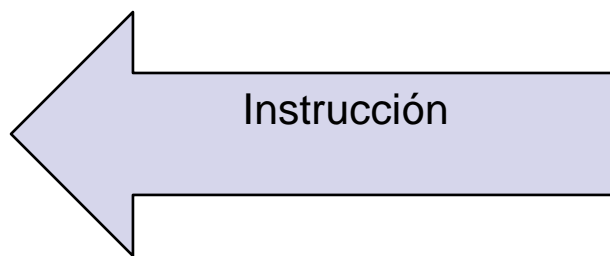
# Arreglos: count

- PHP Implementa la función **Count**, que recibe por parámetro de entrada el nombre de la variable arreglo, y retorna la cantidad de elementos almacenados en el arreglo.

\$misNumeros = 

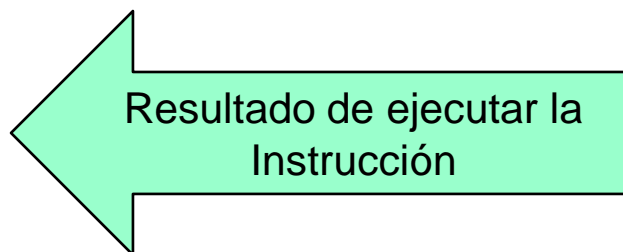
45	-5	187
----	----	-----

**\$n = count (\$misNumeros);**



\$n=

3
---





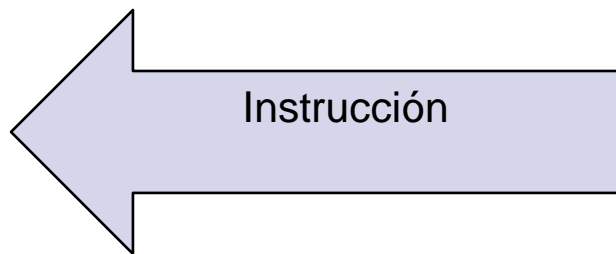
# Arreglos: count

- PHP Implementa la función **Count**, que recibe por parámetro de entrada el nombre de la variable arreglo, y retorna la cantidad de elementos almacenados en el arreglo.

\$misNombres = 

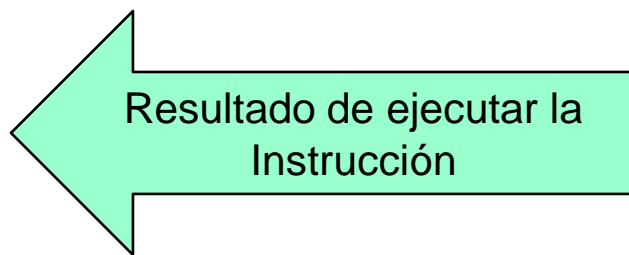
"lucia"	"carlos"	"euge"	"mario"	"marta"
---------	----------	--------	---------	---------

**\$n = count (\$misNombres);**



\$n= 

5
---



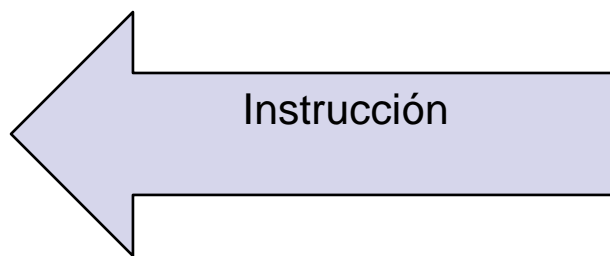


# Arreglos: count

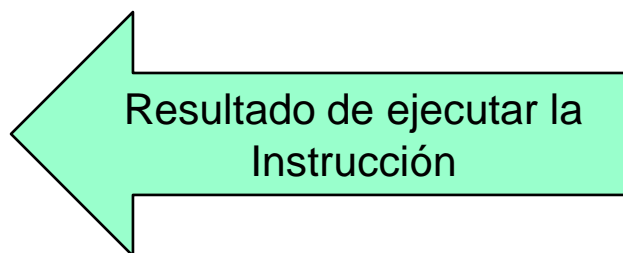
- PHP Implementa la función **Count**, que recibe por parámetro de entrada el nombre de la variable arreglo, y retorna la cantidad de elementos almacenados en el arreglo.

```
$misNumeros = [ ]
```

```
$n = count ($misNumeros);
```



```
$n= 0
```





# Arreglos: count

- PHP Implementa la función **Count**, que recibe por parámetro de entrada el nombre de la variable arreglo, y retorna la cantidad de elementos almacenados en el arreglo.

```
/**
```

```
 * obtiene la cantidad de elementos de un arreglo
```

```
 * @param array $arreglo
```

```
 * @return int
```

```
*/
```

```
function count( $arreglo )
```

Descripción

Parámetro de entrada

Tipo de retorno

Documentación  
De la función  
COUNT



¿Cómo crear/inicializar  
una variable de tipo arreglo?

¿Cómo acceder a los datos que  
contiene?



\$misNumeros =

45	-5	187
----	----	-----

**ARREGLOS:**

- 1. INDEXADOS o UNIDIMENSIONALES**
- 2. ASOCIATIVOS**
- 3. MULTIDIMENSIONALES**



# 1) ARREGLO

## Indexado o Unidimensional







# 1) arreglo indexado o unidimensional



Un arreglo indexado es un arreglo que utiliza **índices numéricos** para acceder a los valores almacenados en el arreglo.



Acordamos que los arreglos indexados con los que trabajaremos almacenan el 1er elemento en el índice 0 (cero)

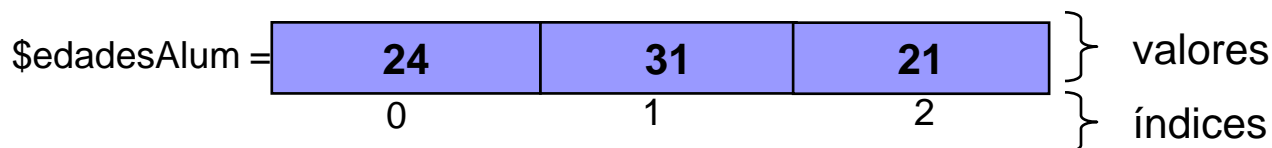


# CREAR un Arreglo Indexado

¿Cuáles son las instruc. para crear un arreglo indexado?

Ejemplo 1: un arreglo con 3 edades de alumnos.

Representación:



Implementación:

Opción 1 para inicializar:

```
$edadesAlum = array( 24, 31, 21 );
```

Opción 2 para inicializar:

```
$edadesAlum = [24, 31, 21 ];
```

} A Partir de la versión  
PHP 5.4

Opción 3 para inicializar)

```
$edadesAlum = array( ); o bien $edadesAlum = [ ];
```

← Inicializo el arreglo en vacio

```
$edadesAlum[0] = 24;
```

```
$edadesAlum[1] = 31;
```

```
$edadesAlum[2] = trim(fgets(STDIN));
```

← El primer elemento se  
almacena la posición o  
índice 0 (cero)

```
$cantEdades = count($edadesAlum);
```

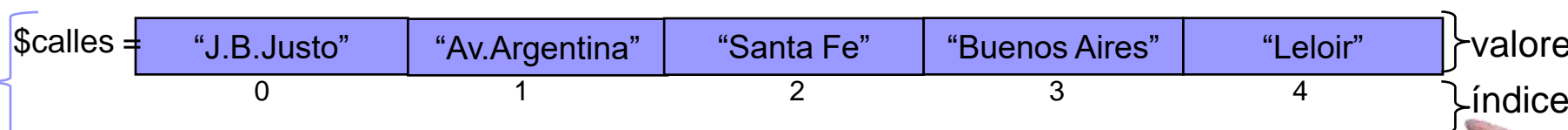


# CREAR un Arreglo Indexado

¿Cuáles son las instruc. para crear un arreglo indexado?

Ejemplo 2: un arreglo con 5 nombres de calles de la ciudad

Representación:



**Opción 1 para inicializar)**

```
$calles = array( "J.B.Justo", "Av.Argentina", "Santa Fe", "Buenos Aires", "Leloir" );
```

**Opción 2 para inicializar)**

```
$calles = [ "J.B.Justo", "Av.Argentina", "Santa Fe", "Buenos Aires", "Leloir" ] ;
```

**Opción 3 para inicializar)**

```
$calles = array( ); o bien $calles = [ ];
```

```
$calles[0] = "J.B.Justo";
```

```
$calles[1] = "Av.Argentina";
```

```
$calles[2] = "Santa Fe";
```

```
$calles[3] = "Buenos Aires";
```

```
$calles[4] = "Leloir";
```

Inicializo el arreglo en vacio

El primer elemento se almacena la posición o índice 0 (cero)

```
$cantidadCalles = count($calles);
```



# ACCEDER un Arreglo Indexado

¿Cómo acceder a los elementos del arreglo indexado?

Los arreglos unidimensionales o indexados  
se **acceden** así:

**\$nombreArreglo** *indice* **]**

Nombre  
del arreglo

Corchete  
abre

Literal entero  
(ej.: 3 )  
ó  
variable que  
Almacena un  
valor entero:  
(ej: \$i)

Corchete  
cierra



# Arreglos Indexados

## ¿Cómo acceder a los elementos del arreglo indexado?

\$misNombres =	“lucia”	“carlos”	“euge”	“mario”	“marta”
	0	1	2	3	4

a) Imprimir en pantalla el 3er nombre almacenado en el arreglo:

```
<?php
```

```
$misNombres = [“lucia”, “carlos”, “euge”, “mario”, “marta”]; } Creo el arreglo
```

```
echo “nombre es: “. $misNombres[ 2 ]; } Accedo al valor  
almacenado en  
la posición 2
```



# Arreglos Indexados

¿Cómo acceder a los elementos del arreglo indexado?

\$misNombres =	"lucia"	"carlos"	"euge"	"mario"	"marta"
	0	1	2	3	4

a) Imprimir en pantalla el 3er nombre almacenado en el arreglo:

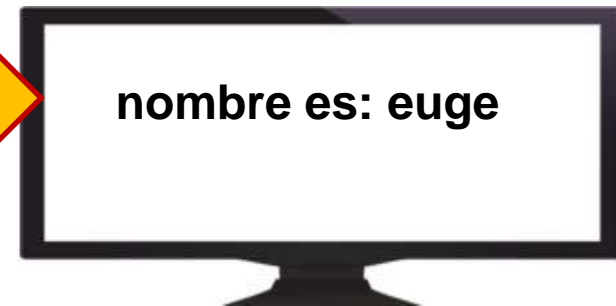
```
<?php
```

```
$misNombres = ["lucia", "carlos", "euge", "mario", "marta"] ;
```

```
echo "nombre es: ". $misNombres[ 2 ] ;
```

**EJECUTAR EL PROGRAMA:**

\$misNombres =	"lucia"	"carlos"	"euge"	"mario"	"marta"
	0	1	2	3	4





# Arreglos Indexados

## ¿Cómo acceder a los elementos del arreglo indexado?

\$misNombres =	"lucia"	"carlos"	"euge"	"mario"	"marta"
	0	1	2	3	4

b) Imprimir en pantalla el nombre elegido por un usuario, la elección es por número de nombre:

```
<?php
```

```
$misNombres = ["lucia", "carlos", "euge", "mario", "marta"] ; } Creo el arreglo
```

```
echo "Ingrese el número de nombre a imprimir:" ;
```

```
$nro = trim(fgets(STDIN));
```

```
echo "nombre es: ". $misNombres[$nro]; } Accedo al valor almacenado en  
la posición indicada por usuario
```



# Arreglos Indexados

## ¿Cómo acceder a los elementos del arreglo indexado?

b) Imprimir en pantalla el nombre elegido por un usuario, la elección es por número de nombre:

```
<?php
```

```
$misNombres = ["lucia", "carlos", "euge", "mario", "marta"] ;
```

```
echo "Ingrese el número de nombre a imprimir:" ;
```

```
$nro = trim(fgets(STDIN));
```

```
echo "nombre es: ". $misNombres[$nro];
```

### EJECUTAR EL PROGRAMA:

\$misNombres =

"lucia"	"carlos"	"euge"	"mario"	"marta"
0	1	2	3	4



\$nro =



Ingrese le número de nombre a imprimir  
nombre es: carlos





# Arreglos Indexados

## ¿Cómo acceder a los elementos del arreglo indexado?

b) Imprimir en pantalla el nombre elegido por un usuario, la elección es por número de nombre:

```
<?php
```

```
$misNombres = ["lucia", "carlos", "euge", "mario", "marta"] ;
```

```
echo "Ingrese el número de nombre a imprimir:" ;
```

```
$nro = trim(fgets(STDIN));
```

```
echo "nombre es: ". $misNombres[$nro];
```

### EJECUTAR EL POGRAMA:

\$misNombres =	"lucia"	"carlos"	"euge"	"mario"	"marta"
	0	1	2	3	4

\$nro =

8



Ingrese le número de nombre a imprimir  
nombre es: **ERROR: Undefined offset!!**



# Arreglos Indexados

## ¿Cómo acceder a los elementos del arreglo indexado?

\$misNombres =	“lucia”	“carlos”	“euge”	“mario”	“marta”
	0	1	2	3	4

b) Imprimir en pantalla el nombre elegido por un usuario, la elección es por número de nombre:

**<?php**

```
$misNombres = [“lucia”, “carlos”, “euge”, “mario”, “marta”] ; } Creo el arreglo
```

```
echo “Ingrese el número de nombre a imprimir:” ;
```

```
$nro = trim(fgets(STDIN));
```

```
If ($nro>=0 && $nro< count($misNombres) {
```

```
    echo “nombre es: ”. $misNombres[$nro];
```

} Acceso al valor almacenado en la posición indicada por usuario

```
}
```

```
else{
```

```
    echo “no existe un nombre para el numero “. $nro }
```

} Mje de error si no existe la posición

```
}
```



# Arreglos Indexados

## Ejercicio:

Almacenar en un arreglo Indexado nombres de países, solicitando a un usuario primero la cantidad de países, y segundo el nombre de cada país. (utilizar ciclo definido)

### **Antes de realizar escribir instrucciones PHP, analicemos:**

- 1) Si tengo que almacenar nombres de países, ¿De qué tipo de dato es la Colección a implementar?
- 2) ¿cómo voy a nombrar o etiquetar a la colección?
- 3) Por ejemplo: Cómo representar en papel el arreglo de 5 países:  
Argentina, Chile, Perú, Brasil y Bolivia
- 4) ¿Qué retorna la función **count** teniendo como parámetro de entrada a la colección de ejemplo?



# Arreglos Indexados

## Ejercicio:

Almacenar en un arreglo Indexado nombres de países, solicitando a un usuario primero la cantidad de países, y segundo el nombre de cada país. (utilizar ciclo definido)

**Antes de realizar el ejercicio responder:**

1) Si tengo que almacenar nombres de países, ¿De qué tipo de dato es la Colección a implementar?

array de string

2) Cómo voy a nombrar a la colección? **\$países**

3) Por ejemplo: Cómo representar el arreglo de 4 países: Argentina, Chile, Perú, Brasil

\$países =	"Argentina"	"Chile"	"Perú"	"Brasil"
	0	1	2	3

4) ¿Qué retorna la función **count** teniendo como parámetro de entrada a esta colección de ejemplo?

retorna el entero 5

*/\* Programa principal \*/*

**echo** "ingrese cant paises";

\$cantPaises = trim(fgets(STDIN));

\$países = [ ];

**echo** count(\$países) . "\n";

**for** (\$i=0; \$i<\$cantPaises; \$i++){

**echo** "Pais: ";

**\$países[ \$i ]** = trim(fgets(STDIN));

}

**echo** count(\$países) . "\n";