



Introducción a los arreglos unidimensionales (Importancia)

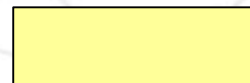
Importancia de usar arreglos unidimensionales



Examina el siguiente código que permite leer 5 datos enteros desde teclado:

```
int x;  
  
for(int i=1;i<=5;i++)  
{  
    cout<<"Ingrese valor "<<i<<": ";  
    cin>>x;  
}
```

x



Se esta haciendo uso de una **sola variable** para almacenar en cada iteración del for el valor leído.

Importancia de usar arreglos unidimensionales



```
int x;  
  
for(int i=1;i<=5;i++)  
{  
    cout<<"Ingrese valor "<<i<<": ";  
    cin>>x;  
}  
  
cout<<"\nEl segundo valor leído fue: ";  
cout<<????
```



Después del for no podríamos tener acceso (por ejemplo) al segundo dato ingresado ya que no existe.
Esto se debe a que el valor de x se habrá quedado con el último dato leído.

Importancia de usar arreglos unidimensionales



Podríamos usar **5 variables** para almacenar cada dato:

```
int x1,x2,x3,x4,x5;  
cout<<"Ingrese valor 1: ";  
cin>>x1;  
cout<<"Ingrese valor 2: ";  
cin>>x2;  
cout<<"Ingrese valor 3: ";  
cin>>x3;  
cout<<"Ingrese valor 4: ";  
cin>>x4;  
cout<<"Ingrese valor 5: ";  
cin>>x5;
```



¿Pero, en caso se desearan leer y almacenar N datos ?

Importancia de usar arreglos unidimensionales



Usar un arreglo unidimensional sería lo mas factible:

```
int numeros[5];  
  
for(int i=0;i<5;i++)  
{  
    cout<<"Ingrese valor: "<<i+1<<": ";  
    cin>>numeros[i];  
}
```

Los datos leídos se van almacenando en cada posición del arreglo

Se crea un arreglo unidimensional de nombre **numeros** con capacidad para almacenar máximo 5 datos de tipo **entero**

<i>numeros</i>	
0	
1	
2	
3	
4	

Importancia de usar arreglos unidimensionales



```
int numeros[5];

for(int i=0;i<5;i++)
{
    cout<<"Ingresa valor: "<<i+1<<": ";
    cin>>numeros[i];
}

cout<<"\nEl segundo valor leído fue: ";
cout<<numeros[1];
```

Después del for ya se podría tener acceso (por ejemplo) al segundo dato leído ya que esta almacenado en el arreglo.

	<i>numeros</i>
0	34
1	12
2	56
3	67
4	134

Importancia de usar arreglos unidimensionales



Los arreglos no solamente pueden almacenar datos enteros, pueden también almacenar, datos reales , incluso hasta cadenas

Ejemplo:

```
string paises[5];  
  
for(int i=0;i<5;i++)  
{  
    cout<<"Ingrese pais: "<<i+1<<" : ";  
    getline(cin,paises[i]);  
}
```

Se crea un arreglo unidimensional de nombre **paises** con capacidad para almacenar máximo 5 cadenas.

Los datos leídos se van almacenando en cada posición del arreglo

<i>paises</i>	
0	
1	
2	
3	
4	

Importancia de usar arreglos unidimensionales



```
string paises[5];

for(int i=0;i<5;i++)
{
    cout<<"Ingrese pais: "<<i+1<<": ";
    getline(cin,paises[i]);
}

cout<<"\nEl primer pais leido fue: ";
cout<<paises[0];
```

Después del for ya se podría tener acceso (por ejemplo) a la primera cadena leída ya que esta almacenada en el arreglo.

<i>paises</i>	
0	"Peru"
1	"Brasil"
2	"Chile"
3	"Ecuador"
4	"Colombia"



En la siguiente sesión presencial , estudiaremos detalladamente el uso de arreglos en la programación