

Fundamentos de la Programación

¿Qué es un arreglo?

Un **array** o **arreglo** es un **conjunto finito y ordenado de elementos homogéneos**. La característica de ser "**ordenado**" implica que cada elemento, desde el primero hasta el último, puede ser identificado. Además, los elementos de un arreglo son **homogéneos**, lo que significa que son del mismo tipo de datos. Por ejemplo, un arreglo puede estar compuesto únicamente por elementos de tipo cadena, otro solo por elementos de tipo entero, etcétera, pero no puede contener datos de tipos distintos.

Otra característica fundamental de los arreglos es su incapacidad de ser redimensionados. Al declararlos, se establece tanto el tipo de dato que contendrán como su capacidad de almacenamiento, es decir, cuántos "espacios" tienen disponibles para guardar información. Esta limitación los hace **finitos** y les impide alterar su tamaño una vez definidos.

Para dar un ejemplo y comprender mejor su funcionamiento, imagina que eres un chef en un restaurante y estás organizando los estantes de tu despensa. Decides utilizar estantes con compartimentos, donde cada estante puede contener una cierta cantidad de compartimentos para los ingredientes.

Al principio, determinas que necesitas tres estantes para organizar tus ingredientes: uno para las especias y condimentos, otro para los productos enlatados y uno más para los granos y cereales. Asignas a cada tipo de ingrediente su estante correspondiente y colocas los productos en sus respectivos compartimentos.

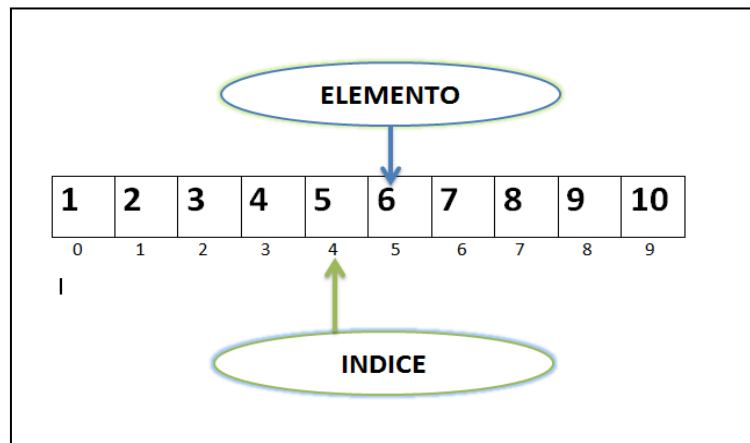
Este enfoque se asemeja al concepto de arreglos. Cada estante representa un "arreglo" de un tipo de dato específico, y cada compartimento es un espacio en el arreglo.. Sin embargo, una vez que has decidido cuánto espacio necesitas no puedes agregar más estantes ni ampliar la cantidad de los compartimentos existentes. El espacio disponible está fijo, al igual que en los arreglos, donde la capacidad de almacenamiento se define al principio y no se puede cambiar después.

Los arreglos pueden tener múltiples dimensiones, aunque en la práctica se utilizan comúnmente en una o dos dimensiones para facilitar su manipulación. En este contexto, se les denomina **vectores o matrices**, según corresponda.

¿Qué es un vector?

El tipo más elemental de arreglo es el **arreglo unidimensional o vector**. Este consiste en una secuencia de elementos, dispuestos uno detrás del otro, y posee las siguientes características:

- Se **identifica** por un único nombre de variable.
- Sus elementos se almacenan en posiciones dentro del vector, cada una asociada a un subíndice.
- Se puede acceder a cada uno de sus elementos mediante su subíndice, ya sea de manera ordenada o aleatoria.
- Su **tamaño es finito**, lo que implica que una vez que se define, no puede cambiar. El tamaño se refiere a la cantidad máxima de elementos que puede contener el vector.



El índice o subíndice es un número entero que identifica cada elemento dentro del vector, independientemente del tipo de dato que contenga. Un vector de tamaño N consta de N subíndices que se suceden en orden creciente. Por ejemplo: 0 - 1 - 2 - 3 - 4 - 5 - 6 - N. El valor inicial del primer subíndice varía según el lenguaje de programación; la mayoría de los lenguajes modernos comienzan con el índice cero. Por lo tanto, en PSeInt, los subíndices comenzarán en cero y los valores posibles irán desde 0 hasta N-1.

Declaración

```
Definir nombre_vector como Tipo_de_Dato  
Dimension nombre_vector(tamaño)
```

La declaración 'Dimension' se utiliza para asignar tamaño a un vector de cualquier tipo de dato simple, como entero, real, cadena o lógico. Este tamaño debe ser un número entero o una variable entera; no puede ser un número con decimales. La dimensión especificada determina cuántos elementos puede almacenar el vector. Es importante tener en cuenta que una vez declarado, el tamaño del vector no puede cambiar.

ASIGNAR ELEMENTOS A UN ARREGLO

Cuando desees ingresar un elemento en tu arreglo, selecciona el subíndice donde desees almacenarlo. Luego, invoca tu vector por su nombre, seguido de paréntesis con el subíndice deseado. Utiliza el signo igual (que es el operador de asignación) seguido del elemento que desees guardar. Recuerda que el elemento a guardar debe coincidir con el tipo de dato de tu arreglo; por ejemplo, si tu arreglo es de tipo entero, solo puedes guardar números enteros. Además, asegúrate de que los subíndices sean válidos según el tamaño de tu arreglo. Por ejemplo, si tu arreglo tiene un tamaño de 5, no podrás usar el subíndice 6, ya que este no existe.

Formas de asignar valores a los arreglos en PSeInt:

Existen diversas formas de asignar valores a los arreglos en PSeInt. Una de ellas es la asignación manual, donde los valores se ingresan uno por uno en cada posición del arreglo. Otra forma es recorrer el arreglo utilizando un bucle y asignar valores según la necesidad, como de uno en uno o utilizando fórmulas matemáticas para calcular los valores.

Ejemplo: Asignación Manual:

```
// Asignación manual de valores a cada posición  
nombre_arreglo(0) = 10  
nombre_arreglo(1) = 20  
nombre_arreglo(2) = 30  
nombre_arreglo(3) = 40  
nombre_arreglo(4) = 50
```

En este ejemplo, los valores se asignan manualmente a cada posición del arreglo.

Ejemplo: Recorriendo el Arreglo:

```
// Asignación de valores usando un bucle Para
Para i ← 0 Hasta 4 Con Paso 1 Hacer
    nombre_arreglo(i) = i * 10
Fin Para
```

En este caso, se utiliza un bucle Para para recorrer el arreglo y asignar valores de manera dinámica, multiplicando el índice por 10 para cada posición del arreglo.

Mostrar o acceder a elementos de un arreglo:

Cuando necesitas mostrar o acceder a un elemento específico de tu arreglo, simplemente escribe el nombre del arreglo seguido de corchetes o paréntesis con el índice del elemento que deseas obtener.

Por ejemplo: `Escribir nombre_arreglo(0)`

Estas líneas te permiten acceder al primer elemento del arreglo `nombre_arreglo`.

Si deseas mostrar todos los elementos del arreglo, puedes utilizar una estructura Para que recorra todos los subíndices del arreglo y los muestre uno por uno.

```
Para i ← 0 Hasta 4 Con Paso 1 Hacer
    Escribir nombre_arreglo[i]
Fin Para
```

Por ejemplo:

En este bucle, la variable `i` recorre todos los subíndices del arreglo, desde 0 hasta 4. Es importante recordar que los arreglos comienzan en el índice 0, por lo que si el tamaño del arreglo es 5, necesitamos que nuestro bucle Para vaya desde 0 hasta 4 para recorrer todos los elementos del arreglo.

Uso en Subprogramas

Los arreglos pueden ser pasados como parámetros a un subprograma (función o procedimiento) de la misma manera que las variables escalares. Sin embargo, **es importante tener en cuenta que los arreglos**, a diferencia de los tipos de datos simples, **se pasan siempre como parámetros "Por Referencia"**. Esto significa que, al utilizar arreglos en nuestros subprogramas, generalmente los modificamos, rellenamos o mostramos.

Ejemplo de Función:

```
Funcion variable_de_retorno ← Nombre (vector por referencia)
    Definir variable_de_retorno como Tipo_de_Dato
    <acciones>
```

En este ejemplo, la función Nombre recibe un arreglo como parámetro por referencia. Puedes realizar operaciones sobre este arreglo dentro de la función y devolver un valor al final.

Ejemplo de Procedimiento:

```
SubProceso Nombre (vector por referencia)
    <acciones>
FinSubProceso
```

En este ejemplo, el procedimiento Nombre recibe un arreglo como parámetro por referencia. Puedes realizar cualquier operación sobre este arreglo dentro del procedimiento.