

Arreglos

Juan Pablo Cobá Juárez Pegueros
Algoritmos y Programación Básica
Bioingeniería Médica
Facultad de Medicina
2015

There are two types of people.

```
if (Condition)
{
    Statements
    /*
     *
     */
}
```

```
if (Condition) {
    Statements
    /*
     *
     */
}
```

Programmers will know.

Contenido

- Objetivo
- ¿Qué son los Arreglos?
- ¿Cómo son los Arreglos?
- Características de los Arreglos
- Dificultad con los Arreglos
- Tipos de Arreglos
- Reglas para los identificadores
- Como se declara un Arreglo unidimensional
- Representación de un Arreglo en memoria
- Elementos que conforman un Arreglo
- Asignar elementos a un Arreglo
- Imprimir los elementos de un Arreglo
- Reglas para los Arreglos
- Operaciones con elementos de un Arreglo
- Inicialización de los Arreglos unidimensionales
- Inicialización de un Arreglo adimensional vacío
- Inicialización de un Arreglo adimensional con valores iniciales
- Operaciones con arreglos
- Ejemplo lectura de un arreglo
- Ejemplo escribir de un arreglo
- Ejemplo asignación de un arreglo
- Bibliografía

Objetivo: Utilizar las estructuras de datos estáticas, codificándolas en pseudocódigo para la manipulación y alojamiento eficiente de datos en memoria estática

Tema: Arreglos

Unidad 5. Introducción a las Estructuras Estáticas



¿Qué son los arreglos?

Al inicio del semestre se definieron los tipos de datos que se encuentran divididos de la siguiente manera:

- ▶ Datos simples
- ▶ Datos estructurados

Estos últimos son los que nos interesan permiten resolver ciertos problemas de manera más rápida que si lo intentáramos con los de tipo simple

¿Qué son los arreglos?

- ▶ Un arreglo es un tipo de dato estructurado que almacena en una variable un conjunto limitado de datos del mismo tipo almacenados en direcciones de memoria contigua donde la dirección más baja corresponde al inicio del elemento y la dirección más alta al último, su identificador apunta al primer elemento del arreglo.
- ▶ Cada elemento puede ser identificado de manera única de acuerdo a la posición dentro del arreglo donde se encuentre

Definición de un arreglo

- ▶ “Un arreglo es una colección finita, homogénea y ordenada de elementos ”
- ▶ Finita por que todo arreglo tiene un límite
- ▶ Homogénea por que todos los elemento del arreglo deben ser del mismo tipo
- ▶ Ordenada por que se puede determinar cual es el primer elemento, cual es el segundo , etc

¿Cómo son los arreglos?



¿Cómo son los arreglos?

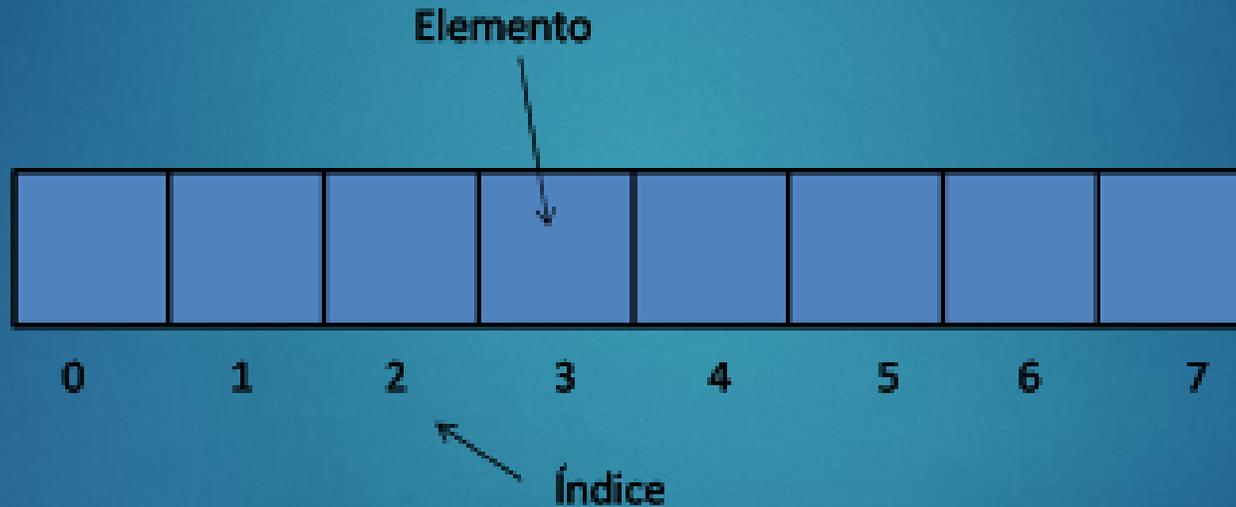
- ▶ Gráficamente un arreglo puede representarse de la siguiente forma



- ▶ Básicamente es un contenedor con casillas donde podemos almacenar cualquier tipo de dato (Fila o cola)

¿Cómo son los arreglos?

- ▶ Dentro de estas casillas podemos almacenar los datos



- ▶ Los índices nos ayudan a identificar la casilla donde almacenamos un dato

Características de los arreglos

- ▶ Son una lista finita de elementos del mismo tipo
- ▶ En memoria un arreglo se guarda en segmentos de memoria contiguos
- ▶ Tiene un identificador que representa a todos los elementos que conforman el arreglo.
- ▶ Podemos acceder a cualquier elemento del arreglo utilizando índices
- ▶ Para acceder a un elemento en particular debemos conocer el identificador del arreglo y el índice del elemento

Dificultad con los arreglos

- ▶ Con los arreglos debemos tener cuidado por las siguientes razones:
- ▶ Un arreglo muy grande es ineficiente si solo se ocupan pocos de sus elementos (desperdicio de memoria)
- ▶ Un arreglo muy pequeño traerá problemas si me quedo sin espacio para almacenar datos
- ▶ En cualquiera de los casos no produce error, solo si lo codificamos si nos quedamos sin memoria el programa dejara de trabajar

Tipos de arreglos

- ▶ Los tipos de arreglos varia dependiendo de la cantidad de elementos y como se organizan la clasificación más sencilla es la siguiente:
- ▶ Unidimensional (vectores o listas)
- ▶ Bidimensional (tablas o matrices)
- ▶ Multidimensional (más de dos dimensiones)
- ▶ Los más utilizados son los unidimensionales y los bidimensionales

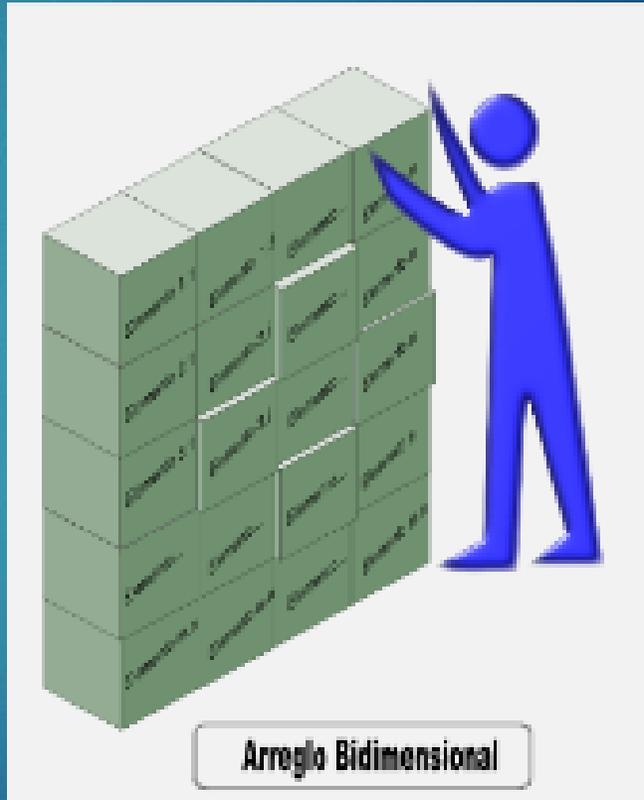
Arreglo unidimensional

- Compuesto por una fila
- Puedes ser visto como lista de elementos
- Comúnmente sus identificadores son i
- El índice inicia en 0



Arreglo bidimensional

- Compuesto por filas y columnas
- Puedes ser visto como una matriz
- Comúnmente sus identificadores son i y j
- El índice inicia en $0,0$



Arreglos multidimensionales

- Compuesto por filas , columnas y profundidad
- Puedes ser visto como un cubo de rubik
- Comúnmente sus identificadores son i, j, k
- El índice inicia en $0,0,0$



Reglas para los identificadores

- ▶ Debe de comenzar con una letra mayúscula o minúscula, no pueden contener espacio en blanco
- ▶ Sensible a mayúsculas y minúsculas
- ▶ Letras, dígitos y carácter bajo están permitidos después del primer carácter
- ▶ No pueden existir identificadores iguales
- ▶ No se pueden utilizar palabras reservadas
- ▶ Deben ser nombres significativos a la función que realizaran
- ▶ No utilizar nombre muy largos

¿Cómo se declara un arreglo unidimensional

Seudocódigo

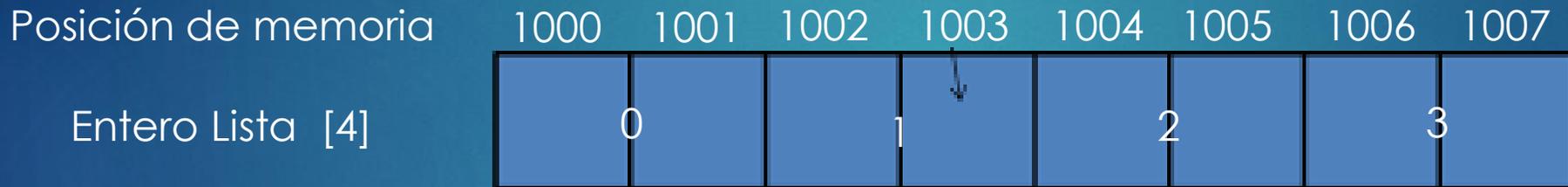
- ▶ Tipo_dato identif_arreglo [tam_arreglo]

Donde

- ▶ **Tipo_dato** se refiere al tipo de dato del arreglo puede ser entero, real, carácter, etc
- ▶ identif_arreglo es el nombre que identifica a todo el arreglo
- ▶ Tam_arreglo es la cantidad de elementos que forman parte del arreglo

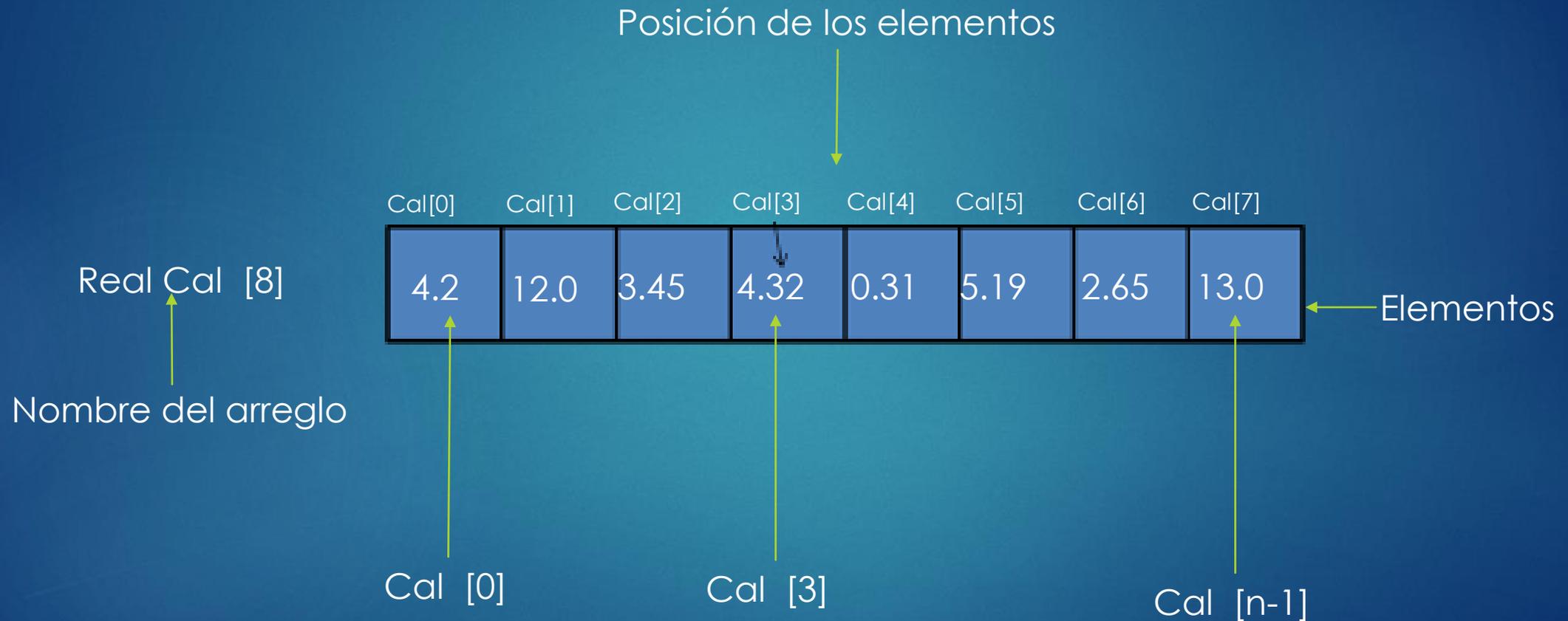
Representación de un arreglo en memoria

- ▶ La cantidad de arreglos y su tamaño dependerá de la memoria libre disponible



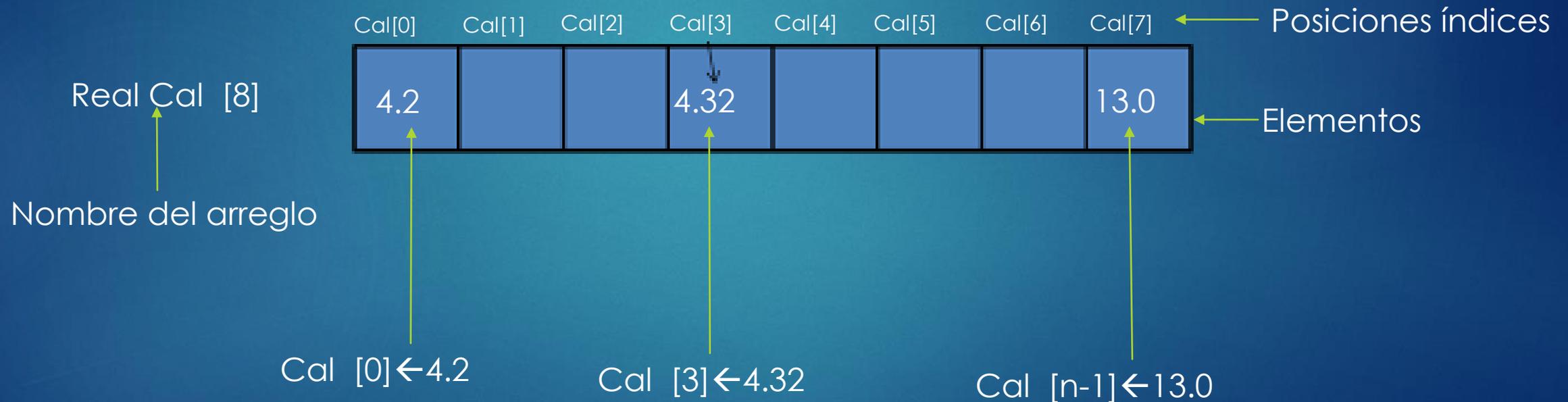
- ▶ Los tipos de datos enteros requieren de dos localidades de memoria

Elementos que conforman un arreglo



Asignar elementos a un arreglo

- Para asignar un valor al elemento debemos referenciar la posición donde queremos guardar el dato y utilizar el operador de asignación “←”



Imprimir los elementos de un arreglo

- ▶ Para poder imprimir los elementos de un arreglo podemos asignarlos a una variable
- ▶ Otra opción es imprimir directamente el contenido de la posición que deseamos imprimir

Cal[0]	Cal[1]	Cal[2]	Cal[3]	Cal[4]	Cal[5]	Cal[6]	Cal[7]
4.2	12.0	3.45	4.32	0.31	5.19	2.65	13.0

Opción 1

$a \leftarrow \text{Cal}[0]$
Imprimir "el valor de la posición 1 es ",a

Opción 2

Imprimir "el valor de la posición 8 es ",Cal[7]

Reglas para los arreglos

- ▶ Los índices (posición) de los elementos solo pueden ser enteros
- ▶ Los elementos del arreglo son del tipo con el que fue declarado el arreglo
- ▶ Se permite realizar operaciones con los elementos del arreglo incluso pueden ser diferentes operaciones por elemento del arreglo
- ▶ Es indispensable anteponer el identificador del arreglo a la posición con la que queremos trabajar

Operaciones con elementos de un arreglo

- ▶ Dividir el contenido de un elemento en el arreglo
- ▶ Sumar el contenido de de los elementos que se encuentran dentro del arreglo
- ▶ las operaciones aritmeticas con los indices están permitidas

División

$$B \leftarrow \text{Cal}[n] / 2$$

Sumatoria

$$A \leftarrow \text{Cal}[0] + \text{Cal}[1] + \text{Cal}[2] + \text{Cal}[3]$$

Operaciones con índices

$$\text{Si } i = 2 \text{ y } j = 4 \\ \text{Cal}[i+j] \leftarrow 10$$

Inicialización de arreglos unidimensionales

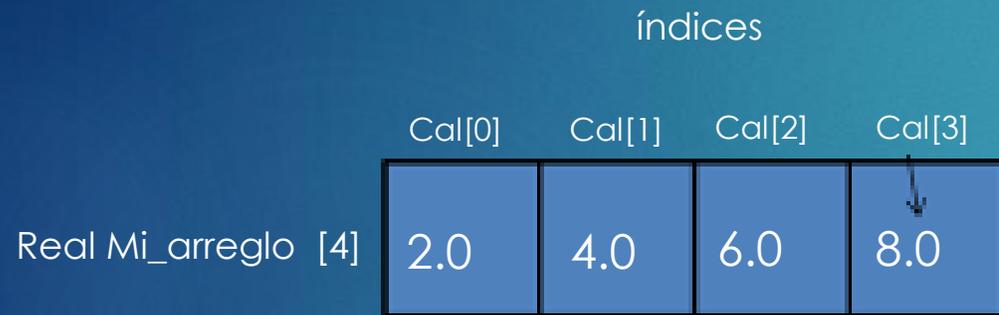
- ▶ Podemos inicializar un arreglo de las siguiente manera:
 - ▶ Vacío
 - ▶ Con valores iniciales
- ▶ Según las necesidades de nuestro algoritmo podemos utilizar cualquiera de las 2 opciones.

Inicialización de un arreglo adimensional vacío



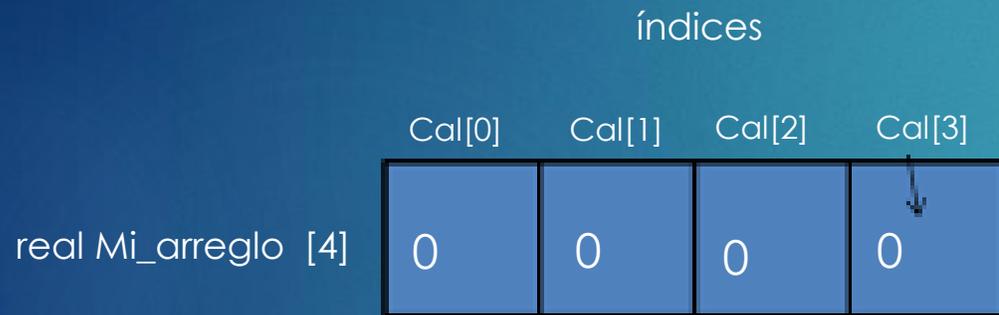
- ▶ Forma general
- ▶ Tipo_dato identif_arreglo [tam_arreglo]
- ▶ Ejemplo
- ▶ `real Mi_arreglo [4]`

Inicialización de un arreglo adimensional con valores iniciales



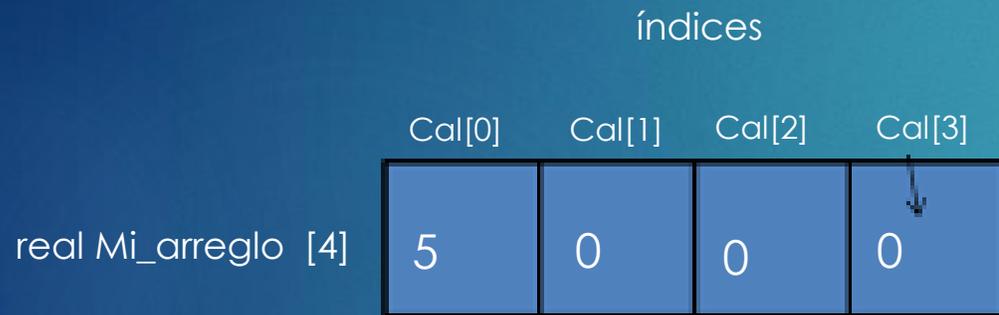
- ▶ Forma general
- ▶ Tipo_dato identif_arreglo [tam_arreglo] ← {valores}
- ▶ Ejemplo
- ▶ `real Mi_arreglo [4] ← {2.0,4.0,6.0,8.0}`
- ▶ Los elemntos se deben separa por “,” (comas)

Inicialización de un arreglo adimensional con valores iniciales



- ▶ Forma general
- ▶ `Tipo_dato identif_arreglo [tam_arreglo] = {0};`
- ▶ Ejemplo
- ▶ `real Mi_arreglo [4] ← {0};`
- ▶ Llena un arreglo unidimensional con puros ceros

Inicialización de un arreglo unidimensional con valores iniciales



- ▶ Forma general
- ▶ Tipo_dato identif_arreglo [tam_arreglo] ← {5};

- ▶ Ejemplo

```
real Mi_arreglo [4] ← {5};
```

Llenar un arreglo unidimensional con el primer elemento con el valor 5 y los demás se llenan en 0

Operaciones con arreglos

- ▶ Las operaciones mas comunes con arreglos son:
- ▶ Lectura/escritura
- ▶ Asignación
- ▶ Actualización
 - ▶ Inserción
 - ▶ Eliminación
 - ▶ Modificación
- ▶ Ordenación
- ▶ Búsqueda

Operaciones con arreglos

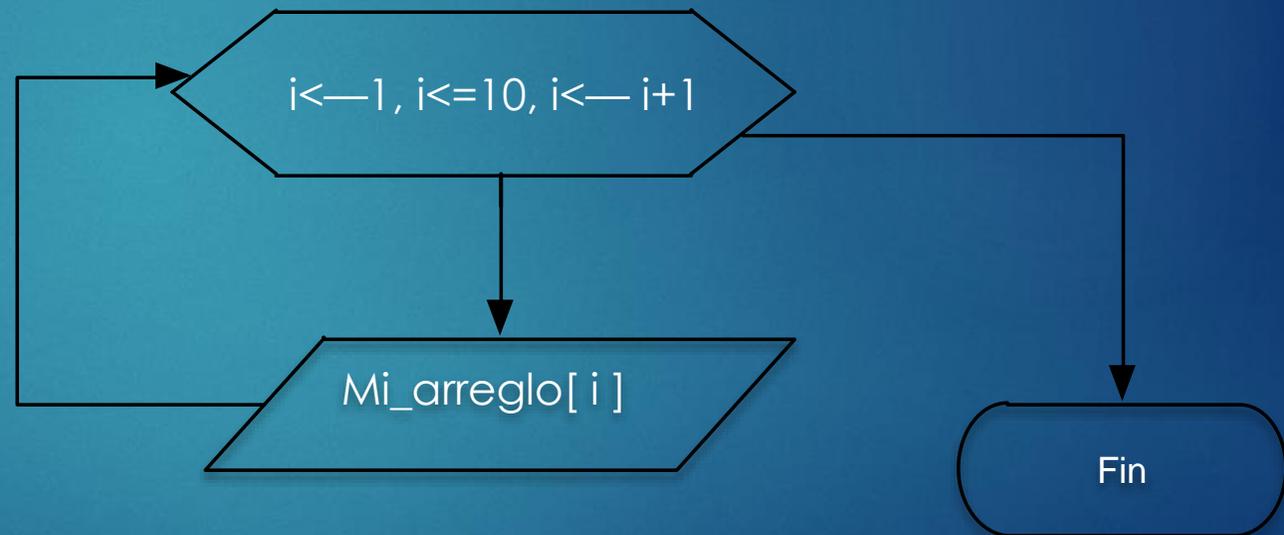
- ▶ Se debe tomar en cuenta que las operaciones anteriores no pueden realizarse de manera global por lo que se deben aplicar sobre cada elemento.
- ▶ Por lo que regularmente nos ayudamos de la estructura de repetición “para” para aplicar las operaciones

Ejemplo lectura de un arreglo

Pseudocodigo

- ▶ INICIO
- ▶ Mi_arreglo [10]: entero
- ▶ i : entero
- ▶ Para ($i \leftarrow 1, i \leq 10, i \leftarrow i+1$)
- ▶ Leer Mi_arreglo[i]
- ▶ Finpara
- ▶ FIN

Diagrama de flujo

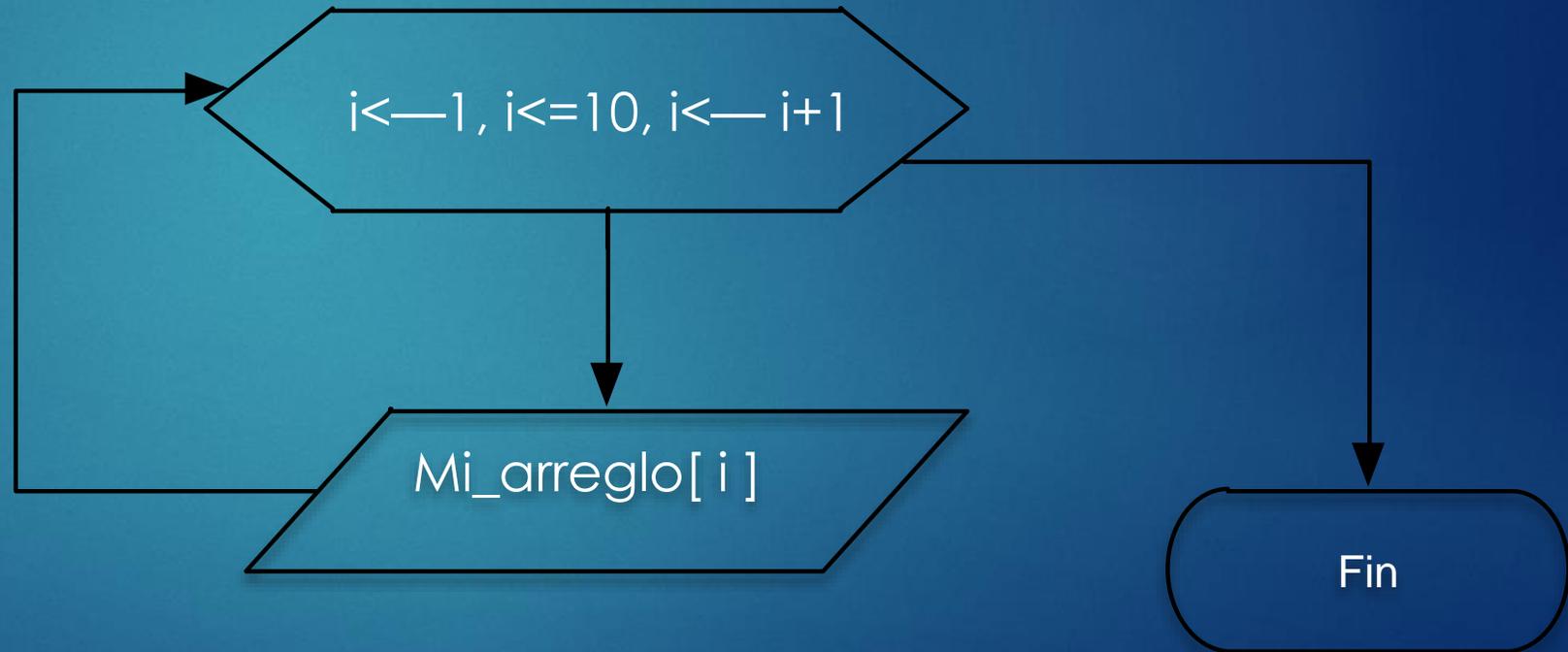


Ejemplo escribir de un arreglo

Pseudocodigo

- ▶ INICIO
- ▶ Mi_arreglo : entero
- ▶ i : entero
- ▶ Para ($i \leftarrow 1, i \leq 10, i \leftarrow i+1$)
- ▶ Escribir Mi_arreglo[i]
- ▶ Finpara
- ▶ FIN

Diagrama de flujo

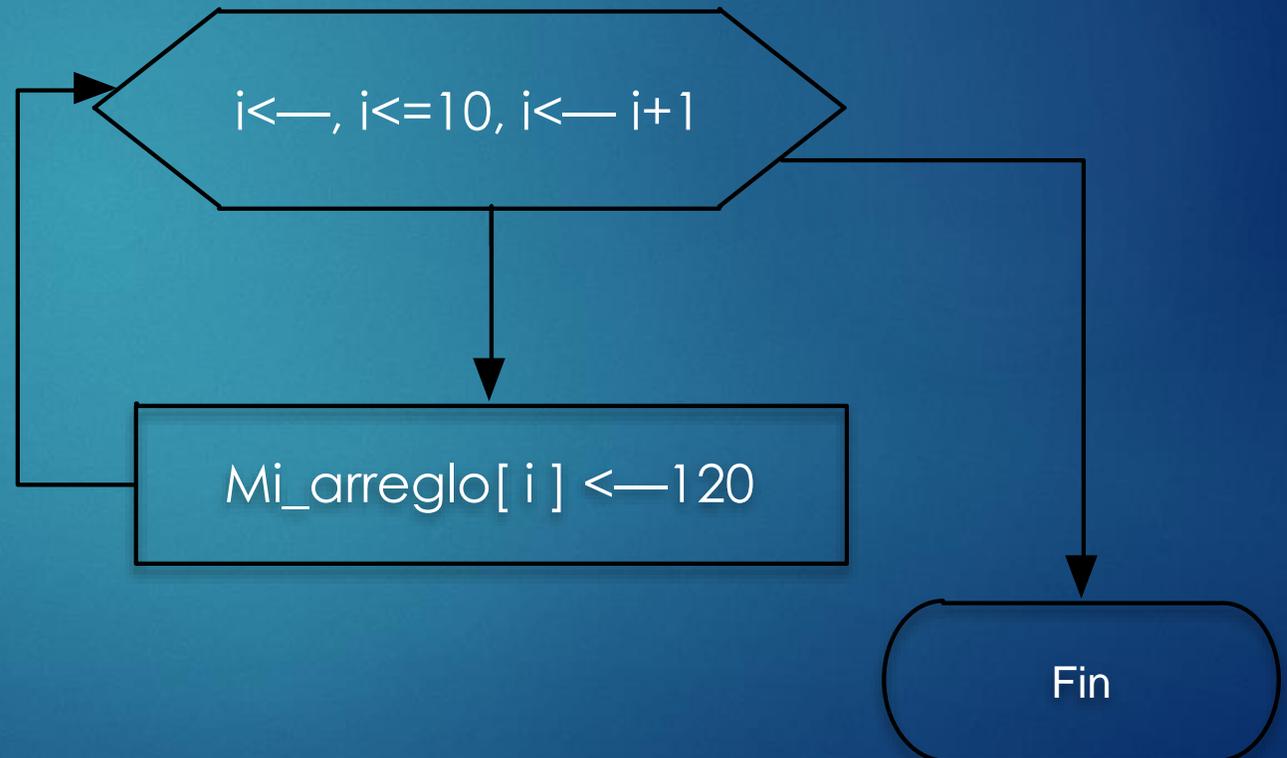


Ejemplo asignación de un arreglo

Pseudocódigo

- ▶ INICIO
- ▶ Mi_arreglo : entero
- ▶ i : entero
- ▶ Para ($i \leftarrow 1, i \leq 10, i \leftarrow i+1$)
- ▶ Hacer $Mi_arreglo[i] \leftarrow 120+i$
- ▶ Fin para
- ▶ FIN

Diagrama de flujo



Ejercicio

Diseñar un Algoritmo que permite guardar 10 calificaciones en un arreglo, y muestre en pantalla el promedio de las calificaciones introducidas, así como el número de calificaciones reprobatorias así como el número de calificaciones aprobatorias.

Calificación reprobatoria ≤ 5.9

Calificación aprobatoria ≥ 6

Validar que el usuario no introduzca valores fuera de rango

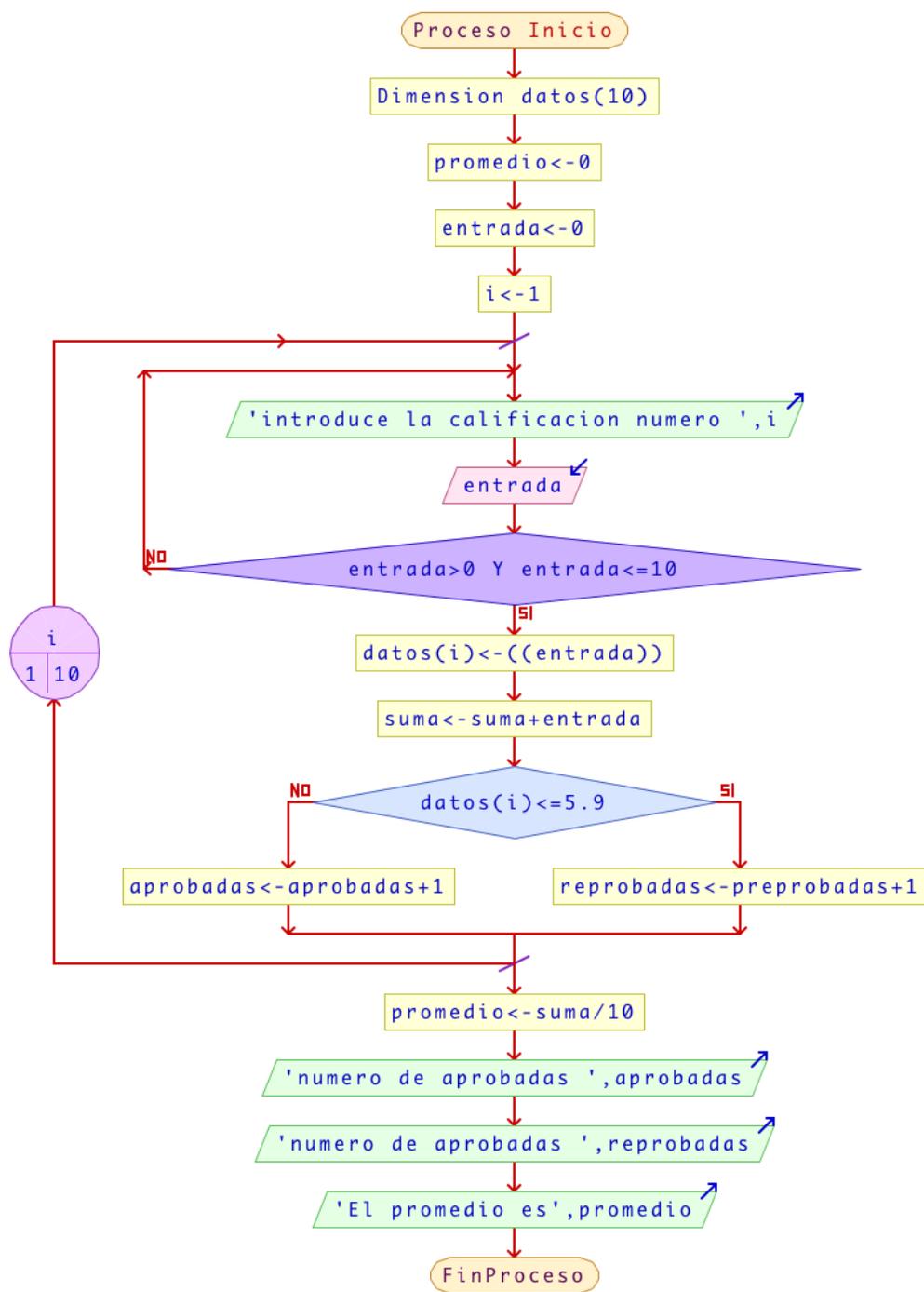
Solución

```
Inicio
Real datos[10]
Real promedio<-0
Real entrada<-0
Entero i<-1 de tipo entero
Para i<-1, i<=10, i<-i+1

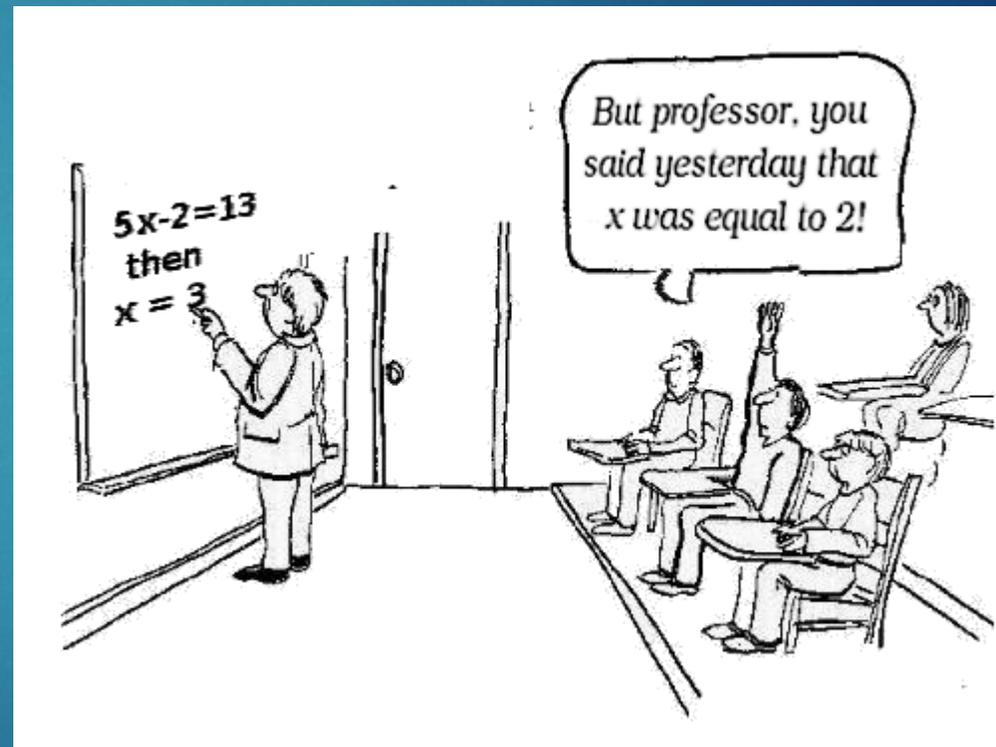
    Repetir
        Escribir "introduce la calificación numero ", i
        Leer entrada
        Hasta (entrada > 0 y entrada <= 10)

        datos[i]<-entrada
        suma<-suma+entrada
        Si datos[i]<=5.9 Entonces
            reprobadas<-reprobadas+1
        Sino
            aprobadas<-aprobadas+1
        Fin Si
    Fin Para
promedio<-suma/10
Escribir "numero de aprobadas ", aprobadas
Escribir "numero de reprobadas ", reprobadas
Escribir "El promedio es", promedio
Fin
```

Diagrama de Flujo



Gracias !!!!!!!



Bibliografía

- ▶ Diseño de algoritmos y su codificación en lenguaje C / María Adriana Corona Nakamura, María de los Angeles Ancona Valdez. 1a ed. México : Mcgraw Hill, 2011.
- ▶ Metodología de la programación: Algoritmos, diagramas de FLUJO El y Programas / Osvaldo Cairó Battistutti. Ed 3a. México, D. F. : Alfaomega, 2005.
- ▶ Diseño de programas / Antonio Menchén Peñuela. México, D.F. : Alfaomega, 2010.
- ▶ Análisis y diseño de **algoritmos** : implementaciones en C y pascal / Gustavo López, Ismael Jeder, Augusto Vega. Buenos Aires ; México, D. F. : Alfaomega, c2009.