Arreglos

Fundamentos de programación

Agenda

- Introducción a la utilización de arreglos.
- Definición de arreglos.
- Inicialización de arreglos.
- Utilización de arreglos con funciones.

Introducción a arreglos

- **Problema:** Escriba un programa en C que permita almacenar las notas de cada uno de los 10 estudiantes que posee un profesor. El programa además permitirá obtener el promedio de notas del curso.

Introducción a arreglos

- Para el problema anterior debemos crear 10 variables del mismo tipo (entero), una para cada estudiante:
  - `int n1,n2,n3,n4,n5,n6,n7,n8,n9,n10;`

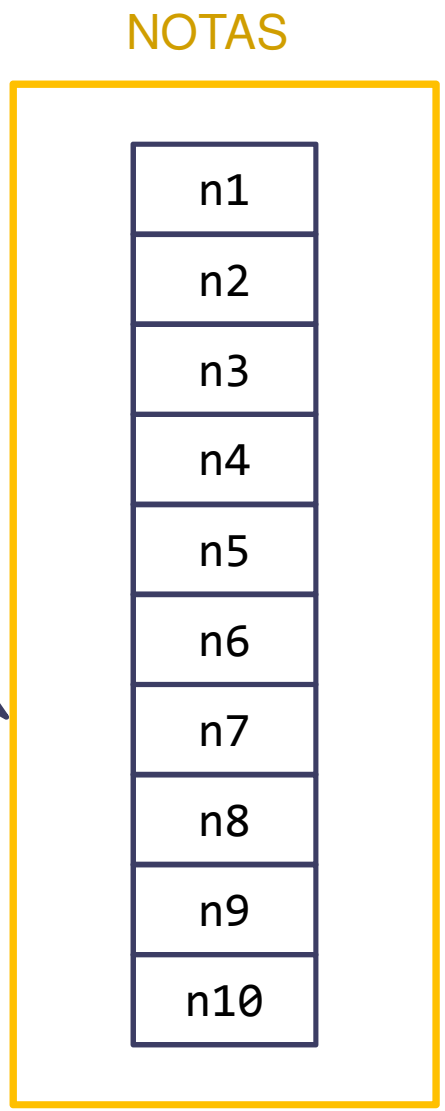
Introducción a arreglos

```
int n1,n2,n3,n4,n5,n6,n7,n8,n9,n10;
float promedio;

printf("Ingrese la nota del estudiante 1: ");
scanf("%i",&n1);
printf("Ingrese la nota del estudiante 2: ");
scanf("%i",&n2);
printf("Ingrese la nota del estudiante 3: ");
scanf("%i",&n3);
printf("Ingrese la nota del estudiante 4: ");
scanf("%i",&n4);
printf("Ingrese la nota del estudiante 5: ");
scanf("%i",&n5);
printf("Ingrese la nota del estudiante 6: ");
scanf("%i",&n6);
printf("Ingrese la nota del estudiante 7: ");
scanf("%i",&n7);
printf("Ingrese la nota del estudiante 8: ");
scanf("%i",&n8);
printf("Ingrese la nota del estudiante 9: ");
scanf("%i",&n9);
printf("Ingrese la nota del estudiante 10: ");
scanf("%i",&n10);

promedio = (float)(n1+n2+n3+n4+n5+n6+n7+n8+n9+n10)/10;
printf("Promedio del curso es %.2f",promedio);
```

Agrupar variables del mismo tipo (enteros) y que contienen la misma información lógica (notas)

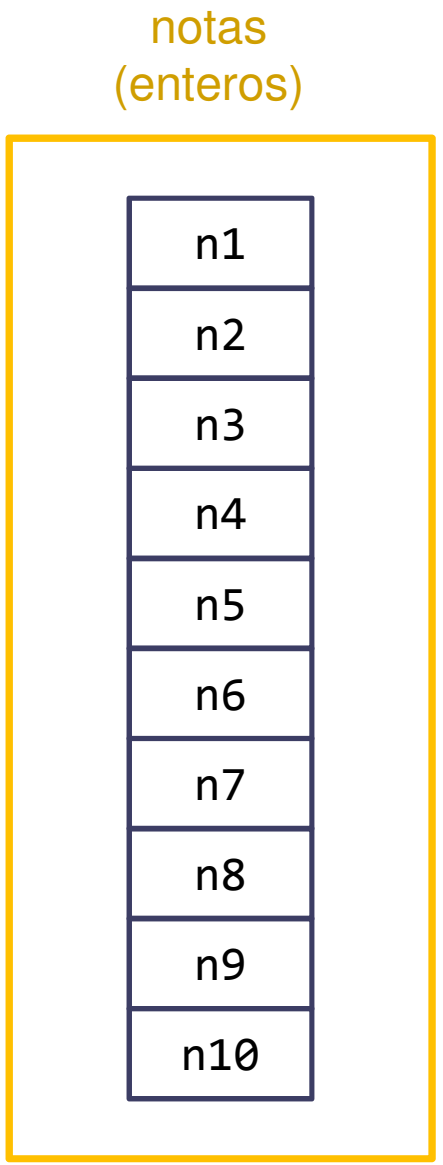


Arreglos
Definición

- Grupo consecutivo de bloques o espacios en memoria.
- Identificados con un nombre y tipo de dato.
  - Arreglo de 10 enteros denominado `notas`.

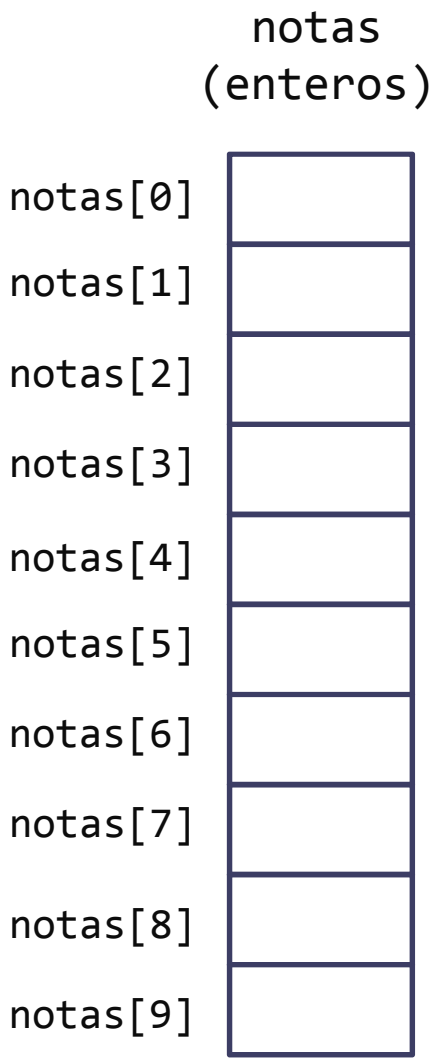
```
int notas[10];
```

```
tipo nombre[tamaño];
```



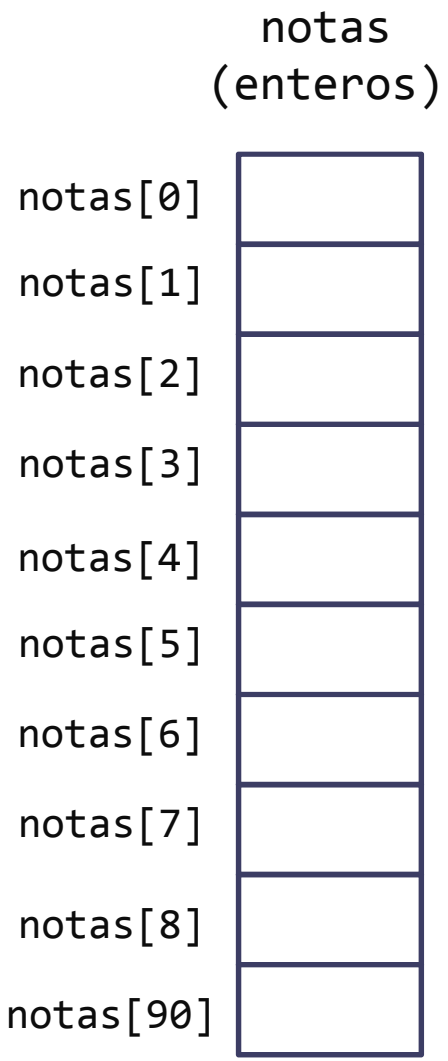
Arreglos
Elementos

- Cada elemento del arreglo es una variable como tal.
- Para referenciar un elemento (celda) del arreglo se requiere del nombre del arreglo junto a la posición (entero) a la cual se desea referenciar.
- La posición de la primera celda del arreglo es la 0. Un arreglo de N elementos comenzará desde la celda `0` hasta la celda `n-1`.



Arreglos
Elementos

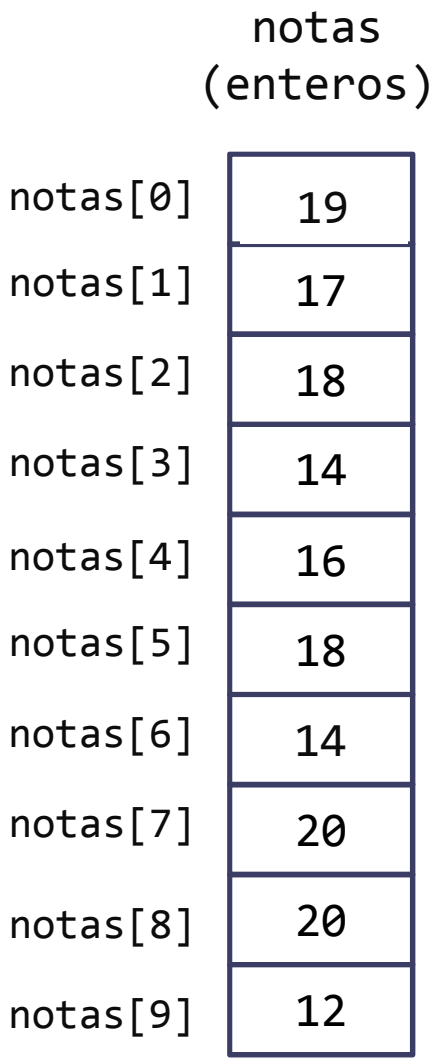
- La posición puede ser una variable entera:
  - `int pos=3;`
  - `notas[pos];`
- La posición puede ser una expresión:
  - `int a=3,b=1;`
  - `notas[a+1];`
  - `notas[a+b];`



Arreglos
Elementos

- Obtener un elemento de un arreglo:
  - `notas[0]`
  - `16`
  - `notas[6]`
  - `19`
- Cambiar el valor de un elemento de un arreglo:

```
notas[0] = 19;
notas[6] = 14;
```



Inicialización de arreglos

- Se la realiza en la declaración.
- Se utilizan llaves que delimitan el valor de cada elemento.
  - `int num[5] = {1, 2, 3, 4, 5};`

Arreglo de 5 elementos tipo entero llamado num

Inicialización de los 5 elementos

Inicialización de arreglos

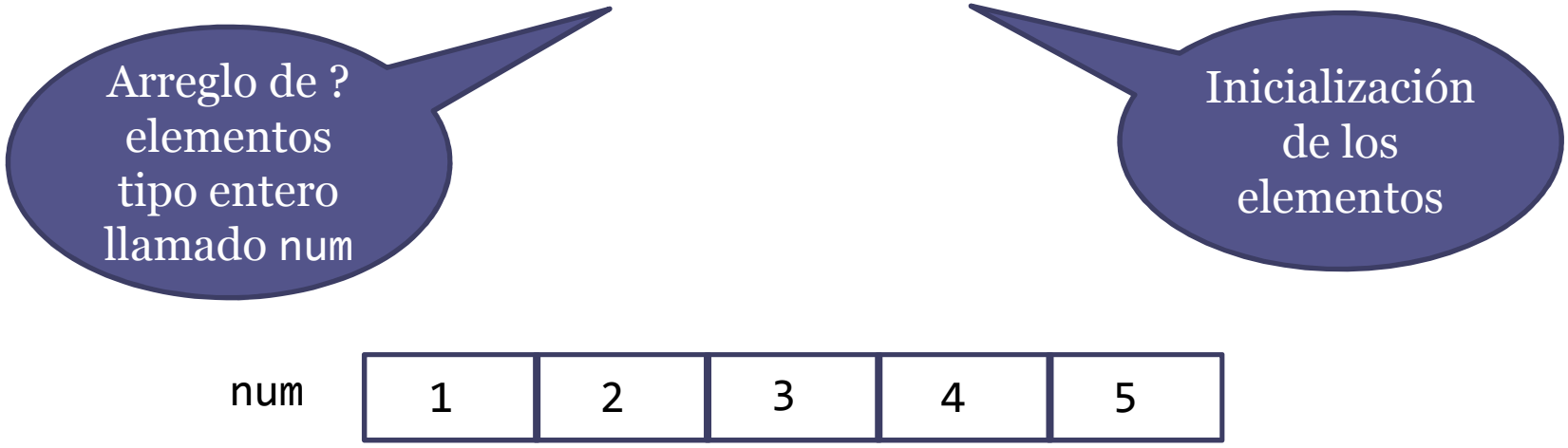
- Si no hay suficientes inicializadores, los elementos que están mas a la derecha se les asignan 0 (cero)
  - `int num[5] = {0};`

Si hay mas inicializadores que los necesitados existe un error de sintaxis, ya que no existe una revisión de los límites

```
int num[5] = {1, 2, 3, 4, 5, 6};
```

Inicialización de arreglos

- Si la dimensión de un arreglo es omitida, los inicializadores la determinarían
  - `int n[ ] = {1, 2, 3, 4, 5}`



Ejemplo

- Realicemos nuestro ejercicio utilizando arreglos.

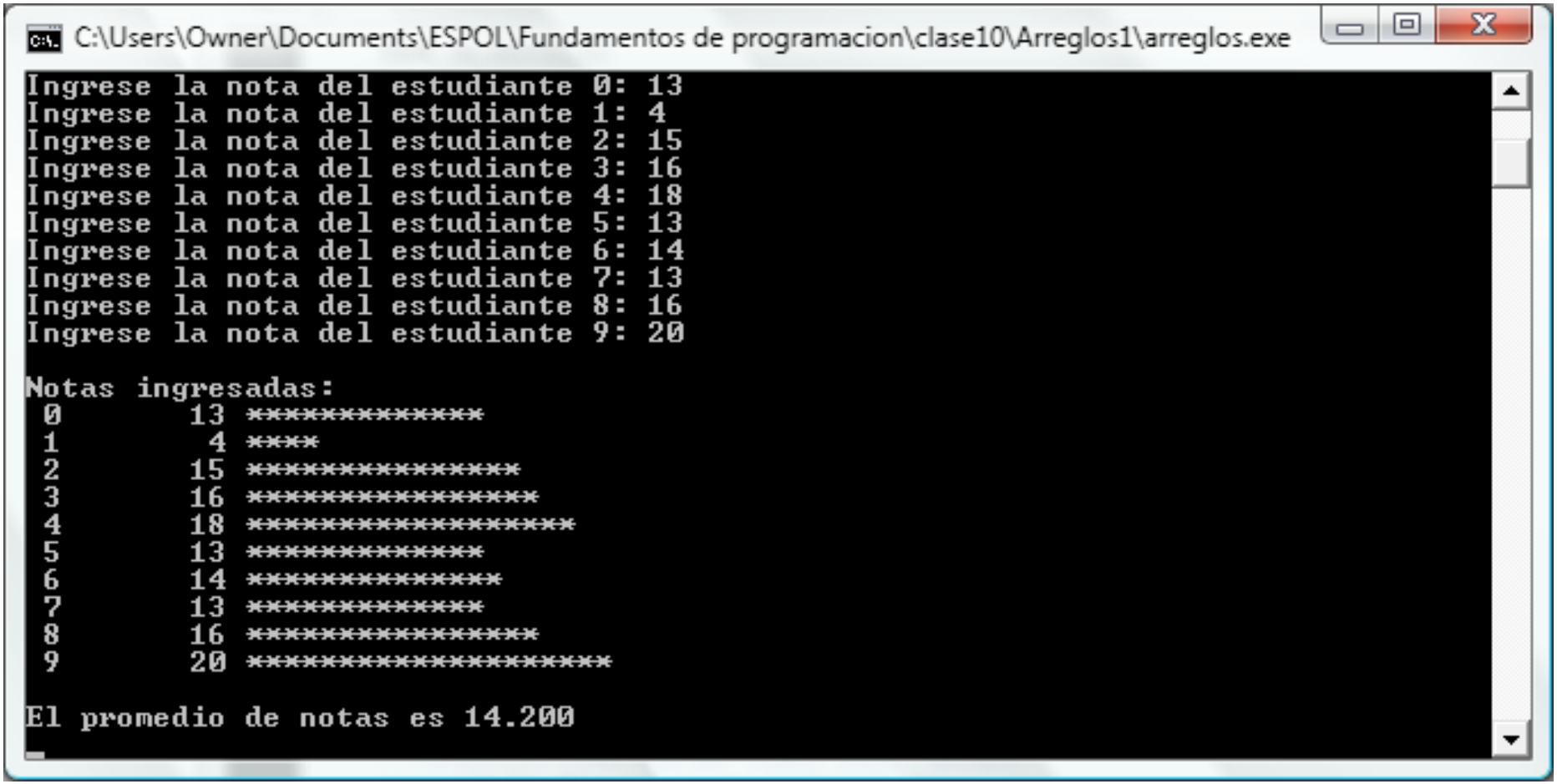
Ejemplo

- Modifiquemos nuestro programa para que el número de alumnos sea una constante simbólica.
- La directiva #define nos permite setear constantes dentro del código. Cuando utilicemos estos valores, el preprocesador los reemplazará automáticamente con el valor que se indique.

```
#define N_NOTAS 10
```

Ejemplo

- Suponga que el profesor quisiera visualizar las notas de forma gráfica como histogramas horizontales.

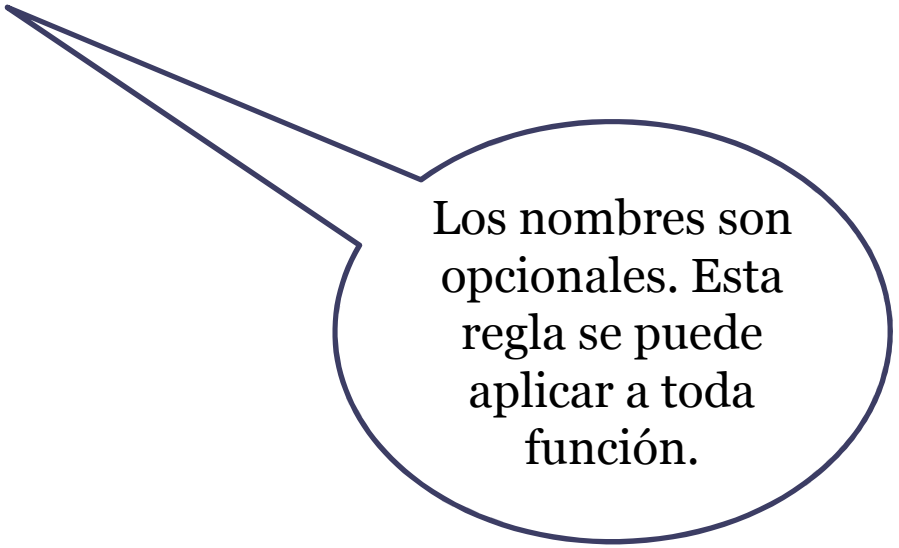


Paso de arreglos a funciones

- Especificar el nombre del arreglos sin brackets
  - `int cartas[5];`
  - `baraja(cartas,5);`
- Usualmente la dimensión del arreglo se pasa como parámetro.
- El arreglo siempre es pasado *por referencia*.
  - Cualquier cambio afecta el contenido original el arreglo.

Paso de arreglos a funciones

- Función prototipo:
  - `void baraja(int cartas[20],int tamanio);`
  - `void baraja(int [], int);`



En esta clase Ud. aprendió

- Almacenar un grupo de datos del mismo tipo en arreglos.
- Identificar los problemas que pueden ser resueltos utilizando arreglos.
- Analizar y resolver problemas que involucren la utilización de arreglos.
- Enviar arreglos a funciones.
- Diseñar funciones para la manipulación de arreglos.