

Tipos de Datos y Estructuras

Dr. Cosijopii García

4 de marzo de 2025

1.1. Tipos de Datos Simples I

- ▶ **Enteros:** int, short, long
- ▶ **Reales:** float, double
- ▶ **Caracteres:** char
- ▶ **Booleanos:** _Bool o bool (con stdbool.h)

Ejemplo en C:

```
1  #include <stdio.h>
2  #include <stdbool.h>
3
4  int main() {
5      int entero = 10;
6      float real = 3.14;
7      char caracter = 'A';
8      bool booleano = true;
9
10     printf("Entero: %d\n", entero);
11     printf("Real: %.2f\n", real);
```

1.1. Tipos de Datos Simples II

```
12     printf("Caracter: %c\n", caracter);  
13     printf("Booleano: %d\n", booleano);  
14  
15     return 0;  
16 }
```

1.2. Tipos de Datos Abstractos I

- ▶ Listas
- ▶ Pilas (Stacks)
- ▶ Colas (Queues)
- ▶ Conjuntos y diccionarios

Ejemplo en C (Implementación simple de una pila usando arreglo):

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  #define MAX 100
5
6  typedef struct {
7      int data[MAX];
8      int top;
9  } Stack;
10
```

1.2. Tipos de Datos Abstractos II

```
11 void init(Stack *s) {
12     s->top = -1;
13 }
14
15 int push(Stack *s, int value) {
16     if(s->top >= MAX - 1)
17         return -1; // Overflow
18     s->data[++s->top] = value;
19     return 0;
20 }
21
22 int pop(Stack *s, int *value) {
23     if(s->top == -1)
24         return -1; // Underflow
25     *value = s->data[s->top--];
26     return 0;
27 }
28
29 int main() {
30     Stack s;
```

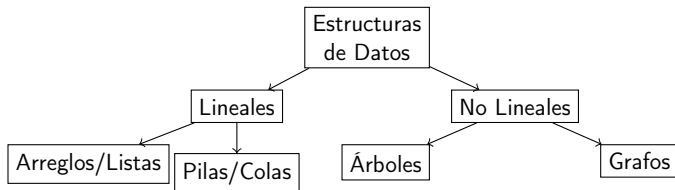
1.2. Tipos de Datos Abstractos III

```
31     init(&s);  
32     push(&s, 10);  
33     push(&s, 20);  
34     int val;  
35     pop(&s, &val);  
36     printf("Valor extra do: %d\n", val);  
37     return 0;  
38 }
```

1.3. Clasificación de las Estructuras de Datos: Lineales y No Lineales

- ▶ **Estructuras Lineales:**
 - ▶ Arreglos, listas enlazadas, pilas, colas.
- ▶ **Estructuras No Lineales:**
 - ▶ Árboles, grafos.

Diagrama (usando TikZ en caso especial):



1.3.2. Estructuras Dinámicas y Estáticas I

- ▶ **Estáticas:**

- ▶ Arreglos de tamaño fijo.

- ▶ **Dinámicas:**

- ▶ Listas enlazadas, vectores dinámicos.

Ejemplo en C (Uso de memoria dinámica para un arreglo):

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main() {
5      int n = 5;
6      int *arr = (int *)malloc(n * sizeof(int));
7      if(arr == NULL) {
8          printf("Error de asignaci n\n");
9          return 1;
10     }
11     for(int i = 0; i < n; i++) {
12         arr[i] = i * 10;
```


1.3.2. Estructuras Dinámicas y Estáticas II

```
13     }  
14     for(int i = 0; i < n; i++) {  
15         printf("arr[%d] = %d\n", i, arr[i]);  
16     }  
17     free(arr);  
18     return 0;  
19 }
```

1.4. Operaciones con Arreglos y Estructuras I

► Arreglos:

- Acceso por índice.
- Búsqueda secuencial y binaria.
- Inserción y eliminación (movimiento de elementos).

► Estructuras (listas, pilas, colas):

- Inserción y eliminación en posiciones específicas.
- Operaciones especializadas: push, pop, enqueue, dequeue.

Ejemplo en C (Inserción en un arreglo):

```
1 #include <stdio.h>
2 #define MAX 100
3
4 int main() {
5     int arr[MAX] = {1, 2, 3, 4, 5};
6     int n = 5, pos = 2, value = 99;
7     // Desplazar elementos para insertar
8     for (int i = n; i > pos; i--) {
9         arr[i] = arr[i - 1];
```

1.4. Operaciones con Arreglos y Estructuras II

```
10     }
11     arr[pos] = value;
12     n++;
13     // Imprimir arreglo
14     for (int i = 0; i < n; i++) {
15         printf("%d ", arr[i]);
16     }
17     printf("\n");
18     return 0;
19 }
```

Resumen

- ▶ Los **tipos de datos simples** incluyen enteros, reales, caracteres y booleanos.
- ▶ Los **tipos de datos abstractos** permiten definir estructuras como listas, pilas y colas.
- ▶ Las **estructuras de datos** se clasifican en:
 - ▶ **Lineales** vs. **No lineales**.
 - ▶ **Estáticas** vs. **Dinámicas**.
- ▶ Las operaciones en arreglos y estructuras se implementan de forma distinta según su naturaleza.
- ▶ Los ejemplos en C muestran aplicaciones prácticas de estos conceptos.

¡Gracias por su atención!