

## **Estructuras de Datos. Definición. Clasificación.**

### **¿Qué es una estructura de datos?**

Es un conjunto de variables relacionadas entre si y que se puede operar como un todo bajo un nombre único.

### **¿Qué estructuras de datos conoce?**

Registro, Arreglo, Lista, Arbol.

### **¿Cómo clasifica a las estructuras de datos que conoce de acuerdo a:**

**Alocación de memoria**

**Linealidad**

**Acceso**

**Los datos que la componen**

Registro: Alocación estática, lineal (los campos ocupan posiciones contiguas en memoria), Acceso directo, Heterogénea

Arreglo: Alocación estática, Lineal, Acceso directo, Homogénea

Lista: Alocación dinámica, lineal, acceso secuencial, Homogénea

Arbol: Alocación dinámica, no lineal, acceso secuencial, Homogénea

### **¿Cuándo se dice que una estructura de datos es dinámica? y ¿estática?**

Es dinámica cuando la cantidad de elementos que contiene varía durante la ejecución del programa, en cambio estática la cantidad es fija

### **¿A qué llamamos estructuras de datos homogéneas y heterogéneas?**

Es homogénea si los datos que la componen son del mismo tipo, en cambio heterogénea pueden ser de distintos tipos

### **¿A qué llamamos estructuras de datos lineales y no lineales?**

Lineal cuando está formada por ninguno, uno o más elementos que guardan una relación de adyacencia donde a cada elemento le sigue 1 y le precede uno, solamente.

Una estructura es no lineal si para un elemento dado pueden existir 0, 1 o más elementos que la suceden y/o preceden.

### **¿Qué diferentes accesos a las estructuras conoces?**

Acceso secuencial: para acceder a un elemento debo respetar un orden.

Acceso directo: puedes acceder a un elemento directamente.

## **Registro**

### **¿Cómo define el tipo de dato registro?**

Permite agrupar diferentes clases de datos en una estructura única, compuesto por campos

### **¿Cómo es el acceso a los campos?**

Directo. registro.campo

### **¿Es importante el orden de los campos? ¿Qué tipos de datos pueden contener los campos?**

No importa el orden de los campos, los campos pueden contener datos elementales o bien otras estructuras de datos

### **¿Una variable tipo puntero puede ser campo de un registro? Por qué.**

Si, la verdad no encuentro entiendo a donde apunta esta pregunta.. supongo que lo importante es destacar que el puntero es un dato simple que apunta a una dirección de memoria, pero si fuera otra estructura como un vector tampoco habría problema

### **Mencione las operaciones permitidas para el tipo de dato registro.**

Asignación, comparación, lectura/escritura.

### **Arreglos**

#### **¿Cómo defines la estructura de datos ARREGLO?**

Es una estructura de datos compuesta de elementos que se almacenan consecutivamente en la memoria que permite accederlos por medio de una variable índice que da la posición del componente dentro de la estructura de datos

#### **¿Cuáles son sus operaciones?**

- asignación de contenido a sus elementos
- Lectura/ Escritura
- Recorridos
- Cargar datos en un vector
- Agregar elementos al final
- Insertar elementos
- Borrar elementos
- Buscar un elemento

#### **¿Cómo describirías la operación de agregar, insertar y borrar un elemento de un vector?**

agregar: Si hay lugar en el vector, colocar el nuevo elemento en la posición de la dimensión lógica + 1 y aumentar la dimensión lógica

insertar: Si hay lugar en el vector, desde 'dimensión lógica' down to 'posición donde quiero insertar' -> me guardo lo que hay en la posición anterior a la actual, luego coloco el nuevo elemento en la posición deseada y aumento la dimensión lógica

borrar: Desde 'la posición que quiero borrar' hasta dimensión lógica menos 1, asigno el elemento de la posición siguiente, luego disminuyo la dimensión lógica

### **¿A qué le llamamos dimensión lógica y física del arreglo?**

Dimensión lógica: Se determina cuando se cargan los contenidos del arreglo, indica la cantidad de posiciones de memoria ocupadas con contenidos cargados desde la posición inicial

Dimensión física: Se determina al momento de la declaración y determina su ocupación de memoria. La cantidad de memoria no variará durante la ejecución del programa

### **¿De qué tipo de dato se pueden declarar los elementos y el índice de un arreglo?**

Los elementos pueden ser de cualquier tipo de datos de asignación estática (entero, real, caracter, logico, string, registro, otro arreglo) mientras que el índice debe ser de un tipo caracter, entero, subrango.

### **¿Cómo es el acceso a sus elementos?**

Los componentes pueden recuperarse en cualquier orden, indicando simplemente su posición por eso es una estructura de datos de Acceso Directo, como el acceso es a través de un índice también se la denomina indexada

### **¿Cómo se declara el tipo de dato arreglo en Pascal?**

arreglo = Array[1..dimF] of tipo

## **Listas**

### **¿Cómo defines la estructura de datos LISTA?**

Es una colección de elementos homogéneos, con una relación lineal que los vincula, es decir que cada elemento tiene un único predecesor (salvo el primero) y un único sucesor (salvo el último)

Los elementos que la componen no ocupan posiciones consecutivas, es decir que pueden aparecer dispersos en la memoria, pero mantienen un orden interno

### **¿Cuáles son sus operaciones?**

Crear una lista vacía, Agregar un elemento al principio, Agregar al final, Insertar un elemento en una lista ordenada, Recorrer, Acceder al K-ésimo elemento de la lista, Eliminar un elemento, Combinar 2 listas (merge)

### **¿Cómo se declara el tipo de dato lista en Pascal?**

```
lista = ^nodo
nodo = record
    dato : dato;
    siguiente : lista;
end;
```

**¿Qué análisis puedes hacer del uso de la memoria al trabajar con una lista?**

Si se supone igual cantidad de datos en un vector y una lista, se puede afirmar que los vectores son más económicos ya que las listas requieren espacio extra para los enlaces

Si no se conoce la cantidad que contendrá cada estructura, no se puede declarar fácilmente los vectores teniendo que reservar en memoria más espacio del que realmente se necesita, mientras que las listas solo utilizan el espacio que se necesita.