



**Nombre del alumno: Johanne Joaquín Arriaga Díaz.**

**Nombre del profesor: Icel Bernardo Lepe Arriaga.**

**Materia: Algoritmos y estructuras de datos.**

**Nombre del trabajo: Ensayo de la unidad II.**

**PASIÓN POR EDUCAR**

**Grado: Quinto cuatrimestre**

**Grupo: ISC13SDC0119-F**

Frontera Comalapa, Chiapas a 13 de Marzo de 2021

## TIPOS DE DATOS ABSTRACTOS FUNDAMENTALES

Definimos a la abstracción como hacer más pequeño un grupo de objetos datos u otras cosas haciendo pequeños grupos de estos con ciertas características o haciendo más pequeño el grupo tomando solo en cuenta lo más relevante y es un proceso cognitivo humano esencial para la comprensión de fenómenos o situaciones complejas y presenta dos caras complementarias:

1. Lo primero que se hace es relacionar las características de los datos del grupo en los cuales se va a bazar la resolución del problema y se seleccionan.
2. Y después se ignora el resto de los detalles que no son necesarios para la tarea en curso de los elementos que se abstraen.

Y ya que los datos a estudiar ahora son más pequeños podemos estudiarlos de mejor manera y que están en diferentes grupos mucho más abstraídos y en orden jerárquico de importancia, es tan importante este método en la vida humana que está presente en casi todo, que va desde el orden de una empresa hasta algo como un programa informático.

Así entramos en el tema de la abstracción funcional en programas informáticos “que surge de la simple idea de crear procedimientos y funciones e invocarlos en diferentes partes del programa mediante un nombre”. La parte que se ignora (“la información que se oculta”) en este proceso es la de cómo el procedimiento o función realiza su tarea. La parte con la que nos quedamos es la signatura o interfaz: los parámetros de entrada y salida (y sus tipos), y la descripción de la tarea que realiza, lo que quiere decir es que los procedimientos primarios se ocultan pero los procesos de interacción con el usuario se dejan.

### El TAD lista

Las listas son estructuras lineales muy flexibles, y esto permite hacerlas crecer y acortarse según se requiera, y nos hace fácil la abstracción o poder agregar más objetos a la lista insertando o suprimiendo elementos. Aunque también puede hacerse con vectores, con la excepción de que las listas son más fáciles de crear y procesar en muchos programas.

### DESCRIPCIÓN MATEMÁTICA DE LAS LISTAS

Una lista es una secuencia de elementos que van de cero o más, y son de un tipo determinado, podemos por ejemplo llamarlo “elem\_t”, por ejemplo, int que maneja enteros o double que maneja números con punto decimal. A menudo representamos una lista en forma impresa como una sucesión de elementos entre paréntesis, separados por comas. Algo importante de la lista es que está ordenada en forma lineal, es decir, para cada elemento  $a_i$  existe un sucesor  $a_{i+1}$  (si  $i < n - 1$ ) y un predecesor  $a_{i-1}$  (si  $i > 0$ ). Este orden es parte de la lista, es decir, dos listas son iguales si tienen los mismos elementos y en el mismo orden.

## OPERACIONES ABSTRACTAS SOBRE LISTAS

Una operación muy típica es eliminar todos los elementos duplicados de la misma con una regla en el código que elimine datos que se repitan haciendo que el dato a comparar incremente en 1 por ejemplo  $i+1$  y con la regla con  $j < i$ , eliminando los elementos iguales a  $i$ . Notar que no hace falta revisar los elementos anteriores a  $i$  (es decir, los elementos  $j$  con  $j < i$ ), ya que, por construcción, todos los elementos de 0 hasta  $i$  son distintos.

También está la opción de insertar elementos a la lista haciendo que el programa recorra la lista y en cierta posición deseada inserte o suprima el objeto.

### El TAD pila

Hablando de las listas algo básico es el hecho de que todas las operaciones de inserción y borrado se producen en uno de los extremos de la lista. Como cuando ocpamos muchas cosas y las apilamos una sobre otra por ende tenemos acceso a lo que esta hasta arriba y por lo tanto en el extremo y si queremos acceder a algún objeto que se encuentra más abajo debemos sacar los primeros objetos de arriba y ponerlos en algún lugar para poder acceder al mismo. La Pila es el típico ejemplo de la estructura tipo “LIFO” (por “Last In First Out”, es decir “el último en entrar es el primero en salir”).

### EL TAD COLA

Sucede lo inverso a la pila porque aquí el primero en entrar es el primero en salir y es un contenedor de tipo "FIFO" (por "First In First Out"). Un ejemplo es la cola de la caja en el supermercado. Y se usa más que nada como contenedor donde almacenar una serie de objetos que deben ser procesados y por eso se considera como buffer o pulmón, manteniendo el orden en el que ingresaron los datos.

### TIEMPO DE EJECUCIÓN

El tiempo de ejecución se basa a la cantidad de objetos en la lista o matriz de datos ya que. El lazo de un dato se ejecuta  $n - 1$  veces, y el lazo interno se ejecuta, en el peor caso  $j - 1$  veces, con lo cual el costo del algoritmo es, en el peor caso  $O(n^2)$ . En el mejor caso, el lazo interno no se ejecuta ninguna vez, de manera que sólo cuenta el lazo externo que es  $O(n)$ . En general, el tiempo de ejecución del algoritmo dependerá de cuantas posiciones deben ser desplazadas (es decir  $p - j$ ) para cada  $j$ .

### PARTICULARIDADES AL ESTAR LAS SECUENCIAS PARES E IMPARES ORDENADAS

Al estar las posiciones pares e impares ordenadas entre sí puede ocurrir que en promedio el desplazamiento sea menor, de hecho, generando vectores en forma aleatoria, pero tales que sus posiciones pares e impares estén ordenada se llega a la conclusión que el desplazamiento promedio es  $O(\sqrt{n})$ , de manera que el algoritmo resulta ser  $O(n^{3/2})$ . Esto representa una gran ventaja contra el  $O(n^2)$  del algoritmo de ordenamiento original.

Como resolución al ser un tema extenso sabemos que las listas vectores o matrices son una forma muy eficaz de depurar datos siendo una buena forma de abstracción y de inserción y hacen muy fácil la tarea de manejar datos siendo algo también complicado programarlos aunque siendo exitoso el programa nos ahorraría demasiado tiempo, entonces las listas tienen varios usos y hay varias maneras de manejarlas como lo son algoritmo de intercalación con una cola auxiliar que se basa en datos auxiliares.