



# UAI

**Universidad Abierta Interamericana**

## **EVALUACIÓN PARCIAL – RESOLUCIÓN DE CASOS**

ALUMNO:

- Di Salvio Tiago
  - 42727210
  - [tiago.disalvio@alumnos.uai.edu.ar](mailto:tiago.disalvio@alumnos.uai.edu.ar)

CARRERA: Ingeniería en Sistemas Informáticos

ASIGNATURA: Programación I

FECHA: 04/10/2021

**1. Si deseo almacenar en una variable la edad de una persona, ¿Qué tipo de dato es el más adecuado? Justifique**

La variable mas adecuada para utilizar es el tipo “byte”. Ya que esta representa a un número entero real positivo; tiene un rango de valores desde 0 hasta 255.

Por lo general se utiliza la variable “int”. Pero en este caso no la veo necesaria, porque representa a un número entero real; posee un rango de valores desde -2.147.483.648 hasta 2.147.483.647

No es posible cumplir años negativos.

**2. ¿Qué es una función recursiva?**

Las funciones recursivas son funciones que se invocan a sí mismas.

Son equivalentes a estructuras bucle o ciclo, permiten especificar muchos problemas de un modo más simple y natural que éstos, de ahí su importancia en programación.

Es una alternativa al uso de bucles. Cuando un programa invoca a una función que llama a otra, y así sucesivamente, las variables y valores de los parámetros de cada llamada a cada función se guardan en la pila o stack, junto con la dirección de la siguiente línea de código a ejecutar.

**3. Enumere los tipos de listas que existen y explique qué caracteriza a cada una**

**I. Listas simples enlazadas**

Estas poseen un enlace por nodo. Dicho enlace apunta al siguiente nodo (o indica que tiene la dirección en memoria del siguiente nodo) en la lista, o al valor NULL o a la lista vacía, si es el último nodo.

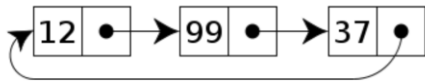
**II. Listas doblemente enlazadas**

Cada nodo tiene dos enlaces: uno apunta al nodo anterior, o apunta al valor NULL si es el primer nodo; y otro que apunta al nodo siguiente, o apunta al valor NULL si es el último nodo.

En algún lenguaje de muy bajo nivel, XOR-Linking (tipo de estructura de datos) ofrece una vía para implementar listas doblemente enlazadas, usando una sola palabra para ambos enlaces, esta técnica no se suele implementar.

### III. Listas enlazadas circulares

En este tipo de lista, el primer y el último nodo están unidos. Esto se puede hacer tanto para listas enlazadas simples como para las doblemente enlazadas. Para recorrer una lista enlazada circular podemos empezar por cualquier nodo y seguir la lista en cualquier dirección hasta que se regrese hasta el nodo original. Desde otro punto de vista, las listas enlazadas circulares pueden ser vistas como listas sin comienzo ni fin. Este tipo de listas es el más usado para dirigir buffers para “ingerir” datos, y para visitar todos los nodos de una lista a partir de uno dado.



*Una lista enlazada circular que contiene tres valores enteros*

### IV. Listas enlazadas simples circulares

Cada nodo tiene un enlace, similar al de las listas enlazadas simples, excepto que el siguiente nodo del último apunta al primero. Como en una lista enlazada simple, los nuevos nodos pueden ser solo eficientemente insertados después de uno que ya tengamos referenciado. Por esta razón, es usual quedarse con una referencia solamente al último elemento en una lista enlazada circular simple, esto nos permite rápidas inserciones al principio, y también permite accesos al primer nodo desde el puntero del último nodo.

### V. Listas enlazadas doblemente circulares

Cada nodo posee dos enlaces, similares a los de la lista doblemente enlazada, excepto que el enlace anterior del primer nodo apunta al último y el enlace siguiente del último nodo, apunta al primero.

Como en una lista doblemente enlazada, las inserciones y eliminaciones pueden ser hechas desde cualquier punto con acceso a algún nodo cercano. Aunque estructuralmente una lista circular doblemente enlazada no tiene ni principio ni fin, un puntero de acceso externo puede establecer el nodo apuntado que está en la cabeza o al nodo cola, y así mantener el orden tan bien como en una lista doblemente enlazada.

#### 4. ¿Qué significa TDA? Ejemplifique.

Un Tipo de Dato Abstracto (TDA) es un conjunto de datos u objetos al cual se le asocian operaciones. Puede ser implementado utilizando distintas estructuras de datos y proveer la misma funcionalidad, una abstracción es la simplificación de un objeto o de un proceso de la realidad en la que sólo se consideran los aspectos más relevantes.

La abstracción se utiliza por los programadores para dar sencillez de expresión al algoritmo.

TDA = valores + operaciones

Ejemplo de TDA y Circulo

