

UT3_PD1

| | |
|------------|-----------------|
| ▼ Property | COMPLETED |
| 📅 Date | @April 30, 2022 |
| ☰ BLOCKED | Subir dibujos |

Ejercicio 1:

- a

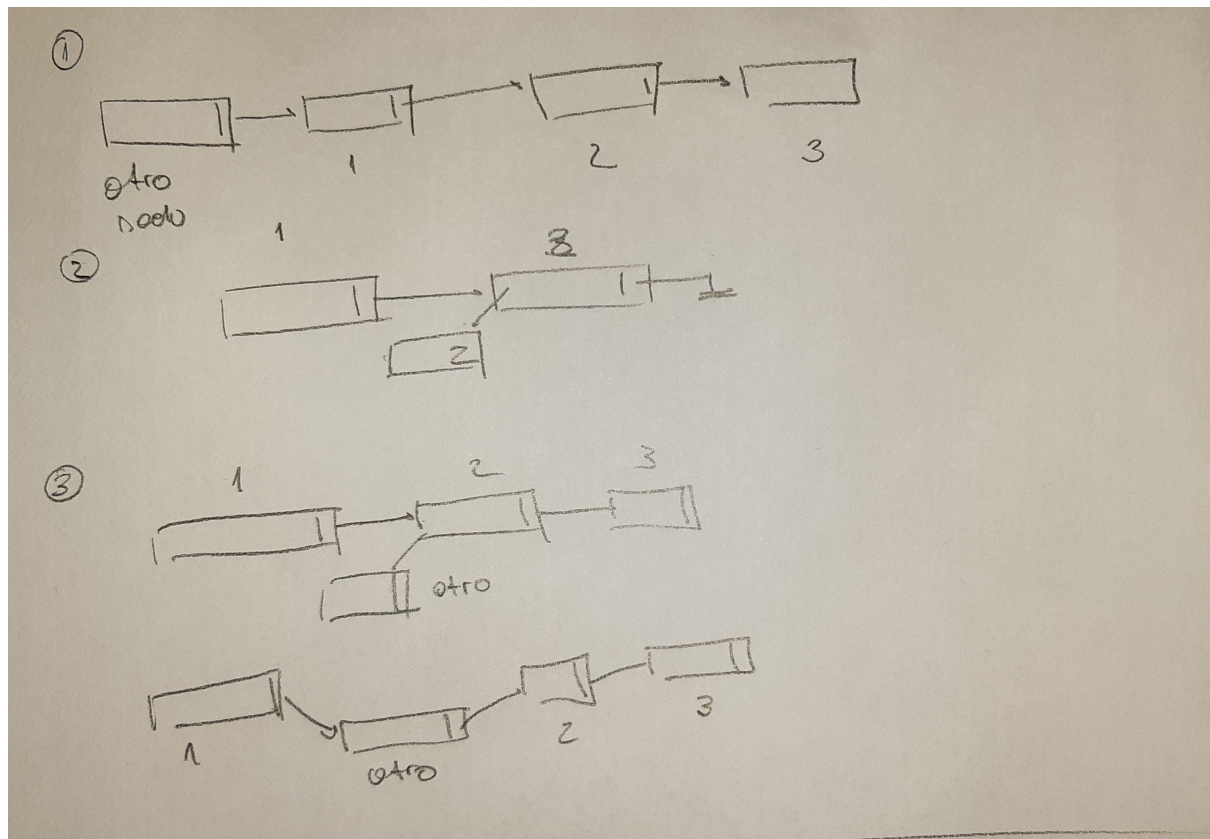
Ejercicio 2:

- c (Porque el nodo2 deja de tener referencia entonces queda como candidato a ser eliminado por el garbage collector).

Ejercicio 3:

- b

Dibujos 1,2,3:



Ejercicio 4:

- d

Se cae a pedazos porque en un momento actual = nulo (cuando sale del mientras) y no se le puede hacer un nulo.siguiente

Ejercicio 5:

- a

Ejercicio 6:

Clase Alumno —> Una clase con dos atributos: Identificador (char[] de 8 digitos es decir 16 bytes), contadorClases (int 4 bytes)

Objeto alumno 16 +4 (espacio de memoria de cada elemento) es decir 20 bytes

- CASO LISTA ENLAZADA:

La lista enlazada tiene espacio "ilimitado", es decir que le agregamos un nodo al final y listo

Espacio de memoria, cada nodo contiene un alumno

es decir 24 bytes (+4 x el espacio de referencia del objeto).

- CASO ARRAY:

Solo el objeto que son 20 bytes x alumno

PROBLEMA: El array debera redimensionarse, para esto, cuando se llene el array se debera crear un nuevo array

En el caso de saber la cantidad especifica es preferible un Array, pues es mas eficiente en memoria, pero si no se supiese deberiamos redimensionar, y esto implica crear un nuevo array de suficiente tamaño para albergar a todos los alumnos, mientras exista el otro array lleno. Muy poco eficiente

Por ejemplo si tenemos 10 alumnos la diferencia es de 40 bytes, en 100 400 bytes y en 10 000 40 000 bytes. Siempre que se sepa es preferible el Array.