



## Taller 2 Estructuras de datos

# FUNDAMENTOS DE ANÁLISIS Y DISEÑO DE ALGORITMOS

Carlos Andres Delgado S, Ing \*

Mayo 2017

Este taller se puede trabajar en grupos de hasta 3 personas. Se debe entregar el código fuente de su solución y un breve informe en formato PDF que contenga:

- Breve explicación de como soluciona el problema.
- Como se ingresan entradas a su algoritmo (lectura archivo, digitación entradas, etc).
- Que formato tiene la salida (impresión en pantalla, archivo).
- Presentación salida (Cómo se interpreta su salida, es decir como se puede entender).
- Ejemplo de ejecución de su algoritmo.

**Importante:** En este taller se propone **Java** como lenguaje de trabajo para la solución, pero usted es libre de usar cualquier otro lenguaje de su preferencia siempre y cuando maneje Objetos y Clases.

### 1. Combinando listas ordenadas

[25 puntos]

Desarrolle la operación **combinar(Lista L1, Lista L2)** que recibe dos listas simplemente enlazadas,

\* carlos.andres.delgado@correounivalle.edu.co

cada una ordenada ascendentemente, y devuelve una lista L3 que tiene los elementos ordenados de ambas listas. Las listas pueden tener elementos repetidos. A continuación se muestra un ejemplo.

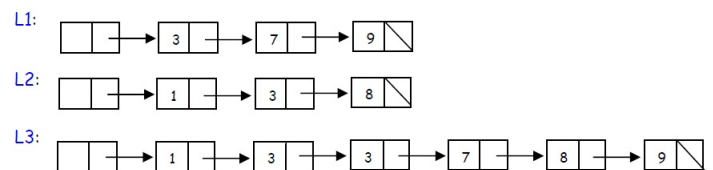


Figura 1: Ejemplo combinando lista enlazadas

### 2. El inventario [30 puntos]

En un museo se guarda la información del inventario de sus obras utilizando una lista simplemente enlazadas en la que el campo dato es un objeto de la clase Obra. Una Obra en el inventario tiene dos atributos **nombre y cantidad**.

- Desarrolle la operación **agregarReplika(String nombre)**, de tal forma que si no existe la obra con el nombre especificado, se crea en el inventario con cantidad 1. Si ya

existe, se aumenta en 1 su cantidad. La lista debe estar ordenada de acuerdo el nombre.

- Desarrolle la operación **venderReplica(String nombre)**, de tal forma que si no hay obras disponibles con el nombre especificado se indique con un mensaje. En caso contrario, se disminuye en 1 su cantidad. Si una obra llega a la cantidad 0, se elimina el nodo de la lista.
- Desarrolle la operación **ListarReplicas()** que muestra para cada obra, su nombre y la cantidad disponible

Suponga que se tiene la siguiente lista:

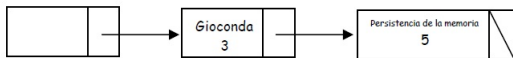


Figura 2: Lista enlazada de ejemplo

En la siguiente figura si se utiliza la operación **agregarReplica("El hombre de Vitrubio")**, la lista quedará de la siguiente manera:

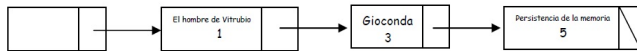


Figura 3: Lista enlazada con operación agregar réplica

Si después se utiliza la operación **venderReplica("Gioconda")**, la lista debe tener el siguiente estado:

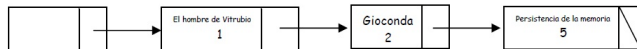


Figura 4: Lista enlazada con operación vender réplica

### 3. Implementando una pila con colas [20 puntos]

Utilizando la estructura Cola, desarrolle las siguientes operaciones básicas de una Pila:

- public boolean **estaVaciaPilaConColas()**

- public void **pushPilaConColas(Object o)**

- public Object **popPilaConColas()**

- public void **mostrarPilaConColas()**

### 4. Dominó modificado [25 puntos]

Se requiere una aplicación para el juego dominó modificado, en donde sólo existen para cada posible par de números únicos desde el 1 hasta el 9 (Ejemplo: 1 + 2, 1 + 3, 1 + 4, 1 + 5, 1 + 6, 1 + 7, 1 + 8, 1 + 9, 2 + 3, 2 + 4, 2 + 5, ...). Note que no existen las piezas simétricas 1 + 2 y 2 + 1 en el conjunto, ya que cada una de estas piezas puede convertirse en la otra rotándola. Implemente una estructura de datos que permita:

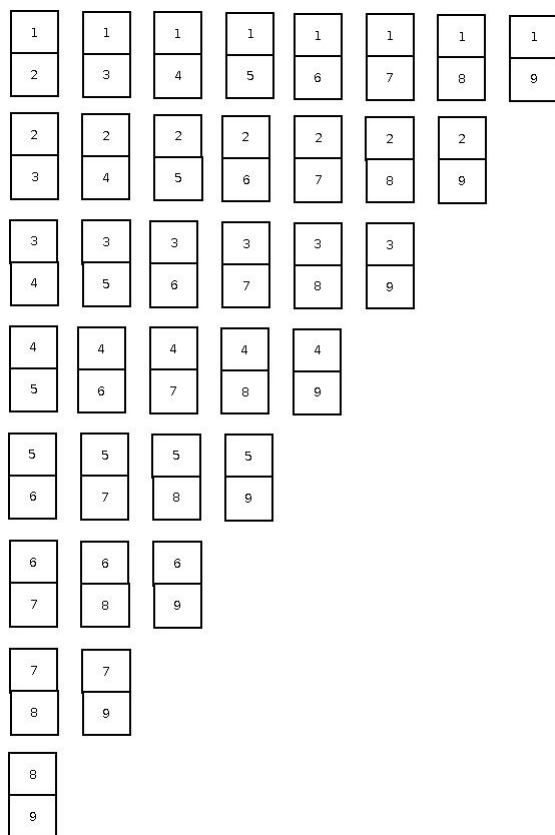
- Crear inicial el conjunto de 36 fichas de dominó. Cada ficha tiene dos estados, el primero que indica si se encuentra utilizada o disponible y el segundo su rotación hacia la derecha (0°, 90°, 180°, 270°). Se asumen que la forma mostrada en la figura 5 están a 0° de rotación, para ver las rotaciones de una ficha se puede consultar la figura 6.

- Extraer una ficha, marcando esta como utilizada.

- Ingresar una ficha y en el caso que este marcada como utilizada, se marca como disponible.

- Un método para buscar una ficha a partir de sus valores  $A$ ,  $B$  e imprimir si esta está disponible o no y su rotación. Por ejemplo para buscar la ficha 1, 2 los valores son  $A = 1$ ,  $B = 2$ .

En la siguiente figura se muestra el conjunto de fichas disponibles:



Justifique en términos de operaciones: inserción y borrado porque su estructura de datos es apropiada para solucionar este problema.

Figura 5: Fichas de dominó

La rotación de una ficha se puede ver de la siguiente manera:

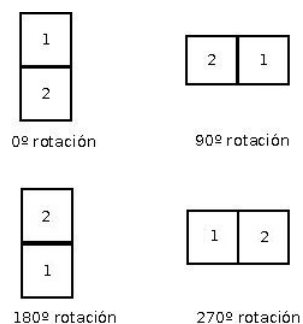


Figura 6: Rotaciones de una ficha de dominó