13-9-2021

Tipo de dato abstracto (TDA)

Tarea: 2

Materia: Estructura de datos

Sección: D01.

Código: 216584703

Carrera: Ingeniería en computación.

Nombre alumno: Padilla Pérez Jorge Daray

Nombre profesor: Julio Esteban Valdes Lopez

Índice general

¿Qué es un TDA?

Tipo de dato abstracto (en adelante *TDA*) es un conjunto de datos u objetos al cual se le asocian *operaciones*. El TDA provee de una interfaz con la cual es posible realizar las operaciones permitidas, abstrayéndose de la manera en como estén implementadas dichas operaciones. Esto quiere decir que un mismo TDA puede ser implementado utilizando distintas estructuras de datos y proveer la misma funcionalidad.

El paradigma de orientación a objetos permite el *encapsulamiento* de los datos y las operaciones mediante la definición de *clases* e *interfaces*, lo cual permite *ocultar* la manera en cómo ha sido implementado el TDA y solo permite el acceso a los datos a través de las operaciones provistas por la interfaz.

TDA lista

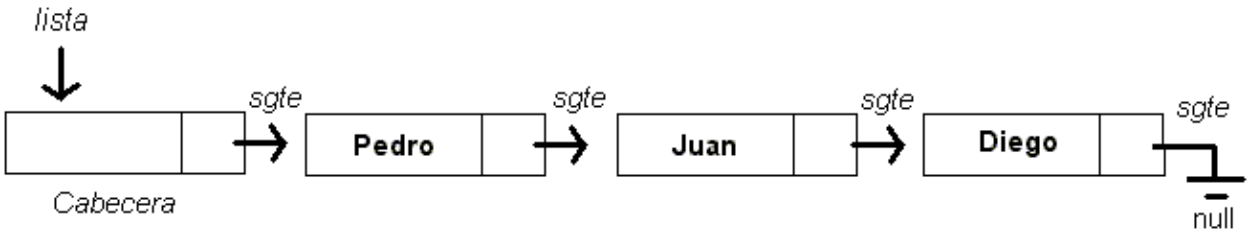
Una *lista* se define como una serie de *N* elementos E_1, E_2, ..., E_N, ordenados de manera consecutiva, es decir, el elemento E_k (que se denomina *elemento k-ésimo*) es previo al elemento E_{k+1}. Si la lista contiene 0 elementos se denomina como *lista vacía*.

Las operaciones que se pueden realizar en la lista son: insertar un elemento en la posición *k*, borrar el k-ésimo elemento, buscar un elemento dentro de la lista y preguntar si la lista esta vacía.

Una manera simple de implementar una lista es utilizando un arreglo. Sin embargo, las operaciones de inserción y borrado de elementos en arreglos son ineficientes, puesto que para insertar un elemento en la parte media del arreglo es necesario mover todos los elementos que se encuentren delante de él, para hacer espacio, y al borrar un elemento es necesario mover todos los elementos para ocupar el espacio desocupado. Una implementación más eficiente del TDA se logra utilizando listas enlazadas.

A continuación se presenta una implementación en Java del TDA utilizando listas enlazadas y sus operaciones asociadas:

- estaVacia(): devuelve *verdadero* si la lista esta vacía, falso en caso contrario.
- insertar(x, k): inserta el elemento *x* en la k-ésima posición de la lista.
- buscar(x): devuelve la posición en la lista del elemento *x*.
- buscarK(k): devuelve el k-ésimo elemento de la lista.
- eliminar(x): elimina de la lista el elemento *x*.



En la implementación con listas enlazadas es necesario tener en cuenta algunos detalles importantes: si solamente se dispone de la referencia al primer elemento, el añadir o remover en la primera posición es un caso especial, puesto que la referencia a la lista enlazada debe modificarse según la operación realizada. Además, para eliminar un elemento en particular es necesario conocer el elemento que lo antecede, y en este caso, ¿qué pasa con el primer elemento, que no tiene un predecesor?

Para solucionar estos inconvenientes se utiliza la implementación de lista enlazada con nodo cabecera. Con esto, todos los elementos de la lista tendrán un elemento previo, puesto que el previo del primer elemento es la cabecera. Una lista vacía corresponde, en este caso, a una cabecera cuya referencia siguiente es *null*.

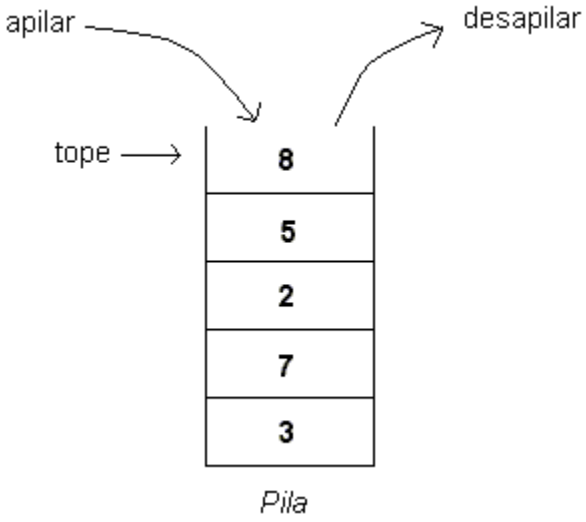
TDA pila

Una *pila* (*stack* o *pushdown* en inglés) es una lista de elementos de la cual sólo se puede extraer el último elemento insertado. La posición en donde se encuentra dicho elemento se denomina *tope* de la pila. También se conoce a las pilas como *listas LIFO* (LAST IN - FIRST OUT: el último que entra es el primero que sale).

La interfaz de este TDA provee las siguientes operaciones:

- **apilar(x)**: inserta el elemento *x* en el tope de la pila (**push** en inglés).
- **desapilar()**: retorna el elemento que se encuentre en el tope de la pila y lo elimina de ésta (**pop** en inglés).
- **tope()**: retorna el elemento que se encuentre en el tope de la pila, pero sin eliminarlo de ésta (**top** en inglés).
- **estaVacia()**: retorna *verdadero* si la pila no contiene elementos, *falso* en caso contrario (**isEmpty** en inglés).

Nota: algunos autores definen **desapilar** como sacar el elemento del tope de la pila sin retornarlo.



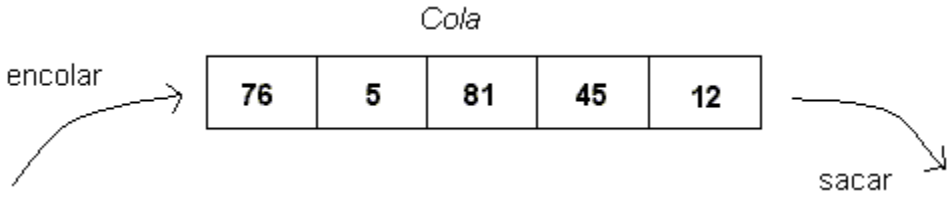
TDA cola

Una *cola* (*queue* en inglés) es una lista de elementos en donde siempre se insertan nuevos elementos al final de la lista y se extraen elementos desde el inicio de la lista. También se conoce a las colas como *listas FIFO* (FIRST IN - FIRST OUT: el primero que entra es el primero que sale).

Las operaciones básicas en una cola son:

- **encolar(x)**: inserta el elemento *x* al final de la cola (**enqueue** en inglés).
- **sacar()**: retorna el elemento que se ubica al inicio de la cola (**dequeue** en inglés).
- **estaVacia()**: retorna *verdadero* si la cola esta vacía, *falso* en caso contrario.

Al igual que con el TDA pila, una cola se puede implementar tanto con arreglos como con listas enlazadas. A continuación se verá la implementación usando un arreglo.



Referencias bibliográficas

N.A. (2012). Estructuras c++. 13 de septiembre de 2021, de uv Sitio web: Lenguaje de Programación: C++ Estructuras (cimat.mx)

N.A. (2017). Tipos de datos abstractos. 13 septiembre 2021, de dcc Sitio web: https://users.dcc.uchile.cl/~bebustos/apuntes/cc30a/TDA/