

Estructura de Datos

Tercer encuentro

Contenido

- Pilas.
- Operaciones con pilas.
 - Apilar, Cima y Desapilar.
- Colas.
- El: Operaciones con colas.
 - Añadir, Primero y Avanzar.
- Colas con prioridad.

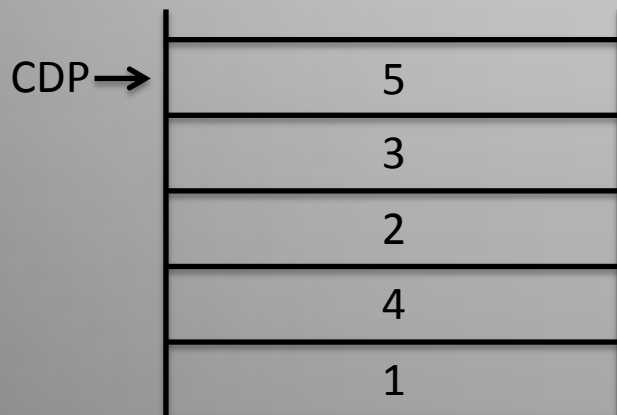
TDA Pila

Una pila es un conjunto dinámico que obedece a la propiedad LIFO. Es decir los elementos de una pila se van añadiendo uno sobre otro de forma tal, que a la hora de desapilarlos el último que se puso es el primero en salir.

LIFO es el acrónimo correspondiente a, ***last-in-first-out***, cuya traducción literal sería: último en entrar, primero en salir.

TDA Pila

Todas las estructuras de datos se basan en una lista enlazada y sus operaciones se construyen a partir de las operaciones definidas para una lista. Gráficamente una pila representa un conjunto de elementos apilados uno sobre el otro. Por ejemplo una Pila de números enteros.



Cada elemento de la pila representa un nodo de la lista.

CDP (cima de Pila): Apunta al elemento insertado más recientemente.

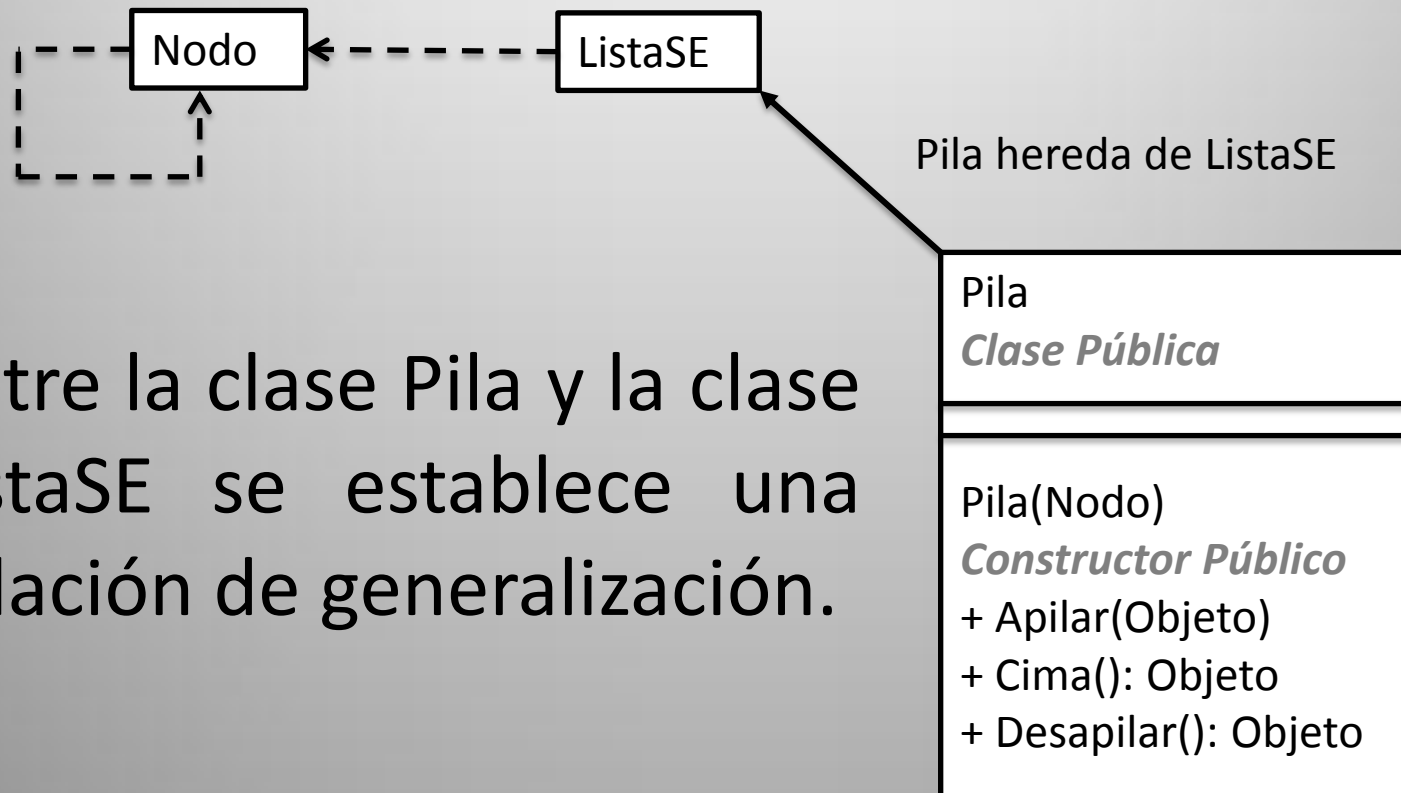
Operaciones con Pilas

En las pilas se definen tres operaciones básicas:

- **Apilar (S, x)**: Inserta x en S (siempre al inicio).
- **Cima(S)**: Devuelve el elemento que fue insertado más recientemente en S.
- **Desapilar(S)**: Elimina el elemento que fue insertado más recientemente en S y lo devuelve.

La nueva clase Pila es heredera de la clase ListaSE.

Jerarquía



Entre la clase Pila y la clase ListaSE se establece una relación de generalización.

Clase Pila

Constructor Pila(cabeza: Nodo)

inicio

// Llamar al constructor de la clase base
base(cabeza)

fin

Apilar(dato: Objeto)

inicio

este.Insertar(dato, 1)

fin

Cima(): Objeto

inicio

este.Inicio()
devolver este.Actual()

fin

Desapilar(): Objeto

var

temp: Objeto

inicio

temp ← este.Cima()
este.Eliminar(1)
devolver temp

fin

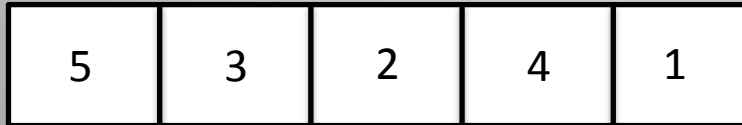
TDA Cola

Una cola es un conjunto dinámico que obedece a la propiedad FIFO. Es decir los elementos de una cola se van añadiendo uno a la derecha del otro de forma tal, que a la hora de avanzar el primero que se puso es el primero en salir.

FIFO es el acrónimo correspondiente a ***first-in-first-out***, cuya traducción literal sería: último en entrar, primero en salir.

TDA Cola

Gráficamente una cola representa un conjunto de elementos uno a la derecha del otro. Donde el elemento que mas tiempo lleva en la cola es el de la izquierda. Por ejemplo una Pila de números enteros.



Cada elemento de la cola representa un nodo de la lista.

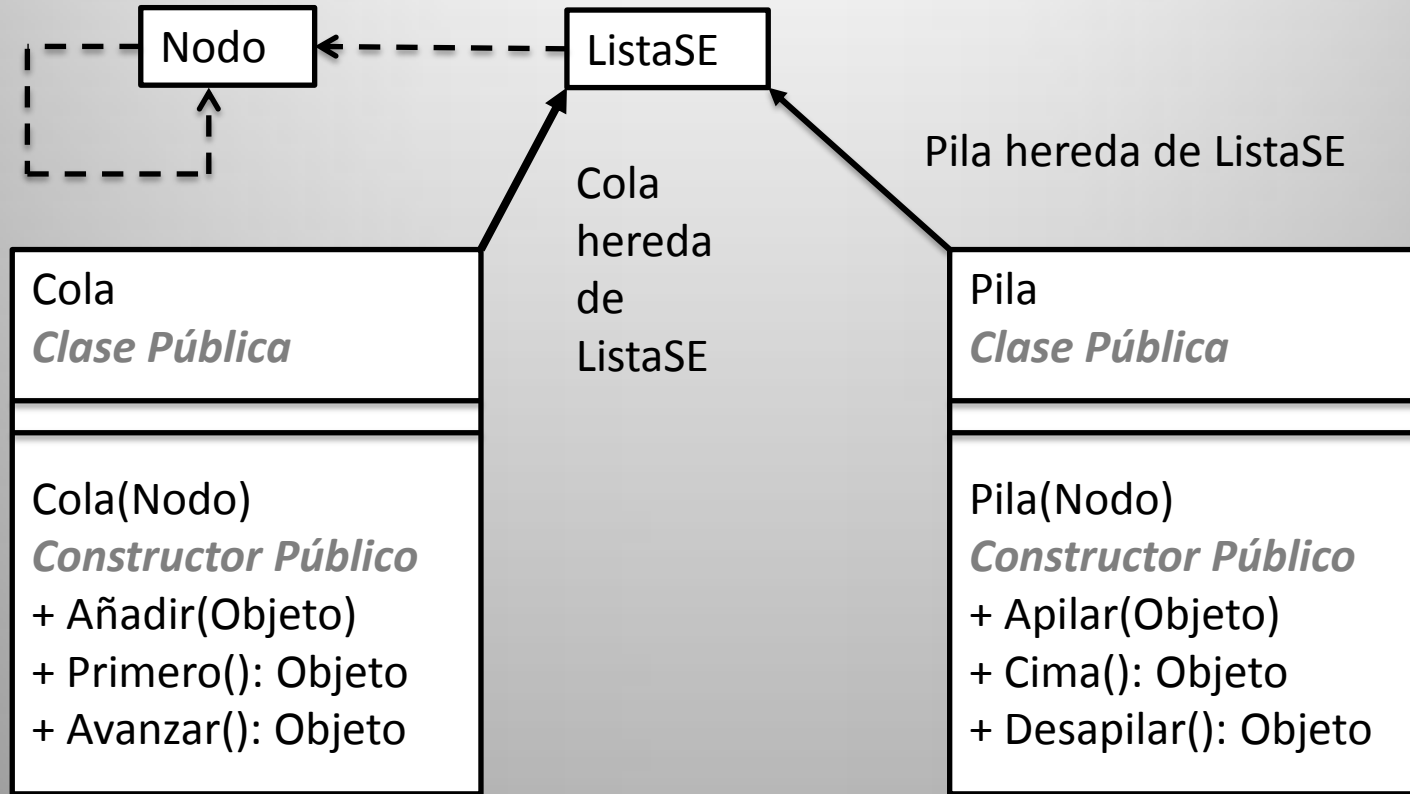
Operaciones con Colas

En las colas se definen tres operaciones básicas:

- **Añadir(Q, x)**: Inserta x en S (siempre al final).
- **Primero(Q)**: Devuelve el elemento que lleve más tiempo en Q.
- **Avanzar(Q)**: Elimina el elemento que lleve más tiempo en Q y lo devuelve.

La nueva clase Cola es heredera de la clase ListaSE.

Jerarquía



Entre la clase Cola y la clase ListaSE se establece una relación de generalización.

Colas con prioridad

A menudo es útil pensar en los valores de clave asociados con los elementos de conjuntos dinámicos como prioridades. En este caso, dado que las claves obedecen una relación de orden total, podemos identificar aquellos elementos con las prioridades más altas (más bajas).

Ejemplo: Una cola de la bodega, dónde las embarazadas tienen prioridad.

Colas con prioridad (Ejemplo)

Planificación de trabajos (procesos) en un sistema informático multiusuario. Es simple utilizar una cola para atender los procesos, con la filosofía, primero en llegar, primero en ser atendido. Sin embargo cuando trabajos cortos quedan «atrapados» en la cola detrás de trabajos largos en ejecución. Generalmente es mejor dejar que los trabajos cortos se ejecuten primero. Si asignamos prioridades a los trabajos basándonos en la cantidad de tiempo de procesador que requieren, entonces puede emplearse una cola con prioridad para planificar los trabajos de tal forma que el trabajo pendiente que requiera menor cantidad de tiempo se ejecutará siempre a continuación.

Colas con prioridad (Operaciones)

- **Insertar(P, x)**: Añade el elemento x a P.
- **EncontrarMin(P)**: Devuelve el elemento de P con la prioridad más alta (menor valor de clave). Si P está vacía, produce un error.
- **EliminarMin(P)**: Quita y devuelve el elemento de P con la prioridad más alta (menor valor de clave). Si P está vacía, produce un error.

Estas operaciones especifican en realidad lo que se denomina el ***TAD Cola de prioridad min***, cuyo complemento obvio es el ***TAD Cola de prioridad max***.

Estudio independiente

Definir algoritmos para las operaciones de la clase Cola.