

TDA Lista

Nombre: Lista

Invariantes:

- primero: puntero al primer nodo de la lista.
- último: puntero al último nodo de la lista.
- cursor: puntero al nodo actual en la lista.
- cantidad: cantidad de elementos en la lista.
- MAX_SIZE: entero que indica la cantidad máxima de la lista.

Operaciones:

- Lista: -> Lista
- ~Lista
- alta: Dato x int ->
- baja: int ->
- esta_vacia: -> bool
- consulta: int -> Dato
- obtener_cantidad: -> int
- hay_siguiente: -> bool
- siguiente: -> Dato
- reiniciar_cursor: ->
- mas_antiguo: -> dato
- promedio: -> double (numero)

Detalle de las operaciones:

- Constructor
 - Descripción: Crea una nueva lista vacía.
 - Postcondiciones: Se crea una nueva lista vacía con primero y cursor apuntando a nullptr y cantidad igual a 0.
- Destructor
 - Descripción: Libera los recursos asignados a la lista.
 - Postcondiciones: Libera la memoria asignada a todos los nodos de la lista.
- Alta
 - Descripción: Se agrega un elemento en la última posición de la lista circular. El nuevo elemento se convierte en el último de la lista y su siguiente apunta

al primer elemento de la lista, actualizando así la estructura de lista circular. La cantidad de elementos en la lista aumenta en uno.

- Precondiciones: Se le pasa un dato d, que en este caso será entero..
- Postcondiciones: Se agrega el dato d a la lista en ultima posicion.

- Baja

- Descripción: Elimina el elemento en la posición especificada de la lista.
- Precondiciones: $0 < pos \leq obtener_cantidad()$.
- Postcondiciones: Se elimina el elemento en la posición pos de la lista.

- Está vacía

- Descripción: Verifica si la lista está vacía.
- Postcondiciones: Devuelve true si la lista está vacía, false en caso contrario.

- Consulta

- Descripción: Devuelve el dato en la posición especificada de la lista.
- Precondiciones: $0 < pos \leq obtener_cantidad()$.
- Postcondiciones: Devuelve el dato en la posición pos de la lista.

- Obtener cantidad

- Descripción: Devuelve la cantidad de elementos en la lista.
- Postcondiciones: Devuelve la cantidad de elementos en la lista.

- Hay siguiente

- Descripción: Verifica si hay un siguiente elemento en la lista.
- Postcondiciones: Devuelve true si hay un siguiente elemento, false en caso contrario.

- Siguiente

- Descripción: Devuelve el siguiente dato en la lista y actualiza el cursor.
- Precondiciones: `hay_siguiente() == true`.
- Postcondiciones: Devuelve el siguiente dato en la lista y actualiza el cursor.

- Reiniciar cursor

- Descripción: Reinicia el cursor al inicio de la lista.
- Postcondiciones: El cursor se coloca al inicio de la lista.

- Más antiguo

- Descripción: Esta función devuelve el dato almacenado en el primer nodo de la lista.

- Precondición: La lista no debe estar vacía.
- Postcondición: Devuelve el dato almacenado en el primer nodo de la lista.
- Promedio
 - Descripción: Esta función calcula el promedio de los datos almacenados en todos los nodos de la lista.
 - Precondición: La lista no debe estar vacía.
 - Postcondición: Devuelve el promedio de los datos almacenados en todos los nodos de la lista.

TDA Dato

Nombre:

Dato

Invariantes:

- dato: Tipo Dato
- siguiente: Puntero a Nodo

Operaciones:

- Dato: -> Dato
- ~Dato
- obtener_dato: -> Dato
- obtener_siguiente: -> Nodo*
- cambiar_dato: Dato ->
- cambiar_siguiente: Nodo* ->

Detalle de las operaciones:

- Constructor
 - Descripción: Crea un nuevo objeto Dato.
 - Postcondiciones: Se crea un nuevo objeto Dato.
- Destructor
 - Descripción: Libera los recursos asignados al objeto Dato.
 - Postcondiciones: Libera la memoria asignada al objeto Dato.
- Obtener dato
 - Descripción: Devuelve el dato almacenado en el nodo.
 - Postcondiciones: Devuelve el dato almacenado en el nodo.

- Obtener siguiente
 - Descripción: Devuelve un puntero al siguiente nodo en la lista.
 - Postcondiciones: Devuelve un puntero al siguiente nodo en la lista, o nullptr si el nodo actual es el último en la lista.
- Cambiar dato
 - Descripción: Actualiza el dato almacenado en el nodo.
 - Postcondiciones: Actualiza el dato almacenado en el nodo con el valor proporcionado.
- Cambiar siguiente
 - Descripción: Establece el siguiente nodo en la secuencia.
 - Postcondiciones: Establece el siguiente nodo en la secuencia como el nodo pasado como argumento.