

Nombre: _____ Rut: _____

Estructuras de datos Control 2

Prof: Fabián Riquelme Csori
2017-II

1. Además de las FIFO (first-in, first-out) y las LIFO (last-in, first-out) existen las FINO (first-in, never-out). Para definir el correspondiente TDA, determine:
 - a) Cuatro operaciones para trabajar sobre él. [12 pts]
 - b) Cuatro axiomas que regulen el funcionamiento de dichas operaciones. [12 pts]
2. ¿En qué casos es mejor una estructura de datos Vector que una Lista? [12 pts]
3. Considere el siguiente pseudocódigo, que define una lista doblemente enlazada de nodos:

```
struct Node {  
    data;           // Dato almacenado en el nodo  
    next;           // Puntero al nodo siguiente (NULL para el último nodo)  
}  
struct List {  
    Node FirstNode; // La lista apunta al primer nodo; NULL si está vacía  
    Node LastNode;  // La lista apunta al último nodo; NULL si está vacía  
}
```

Para insertar un nodo `newNode` después de un nodo `node`, se define la función `insertAfter`:

```
function insertAfter(Node node, Node newNode) {  
    newNode.next := node.next;  
    node.next := newNode;  
}
```

Para insertar un nodo `newNode` al inicio de la lista `list`, se define la función `insertBeginning`:

```
function insertBeginning(List list, Node newNode) {  
    newNode.next := list.firstNode;  
    list.firstNode := newNode;  
}
```

A partir de lo anterior, defina:

- a) Una función `insertEnd`, que inserta un nodo `newNode` al final de `list`. [12 pts]
- b) Una función `removeAfter`, que elimina un nodo `Node`. [12 pts]

Estructuras de datos

Control 2 - Pauta

Prof: Fabián Riquelme Csori
2017-II

1. a) Son respuestas válidas `size()`, `isEmpty()`, `push(element)`, `top()` o `create()`.
Las operaciones tipo `pop()` o de acceso directo a cualquier índice no deberían estar permitidas. [12 pts]
1. b) Cualquiera análoga a las vistas en clases para Pilas o Colas son válidas, salvo aquellas que consideren las operaciones `pop()`. [12 pts]
2. Cuando no se requieren tantas operaciones de inserción, extracción o desplazamiento de datos durante la ejecución.
Cuando se requiere acceder frecuentemente a los datos. [12 pts]
3. a) La función queda definida por:

```
function insertEnd(List list, Node newNode) {  
    (list.LastNode).next := newNode;  
}
```

 [12 pts]
3. b) La función queda definida por:

```
function removeAfter(Node node) {  
    aux := node.next;  
    node.next := node.next.next;  
    destroy aux;  
}
```

 [12 pts]