

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

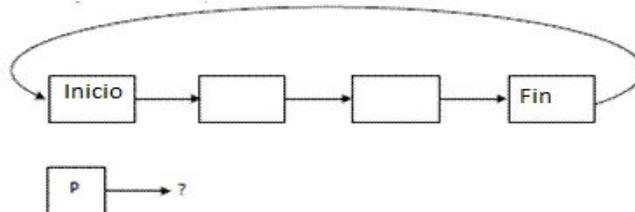
FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN
ESTRUCTURAS DE DATOS
PRIMERA EVALUACIÓN - I TÉRMINO 2016

Nombre: _____ Matrícula: _____

TEMA 1 (15 PUNTOS)

Seleccione la respuesta correcta y **justifique** su respuesta.

1. En el peor escenario, el número de comparaciones necesarias para buscar un elemento dado en una lista simplemente enlazada de longitud N es:
 - a. 1
 - b. N^2
 - c. $N/2$
 - d. N
2. Considere una lista simplemente enlazada con una referencia al nodo inicio y otra referencia al nodo final, Cuál de las siguientes operaciones dependen del tamaño de la lista?
 - a. Eliminar el primer elemento
 - b. Insertar un elemento al inicio
 - c. Eliminar el último elemento
 - d. Insertar un elemento al final
3. Una lista circular simplemente enlazada es utilizada para representar una Cola. Una variable p es utilizada para acceder a la Cola. A cuál nodo debería apuntar la variable p de tal forma que las operaciones encolar y desencolar tomen un tiempo de $O(1)$?



- a. Al nodo fin
 - b. Al nodo inicio
 - c. No es posible con un solo nodo
 - d. El nodo siguiente del inicio
4. Cuál de las siguientes afirmaciones es verdadera con respecto a la implementación de una **cola** utilizando una lista simplemente enlazada?
 - a. En la operación encolar, si los nodos son insertados al inicio de la lista, entonces en la operación desencolar, los nodos son removidos desde el final de la lista
 - b. En la operación encolar, si los nodos son insertados al final de la lista, entonces en la operación desencolar, los nodos son removidos desde el inicio de la lista
 - c. Las dos afirmaciones anteriores son correctas
 - d. Ninguna de las anteriores

- 5.Cuál de las siguientes afirmaciones es verdadera con respecto a la implementación de una **pila** utilizando una lista simplemente enlazada?
- En la operación push, si los nodos son insertados al inicio de la lista, entonces en la operación pop, los nodos son removidos desde el final de la lista
 - En la operación push, si los nodos son insertados al final de la lista, entonces en la operación pop, los nodos son removidos desde el inicio de la lista
 - Las dos afirmaciones anteriores son correctas
 - Ninguna de las anteriores

TEMA 2 (5 PUNTOS)

Para cada uno de los siguientes escenarios escoja la “mejor” estructura de datos. Puede seleccionar una o una combinación de las siguientes y **justifique** su respuesta:

- Lista basada en arreglos
 - Lista Simplemente Enlazada
 - Lista Doblemente Enlazada
 - Lista Circular Simplemente Enlazada
 - Lista Circular Doblemente Enlazada
 - Pila
 - Cola
 - Cola de Prioridad
1. Los navegadores web como: Firefox, Chrome, Internet Explorer, entre otros. almacenan las direcciones web de los sitios que los usuarios visitan. Los navegadores web brindan a los usuarios la funcionalidad de poder regresar a los sitios que recientemente has visitado.
 2. Las urbanizaciones privadas cuentan con un sistema de control de acceso a sus diferentes etapas. Los residentes tienen una tarjeta de proximidad que es utilizada sobre un lector que autoriza su ingreso a la etapa. Los visitantes tienen que esperar a que un guardia de seguridad tome sus datos y solicite autorización al residente para permitir el ingreso del visitante.
 3. Una empresa de marketing fabrica letreros de publicidad continua, es decir, las publicidades se repiten cada cierto tiempo en un orden preestablecido. Estos letreros permiten al usuario visualizar una publicidad de un producto y cambian automáticamente.

TEMA 3 (25 PUNTOS)

1. **(10 puntos)** Implemente el método **borrarBase**, que elimina el elemento situado en la base (o fondo) de la pila y lo devuelve como resultado. Además, la pila debe quedar con los elementos originales sin la base.

```
public static Object borrarBase(Stack<Object> pila)
```

2. **(15 puntos)** Implemente el método **topeBase**, que intercambia el elemento del tope de una pila con el elemento de la base.

```
public static void topeBase(Stack<Object> pila)
```

TEMA 4 (25 PUNTOS)

1. **(10 puntos)** Utilizando la clase LinkedList de Java. Implemente el método estático **removeDuplicados**, que recibe como parámetro un LinkedList con números en cualquier orden y modifica la lista para que no contenga elementos repetidos.

```
public static void removeDuplicados(LinkedList<Integer> lista) {}
```

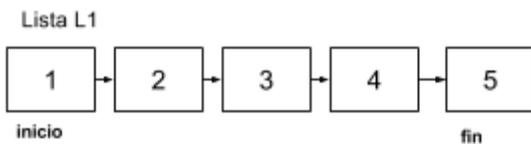
2. **(15 puntos)** Considere la siguiente definición de la Lista Simplemente Enlazada.

```
public class NodoLista {  
    private int dato;  
    private NodoLista siguiente;  
}  
public class ListaSimplementeEnlazada {  
    private NodoLista primero;  
    private NodoLista ultimo;  
}
```

Se le solicita implementar dentro de la ListaSimplementeEnlazada el método **splitReverse** que recibe un NodoLista **p** como parámetro y modifica la lista de tal forma que los nodos después de **p** estén ubicados ahora al comienzo de la lista. **Este método debe tomar como tiempo $O(1)$.**

```
private void splitReverse(NodoLista p) {}
```

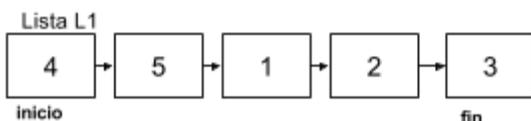
Por ejemplo:



NodoLista p



splitReverse(p);



TEMA 5 (30 PUNTOS)

El I.E.S.S ha decidido implementar el servicio de telemedicina en su call center de atención a los afiliados, la finalidad de este servicio es descongestionar la atención de consulta externa en los diferentes hospitales del país. Para llevar a cabo este propósito, el I.E.S.S ha decidido incorporar un grupo de médicos de Medicina General para que sean parte del grupo de agentes que atienden las llamadas en el call center.

Un afiliado tiene un número de cédula, nombres, apellidos y un historial de dolencias.

Una dolencia tiene un nombre y un grupo de síntomas.

Un médico tiene un número de cédula, nombres, apellidos y un histórico de afiliados que ha atendido.

El call center atiende las llamadas de los afiliados conforme han sido recibidas por la central telefónica. El afiliado es asignado a un médico bajo el criterio del médico que ha atendido menos afiliados.

A usted se le solicita:

- a) **(5 puntos)** Definir los TDAs necesarios.

- b) **(10 puntos)** Implementar el procedimiento **medicosTratantes** que recibe una lista de médicos y una dolencia y retorna una lista de médicos que han tratado esa dolencia.

```
public static LinkedList<Medico> medicosTratantes(LinkedList<Medico> medicos, Dolencia c)
```

- c) **(15 puntos)** Implementar el procedimiento **atenderLlamadas** que simula la atención de las llamadas de los afiliados en el call center. El procedimiento recibe una lista de médicos y una cola que representa las llamadas de los afiliados:

```
public static void atenderLlamadas(LinkedList<Medico> medicos, Queue<Afiliado> llamadas)
```