**UNIDAD TEMÁTICA 3: Listas, Pilas y Colas**

**PRACTICOS DOMICILIARIOS INDIVIDUALES**

# Ejercicios #1-5 Lista encadenada

Los nodos de una lista simplemente encadenada tienen dos atributos:

* DATOS, de tipo “dato”.
* SIGUIENTE, de tipo “nodo de lista”, que hace referencia al nodo siguiente en la lista.

# Ejercicio #1

Sean **nodo1**, **nodo2** y **nodo3** tres nodos consecutivos de una lista (nodo2 es el siguiente a nodo1 y nodo3 es el siguiente a nodo2).

Analice el siguiente fragmento de código (utilice dibujos o diagramas para clarificar qué es lo que sucede):

Nuevo nodo otroNodo

otroNodo.siguiente  nodo1 nodo2.siguiente  nodo3

1. Inserta “otroNodo” en la lista, quedando como anterior a nodo1.
2. Inserta “otroNodo” en la lista, quedando entre nodo1 y nodo2.
3. Elimina nodo2 de la lista.
4. No tiene ningún efecto sobre la lista.

Crea el nuevo nodo y lo hace anterior a nodo1. A

Sean **nodo1**, **nodo2** y **nodo3** tres nodos consecutivos de una lista (nodo2 es el siguiente a nodo1 y nodo3 es el siguiente a nodo2).

Analice el siguiente fragmento de código (utilice dibujos o diagramas para clarificar qué es lo que sucede):

Nuevo nodo otroNodo

otroNodo  nodo1.siguiente

nodo1.siguiente  nodo3

1. Inserta “otroNodo” en la lista, quedando como anterior a nodo1.
2. Inserta “otroNodo” en la lista, quedando entre nodo1 y nodo2.
3. Elimina nodo2 de la lista.
4. No tiene ningún efecto sobre la lista.

Crea un nuevo nodo (otroNodo) y hace que apunte a nodo2. Luego nodo1.siguiente pasa a ser nodo3. Se podría decir que se elimina a nodo2 de la lista y otroNodo toma su lugar.

# Ejercicio #3

Sean **nodo1**, **nodo2** y **nodo3** tres nodos consecutivos de una lista (nodo2 es el siguiente a nodo1 y nodo3 es el siguiente a nodo2).

Analice el siguiente fragmento de código (utilice dibujos o diagramas para clarificar qué es lo que sucede) y responda las preguntas proyectadas en pantalla:

Nuevo nodo otroNodo

otroNodo.siguiente  nodo1.siguiente nodo1.siguiente  otroNodo

1. Inserta “otroNodo” en la lista, quedando como anterior a nodo1.
2. Insert “otroNodo” en la lista, quedando entre nodo1 y nodo2.
3. Elimina nodo2 de la lista.
4. Dará error en tiempo de ejecución si nodo1 es el primero o nodo3 es el último.

Crea un nuevo nodo el cual pasara a ser parte de la lista, encontrándose posicionado entre nodo1 y nodo2.

**Ejercicio #**4

Analice el siguiente fragmento de código (utilice dibujos o diagramas para clarificar qué es lo que sucede) y responda las preguntas proyectadas en pantalla:

Nuevo nodo otroNodo

Nuevo nodo nodoActual

nodoActual  primero

mientras nodoActual <> nulo hacer nodoActual  nodoActual.siguiente fin mientras

nodoActual.siguiente  otroNodo

1. Inserta correctamente “otroNodo” en la lista, quedando como último nodo.
2. Inserta correctamente “otroNodo” en la lista, quedando como primer nodo.
3. El algoritmo está mal hecho, ya que dará error en tiempo de ejecución si la lista está vacía.
4. El algoritmo está mal hecho, ya que dará siempre error en tiempo de ejecución.

Respuesta: A

# Ejercicio #5

Analice el siguiente fragmento de código (utilice dibujos o diagramas para clarificar qué es lo que sucede) y responda las preguntas proyectadas en pantalla:

Nuevo nodo otroNodo   
Nuevo nodo nodoActual   
nodoActual  primero

mientras nodoActual.siguiente <> nulo hacer nodoActual  nodoActual.siguiente   
fin mientras

nodoActual.siguiente  otroNodo

1. Inserta correctamente “otroNodo” en la lista, quedando como último nodo.
2. Inserta correctamente “otroNodo” en la lista, quedando como primer nodo.
3. El algoritmo está mal hecho, ya que dará error en tiempo de ejecución si la lista está vacía.
4. El algoritmo está mal hecho, ya que dará siempre error en tiempo de ejecución.

Si la lista esta vacia da error en tiempo de ejecución, ya que null no tiene un atributo siguiente.