

Nombre Estudiante:	Código:	Grupo:
Profesor:	Fecha: 19 de noviembre de 2018	

Unidad 6: Estructuras y Algoritmos Recursivos

Al finalizar esta unidad, el estudiante estará en capacidad de:

OE5.1 Utilizar estructuras enlazadas de objetos para modelar grupos de atributos no primitivos de tamaño flexible.

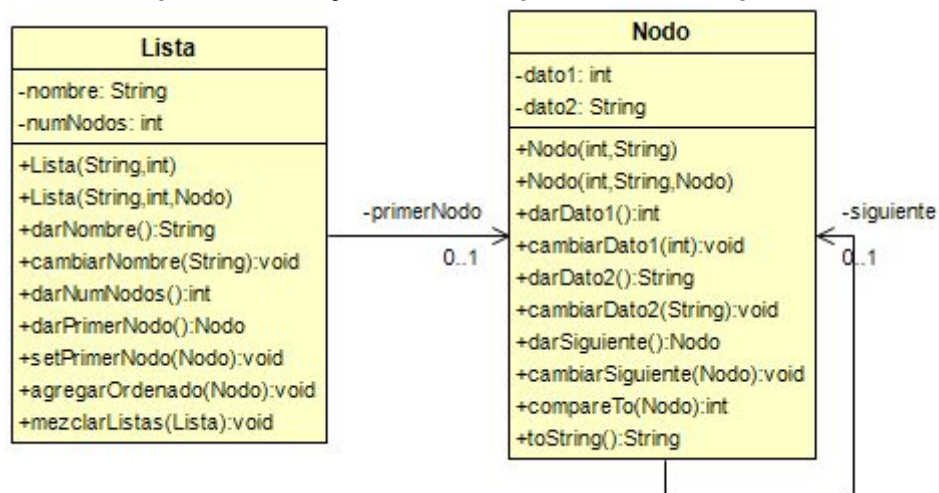
OE5.2 Escribir los algoritmos necesarios para manipular estructuras lineales que almacenan sus elementos enlazándolos entre ellos

OE6.1 Emplear el concepto de recursividad como una alternativa a la estructura de control iterativa.

OE6.2 Aplicar la computación recursividad en la solución de problemas de naturaleza inherentemente autocontenida.

Enunciado

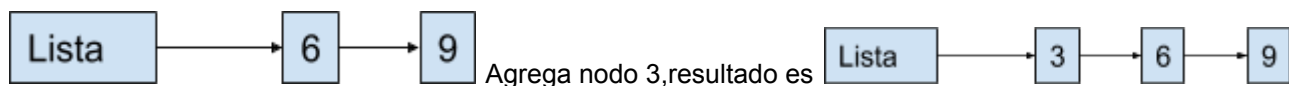
Se le ha solicitado a usted que implemente las siguientes operaciones de listas enlazadas. Para los enunciados que se presentan a continuación debe tener como referencia el siguiente diagrama de clase. Antes de escribir su solución por favor **lea el enunciado completo de ese ejercicio**. Usted puede definir e **implementar métodos auxiliares**.



1. [35%][OE5.1, OE5.2, OE6.1, OE6.2] Adicionar un nodo de forma ordenada

Desarrollar un método recursivo en Java que recibe como argumento un nodo y lo agrega a la lista de nodos ordenada según el orden natural. La lista construida hasta el momento de adicionar el nodo ya está ordenada.

Ejemplo:

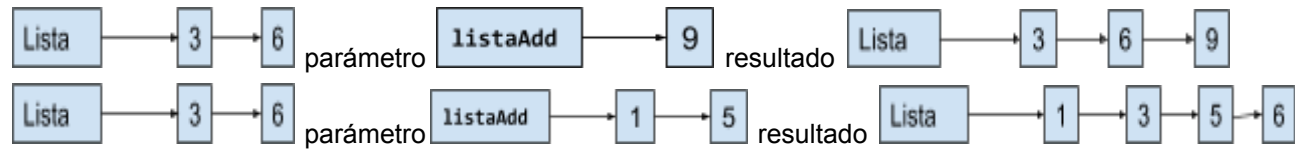


```
public void agregarOrdenado(Nodo nodo) {
```

```
}
```

2. [35%][OE5.1, OE5.2, OE6.1, OE6.2] Mezclar dos listas en forma ordenada

Desarrollar un método recursivo en Java que recibe como argumento una lista simple ordenada y la mezcla de forma ordenada con la lista de nodos ordenada apuntada por el atributo primerNodo. La mezcla de las dos listas debe hacerse según el criterio de orden natural. La lista construida hasta el momento de mezclar la lista y la lista que recibe como parámetro ya están ordenadas. Ejemplo:

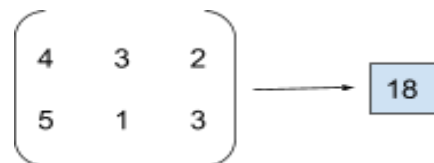


```
public void mezclarListas(Lista listaAdd) {
```

```
}
```

3. [35%][OE6.1] Sumar elementos de la matriz

Desarrollar un método recursivo en lenguaje Java que recibe como argumento una matriz m [][] de enteros y obtiene como resultado la suma de todos los elementos de la matriz. La solución debe ser desarrollada usando recursión y puede usar métodos auxiliares si lo requiere (estos deberán ser implementados). Ejemplo:



```
public int sumarMatriz ( int [ ][ ] m ) {
```

```
}
```