**Ejercicio:**

Escribir el código de los métodos de la clase **Lista** implementadacomo una lista enlazada simple.

La declaración completa de la **clase Lista** es la siguiente:

**class Lista**

protected int **tam**;

protected Nodo **primero**;

protected Nodo **recorrido**;

**public Lista()**

/\* Crea una nueva lista, vacía \*/

**public void vaciar()**

/\* vacia una lista ya existente \*/

**public boolean esVacia()**

/\* Devuelve true si la lista no tiene ningún elemento,

false en caso contrario \*/

**public int longitud()**

/\* Devuelve el numero de elementos de la lista \*/

**public void insertarPrimero( Object x );**

/\* Inserta X en la primera posición de la lista \*/

**private Nodo devuelvePos( int pos )**

/\* Precondición: 1 <= pos <= tam

Devuelve el nodo de la posición pos \*/

**public void insertarPos( Object x, int pos)**

/\* Inserta X en la posición Pos de la lista,

1 <= pos <= tam + 1 \*/

**public void modificarPos( Object x, int pos )**

/\* Modifica el valor del elemento situado en la posicion Pos

de la lista, 1 <= pos <= tam \*/

**public void borrarPos( int pos )**

/\* Borra el elemento situado en la posición Pos de la lista,

1 <= pos <= tam \*/

**public void borrar( Object x );**

/\* Borra el elemento de la lista cuyo atributo Elemento

sea igual a X }

**public Object extraerPos( int posi )**

/\* Devuelve la referencia al elemento situado

en la posición Pos de la lista \*/

**public int buscar( Object x )**

/\* Busca el elemento X en la lista y devuelve su posición

si existe. Si no existe devuelve 0 }

**/\* OPERACIONES DE RECORRIDO \*/**

**public void inicioRecorrido()**

/\* Inicializa el recorrido en la lista a su primer elemento \*/

**public Object getElemento()**

/\* Coge el elemento en curso en el recorrido,

devuelve su referencia (sin copia)

y avanza el recorrido al siguiente \*/

**public boolean finRecorrido()**

/\* Devuelve true si se ha recorrido ya el último elemento

de la lista, false en caso contrario \*/

{