11/03/15

# Invertir lista

Public void Invertir( ){

If (inicio == null){

System.out.println(“Lista vacía”);

Return;

}

If (inicio.siguiente == null){

Return;

}

Lista nevalista = new Lista( );

Nodo temp = inicio;

While (temp != null){

Nuevalista.insertarInicio(temp);

Temp = temp.siguiente;

}

Inicio = nuevalista.getInicio( );

}

# Buscar Elemento

Public Nodo BuscarElemento(Nodo elemen){

Nodo p = inicio;

While (p != null){

Equals es una función de **Object** y se encarga de hacer la comparación.

If (p.equals(elemen))

Return p;

p = p.siguiente;

}

System.out.println(“Elemento no encontrado”);

}

# **Lista Circular Doble:**

* Inicio
* Final
* Anterior
* Siguiente

# Insertar al Inicio

Public void InsertarInicio(Object valor){

Ìf (inicio == null){

Inicio = new Nodo(valor);

Final = inicio;

Inicio.siguiente = final;

Inicio.anterior = inicio;

}

Else {

Nodo p = new Nodo(valor);

Inicio.siguiente = p;

Inicio.anterior = final;

Final.siguiente=inicio;

}

}

# Insertar en Posición.

Public void InsertarPos(Object Elemen, int pos) {  
 if (pos == 0){

InsertarInicio(Elemen);

Return;

}

Nodo nuevoNodo = new Nodo(Elemen);

Nodo p = Inicio;

Int i = 0;

While (p != final && i< pos){

i++;

p =p.siguiente;

Nodo anterior = p.anterior;

Anterior.siguiente = nuevoNodo;

nuevoNodo.anterior = anterior;

nuevoNodo.siguiente = p;

p.anterior = nuevoNodo;

}

If (pos == final && i>pos){

InsertarFinal(Elemen);

}

}