



UNIVERSIDAD NACIONAL DE CAJAMARCA
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA DE SISTEMAS



ALGORITMOS Y ESTRUCTURA DE DATOS II

06: LISTAS CIRCULARES SIMPLES

INTRODUCCIÓN

Semana 06 - Listas circulares simples



OBSERVE



- Para tener acceso a la lista se tiene un nodo especial que permite tener acceso a la lista completa
 - Nodo lista;
- Pero para mejorar el rendimiento de las listas al realizar las diferentes operaciones se estableció dos referencias a la lista
 - Nodo inicio
 - Nodo fin
- ¿Qué pasa si la lista es circular?



RECORDEMOS

- ¿Qué tipos de listas sea estudiado hasta el momento?
- ¿Qué características han tenido cada una de las listas? Ventajas y desventajas
- ¿En dónde puedo aplicar los conceptos estudiados de listas en problemas de la vida diaria?
- ¿Qué operaciones puedo implementar en cada lista y qué casos especiales tiene cada uno de ellas?



¿CÓMO LO RESOLVERÍA?



- ¿Qué pasaría si le piden implementar una lista circular simple y en ella crear un método que le permita determinar si un elemento existe o no, y de existir que muestre su siguiente?

LOGRO ESPERADO



- Al finalizar la sesión, el estudiante presenta un informe que describe los métodos para manipular una lista circular simple, a través de diagramas con sus respectivos algoritmos e implementación en java verificando el funcionamiento adecuado de cada método y sus casos particulares.



DESARROLLO DEL TEMA

Estructura de Datos: listas circulares simples



LISTAS CIRCULARES

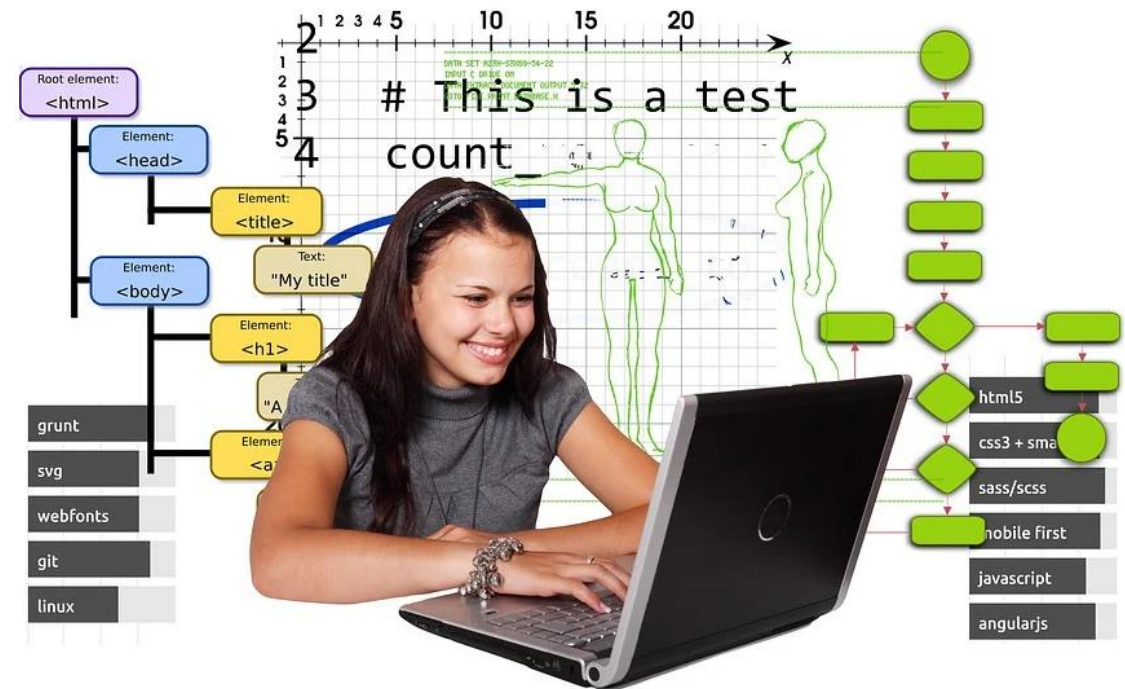


- Es una lista lineal en la que el último nodo a punta al primero.
- En éstas listas no existen casos especiales, cada nodo siempre tiene uno anterior y uno siguiente.
- A pesar de que las listas circulares simplifiquen las operaciones sobre ellas, también introducen algunas complicaciones. Por ejemplo, en un proceso de búsqueda, no es tan sencillo dar por terminada la búsqueda cuando el elemento buscado no existe.

LISTAS CIRCULARES



- Las operaciones que se pueden realizar sobre ellas:
 - Añadir o insertar elementos.
 - Buscar o localizar elementos.
 - Borrar elementos.
 - Recorrer la lista.
- Cada una de éstas operaciones podrá tener varios casos especiales, por ejemplo, tendremos que tener en cuenta cuando se inserte un nodo en una lista vacía, o cuando se elimina el único nodo de una lista.



CLASE NODO, MÉTODO CONSTRUCTOR Y ESTÁ VACÍA



```
public class Nodo {
    String dato;
    Nodo sgt;
    public Nodo(String dato, Nodo sgt) {
        this.dato = dato;
        this.sgt = sgt;
    }
    public Nodo(String dato) {
        this.dato = dato;
        this.sgt = null;
    }
    public String getDato() {
        return dato;
    }
    public void setDato(String dato) {
        this.dato = dato;
    }
    public Nodo getSgt() {
        return sgt;
    }
    public void setSgt(Nodo sgt) {
        this.sgt = sgt;
    }
    @Override
    public String toString() {
        return "" + dato;
    }
}
```

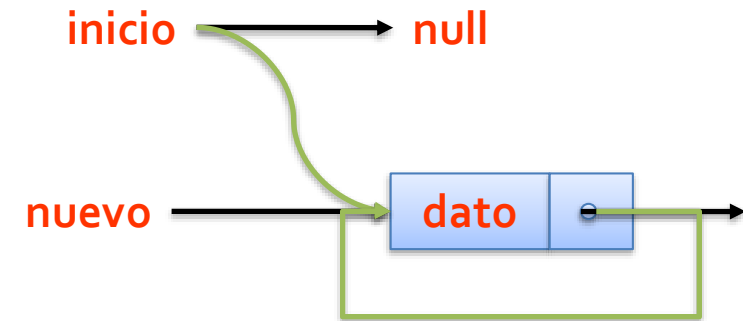
```
public ListaCircularS() {
    this.inicio = null;
    this.size = 0;
}
```

```
private boolean estaVacia() {
    return inicio == null;
}
```



INSERTAR NODOS EN UNA LISTA ENLAZADA

- **Insertar un nodo en una lista vacía:**
 - Partimos de que ya se tiene el nodo a insertar y la lista vacía.
 - Se hace que inicio referencie a nuevo,
 - El siguiente de inicio referencia a nuevo



```
inicio = null;
```

```
Nodo nuevo = new Nodo(dato);
```

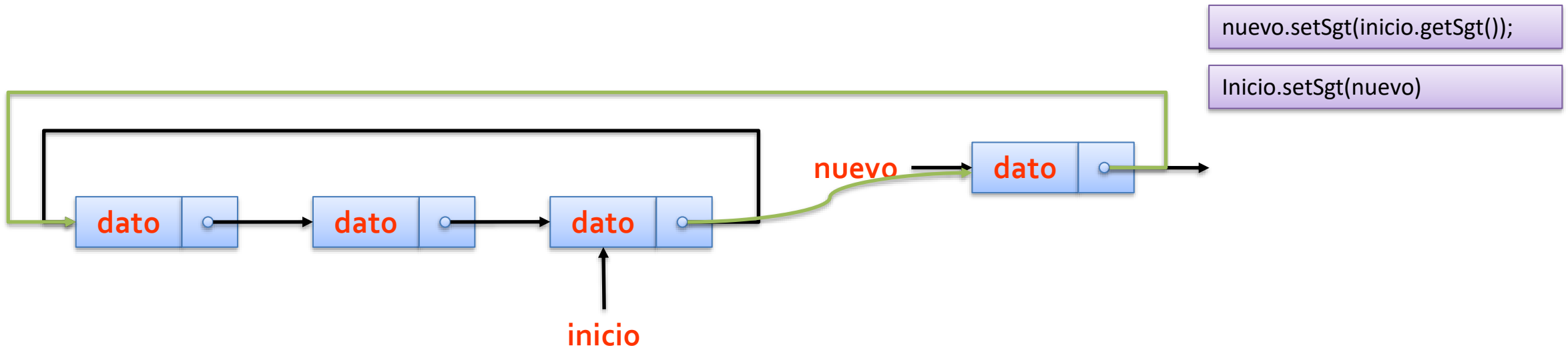
```
Inicio = nuevo;
```

```
inicio.setSgt(nuevo);
```



INSERTA UN NODOS EN UNA LISTA CIRCULAR SIMPLE

- **Insertar un elemento en una lista circular no vacía:**
 - Se parte de una lista no vacía y el nodo a insertar referenciado por nuevo
 - El siguiente de nuevo referencia al siguiente de inicio.
 - A continuación el siguiente de inicio referencia a nuevo.





MOSTRAR UNA LISTA CIRCULAR

- Para mostrar la lista se puede realizar a través del método toString()
- Para ello se inicializa la iterador en inicio. Esta variable permite recorrer la lista.
- Para el Nodo referenciado por iterador obtenemos el dato y se lo muestra o acumula en un String.
- El iterador apunta al siguiente nodo
- Se repite los dos pasos anteriores mientras iterador sea diferente de inicio

```
Nodo iterador = inicio;
```

```
Iterador.getsgt();
```

```
iterador = iterador.getSgt()
```

```
while (iterador != inicio)
```




BUSCAR UN ELEMENTO EN UNA LISTA CIRCULAR

- Se realiza a través de un método que reciba el valor a buscar devuelva un valor booleano indicando si se encontró
- Se inicializa la variable encontrado en falso.
- Si la lista no está vacía se inicializa iterador en inicio.
- Mientras el dato obtenido por iterador sea diferente del valor buscado y el siguiente de iterador sea diferente de inicio, el iterador apunta al siguiente nodo.
- Al salir del mientras, si el dato que es referenciado por iterador es igual al valor buscado se cambia el estado de encontrado por verdadero . Finalmente se retorna encontrado.

```
boolean encontrado = false;
```

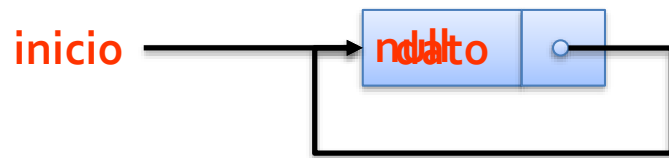
```
Nodo iterador = inicio;
```

```
iterador = iterador.getSgt();
```

```
if(iterador.getDato() == valor)  
    encontrado = true;
```

ELIMINAR UN NODO EN UNA LISTA CIRCULAR SIMPLE

- **Eliminar el único nodo de una lista circular**
 - Si la lista tiene un solo elemento se hace que inicio tome el valor de null
 - El recolector de basura elimina el nodo.

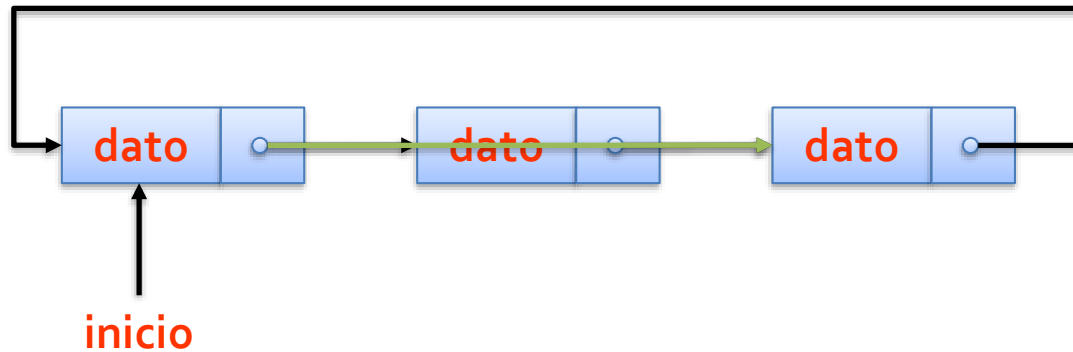


- **Eliminar un nodo cualquiera de la lista circular**
 - Para este caso, se necesita localizar el nodo anterior al que se quiere eliminar.
 - Como el inicio de la lista puede ser cualquier nodo se hace que inicio referencie a la nodo anterior a eliminar.
 - Una vez localizado el nodo anterior referenciado por inicio se hace que inicio.sgt referencie al siguiente de inicio.sgt



ELIMINAR UN NODO EN UNA LISTA CIRCULAR SIMPLE

- **Eliminar un nodo en una lista circular no vacía**
 - Se parte de una lista circular no vacía y que inicio apunte al nodo anterior a eliminar
 - Para que inicio referencie al nodo anterior a eliminar, se hace que inicio referencie a su siguiente, mientras que el siguiente de inicio sea diferente de auxiliar. Auxiliar previamente ha sido inicializado en inicio.
 - A continuación el siguiente de inicio referencie al siguiente de inicio.siguiente.



```
Nodo auxiliar = inicio;  
//lista apunta al nodo anterior a eliminar  
do {  
    if(inicio.getSgt().getDato() != valor)  
        inicio = inicio.getSgt();  
} while (inicio.getSgt().getDato() != valor && inicio != auxiliar);
```

```
inicio.setSgt(inicio.getSgt().getSgt());
```

EVALUACIÓN DEL TEMA DESARROLLADO

Reflexionemos!



PREGUNTAS DE REPASO

- ¿Qué diferencias encuentra en una lista circular respecto de las listas enlazadas?
- Describa el algoritmo que permite eliminar un nodo de la lista circular.
- Implemente un programa que verifique el funcionamiento del método de buscar y modifique su programa de tal manera que si el elemento existe muestre el elemento y su siguiente



EJERCICIOS DE APLICACIÓN



EJERCICIOS DE APLICACIÓN



- Dada una lista circular simple, crear un método que calcule su promedio, y determine si el promedio es igual a alguno de los elementos de la lista
- Crear una lista circular siempre de n nombres e ir borrando cada x veces hasta que quede solo un nombre. Ejemplo $n = 5$, $x = 3$
 - **Ana** – Rosa – **Pedro** – Juan – María
 - **Ana** – Rosa – **Juan** – María
 - **Rosa** – Juan – **María**
 - **Rosa** – Juan
 - Juan