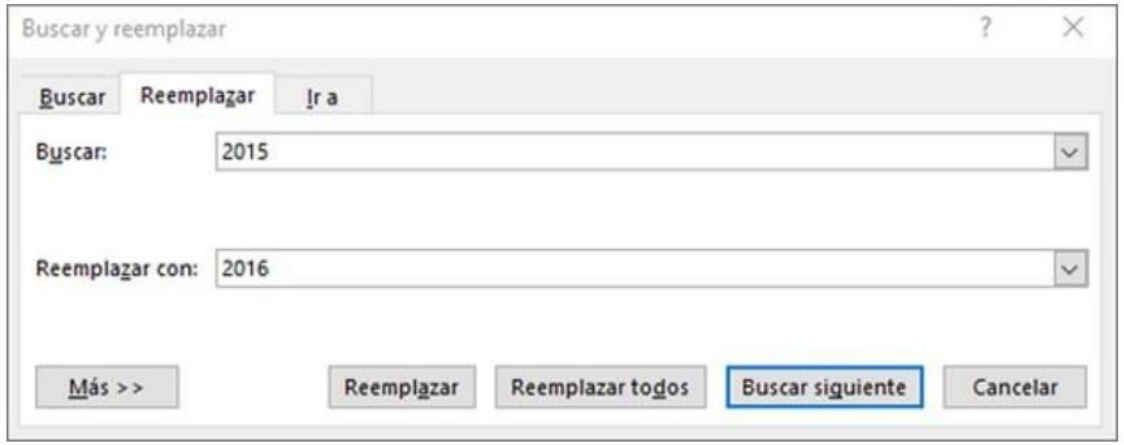
UNIVERSIDAD NACIONAL DE CAJAMARCA
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA DE SISTEMAS

ALGORITMOS Y ESTRUCTURA DE DATOS II
07: LISTAS CIRCULARES DOBLEMENTE ENLAZADAS

INTRODUCCIÓN

Semana 07 - Listas circulares doblemente enlazadas

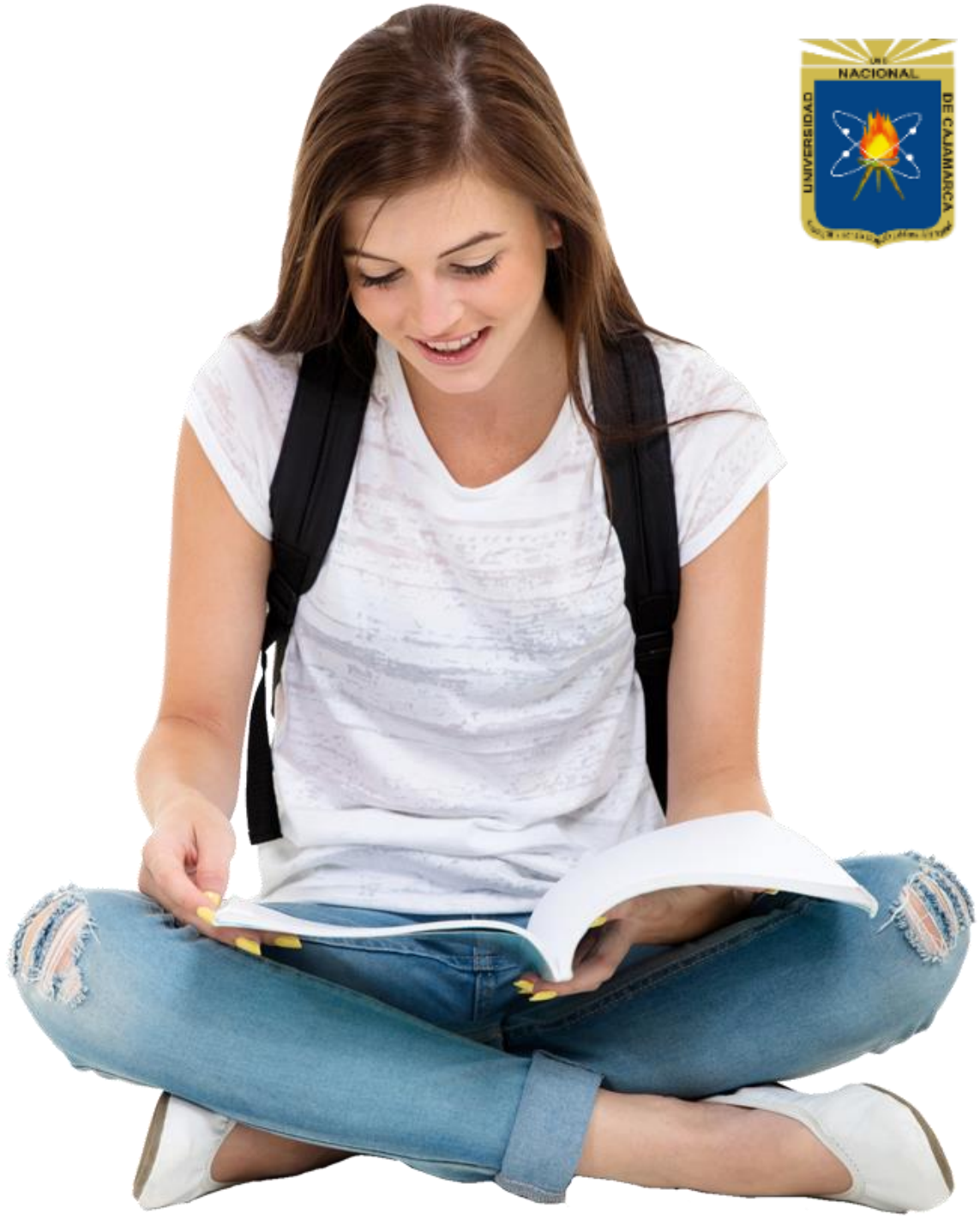
PROCESADOR DE TEXTO



- En algún momento en la revisión de un documento, estando a la mitad no ha tenido que buscar una palabra desde ese punto hacia adelante, incluso si llega al final del documento, este continúa la revisión desde un inicio.
- Pero también puede retroceder y revisar en sentido contrario.
- Observe que en este ejemplo se usa las listas dobles. ¿Identifica cuál?

RECORDEMOS

- ¿Cuál es la característica principal de una lista circular simple?
- Describa el método utilizado para mostrar los elementos de una lista circular simple
- ¿Cómo evita entrar en un bucle al recorrer o buscar en una lista circular simple
- Describa los métodos para insertar y eliminar en una lista circular siemple



¿CÓMO LO RESUELVO?

- Podría describir los pasos para realizar una búsqueda de una elemento (objeto) en una lista circular doblemente enlazada.

LOGRO ESPERADO

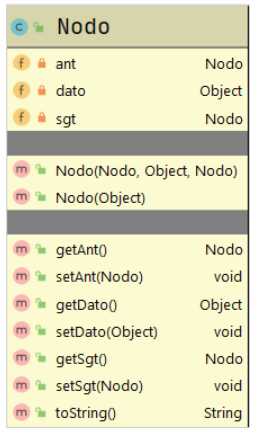
- Al término de la sesión, el estudiante implementa un menú de opciones con todas las tareas que permitan manipular una lista circular, verificando el funcionamiento de cada opción para cada caso particular.

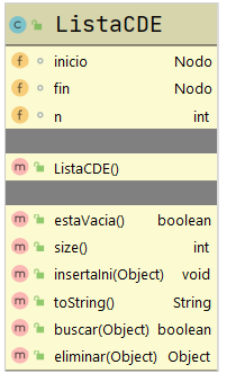
LISTA CIRCULAR DOBLEMENTE ENLAZADA

Estructura de Datos

LISTAS ENLAZADAS DOBLES

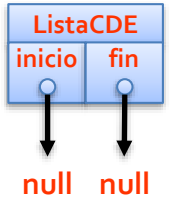
- En una LE doblemente circular, cada nodo tiene dos enlaces, similares a los de la lista doblemente enlazada, excepto que el enlace anterior del primer nodo apunta al último y el enlace siguiente del último nodo, apunta al primero.
- Aunque estructuralmente una lista circular doblemente enlazada no tiene ni principio ni fin, una referencia o referencias de acceso externo pueden establecer el nodo referenciado que está en el inicio o al final.
- Un nodo de una lista circular doblemente enlazada tiene dos referencias para enlazar con los nodos izquierdo y derecho, además de la parte correspondiente al campo dato.
- Se tiene la clase ListaCDE que tiene las referencias inicio y fin. Cuando una lista está vacía las referencias valen null.





INSERTA AL INICIO EN UNA LISTA VACÍA

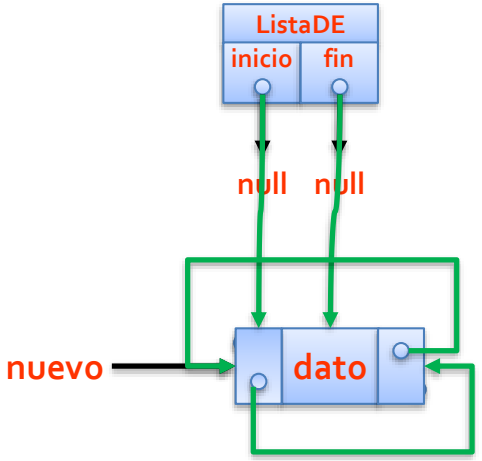
- Se tiene la clase Lista Circular Doblemente Enlazada (ListaCDE) con dos referencias al inicio y fin de la lista
- Insertamos en una lista vacía
  - Se parte de una lista vacía y el nodo a insertar (nuevo)




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

INSERTA AL INICIO EN UNA LISTA VACÍA

- Luego inicio y fin referencian nuevo
- Las referencias anterior y siguiente de nuevo (o inicio y fin) referencian al mismo nodo
  - El siguiente de fin referencia a nuevo
  - El anterior de inicio referencia a nuevo




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

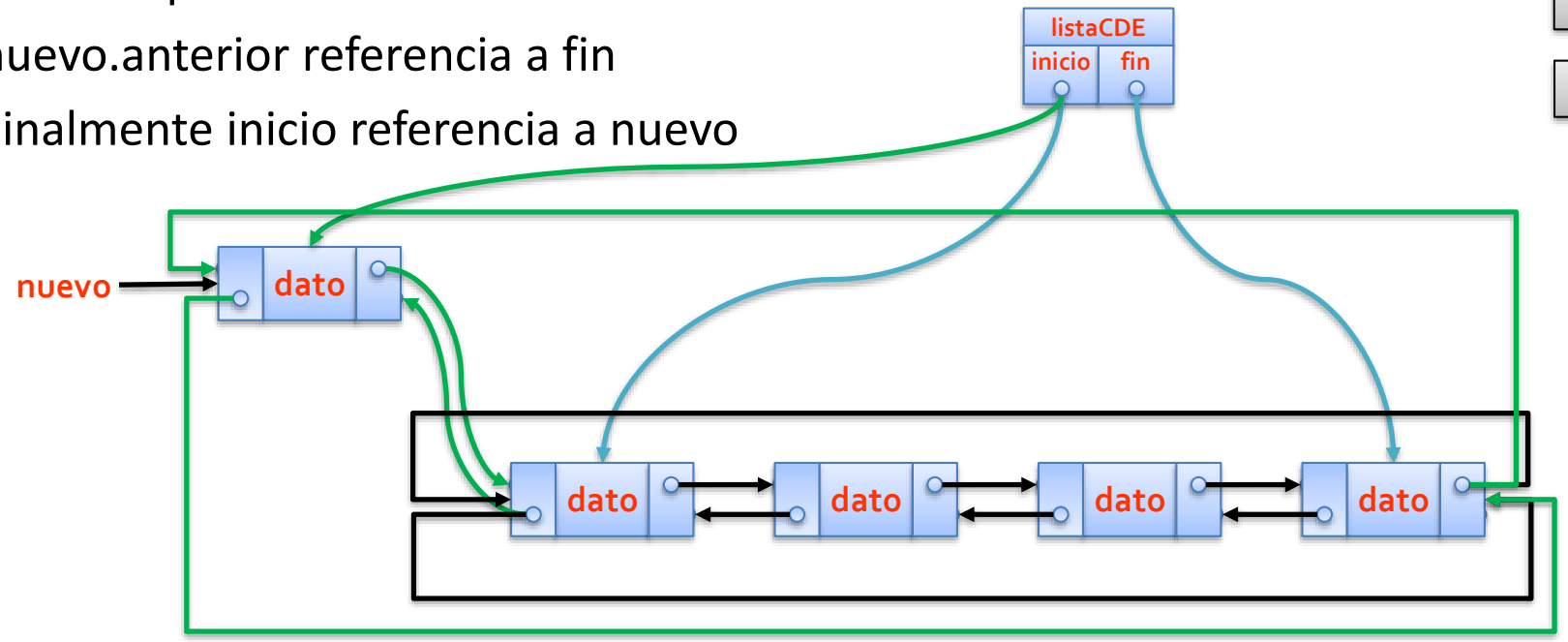
```
inicio = fin = nuevo;
```

```
fin.setSgt(nuevo);
```

```
fin.setSgt(nuevo);
```

INSERTA AL INICIO EN UNA LISTA NO VACÍA

- Si la lista no está vacía
  - Se parte de una lista con elementos y el nodo a insertar (nuevo)
  - Luego nuevo.sgt referencie a inicio
  - fin.sgt referencia a nuevo
  - Se hace que inicio.ant referencie a nuevo
  - nuevo.anterior referencia a fin
  - Finalmente inicio referencia a nuevo

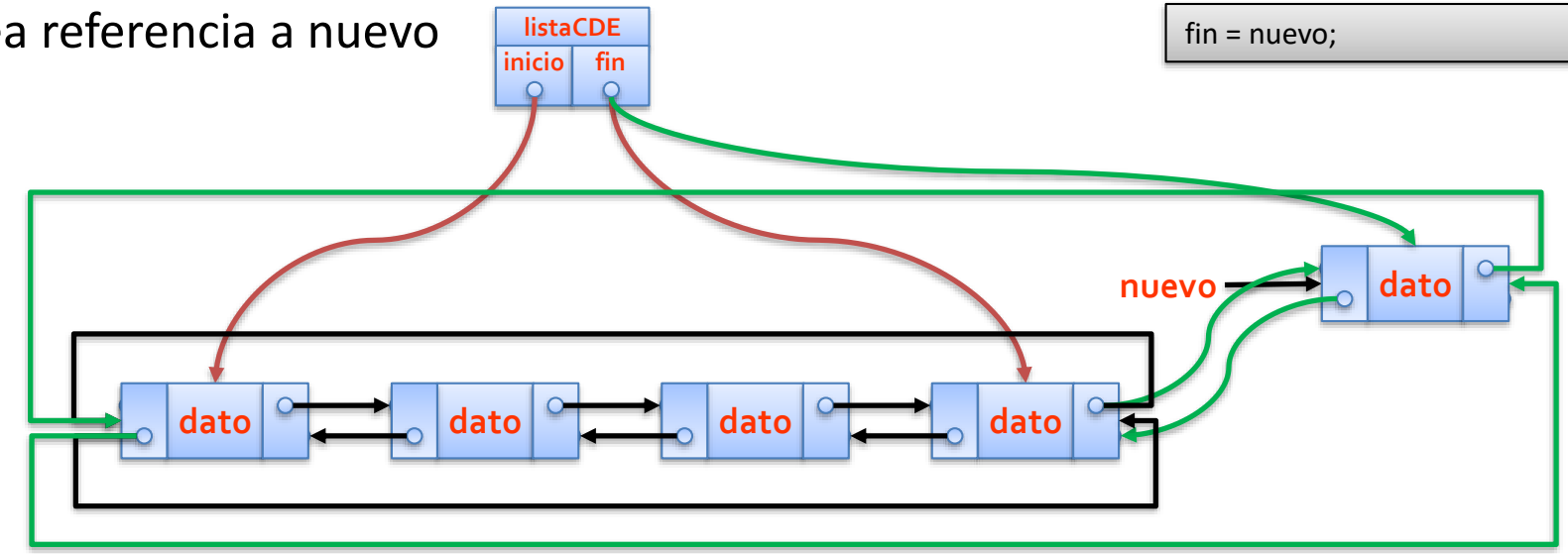



```
Nodo nuevo = new Nodo(dato);
nuevo.sgt = inicio;
fin.sgt = nuevo;
inicio.ant = nuevo;
nuevo.ant = fin;
inicio = nuevo;
```

INSERTAR AL FINAL EN UNA LISTA

- Insertamos en una lista no vacía
  - Se parte de una lista con elementos y el nodo a insertar
  - Se hace que nuevo.sgt referencie a inicio
  - sgt de fin referencie a nuevo
  - inicio.anterior referencia a nuevo
  - Luego el ant de nuevo referencia a fin
  - Finalmente fin sea referencia a nuevo

```
Nodo nuevo = new Nodo (dato);
nuevo.sgt = inicio;
fin.sgt = nuevo;
inicio.anterior = nuevo;
nuevo.anterior = fin;
fin = nuevo;
```

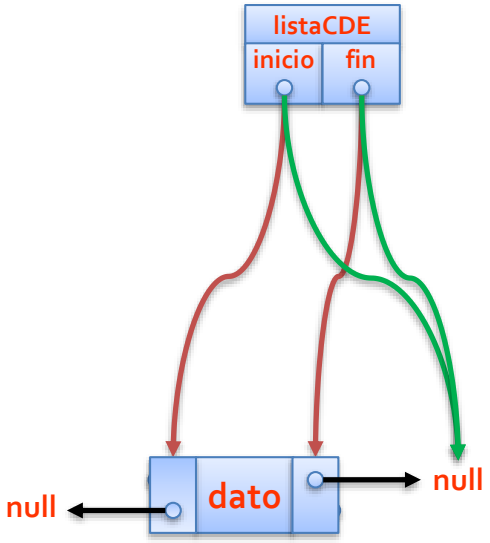


BUSCAR UN ELEMENTO

- Para este método se establece que retorne un booleano que indique si ha sido encontrado o no.
- Se realiza la búsqueda solo si es que la lista no está vacía.
- Se establece un nodo auxiliar en inicio.
- En un bucle hacer mientras auxiliar sea diferente de inicio y encontrado sea falso
- Se pregunta si el dato buscado coincide con el dato de auxiliar, de lo contrario auxiliar apunta al nodo siguiente.

ELIMINAR EL ÚNICO ELEMENTO EN UNA LISTA

- Si la lista tiene un solo elemento
  - Inicio referencia a null
  - Fin referencia a null



inicio = fin = null

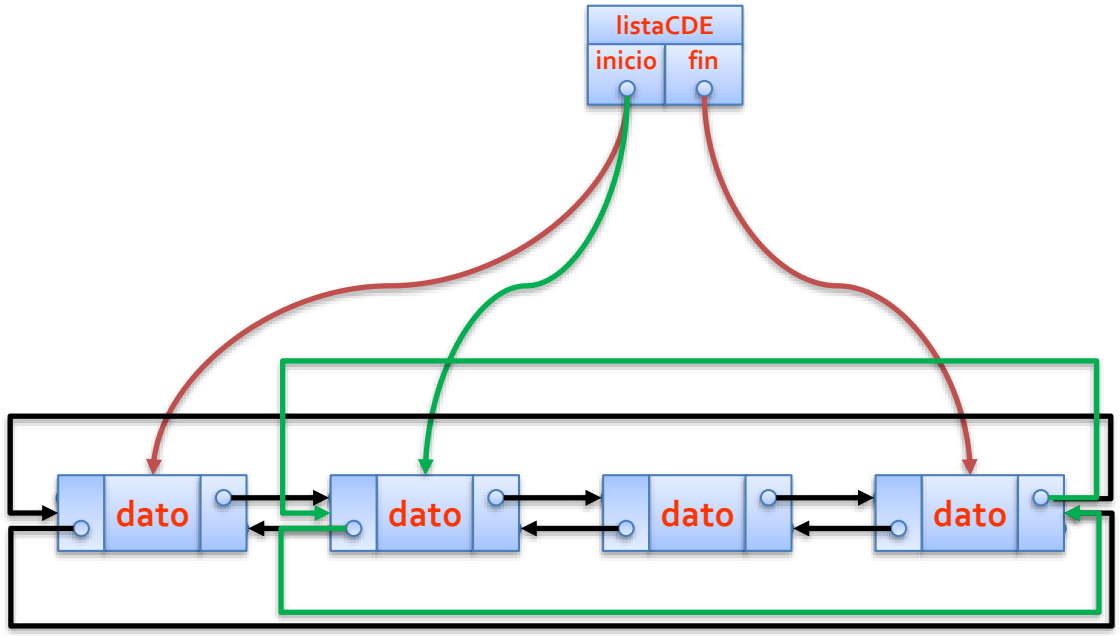
SI EL ELEMENTO A ELIMINAR ES EL INICIO

- Si la lista tiene más de un elemento
  - Siguiente de fin referencia al siguiente de inicio
  - inicio referencia inicio.sgt
  - Inicio.anterior referencia a fin

```
fin.sgt = inicio.sgt;
```

```
Inicio = inicio.sgt;
```

```
inicio.ant = null;
```



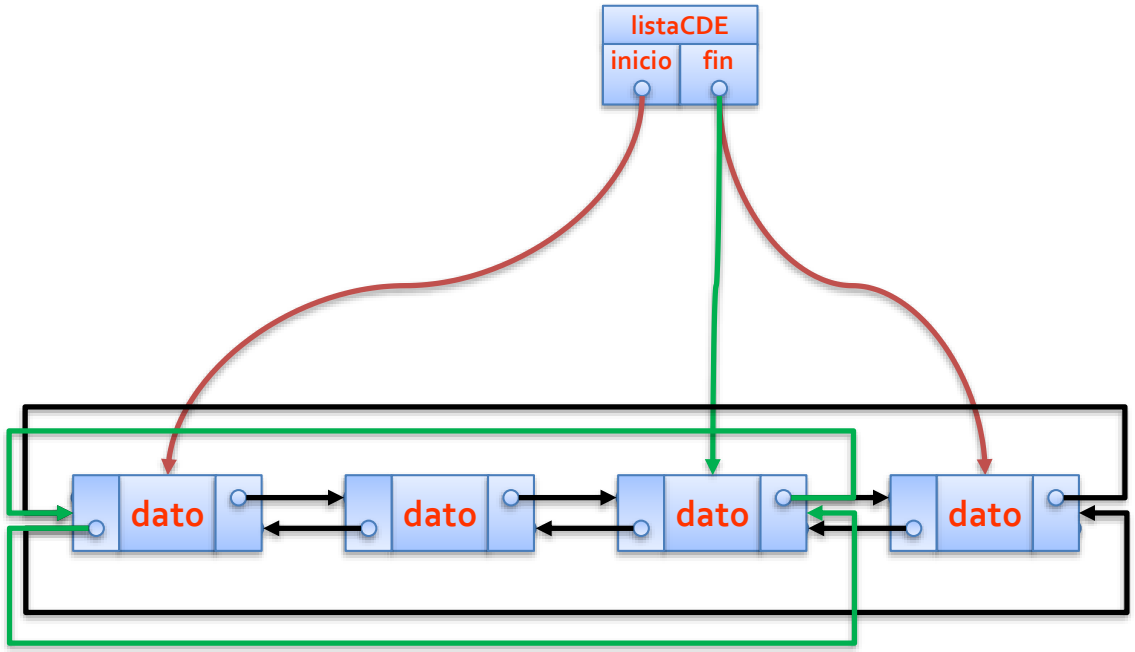
SI EL ELEMENTO A ELIMINAR ES EL FINAL

- Si la lista tiene más de un elemento
  - Anterior de inicio referencia al anterior de fin
  - Fin referencia al anterior de fin
  - fin.sgt referencia a inicio

```
inicio.ant = fin.ant;
```

```
fin = fin.ant;
```

```
fin.sgt = inicio;
```

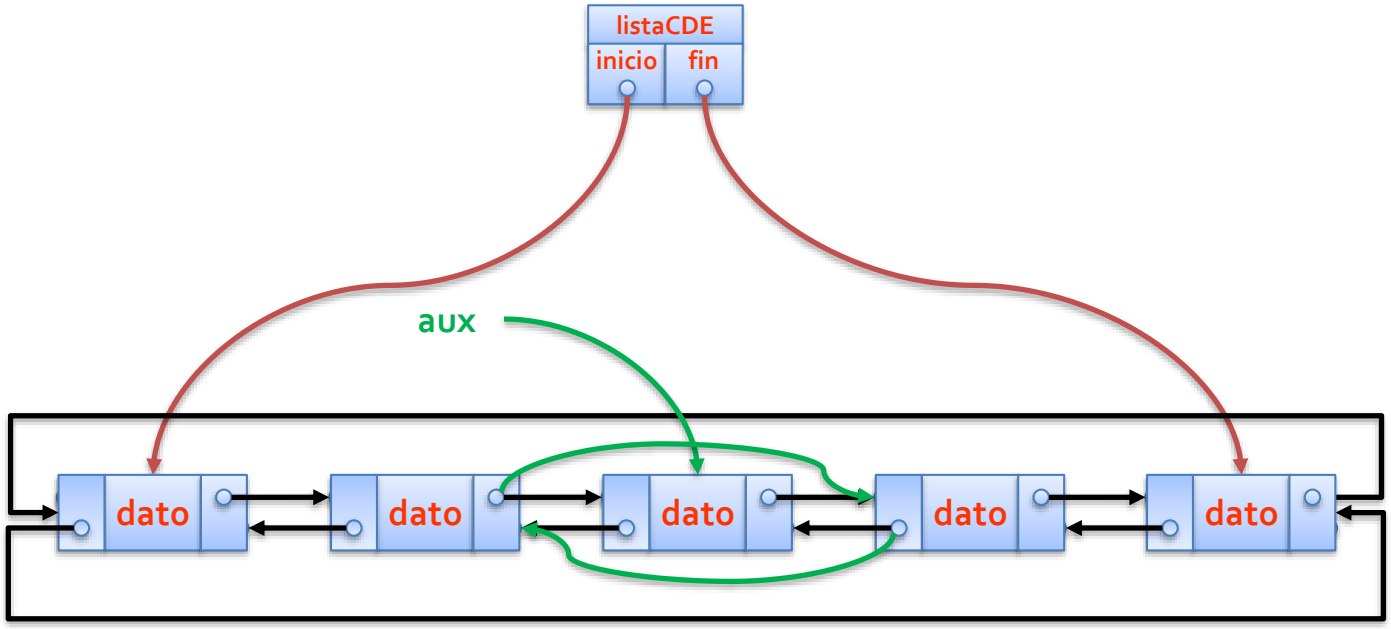


ELIMINAR UN NODO EN UNA POSICIÓN DIFERENTE DE INICIO Y FIN

- Una vez ubicado el nodo a eliminar y referenciado por auxiliar (aux)
- El siguiente del nodo anterior a auxiliar referencia al siguiente de auxiliar
- El anterior del nodo siguiente a auxiliar referencia al anterior de auxiliar

```
aux.ant.sgt = aux.sgt;
```

```
aux.sgt.ant = aux.ant;
```



OTRAS FUNCIONES

```
@Override
public String toString() {
    if (!estaVacia()) {
        String s = "\n[";
        Nodo aux = inicio;
        do {
            s += aux.getDato().toString() + ", ";
            aux = aux.getSgt();
        } while (aux != inicio);
        s = s.substring(0, s.length() - 2);
        s += "]\n";
        return s;
    }
    return "[ ]";
}
```

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

```
public class ListaCDE {
    Nodo inicio;
    Nodo fin;
    int n;

    public ListaCDE() {
        this.inicio = null;
        this.fin = null;
        this.n = 0;
    }
}
```

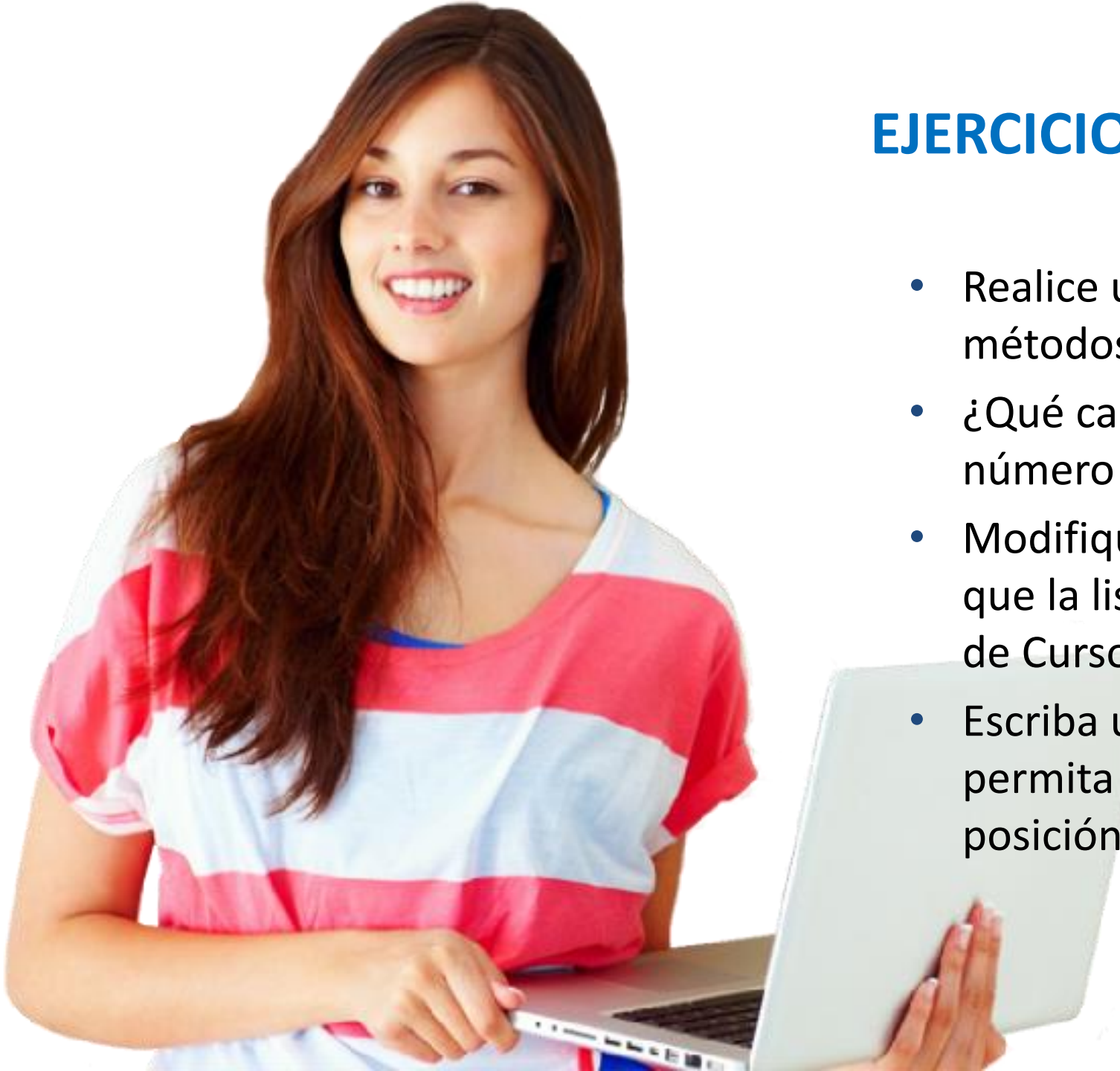
EVALUACIÓN DEL TEMA DESARROLLADO

Reflexionemos!

REVISEMOS LO APRENDIDO

- Podrías transformar el programa de prueba en un menú de opciones.
- Hasta el momento cómo haces para plantear tus algoritmos.
- Describa el algoritmo para realizar una búsqueda
- ¿Qué casos se tiene al momento de eliminar un nodo?
  - Describa los pasos para los diferentes casos
- ¿Cómo puedo incluir en la lista información de un curso, fracción, números, string, etc.?

EJERCICIOS DE APLICACIÓN



EJERCICIOS DE APLICACIÓN

- Realice un menú de opciones con los métodos estudiados.
- ¿Qué cambios se haría para saber el número de elementos de la lista?
- Modifique el programa de tal forma que la lista esté formada por los datos de Cursos (nombre, créditos)
- Escriba un método de tal forma que permita insertar un nodo en una posición cualquiera