

Programación II

Tema 3: Estructuras de datos básicas con memoria dinámica

Departamento de Lenguajes y Sistemas Informáticos
Universitat Jaume I

Curso 2024/2025

EI(MT)1008 – Programación II

1 / 33

Estructuras de datos

Estructura de datos

Representación de datos junto con las operaciones permitidas sobre dichos datos.

- Especificación (ej.: List)
- Implementación (ej.: ArrayList y LinkedList)

EI(MT)1008 – Programación II

3 / 33

Contenido

Listas enlazadas

Pilas

Colas

EI(MT)1008 – Programación II

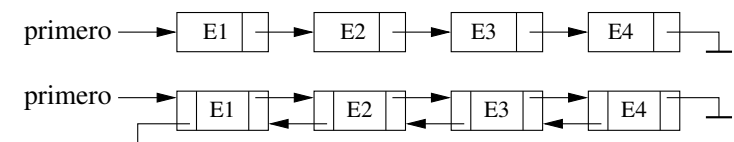
2 / 33

Listas enlazadas

Secuencia de objetos creados dinámicamente (*nodos*), cada uno de los cuales se refiere a:

- su sucesor en la lista (*enlace simple*).
- su sucesor y a su antecesor en la lista (*enlace doble*).

Ejemplo: listas con cuatro elementos

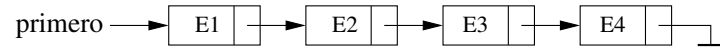


EI(MT)1008 – Programación II

4 / 33

Listas con enlace simple

Ejemplo: lista con cuatro elementos



```
// De momento
class Nodo {
    Tipo dato; // Dato almacenado en el nodo
    Nodo sig; // Referencia al siguiente nodo
}
```

- ▶ Enlace al primer nodo.
- ▶ El acceso a cualquier otro nodo requiere recorrer todos sus antecesores.
- ▶ El sucesor del último nodo es null.

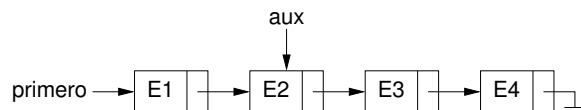
Listas con enlace simple: recorrido de sus elementos

Ejemplo: contar las veces que aparece un elemento en una lista

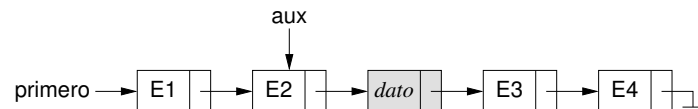
```
// Suponiendo que Tipo es String
int veces = 0;
for (Nodo aux = primero; aux != null; aux = aux.sig)
    if (aux.dato.equals(elemento))
        veces++;
```

Listas con enlace simple: añadir un elemento

Dada la siguiente lista enlazada:



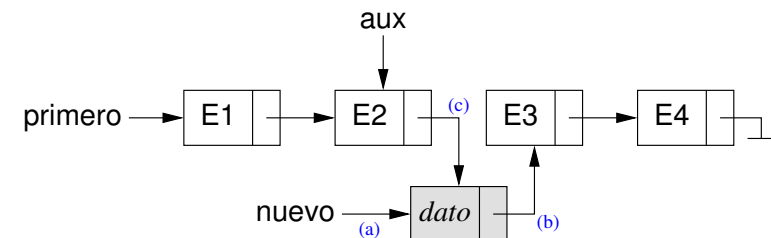
queremos insertar un nuevo nodo tras el nodo apuntado por aux:



Listas con enlace simple: añadir un elemento (cont.)

Ejemplo: inserción tras el nodo apuntado por aux

```
// Suponiendo que aux != null
Nodo nuevo = new Nodo();
nuevo.dato = dato;
nuevo.sig = aux.sig;
aux.sig = nuevo;
```



Listas con enlace simple: añadir un elemento (cont.)

Ejemplo: inserción al final de una lista no vacía

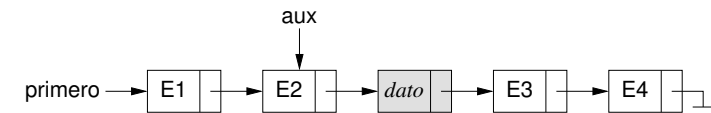
```
// Hacer que aux apunte al último nodo
Nodo aux;
for (aux = primero; aux.sig != null; aux = aux.sig);
// Insertar un nuevo nodo detrás de aux
...
```

Ejemplo: inserción al comienzo de una lista

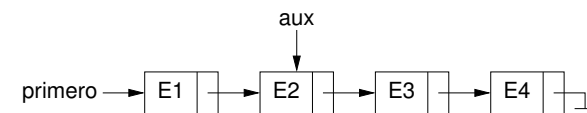
```
Nodo nuevo = new Nodo();
nuevo.dato = dato;
nuevo.sig = primero;
primero = nuevo;
```

Listas con enlace simple: borrar un elemento

Dada la siguiente lista enlazada:



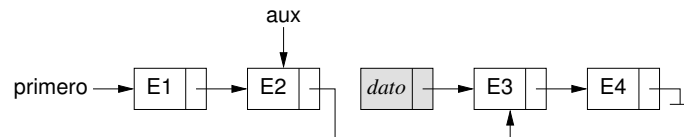
queremos borrar el nodo siguiente del nodo apuntado por aux:



Listas con enlace simple: borrar un elemento (cont.)

Ejemplo: borrado del sucesor del nodo apuntado por aux

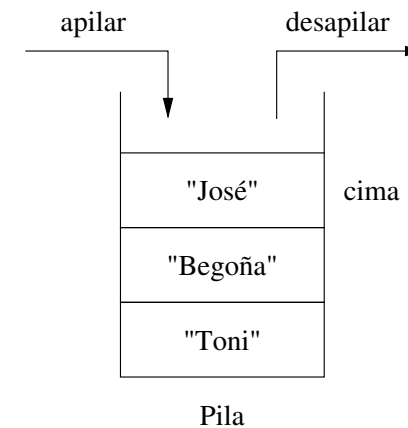
```
// Suponiendo que aux != null y que aux.sig != null
aux.sig = aux.sig.sig;
```



Ejemplo: borrado del primer nodo de una lista no vacía

```
primero = primero.sig;
```

Pilas

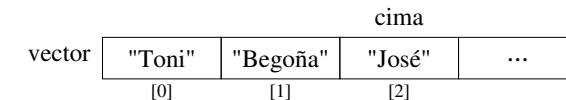


Pila de cadenas: especificación

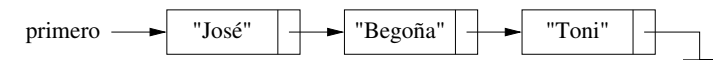
```
public interface PilaString {  
    void apilar(String elemento);  
  
    // Excepción si la pila está vacía  
    String desapilar();  
  
    // Excepción si la pila está vacía  
    String cima();  
  
    boolean esVacía();  
}
```

Pila de cadenas: implementación

- ▶ Usando un vector:



- ▶ Usando nodos:



Pila de cadenas: implementación con un vector

```
public class PilaStringVector implements PilaString {  
    private static final int CAPACIDAD_INICIAL = 10;  
    private String[] vector;  
    private int posCima;  
  
    public PilaStringVector() {  
        vector = new String[CAPACIDAD_INICIAL];  
        posCima = -1;  
    }  
    ...  
}
```

Pila de cadenas: implementación con un vector (cont.)

```
public class PilaStringVector implements PilaString {  
    ...  
    public void apilar(String elemento) {  
        if (posCima == vector.length - 1) // Vector lleno  
            redimensionar();  
        vector[++posCima] = elemento;  
    }  
  
    private void redimensionar() {  
        String[] nuevo = new String[2 * vector.length];  
        for (int i = 0; i < vector.length; i++)  
            nuevo[i] = vector[i];  
        vector = nuevo;  
    }  
    ...  
}
```

Pila de cadenas: implementación con un vector (cont.)

```
public class PilaStringVector implements PilaString {
    ...
    public String desapilar() {
        if (esVacía())
            throw new EmptyStackException();
        String elemento = vector[posCima];
        vector[posCima] = null; // Eliminar referencia antigua
        posCima--;
        return elemento;
    }
    ...
}
```

Pila de cadenas: implementación con un vector (cont.)

```
public class PilaStringVector implements PilaString {
    ...
    public String cima() {
        if (esVacía())
            throw new EmptyStackException();
        return vector[posCima];
    }

    public boolean esVacía() {
        return posCima < 0;
    }
}
```

Pila de cadenas: ejemplo de uso

```
...
String[] nombres = {"Toni", "Begoña", "José"};

PilaString pila = new PilaStringVector();

for (int i = 0; i < nombres.length; i++)
    pila.apilar(nombres[i]);

while (!pila.esVacía())
    System.out.println(pila.desapilar());
...
```

Pila de cadenas: implementación con nodos

```
public class PilaStringNodos implements PilaString {
    private static class Nodo {
        String dato;
        Nodo sig;

        Nodo(String dato, Nodo sig) {
            this.dato = dato;
            this.sig = sig;
        }
    }

    private Nodo primero;

    // Constructor (por defecto)
    ...
}
```

Pila de cadenas: implementación con nodos (cont.)

```
public class PilaStringNodos implements PilaString {
    ...
    public void apilar(String elemento) {
        primero = new Nodo(elemento, primero);
    }

    public String desapilar() {
        if (esVacía())
            throw new EmptyStackException();
        String elemento = primero.dato;
        primero = primero.sig;
        return elemento;
    }
    ...
}
```

Pila de cadenas: implementación con nodos (cont.)

```
public class PilaStringNodos implements PilaString {
    ...
    public String cima() {
        if (esVacía())
            throw new EmptyStackException();
        return primero.dato;
    }

    public boolean esVacía() {
        return primero == null;
    }
}
```

Pila de cadenas: ejemplo de uso

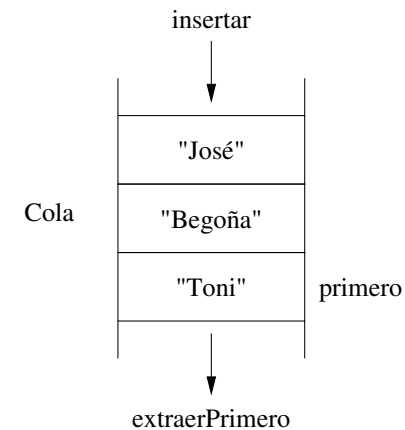
```
...
String[] nombres = {"Toni", "Begoña", "José"};

PilaString pila = new PilaStringNodos();

for (int i = 0; i < nombres.length; i++)
    pila.apilar(nombres[i]);

while (!pila.esVacía())
    System.out.println(pila.desapilar());
...
```

Colas

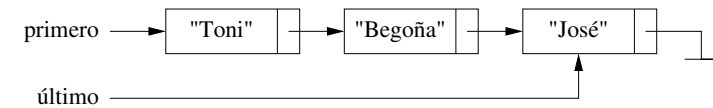


Cola de cadenas: especificación

```
public interface ColaString {  
    void insertar(String elemento);  
  
    // Excepción si la cola está vacía  
    String extraerPrimero();  
  
    // Excepción si la cola está vacía  
    String consultarPrimero();  
  
    boolean esVacía();  
}
```

Cola de cadenas: implementación

- ▶ Usando un vector *circular* y dos índices (no lo estudiaremos)
- ▶ Usando nodos:



Cola de cadenas: implementación con nodos

```
public class ColaStringNodos implements ColaString {  
    private static class Nodo {  
        String dato;  
        Nodo sig;  
  
        Nodo(String dato, Nodo sig) {  
            this.dato = dato;  
            this.sig = sig;  
        }  
    }  
  
    private Nodo primero;  
    private Nodo último;  
  
    // Constructor (por defecto)  
    ...  
}
```

Cola de cadenas: implementación con nodos (cont.)

```
public class ColaStringNodos implements ColaString {  
    ...  
    public void insertar(String elemento) {  
        Nodo nuevo = new Nodo(elemento, null);  
        if (esVacía()) {  
            primero = nuevo;  
            último = nuevo;  
        } else {  
            último.sig = nuevo;  
            último = nuevo;  
        }  
    }  
    ...  
}
```

Cola de cadenas: implementación con nodos (cont.)

```
public class ColaStringNodos implements ColaString {
    ...
    public String extraerPrimero() {
        if (esVacía())
            throw new NoSuchElementException();
        String elemento = primero.dato;
        primero = primero.sig;
        if (primero == null)
            último = null;
        return elemento;
    }
    ...
}
```

Cola de cadenas: implementación con nodos (cont.)

```
public class ColaStringNodos implements ColaString {
    ...
    public String consultarPrimero() {
        if (esVacía())
            throw new NoSuchElementException();
        return primero.dato;
    }

    public boolean esVacía() {
        return primero == null;
    }
}
```

Cola de cadenas: ejemplo de uso

```
...
String[] nombres = {"Toni", "Begoña", "José"};

ColaString cola = new ColaStringNodos();

for (int i = 0; i < nombres.length; i++)
    cola.insertar(nombres[i]);

while (!cola.esVacía())
    System.out.println(cola.extraerPrimero());
...
```