

Miguel Lozano Arvizu

Scribd

Implementación de una Lista Simplemente Enlazada  
con Nodos en Java.

Clase Nodo

```
public class Nodo {  
    private int dato;  
    private Nodo siguiente;
```

```
    public Nodo (int dato) {  
        this.dato = dato;  
        this.siguiente = null;
```

```
    public int getData () {  
        return dato;
```

3/15

```
    public void setData () {  
        this.dato = dato;
```

}

```
    public Nodo getSigiente () {  
        return siguiente;
```

3

```
    public void setSigiente () {  
        this.siguiente = siguiente;
```

3

3

sutta guinea

Defini encapsulamiento en la clase Nodo y que sea genérica con dato genérico.

```
public class Nodo <T> {
```

```
    private T dato;
```

```
    private Nodo <T> siguiente;
```

```
    public Nodo (T dato) {
```

```
        this.dato = dato;
```

```
        this.siguiente = null;
```

```
}
```

// getters y setters

```
    public T getData () {
```

```
        return dato;
```

```
}
```

o/

```
    public void setData (T dato) {
```

```
        this.dato = dato;
```

```
}
```

```
    public Nodo <T> getSiguiente () {
```

```
        return siguiente;
```

```
}
```

```
    public void setSiguiente (Nodo <T> siguiente) {
```

```
        this.siguiente = siguiente;
```

```
}
```

```
}
```

25/10/2017 + Ejemplos: Pintar sobre un tablero de ajedrez con Nodos y vértices. Store datos y solo se acceden a través de métodos (obligatoriamente).

Play off Yo!

Clase listaEnlazada

```
public class ListaEnlazada {
    Nodo cabeza; // Primer nodo - la cabeza
    // Constructor ~
    public ListaEnlazada() {
        this.cabeza = null;
    }
    // Metodos de las operaciones.
```

- Corregido - Metodos -

```
public class ListaEnlazada <T> {
    private Nodo <T> cabeza; // Encapsulamiento
    public ListaEnlazada() { ← // Constructor
        this.cabeza = null;
    }
    public Nodo <T> get Cabeza () {
        return cabeza;
    }
    // Metodos de los operaciones.
    // Agregar
    public void agregar (T dato) {
        Nodo <T> nuevo = new Nodo <> (dato);
        nuevo.siguiente = cabeza;
        cabeza = nuevo;
    }
}
```

// Eliminar

```
public void eliminar (T dato) {  
    if (cabeza == null) {  
        return;  
    }  
    if (cabeza.dato.equals (dato)) {  
        cabeza = cabeza.siguiente;  
        return;  
    }
```

```
Nodo < T > actual = cabeza;  
while (actual.siguiente != null) {  
    if (actual.siguiente == actual.siguiente.siguiente);  
        return;  
    }  
    actual = actual.siguiente;  
}
```

// Imprimir

```
public void imprimir () {  
    Nodo < T > actual = cabeza;  
    while (actual != null) {  
        System.out.println (actual.dato);  
        actual = actual.siguiente;  
    }  
}
```

11 Buscar dato

```
public boolean buscar (T dato) {  
    Nodo<T> actual = cabecera;  
    while (actual != null) {  
        if (actual.dato.equals (dato)) {  
            return true;  
        }  
        actual = actual.siguiente;  
    }  
    return false;  
}
```

3 - Dibujo

3.1 - Agregar al inicio (agregarAlInicio (int dato)).

```
public void agregarAlInicio (int dato) {  
    Nodo nuevo = new Nodo (dato);  
    nuevo.setSiguiente (cabeza);  
    cabeza = nuevo;
```

}

3.2 - Eliminar al Inicio (eliminarAlInicio()).

```
public void eliminarInicio () {  
    if (cabeza != null) {  
        cabeza = cabeza.getSiguiente ();
```

}

}

3.3 Recorrer / Imprimir (imprimirLista())

```
public void imprimirLista () {  
    Nodo actual = cabeza;  
    while (actual != null) {  
        System.out.println (actual.getDato () + " ");  
        actual = actual.getSiguiente ();
```

}

```
    System.out.print ("null");
```

}

#### 4.- Caso Práctico: Gestión de un Historial de Tracciones

```
public class Transacciones {  
    public static void main (String [] args) {  
        // Inicializar lista  
        ListaEnlazada lista = new ListaEnlazada ();  
  
        lista.agregarInicio (150);  
        lista.agregarInicio (45);  
        lista.agregarInicio (200);  
        lista.agregarInicio (75);  
  
        System.out.println ("Lista Actual");  
        lista.imprimir ();  
  
        lista.eliminarInicio ();  
  
        System.out.println ("\nDespués de la cancelación");  
        lista.imprimir ();  
    }  
}
```

- Desafío

```
Public void agregarFinal (int dato) {  
    Nodo nuevo = new Nodo (dato);  
    if  
        (cabeza == null) {  
            cabeza = nuevo;  
            return;  
    }
```

Miguel Lozano Arizmendi GTT0141

S M A

Scribd

(51)

```
Nodo actual = cabeza;
while (actual.getsiguiente() != null) {
    }
    actual.setsiguiente (nuevo);
}
```