

Universidad Tecnológica de Izúcar de Matamoros

Estructura de Datos

Unidad III - Listas

3.2 Tipos de Listas

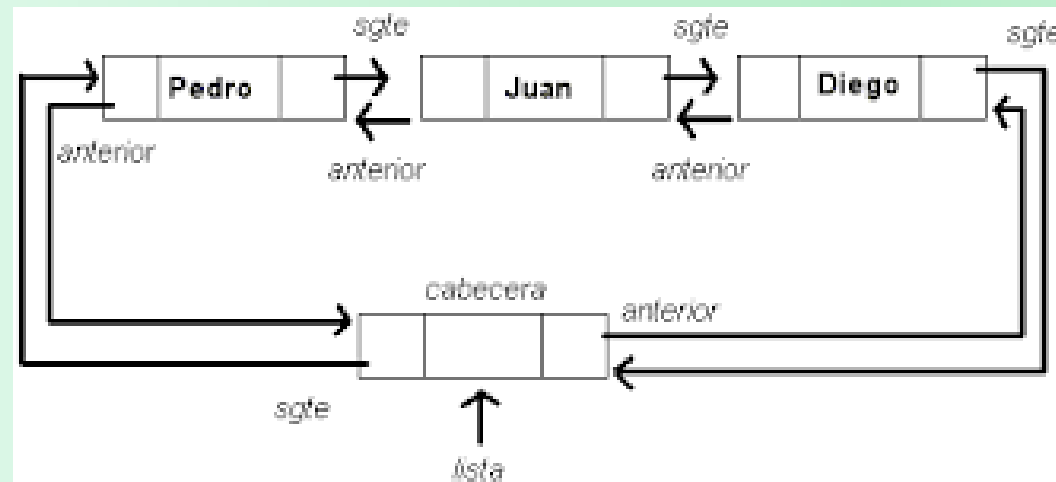


Tabla de Contenido

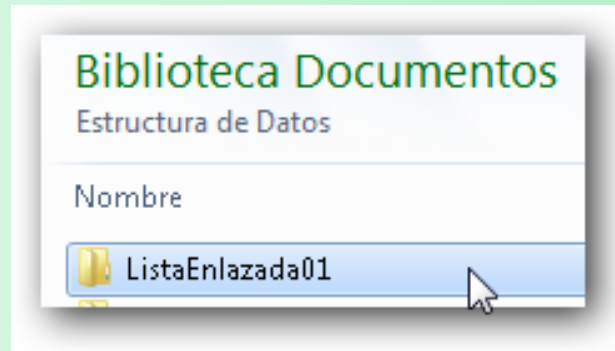
- | | |
|--|--|
| 3. Objetivo | 11. Agregando un Archivo Main Class |
| 4. Creando el Directorio | |
| 5. Abriendo el IDE de Desarrollo | 12. Probando el Funcionamiento |
| 6. Declarando la Clase del Nodo | 13. Comprobamos si la Lista está Vacía |
| 7. Método para Insertar un Objeto | 14. Referencias. |
| 8. Método para Verificar la Lista | |
| 9. Devolver el Primero y el Último de la Lista | |
| 10. Devolver el Último de la Lista | |

Objetivo

- El alumno elaborará programas usando listas para manipular datos de forma organizada.

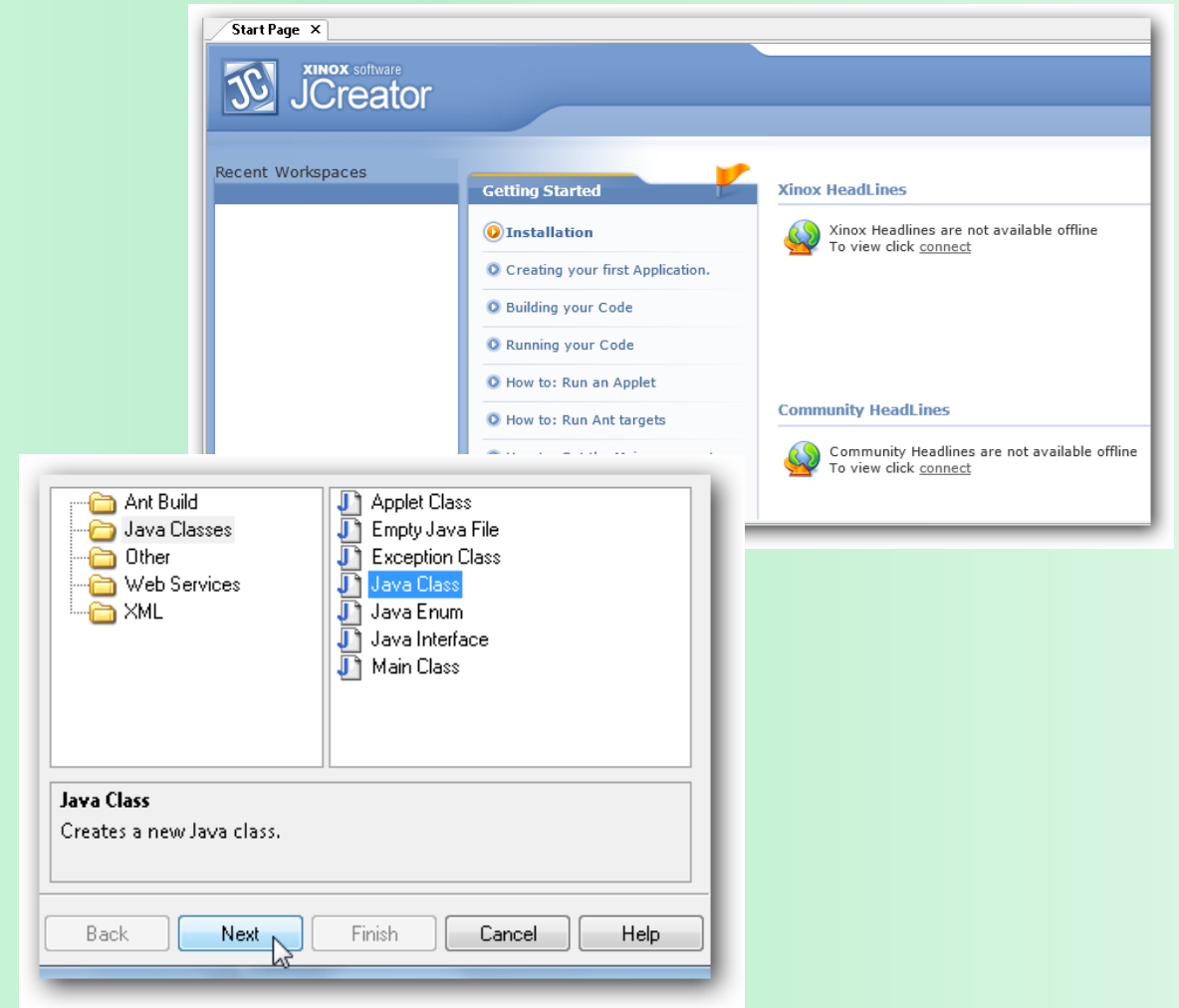
Creando el Directorio

- Crear una carpeta dentro de su directorio de archivos con un nombre identificable (pe. ListaEnlazada).



Abriendo el IDE de Desarrollo

- Abrir nuestro IDE de desarrollo (Jcreator, Netbeans, Eclipse, etc.).
- Creamos una clase Java y presionamos el botón Next.
- Le agregamos el nombre de MiListaEnlazada y agregamos el siguiente código java.



Declarando la Clase del Nodo

```
9 public class MiListaEnlazada {  
10  
11     //Declaramos la clase del nodo.  
12     private class Nodo {  
13         protected Object dato;  
14         protected Nodo sig;  
15  
16         //Declaramos el constructor  
17         public Nodo(){  
18             sig = null;  
19             dato = new Object();  
20         }  
21  
22         //Le pasamos un objeto al nodo  
23         public Nodo(Object p){  
24             sig = null;  
25             dato = p;  
26         }  
27     }
```

```
29         //Atributos  
30         private Nodo primero;
```

- Agregamos el siguiente código Java para declarar la clase con el nombre Nodo.
- Declaramos un constructor con el nombre Nodo.
- Agregamos el código para mandarle un objeto al nodo.
- Agregamos los atributos.

Método para Insertar un Objeto

- Agregamos el siguiente código que permite insertar un objeto en la lista.

```
32 //Inserta un objeto al principio de la lista
33 public void Insertar_Primerio(Object p){
34     Nodo nuevo = new Nodo(p);
35     if(primerio == null){
36         primerio=nuevo;
37     }
38     else{
39         nuevo.sig = primerio;
40         primerio = nuevo;
41     }
42 }
```

Método para Verificar la Lista

- El siguiente método permitirá revisar si la lista está vacía o no

```
44 //Indica si la lista está vacía o no.  
45 public boolean EstaVacía(){  
46     return primero == null;  
47 }
```


Devolver el Primero y el Último de la Lista

- Agregar el siguiente código el cual permitirá devolver el primero de la lista.
- Agregar el siguiente código el cual permitirá devolver el último de la lista.

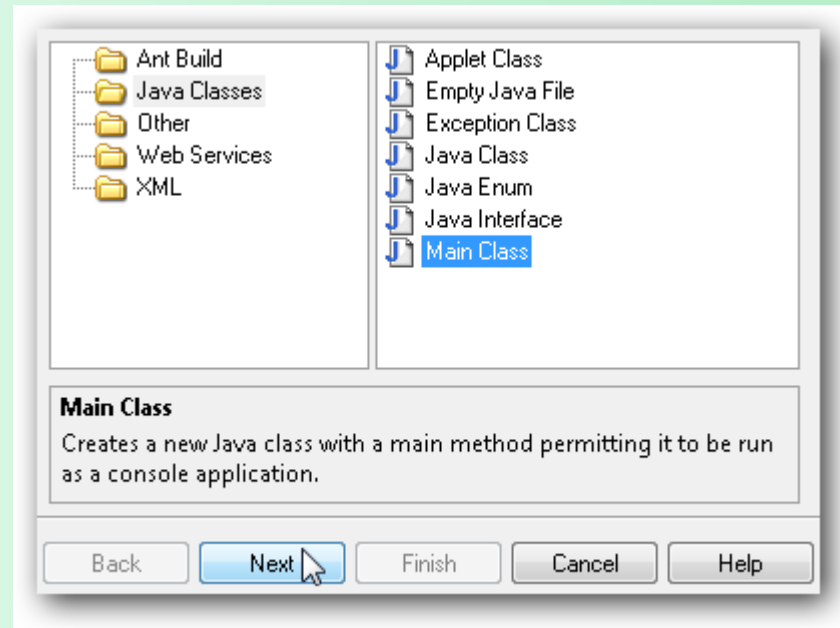
```
49 //Devuelve el primero de la lista.|
50 public Object Primero(){
51     Object elemen = null;
52     if(!EstaVacia()){
53         elemen = (Object) primero.dato;
54     }
55     return elemen;
56 }
```

```
58 //Devuelve el último de la lista.
59 public Object Ultimo(){
60     Object elemen = null;
61     Nodo aux = null;
62     if(!EstaVacia()){
63         aux = primero;
64         while(aux.sig != null){
65             aux = aux.sig;
66             elemen = (Object) aux.dato;
67         }
68     }
69     return elemen;
70 }
```

Nota: Guardamos los cambios y creamos un nuevo archivo.

Devolver el Último de la Lista

- Creamos un nuevo archivo de tipo Main Class y le agregamos un nombre (p.e. MiEjemplo).



Agregando un Archivo Main Class

1. Creamos la lista.
2. Le agregamos valores.
3. Invocamos los métodos Primero y Ultimo.

```
11 public static void main(String[] args) {  
12     //Creamos la lista.  
13     MiListaEnlazada lista = new MiListaEnlazada();  
14  
15     //Añadismo datos  
16     lista.Insertar_Primer("Jonathan");  
17     lista.Insertar_Primer("Oscar");  
18     lista.Insertar_Primer("Guadalupe");  
19     lista.Insertar_Primer("Coraima");  
20  
21     //Mostramos el primer elemento.  
22     System.out.println("El primer elemento es: " + lista.Primer());  
23     System.out.println("El último elemento es: " + lista.Ultimo());  
24  
25 }
```

Probando el Funcionamiento

- Compilamos nuestra lista, en caso de tener errores corregirlos con base en los códigos de las imágenes.
- El resultado debe ser similar al siguiente.



```
General Output
-----Configuration:
El primer elemento es: Coraima
El último elemento es: Jonathan

Process completed.
|
```

Comprobamos si la Lista está Vacía

- Invocamos el método EstaVacia() y lo agregamos dentro de una estructura IF para comprobar, esto devolverá un TRUE si la lista está vacía o un FALSE, si no está vacía.

```
25 //Preguntamos si la lista está vacía.  
26 if(lista.EstaVacia()==true){  
27     System.out.println("La lista está vacía.");  
28 }else{  
29     System.out.println("La lista no está vacía.");  
30 }
```

General Output

```
-----Configuration: <Default>  
El primer elemento es: Coraima  
El último elemento es: Jonathan  
La lista no está vacía.  
  
Process completed.
```

Agregando los Nuevos Métodos

- Método para contar los elementos de la lista.



```
87 //Devolver el número de elementos de la lista.  
88 //Usar un incrementador.  
89 public int Cuantos_Elementos(){  
90     Nodo aux = null;  
91     int i = 0;  
92     aux = primero;  
93     while(aux != null){  
94         aux = aux.sig;  
95         i++;  
96     }  
97     return i;  
98 }
```

```
100 public void mostrar(){  
101     Nodo temp = primero;  
102     if(temp != null){  
103         System.out.println("\nLos datos de la lista son: ");  
104         while(temp != null){  
105             System.out.println(temp.dato.toString() + " ");  
106             temp = temp.sig;  
107         }  
108         //System.out.println("\n");  
109     }  
110 }
```

- Método para mostrar los elementos de la lista.



Invocando los Métodos en el Main

- Dentro del main invocamos el método Cuantos_Elementos() y nos devolverá un número.
- Al invocar el método mostrar(), nos debería reflejar los datos agregados en la lista.

```
33 //Mostramos el total de elementos.  
34 System.out.println("\nEl total de elementos es de: " + lista.Cuantos_Elementos());  
35 //Mostramos los datos de la lista.  
36 lista.mostrar();
```

General Output

El total de elementos es de: 4

Los datos de la lista son:

Coraima

Guadalupe

Oscar

Jonathan

Referencias

- Aho, A., Hopcroft, J., & Ullman, J. (1988). *Estructuras de Datos y Algoritmos*. Delaware: Addison Wesley Iberoamericana.
- Joyanes Aguilar, L., & Zahonero Martínez, I. (2008). *Estructuras de Datos en Java*. España: McGraw-Hill.
- Ureña Gómez, F. (2 de Septiembre de 2014). *Listas enlazadas en java*. Obtenido de <http://www.discoduroderoer.es/listas-enlazadas-en-java/>
- Weiss, M. (2000). *Estructuras de Datos en Java*. España: Addison Wesley.