



	<p align="center">UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		
Aprobación: 2022/03/01	Código: GUIA-PRLE-001	Página: 1

INFORME DE LABORATORIO

INFORMACIÓN BÁSICA					
ASIGNATURA:	Estructuras de datos y algoritmos				
TÍTULO DE LA PRÁCTICA:	<i>Revisión de elementos de programación (Parte I)</i>				
NÚMERO DE PRÁCTICA:	<i>02</i>	AÑO LECTIVO:	<i>2023-A</i>	NRO. SEMESTRE:	<i>III</i>
FECHA DE PRESENTACIÓN	<i>21/05/2022</i>	HORA DE PRESENTACIÓN			
INTEGRANTE (s): <ul style="list-style-type: none"> - Borda Espinoza Gabriela - Condori Pinto Juan José - Gordillo Mendoza, Jose Alonzo - Lopez Arela, Ower Frank 				NOTA:	
DOCENTE(s): <i>Mg. Edith Giovanna Cano Mamani</i>					

SOLUCIÓN Y RESULTADOS
<p>1. SOLUCIÓN DE EJERCICIOS/PROBLEMAS</p> <p>REPOSITORIO: https://github.com/jcondoripin/EdaC_lab2</p> <ul style="list-style-type: none"> Invertir un matriz de enteros (2 pts): <p>Ejemplo: A=[1 2 3] -> Ain=[3 2 1]</p> <pre>public int[] invertirArray(int[] A){ /** */ //Procedimiento para invertir la matriz /** */ return Ain; }</pre>

	<p align="center">UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		
Aprobación: 2022/03/01	Código: GUIA-PRLE-001	Página: 2

Get Started
EJERCICIO01.JAVA

home > usuario > Descargas > EJERCICIO01.JAVA

```

1 public class Main {
2
3     public static void main(String[] args) {
4
5         int array[] = {1, 2, 3};
6
7         int invertido[] = new int[array.length];
8
9         for (int i = invertido.length - 1, j = 0; i >= 0; i--, j++) {
10             invertido[i] = array[j];
11         }
12
13         for (int i = 0; i < invertido.length; i++) {
14             System.out.println(invertido[i]);
15         }
16     }
17 }



```

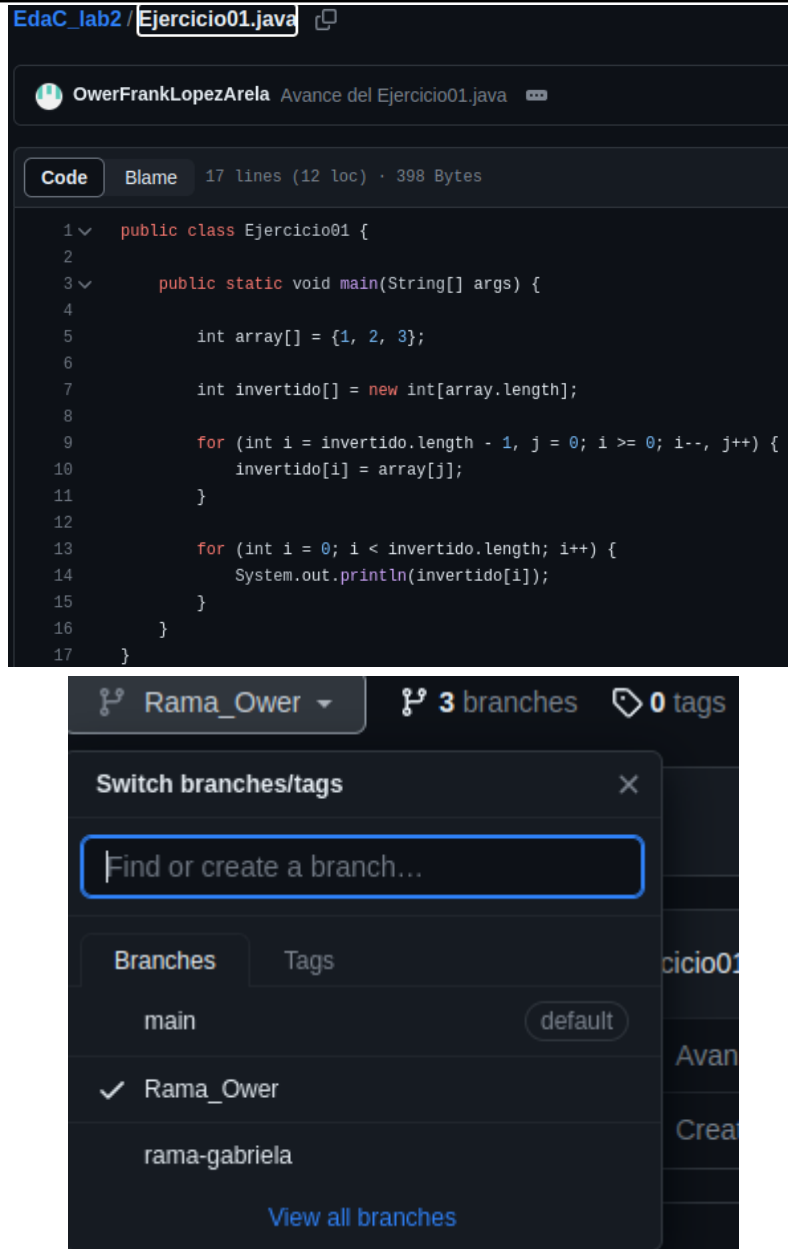
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```

• usuario@usuario-ThinkCentre-M93p:~$ ls
03-hola-al- CarpetaEjercicios Descargas Documentos ejemplo.txt Escritorio Imágenes Música my_project Pharo Plantillas Público snap Videos workspace
• usuario@usuario-ThinkCentre-M93p:~$ cd Descargas/
• usuario@usuario-ThinkCentre-M93p:~/Descargas$ java EJERCICIO01.JAVA
3
2
1
• usuario@usuario-ThinkCentre-M93p:~/Descargas$

```

	<p align="center">UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
<p align="center">Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLE-001</p>	<p>Página: 3</p>



```

1 public class Ejercicio01 {
2
3     public static void main(String[] args) {
4
5         int array[] = {1, 2, 3};
6
7         int invertido[] = new int[array.length];
8
9         for (int i = invertido.length - 1, j = 0; i >= 0; i--, j++) {
10             invertido[i] = array[j];
11         }
12
13         for (int i = 0; i < invertido.length; i++) {
14             System.out.println(invertido[i]);
15         }
16     }
17 }

```

- **Rotación a la Izquierda (3 pts)**

Ejemplo:

Si d=2

A=[1 2 3 4 5] -> Aiz=[3 4 5 1 2]

```

public int[] rotarIzquierdaArray(int[] A){
    /** */
    //Procedimiento para rotar la matriz
    /** */
    return Aiz;
}

```

Código:

```
package ejercicio02;
import java.util.*;

public class ejercicio02 {

    public static ArrayList<Integer> A = new ArrayList<Integer>();
    public static ArrayList<Integer> Aiz = new ArrayList<Integer>();
    public static int d = 0;

    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);


        crearMatriz(A);
        System.out.print("La matriz original es:\nA = [");
        mostrarMatriz(A);

        System.out.print("Introduzca un número menor a "+ A.size() +": ");
        d = scan.nextInt();
        Aiz = rotarIzquierdaArray(A);

        System.out.print("La matriz invertida es: \nAiz = [");
        mostrarMatriz(Aiz);
    }

    public static ArrayList<Integer> crearMatriz(ArrayList<Integer> matriz) {
        int length = (int)(Math.random()*8 +3);
        Integer aux;

        for (int i = 0; i < length; i++) {
            aux = (int)(Math.random()*10 +1);
            matriz.add(aux);
        }
        return matriz;
    }
}
```



	<p>UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
<p>Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLE-001</p>	<p>Página: 5</p>

```
public static ArrayList<Integer> rotarIzquierdaArray(ArrayList<Integer> A) {
    int aux = d;
    for (int i = aux; i < A.size(); i++) {
        Aiz.add(A.get(aux));
        aux++;
    }
    for (int z = 0; z < d; z++) {
        Aiz.add(A.get(z));
    }
    return Aiz;
}
```

```
public static void mostrarMatriz(ArrayList<Integer> matriz) {
    for (int i = 0; i < matriz.size(); i++) {
        System.out.print(matriz.get(i) + " ");
    }
    System.out.println("]");
}
}
```

Ejecución:

```
PS C:\Gabriela\2023A\EDA - LAB\LAB02> & 'C:\Program Files\Java\jdk-
X:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\isaac\AppData
3f7cef7b1ceede0b9ba4f49501cc5\redhat.java\jdt_ws\LAB02_23555b53\bin'
La matriz original es:
A = [5 4 8 2 4 9 6 3 ]
Introduzca un número menor a 8: 3
La matriz invertida es:
Aiz = [2 4 9 6 3 5 4 8 ]
PS C:\Gabriela\2023A\EDA - LAB\LAB02> |
```

	<p>UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
<p>Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLE-001</p>	<p>Página: 6</p>

- **Triángulo recursivo (4 pts)**

Ejemplo:

b=5

*

**

```
public void trianguloRecursivo(int base){
```

```
/** */
```

```
//Procedimiento para imprimir triángulo
```


```
/** */
```

```
}
```

Código:

```
1  import java.util.Scanner;
2  public class ejercicio04 {
3      public static void main(String[] args) {
4          Scanner sc=new Scanner (System.in);
5          System.out.println("Ingresa la base del triangulo a imprimir: ");
6          int base = sc.nextInt();
7          trianguloRecursivo(base);
8      }
9
10     public static void trianguloRecursivo(int b) {
11         if (b > 0) {
12             trianguloRecursivo(b - 1); //uso del metodo recursivamente, disminuyendo la base
13             /*Como vemos se pone antes el metodo para que asi salga escalonadamente de manera
14             descendente como se nos pide en el ejercicio*/
15             for (int i = 0; i < b; i++) {
16                 System.out.print("*");//Impresion de asteriscos en cada nivel
17             }
18             System.out.println();//Salto de linea
19         }
20     }
21 }
```

Ejecución:

	<p>UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
<p>Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLE-001</p>	<p>Página: 7</p>

```
PS C:\Users\user\Desktop\EDA lab02\EdaC_lab2> javac ejercicio04.java
PS C:\Users\user\Desktop\EDA lab02\EdaC_lab2> java .\ejercicio04.java
Ingresa la base del triangulo a imprimir:
6
*
**
***
****
*****
*****
PS C:\Users\user\Desktop\EDA lab02\EdaC_lab2> java .\ejercicio04.java
Ingresa la base del triangulo a imprimir:
3
*
**
***
```

- **Lista (11 pts)**

Implementa una Lista usando POO con tipos genéricos siguiendo los estándares de Java. (Los métodos para una lista).

<https://docs.oracle.com/javase/7/docs/api/java/util/List.html>

Puede ignorar los siguientes métodos:

```
hashCode()
iterator()
listIterator()
listIterator(int index)
retainAll(Collection<?> c)
toArray()
toArray(T[] a)
```

(i) Implemente una clase Node donde T es un tipo genérico, esta clase debe contener al menos dos propiedades.

<https://docs.oracle.com/javase/tutorial/java/generics/types.html>

T data: la información almacenada en el nodo

Node<T> nextNode: una referencia al siguiente nodo

Clase Nodo<T>:

```
public class Node<T> {  
    private T data;  
    private Node<T> nextNode;  
    public Node(T data, Node<T> nextNode) {  
        this.data = data;  
        this.nextNode = nextNode;  
    }  
    public Node(T data) {  
        this.data = data;  
        this.nextNode = null;  
    }  
    public T getData() {  
        return this.data;  
    }  
    public void setData(T data) {  
        this.data = data;  
    }  
    public boolean hasNext() {  
        if(this.nextNode != null)  
            return true;  
        return false;  
    }  
    public Node<T> getNext() {  
        return this.nextNode;  
    }  
    public void setNext(Node<T> nextNode) {  
        this.nextNode = nextNode;  
    }  
}
```

(ii) Implementar una clase List esta clase debe contener al menos esta propiedad
Node<T> root: la referencia sobre el nodo inicial

Clase ListNode:

```
public class ListNode<T>{  
    private Node<T> root;
```



```
public ListNode(Node<T> root) {
    this.root = root;
}

public ListNode() {
    this.root = null;
}

public Node<T> getRoot() {
    return this.root;
}

@Override
public String toString() {
    Node<T> aux = this.root;
    String string = "";
    while(aux != null) {

        string = string + "[" + aux.getData() + "] => ";

        aux = aux.getNext();
    }
    string = string + "NULL";
    return string;
}

// Métodos de lista
public void add(T elem) { // Añade al final
    Node<T> e = new Node<T>(elem);
    if(root == null) {
        this.root = e;
    } else {
        Node<T> aux = root;
        while(aux.hasNext()) {
            aux = aux.getNext();
        }
        aux.setNext(e);
    }
}

public void add(int index, T elem) { // Añade en una posición
    Node<T> e = new Node<T>(elem);
    if(index == 0) { // Inserción al inicio
```

```
e.setNext(this.root);
this.root = e;
} else { // Interno de la lista
    Node<T> aux = this.root;
    int pos = 1;
    while(aux.hasNext() && pos < index) {
        aux = aux.getNext();
        pos++;
    }
    if(pos != index) {
        System.out.println("Index not found");
    } else {
        e.setNext(aux.getNext());
        aux.setNext(e);
    }
}
}

public void clear() {
    this.root = null;
}

public boolean contains(Object o) {
    Node<T> aux = this.root;
    while(aux.hasNext()) {
        aux = aux.getNext();
        if(o.equals(aux.getData())) {
            return true;
        }
    }
    return false;
}

public T get(int index) {
    Node<T> aux = this.root;
    int pos = 0;
    while(aux.hasNext() && pos < index) {
        aux = aux.getNext();
        pos++;
    }
    if(pos == index) {
```

```
        return aux.getData();
    } else {
        System.out.println("Index not found");
        return null;
    }
}

public int size() {
    Node<T> aux = root;
    int siz = 0;
    while(aux.hasNext()) {
        aux = aux.getNext();
        siz++;
    }
    return siz;
}

public boolean isEmpty() {
    return this.root == null;
}

public T remove(int index) {
    if(this.isEmpty()) {
        return null;
        // Arrojar excepcion
    } else {
        Node<T> aux = this.root;
        if(index == 0) {
            this.root = aux.getNext();
            return aux.getData();
        } else {
            int pos = 1;
            while(aux.hasNext() && pos < index) {
                aux = aux.getNext();
                pos++;
            }
            Node<T> aux1 = aux.getNext();
            aux.setNext(aux1.getNext());
            return aux1.getData();
        }
    }
}
```

```
}

public boolean remove(Object o) {
    if(this.isEmpty()) {
        // Underflow error
        return false;
    } else {
        Node<T> aux = this.root;
        if(aux.equals(o)) {
            this.root = aux.getNext();
            return true;
        } else {
            while(aux.hasNext()) {
                if(aux.getNext().equals(o)) {
                    aux.setNext(aux.getNext().getNext());
                    return true;
                }
                aux = aux.getNext();
            }
            return false;
        }
    }
}

public static void main(String[] args) {
    ListNode<Integer> a = new ListNode<>();
    a.add(1);
    a.add(2);
    a.add(3);

    // [1] => [2] => [3] => NULL
    System.out.println(a);
}
}
```

Ejecución:

	<p style="text-align: center;">UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
<p style="text-align: center;">Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLE-001</p>	<p>Página: 13</p>

```
PS C:\Users\HP\Desktop\UNSA\AÑO 2 - SEMESTRE 3\EDA_LAB> & 'C:\Program Files\Java\jdk-18.0.1\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\HP\AppData\Roaming\Code\User\workspaceStorage\382730ee4190bf30ffd32eeba5321b88\re-dhat.java\jdt_ws\EDA_LAB_cab6e1a1\bin' 'ListNode'
[1] => [2] => [3] => NULL
PS C:\Users\HP\Desktop\UNSA\AÑO 2 - SEMESTRE 3\EDA_LAB>
```

2. SOLUCIÓN DEL CUESTIONARIO

- **¿Qué diferencia hay entre un List y un ArrayList en Java?**

En primer lugar list es una interfaz en Java que define el comportamiento general de una lista, mientras que ArrayList es una clase concreta que implementa la interfaz List. Esto significa que ArrayList proporciona una implementación específica de los métodos definidos en la interfaz List. Puedes crear una instancia de ArrayList y asignarla a una variable de tipo List para aprovechar la flexibilidad y el polimorfismo de las interfaces.

También en términos de eficiencia en el acceso a los elementos, ArrayList es más eficiente que algunas otras implementaciones de List, como LinkedList.

Por tanto, List es una interfaz que define el comportamiento general de una lista en Java (siendo un “molde”), mientras que ArrayList es una implementación específica de esa interfaz que proporciona un arreglo dinámico para almacenar los elementos.

- **¿Qué beneficios y oportunidades ofrecen las clases genéricas en Java?**

Las clases genéricas permiten escribir un único código que puede ser utilizado con diferentes tipos de datos. Esto promueve la reutilización del código.

Al utilizar clases genéricas, se evita el uso excesivo de conversiones de tipo (casting) y se optimiza el rendimiento del código, ya que no es necesario realizar verificaciones o conversiones innecesarias.

Las clases genéricas permiten una programación más modular y extensible, ya que se pueden definir clases genéricas que se adapten a diferentes tipos de datos. Esto facilita la creación de componentes genéricos y reutilizables en diversos contextos.

3. CONCLUSIONES

- En conclusión, el uso de List, ArrayList y clases genéricas en Java ofrece una manera flexible y eficiente de trabajar con colecciones de objetos. Proporciona modularidad, reutilización de código y seguridad en tiempo de compilación. La interfaz List define una abstracción común para trabajar con colecciones ordenadas, mientras que ArrayList es una implementación específica que ofrece acceso rápido y operaciones eficientes. Las clases genéricas permiten crear estructuras de datos y algoritmos que funcionen con diferentes tipos de objetos, evitando errores de tipo en tiempo de ejecución.

	<p style="text-align: center;">UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
<p style="text-align: center;">Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLE-001</p>	<p>Página: 14</p>

- En conjunto, estas características hacen de List, ArrayList y las clases genéricas herramientas fundamentales en el desarrollo de aplicaciones Java.

RETROALIMENTACIÓN GENERAL

REFERENCIAS Y BIBLIOGRAFÍA

- <https://www.w3schools.com/java/>
- <https://www.eclipse.org/downloads/packages/release/2022-03/r/eclipse-ide-enterprise-javaand-web-developers>
- <https://docs.oracle.com/javase/7/docs/api/java/util/List.html>
- <https://docs.oracle.com/javase/tutorial/java/generics/types.html>