

**UNIVERSIDAD NACIONAL DE SAN AGUSTÍN DE AREQUIPA**

**FACULTAD DE PRODUCCION Y SERVICIOS**

**ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS**



**Curso:** Laboratorio de Análisis y Diseño de Algoritmos

## **Práctica 2**

**Estudiante:** Tapara Quispe, Fabiola Grissel

**Docente:** Alex Josue Florez Farfan

**Sección:** “B”

Arequipa - Perú

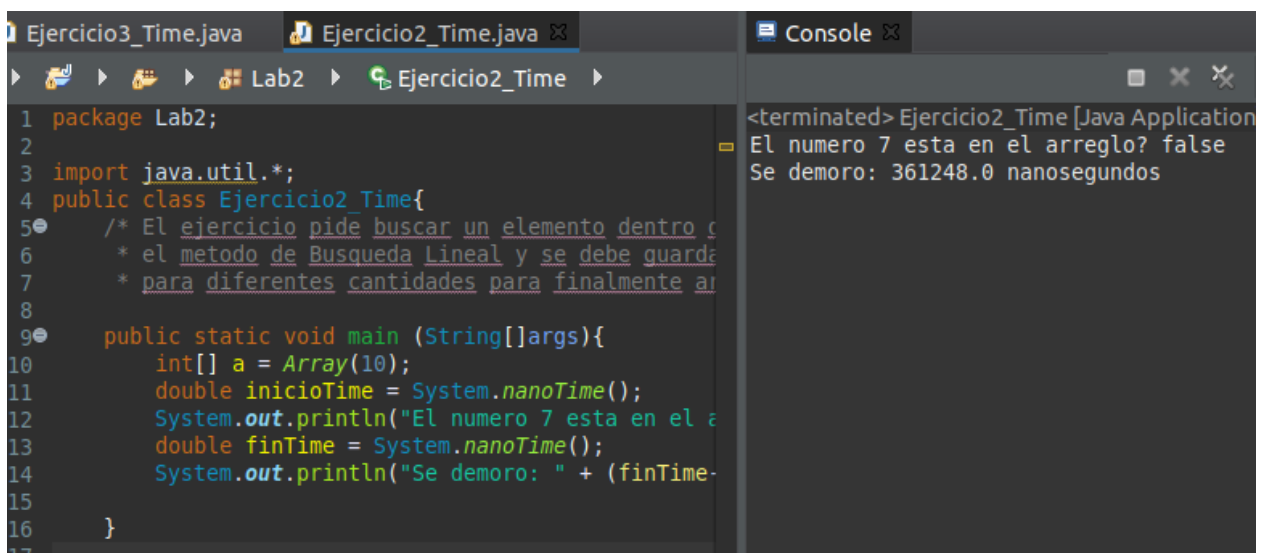
Setiembre 2021

El presente informe complementa los ejercicios 2 y 3 agregando un medidor de tiempo

## Ejercicio2\_Timer

El ejercicio pide buscar un elemento dentro de un arreglo de enteros usando el método de Búsqueda Lineal y se debe guardar los tiempos de ejecuciones para diferentes cantidades para finalmente armar una gráfica de su crecimiento usando nanosegundos.

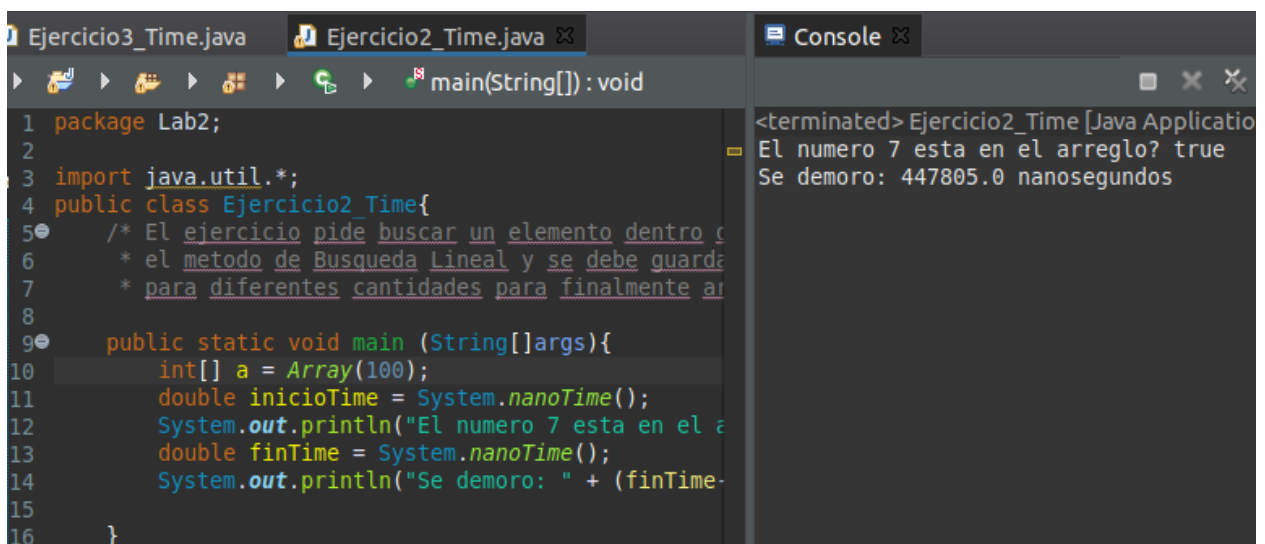
Figura1 Probando con un array de tamaño 10



```
1 package Lab2;
2
3 import java.util.*;
4 public class Ejercicio2_Timer{
5     /* El ejercicio pide buscar un elemento dentro de
6      * el metodo de Busqueda Lineal y se debe guardar
7      * para diferentes cantidades para finalmente ar
8
9     public static void main (String[]args){
10         int[] a = Array(10);
11         double inicioTime = System.nanoTime();
12         System.out.println("El numero 7 esta en el a
13         double finTime = System.nanoTime();
14         System.out.println("Se demoro: " + (finTime-
15
16     }
```

```
<terminated> Ejercicio2_Time [Java Application
El numero 7 esta en el arreglo? false
Se demoro: 361248.0 nanosegundos
```

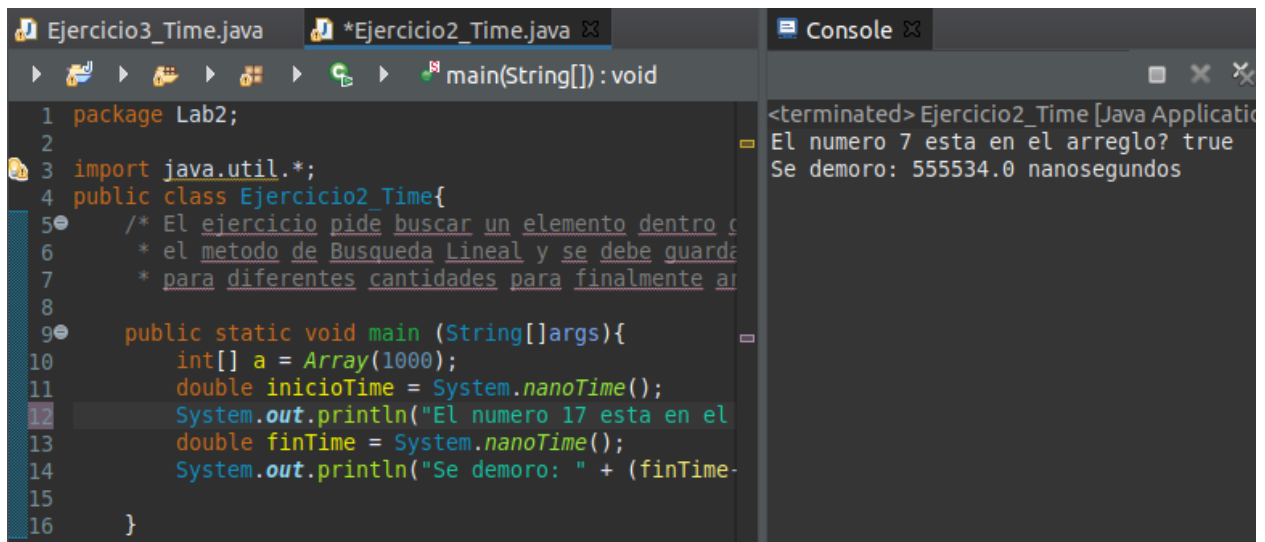
Figura 2 Probando con un array de tamaño 100



```
1 package Lab2;
2
3 import java.util.*;
4 public class Ejercicio2_Timer{
5     /* El ejercicio pide buscar un elemento dentro de
6      * el metodo de Busqueda Lineal y se debe guardar
7      * para diferentes cantidades para finalmente ar
8
9     public static void main (String[]args){
10         int[] a = Array(100);
11         double inicioTime = System.nanoTime();
12         System.out.println("El numero 7 esta en el a
13         double finTime = System.nanoTime();
14         System.out.println("Se demoro: " + (finTime-
15
16     }
```

```
<terminated> Ejercicio2_Time [Java Applicatio
El numero 7 esta en el arreglo? true
Se demoro: 447805.0 nanosegundos
```

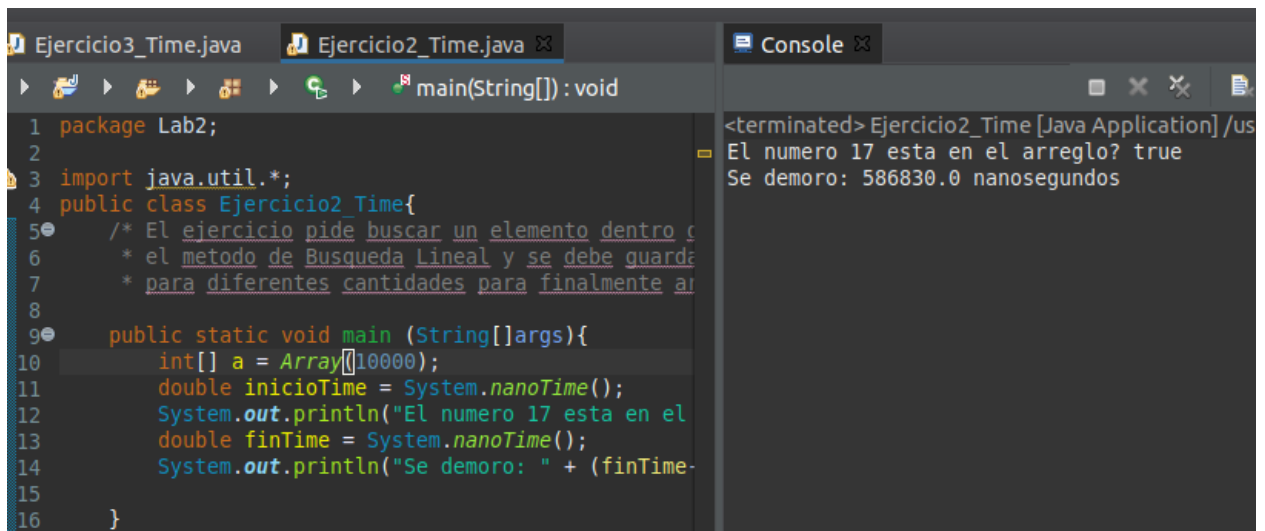
Figura 3 Probando con un array de tamaño 1000



```
1 package Lab2;
2
3 import java.util.*;
4 public class Ejercicio2_Time{
5     /* El ejercicio pide buscar un elemento dentro de un array
6      * el metodo de Busqueda Lineal y se debe guardar el tiempo
7      * para diferentes cantidades para finalmente imprimirlo
8
9     public static void main (String[]args){
10         int[] a = Array(1000);
11         double inicioTime = System.nanoTime();
12         System.out.println("El numero 17 esta en el arreglo? true");
13         double finTime = System.nanoTime();
14         System.out.println("Se demoro: " + (finTime - inicioTime));
15     }
16 }
```

<terminated> Ejercicio2\_Time [Java Application] /usr/bin/java -Djava.library.path=... -jar ...  
El numero 7 esta en el arreglo? true  
Se demoro: 555534.0 nanosegundos

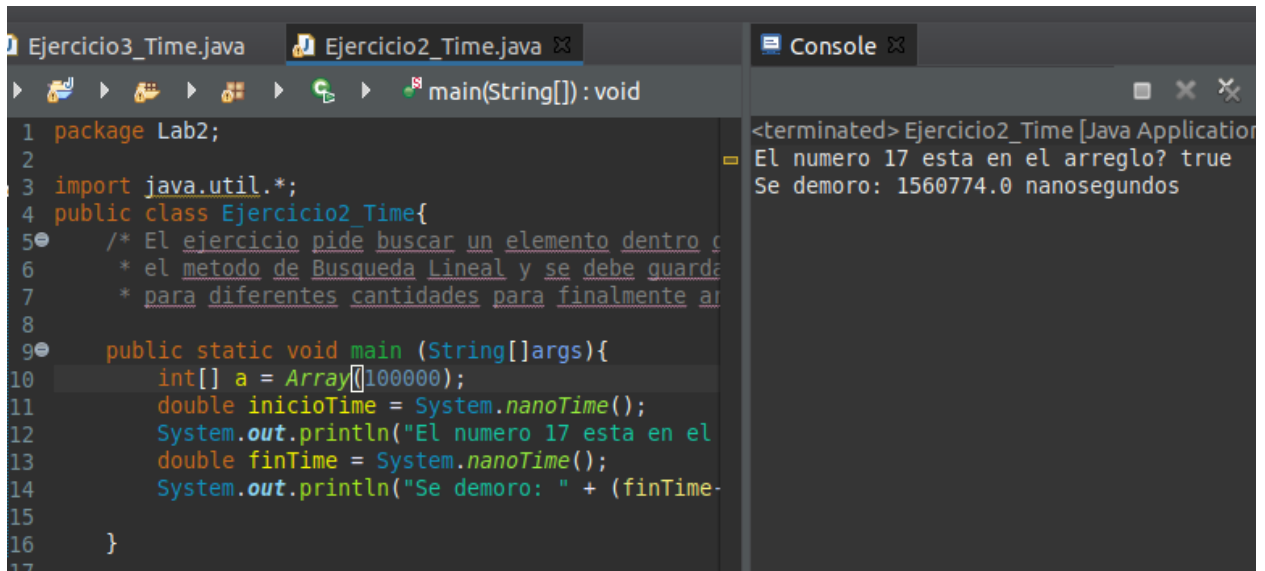
Figura 4 Probando con un array de tamaño 10000



```
1 package Lab2;
2
3 import java.util.*;
4 public class Ejercicio2_Time{
5     /* El ejercicio pide buscar un elemento dentro de un array
6      * el metodo de Busqueda Lineal y se debe guardar el tiempo
7      * para diferentes cantidades para finalmente imprimirlo
8
9     public static void main (String[]args){
10         int[] a = Array(10000);
11         double inicioTime = System.nanoTime();
12         System.out.println("El numero 17 esta en el arreglo? true");
13         double finTime = System.nanoTime();
14         System.out.println("Se demoro: " + (finTime - inicioTime));
15     }
16 }
```

<terminated> Ejercicio2\_Time [Java Application] /usr/bin/java -Djava.library.path=... -jar ...  
El numero 17 esta en el arreglo? true  
Se demoro: 586830.0 nanosegundos

Figura 5 Probando con un array de tamaño 10^5



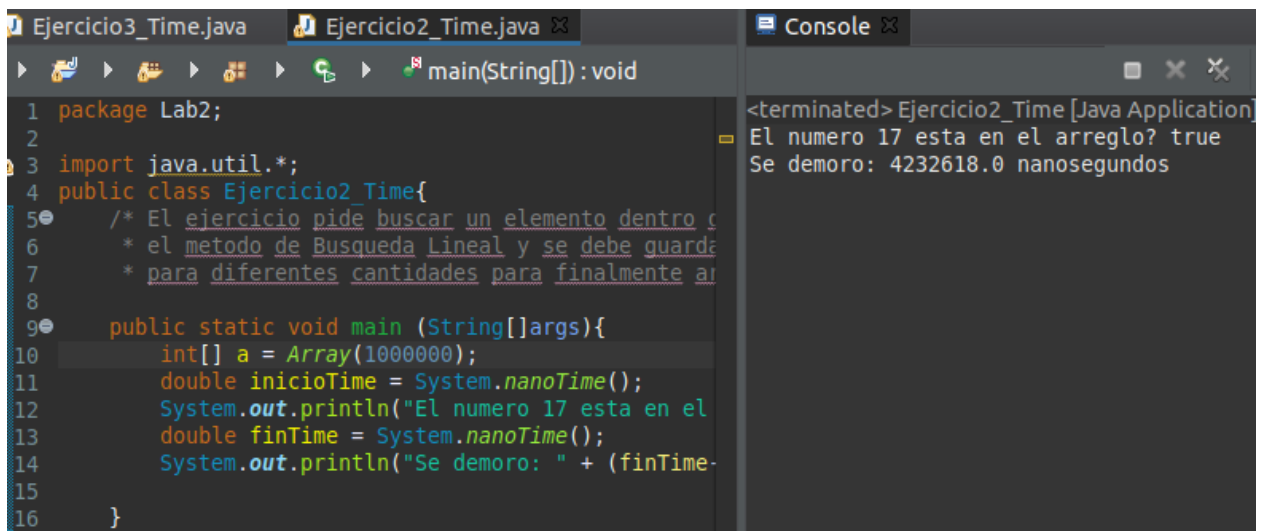
The screenshot shows an IDE with two tabs: Ejercicio3\_Time.java and Ejercicio2\_Time.java. The active tab is Ejercicio2\_Time.java, which contains the following code:

```
1 package Lab2;
2
3 import java.util.*;
4 public class Ejercicio2_Time{
5     /* El ejercicio pide buscar un elemento dentro de un array
6      * el metodo de Busqueda Lineal y se debe guardar el tiempo
7      * para diferentes cantidades para finalmente imprimirlo
8      */
9     public static void main (String[] args){
10         int[] a = Array(100000);
11         double inicioTime = System.nanoTime();
12         System.out.println("El numero 17 esta en el arreglo? true");
13         double finTime = System.nanoTime();
14         System.out.println("Se demoro: " + (finTime - inicioTime));
15     }
16 }
```

The console output shows the following text:

```
<terminated> Ejercicio2_Time [Java Application]
El numero 17 esta en el arreglo? true
Se demoro: 1560774.0 nanosegundos
```

Figura 4 Probando con un array de tamaño  $10^6$



The screenshot shows an IDE with two tabs: Ejercicio3\_Time.java and Ejercicio2\_Time.java. The active tab is Ejercicio2\_Time.java, which contains the following code:

```
1 package Lab2;
2
3 import java.util.*;
4 public class Ejercicio2_Time{
5     /* El ejercicio pide buscar un elemento dentro de un array
6      * el metodo de Busqueda Lineal y se debe guardar el tiempo
7      * para diferentes cantidades para finalmente imprimirlo
8      */
9     public static void main (String[] args){
10         int[] a = Array(1000000);
11         double inicioTime = System.nanoTime();
12         System.out.println("El numero 17 esta en el arreglo? true");
13         double finTime = System.nanoTime();
14         System.out.println("Se demoro: " + (finTime - inicioTime));
15     }
16 }
```

The console output shows the following text:

```
<terminated> Ejercicio2_Time [Java Application]
El numero 17 esta en el arreglo? true
Se demoro: 4232618.0 nanosegundos
```

Figura 4 Probando con un array de tamaño  $10^7$

```

1 package Lab2;
2
3 import java.util.*;
4 public class Ejercicio2_Time{
5     /* El ejercicio pide buscar un elemento dentro de un array
6      * el metodo de Busqueda Lineal y se debe guardar el tiempo
7      * para diferentes cantidades para finalmente armar una tabla
8
9     public static void main (String[]args){
10         int[] a = Array(10000000);
11         double inicioTime = System.nanoTime();
12         System.out.println("El numero 17 esta en el arreglo? true");
13         double finTime = System.nanoTime();
14         System.out.println("Se demoro: " + (finTime - inicioTime));
15     }
16 }

```

```

<terminated> Ejercicio2_Time [Java Application]
El numero 17 esta en el arreglo? true
Se demoro: 1.1023647E7 nanosegundos

```

Figura 4 Probando con un array de tamaño  $10^8$

```

1 package Lab2;
2
3 import java.util.*;
4 public class Ejercicio2_Time{
5     /* El ejercicio pide buscar un elemento dentro de un array
6      * el metodo de Busqueda Lineal y se debe guardar el tiempo
7      * para diferentes cantidades para finalmente armar una tabla
8
9     public static void main (String[]args){
10         int[] a = Array(100000000);
11         double inicioTime = System.nanoTime();
12         System.out.println("El numero 17 esta en el arreglo? true");
13         double finTime = System.nanoTime();
14         System.out.println("Se demoro: " + (finTime - inicioTime));
15     }
16 }

```

```

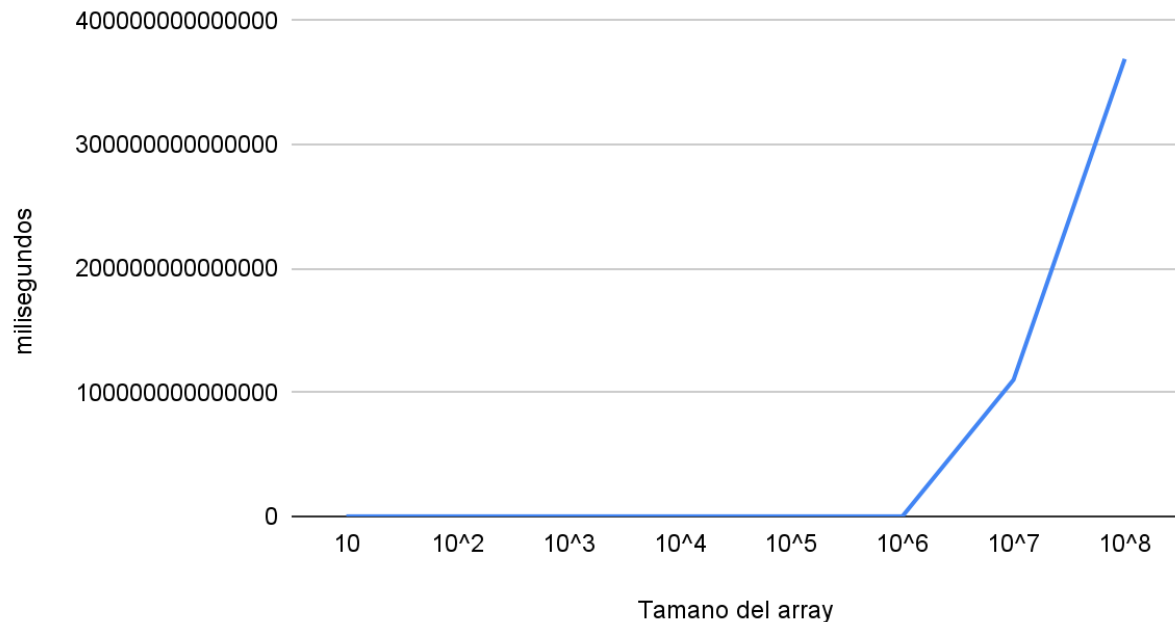
<terminated> Ejercicio2_Time [Java Application]
El numero 17 esta en el arreglo? true
Se demoro: 3.6910924E7 nanosegundos

```

En resumen se armó la siguiente tabla:

Tamano del array	nanosegundos
10	361248
$10^2$	447805
$10^3$	555534
$10^4$	586830
$10^5$	1560774
$10^6$	4232618
$10^7$	1,10E+14
$10^8$	3,69E+14

## Nanosegundos frente a Tamano del array



### Ejercicio3\_Timer

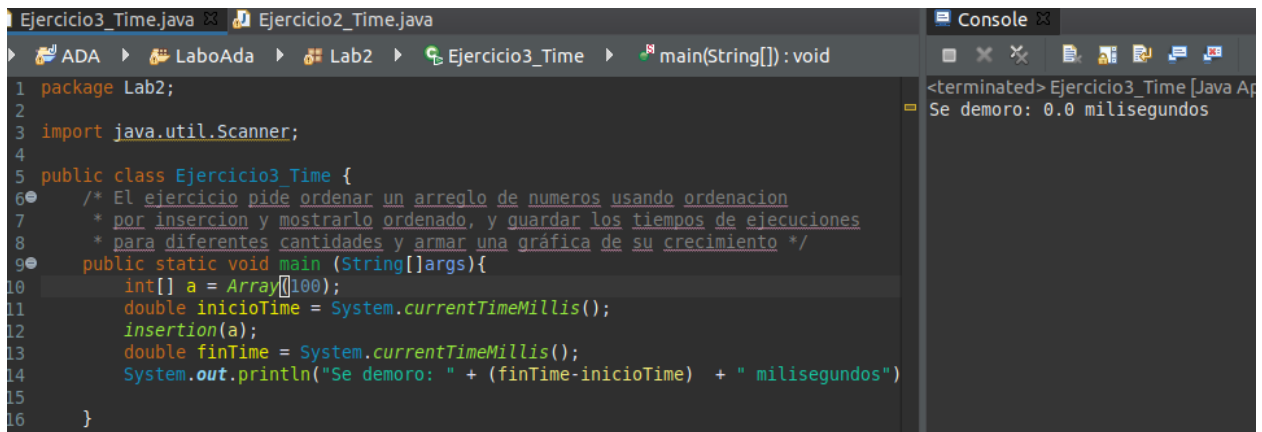
El ejercicio pide ordenar un arreglo de números usando ordenación por inserción y mostrarlo ordenado, y guardar los tiempos de ejecuciones para diferentes cantidades y armar una gráfica de su crecimiento.

Figura 5 Probando con un array de tamaño  $10^1$

```
1 package Lab2;
2
3 import java.util.Scanner;
4
5 public class Ejercicio3_Time {
6     /* El ejercicio pide ordenar un arreglo de numeros usando ordenacion
7      * por insercion y mostrarlo ordenado, y guardar los tiempos de ejecuciones
8      * para diferentes cantidades y armar una gráfica de su crecimiento */
9     public static void main (String[] args){
10         int[] a = Array(10);
11         double inicioTime = System.currentTimeMillis();
12         insertion(a);
13         double finTime = System.currentTimeMillis();
14         System.out.println("Se demoro: " + (finTime-inicioTime) + " milisegundos")
15     }
16 }
```

Console: <terminated> Ejercicio3\_Time [Ja  
Se demoro: 0.0 milisegundos

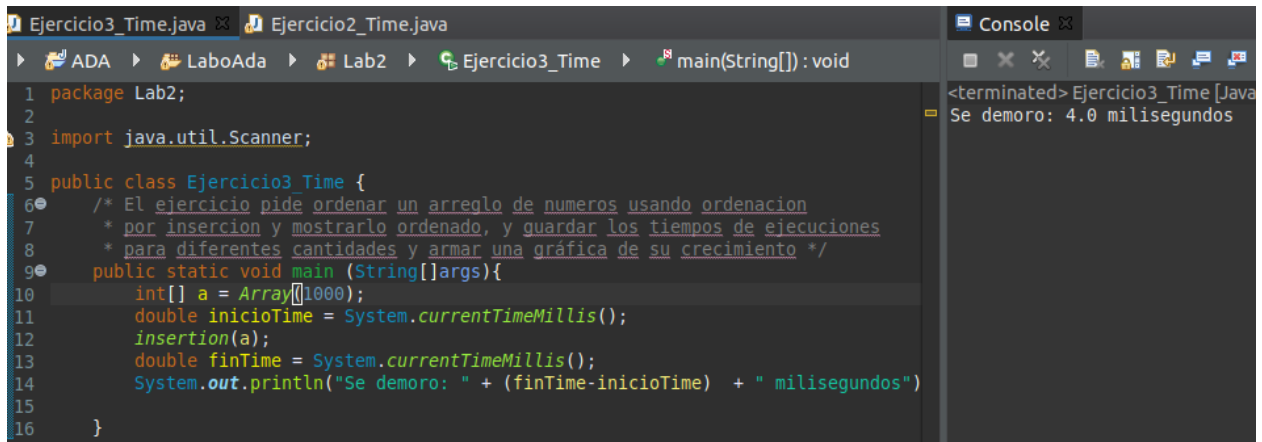
Figura 6 Probando con un array de tamaño  $10^2$



```
1 package Lab2;
2
3 import java.util.Scanner;
4
5 public class Ejercicio3_Time {
6     /* El ejercicio pide ordenar un arreglo de numeros usando ordenacion
7      * por insercion y mostrarlo ordenado, y guardar los tiempos de ejecuciones
8      * para diferentes cantidades y armar una gráfica de su crecimiento */
9     public static void main (String[]args){
10         int[] a = Array[100];
11         double inicioTime = System.currentTimeMillis();
12         insertion(a);
13         double finTime = System.currentTimeMillis();
14         System.out.println("Se demoro: " + (finTime-inicioTime) + " milisegundos")
15     }
16 }
```

<terminated> Ejercicio3\_Time [Java Ap  
Se demoro: 0.0 milisegundos

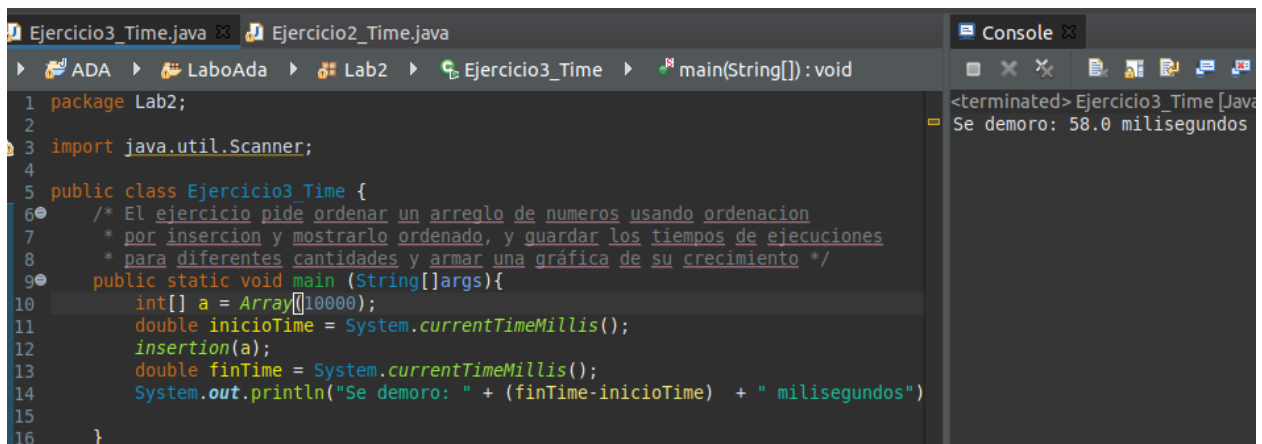
Figura 7 Probando con un array de tamaño  $10^3$



```
1 package Lab2;
2
3 import java.util.Scanner;
4
5 public class Ejercicio3_Time {
6     /* El ejercicio pide ordenar un arreglo de numeros usando ordenacion
7      * por insercion y mostrarlo ordenado, y guardar los tiempos de ejecuciones
8      * para diferentes cantidades y armar una gráfica de su crecimiento */
9     public static void main (String[]args){
10         int[] a = Array[1000];
11         double inicioTime = System.currentTimeMillis();
12         insertion(a);
13         double finTime = System.currentTimeMillis();
14         System.out.println("Se demoro: " + (finTime-inicioTime) + " milisegundos")
15     }
16 }
```

<terminated> Ejercicio3\_Time [Java  
Se demoro: 4.0 milisegundos

Figura 8 Probando con un array de tamaño  $10^4$



```
1 package Lab2;
2
3 import java.util.Scanner;
4
5 public class Ejercicio3_Time {
6     /* El ejercicio pide ordenar un arreglo de numeros usando ordenacion
7      * por insercion y mostrarlo ordenado, y guardar los tiempos de ejecuciones
8      * para diferentes cantidades y armar una gráfica de su crecimiento */
9     public static void main (String[]args){
10         int[] a = Array[10000];
11         double inicioTime = System.currentTimeMillis();
12         insertion(a);
13         double finTime = System.currentTimeMillis();
14         System.out.println("Se demoro: " + (finTime-inicioTime) + " milisegundos")
15     }
16 }
```

<terminated> Ejercicio3\_Time [Java  
Se demoro: 58.0 milisegundos

Figura 9 Probando con un array de tamaño  $10^5$

```

Ejercicio3_Time.java Ejercicio2_Time.java
└─ ADA └─ LaboAda └─ Lab2 └─ Ejercicio3_Time └─ main(String[]): void
1 package Lab2;
2
3 import java.util.Scanner;
4
5 public class Ejercicio3_Time {
6     /* El ejercicio pide ordenar un arreglo de numeros usando ordenacion
7      * por insercion y mostrarlo ordenado, y guardar los tiempos de ejecuciones
8      * para diferentes cantidades y armar una gráfica de su crecimiento */
9     public static void main (String[] args){
10         int[] a = Array[100000];
11         double inicioTime = System.currentTimeMillis();
12         insertion(a);
13         double finTime = System.currentTimeMillis();
14         System.out.println("Se demoro: " + (finTime-inicioTime) + " milisegundos")
15     }
16 }

```

Console

```

<terminated> Ejercicio3_Time [Java Ap
Se demoro: 5367.0 milisegundos

```

En resumen se armó la siguiente tabla:

Tamaño del array	milisegundos
10	0
10 <sup>2</sup>	0
10 <sup>3</sup>	4
10 <sup>4</sup>	58
10 <sup>5</sup>	5367
10 <sup>6</sup>	213636

Milisegundos frente a Tamano del array

