

# UNIVERSIDAD NACIONAL DE SAN CRISTÓBAL DE HUAMANGA

FACULTAD DE INGENIERÍA MINAS, GEOLOGÍA Y CIVIL

ESCUELA DE FORMACION PROFESIONAL DE INGENIERÍA DE SISTEMAS



## INFORME DE LABORATORIO 03

### ARREGLOS

09 octubre 2019

Jueves 2 - 5 pm

ASIGNATURA: IS142 Programación Orientada a Objetos

ALUMNO: Ramos Conga, Christian Yordi

DOCENTE: M.sc. Ing. Fredy Barrientos

AYACUCHO-PERÚ  
2019

## LABORATORIO 004

### 1. Objetivos

#### 1.1. Objetivo General

Poner en práctica el concepto de arreglos de una dimensión en Java.

#### 1.2. Objetivos Específicos

☑ Declarar, instanciar, inicializar, asignar y leer datos de un arreglo.

☑ Iterar por medio de un ciclo for los elementos de un arreglo.

### 2. Requisitos

Este laboratorio requiere de conocimientos previos de manejo de variables, cadenas, operadores y estructuras de control en Java.

### 3. Conceptos relacionados

#### 3.1. Arreglos en Java

En Java la estructura de datos más simple es el arreglo. Un arreglo de datos básicamente es un conjunto de información, la cual nos permite agrupar información normalmente de un tipo en particular.

Por ejemplo, podemos tener dos arreglos, uno de tipo enteros y otro de tipos Estudiante. Es decir, que podemos declarar arreglos que contengan cualquier tipo de datos, ya sea de tipo primitivos o de tipo Object.

No todos los elementos del arreglo deben contener valores. Por ejemplo, si el arreglo de enteros fuera de 10 elementos, pero solo tuviera 7 valores, los 3 últimos valores tendrían su valor por default del tipo declarado, en este caso como es de tipo int, el valor por default para el tipo int es 0.

En el caso del arreglo de tipo object si fuera de 10 elementos, y tuviera solo 5 objetos de tipo estudiante definidos, entonces los 5 restantes su valor seria null, ya que ese es el tipo por default para los tipo Object.

#### 3.2. Declaración de arreglos

A continuación, veamos la sintaxis para declarar un arreglo de una dimensión:

tipo [] nombreArreglo ó tipo nombreArreglo [];

Ejemplos:

☑ Declaración de arreglos de tipo primitivo:

int enteros [];

☐ Declaración de arreglos de tipo Object:

```
String nombres [];
```

```
Estudiante estudiantes [];
```

Declarar un arreglo es como declarar una variable excepto que vamos a utilizar los corchetes [], los cuales se pueden utilizar en dos partes, ya sea antes del nombre de la variable o después del nombre de la variable.

Debido a que los arreglos es una colección de datos, normalmente el nombre de un arreglo es en plural, para que fácilmente con solo leer la variable reconozcamos que se trata de una colección de datos, y en este caso de un arreglo, aunque más adelante veremos que se puede tratar no solamente de arreglos, sino de otras estructuras de datos.

### **3.3. Instanciar arreglos**

A continuación, veremos cómo inicializar un arreglo de una dimensión, ya que hasta el momento con sólo declarar una variable de tipo arreglo la JVM no sabe cuan largo es este arreglo, para ello debemos inicializarlo, veamos como:

```
nombreArreglo = new tipo[largo];
```

### **3.4. Inicializar los elementos de un arreglo**

A continuación, veremos cómo inicializar los elementos de un arreglo:

```
nombreArreglo[indice] = valor;
```

### **3.5. Extraer elementos de un arreglo**

A continuación, veremos cómo extraer elementos de un arreglo:

```
variableReceptora = nombreArreglo[índice];
```

## **4. Desarrollo del laboratorio**

### **4.1. Arreglos de una dimensión en Java**

## **5. Assignment**

La Tarea Académica N° 03 (TA03) consiste en crear un proyecto de nombre

TA03\_Nombre\_ApellidoPaterno, y el paquete debe llamarse arreglos. A continuación, agregar una clase main para cada ejercicio, por ejemplo, para el ejercicio 1, la clase main debe llamarse Ejercicio001.

La estructura final del proyecto deberá tener el siguiente aspecto:

La TA03 está compuesta por los siguientes ejercicios:

1. Escribe un programa que genere 100 números aleatorios del 0 al 20 y que los muestre por pantalla separados por espacios. El programa pedirá entonces por teclado dos valores y a continuación cambiará todas las ocurrencias del primer valor por el segundo en la lista generada anteriormente. Los números que se han cambiado deben aparecer entre comillas.

```
6 package arreglos;
7 import java.util.Scanner;
8 /**
9  *
10  * @author Añapa
11  */
12 public class Ejercicio001 {
13
14     /**
15      * @param args the command line arguments
16      */
17     public static void main(String[] args) {
18         // TODO code application logic here
19         int[] numero = new int[100];
20         int i;
21
22
23         // Rellena el array con números aleatorios
24         for (i = 0; i < 100; i++) {
25
26             numero[i] = (int) (Math.random() * 21 );
27             System.out.print(numero[i] + " ");
28         }
29
30         Scanner leer = new Scanner(System.in);
31         System.out.print("\nIntroduzca un número de los que se han mostrado: ");
32         int valor1 = leer.nextInt();
33         System.out.print("Introduzca el valor por el que quiere sustituirlo: ");
34         int valor2 = leer.nextInt();
35
36         System.out.println("\n### REEMPLAZANDO EL NUMERO INGRESADO ###");
37
38         for (i = 0; i < 100; i++) {
39             if (numero[i] == valor1) {
40                 numero[i] = valor2;
41                 System.out.print( "\"" + numero[i] + "\" ");
42             } else {
43                 System.out.print( numero[i] + " ");
44             }
45         }
46     }
47 }
```

```
31 | int valor1 = leer.nextInt();
```

Output - arreglos (run)

run:

3 6 19 17 5 1 12 13 8 16 6 14 12 14 1 19 15 20 1 16 1 5 4 3 1 19 14 18 20 4 12 1 1 14 13

Introduzca un número de los que se han mostrado: 1

Introduzca el valor por el que quiere sustituirlo: 123

### REEMPLAZANDO EL NUMERO INGRESADO ###

3 6 19 17 5 "123" 12 13 8 16 6 14 12 14 "123" 19 15 20 "123" 16 "123" 5 4 3 "123" 19 14 1

2. Realiza un programa que pida 10 números por teclado y que los almacene en un array. A continuación, se mostrará el contenido de ese array junto al índice (0 – 9) utilizando para ello una tabla. Seguidamente el programa pasará los primos a las primeras posiciones, desplazando el resto de los números (los que no son primos) de tal forma que no se pierda ninguno. Al final se debe mostrar el array resultante.

Por ejemplo:

```

6 package arreglos;
7 import java.util.Scanner;
8 /**
9  *
10  * @author Añapa
11  */
12 public class Ejercicio002 {
13
14     /**
15      * @param args the command line arguments
16      */
17     public static void main(String[] args) {
18         int[] num = new int[10];
19         int[] resultado = new int[10];
20         int contador = 0;
21         int cuentaArray = 0;
22         int prim = 0;
23         int prueba = 0;
24
25         System.out.println("Introduzca 10 números");
26         //primero pedimos los números por teclado
27         for (int i = 0; i < 10; i++) {
28             Scanner leer = new Scanner(System.in);
29             num[i] = leer.nextInt();
30         }
31
32         System.out.println("Índice\tValor");
33         //mostramos el array
34         for (int i = 0; i < 10; i++) {
35             System.out.println(i + "\t" + num[i]);
36         }
37         //comprobamos cuales son primos y los posicionamos primero
38         for (int i = 0; i < 10; i++) {
39             do {
40                 prim++;
41                 prueba = num[i] % prim;
42
43                 if (prueba == 0) {
44                     contador++;
45                 } else {
46                     //nada
47                 }
48             } while (prim != num[i]);
49
50             if (contador == 2) {
51                 resultado[cuentaArray] = num[i];
52                 cuentaArray++;
53             } else {
54                 //nada

```

```

50         if (contador == 2) {
51             resultado[cuentaArray] = num[i];
52             cuentaArray++;
53         } else {
54             //nada
55         }
56
57         contador = 0;
58         prim = 0;
59     } //for
60
61     //ahora metemos en el array resultante los que quedan
62     for (int i = 0; i < 10; i++) {
63         do {
64             prim++;
65             prueba = num[i] % prim;
66
67             if (prueba == 0) {
68                 contador++;
69             } else {
70                 //nada
71             }
72         } while (prim != num[i]);
73
74         if (contador != 2) {
75             resultado[cuentaArray] = num[i];
76             cuentaArray++;
77         } else {
78             //nada
79         }
80
81         contador = 0;
82         prim = 0;
83     } //for
84
85     System.out.println(" \nArray final");
86     //por último mostramos el array resultante
87     System.out.println("índice\tValor");
88     //mostramos el array
89     for (int i = 0; i < 10; i++) {
90         System.out.println(i + "\t" + resultado[i]);
91     }
92 }
93
94     // TODO code application logic here
95

```

Output - arreglos (run)



```
run:
Introduzca 10 números
```

```
1
2
3
2
5
6
4
5
87
7
```

Índice	Valor
0	1
1	2
2	3
3	2
4	5
5	6
6	4
7	5
8	87
9	7

```
Array final
```

Índice	Valor
0	2
1	3
2	2
3	5
4	5
5	7
6	1
7	6
8	4
9	87

```
BUILD SUCCESSFUL (total time: 16 seconds)
```

```
|
```



**3. Escribe un programa que rellene un array de 100 elementos con números enteros aleatorios comprendidos entre 0 y 500 (ambos incluidos). A continuación, el programa mostrará el array y preguntará si el usuario quiere destacar el máximo o el mínimo. Seguidamente se volverá a mostrar el array escribiendo el número destacado entre dobles asteriscos.**

**Ejemplo:**

```

6 package arreglos;
7 import java.util.Scanner;
8 /**
9  *
10  * @author Añapa
11  */
12 public class Ejercicio003 {
13
14     /**
15      * @param args the command line arguments
16      */
17     public static void main(String[] args) {
18         int[] numero = new int[100];
19         int i;
20         // Rellena el array con números aleatorios
21         for (i = 0; i < 100; i++) {
22
23             numero[i] = (int) (Math.random() * 501 );
24             System.out.print(numero[i] + " ");
25         }
26         Scanner leer = new Scanner(System.in);
27         System.out.print("\ndestacar el máximo o mínimo valor: ");
28         int valor1 = leer.nextInt();
29
30
31         System.out.println("\n### DESTANCANDO EL MINIMO O MAXIMO VALOR ###");
32
33         for (i = 0; i < 100; i++) {
34             if (numero[i] == valor1) {
35
36                 System.out.print( "***" + numero[i] + "***");
37             } else {
38                 System.out.print( numero[i] + " ");
39             }
40         }
41         // TODO code application logic here
42     }
43
44 }

```

Output - arreglos (run)



run:

324 62 422 326 80 18 201 267 351 138 196 98 271 69 32 274 290 7 81 473 248 323 203 290 29

destacar el máximo o mínimo valor: 323

### DESTANCANDO EL MINIMO O MAXIMO VALOR ###

324 62 422 326 80 18 201 267 351 138 196 98 271 69 32 274 290 7 81 473 248 \*\*323\*\*203 290

|

Finalmente, debe subir el proyecto y el informe (.pdf) al repositorio remoto (Classroom de GitHub).

Url para activar el repositorio remoto: [https://classroom.github.com/a/9fcLti\\_9](https://classroom.github.com/a/9fcLti_9)