

# Informe de Laboratorio 02

## Tema: Arreglos

Nota

Estudiante	Escuela	Asignatura
Roni Companocca Checco rcompanocca@unsa.edu.pe	Escuela Profesional de Ingeniería de Sistemas	Programación Semestre: II Código: 20210558

Laboratorio	Tema	Duración
02	Arreglos	04 horas

Semestre académico	Fecha de inicio	Fecha de entrega
2023 - B	Del 12 Septiembre 2023	Al 13 Septiembre 2023

## 1. TAREA

- Crear e inicializar arreglos
- Uso del atributo length de arreglos
- Copiar valores de un arreglo a otro mediante System.arraycopy()
- Crear métodos que reciban arreglos como parámetros
- Valorar el uso de arreglos estándar frente a las variables simples

## 2. EQUIPOS, MATERIALES Y TEMAS UTILIZADOS

- Sistema Operativo Windows
- OpenJDK 64-Bits 17.0.7.
- Git 2.39.2.
- Cuenta en GitHub con el correo institucional.

### 3. URL DE REPOSITORIO GITHUB

- URL para el Repositorio GitHub.
- <https://github.com/RONI-COMPANOCCA-CHECCO>
- URL para el laboratorio 02 en el Repositorio GitHub.
- <https://github.com/RONI-COMPANOCCA-CHECCO/FP2-LAB2>
- <https://github.com/RONI-COMPANOCCA-CHECCO/FP2-LAB2>

### 4. ACTIVIDADES

- JUEGO DEL AHORCADO
- En este ejercicio se le solicita a usted implementar el juego del ahorcado utilizando el código parcial que se le entrega.
- Deberá considerar que:
- El juego valida el ingreso de letras solamente. En caso el usuario ingrese un carácter equivocado le dará el mensaje de error y volverá a solicitar el ingreso
- El juego supone que el usuario no ingresa una letra ingresada previamente
- El método `ingreseLetra()` debe ser modificado para incluir las consideraciones de validación
- Puede crear métodos adicionales

### 5. CUESTIONARIO

- ¿Se pueden clonar los arreglos? Explique 2 formas.
- ¿Qué ventajas tiene el uso de los arreglos?
- Investigue la clase `Arrays` y describa 3 métodos importantes que ofrece dicha clase. (fuera de los vistos en el marco conceptual)

### 6. REFERENCIAS

- M. Aedo, "Fundamentos de Programación 2 - Tópicos de Programación Orientada a Objetos", Primera Edición, 2021, Editorial UNSA.
- <https://github.com/rescobedoq/programacion.git>
- J. Dean, "Introduction to programming with Java: A Problem Solving Approach", Third Edition, 2021, McGraw-Hill.
- C. T. Wu, "An Introduction to Object-Oriented Programming with Java", Fifth Edition, 2010, McGraw-Hill.
- P. Deitel, "Java How to Program", Eleventh Edition, 2017, Prentice Hall.

## 7. CODIGO

```
// RONI COMPANOCCA CHECCO
// CUI: 20210558
// LABORATORIO 01
// FUNDAMENTOS DE PROGRAMACION

import java.util.Scanner;
public class hanged {
    public static void main(String []args){

        String ahor1 = " +---+ \n"+
            " | | \n" +
            " | \n" +
            " | \n" +
            " | \n" +
            " | \n" +
            "=====";

        String ahor2 = " +---+ \n"+
            " | | \n" +
            " 0 | \n" +
            " | \n" +
            " | \n" +
            " | \n" +
            "=====";

        String ahor3 = " +---+ \n"+
            " | | \n" +
            " 0 | \n" +
            " | | \n" +
            " | \n" +
            " | \n" +
            "=====";

        String ahor4 = " +---+ \n"+
            " | | \n" +
            " 0 | \n" +
            " /| | \n" +
            " | \n" +
            " | \n" +
            "=====";

        String ahor5 = " +---+ \n"+
            " | | \n" +
            " 0 | \n" +
            " /|\ | \n" +
            " | \n" +
            " | \n" +
            "=====";

        String ahor6 = " +---+ \n"+
            " | | \n" +
            " 0 | \n" +
            " /|\ | \n" +
            " / | \n" +
            "=====";
```

```
        " | \n" +
        "=====";

String ahor7 = " +---+ \n"+
        " | | \n" +
        " 0 | \n" +
        " /|\ | \n" +
        " / \ | \n" +
        " | \n" +
        "=====";

String [] figuras = {ahor1, ahor2, ahor3,ahor4,ahor5,ahor6,ahor7};
int contador = 1;
int contadorError = 1;
String letra;
String progreso="";
String [] palabras = {"programacion", "java", "identacion", "clases","objetos",
        "desarrollador", "pruebas"};
String palSecreta = getPalabraSecreta(palabras);
for(int i=0;i<palSecreta.length();i++) {
    progreso=progreso+"_ ";
}
System.out.println(figuras[0]);
mostrarBlancos(palSecreta);
System.out.println("\n");
while(contadorError<=6){
    letra = ingreseLetra();
    if (letraEnPalabraSecreta(letra, palSecreta)) {
        progreso=mostrarBlancosActualizados(letra, palSecreta, progreso);
    }else {
        System.out.println(figuras[contadorError]);
        contadorError++;
        progreso=mostrarBlancosActualizados(letra, palSecreta, progreso);
    }
    contador++;

// COMPLETAR PARA INDICAR SI GANO, PERDIO Y CUANTOS TURNOS NECESITO
/* EN ESTA PARTE COMPLETAMOS SI GANA, PARA LO CUAL EL PROGRESO QUE TENEMOS VAMOS
CONTANDO
ANTES DE AHORCARNO, ESTA PARTE NOS PERMITE IMPRIMIR EN LA PANTALLA EL NUMERO DE
INTENTOS PARA
PODER LOGRARLO */
for(int i=0;i<progreso.length();i++) {
    if(!progreso.substring(i, i+1).equals("_")) {
        if(i==progreso.length()-1) {
            System.out.println("Ganaste!");
            System.out.println("Nmero de intentos: "+contador);
            System.exit(0);
        }
    }else {
        break;
    }
}
System.out.println("\n");
}

// SI LO PERDEMOS ENTONCES ESTE NOS SALE EN PANTALLA, EN EL MOMENTO DE AHORCARNOS
```

```
System.out.println("Perdiste :(");
System.out.println("La palabra secreta era: "+palSecreta);
}

/* ESTE METODO NOS PERMITE ALEATORIAMENTE DAR LA PALABRA SECRETA QUE YA LO TENEMOS, DE
   TODAS LAS
   * PALABRAS SECRETAS NOS SELECCIONA A UNA DE ELLAS PARA QUE EL USUARIO PUEDA COMPLETAR.
   */
public static String getPalabraSecreta(String [] lasPalabras){
    String palSecreta;
    int ind;
    int indiceMayor = lasPalabras.length -1;
    int indiceMenor =0;
    ind = (int) ((Math.random() * (indiceMayor - indiceMenor + 1) + indiceMenor));
    return lasPalabras[ind];
}

/*
   * EN ESTE METODO VACIO MOSTRAMOS LO ESPACIOS QUE VA A OCUPAR LA LETRA QUE HA SIDO
   ESCOGIDO
   * ALEATORIAMENTE.
   */
public static void mostrarBlancos(String palabra) {
    for(int i=0; i< palabra.length(); i++) {
        System.out.print("_ ");
    }
}

/*
   * EN ESTE METODO INGRESAMOS UNA LETRA QUE AUTOMATICAMENTE VA SER VALIDADO POR EL METODO
   LETRAVALIDO,
   * YA QUE LOS NUMEROS O OTRAS TECLAS NO SERAN ADMITIDOS Y UNA VES QUE LA LETRA INGRESADA
   SEA VALIDA,
   * NOS IMPRIMIRA EN LA PANTALLA SI Y SOLO SI ES PARTE DE LA PALABRA SECRETO.
   */
public static String ingreseLetra(){
    String laLetra;
    Scanner sc = new Scanner(System.in);
    System.out.println("Ingrese letra: ");
    laLetra = sc.next();
    while(laLetra.length()!= 1 || !letraValido(laLetra)){
        System.out.println("Ingrese letra: ");
        laLetra = sc.next();
    }return laLetra;
}

/*ESTE METODO NOS PERMITE SELECCIONAR LAS LETRAS COORECTAS, QUE SI FORMAN PARTE DE LA
   PALABRA
   * SECRETO SINO FORMA PARTE NO ADMITIRA.
   */
public static boolean letraEnPalabraSecreta(String letra, String palSecreta ){
    for(int i=0;i<palSecreta.length();i++) {
        if(letra.equals(palSecreta.substring(i, i+1))) {
            return true;
        }
    }
}
```

```
return false;
}

/*
 * ESTE METODO NOS PERMITE GUARDAR LAS LETRAS CORRECTAS DE LA PALABRA SECRETO QUE
   INGRESAMOS LOS
USUARIOS, CADA VES QUE INGRESAMOS UNA LETRA QUE SI PERTENECE LO GUARDA Y NOS IMPRIME EN LA
PANTALLA
MIENTRAS AVANZAMOS CON ADIVINAR LA PALABRA SECRETO.
 */
public static String mostrarBlancosActualizados(String letra,String palSecreta,String
progreso){
    System.out.println("PROCESANDO.....");
    String avance1="";
    for(int i=0;i<palSecreta.length();i++) {
        if(letra.equals(palSecreta.substring(i, i+1))) {
            avance1=avance1+letra+" ";
        }else{
            avance1=avance1+progreso.substring(2*i, 2*i+2);
        }
    }
    System.out.print(avance1);
    return avance1;
}

/*EN ESTE METODO LO QUE HACEMOS ES VALIDAR LAS LETRAS QUE VAMOS A USAR AL MOMENTO DE
INGRESAR POR EL
 * TECLADO, PARA QUE NO NOS PUEDA SALIR EL ERROR.
 */
public static boolean letraValido(String laLetra) {
    String abecedario[]=
        {"a","b","c","d","e","f","g","h","i","j","k","l","m","n",
" ", "o","p","q","r","s","t","u","v","w","x","y","z"};
    for(int i=0;i<abecedario.length;i++) {
        if(laLetra.equalsIgnoreCase(abecedario[i])) {
            return true;
        }
    }
    return false;
}
}
```

## 8. EJECUCION CUANDO NO SE AHORCA

```
+---+
|   |
|   |
|   |
|   |
|   |
=====
- - - - -
Ingrese letra:
```

```
P
+---+
|   |
0   |
    |
    |
=====
PROCESANDO.....
- - - - -

Ingrese letra:
I
+---+
|   |
0   |
|   |
    |
    |
=====
PROCESANDO.....
- - - - -

Ingrese letra:
D
+---+
|   |
0   |
/|  |
    |
    |
=====
PROCESANDO.....
- - - - -

Ingrese letra:
d
PROCESANDO.....
- d - - - - -

Ingrese letra:
i
PROCESANDO.....
i d _ _ _ i _ _

Ingrese letra:
e
PROCESANDO.....
i d e _ _ _ i _ _

Ingrese letra:
n
PROCESANDO.....
i d e n _ _ _ i _ _

Ingrese letra:
```

```
t
PROCESANDO.....
i d e n t _ _ i _ n

Ingrese letra:
a
PROCESANDO.....
i d e n t a _ i _ n

Ingrese letra:
c
PROCESANDO.....
i d e n t a c i _ n

Ingrese letra:
o
PROCESANDO.....
i d e n t a c i o n Ganaste!
Nmero de intentos: 12
```

## 9. EJECUCION CUANDO SE AHORCA

```
+---+
|   |
|   |
|   |
|   |
=====
- - - - -

Ingrese letra:
e
PROCESANDO.....
- - - e - - -

Ingrese letra:
j
+---+
|   |
0   |
|   |
|   |
=====
PROCESANDO.....
- - - e - - -

Ingrese letra:
r
PROCESANDO.....
- r - e - - -

Ingrese letra:
```



```
o
+---+
|   |
0   |
|   |
    |
=====
PROCESANDO.....
_ r _ e _ _ _

Ingrese letra:
l
+---+
|   |
0   |
/|   |
    |
=====
PROCESANDO.....
_ r _ e _ _ _

Ingrese letra:
W
+---+
|   |
0   |
/|\  |
    |
=====
PROCESANDO.....
_ r _ e _ _ _

Ingrese letra:
T
+---+
|   |
0   |
/|\  |
/   |
    |
=====
PROCESANDO.....
_ r _ e _ _ _

Ingrese letra:
P
+---+
|   |
0   |
/|\  |
/ \  |
    |
=====
```

PROCESANDO.....

\_ r \_ e \_ \_ \_

Perdiste :(

La palabra secreta era: pruebas