Informe de Laboratorio 02

Tema: Arreglos Estándar

Nota

Estudiante	Escuela	Asignatura
Hernan Andy Choquehuanca Zapana hchoquehuancaz@unsa.edu.pe	Escuela Profesional de Ingeniería de Sistemas	Fundamentos de la Programación II Semestre: II Código: 20232191

Laboratorio	Tema	Duración
02	Arreglos Estándar	02 horas

Semestre académico	Fecha de inicio	Fecha de entrega
2023 - B	Del 18 Septiembre 2023	Al 20 Septiembre 2023

1. Tarea

- 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.
- Utilizar Git para evidenciar su trabajo.

2. Equipos, materiales y temas utilizados

- Sistema Operativo Windows 11 Pro 22H2 64 bits.
- VIM 9.0.
- Visual Studio Code
- Git 2.41.1.

- Cuenta en GitHub con el correo institucional.
- Variables Simples
- Arreglos Estándar
- Métodos

3. URL de Repositorio Github

- URL del Repositorio GitHub para clonar o recuperar.
- `https://github.com/hernanchoquehuanca/fp2-23b.git`
- URL para el laboratorio 02 en el Repositorio GitHub.
- `https://github.com/hernanchoquehuanca/fp2-23b/tree/main/fase01/lab02`

4. Actividades con el repositorio GitHub

4.1. Commits

4.1.1. Actividad 1 : Implementar el juego del ahorcado utilizando el código parcial que se entregó :

- Primero acomodamos y copiamos el código para luego empezar a completarlo
- El código fue el siguiente:

Listing 1: Ejercicio01.java

```java
package fase01.lab02;
import java.util.Scanner;
public class Ejercicio01 {
  public static void main(String []args){
    String ahor1 = " +---+ \n" +
                   " |   |  \n" +
                   "     |  \n" +
                   "     |  \n" +
                   "     |  \n" +
                   "     |  \n" +
                   "========= ";
//
    String ahor2 = " +---+  \n"+
                   " |   |   \n"+
                   " O   |   \n"+
                   "     |   \n"+
                   "     |   \n"+
                   "     |   \n"+
                   "=========";
//
    String ahor3 = " +---+  \n"+
                   " |   |   \n"+
                   " O   |   \n"+
                   " |   |   \n"+
                   "     |   \n"+
```

```
26                  "     |   \n"+
27                  "=========";
28  //
29      String ahor4 = " +---+  \n"+
30                  " |   |   \n"+
31                  " O   |   \n"+
32                  "/|   |   \n"+
33                  "     |   \n"+
34                  "     |   \n"+
35                  "=========";
36  //
37      String ahor5 = " +---+  \n"+
38                  " |   |   \n"+
39                  " O   |   \n"+
40                  "/|\\  |   \n"+
41                  "     |   \n"+
42                  "     |   \n"+
43                  "=========";
44  //
45      String ahor6 = " +---+  \n"+
46                  " |   |   \n"+
47                  " O   |   \n"+
48                  "/|\\  |   \n"+
49                  "/    |   \n"+
50                  "     |   \n"+
51                  "=========";
52  //
53      String ahor7 = " +---+  \n"+
54                  " |   |   \n"+
55                  " O   |   \n"+
56                  "/|\\  |   \n"+
57                  "/ \\  |   \n"+
58                  "     |   \n"+
59                  "=========";
60  //
61      String [] figuras = {ahor1, ahor2, ahor3,ahor4,ahor5,ahor6,ahor7};
62      int contador = 1;
63      String letra;
64
65      String [] palabras = {"programacion", "java", "identacion", "clases",
66      "objetos", "desarrollador", "pruebas"};
67
68      String palSecreta = getPalabraSecreta(palabras);
69      System.out.println(figuras[0]);
70
71      mostrarBlancos(palSecreta);
72
73      System.out.println("\n");
74
75      while(contador <= 6){
76        letra = ingreseLetra();
77        if (letraEnPalabraSecreta(letra, palSecreta))
78          mostrarBlancosActualizados(letra);
79        else
80          System.out.println(figuras[contador]);
81        contador = contador +1;
```

```
    }

    //COMPLETAR PARA INDICAR SI GAN, PERDI Y CUNTOS TURNOS NECESIT
    System.out.println("\n");
  }

  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];
  }

  public static void mostrarBlancos(String palabra){
    for(int i=0; i< palabra.length(); i++)
      System.out.print("_ " );

  }

  public static String ingreseLetra(){
    String laLetra;
    Scanner sc = new Scanner(System.in);
    System.out.println("Ingrese letra: ");
    laLetra = sc.next();
    while(laLetra.length()!= 1){
      System.out.println("Ingrese letra: "); //COMPLETAR PARA VALIDAR CARACTERES PERMITIDOS
      laLetra = sc.next();
    }
    return laLetra;
  }

  public static boolean letraEnPalabraSecreta(String letra, String palSecreta ){
  //COMPLETAR
    return false;
  }

  public static void mostrarBlancosActualizados(String letra){
  //COMPLETAR
    System.out.println("PROCESANDO.....");
  }
}
```

Listing 2: Commit: Copiando el código proporcionado y acomodandolo

```
$ git add .
$ git commit -m "Copiando el codigo proporcionado y acomodandolo"
$ git push -u origin main
```

- Para validar el ingreso de letras aceptadas se hizo uso de los códigos ASCII
- Además para no usar dos condicionales que para mayúsculas y minúsculas, se usa el método "toUpperCase()"para validar el rango entre 65 y 90
- El método fue el siguiente:

Listing 3: Ejercicio01.java

```java
public static String ingreseLetra(){
  String laLetra;
  Scanner sc = new Scanner(System.in);
  System.out.println("Ingrese letra: ");
  laLetra = sc.next().toUpperCase();
  char c = laLetra.charAt(0);
  while(laLetra.length()!= 1 || (int)c < 65 || (int)c > 90){
    System.out.println("ERROR CARACTER NO ADMITIDO - Ingrese letra: ");
    laLetra = sc.next();
  }
  return laLetra;
}
```

Listing 4: Commit: Completando el metodo ingreseLetra usando codigo ascii para validar las letras

```
$ git add .
$ git commit -m "Completando el metodo ingreseLetra usando codigo ascii para validar
    las letras"
$ git push -u origin main
```

- Para el método "letraEnPalabraSecreta"simplemente usaremos un bucle for que recorra por cada char de "palSecretaçon una condicional (if) que retorne true si en algun char coincide el ingresado por el jugador con el de la palabra.
- En caso de no haber coincidencias, al finalizar el bucle for se retornará falso.

Listing 5: Ejercicio01.java

```java
public static boolean letraEnPalabraSecreta(String letra, String palSecreta ){
  for (int i = 0; i < palSecreta.length(); i++) {
    if (palSecreta.charAt(i) == letra.charAt(0))
      return true;
  }
  return false;
}
```

Listing 6: Commit: Completando el metodo letraEnPalabraSecreta verificando char por char si hay una coincidencia

```
$ git add .
$ git commit -m "Completando el metodo letraEnPalabraSecreta verificando char por char
    si hay una coincidencia"
$ git push -u origin main
```

- En el método "mostrarBlancosActualizados"hubieron dos versiones con modificaciones.
- Primero, se intentó usar el método para retornar una variable que guarde las letras adivinadas anteriormente a la vez que los imprime, pero al momento de guardar la palabra sería tedioso, ya que guardaría un string donde cada letra tenga en medio.
- Segundo, se creó un nuevo String para guardar los datos y mantener el String "blancos", luego lo retorna para que en las siguientes veces se muestre la palabra que ha ido completando con los anteriores aciertos.

Listing 7: Ejercicio01.java

```
1  public static String mostrarBlancosActualizados(String letra, String palabra, String
       blancos){
2    System.out.println("PROCESANDO...");
3    // la variable newBlancos sirve para guardar los valores de blancos y no perderlos
4    String newBlancos = "";
5    char c = letra.charAt(0);
6    for(int i = 0; i < palabra.length(); i++) {
7      if (palabra.charAt(i) == c)
8        newBlancos += c + " ";
9      else
10       newBlancos += blancos.charAt(i * 2) + " "; // asi usaremos el antiguo blancos
11   }
12   System.out.println(blancos);
13   // retornamos blancos y asi guardarlos para la proxima coincidencia de la palabraSecreta
14   return newBlancos;
15 }
```

Listing 8: Commit: Segunda version del metodo con cambios para mostrar las letras adivinadas previamente al jugador

```
$ git add .
$ git commit -m "Segunda version del metodo con cambios para mostrar las letras
    adivinadas previamente al jugador"
$ git push -u origin main
```

- Hubieron más modificaciones para que el String "blancos"pueda guardar el avance del jugador y a su vez nos sirva para el método "letraEnPalabraSecreta"
- Además se modificó el método "mostrarBlancos"para que al crear el contenido del String "blancos"lo haga sin espacios en cada posición par
- Y la tercera modificación fue en el método "mostrarBlancosActualizadosza que ahora que "blancos"no contiene espacios, se le agregarán sólo al momento de imprimirlos

- El código con todos los cambios mencionados anteriormente quedaría de la siguiente manera :

Listing 9: Ejercicio01.java

```
1  package fase01.lab02;
2  import java.util.Scanner;
3  public class Ejercicio01 {
4    public static void main(String []args){
5      String ahor1 = " +---+ \n" +
6                     " |   | \n" +
7                     "     | \n" +
8                     "     | \n" +
9                     "     | \n" +
10                    "     | \n" +
11                    "========= ";
12 //
13     String ahor2 = " +---+ \n"+
14                    " |   |   \n"+
```

```
                    " O   |   \n"+
                    "     |   \n"+
                    "     |   \n"+
                    "     |   \n"+
                    "=========";
//
    String ahor3 = " +---+ \n"+
                    " |   |   \n"+
                    " O   |   \n"+
                    " |   |   \n"+
                    "     |   \n"+
                    "     |   \n"+
                    "=========";
//
    String ahor4 = " +---+ \n"+
                    " |   |   \n"+
                    " O   |   \n"+
                    "/|   |   \n"+
                    "     |   \n"+
                    "     |   \n"+
                    "=========";
//
    String ahor5 = " +---+ \n"+
                    " |   |   \n"+
                    " O   |   \n"+
                    "/|\\ |   \n"+
                    "     |   \n"+
                    "     |   \n"+
                    "=========";
//
    String ahor6 = " +---+ \n"+
                    " |   |   \n"+
                    " O   |   \n"+
                    "/|\\ |   \n"+
                    "/    |   \n"+
                    "     |   \n"+
                    "=========";
//
    String ahor7 = " +---+ \n"+
                    " |   |   \n"+
                    " O   |   \n"+
                    "/|\\ |   \n"+
                    "/ \\ |   \n"+
                    "     |   \n"+
                    "=========";
//
    String [] figuras = {ahor1, ahor2, ahor3,ahor4,ahor5,ahor6,ahor7};
    int contador = 1;
    String letra;
// posiblemente en lugar de identacion seria : indentacion;
    String [] palabras = {"programacion", "java", "identacion", "clases",
    "objetos", "desarrollador", "pruebas"};

    String palSecreta = getPalabraSecreta(palabras);
    System.out.println(figuras[0]);
    String blancos = mostrarBlancos(palSecreta);
```

```
71
72      System.out.println("\n");
73
74      while(contador <= 6){
75        letra = ingreseLetra();
76        if (letraEnPalabraSecreta(letra, palSecreta)){
77          blancos = mostrarBlancosActualizados(letra, palSecreta, blancos);
78          if ()
79        }
80        else
81          System.out.println(figuras[contador]);
82        contador = contador +1;
83      }
84
85      if () {
86
87      }
88      //COMPLETAR PARA INDICAR SI GAN, PERDI Y CUNTOS TURNOS NECESIT
89      System.out.println("\n");
90    }
91
92    public static String getPalabraSecreta(String [] lasPalabras){
93      //String palSecreta;
94      int ind;
95      int indiceMayor = lasPalabras.length -1;
96      int indiceMenor = 0;
97      ind = (int) (Math.random() * (indiceMayor - indiceMenor + 1) + indiceMenor);
98      return lasPalabras[ind];
99    }
100
101   public static String mostrarBlancos(String palabra){
102     String blancos = "";
103     for(int i = 0; i < palabra.length(); i++) {
104       blancos += "_";
105     }
106     for (int i = 0; i < blancos.length(); i++) {
107       System.out.println(blancos.charAt(i) + " ");
108     }
109     return blancos;
110   }
111
112   public static String ingreseLetra(){
113     String laLetra;
114     Scanner sc = new Scanner(System.in);
115     System.out.println("Ingrese letra: ");
116     laLetra = sc.next().toUpperCase();
117     char c = laLetra.charAt(0);
118     while(laLetra.length()!= 1 || (int)c < 65 || (int)c > 90){
119       System.out.println("ERROR CARACTER NO ADMITIDO - Ingrese letra: ");
120       laLetra = sc.next();
121     }
122     return laLetra;
123   }
124
125   public static boolean letraEnPalabraSecreta(String letra, String palSecreta ){
126     for (int i = 0; i < palSecreta.length(); i++) {
```

```
127       if (palSecreta.charAt(i) == letra.charAt(0))
128         return true;
129     }
130     return false;
131   }
132
133   public static String mostrarBlancosActualizados(String letra, String palabra, String
          blancos){
134     System.out.println("PROCESANDO...");
135     // la variable newBlancos sirve para guardar los valores de blancos y no perderlos
136     String newBlancos = "";
137     char c = letra.charAt(0);
138     for(int i = 0; i < palabra.length(); i++) {
139       if (palabra.charAt(i) == c)
140         newBlancos += c;
141       else
142         newBlancos += blancos.charAt(i); // asi usaremos el antiguo blancos
143     }
144     for (int i = 0; i < blancos.length(); i++) {
145       System.out.println(blancos.charAt(i) + " ");
146     }
147     // retornamos blancos y asi guardarlos para la proxima coincidencia de la palabraSecreta
148     return newBlancos;
149   }
150 }
```

- A pesar de las modificaciones anteriores existían errores que aparecieron al momento de compilar.
- Primero, se agregó la condicional que determinaría si el jugador alcanzó el límite de intentos (6) o si completo la palabra secreta, esto usando break que terminará el bucle cuando se haya completado la palabra.
- Segundo, se implementó los mensajes al finalizar el juego, tanto si se ganase o perdiese.
- Tercero, se modificó el método "mostrarBlancosza que imprimía con saltos de línea, lo cual no era correcto visualmente.
- Cuarto, al momento de recibir la letra ingresada se aplicaba el toUpperCase() para que la letra quede modificada desde que se recibe.
- Finalmente para que el juego funcione correctamente se agregó el contador de intentos a la condicional que se ejecuta cuando el usuario no acierta la letra, de está manera sería posible ganar si la palabra secreta tiene más de 6 letras distintas

- El código con todos los cambios ya mencionados es el siguiente:

Listing 10: Ejercicio01.java

```
// Laboratorio Nro 02 - Ejercicio01
// Autor : Hernan Andy
//  Colabor  : -
// Tiempo : -

package fase01.lab02;
import java.util.Scanner;
public class Ejercicio01 {
  public static void main(String []args){
    String ahor1 = " +---+ \n" +
                   " |   |  \n" +
                   "     |  \n" +
                   "     |  \n" +
                   "     |  \n" +
                   "     |  \n" +
                   "========= ";
//
    String ahor2 = " +---+  \n"+
                   " |   |   \n"+
                   " O   |   \n"+
                   "     |   \n"+
                   "     |   \n"+
                   "     |   \n"+
                   "=========";
//
    String ahor3 = " +---+  \n"+
                   " |   |   \n"+
                   " O   |   \n"+
                   " |   |   \n"+
                   "     |   \n"+
                   "     |   \n"+
                   "=========";
//
    String ahor4 = " +---+  \n"+
                   " |   |   \n"+
                   " O   |   \n"+
                   "/|   |   \n"+
                   "     |   \n"+
                   "     |   \n"+
                   "=========";
//
    String ahor5 = " +---+  \n"+
                   " |   |   \n"+
                   " O   |   \n"+
                   "/|\\  |   \n"+
                   "     |   \n"+
                   "     |   \n"+
                   "=========";
//
    String ahor6 = " +---+  \n"+
                   " |   |   \n"+
                   " O   |   \n"+
```

```
                    "/|\\  |   \n"+
                    "/    |   \n"+
                    "     |   \n"+
                    "=========";
//
    String ahor7 = " +---+  \n"+
                    " |   |   \n"+
                    " O   |   \n"+
                    "/|\\  |   \n"+
                    "/ \\  |   \n"+
                    "     |   \n"+
                    "=========";
//
    String [] figuras = {ahor1, ahor2, ahor3,ahor4,ahor5,ahor6,ahor7};
    int contador = 1;
    String letra;
// posiblemente en lugar de identacion seria : indentacion;
    String [] palabras = {"programacion", "java", "identacion", "clases",
    "objetos", "desarrollador", "pruebas"};

    String palSecreta = getPalabraSecreta(palabras);
    System.out.println(figuras[0]);
    String blancos = mostrarBlancos(palSecreta);

    System.out.println("\n");

    while(contador <= 6){
      letra = ingreseLetra();
      if (letraEnPalabraSecreta(letra, palSecreta)){
        blancos = mostrarBlancosActualizados(letra, palSecreta, blancos);
        if (palSecreta.equals(blancos))
          break;
      }
      else {
        System.out.println(figuras[contador]);
        // incluyendo el contador para que sirva al momento de contar los errores, y asi
            exista posibilida de ganar
        // si la palabra contiene mas de 6 letras distintas
        contador = contador +1;
      }
    }

    if (contador == 7){
      System.out.println("Perdiste");
      System.out.println("La palabra secreta era : " + palSecreta);
    } else {
      System.out.println("Ganaste");
      System.out.println("Nro de intentos fallidos: " + contador);
    }
    System.out.println("\n");
  }

  public static String getPalabraSecreta(String [] lasPalabras){
    //String palSecreta;
    int ind;
    int indiceMayor = lasPalabras.length -1;
```

```
108   int indiceMenor = 0;
109   ind = (int) (Math.random() * (indiceMayor - indiceMenor + 1) + indiceMenor);
110   return lasPalabras[ind];
111 }
112
113 public static String mostrarBlancos(String palabra){
114   String blancos = "";
115   for(int i = 0; i < palabra.length(); i++) {
116     blancos += "_";
117   }
118   for (int i = 0; i < blancos.length(); i++) {
119     System.out.print(blancos.charAt(i) + " ");
120   }
121   return blancos;
122 }
123
124 public static String ingreseLetra(){
125   String laLetra;
126   Scanner sc = new Scanner(System.in);
127   System.out.println("Ingrese letra: ");
128   laLetra = sc.next().toLowerCase();
129   char c = laLetra.charAt(0);
130   while(laLetra.length()!= 1 || (int)c < 97 || (int)c > 122){
131     System.out.println("ERROR - CARACTER NO ADMITIDO - Ingrese letra: ");
132     laLetra = sc.next().toLowerCase();
133     c = laLetra.charAt(0);
134   }
135   return laLetra;
136 }
137
138 public static boolean letraEnPalabraSecreta(String letra, String palSecreta ){
139   for (int i = 0; i < palSecreta.length(); i++) {
140     if (palSecreta.charAt(i) == letra.charAt(0))
141       return true;
142   }
143   return false;
144 }
145
146 public static String mostrarBlancosActualizados(String letra, String palabra, String
      blancos){
147   System.out.println("PROCESANDO...");
148   // la variable newBlancos sirve para guardar los valores de blancos y no perderlos
149   String newBlancos = "";
150   char c = letra.charAt(0);
151   for(int i = 0; i < palabra.length(); i++) {
152     if (palabra.charAt(i) == c)
153       newBlancos += c;
154     else
155       newBlancos += blancos.charAt(i); // asi usaremos el antiguo blancos
156   }
157   for (int i = 0; i < newBlancos.length(); i++) {
158     System.out.print(newBlancos.charAt(i) + " ");
159   }
160   System.out.println("");
161   // retornamos blancos y asi guardarlos para la proxima coincidencia de la palabraSecreta
162   return newBlancos;
```

```
163   }
164 }
165 /*
166 Luego de haber terminado el codigo, siento que el contador del main deberia incluirse dentro
        del else,
167 ya que como se presenta ahora es imposible ganar si es que le toca palabras con mas de 6
        letras distintas
168 como lo es programacion y el incluirlo dentro del else ayudara a ello
169  */
```

```
PowerShell 7.3.7
PS E:\fp2-23b\fase01\lab02>  & 'C:\Program Files\Eclipse Adoptium\jdk-17.0.8.7-hotspo
t\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\MSI\AppData
\Roaming\Code\User\workspaceStorage\eea1fe189d4f1f22b99ca1ec938a156d\redhat.java\jdt_
ws\jdt.ls-java-project\bin' 'fase01.lab02.Ejercicio01'
 +---+
 |   |
     |
     |
     |
     |
=========
_ _ _ _ _ _ _

Ingrese letra:
a
PROCESANDO...
_ _ _ _ _ a _
Ingrese letra:
s
PROCESANDO...
_ _ _ _ _ a s
Ingrese letra:
3
ERROR - CARACTER NO ADMITIDO - Ingrese letra:
p
PROCESANDO...
p _ _ _ _ a s
Ingrese letra:
r
PROCESANDO...
p r _ _ _ a s
Ingrese letra:
U
PROCESANDO...
p r u _ _ a s
Ingrese letra:
e
PROCESANDO...
p r u e _ a s
Ingrese letra:
b
PROCESANDO...
p r u e b a s
Ganaste
Nro de intentos fallidos: 1


PS E:\fp2-23b\fase01\lab02>
```

4.2. Estructura de laboratorio 01

- El contenido que se entrega en este laboratorio es el siguiente:

```
lab01
|---|--- Ejercicio01.java
|--- latex
   |--- img
   |   |--- logo_abet.png
   |   |--- logo_episunsa.png
   |   |--- logo_unsa.jpg
   |   |--- prueba01.png
   |--- Informe_lab02.pdf
   |--- Informe_lab02.tex
   |--- src
       |---|--- Ejercicio01.java
       |---|--- Ejercicio01v1.java
       |---|--- Ejercicio01v2.java
       |---|--- Ejercicio01v3.java
       |---|--- Ejercicio01v4.java
       |---|--- Ejercicio01v5.java
```

5. Rúbricas

5.1. Entregable Informe

Tabla 1: Tipo de Informe

Informe	
Latex	El informe está en formato PDF desde Latex, con un formato limpio (buena presentación) y facil de leer.

5.2. Rúbrica para el contenido del Informe y demostración

- El alumno debe marcar o dejar en blanco en celdas de la columna **Checklist** si cumplio con el ítem correspondiente.
- Si un alumno supera la fecha de entrega, su calificación será sobre la nota mínima aprobada, siempre y cuando cumpla con todos lo items.
- El alumno debe autocalificarse en la columna **Estudiante** de acuerdo a la siguiente tabla:

Tabla 2: Niveles de desempeño

	Nivel			
Puntos	Insatisfactorio 25 %	En Proceso 50 %	Satisfactorio 75 %	Sobresaliente 100 %
2.0	0.5	1.0	1.5	2.0
4.0	1.0	2.0	3.0	4.0

Tabla 3: Rúbrica para contenido del Informe y demostración

Contenido y demostración		Puntos	Checklist	Estudiante	Profesor
1. GitHub	Hay enlace URL activo del directorio para el laboratorio hacia su repositorio GitHub con código fuente terminado y fácil de revisar.	2	X	2	
2. Commits	Hay capturas de pantalla de los commits más importantes con sus explicaciones detalladas. (El profesor puede preguntar para refrendar calificación).	4	X	3	
3. Código fuente	Hay porciones de código fuente importantes con numeración y explicaciones detalladas de sus funciones.	2	X	2	
4. Ejecución	Se incluyen ejecuciones/pruebas del código fuente explicadas gradualmente.	2	X	0.5	
5. Pregunta	Se responde con completitud a la pregunta formulada en la tarea. (El profesor puede preguntar para refrendar calificación).	2	X	2	
6. Fechas	Las fechas de modificación del código fuente estan dentro de los plazos de fecha de entrega establecidos.	2	X	2	
7. Ortografía	El documento no muestra errores ortográficos.	2	X	2	
8. Madurez	El Informe muestra de manera general una evolución de la madurez del código fuente, explicaciones puntuales pero precisas y un acabado impecable. (El profesor puede preguntar para refrendar calificación).	4	X	3	
Total		20		16.5	

6. Referencias

- https://docs.oracle.com/javase/tutorial/java/nutsandbolts/variables.html
- https://docs.oracle.com/javase/8/docs/api/java/util/Arrays.html
- https://docs.oracle.com/javase/tutorial/java/javaOO/methods.html