


	<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: 1</p>

## INFORME DE LABORATORIO

### (formato estudiante)

INFORMACIÓN BÁSICA					
<b>ASIGNATURA:</b>	<i>Fundamentos de la programación 2</i>				
<b>TÍTULO DE LA PRÁCTICA:</b>	<i>Definición de Clases de Usuario Clase Soldado</i>				
<b>NÚMERO DE PRÁCTICA:</b>	<i>09</i>	<b>AÑO LECTIVO:</b>	<i>2023</i>	<b>NRO. SEMESTRE:</b>	<i>2do Semestre</i>
<b>FECHA DE PRESENTACIÓN</b>	<i>25/12/2023</i>	<b>HORA DE PRESENTACIÓN</b>	<i>21/30/00</i>		
<b>INTEGRANTE (s)</b> <i>Juan Diego Gutiérrez Ccama</i>				<b>NOTA (0-20)</b>	<i>Nota colocada por el docente</i>
<b>DOCENTE(s):</b> <i>Linno Jose Pinto Oppe</i>					

RESULTADOS Y PRUEBAS
<b>I. EJERCICIOS RESUELTOS:</b>

```
*Soldado.java x Videojuego.java
1
2 public class Soldado {
3     private String nombre;
4     private int nivelAtaque;
5     private int nivelDefensa;
6     private int nivelVida;
7     private int velocidad;
8     private String actitud;
9     private int vidaActual;
10    private boolean vive;
11    private int posFila;
12    private char posCol;
13    private char figura;
14    public Soldado(String n) {
15        nombre = n;
16    }
17    public int atacar() {
18        actitud = "ofensiva";
19        return avanzar();
20    }
21    public void defender() {
22        velocidad = 0;
23        actitud = "defensiva";
24    }
25    public int avanzar() {
26        return velocidad+1;
27    }
28    public void retroceder() {
29        defender();
30        actitud = "defensiva";
31        velocidad = velocidad - 1;
32    }
33    public void serAtacado() {
34        nivelVida = nivelVida - 1;
35    }
36    public void huir() {
37        actitud = "fuga";
38    }
39    public void morir() {
40        actitud = "muerto";
41        nivelVida = 0;
42    }
43    //sets
```

## II. PRUEB

```

43 //sets
44 public void setVidaActual(int a) {
45     nivelVida = a;
46 }
47 public void setFila(int n){
48     posFila = n;
49 }
50 public void setCol(char n){
51     posCol = n;
52 }
53 //tres constructores sobrecargados
54 public Soldado(String nom, char fig){
55     nombre = nom;
56     nivelAtaque = (int)(Math.random() * 5 + 1);
57     nivelDefensa = (int)(Math.random() * 5 + 1);
58     nivelVida = (int)(Math.random() * 5 + 1);
59     vidaActual = nivelVida;
60     velocidad = 0;
61     actitud = "Defensiva";
62     vive = true;
63     posFila = (int)(Math.random() * 10 + 1);
64     posCol = numCol();
65     figura = fig;
66 }
67 public Soldado(String nom, int nivAtaq, int nivDef, int vid, char fig){
68     nombre = nom;
69     nivelAtaque = nivAtaq;
70     nivelDefensa = nivDef;
71     nivelVida = vid;
72     vidaActual = vid;
73     velocidad = 0;
74     actitud = "Defensiva";
75     vive = true;
76     posFila = (int)(Math.random() * 10 + 1);
77     posCol = numCol();
78     figura = fig;
79 }
80 public Soldado(String nom, int nivAtaq, int nivDef, int nivVida, int nivAct, int vel,
81     String act, boolean vivir, int pFila, char pCol, char fig){
82     nombre = nom;
83     nivelAtaque = nivAtaq;
84     nivelDefensa = nivDef;
85     nivelVida = nivVida;
86     vidaActual = nivAct;
87     velocidad = vel;
88     actitud = act;
89     vive = vivir;
90     posFila = pFila;
91     posCol = pCol;
92     figura = fig;
93 }

```

```
*Soldado.java x Videojuego.java
85     nivelVida = nivVida;
86     vidaActual = nivAct;
87     velocidad = vel;
88     actitud = act;
89     vive = vivir;
90     posFila = pFila;
91     posCol = pCol;
92     figura = fig;
93 }
94 public static char numCol() {
95     String a = "abcdefghij";
96     int n = (int)(Math.random() * a.length());
97     char car = a.charAt(n);
98     return car;
99 }
100 public String toString(){
101     return "Nombre: " + nombre + " Vida:" + vidaActual + " Fila:" + posFila +
102           " Columna:" + posCol + " Actitud:" + actitud;
103 }
104 //gets
105 public String getNombre(){
106     return nombre;
107 }
108 public int getAtaque(){
109     return nivelAtaque;
110 }
111 public int getDefensa(){
112     return nivelDefensa;
113 }
114 public int getNivVidAct(){
115     return vidaActual;
116 }
117 public int getVida(){
118     return nivelVida;
119 }
120 public int getVelocidad(){
121     return velocidad;
122 }
123 public String getActitud(){
124     return actitud;
125 }
126 public boolean getVive(){
127     return vive;
128 }
129 public int getFila(){
130     return posFila;
131 }
132 public char getColumna(){
133     return posCol;
134 }
135 public char getFigura(){
136     return figura;
137 }
138 }
```

```

1 import java.util.*;
2 public class Videojuego{
3     public static void main(String[] args){
4         Scanner sc = new Scanner(System.in);
5         char[][] tablero = {{
6             {'.', '.', '.', '.', '.', '.', '.', '.', '.', '.'},
7             {'-', '-', '-', '-', '-', '-', '-', '-', '-', '-'},
8             {'.', '.', '.', '.', '.', '.', '.', '.', '.', '.'},
9             {'-', '-', '-', '-', '-', '-', '-', '-', '-', '-'},
10            {'.', '.', '.', '.', '.', '.', '.', '.', '.', '.'},
11            {'-', '-', '-', '-', '-', '-', '-', '-', '-', '-'},
12            {'.', '.', '.', '.', '.', '.', '.', '.', '.', '.'},
13            {'-', '-', '-', '-', '-', '-', '-', '-', '-', '-'},
14            {'.', '.', '.', '.', '.', '.', '.', '.', '.', '.'}}};
15 ArrayList<Soldado> ejercito_1 = new ArrayList<Soldado>();
16 datosEjercito(1, ejercito_1, '%');
17 ArrayList<Soldado> ejercito_2 = new ArrayList<Soldado>();
18 datosEjercito(2, ejercito_2, '*');
19 actTablero(ejercito_1, tablero);
20 actTablero(ejercito_2, tablero);
21 ArrayList<Soldado> ejercito1_cop1 = new ArrayList<Soldado>();
22 ArrayList<Soldado> ejercito1_cop2 = new ArrayList<Soldado>();
23 ArrayList<Soldado> ejercito2_cop1 = new ArrayList<Soldado>();
24 ArrayList<Soldado> ejercito2_cop2 = new ArrayList<Soldado>();
25 boolean continuar = true;
26 System.out.println("Opcion 1: Imprimir Tablero \nOpcion 2: Soldado con mayor vida del ejercito1\n" +
27 "Opcion 3: Soldado con mayor vida del ejercito2\nOpcion 4: Promedio de vida del ejercito 1\n" +
28 "Opcion 5: Promedio de vida del ejercito 2\nOpcion 6: Mostrar datos del ejercito1\n" +
29 "Opcion 7: Mostrar datos del ejercito2\n" + "Opcion 8: 1er metodo de orden Ejercito 1\n" +
30 "Opcion 9: 2do metodo de orden Ejercito 1\n" + "Opcion 10: 1er metodo de orden Ejercito 1\n" +
31 "Opcion 11: 2do metodo de orden Ejercito 2\n" + "Opcion 12: Ejercito ganador\n" +
32 "Opcion 13: Metodo serAtacado al 1er Ejercito\n" + "Opcion 14: Metodo serAtacado al 2do ejercito\n" +
33 "Opcion 15: Probabilidad de Rol por cada ejercito\n" +
34 "Opcion 16: Salir");
35 while(continuar){
36     System.out.println("*****");
37     System.out.println("Ingrese una opcion");
38     int opcion = sc.nextInt();
39     System.out.println("*****");
40     switch(opcion){
41         case 1:
42             System.out.println("El tablero es ");
43             impTablero(tablero);

```

```
*Soldado.java  Videojuego.java x
43         impTablero(tablero);
44         break;
45     case 2:
46         mayorVida(ejercito_1, 1);
47         break;
48     case 3:
49         mayorVida(ejercito_2, 2);
50         break;
51     case 4:
52         promVida(ejercito_1, 1);
53         break;
54     case 5:
55         promVida(ejercito_2, 2);
56         break;
57     case 6:
58         System.out.println("Los datos del ejercito 1 son : ");
59         mostDatos(ejercito_1);
60         break;
61     case 7:
62         System.out.println("Los datos del ejercito 2 son : ");
63         mostDatos(ejercito_2);
64         break;
65     case 8:
66         System.out.println("Primer metodo de orden - Ejercito 1");
67         copiaDatos(ejercito_1, ejercito1_cop1);
68         ordenarBurbuja(ejercito1_cop1);
69         ejercito1_cop1.clear();
70         break;
71     case 9:
72         System.out.println("2do metodo de orden - Ejercito 1");
73         copiaDatos(ejercito_1, ejercito1_cop2);
74         ordenarSeleccion(ejercito1_cop2);
75         ejercito1_cop2.clear();
76         break;
77     case 10:
78         System.out.println("1er metodo de orden - Ejercito 2");
79         copiaDatos(ejercito_2, ejercito2_cop1);
80         ordenarBurbuja(ejercito2_cop1);
81         ejercito2_cop1.clear();
82         break;
83     case 11:
84         System.out.println("2do metodo de orden - Ejercito 2");
85         copiaDatos(ejercito_2, ejercito2_cop2);
```

```

85      copiaDatos(ejercito_2, ejercito2_cop2);
86      ordenarSeleccion(ejercito2_cop2);
87      ejercito2_cop2.clear();
88      break;
89      case 12:
90          System.out.println("El ganador entre ambos ejercitos es :");
91          ganBatala(ejercito_1, ejercito_2);
92          break;
93      case 13:
94          System.out.println("Sufrieron ataques del ejercito 1:");
95          probAtacado(ejercito_1);
96          break;
97      case 14:
98          System.out.println("Sufrieron ataques del ejercito 2:");
99          probAtacado(ejercito_2);
100         break;
101     case 15:
102         System.out.println("Rol del ejercito 1:");
103         probRol(ejercito_1);
104         System.out.println("Rol del ejercito 2:");
105         probRol(ejercito_2);
106         break;
107     case 16:
108         System.out.println("Saliendo del programa");
109         continuar = false;
110         break;
111     default:
112         System.out.println("Opcion no valida");
113     }
114 }
115 }
116 public static void datosEjercito(int n, ArrayList<Soldado> ejercito, char fig){
117     int nroSoldados = (int)(Math.random() * 10 + 1);
118     System.out.println("El ejercito " + n + " tiene un total de " + nroSoldados + " soldados");
119     for(int i = 0; i < nroSoldados; i++){
120         String nombre = "Soldado " + (i + 1) + "X" + n;
121         ejercito.add(new Soldado(nombre, fig));
122     }
123 }
124 }
125 public static void actTablero(ArrayList<Soldado> ejercito, char[][] tablero){
126     for(int i = 0; i < ejercito.size(); i++){
127         int fila = ejercito.get(i).getFila();

```

```
*Soldado.java  Videojuego.java X
127     int fila = ejercito.get(i).getFila();
128     char columna = ejercito.get(i).getColumna();
129     if(tablero[fila - 1][nroColumna(columna)] != '-'){
130         fila = (int)(Math.random() * 10 + 1);
131         columna = Soldado.numCol();
132     }
133     ejercito.get(i).setFila(fila);
134     ejercito.get(i).setCol(columna);
135     tablero[fila - 1][nroColumna(columna)] = ejercito.get(i).getFigura();
136 }
137 }
138 public static int nroColumna(char n){
139     switch(n){
140         case 'a': return 0;
141         case 'b': return 1;
142         case 'c': return 2;
143         case 'd': return 3;
144         case 'e': return 4;
145         case 'f': return 5;
146         case 'g': return 6;
147         case 'h': return 7;
148         case 'i': return 8;
149         case 'j': return 9;
150         default: return 0;
151     }
152 }
153 public static void impTablero(char[][] tablero){
154     System.out.println("El tablero es :");
155     for(int x = 0; x < tablero.length; x++){
156         for(int y = 0; y < tablero[x].length; y++){
157             System.out.print(tablero[x][y]);
158         }
159         System.out.println();
160     }
161 }
162 public static void mayorVida(ArrayList<Soldado> ejercito, int n){
163     int mayor = ejercito.get(0).getVida();
164     for(int i = 1; i < ejercito.size(); i++){
165         if(ejercito.get(i).getVida() > mayor)
166             mayor = ejercito.get(i).getVida();
167     }
168     System.out.println("Los soldados con mayor vida del ejercito " + n + " son:");
169     for(int j = 0; j < ejercito.size(); j++){
170         if(ejercito.get(j).getVida() == mayor){
171             System.out.println(ejercito.get(j));
172         }
173     }
174 }
175 public static void promVida(ArrayList<Soldado> ejercito, int m){
176     double n = 0;
177     for(int i = 0; i < ejercito.size(); i++){
178         n += ejercito.get(i).getVida();
179     }
180     System.out.println("El promedio de vida del ejercito " + m + " es " + n);
181 }
182 public static void copiaDatos(ArrayList<Soldado> ejercito, ArrayList<Soldado> copia){
183     for(int i = 0; i < ejercito.size(); i++){
184         Soldado pos = ejercito.get(i);
185         Soldado c = new Soldado(pos.getNombre(), pos.getAtaque(), pos.getDefensa(), pos.getVida(), pos.getNivVidAct(), pos.getVeloci
186         copia.add(c);
187     }
188 }
```



```

189 public static void ordenarSeleccion(ArrayList<Soldado> ejercito){
190     for(int i = 0; i < ejercito.size() - 1; i++){
191         for(int j = i + 1; j < ejercito.size(); j++){
192             Soldado may = ejercito.get(i);
193             Soldado men = ejercito.get(j);
194             if(men.getVida() > may.getVida()){
195                 Soldado mayor = new Soldado(men.getNombre(), men.getAtaque(), men.getDefensa(), men.getVida(), men.getNivVidAct(), n
196                 Soldado menor = new Soldado(may.getNombre(), may.getAtaque(), may.getDefensa(), may.getVida(), may.getNivVidAct(), n
197                 ejercito.set(i, mayor);
198                 ejercito.set(j, menor);
199             }
200         }
201     }
202     mostDatos(ejercito);
203 }

```

---

```

203 }
204 public static void ordenarBurbuja(ArrayList<Soldado> ejercito){
205     for(int i = 0; i < ejercito.size() - 1; i++){
206         for(int j = 0; j < ejercito.size() - 1; j++){
207             Soldado may = ejercito.get(i);
208             Soldado men = ejercito.get(i + 1);
209             if(men.getVida() > may.getVida()){
210                 Soldado mayor = new Soldado(men.getNombre(), men.getAtaque(), men.getDefensa(), men.getVida(), men.getNivVidAct(), n
211                 Soldado menor = new Soldado(may.getNombre(), may.getAtaque(), may.getDefensa(), may.getVida(), may.getNivVidAct(), n
212                 ejercito.set(j, mayor);
213                 ejercito.set(j + 1, menor);
214             }
215         }
216     }
217     mostDatos(ejercito);
218 }
219 public static void mostDatos(ArrayList<Soldado> ejercito){
220     for(Soldado n: ejercito)
221         System.out.println(n);
222 }
223 public static void probAtacado(ArrayList<Soldado> ejercito){
224     for(int i = 0; i < ejercito.size(); i++){
225         int n = (int)(Math.random() * 2);
226         if(n == 0){
227             System.out.println("El soldado " + ejercito.get(i).getNombre() + " fue atacado");
228             ejercito.get(i).serAtacado();
229             System.out.println("Tiene de " + ejercito.get(i).getNivVidAct() + " puntos de vida");
230         }
231     }
232 }
233 public static void probRol(ArrayList<Soldado> ejercito){
234     for(int i = 0; i < ejercito.size(); i++){
235         int n = (int)(Math.random() * 3);
236         if(n == 0){
237             System.out.println("El soldado " + ejercito.get(i).getNombre() + " atacara");
238             ejercito.get(i).atacar();
239         } else if( n == 1){
240             System.out.println("El soldado " + ejercito.get(i).getNombre() + " defendera");
241             ejercito.get(i).defender();
242         } else{
243             System.out.println("El soldado " + ejercito.get(i).getNombre() + " huida");
244             ejercito.get(i).huir();
245         }
246     }
247 }

```

	<p align="center"><b>UNIVERSIDAD NACIONAL DE SAN AGUSTIN</b>  <b>FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS</b>  <b>ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</b></p>	
<p align="center"><b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p><b>Aprobación:</b> 2022/03/01</p>	<p><b>Código:</b> GUIA-PRLE-001</p>	<p align="right"><b>Página:</b> 10</p>

```


245     }
246   }
247 }
248 public static void ganBatalla(ArrayList<Soldado> ejercito1, ArrayList<Soldado> ejercito2){
249     int suma1 = 0, suma2 = 0;
250     for(int i = 0; i < ejercito1.size(); i++){
251         suma1 += ejercito1.get(i).getVida();
252     }
253     for(int j = 0 ; j < ejercito2.size(); j++){
254         suma2 += ejercito2.get(j).getVida();
255     }
256     if(suma1 > suma2){
257         System.out.println("El ganador es el ejercito 1");
258     } else if( suma2 > suma1){
259         System.out.println("El ganador es el ejercito 2");
260     } else {
261         System.out.println("Es un empate");
262     }
263 }
264 }

```

```

Console × Search
Videojuego [Java Application] C:\Users\UJKMjuandi\Downloads\eclipse-java-2023-03-R-
El ejercito 1 tiene un total de 3 soldados
El ejercito 2 tiene un total de 1 soldados
Opcion 1: Imprimir Tablero
Opcion 2: Soldado con mayor vida del ejercito1
Opcion 3: Soldado con mayor vida del ejercito2
Opcion 4: Promedio de vida del ejercito 1
Opcion 5: Promedio de vida del ejercito 2
Opcion 6: Mostrar datos del ejercito1
Opcion 7: Mostrar datos del ejercito2
Opcion 8: 1er metodo de orden Ejercito 1
Opcion 9: 2do metodo de orden Ejercito 1
Opcion 10: 1er metodo de orden Ejercito 1
Opcion 11: 2do metodo de orden Ejercito 2
Opcion 12: Ejercito ganador
Opcion 13: Metodo serAtacado al 1er Ejercito
Opcion 14: Metodo serAtacado al 2do ejercito
Opcion 15: Probabilidad de Rol por cada ejercito
Opcion 16: Salir
*****
Ingrese una opcion
1
*****
El tablero es
El tablero es :
-----
-----
%------
------%
-----
-----
-----
-----
-----%-----
*-----
*****
Ingrese una opcion

```

	<p align="center"><b>UNIVERSIDAD NACIONAL DE SAN AGUSTIN</b>  <b>FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS</b>  <b>ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</b></p>	
<b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		
<b>Aprobación:</b> 2022/03/01	<b>Código:</b> GUIA-PRLE-001	<b>Página:</b> 12

**III. CUESTIONARIO:**

REPOSITORIO: <https://github.com/UJKMjuandi/FundamentosP2>

CONCLUSIONES
<p>La práctica de seguir un diagrama UML y ampliar las clases con nuevos métodos ha sido una experiencia enriquecedora que ha fortalecido mis conocimientos teóricos y ha proporcionado una valiosa aplicación práctica. En primer lugar, la comprensión del diagrama UML fue crucial para analizar las relaciones entre clases, atributos y métodos, estableciendo así la base conceptual del programa.</p> <p>La implementación del código basado en el diagrama UML no solo requirió precisión en la sintaxis, sino también una cuidadosa consideración de las relaciones de asociación entre clases. La fase de ampliación con nuevos métodos permitió explorar la extensión de la funcionalidad del sistema de acuerdo con requisitos específicos. La toma de decisiones sobre qué métodos agregar y cómo interactuarían con las clases existentes fue una tarea</p>

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

desafiante pero esencial.

Al encontrarme con desafíos durante la implementación, como la gestión adecuada de la visibilidad de los atributos y la elección de las estructuras de datos correctas, desarrollé un pensamiento analítico y una comprensión más profunda de los conceptos de programación orientada a objetos.

La fase de pruebas y depuración destacó la importancia de validar la robustez del programa a través de una variedad de escenarios. Este proceso no solo garantizó el correcto funcionamiento del código, sino que también ofreció oportunidades de aprendizaje continuo y mejoras constantes.

Además, la práctica fomentó la colaboración y la retroalimentación. La interacción con compañeros de clase o revisores externos proporcionó perspectivas valiosas que contribuyeron a la mejora del diseño y la legibilidad del código.

### METODOLOGÍA DE TRABAJO

- 1.- Leer los enunciados cuidadosamente para saber que requiere el problemas.
- 2.- verificar el diagrama uml puesto en la practica
- 3.-Empezar a hacer las clase soldado con el uml
- 4.- Realizar el Videojuego con los métodos.

### REFERENCIAS Y BIBLIOGRAFÍA

[1] M. A. Lopez, E. Castro Gutierrez, *Fundamentos de la Programación 2 Topicos de Programación Orientada a Objetos*. Arequipa: UNAS, 2021

### RUBRICA PARA EL CONTENIDO DEL INFORME Y DEMOSTRACIÓN

El alumno debe marcar o dejar en blanco en celdas de la columna Checklist si cumplió 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 ítems.

El alumno debe autocalificarse en la columna Estudiante de acuerdo a la siguiente tabla:

Tabla 1: 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

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

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	1	
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 están dentro de los plazos de fecha de entrega establecidos.	2	X	0	
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		15	

Tabla 2: Rúbrica para contenido del Informe y demostración