
	<b>Curso</b>	DAM on-line			
	<b>Módulo</b>	PROG. SERVICIOS Y PROCESOS			
	<b>Fecha</b>	10-06-2020	<b>Duración</b>	90 minutos	
	<b>Parcial</b>	2 (JUNIO)	<b>Temas</b>	4, 5 y 6	

### Instrucciones generales

El examen del 2º parcial de junio se compone de dos partes:

- Parte preguntas 4 puntos.
- Parte práctica 6 puntos.

La entrega del examen se realizara en un único fichero zip utilizando la plataforma avanza. El nombre del fichero será **apellido1\_apellido2\_nombre\_PSP\_2Parcial\_Junio** y contendrá el documento de texto con las respuestas a las preguntas más el proyecto realizado en netbeans.

Criterios de calificación	Puntuación
Pregunta 1	2 puntos
Pregunta 2	2 puntos
Ejercicio Práctico: <b>Clase Servidor</b>	
Programa principal	1 puntos
Login	2 puntos
Configurar juego	1,5 puntos
Jugar	1,5 puntos
	-----
	10 puntos

<b>Alumno/a:</b>	PC	<b>Total</b>
------------------	----	--------------

**Pregunta 1 Trabajando con URL.** Explica que hace el siguiente código, así como el resultado de su ejecución

```

1 import java.io.IOException;
2 import java.util.Date;
3 import java.net.URL;
4 import java.net.URLConnection;
5 import java.net.MalformedURLException;
6
7 public class Main {
8     public static void main(String[] args) throws IOException {
9         URL aURL = null;
10        try {
11
12            aURL = new URL("http://www.alferez.es/documentos/Jboss.pdf");
13            URLConnection conexion = aURL.openConnection();
14            conexion.connect();
15            System.out.println("1.- " + aURL.getProtocol());
16            System.out.println("2.- " + aURL.getHost());
17            System.out.println("3.- " + aURL.getFile());
18            System.out.println("4.- " + aURL.getPort());
19            System.out.println("5.- " + conexion.getContentType());
20            System.out.println("6.- " + new Date( conexion.getLastModified() ) );
21
22        } catch (MalformedURLException e) {
23            System.out.println("MalformedURLException: " + e);
24        }
25    }
26 }

```

**Pregunta 2 En relación a los ficheros de registros o logs.** ¿Cómo se realiza el registro de actividad? ¿Qué clases e instrucciones debemos utilizar?

## Ejercicio práctico: juegoAdivinanza. (6 puntos)

Un **servidor** almacenara una adivinanza y su solución. Este servidor debe permitir las conexiones de múltiples clientes y realizara lo siguiente:

Solicitará al cliente un nombre de **usuario o login** si es “Administrador” le permitirá configurar el juego solicitándole la adivinanza y la solución, pasando después a jugar.

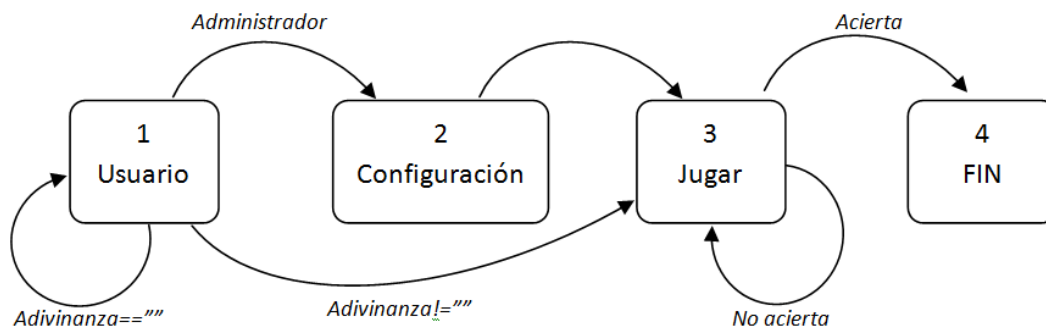
- Si no es “Administrador” y la adivinanza esta vacía le avisara que no esta configurado y volverá a solicitar login.
- Si no es “Administrador” y la adivinanza esta configurada pasara a jugar

Cuando se esta **jugando** el servidor le manda a los clientes la adivinanza y estos la deben resolver.

- Si no la acierta el servidor volverá a mandarle la adivinanza.
- Si la acierta le mostrará el mensaje de felicidades, y ese cliente finalizará.

Mientras pueden seguir jugando el resto de clientes.

Posible diagrama de estados:



Ejemplos de ejecución de los clientes:

### Cliente 1

```
run:
Introduce el usuario
Administrador
El Administrador va a configura el juego
Introduce la adivinanza:
Oro parece, plata no es. Que es?
Continuamos...
Introduce la solución:
platano
Configurado
Oro parece, plata no es. Que es?
```

### Cliente 2

```
run:
Introduce el usuario
Paco
Comienza el juego
Oro parece, plata no es. Que es?
limon
vuelve a intentarlo
Oro parece, plata no es. Que es?
platano
Felicidades
BUILD SUCCESSFUL (total time: 55 seconds)
```

Se facilita el código del cliente

```
Cliente.java
Source History
1 import java.io.DataInputStream;
2 import java.io.DataOutputStream;
3 import java.io.IOException;
4 import java.io.InputStream;
5 import java.io.OutputStream;
6 import java.net.Socket;
7 import java.util.Scanner;
8
9 public class Cliente {
10     static final String HOST = "localhost";
11     static final int Puerto=4000;
12
13     public Cliente() throws IOException {
14         Scanner scanner = new Scanner(System.in);
15         Socket sCliente = new Socket( HOST , Puerto );
16
17         DataInputStream dataIS=new DataInputStream(sCliente.getInputStream());
18         DataOutputStream dataOS= new DataOutputStream(sCliente.getOutputStream());
19
20         String respuesta="", retorno="";
21         boolean fin = false;
22
23         while(!respuesta.equals("Felicidades")){
24             respuesta = dataIS.readUTF();
25             System.out.println(respuesta);
26
27             retorno = scanner.nextLine();
28             dataOS.writeUTF(retorno);
29
30             respuesta = dataIS.readUTF();
31             System.out.println(respuesta);
32         } //FIN
33         sCliente.close();
34     }
35     public static void main( String[] arg ) throws IOException {
36         new Cliente();
37     }
38 }
```

#### Criterios de calificación. Práctica Servidor

Programa principal	1 puntos
Login	2 puntos
Configurar juego	1,5 puntos
Jugar	1,5 puntos