



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



ESCUELA TÉCNICA  
SUPERIOR DE  
INGENIEROS DE  
TELECOMUNICACIÓN

## **Desarrollo de una aplicación web de juego trivial basado en Servlets, Java, JavaScript y HTML5**

**Autor: Laghmari Ibtissam**

**Tutor: López Patiño José Enrique**

Trabajo Fin de Grado presentado en la Escuela Técnica Superior de Ingenieros de Telecomunicación de la Universitat Politècnica de València, para la obtención del Título de Graduado en Ingeniería de Tecnologías y Servicios de Telecomunicación

Curso 2015-16

Valencia, 13 de septiembre de 2016

## **Resumen**

El proyecto consiste en el desarrollo de un juego basado en preguntas y respuestas tipo test, en el que dos jugadores deben responder a preguntas de diferentes categorías, las preguntas están clasificadas en 6 categorías, de forma que cada pregunta pertenece a una única categoría y tiene 4 posibles respuestas de las que solo una es correcta. Cada partida es jugada por dos usuarios que se desplazan por un tablero por turnos alternativos dependiendo de un dado. El tablero contiene casillas de 3 tipos: “pregunta”, “estrella”, “volver a tirar”. Si un jugador acierta una pregunta mantiene su turno, si además la pregunta es estrella la gana, el primer jugador que consigue las 6 estrellas gana el juego.

El juego se desarrolla sobre una plataforma Web, en el que el software del cliente está desarrollado en HTML5, JavaScript, CSS3 y el de servidor en Java, utilizando las librerías Servlet y JDBC.

El documento que se encuentra a continuación recoge una explicación de los objetivos del proyecto, de las tecnologías utilizadas, un análisis de las fases del proyecto, un diseño de la aplicación y un manual de uso.

## **Resum**

El projecte consisteix en el desenvolupament d'un joc basat en preguntes i respostes tipus test en què dos jugadors han de respondre a preguntes de diferents categories. Les preguntes estan classificades en 6 categories de manera que cada pregunta pertany a una única categoria i té 4 possibles respostes de les quals només una és correcta. Cada partida és jugada per dos usuaris que es desplacen per un tauler per torns alternatius depenent d'un donat. El tauler conté caselles de 3 tipus: "pregunta", "estrella", "tornar a tirar". Si un jugador encerta una pregunta manté el seu torn, si a ademes la pregunta és estrella la guanya, el primer jugador que aconsegueix les 6 estrelles guanya el joc.

El joc es desenvolupa sobre una plataforma web, en el qual el programari del client està desenvolupat en HTML5, JavaScript, CSS3 i el de servidor en Java, utilitzant les llibreries Servlet i JDBC.

El document que es troba a continuació recull una explicació dels objectius del projecte, de les tecnologies utilitzades, un anàlisi de les fases del projecte, un disseny de l'aplicació i un manual d'ús.

## **Abstract**

The project involves the development of a game based on question and answer type test in which two players must answer questions from different categories, questions are classified into 6 categories with each one having 4 possible answers while only one of them is correct. Each game is played by two users who move around a board by alternative shifts depending on a dice. The board contains 3 types of boxes: "question", "star", "pull back". if the player answers correctly he keeps playing, if the question is a star he wins this star, the first player to get 6 stars wins the game.

The game takes place on a Web platform, in which the client software is developed in HTML5, JavaScript, CSS3 and server in Java, using the Servlet and JDBC libraries.

The document includes a detailed explanation of the project objectives, the technology used, an analysis phase, as well as a design application and a user's guides.

## **Palabras clave**

- Servlet
- Java
- Jugador
- Partidas
- Categorías
- Estrellas
- Estadísticas
- Web
- HTML5
- JDBC,
- MySQL
- JavaScript
- jQuery
- CSS3

## **Keyword**

- Servlet
- Java
- Player
- Games
- Category
- Stars
- Statistics
- Web
- HTML5
- JDBC,
- MySQL
- JavaScript
- jQuery
- CSS3

## INDICE

INDICE DE FIGURAS .....	7
INDICE DE TABLAS.....	8
1 Introducción.....	9
1.1 Aplicación web.....	9
1.2 Objetivos .....	9
1.2.1 <i>Objetivos respecto al proyecto</i> .....	9
1.2.2 <i>Objetivos personales</i> .....	9
2 Metodología.....	9
2.1 Distribución de las diferentes tareas realizadas .....	10
3 Tecnologías y Herramientas usadas .....	11
3.1 Herramientas .....	11
3.2 Tecnologías .....	17
4 Análisis.....	24
4.1 Requisitos.....	24
4.2 Base de Datos .....	24
4.3 Aplicación .....	24
4.4 Interfaces .....	27
5 Diseño de la aplicación.....	28
5.1 Diseño base de datos .....	28
5.2 Diseño de la aplicación.....	34
6 Conclusiones .....	42
7 Propuesta de trabajo futuro.....	42
8 Bibliografía.....	43
9 Anexo .....	44
9.1 Manual de usuario .....	44

## INDICE DE FIGURAS

Figura 1. Fichero WEB.XML.....	12
Figura 2. Ficheros Servlets.....	13
Figura 3 : Estructura directorios en Tomcat.....	13
Figura 4: Panel de Control Xampp.....	14
Figura 5: Conexión desde el navegador.....	14
Figura 6: Anadir librería Servlet.....	15
Figura 7: Servlets en Bluej.....	16
Figura 8: Base de datos en Mysql Workbench.....	16
Figura 9: Creación de la tabla estadísticas con Kompozer.....	17
Figura 10: Ejemplo de uso de Servlet.....	18
Figura 11: Conexión a la base de datos con JDBC.....	19
Figura 12: Creación del formulario crear cuenta con HTML.....	20
Figura 13: Ficheros CSS.....	21
Figura 14: Llamada a los ficheros CSS.....	21
Figura 15: Ejemplo Código CSS.....	21
Figura 16: Función JavaScript Lanzar dado.....	22
Figura 17: Función JQuery Popup.....	23
Figura 18: Esquema de autenticación de jugadores.....	25
Figura 19: Menú del juego.....	26
Figura 20: Funcionamiento del juego.....	27
Figura 21: Base de datos.....	28
Figura 22: Tabla de usuarios.....	29
Figura 23: Tabla de preguntas.....	30
Figura 24: Tabla de respuestas.....	30
Figura 25: Introducción preguntas y respuestas.....	31
Figura 26: Tabla de categorías.....	31
Figura 27: Inserto categorías.....	32
Figura 28: Tabla de partidas.....	32
Figura 29: Tabla guardar.....	33
Figura 30: Tabla de estrellas.....	33
Figura 31: Tabla de estadísticas.....	34
Figura 32: Código HTML e interfaz de la Index.html.....	35
Figura 33: Interfaces de autenticación del jugador 1.....	35
Figura 34: Interfaces de autenticación del jugador 2.....	36
Figura 35: Interfaz del menú principal.....	36
Figura 36: Lista de partidas existentes.....	37
Figura 37: Interfaces para ver las estadísticas.....	38
Figura 38: Funcionamiento del servlet lanzardado.....	38
Figura 39: Posiciones del jugador.....	39
Figura 40: Elección de categoría.....	39
Figura 41: Icono del turno del jugador.....	39
Figura 42: Pregunta en Popup.....	40
Figura 43: Verificación de respuestas.....	40
Figura 44: Fin del Juego.....	41

Figura 45: Acceder como usuario 1.....	44
Figura 46: Campo vacío.....	45
Figura 47: Información incorrecta.....	45
Figura 48: Crear cuenta usuario 1.....	46
Figura 49: Campo vacío usuario registrado.....	47
Figura 50: Información incorrecta usuario registrado.....	47
Figura 51: Crear cuenta usuario 2.....	48
Figura 52: Usuario 2 registrado.....	48
Figura 53: Menú.....	49
Figura 54: Tablero del juego.....	50
Figura 55: Lanzar el dado.....	51
Figura 56: Elegir entre dos casillas.....	51
Figura 57 : Elegir categoría.....	51
Figura 58: Caso de respuesta verdadera.....	52
Figura 59: Caso de respuesta falsa.....	53
Figura 60: Jugador 2 ha ganado una estrella.....	53
Figura 61: Fin de la partida.....	54
Figura 62: Partidas existentes.....	54
Figura 63: Estadísticas por jugador.....	55
Figura 64: Lista de partidas para estadísticas.....	56
Figura 65: Estadísticas de una partida.....	57

## INDICE DE TABLAS

Tabla 1. Organización temporal de las tareas.....	10
---	----

# 1 Introducción

El presente proyecto Trabajo fin de Grado, Consta de la realización de una aplicación Web de un juego trivial en la que hay preguntas y respuestas tipo test en el que dos jugadores deben responder a preguntas de diferentes categorías, las preguntas están clasificadas en 6 categorías de forma que cada pregunta pertenece a una única categoría y tiene 4 posibles respuestas de las que solo una es correcta. Cada partida es jugada por dos usuarios que se desplazan por un tablero por turnos alternativos dependiendo de un dado. El tablero contiene casillas de 3 tipos: “pregunta”, “estrella”, “volver a tirar”. Si un jugador acierta una pregunta mantiene su turno, si además la pregunta es estrella la gana, el primer jugador que consigue las 6 estrellas gana el juego.

La aplicación se desarrolló con las tecnologías Java, JavaScript, HTML5, CSS3, MySQL y el servidor local Xampp. La motivación de hacer este proyecto ha sido la aplicación de los conocimientos adquiridos en la asignatura de especialidad de telemática sistemas telemáticos de gestión de la información, así como adquirir nuevas competencias durante el proceso de desarrollo del proyecto.

## 1.1 Aplicación web

Una aplicación web es una aplicación desarrollada con lenguajes muy conocidos por los programadores, como HTML, CSS y JavaScript. Este tipo de aplicaciones no tiene la restricción de plataforma como ocurría en las aplicaciones nativas, ya que podemos programar independiente del sistema operativo en el que se usará la aplicación. De esta forma se pueden ejecutar en los diferentes dispositivos sin tener que crear varias aplicaciones. Este tipo de aplicaciones se ejecutan dentro del navegador web y el contenido se adapta al dispositivo simulando una aplicación cualquiera.

## 1.2 Objetivos

### 1.2.1 *Objetivos respecto al proyecto*

El proyecto tiene como objetivo principal desarrollar una aplicación de uso sencillo que permite a dos jugadores competir para ganar 6 estrellas, así como para aprender cosas nuevas en diferentes temas. El juego dispone de 6 categorías con dificultad distinta.

### 1.2.2 *Objetivos personales*

Este proyecto me permitirá profundizar en mis conocimientos adquiridos en la asignatura de especialidad de telemática sistemas telemáticos de gestión de la información, así como adquirir nuevas competencias durante el proceso de desarrollo del proyecto.

# 2 Metodología

## 2.1 Distribución de las diferentes tareas realizadas

A continuación, se detallan todas las tareas que se han realizado en este TFG.

- Tarea 1: Captura de requerimientos.
- Tarea2: Aprendizaje de las herramientas empleadas.
- Tarea 3: Aprendizaje de los lenguajes de programación usados HTML5, JavaScript, jQuery y CSS3.
- Tarea 4: Repaso de MySQL y Servlets adquiridos en la asignatura de sistemas telemáticos para la gestión de la información.
- Tarea 5: Diseño de la base de datos.
- Tarea 6: Creación de las primeras interfaces de la aplicación.
- Tarea 7: Creación de los primeros Servlets (Registro usuarios).
- Tarea 8: Programación del juego (tablero, dado, etc.) y posibilidad de jugar una partida con datos estáticos.
- Tarea 9: Creación de los Servlets necesarios para guardar los datos (Partidas, estadísticas, posiciones, estrellas, etc.) en la base de datos.
- Tarea 10: Corrección de fallos y dar estilo a la aplicación.

A continuación, se muestra una tabla del tiempo dedicados a cada tarea.

<b>Tareas</b>	<b>Duración</b>	<b>Inicio</b>	<b>Fin</b>
Tarea 1	8 Días	02/05/16	11/05/16
Tarea 2	4 Días	16/05/16	19/05/16
Tarea 3	15 Días	23/05/16	10/06/16
Tarea 4	3 Días	13/06/16	15/06/16
Tarea 5	3 Días	16/06/16	20/06/16
Tarea 6	5 Días	21/06/16	27/06/16
Tarea 7	4 Días	28/06/16	01/07/16
Tarea 8	15 Días	06/07/16	26/07/16
Tarea 9	15 Días	27/07/16	16/08/16
Tarea 10	8 Días	17/08/16	26/08/16

**Tabla 1. Organización temporal de las tareas.**

## 3 Tecnologías y Herramientas usadas

### 3.1 Herramientas

#### Xampp

Es una de las herramientas líderes para diseñadores y programadores dedicados a la web, ya que permite crear de forma sencilla y rápida un servidor local completo de los lenguajes de programación para web más usados (PHP, JSP), junto a MySQL, la base de datos que mejor se adapta al tipo de aplicaciones construidas con ellos. [1]

Xampp cuenta con el siguiente contenido:

- Apache
- MySQL
- PHP
- phpMyAdmin
- FileZilla FTP Server 0.9.39
- Tomcat

**Tomcat:** funciona como un contenedor de Servlets, puede funcionar como servidor web por sí mismo.[2]  
Todos los ficheros de la aplicación deben estar incluidos en la carpeta Tomcat en Webapps.

La aplicación Web se ha estructurado según la siguiente jerarquía de subdirectorios:

Directorio raíz: contiene ficheros estáticos (HTML, imágenes, hojas de estilo, etc.) y JSPs.

- Directorio WEB-INF: contiene un fichero web.xml. Este fichero configura la aplicación. Por ejemplo, permite declarar Servlets, asignarles parámetros de inicio, declarar alias y filtros, etc, (ver figura 1).
  - Directorio clases: contiene los ficheros compilados (Servlets) de las clases utilizadas por la aplicación web (ver figura 2).
  - Directorio lib: contiene el fichero .jar del conector con la base de datos que explicaré más adelante.

Resto de subdirectorios: para ficheros estáticos y JSP (ver figura 3).

```
C:\xampp\tomcat\webapps\juego_trivial\WEB-INF\web.xml - Notepad++
Fichier  Édition  Recherche  Affichage  Encodage  Langage  Paramétrage  Macro  Exécution  Compléme
new 1 x  index.html x  index.js x  web.xml x
1  <?xml version="1.0" encoding="ISO-8859-1"?>
2  <web-app>
3      <display-name>Introducirusuario</display-name>
4      <description>Introducir usuario </description>
5      <servlet>
6          <servlet-name>Introducirusuario</servlet-name>
7          <description> Servlet basico </description>
8          <servlet-class>Introducirusuario</servlet-class>
9      </servlet>
10
11     <display-name>Accedercomousuario</display-name>
12     <description>Accedercomousuario</description>
13     <servlet>
14         <servlet-name>Accedercomousuario</servlet-name>
15         <description> Servlet basico </description>
16         <servlet-class>Accedercomousuario</servlet-class>
17     </servlet>
18 <display-name>lanzardado</display-name>
19 <description>lanzardado </description>
20 <servlet>
21     <servlet-name>lanzardado</servlet-name>
22     <description> Servlet basico </description>
23     <servlet-class>lanzardado</servlet-class>
24 </servlet>
25 <display-name>sacarpreguntarespuestas</display-name>
26 <description>sacarpreguntarespuestas</description>
27 <servlet>
28     <servlet-name>sacarpreguntarespuestas</servlet-name>
29     <description> Servlet basico </description>
30     <servlet-class>sacarpreguntarespuestas</servlet-class>
31 </servlet>
32 <display-name>Partida</display-name>
33 <description>Partida</description>
34 <servlet>
35     <servlet-name>Partida</servlet-name>
36     <description> Servlet basico </description>
37     <servlet-class>Partida</servlet-class>
38 </servlet>
```

Figura 1. Fichero WEB.XML.

Nom	Modifié le	Type	Taille
package	11/09/2016 02:28	BlueJ Project File	3 Ko
Accedercomousuario	07/09/2016 13:24	Fichier JAVA	27 Ko
lanzardado	04/09/2016 14:12	Fichier CLASS	42 Ko
lanzardado	04/09/2016 14:12	Fichier CTXT	1 Ko
lanzardado	04/09/2016 14:12	Fichier JAVA	62 Ko
estadisticas	03/09/2016 19:37	Fichier CLASS	18 Ko
estadisticas	03/09/2016 19:37	Fichier CTXT	1 Ko
estadisticas	03/09/2016 19:37	Fichier JAVA	22 Ko
partidasexistentes	02/09/2016 20:19	Fichier CLASS	7 Ko
partidasexistentes	02/09/2016 20:19	Fichier CTXT	1 Ko
partidasexistentes	02/09/2016 20:19	Fichier JAVA	7 Ko
sacarpreguntarespuestas	02/09/2016 15:53	Fichier CLASS	13 Ko
sacarpreguntarespuestas	02/09/2016 15:53	Fichier CTXT	1 Ko
sacarpreguntarespuestas	02/09/2016 15:53	Fichier JAVA	21 Ko
Accedercomousuario	02/09/2016 15:34	Fichier CLASS	19 Ko
Accedercomousuario	02/09/2016 15:34	Fichier CTXT	1 Ko
insertarestrellas	28/08/2016 18:41	Fichier CLASS	5 Ko
insertarestrellas	28/08/2016 18:30	Fichier JAVA	5 Ko
insertarestadisticas	24/08/2016 17:34	Fichier CLASS	4 Ko
insertarestadisticas	24/08/2016 17:34	Fichier JAVA	4 Ko
Partida	23/08/2016 20:21	Fichier CLASS	3 Ko
Partida	23/08/2016 20:21	Fichier JAVA	3 Ko
Partida	12/08/2016 22:39	Fichier CTXT	1 Ko
README	07/03/2016 19:15	Document texte	1 Ko

Figura 2. Ficheros Servlets.

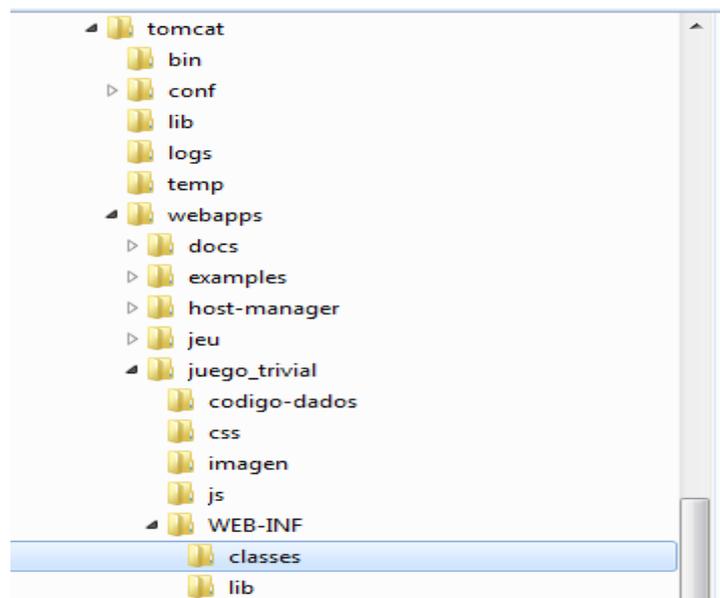


Figura 3 : Estructura directorios en Tomcat.

Para acceder a la aplicación lo primero es iniciar Apache, MySQL y Tomcat desde el panel de gestión de Xampp [3], comprobar el puerto en que está operativo el Tomcat (ver figura 4). Desde un navegador conectarse a la URL: `http://localhost:puerto/nombre_de_mi_carpeta/...` (Ver figura 5).

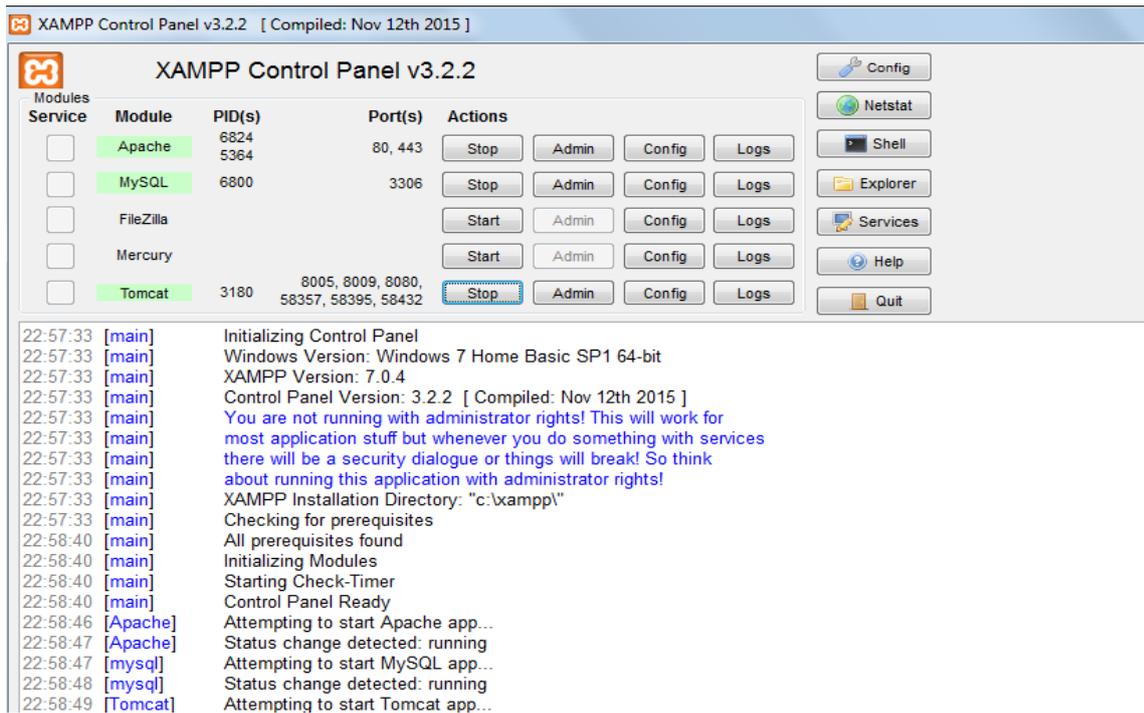


Figura 4: Panel de Control Xampp.

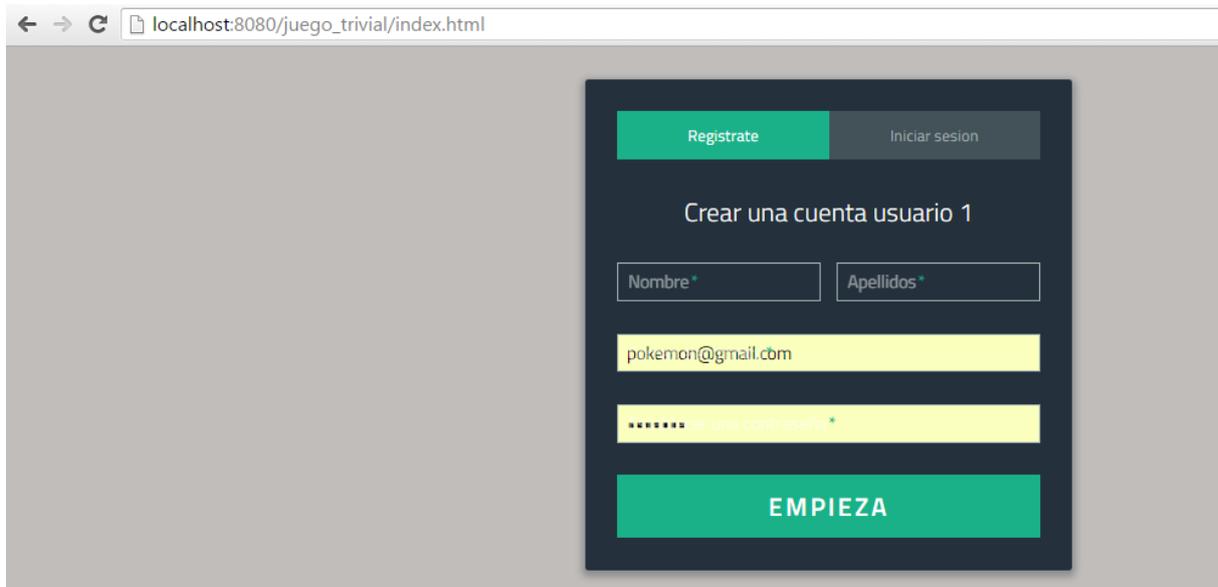


Figura 5: Conexión desde el navegador.

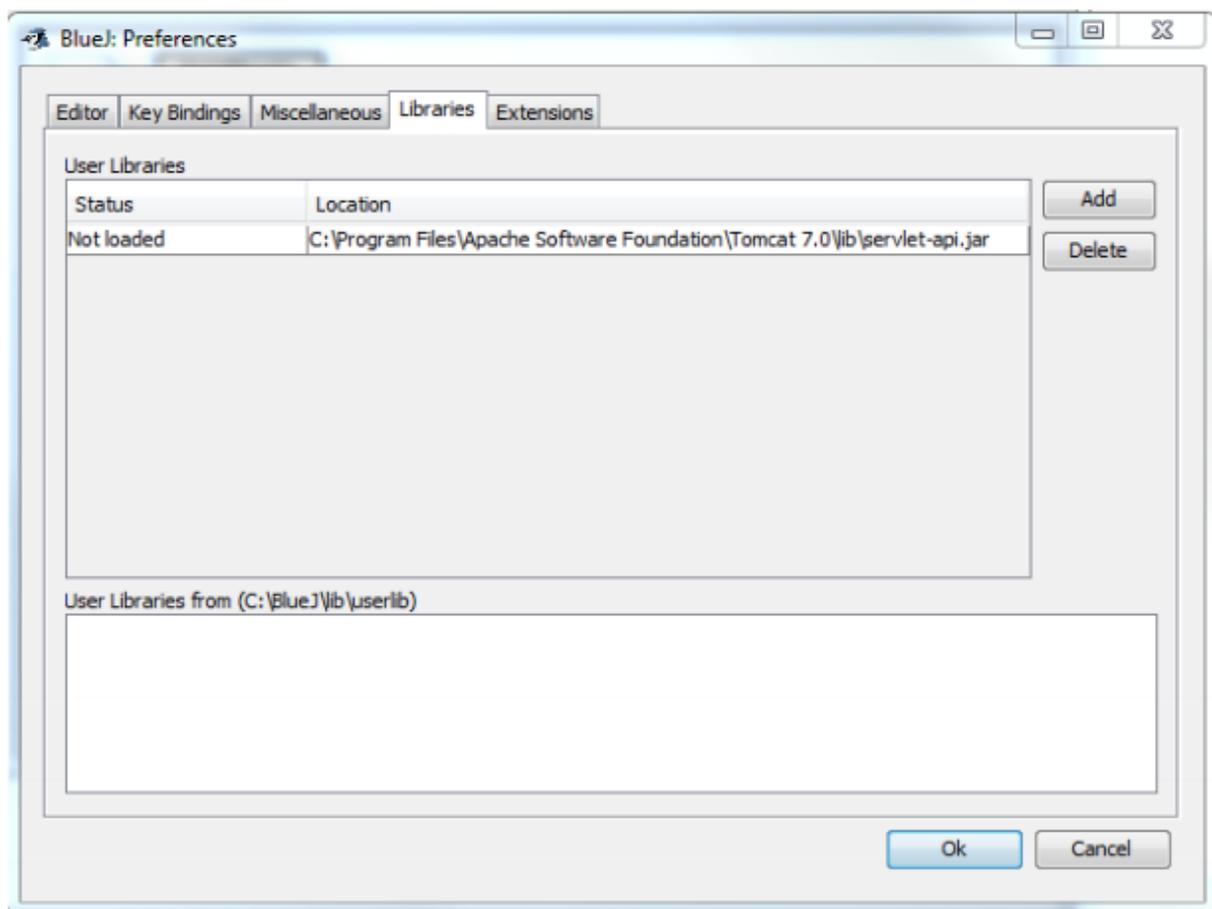
## Bluej

Es un entorno integrado de desarrollo para el lenguaje de programación Java, desarrollado principalmente con propósitos educativos, pero también es adecuado para el desarrollo de software a pequeña escala.

BlueJ fue desarrollado para apoyar la enseñanza y el aprendizaje de la programación orientada a objetos, y como resultado, su diseño difiere de otros entornos de desarrollo. La pantalla principal muestra gráficamente la estructura de clases de una aplicación en desarrollo (en un diagrama muy parecido a UML), y los objetos pueden ser creados y probados interactivamente. Combinado con una interfaz de usuario simple, esta facilidad de interacción permite experimentar de manera fácil con los objetos en desarrollo. Los conceptos de la orientación a objetos (clases, objetos, comunicación a través de llamadas a métodos) son representados visualmente en el diseño de interacción en la interfaz. [4]

En este proyecto se usó Bluej para la compilación de los Servlets que explicare más adelante. [5]

No obstante, como la librería Servlet no es estándar, se ha añadido en el conjunto de librerías que el BlueJ utiliza durante la compilación. Se añade el fichero .jar que contiene las clases de dicha librería. En concreto, este fichero es el fichero servlet-api.jar ubicado en el directorio /lib de la carpeta de instalación de Tomcat (ver figura 6).



**Figura 6: Anadir librería Servlet.**

A continuación, se muestra una figura de los Servlets con Bluej.

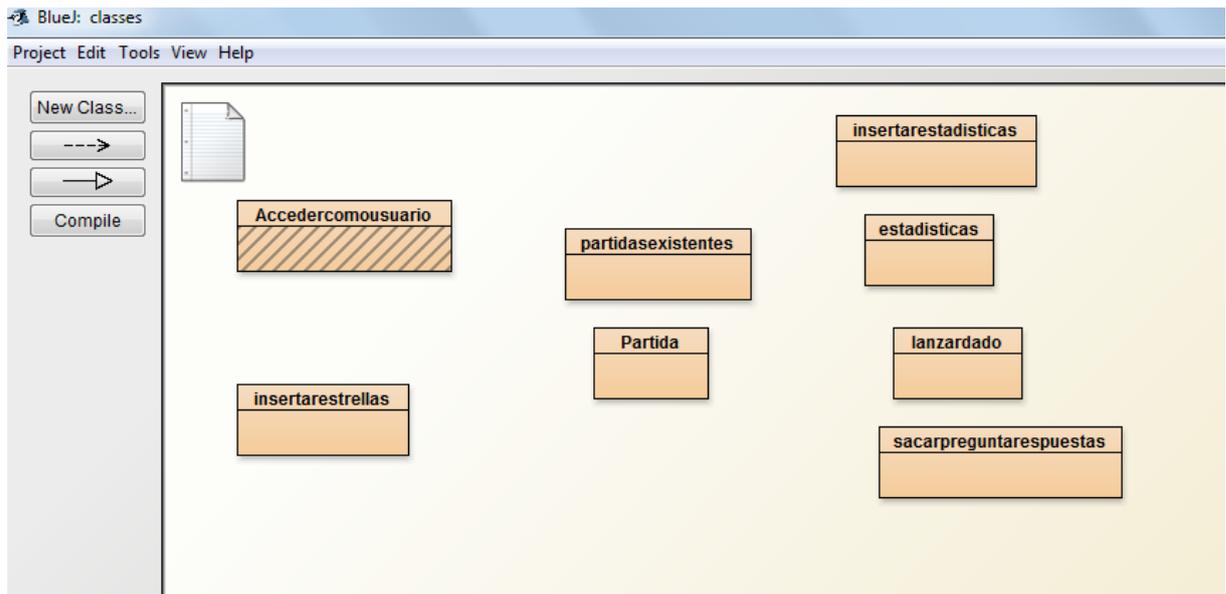


Figura 7: Servlets en Bluej.

## MySQL Workbench 6.3 CE

Es una herramienta visual de diseño de bases de datos que integra desarrollo de software, Administración de bases de datos, diseño de bases de datos, creación y mantenimiento para el sistema de base de datos MySQL. [6]

A continuación, se muestra una figura de la base de datos en MySQL Workbench.

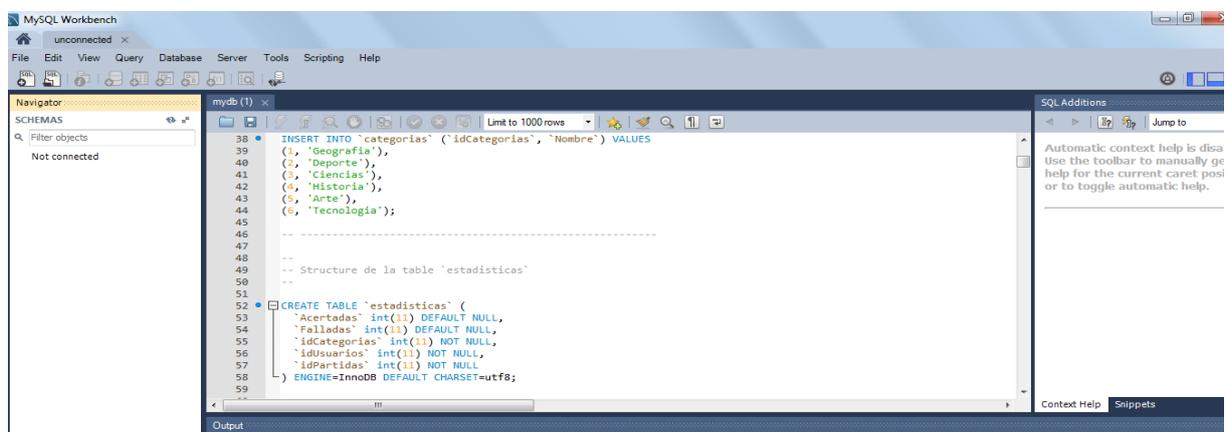


Figura 8: Base de datos en MySQL Workbench.

## Kompozer

Es un editor **WYSIWYG** (What You See Is What You Get = Lo que ves es lo que tienes) de páginas web. Resulta una herramienta de uso fácil, de libre distribución y de uso gratuito. [7]

Se usó en este proyecto para la creación del tablero de juego y las tablas de estadísticas, etc.

A continuación, se muestra una figura de creación de la tabla de estadísticas con la herramienta Kompozer.

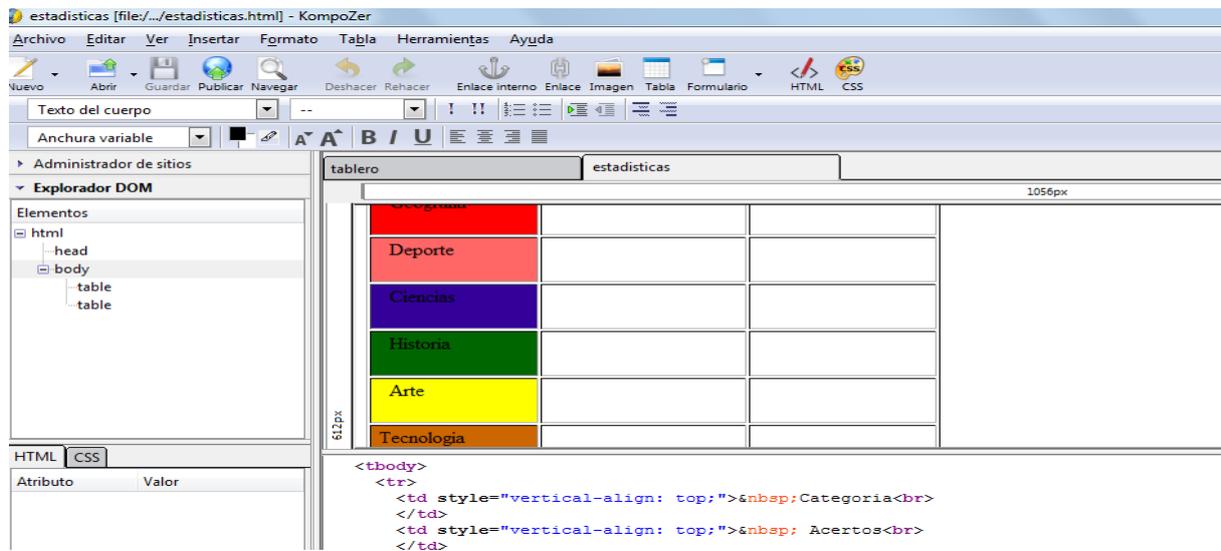


Figura 9: Creación de la tabla estadísticas con Kompozer.

## Notepad++

Es un editor de texto y de código fuente libre con soporte para varios lenguajes de programación. De soporte nativo a Microsoft Windows. Se parece al Bloc de notas en cuanto al hecho de que puede editar texto sin formato y de forma simple. [8]

Se usó en este proyecto para editar HTML5 Y CCS3.

## 3.2 Tecnologías

### Java Servlet

Es un objeto java de una clase que extiende de `javax.servlet.http.HttpServlet`, que nos permite crear aplicaciones web dinámicas es decir, le permite al usuario interactuar con la aplicación (hacer consultas, insertar y eliminar datos...)

Los Servlets admiten peticiones a través del protocolo HTTP. Los Servlets reciben peticiones desde un navegador web, las procesan y devuelven una respuesta al navegador, normalmente en HTML. Para realizar estas tareas podrán utilizar las clases incluidas en el lenguaje Java. [9]

Se han usado en este proyecto como intermediarios entre el cliente (navegador web) y los datos (BBDD).

Como he explicado anteriormente, los Servlets se almacenan en la ruta C:\xampp\tomcat\webapps\juego\_trivial\WEB-INF\classes y se deben declarar en el fichero web.xml que se almacena en la ruta C:\xampp\tomcat\webapps\juego\_trivial\WEB-INF.

A continuación, se muestra un ejemplo de uso de Servlet.

```
public class Partida extends HttpServlet
{
    public void doPost(HttpServletRequest req, HttpServletResponse res)
    {
        Connection con;
        Statement st,st1;
        ResultSet rs;
        String sql,sql1,partida,turno,posicion1,posicion2;

        //comprobar que el driver existe
        try
        {
            System.out.println("Intentando cargar el conector...");
            Class.forName("com.mysql.jdbc.Driver");
            System.out.println("Conectando a la base...");
            con = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb", "root","");
            System.out.println("Conexion con la base de datos establecida");
            partida = req.getParameter("p"); // parte
            posicion1 = req.getParameter("p1"); // position1
            posicion2 = req.getParameter("p2"); // position2
            turno = req.getParameter("t");

            PrintWriter out;
            out=res.getWriter();
            res.setContentType("text/html");

            st=con.createStatement();
            sql="SELECT * FROM guardar WHERE idPartidas="+partida+"";
            rs = st.executeQuery(sql);

            if( rs.next() ){
                st1=con.createStatement();
                sql1="UPDATE guardar SET idposicion1="+posicion1+", idposicion2="+posicion2+", TurnoJugador="+turno+" WHERE idPartidas="+partida+"";
                st1.executeUpdate(sql1);
            }else{
                st1=con.createStatement();
                sql1="INSERT INTO guardar (idPartidas, TurnoJugador, idposicion1, idposicion2) VALUES ('"+partida+"', '"+turno+"', '"+posicion1+"', '"+posicion2+"')";
                st1.executeUpdate(sql1);
            }
            out.println(sql1);
            con.close();
            st.close();
            st1.close();
        }
    }
}
```

**Figura 10: Ejemplo de uso de Servlet.**

Para acceder a la base de datos desde los Servlets se necesita un conector, en este proyecto se ha usado el driver JDBC.

El driver JDBC tiene las siguientes funciones:

- Establece una conexión con una BD.
- Envía sentencias SQL.
- Procesa los resultados.

Se instala el driver JDBC para MySQL [10] y se incluye el fichero .jar dentro de la variable de entorno CLASSPATH para poder compilar y ejecutar y la carpeta de lib de Tomcat.

Se llama al driver desde el Servlet para establecer una conexión con la base de datos (ver figura 11).

```

//comprobar que el driver existe
try
{
    System.out.println("Intentando cargar el conector...");
    Class.forName ("com.mysql.jdbc.Driver");
    System.out.println("Conectando a la base...");
    con = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb", "root","");
    System.out.println("conexion con la base de datos establecida2" );
}
}

```

**Figura 11: Conexión a la base de datos con JDBC.**

## HTML

Es un lenguaje de programación que se utiliza para el desarrollo de páginas de Internet.

Es muy simple y general que sirve para definir otros lenguajes que tienen que ver con el formato de los documentos. El texto en él se crea a partir de etiquetas, también llamadas tags, que permiten interconectar diversos conceptos y formatos. Para la escritura de este lenguaje, se crean etiquetas que aparecen especificadas a través de corchetes o paréntesis angulares: < y >. Entre sus componentes, los elementos dan forma a la estructura esencial del lenguaje, ya que tienen dos propiedades (el contenido en sí mismo y sus atributos). Por otra parte, cabe destacar que el HTML permite ciertos códigos que se conocen como scripts, los cuales brindan instrucciones específicas a los navegadores que se encargan de procesar el lenguaje. Entre los scripts que pueden agregarse, los más conocidos y utilizados son JavaScript y PHP. El marcado estructural es el que estipula la finalidad del texto, aunque no define cómo se verá el elemento. El marcado presentacional, por su parte, es el que se encarga de señalar cómo se verá el texto más allá de su función. Para conocer el código HTML que utiliza una página web, hay que seleccionar Ver código fuente en nuestro navegador (como Internet Explorer o Mozilla Firefox). Al elegir esta opción, se abrirá el editor de texto con el código HTML de la página que se está visualizando. [11]

En este proyecto se usó para definir el contenido de las páginas web.

A continuación, se muestra el código HTML del formulario crear una cuenta.

```

<div class="tab-content">
  <div id="signup">
    <h1>Crear una cuenta usuario 1</h1>

    <form action="/juego_trivial/Accedercomousuario" method="post">

      <div class="top-row">
        <div class="field-wrap">
          <label>
            Nombre<span class="req">*</span>
          </label>
          <input type="text" name="nombre" required autocomplete="off" />
        </div>

        <div class="field-wrap">
          <label>
            Apellidos<span class="req">*</span>
          </label>
          <input type="text" name="apellidos" required autocomplete="off"/>
        </div>

        <div class="field-wrap">
          <label>
            Correo electrónico<span class="req">*</span>
          </label>
          <input type="email" name="correo" required autocomplete="off"/>
        </div>

        <div class="field-wrap">
          <label>
            Establecer una contraseña<span class="req">*</span>
          </label>
        </div>
      </div>
    </form>
  </div>
</div>

```

Figura 12: Creación del formulario crear cuenta con HTML.

## CSS

Es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML.

Al crear una página web, se utiliza en primer lugar el lenguaje HTML para marcar los contenidos, es decir, para designar la función de cada elemento dentro de la página: párrafo, titular, texto destacado, tabla, lista de elementos, etc. Una vez creados los contenidos, se utiliza el lenguaje CSS para definir el diseño de la página. [12]

En este proyecto se usó para especificar el color, tamaño y tipo de letra del texto, separación horizontal y vertical entre elementos, posición de cada elemento, etc.

A continuación, se muestra una figura de todos los archivos CSS usados para dar estilo a la aplicación.

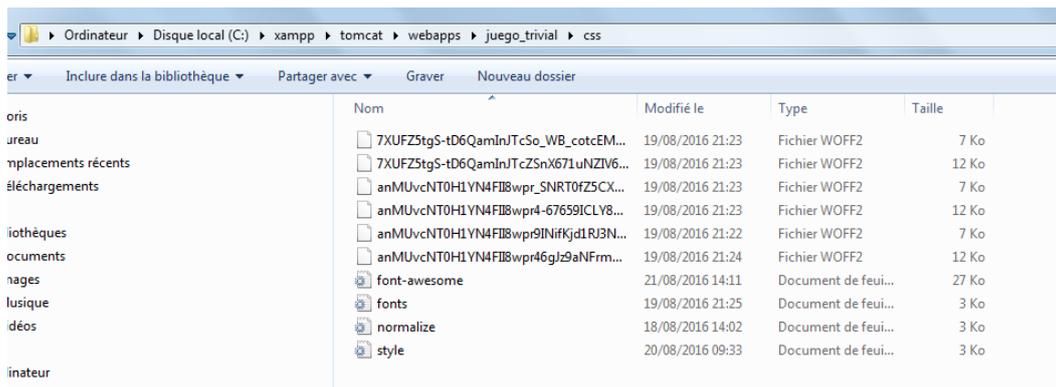


Figura 13: Ficheros CSS.

Se llama a estos archivos desde la página HTML en la parte de cabecera (ver figura 14).

```

<head>
  <meta charset="UTF-8">
  <title>JUEGO TRIVIAL</title>
  <link href='css/fonts.css' rel='stylesheet' type='text/css'>

  <link rel="stylesheet" href="css/normalize.css">
  <link rel="shortcut icon" href="favicon.png">
  <link rel="stylesheet" href="css/style.css">

```

Figura 14: Llamada a los ficheros CSS.

A continuación, se muestra una figura de código CSS para el formulario de registro.

```

}

.form {
  background: rgba(19, 35, 47, 0.9);
  padding: 40px;
  max-width: 600px;
  margin: 40px auto;
  border-radius: 4px;
  box-shadow: 0 4px 10px 4px rgba(19, 35, 47, 0.3);
}

.tab-group {
  list-style: none;
  padding: 0;
  margin: 0 0 40px 0;
}

.tab-group:after {
  content: "";
  display: table;
  clear: both;
}

.tab-group li a {
  display: block;
  text-decoration: none;
  padding: 15px;
  background: rgba(160, 179, 176, 0.25);
  color: #a0b3b0;
  font-size: 20px;
  float: left;
  width: 50%;
  text-align: center;
  cursor: pointer;
  -webkit-transition: .5s ease;
  transition: .5s ease;
}

.tab-group li a:hover {

```

Figura 15: Ejemplo Código CSS.

## JavaScript:

Es un lenguaje de programación interpretado. Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas. Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del Document Object Model (DOM). Tradicionalmente se venía utilizando en páginas web HTML para realizar operaciones y únicamente en la parte del cliente, sin acceso a funciones del servidor. No obstante, a partir de mediados de la década de los 2000, ha habido una proliferación de implementaciones de JavaScript para el lado servidor, surgiendo así "Node.js". Actualmente JavaScript es ampliamente utilizado para enviar y recibir información del servidor junto con ayuda de otras tecnologías como AJAX.[13]

En este proyecto se usó para programar el comportamiento de las páginas web.

A continuación, se muestra una figura de código JavaScript de la función de lanza el dado.

```
out.println("var randomdice;");
out.println("randomdice=Math.round(Math.random()*4)+1;");
//out.println("alert(randomdice)");
out.println("document.images[\\"mydice\\"].src=eval(\\"face\\"+randomdice+\\".src\\");");

out.println("if (currentJ==1) {");

    out.println("ident=currentP1+randomdice;");
    out.println("if (ident>64) {ident=ident-64}");
    out.println(" var color=document.getElementById(ident)");
    out.println("var muestracolor=document.defaultView.getComputedStyle(color).getPropertyValue(\\"backgrou");
    out.println("hlight(ident);");

    out.println("ident1=currentP1-randomdice;");
    out.println("if (ident1<=0) {ident1=65-randomdice}");
    out.println(" var color1=document.getElementById(ident1)");
    out.println("var muestracolor1=document.defaultView.getComputedStyle(color1).getPropertyValue(\\"backgr");
    out.println("hlight(ident1);");
out.println("}");
```

Figura 16: Función JavaScript Lanzar dado.

## jQuery:

Es una biblioteca de JavaScript, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web.

jQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio. [14]

Características del jQuery:

- Selección de elementos DOM.
- Interactividad y modificaciones del árbol DOM, incluyendo soporte para CSS 1-3 y un plugin básico de XPath.
- Eventos.

- Manipulación de la hoja de estilos CSS.
- Efectos y animaciones.
- Animaciones personalizadas.
- AJAX.
- Soporta extensiones.
- Utilidades varias como obtener información del navegador, operar con objetos y vectores, funciones para rutinas comunes, etc.

Se ha usado en este proyecto para las animaciones, pantallas de Popup, la interactividad y llamar a los servlets para enviar y/o recibir datos entre ellos.[15]

En la figura a continuación se muestra la función jQuery de la ventana Popup.

```

167 <script src="js/index.js"></script>
168 <script>
169 $(function() {
170
171     var appendthis = ("<div class='modal-overlay js-modal-close'></div>");
172
173     $('a[data-modal-id]').click(function(e) {
174         e.preventDefault();
175         $("body").append(appendthis);
176         $(".modal-overlay").fadeTo(500, 0.7);
177         //$(".js-modalbox").fadeIn(500);
178         var modalBox = $(this).attr('data-modal-id');
179         $('#'+modalBox).fadeIn($(this).data());
180     });
181
182
183     $(".js-modal-close, .modal-overlay").click(function() {
184         $(".modal-box, .modal-overlay").fadeOut(500, function() {
185             $(".modal-overlay").remove();
186         });
187     });
188
189
190     $(window).resize(function() {
191         $(".modal-box").css({
192             top: ($(window).height() - $(".modal-box").outerHeight()) / 2,
193             left: ($(window).width() - $(".modal-box").outerWidth()) / 2
194         });
195     });
196
197     $(window).resize();
198
199 });
200 </script>
201

```

Figura 17: Función jQuery Popup.

## **4 Análisis**

### **4.1 Requisitos**

Lista de requisitos inicial alcanzada tras varias reuniones entre el tutor y el autor de este TFG:

- La aplicación debe ser sobre preguntas y respuestas tipo test.
- Las preguntas deben ser de diferentes categorías.
- La aplicación debe ser multijugador.
- El jugador debe tener una cuenta, en caso de no tenerla debe registrarse.
- El jugador debe lanzar un dado y elegir entre dos categorías.
- Las partidas del juego se deben guardar en la base de datos para poder seguir jugando en otro momento.
- Se debe poder ver las estadísticas de cada jugador para todas las partidas.
- El Juego se termina cuando un jugador gana las 6 estrellas correspondientes a las 6 categorías de preguntas.

### **4.2 Base de Datos**

Se necesita una base de datos para almacenar la información entre usos de la aplicación. La base de datos tiene que almacenar toda la información de cada usuario, de las partidas, estadísticas, estrellas y también de preguntas y respuestas.

### **4.3 Aplicación**

A continuación, se presenta un esquema general de la aplicación:

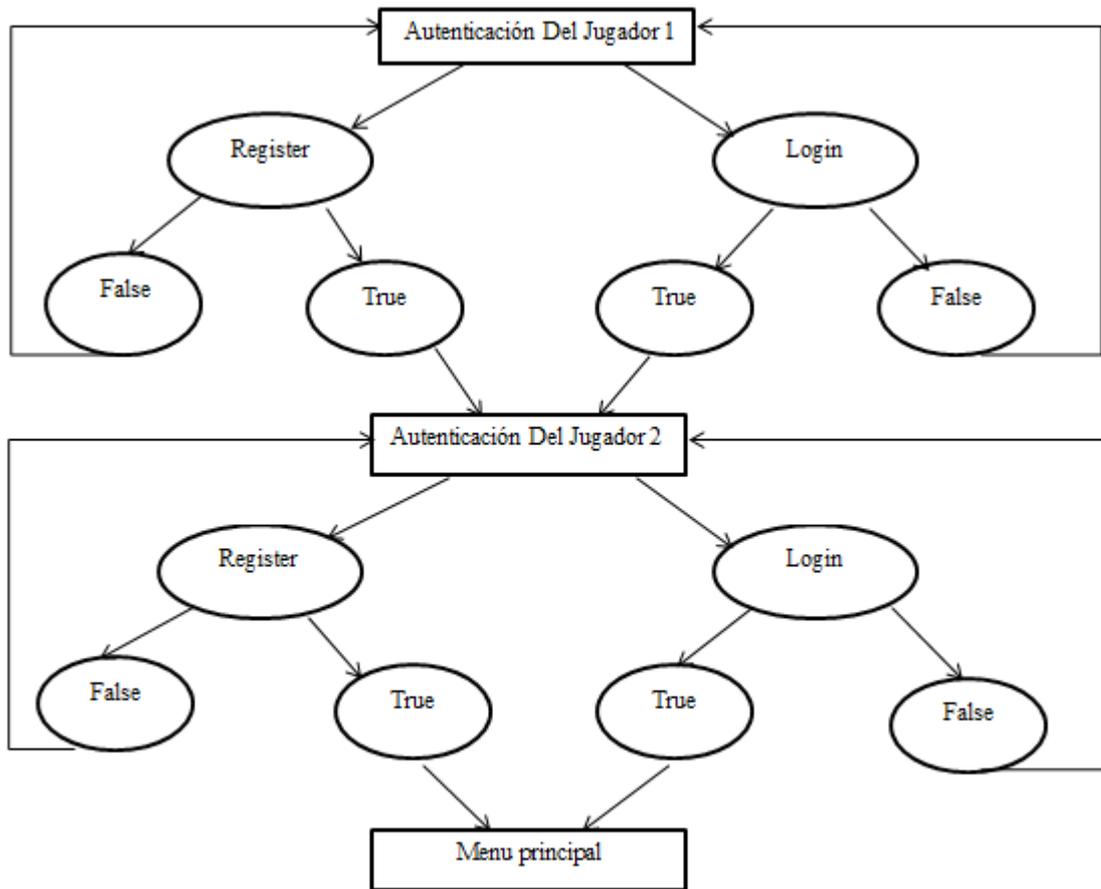


Figura 18: Esquema de autenticación de jugadores.

Lo primero de todo es la autenticación de los jugadores:

Jugador 1 Registrado: Debe introducir su correo y su contraseña, se verifica que no se ha dejado ningún campo vacío, luego se verifica si el usuario está registrado en la base de datos y la contraseña es correcta, en caso de información incorrecta vuelve al formulario de login, si todo es correcto se procede a la autenticación del Jugador 2.

Jugador 1 No registrado: En caso de que el jugador no esté registrado debe crear una cuenta, se verifica primero que se ha rellenado todos los campos, luego se verifica que el correo introducido no lo tiene otro usuario, si la información es incorrecta vuelve al formulario de registro, si todo es correcto se procede a la autenticación del Jugador 2.

Jugador 2 Registrado: Debe introducir su correo y su contraseña, se verifica que no se ha dejado ningún campo vacío luego se verifica si el usuario existe en la base de datos, y la contraseña es correcta, en caso de información incorrecta vuelve al formulario de login si todo es correcto sale el menú del juego.

Jugador 2 No registrado: En caso de que el jugador no este registrado debe crear una cuenta, se verifica primero que se ha rellenado todos los campos, luego se verifica que el correo introducido no lo tiene otro usuario, si la información es incorrecta vuelve al formulario de registro si todo es correcto sale el menú del juego.

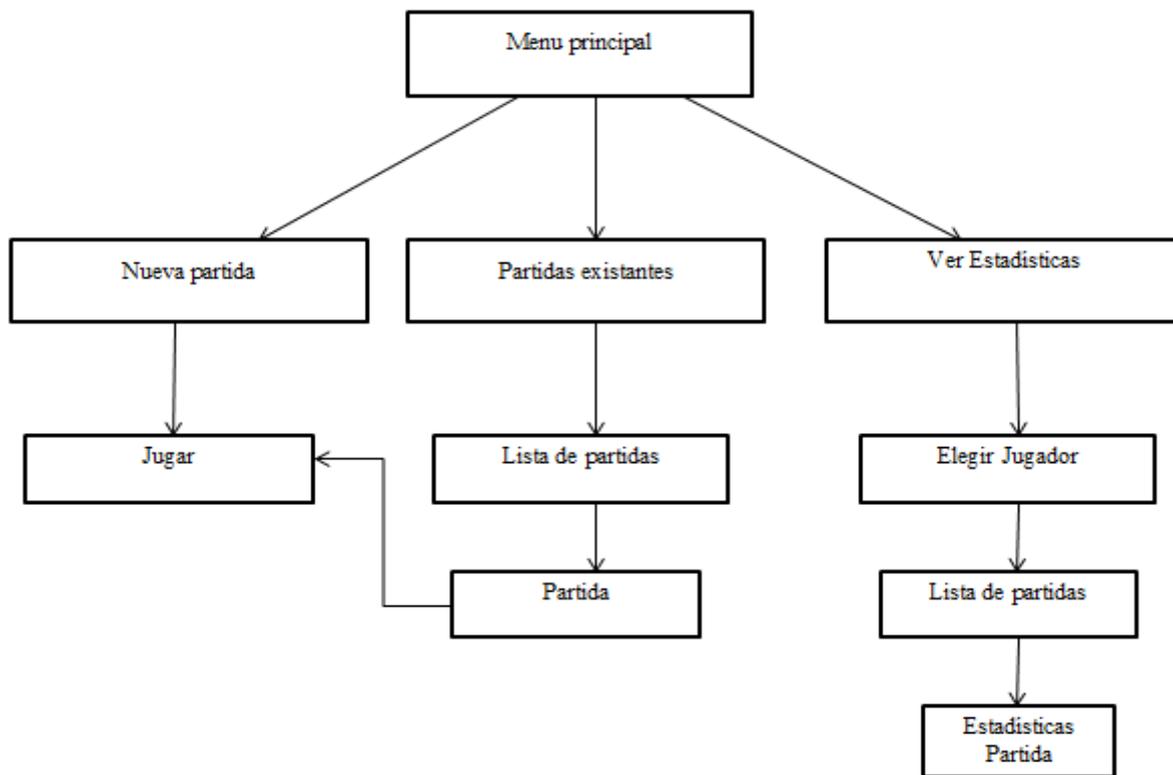


Figura 19: Menú del juego.

### El menú principal

Tiene tres opciones:

- Nueva partida
- Partidas existentes
- Ver estadísticas

**Nueva partida:** el juego contiene un tablero que, a su vez, contiene casillas de diferentes colores, cada color representa una categoría de preguntas de los cuales 6 son preguntas estrellas.

El juego dispone de 6 categorías:

- Rojo: Geografía.
- Rosa: Deporte.
- Azul: Ciencias.
- Verde: Historia.
- Amarillo: Arte.
- Naranja: Tecnología.
- Gris: Volver a tirar.

Los dos jugadores empiezan en la posición 1 del tablero. Al lanzar el dado, el jugador se mueve por las casillas del tablero según el número obtenido. Cada casilla contiene una categoría, que al seleccionarse sale una pregunta y 4 respuestas relativas a la categoría en cuestión, si la respuesta es correcta se pone en verde y el jugador que está jugando no pierde su turno. Si la respuesta es falsa se pone en rojo y el turno se pierde (ver figura 20).

**Partidas existentes:** contiene todas las partidas jugadas entre el jugador 1 y el jugador 2, se puede seguir jugando a las partidas no terminadas.

**Ver estadísticas:** Cada jugador tiene sus estadísticas de cada partida. Podemos ver las estadísticas de todas las partidas de los jugadores.

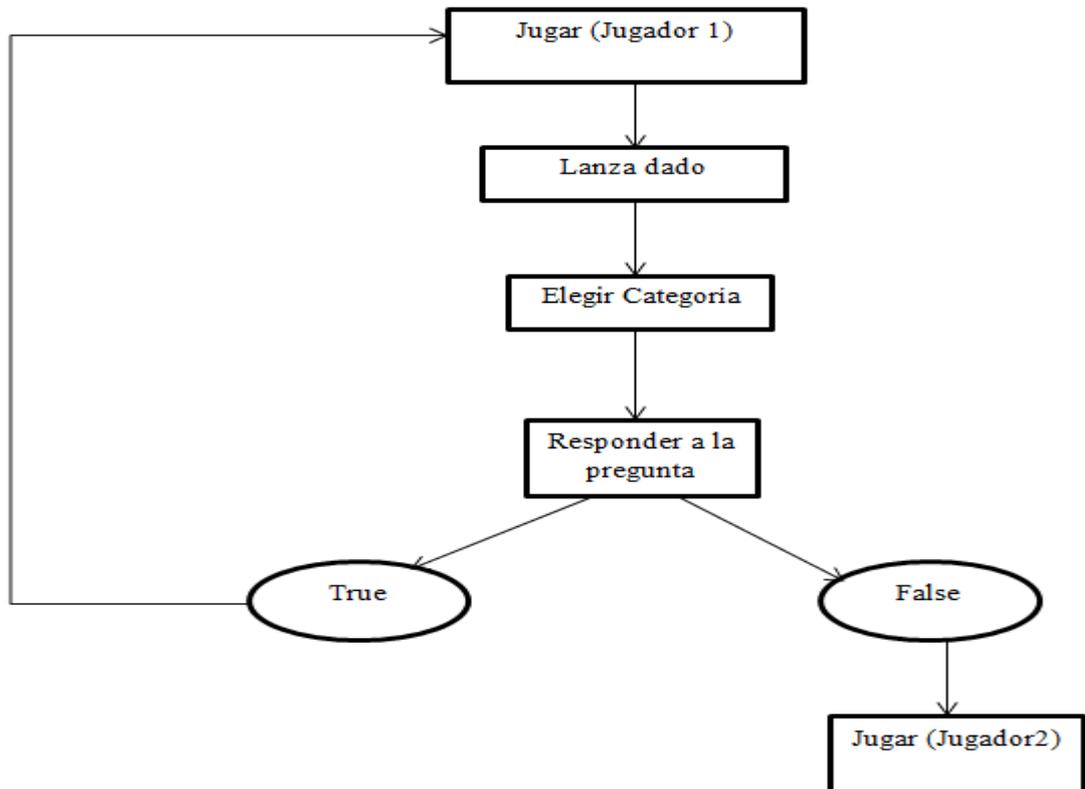


Figura 20: Funcionamiento del juego.

#### 4.4 Interfaces

Las interfaces necesarias para diseñar la aplicación son:

- Pantalla para registro o inicio de sesión para los jugadores.
- Menú principal que contiene tres opciones “nueva partida”, “partidas existentes”, “ver estadísticas”.
- Pantalla de partidas existentes.
- Pantalla de estadísticas.
- Pantalla del juego.

## 5 Diseño de la aplicación

### 5.1 Diseño base de datos

En la figura a continuación se muestra una imagen de todas las tablas de la base de datos que se detallarán más adelante.

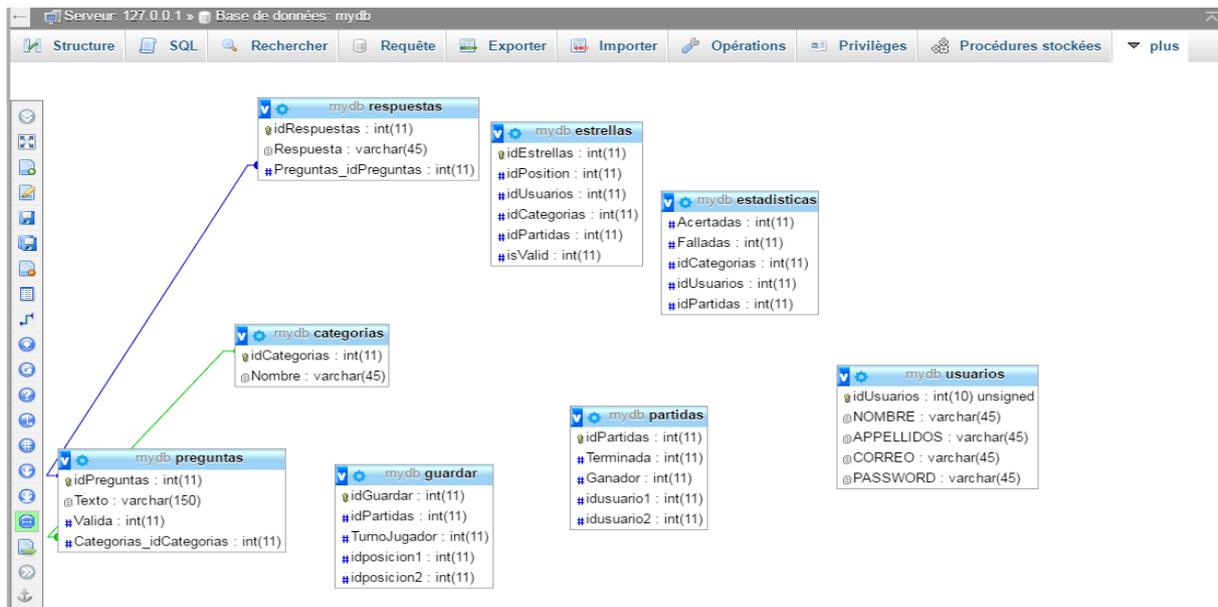


Figura 21: Base de datos.

*Usuarios*: Almacena la información de los usuarios.

Structure de table

#	Nom	Type	Interclassement	Attributs	Null	Défaut	Extra	Action
1	idUsuarios	int(10)		UNSIGNED	Non	Aucune	AUTO_INCREMENT	Modifier Supprimer Primaire Unique Index Spatial plus
2	NOMBRE	varchar(45)			Non	Aucune		Modifier Supprimer Primaire Unique Index Spatial plus
3	APPELLIDOS	varchar(45)			Non	Aucune		Modifier Supprimer Primaire Unique Index Spatial plus
4	CORREO	varchar(45)			Non	Aucune		Modifier Supprimer Primaire Unique Index Spatial plus
5	PASSWORD	varchar(45)			Non	Aucune		Modifier Supprimer Primaire Unique Index Spatial plus

Version imprimable Suggérer des optimisations de structure Suivre la table Déplacer des colonnes Améliorer la structure de la table

Ajouter 1 colonne(s) après PASSWORD Exécuter

**Information**

Espace utilisé		Statistiques	
Données	16 Kio	Format	Compact
Index	0	Interclassement	utf8_general_ci
Total	16 Kio	Prochain index automatique	12
		Création	Dim 28 Août 2016 à 18:49

Figura 22: Tabla de usuarios.

Contiene 5 campos:

- idUsuarios: contiene el ID de cada usuario.
- Nombre: contiene el nombre del usuario.
- Apellidos: contiene los apellidos del usuario.
- Correo: contiene el correo del usuario.
- Password: contiene la contraseña del usuario.

*Preguntas:* Almacena la información de las preguntas.

#	Nom	Type	Interclassement	Attributs	Null	Défaut	Extra	Action
1	idPreguntas	int(11)			Non	Aucune	AUTO_INCREMENT	Modifier Supprimer Primaire Unique plus
2	Texto	varchar(150)			Non	Aucune		Modifier Supprimer Primaire Unique plus
3	Valida	int(11)			Oui	NULL		Modifier Supprimer Primaire Unique plus
4	Categorias_idCategorias	int(11)			Non	Aucune		Modifier Supprimer Primaire Unique plus

Espace utilisé		Statistiques	
Données	16 Kio	Format	Compact
Index	16 Kio	Interclassement	utf8_general_ci
Total	32 Kio	Prochain index automatique	121
		Création	Dim 28 Août 2016 à 18:49

Figura 23: Tabla de preguntas.

Contiene 4 campos:

idPreguntas: contiene el ID de cada pregunta.

Texto: contiene la pregunta.

Valida: contiene el Id de la respuesta válida.

Categorias\_idCategorias: contiene el id de la categoría de la pregunta.

Respuestas: indica la información de las respuestas.

#	Nom	Type	Interclassement	Attributs	Null	Défaut	Extra	Action
1	idRespuestas	int(11)			Non	Aucune	AUTO_INCREMENT	Modifier Supprimer Primaire Unique Index plus
2	Respuesta	varchar(45)			Oui	NULL		Modifier Supprimer Primaire Unique Index plus
3	Preguntas_idPreguntas	int(11)			Non	Aucune		Modifier Supprimer Primaire Unique Index plus

Espace utilisé		Statistiques	
Données	48 Kio	Format	Compact
Index	14 Kio	Interclassement	utf8_general_ci
Total	64 Kio	Prochain index automatique	481
		Création	Dim 28 Août 2016 à 18:49

Figura 24: Tabla de respuestas.

Contiene 3 campos:

idRespuestas: contiene el ID de cada respuesta.

Respuesta: contiene la respuesta.

Preguntas\_idPreguntas: contiene el id de la pregunta de la respuesta.

A continuación, se muestra una figura de la introducción de preguntas y respuestas en la base de datos.

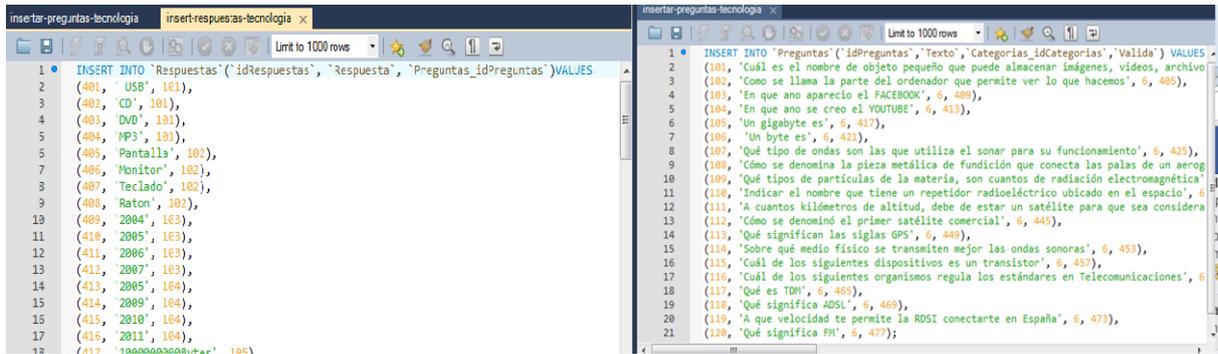


Figura 25: Introducción preguntas y respuestas.

Categorías: Indica la información de las categorías.

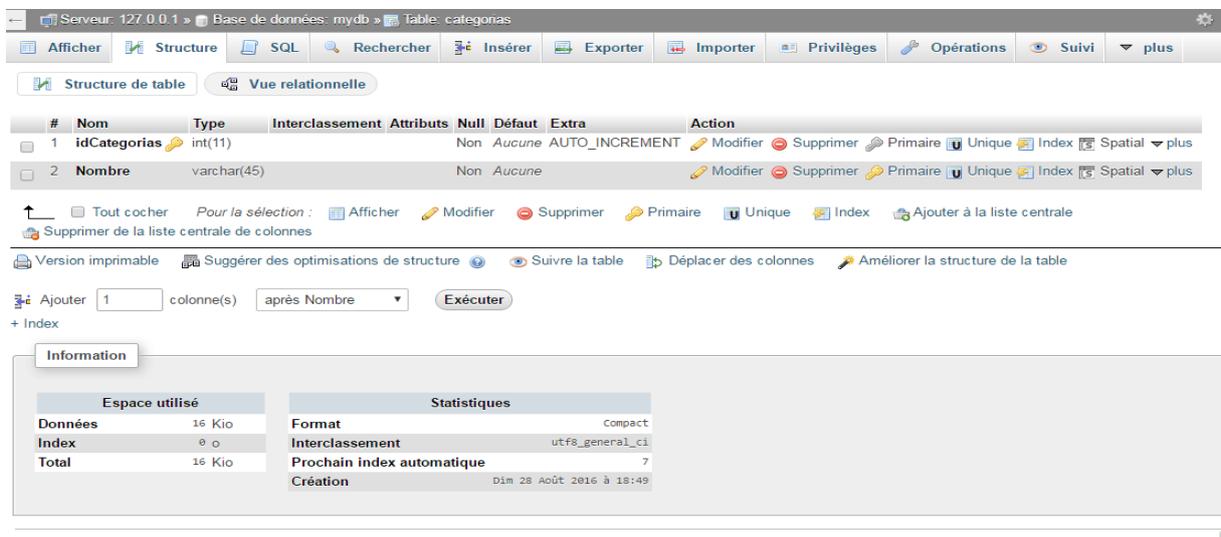


Figura 26: Tabla de categorías.

Contiene dos campos:

idCategorias: contiene el ID de cada categoría.

Nombre: contiene la categoría.

A continuación, se muestra una figura de introducción de categorías.

```

1 INSERT INTO `Categorias` (`idCategorias`,`Nombre`)VALUES
2 (1,'Geografia'),
3 (2,'Deporte'),
4 (3,'Ciencias'),
5 (4,'Historia'),
6 (5,'Arte'),
7 (6,'Tecnologia');
8
9
10
11

```

Figura 27: Introducción categorías.

*Partidas*: indica la información de las partidas.

#	Nom	Type	Interclassement	Attributs	Null	Défaut	Extra	Action
1	idPartidas	int(11)			Non	Aucune	AUTO_INCREMENT	Modifier Supprimer Primaire Unique Index Spatial plus
2	Terminada	int(11)			Oui	NULL		Modifier Supprimer Primaire Unique Index Spatial plus
3	Ganador	int(11)			Oui	NULL		Modifier Supprimer Primaire Unique Index Spatial plus
4	idusuario1	int(11)			Non	Aucune		Modifier Supprimer Primaire Unique Index Spatial plus
5	idusuario2	int(11)			Non	Aucune		Modifier Supprimer Primaire Unique Index Spatial plus

Espace utilisé		Statistiques	
Données	16 Kio	Format	Compact
Index	0 o	Interclassement	utf8_general_ci
Total	16 Kio	Prochain index automatique	6
		Création	Dim 28 Août 2016 à 18:49

Figura 28: Tabla de partidas.

Contiene 5 campos:

idPartidas: contiene el ID de cada partida.

Terminada: es 0 si la partida no está terminada y 1 si se ha terminado.

Ganador: contiene el ID del usuario que ha ganado la partida.

Idusuario1: contiene el ID del Jugador 1.

Idusuario2: contiene el Id del Jugador 2.

*Guardar*: Guarda las partidas jugadas.

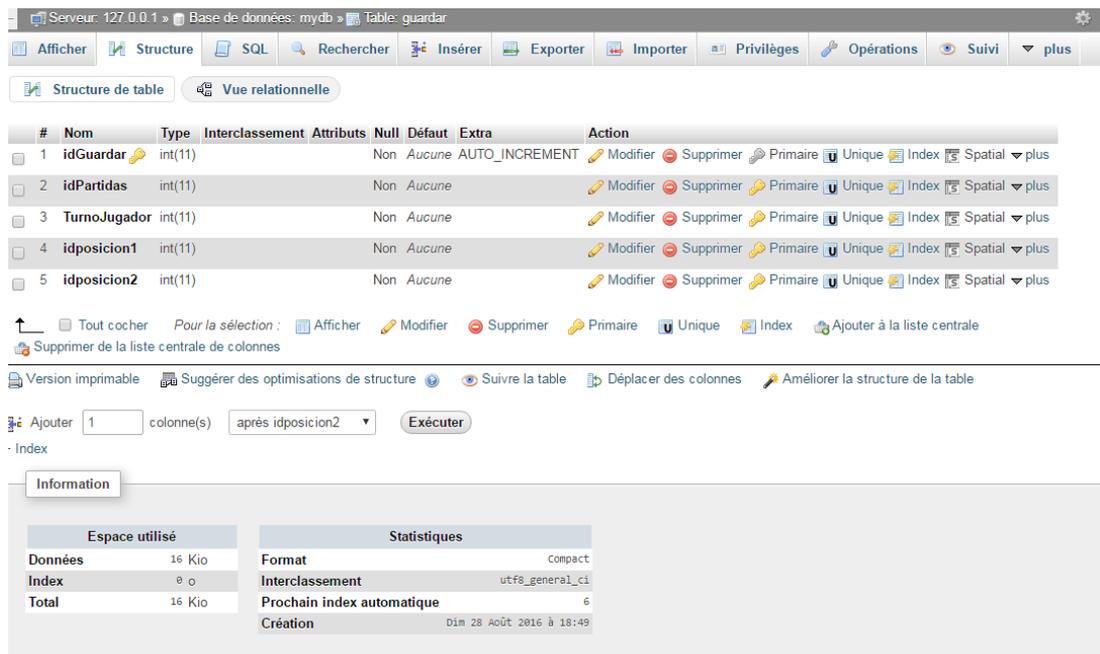


Figura 29: Tabla guardar.

Contiene 5 campos:

idGuardar: Contiene el ID.

idPartidas: Contiene el ID de la partida guardada.

TurnoJugador: Contiene el turno del Jugador.

Idposicion1: Contiene la posición del jugador 1 en el tablero del juego.

Idposicion2: Contiene la posición del jugador2 en el tablero del juego.

*Estrellas*: Indica la información de las estrellas.

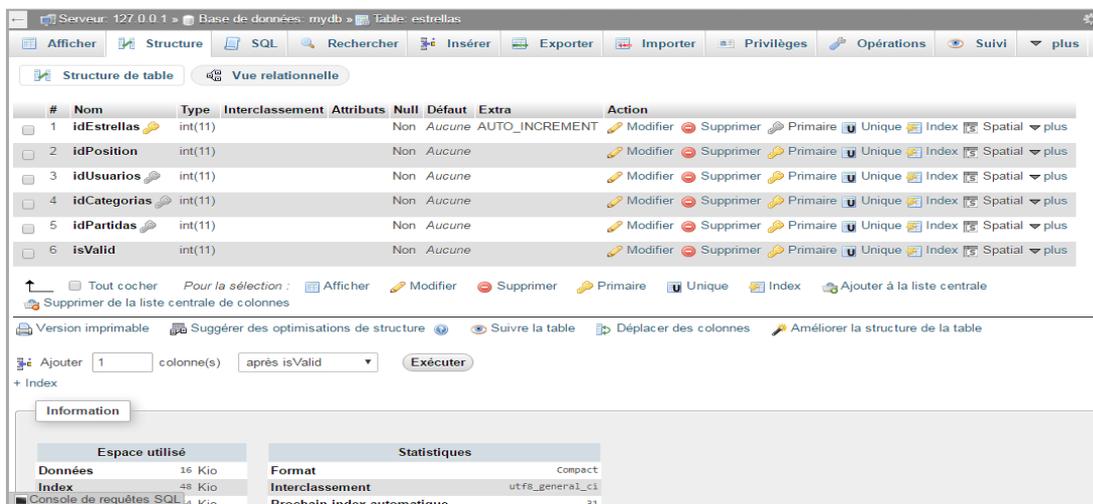


Figura 30: Tabla de estrellas.

Contiene 6 campos:

idEstrellas: contiene el ID de la estrella.

idPosicion: contiene la posición de la estrella en el tablero del juego.  
 idUsuarios: contiene el ID del usuario que ha respondido a la estrella.  
 idCategorias: contiene el ID de la categoría de la estrella.  
 idPartida: contiene el ID de la partida correspondiente.  
 isValid: vale 0 si la respuesta es falsa y 1 si la respuesta es correcta.

*Estadísticas*: Indica la información de las estadísticas.

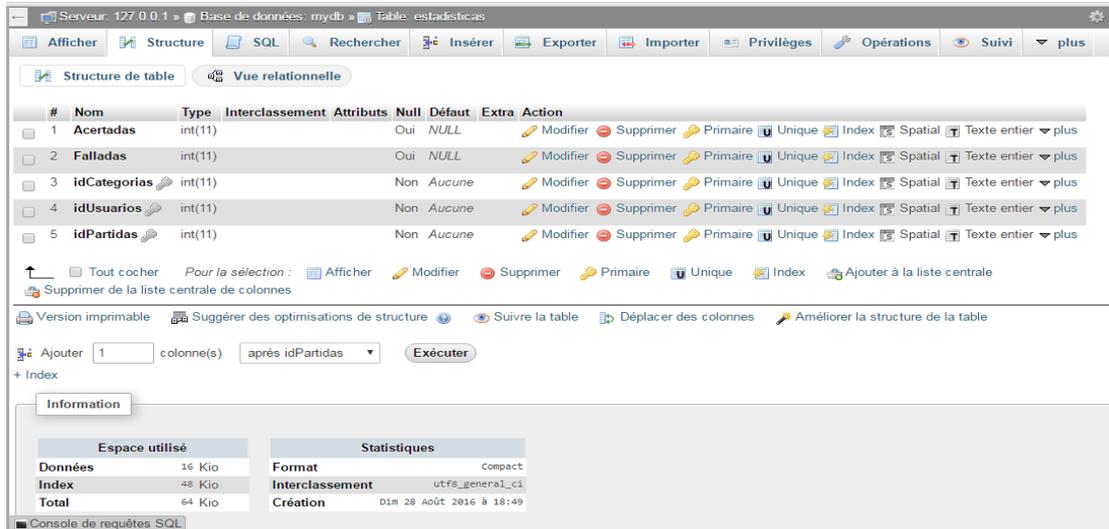


Figura 31: Tabla de estadísticas.

Contiene 5 campos:

idPartidas: contiene el ID de la partida en cuestión.  
 idUsuarios: contiene el ID del usuario en cuestión.  
 idCategorias: contiene el ID de la categoría en cuestión.  
 Acertadas: Contiene el número de preguntas acertadas.  
 Falladas: Contiene el número de preguntas falladas.

## 5.2 Diseño de la aplicación

*Index.html*: Es la interfaz principal de la aplicación, contiene un formulario de registro y otro de login en caso de que el jugador dispone de una cuenta, todo ello implementado en HTML y CSS3 para el estilo. Una vez introducidos los datos en los formularios se llama al Servlet accedercomousuario para validarlos.

La figura a continuación muestra el código HTML del formulario acceder como usuario y la interfaz correspondiente.

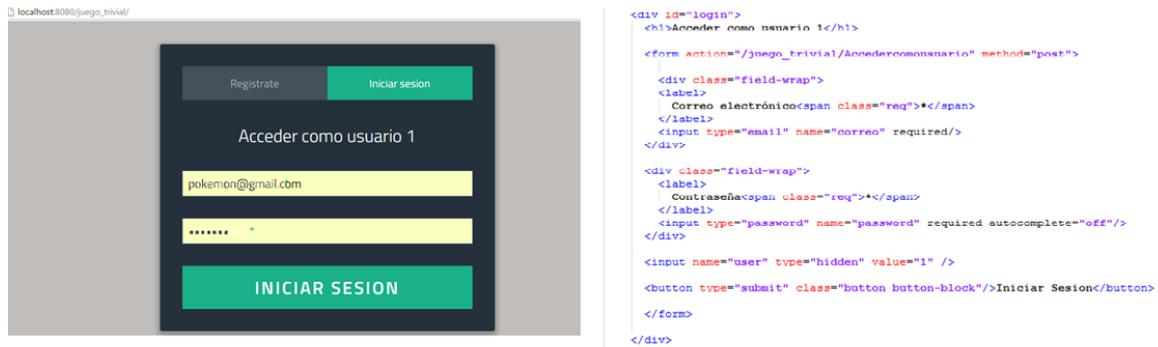


Figura 32: Código HTML e interfaz de la Index.html.

*Accedercomousuario:* Es el Servlet de autenticación de los dos jugadores. Si el usuario 1 ya tiene cuenta verifica primero que no ha dejado ningún campo vacío después accede a la base de datos a la tabla de usuarios y verifica que el correo introducido y la contraseña son correctos, en caso de que el usuario no este registrado el Servlet recibe los datos del formulario, verifica que no se ha dejado ningún campo vacío y luego verifica que el correo introducido no lo tiene otro usuario, si todo es correcto inserta los datos en la base de datos en la tabla de usuarios. Se hace lo mismo para el usuario 2 (ver figuras 33 y 34).

Después de la autenticación de los dos jugadores se accede al menú principal que se implementa desde este Servlet con HTML y CSS3.

El menú tiene tres opciones: Nueva partida, Partida existente, ver estadísticas (ver figura 35).



Figura 33: Interfaces de autenticación del jugador 1.

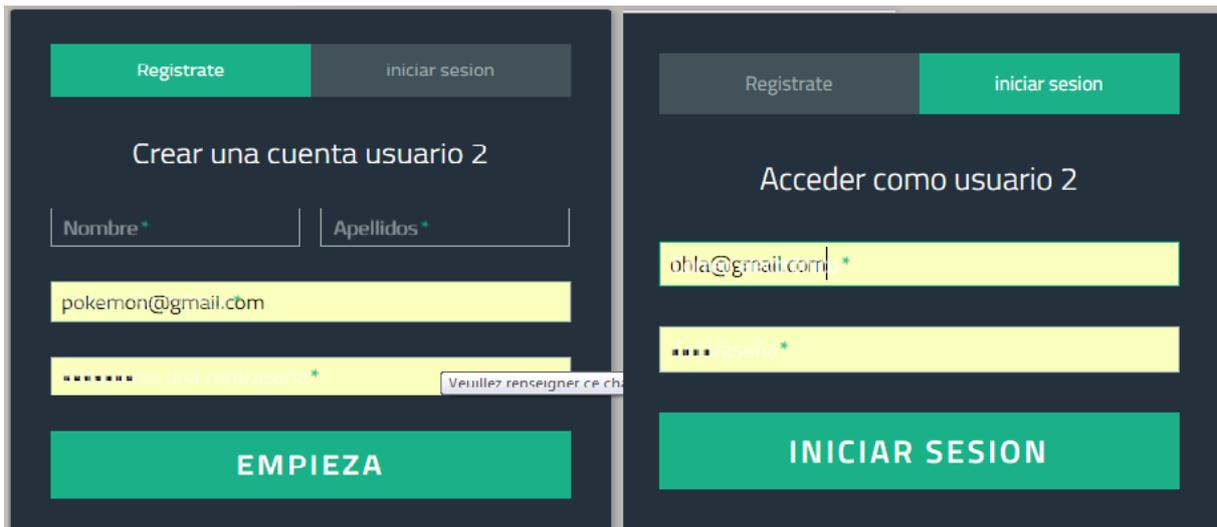


Figura 34: Interfaces de autenticación del jugador 2.



Figura 35: Interfaz del menú principal.

*Partidasexistentes:* Es un Servlet que recibe el ID de los dos jugadores desde el Servlet accedercomo usuario, saca los nombres desde la tabla usuarios de la base de datos, luego accede a la tabla partidas y verifica que existen partidas entre los dos jugadores, sino hay partidas saca un mensaje por pantalla “no se ha registrado ninguna partida”, si existen partidas las saca por pantalla (ver figura 36). Si el valor del campo terminada en la tabla de partidas es 0 la partida aún no se ha terminado se puede seguir jugando, si el valor de terminada es 1 eso significa que la partida ya está terminada. En caso de que la partida no está terminada al hacer clic en ella sale la pantalla del juego ejecutando el Servlet lanzado que explicare más adelante.



Figura 36: Lista de partidas existentes.

*Estadísticas:* Es un Servlet que se lanza al hacer clic en el botón ver estadísticas del menú principal, recibe los ID de los dos jugadores, accede a la base de datos, saca los nombres de los usuarios de la tabla usuarios y muestra por pantalla dos botones con los nombres de los jugadores, al hacer clic en uno de ellos se vuelve a ejecutar el Servlet estadísticas, accediendo a la base de datos y sacando la lista de las partidas de los dos jugadores, al hacer clic en una de las partidas se ejecuta otra vez el Servlet Estadísticas que accede a la base de datos en la tabla estadísticas, nos saca las estadísticas de la partida y el jugador en cuestión por pantalla( ver figura 37).

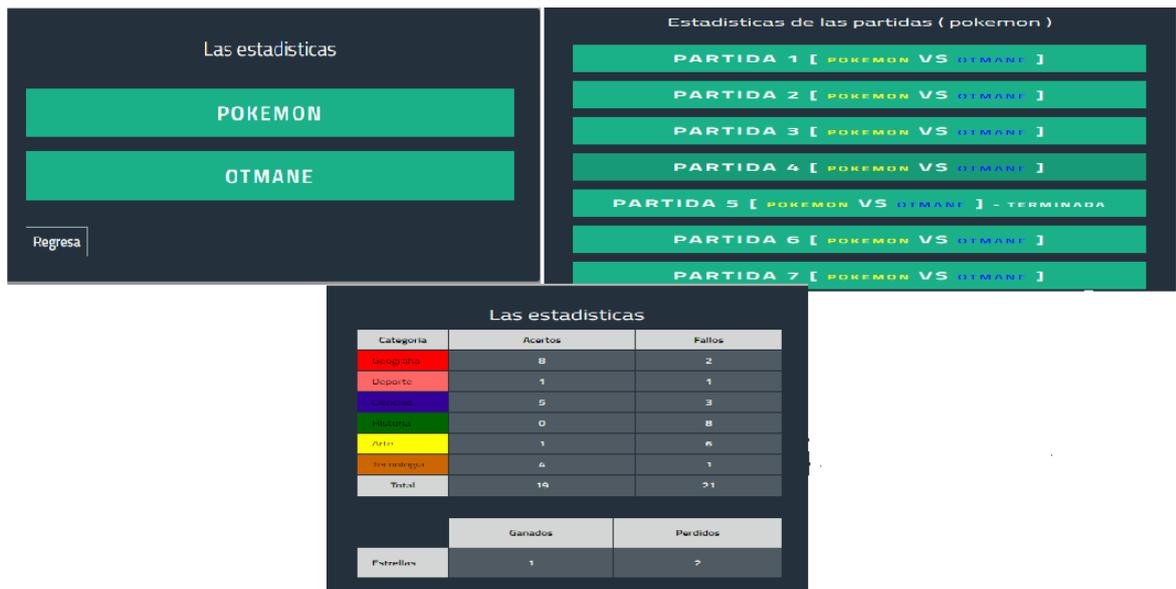


Figura 37: Interfaces para ver las estadísticas.

*Lanzardado*: Es el Servlet más complejo de esta aplicación, a continuación, se muestra un esquema de su funcionamiento.

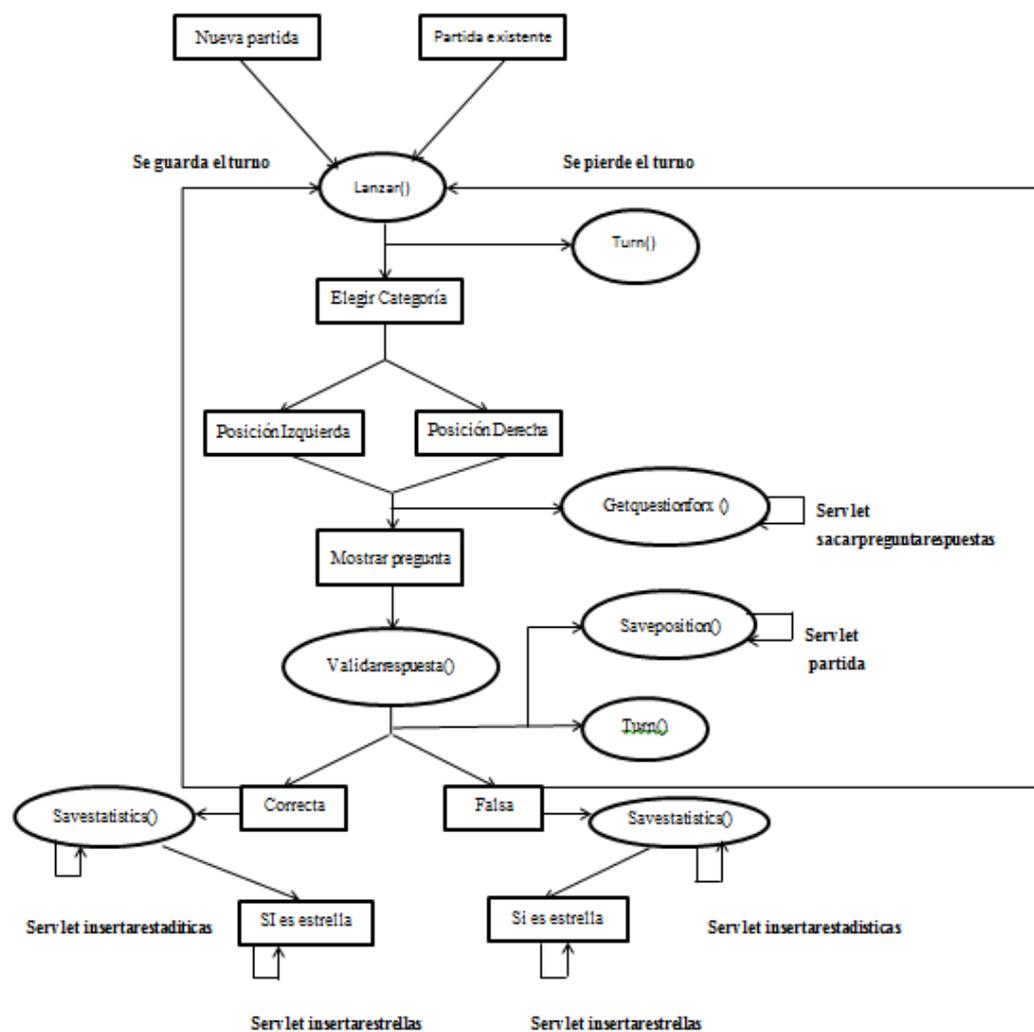


Figura 38: Funcionamiento del Servlet lanzardado.

Si la partida es nueva el Servlet lanzardado la inicia en la base de datos en la tabla de partidas, si la partida ya existe saca la información de ella en la base datos en la tabla guardar donde se guardan las posiciones de los jugadores, el jugador que tiene el turno etc.

Lanzar (): Es una función JavaScript que lanza el dado, saca un número aleatorio n y mueve la ficha de posición del jugador n posiciones a la derecha y n posiciones a la izquierda (ver figura 39), cada posición tiene un color que representa una categoría (ver figura 40).

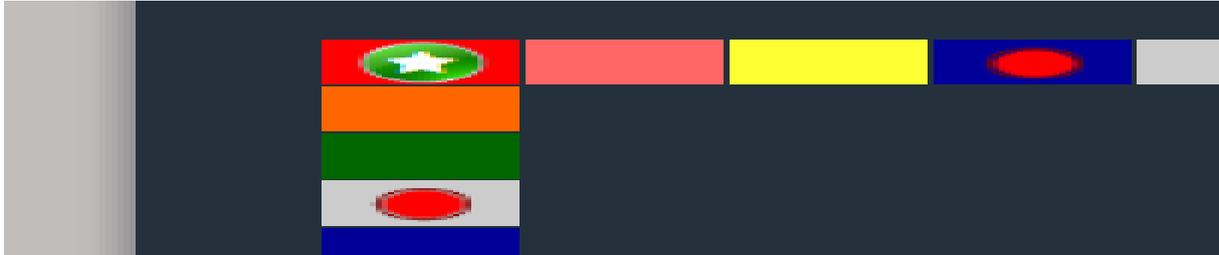


Figura 39: Posiciones del jugador.



Figura 40: Elección de categoría.

Return (): Es una función jQuery que refresca los iconos de los jugadores, cambia la posición del icono en función del turno del jugador (ver figura 41).



Figura 41: Icono del turno del jugador.

GetQuestionForx (): Es una función jQuery que envía una petición al Servlet Sacarpreguntarespuestas con el ID de la categoría y recibe una tabla que contiene una pregunta aleatoria y 4 respuestas de la categoría en cuestión que mostrara en una ventana Popup (ver figura 42).

*Sacarpreguntarespuestas*: Es un Servlet que recibe el ID de la categoría de la pregunta, accede a la base de datos, saca una pregunta y 4 respuesta aleatoriamente y la respuesta correcta correspondiente a la categoría recibida y los devuelve a la función GetQuestionForx ().

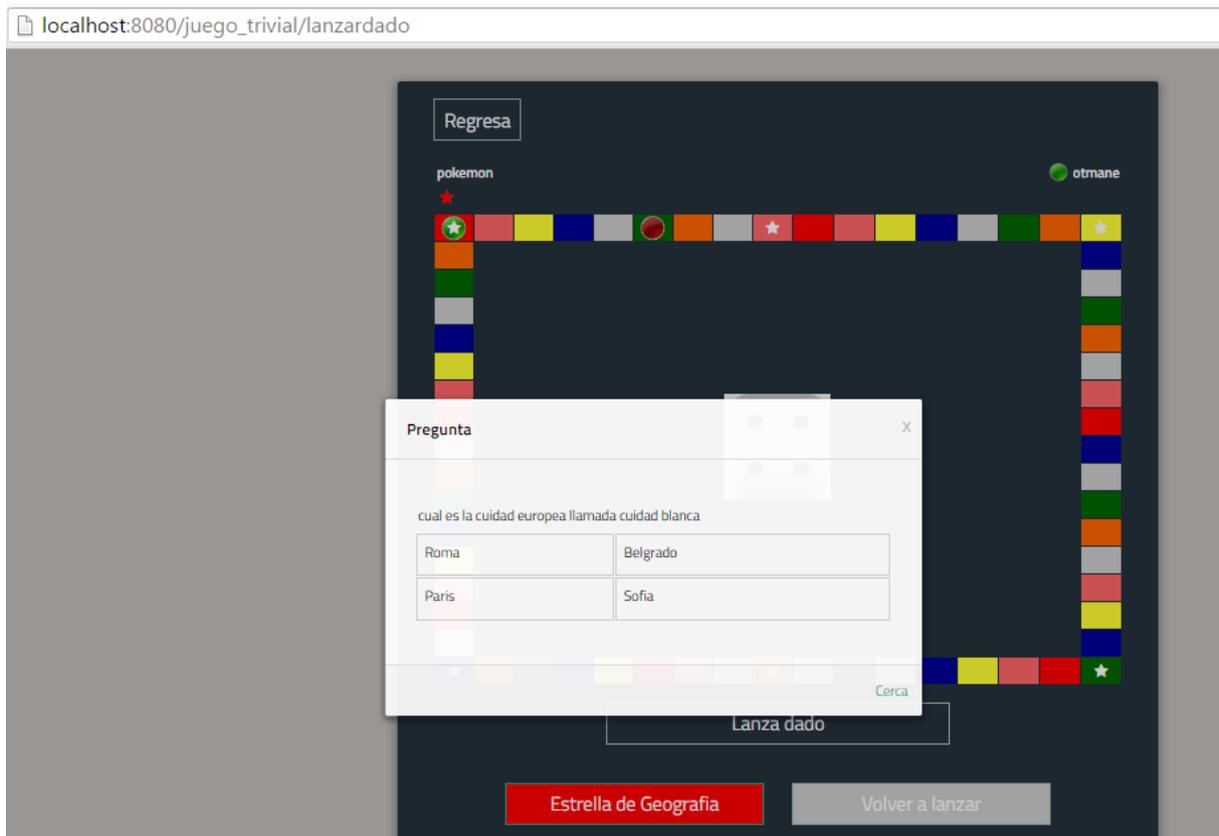


Figura 42: Pregunta en Popup.

Validarrespuesta (): Es una función JavaScript que verifica si la respuesta es correcta o falsa. Recibe la respuesta correcta y la respuesta del jugador, los compare, si son iguales se cambia el color de la respuesta a verde, sino se pone en rojo (ver figura 43).

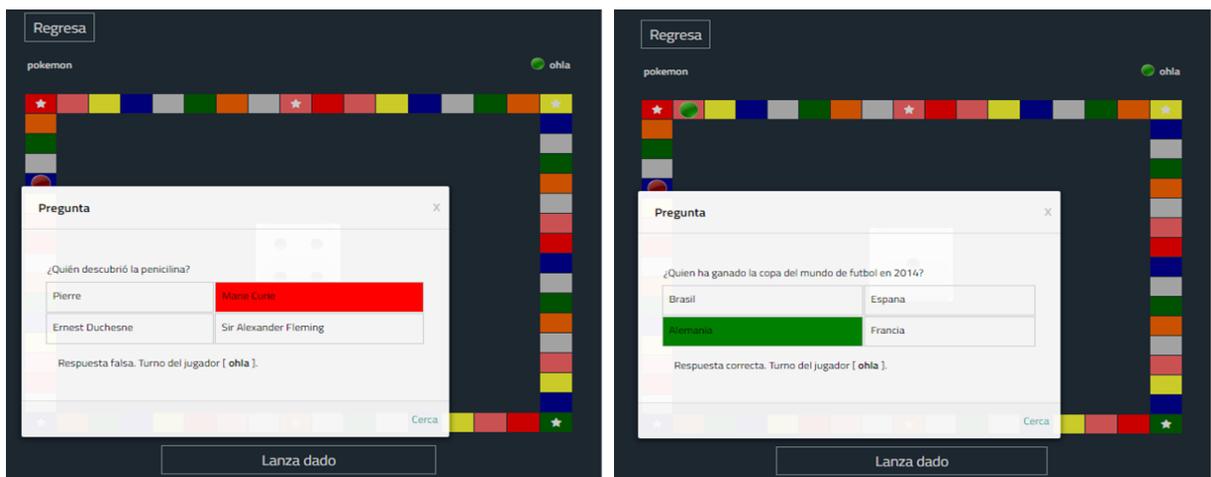


Figura 43: Verificación de respuestas.

Saveposition (): Es una función jQuery que envía una petición post al Servlet partida para actualizar las posiciones de los jugadores.

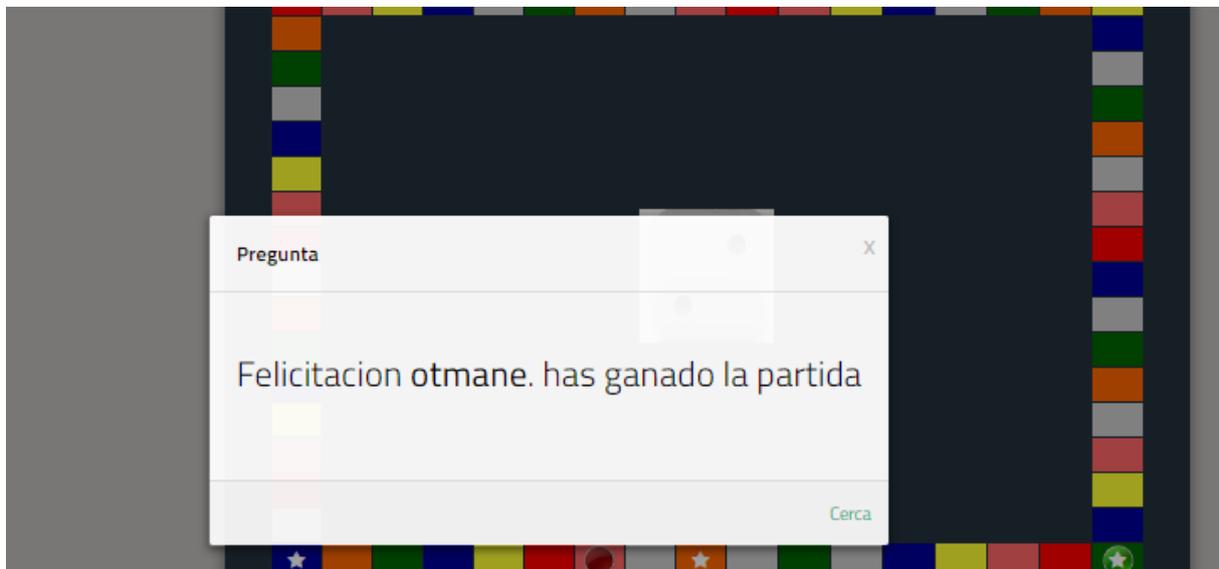
*Partida:* Es un Servlet que recibe el ID de la partida, las posiciones de los dos jugadores y el turno. Accede a la base de datos en la tabla Guardar, verifica si existe la partida. Si existe actualiza los datos, sino esto significa que es una nueva partida y entonces la inserta.

Savestaticis (): Es una función jQuery que envía una petición al Servlet insertarestadisticas para insertar las estadísticas de una partida.

*Insertarestadisticas:* Es un Servlet que recibe los parámetros partida, jugador, estado (si la respuesta es correcta es 1 sino es 0) y la categoría, verifica en la base de datos si existe estadísticas para esta partida, si existe y la respuesta es correcta se suma un punto a acertadas, sino es correcta se suma un punto a falladas. Si la partida no existe en estadísticas la inserta.

Además, si la pregunta es una estrella, la función Savestaticis envía una petición al Servlet insertarestrellas para insertarla, recibe un 1 si el jugador ha obtenido las 6 estrellas y un 0 en el caso contrario. Si la respuesta es un 1 se ejecuta una función jQuery que saca un mensaje de fin del juego en una ventana Popup (ver figura 44).

*Insertarestrellas:* Es un Servlet que recibe los parámetros posición de la estrella en el tablero, Estado de la respuesta, el jugador, la categoría de estrella y la partida. Si la respuesta es correcta, accede a la base de datos en la tabla de estrellas y verifica si la estrella existe, si existe actualiza el campo idvalid sino existe la inserta con idvalid=1 además hace una cuenta de las estrellas ganadas si es 6 se actualiza el campo terminada en la tabla de partidas y se borra la partida de la tabla guardar. Si la respuesta es falsa accede a la base de datos en la tabla de estrellas y verifica si la estrella existe, si existe actualiza el campo idvalid sino existe la inserta con idvalid=0.



**Figura 44:** Fin del Juego.

## 6 Conclusiones

El objetivo de este proyecto ha sido crear una aplicación web de un juego trivial, permitiéndose a dos jugadores competir para ganar 6 preguntas estrellas de diferentes categorías. Para ello se han utilizado varias herramientas y lenguajes de programación muy potentes.

Para realizar este proyecto han sido necesarios muchos conocimientos adquiridos a lo largo de la carrera en especial la asignatura sistemas telemáticos de la gestión de información y las asignaturas de programación. También se han adquirido bastantes nuevos conocimientos, así que el desarrollo de este proyecto ha sido de gran interés. También ha surgido un interés personal para mejorarlo y porque no hacer más aplicaciones web.

## 7 Propuesta de trabajo futuro

La creación de este proyecto abre las puertas a un desarrollo de una herramienta mucho más completa y compleja.

Posibles ampliaciones del proyecto serían:

- Posibilidad de jugar online contra otro jugador es decir sin tener que estar en el mismo local.
- Posibilidad de jugar en diferentes idiomas.
- Añadir nuevos modos de juego.
- Posibilidad de agregar nuevas preguntas en la base de datos.
- Poner varios niveles de dificultad de las preguntas.
- Incluir una sección de rankings en la aplicación para aumentar la competitividad entre los Jugadores.
- Anadir una opción de chat entre los jugadores.
- Anadir una foto de perfil o icono para los jugadores.

## 8 Bibliografía

- [1] XAMPP. <http://www.caezar.net/xampp-servidor-apache-php-tomcat-jsp-perl-mysql.html> [Online]
- [2] Wikipedia, “Tomcat”. <https://es.wikipedia.org/wiki/Tomcat> [Online]
- [3] XAMPP. <http://www.apachefriends.org/> [Online]
- [4] Wikipedia, “Bluej”. <https://es.wikipedia.org/wiki/BlueJ> [Online]
- [5] Bluej. <http://www.bluej.org/> [Online]
- [6] MySQL. <https://www.mysql.com/>. [Online]
- [7] Kompozer. <http://www.ite.educacion.es/formacion/materiales/107/cd/html/html0201.html> [Online]
- [8] Wikipedia, “Notepad++”. <https://es.wikipedia.org/wiki/Notepad%2B%2B> [Online]
- [9] Servlet. <http://losteatinos.com/servlets/servlet.html> [Online]
- [10] JDBC. <http://dev.mysql.com/downloads/connector/j/>. [Online]
- [11] Wikipedia, “HTML”. <https://es.wikipedia.org/wiki/HTML> [Online]
- [12] CSS. [http://librosweb.es/libro/css/capitulo\\_1.html](http://librosweb.es/libro/css/capitulo_1.html) [Online]
- [13] Wikipedia, “JavaScript”. <https://es.wikipedia.org/wiki/JavaScript> [Online]
- [14] Wikipedia, “jQuery”. <https://fr.wikipedia.org/wiki/JQuery> [Online]
- [15] jQuery. <https://jquery.com/> [Online]

## 9 Anexo

### 9.1 Manual de usuario

En este capítulo abordo la explicación de la estructura y funcionamiento de la propia aplicación web. Para tener acceso a la aplicación, es imprescindible tener instaladas las herramientas Xampp, Bluej, MySQL Workbench y un lector de Texto en este caso Notepad++ explicadas en el capítulo 2.

Lo primero de todo es iniciar Apache, Mysql y Tomcat desde el panel de gestión de XAMPP, comprobar el puerto en que está operativo el Tomcat. Desde un navegador conectarse a la URL: `http://localhost:puerto/nombre_de_mi_carpeta/...`

En la página principal se autentifica los dos usuarios que van a jugar una partida.

#### Usuario registrado:

Si el usuario ya está registrado, hace un clic en el botón Iniciar sesión e introduce su correo y su contraseña (ver figura 45), se verifica que no se ha dejado ningún campo vacío, en caso afirmativo lo indica (ver figura 46), luego se verifica si el usuario existe en la base de datos y la contraseña es correcta sino lo indica (ver figura 47).

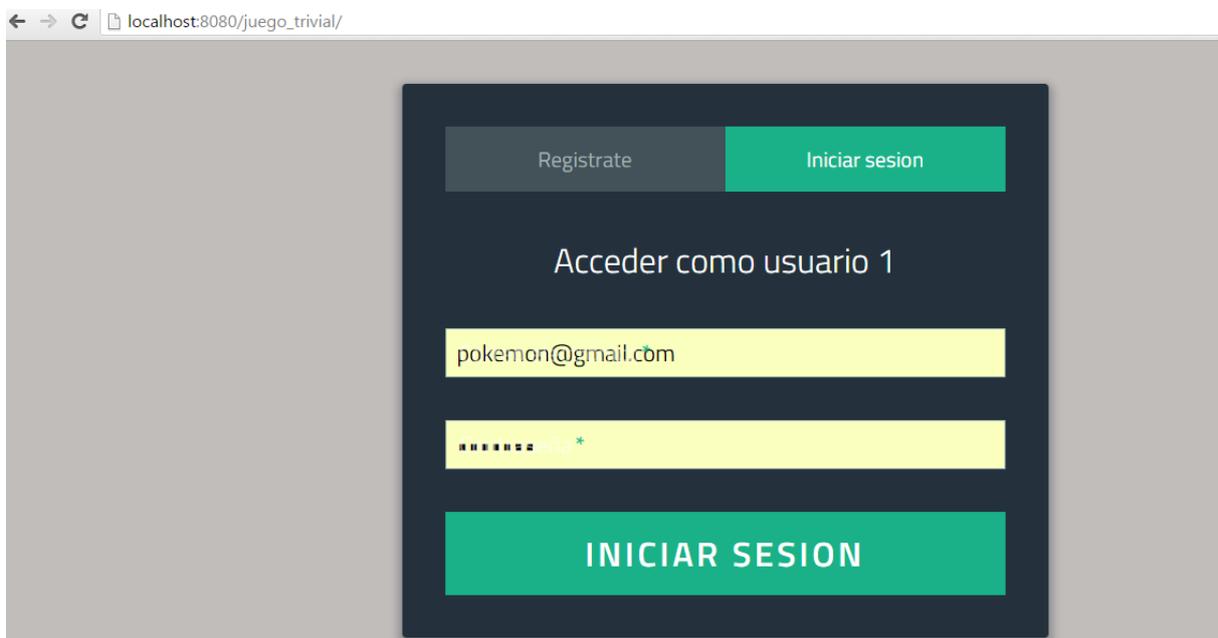


Figura 45: Acceder como usuario 1.

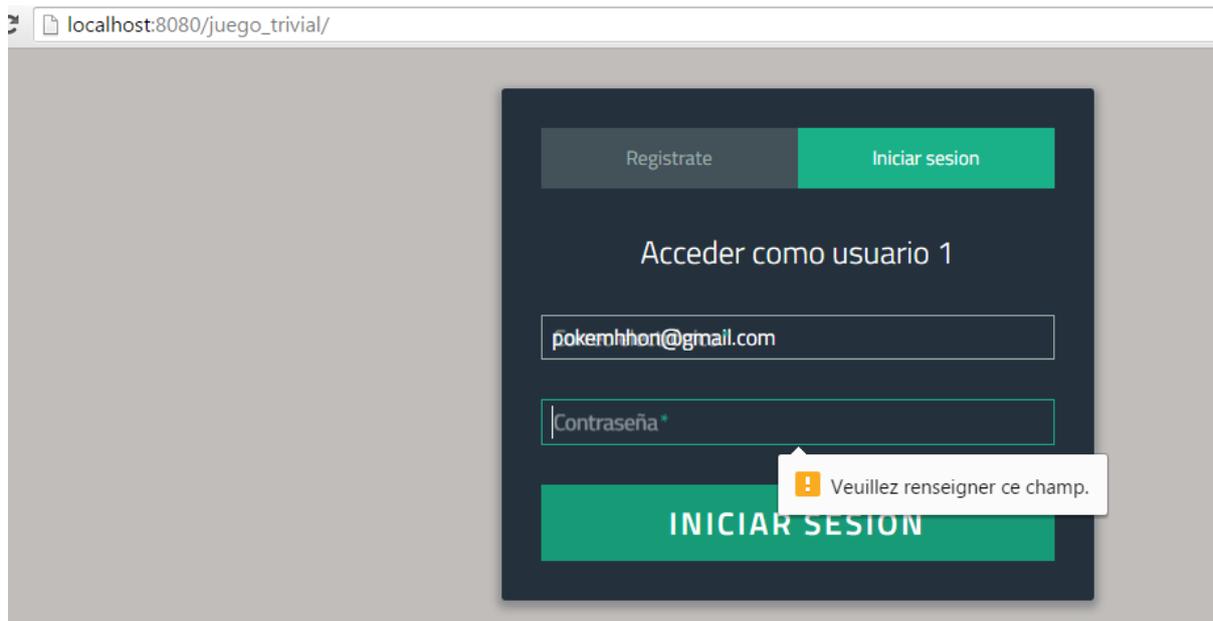


Figura 46: campo vacío.

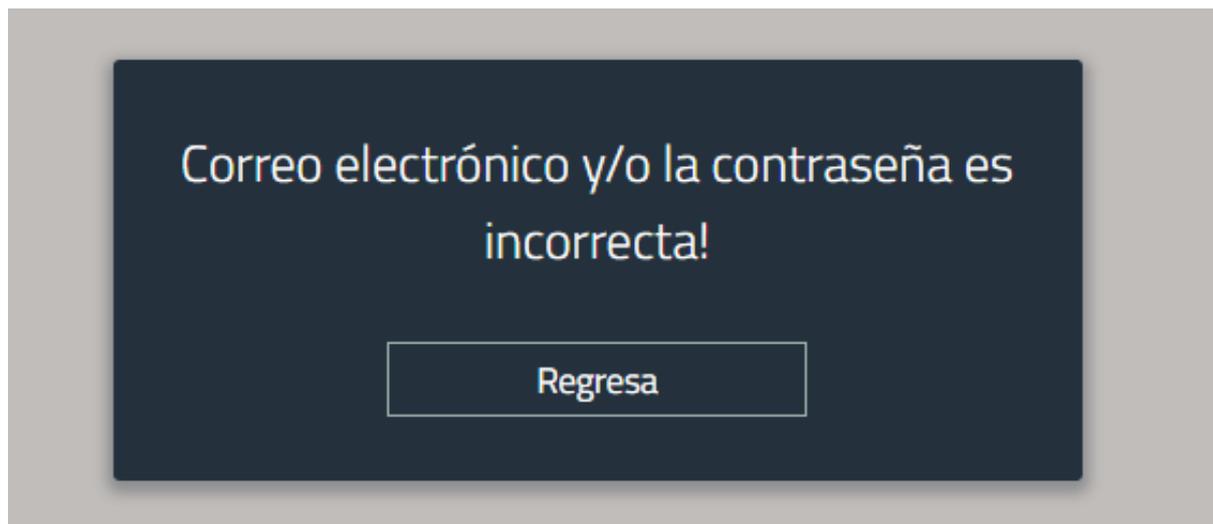


Figura 47: Información incorrecta.

#### Nuevo usuario:

Si el usuario es nuevo, debe crear una cuenta, para ello hace un clic en el botón Regístrate y rellene todos los campos (ver figura 48), se verifica primero que se ha rellenado todos los campos, en caso que se ha dejado algún campo vacío lo indica (ver figura 49), luego se verifica que el correo introducido no lo tiene otro usuario en caso que ya existe en la base de datos lo indica (ver figura 50).

Registrate      Iniciar sesion

## Crear una cuenta usuario 1

hajar      sbai

Nombre      Apellidos

hajar@gmail.com

Correo electrónico

.....

Establecer una contraseña

**EMPIEZA**

Figura 48: crear cuenta usuario 1.

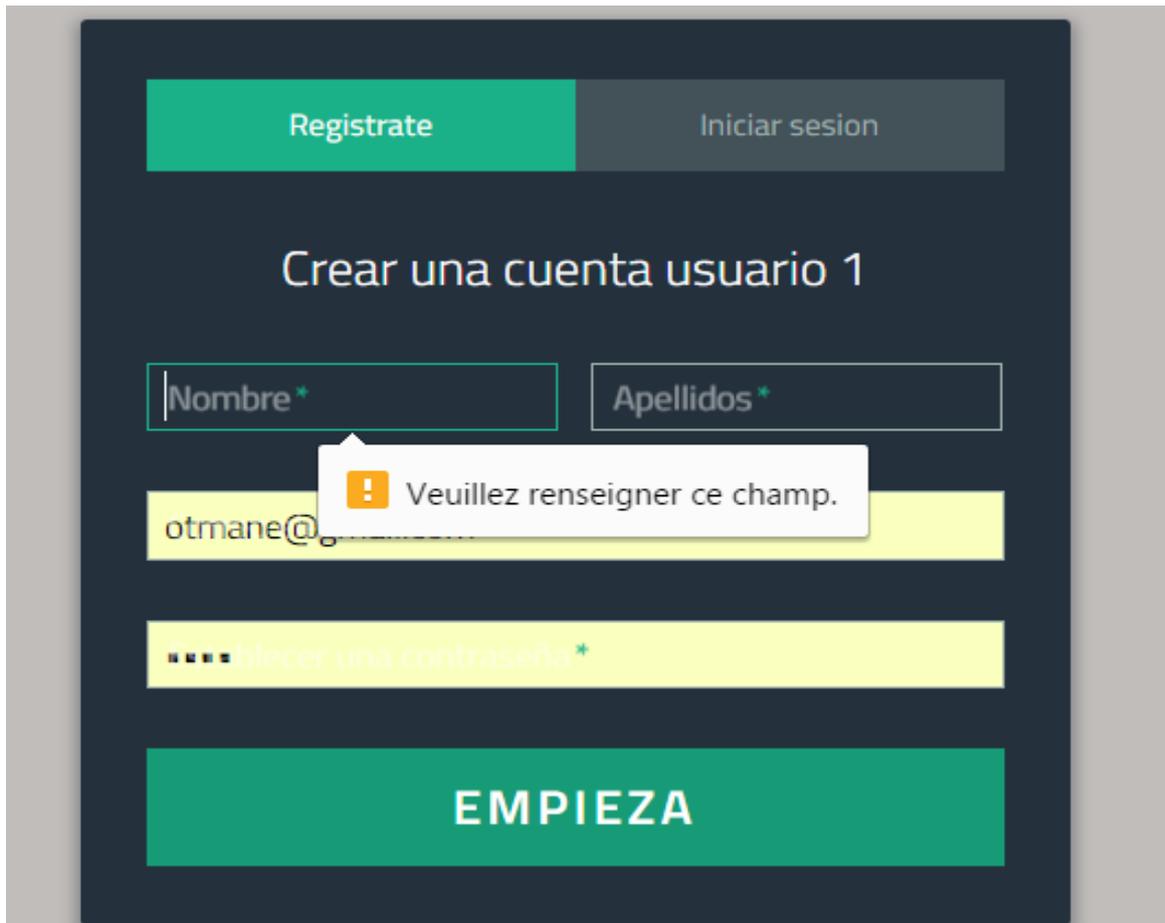


Figura 49: campo vacío usuario registrado.

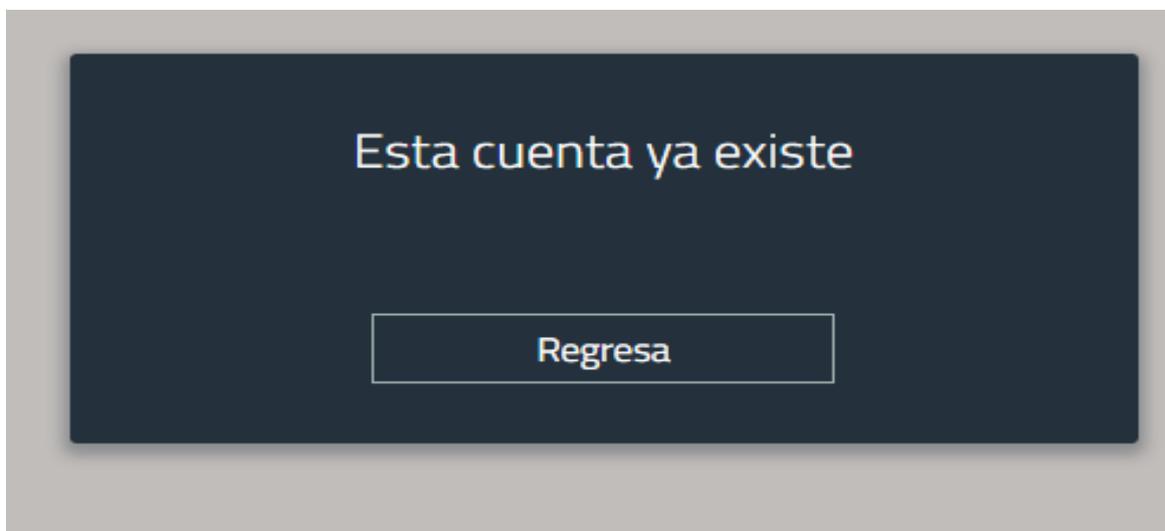


Figura 50: Información incorrecta usuario registrado.

Después de autenticar el usuario 1, habrá que autenticar el usuario 2, se hace el mismo procedimiento (ver figuras 51 y 52).

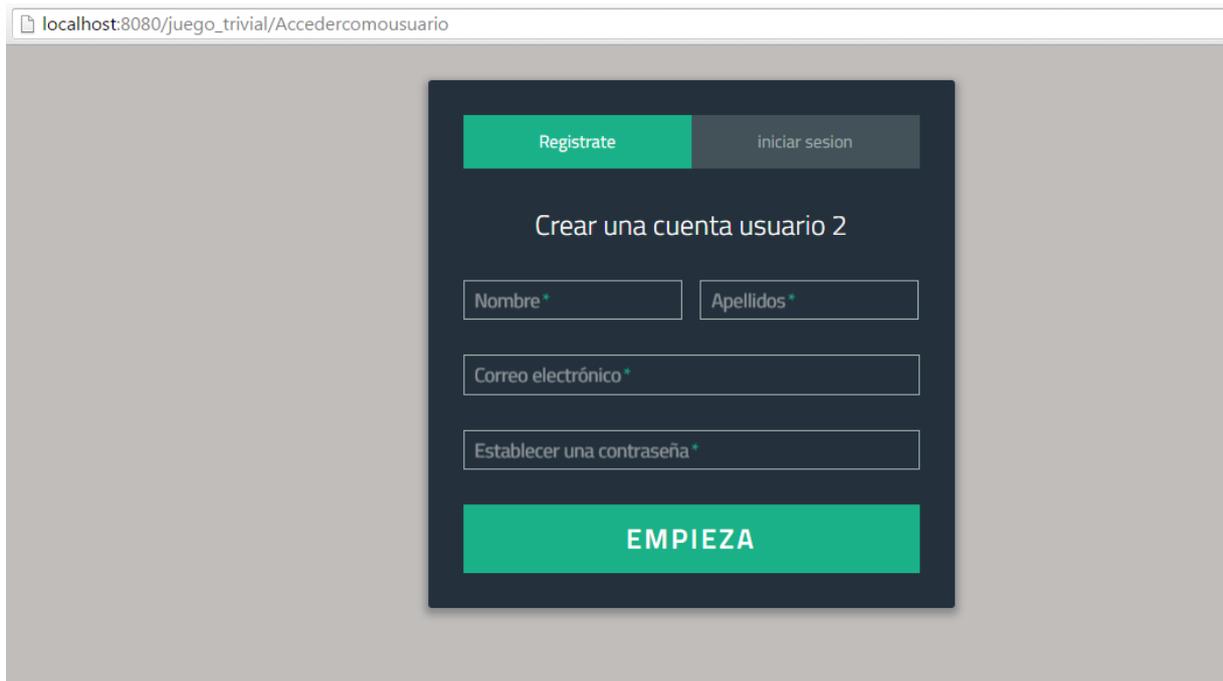


Figura 51: crear cuenta usuario 2.



Figura 52: usuario 2 registrado.

### Menú:

Una vez autenticados los dos jugadores se puede acceder al menú principal (ver figura 53). En este menú tenemos tres opciones:

- Nueva partida: Para empezar una nueva partida entre los dos jugadores.
- Partidas existentes: Para seguir jugando a una partida registrada en la base de datos.
- Ver estadísticas: Para ver las estadísticas de las partidas existentes para cada jugador.



**Figura 53: menú.**

*Nueva partida:* Al hacer clic en el botón nueva partida sale el juego, como podemos observar en la figura 54 el juego contiene un tablero de 64 casillas de diferentes colores, cada color representa una categoría de preguntas, de los cuales 6 son preguntas estrellas.

El juego dispone de 6 categorías:

- Rojo: Geografía.
- Rosa: Deporte.
- Azul: Ciencias.
- Verde: Historia.
- Amarillo: Arte.
- Naranja: Tecnología.
- Gris: Volver a tirar.

Los dos jugadores empiezan en la posición 1 del tablero.

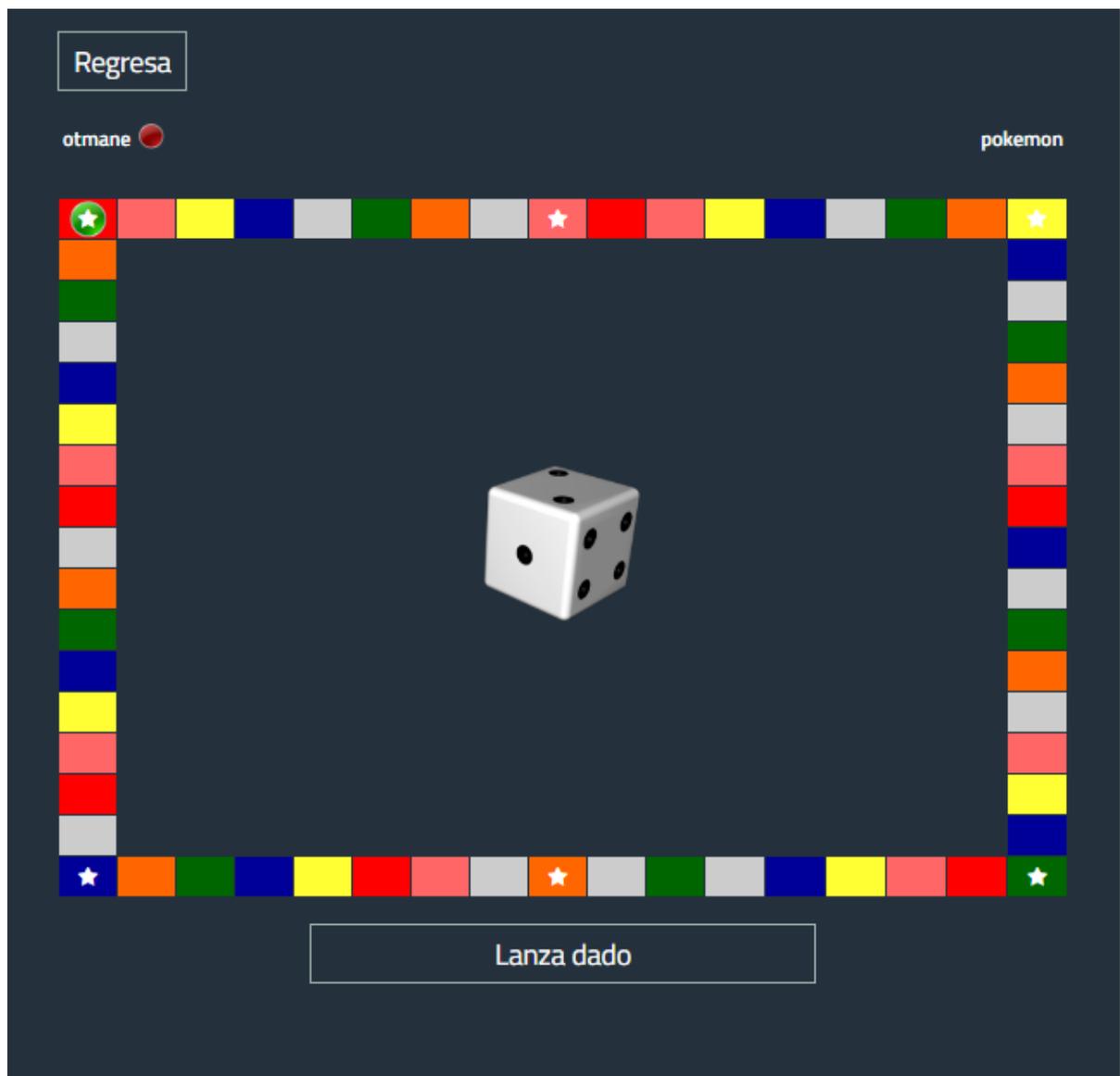


Figura 54: Tablero del juego.

Con el botón lanzar dado, se lanza el dado de la imagen y el jugador se mueve por el tablero según este dado.

Como podemos observar en la figura 55, el jugador 1 (otmane) tiene el turno y está en la posición 1, al lanzar el dado sale el número 3, el jugador puede moverse tres posiciones a la derecha o tres posiciones a la izquierda, esto quiere decir que debe elegir entre la tercera casilla a la derecha y la tercera casilla a la izquierda (ver figura56).

Cada casilla presenta una categoría, en este caso el jugador tiene elección entre volver a tirar o pregunta de ciencias (ver figura 57).

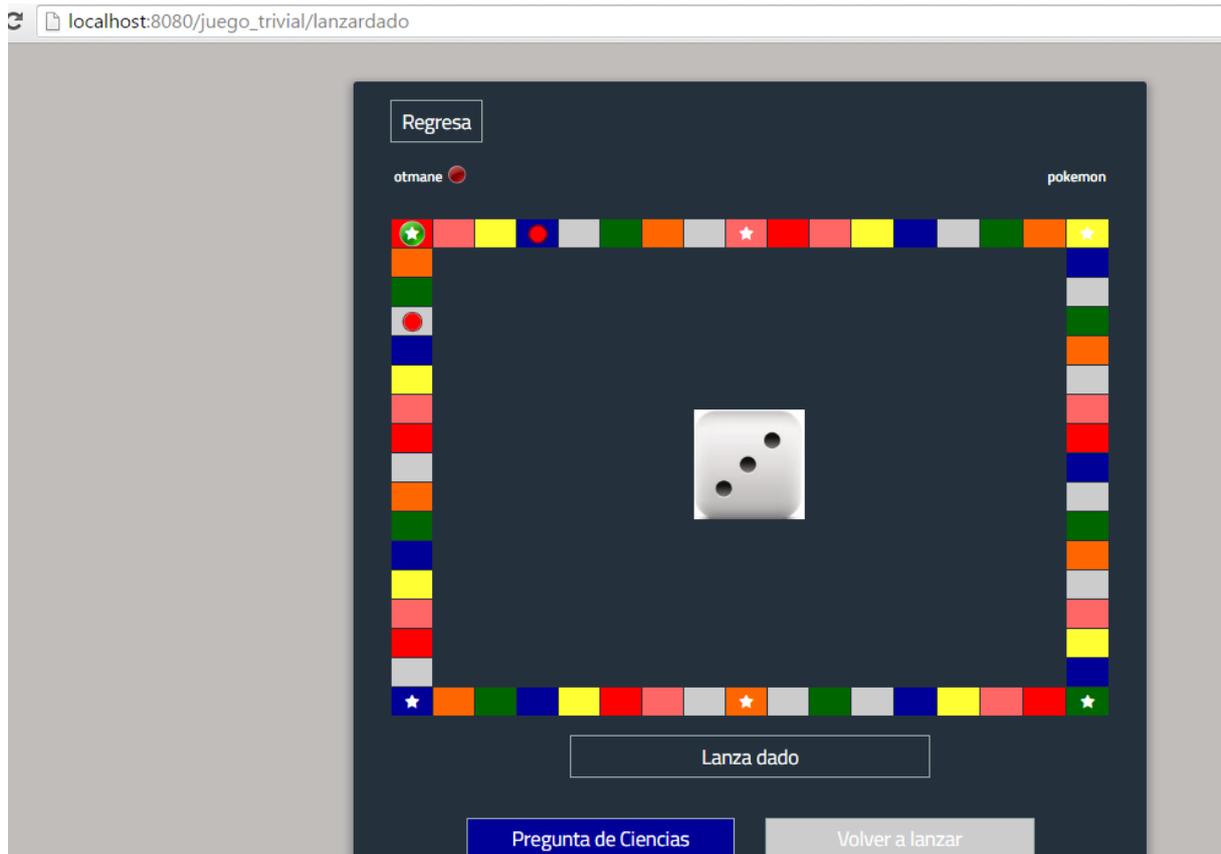


Figura 55: Lanzar el dado.

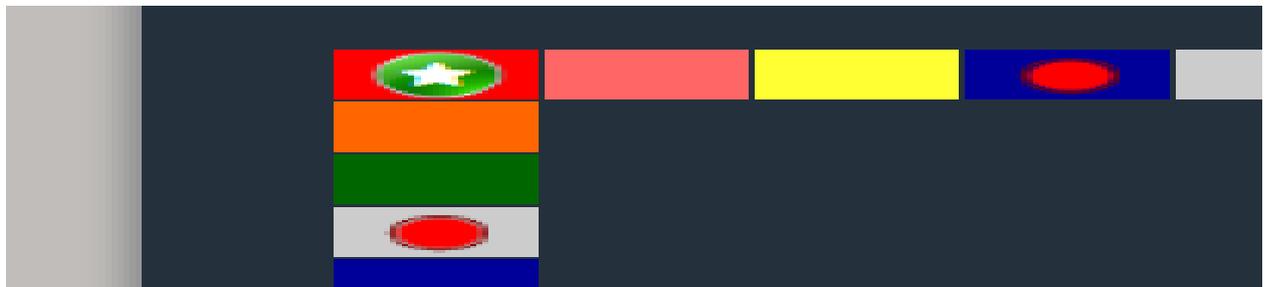


Figura 56: Elegir entre dos casillas.



Figura 57 : Elegir categoría.

Al elegir una categoría sale una pregunta y 4 respuestas, si la respuesta es correcta se pone en verde, se suma un punto en acertadas (parte de estadísticas que explicare posteriormente) a la categoría correspondiente y el jugador que está jugando no pierde su turno. Si la respuesta es falsa, se pone en rojo y se suma un punto en falladas a la categoría correspondiente y el turno se pierde. (Ver figuras 58 y 59).

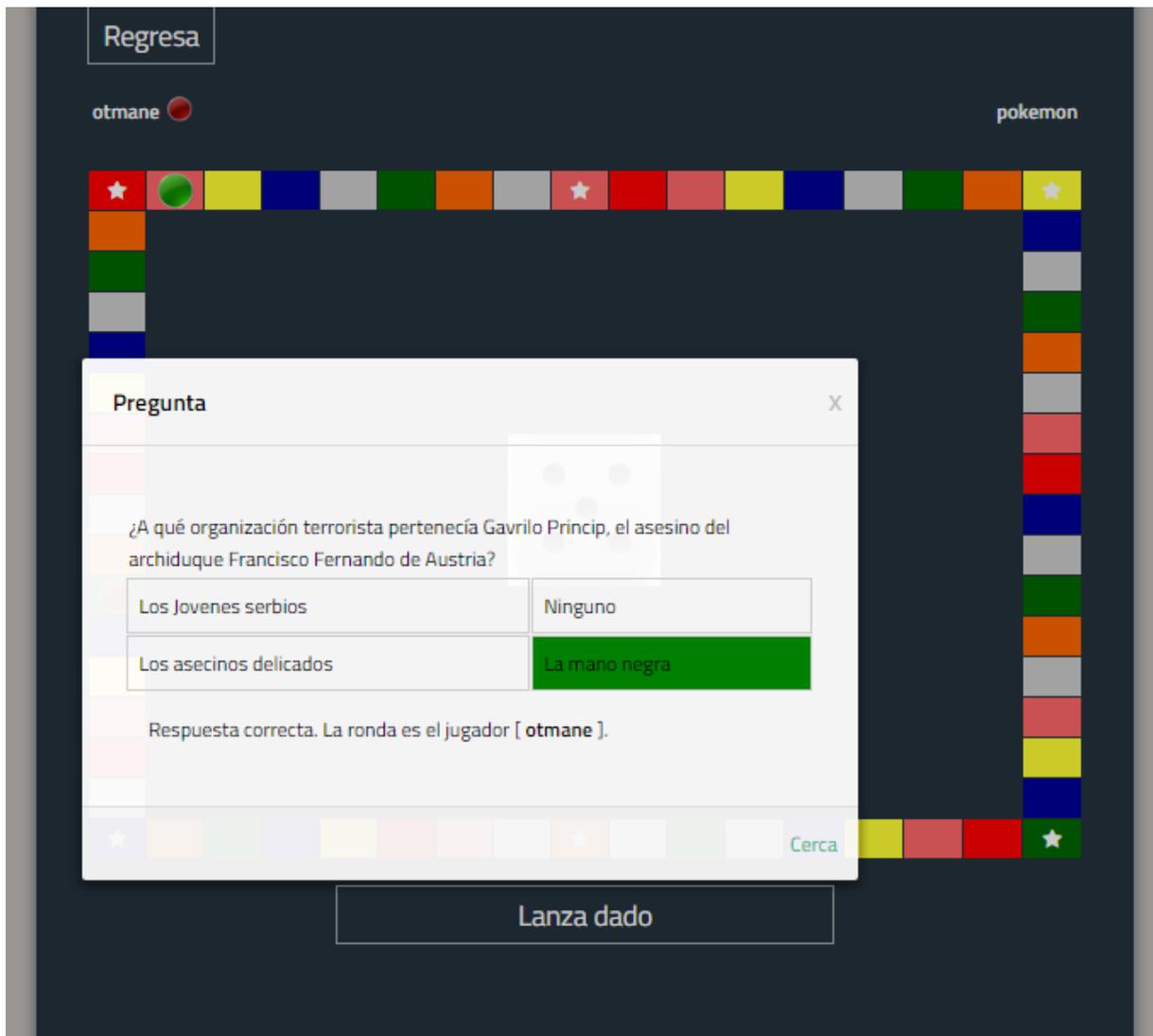


Figura 58: Caso de respuesta verdadera.

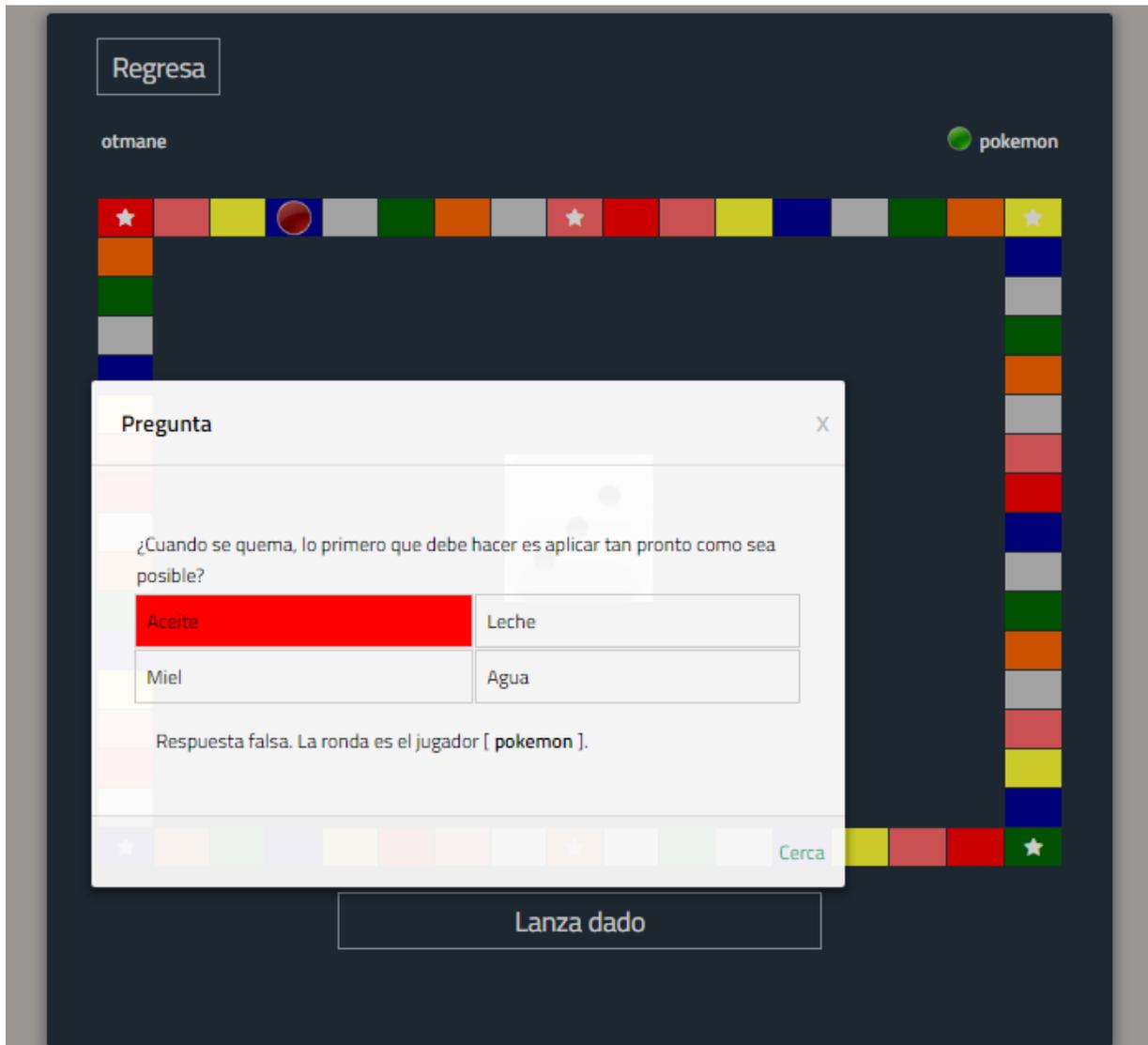


Figura 59: Caso de respuesta falsa.

Cuando un jugador responde a una pregunta estrella correctamente gana la estrella (ver figura 60). Si la pregunta estrella vuelve a caer y el jugador responde mal la pierde. El juego termina cuando un jugador gana las 6 estrellas (ver figura 61).



Figura 60: Jugador 2 ha ganado una estrella.

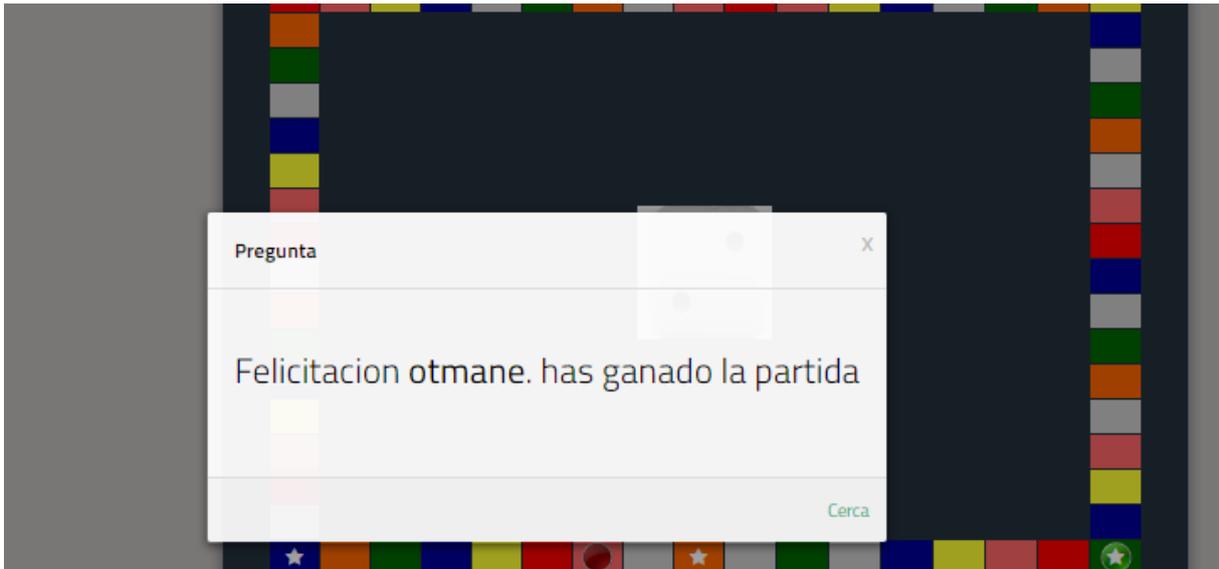


Figura 61: Fin de la partida.

*Partidas existentes:* Para ver las partidas existentes entre los dos jugadores hay que hacer clic en el botón partida existente y salen todas las partidas que han jugado (ver figura 62).

Para seguir jugando una partida, hay que hacer clic en ella. Para eliminarla hay que hacer clic en la cruz correspondiente a la partida.



Figura 62: Partidas existentes.

*Ver estadísticas:* Para ver las estadísticas hay que hacer clic en el botón ver estadísticas. Cada jugador tiene sus estadísticas de cada partida. (Ver figura 63 y 64). Podemos ver las estadísticas de todas las partidas de los 2 jugadores.

En la tabla de estadísticas podemos ver para cada categoría el número de puntos (preguntas) acertados y fallados y el total de puntos acertados y fallados. También podemos ver las estrellas ganadas y perdidas. Al responder correctamente a una estrella se suma un punto a las estrellas ganadas, si la respuesta es falsa se suma un punto a pérdidas (ver figura 65).



Figura 63: Estadísticas por jugador.



Figura 64: Lista de partidas para estadísticas.

## Las estadísticas

Categoría	Acertos	Fallos
Geografía	8	2
Deporte	1	1
Ciencias	5	3
Historia	0	7
Arte	1	6
Tecnología	4	1
Total	19	20

	Ganados	Perdidos
Estrellas	1	2

[Regresa](#)

Figura 65: Estadísticas de una partida.