2021

JPCGames

PROYECTO INTEGRADO

Joan Penyarrocha, Pau Navarro, Carlos Sanchis

INDEX

1. Datos del Equipo

- Joan Peñarrocha

Email: joanpe2400@gmail.com

Teléfono: 637610927

- Pau Navarro

Email: paunavarropalao3@gmail.com

Teléfono: 635925625

- Carlos Sanchis

Email: CarlosS-P79@hotmail.com

Teléfono: 633010575

2. Descripción y Problemáticas

- ## Descripción

  JPCGames es una plataforma de videojuegos que ira recopilando juegos clásicos a medida que se va actualizando y que cuenta con una página web donde se puede descargar esta aplicación.

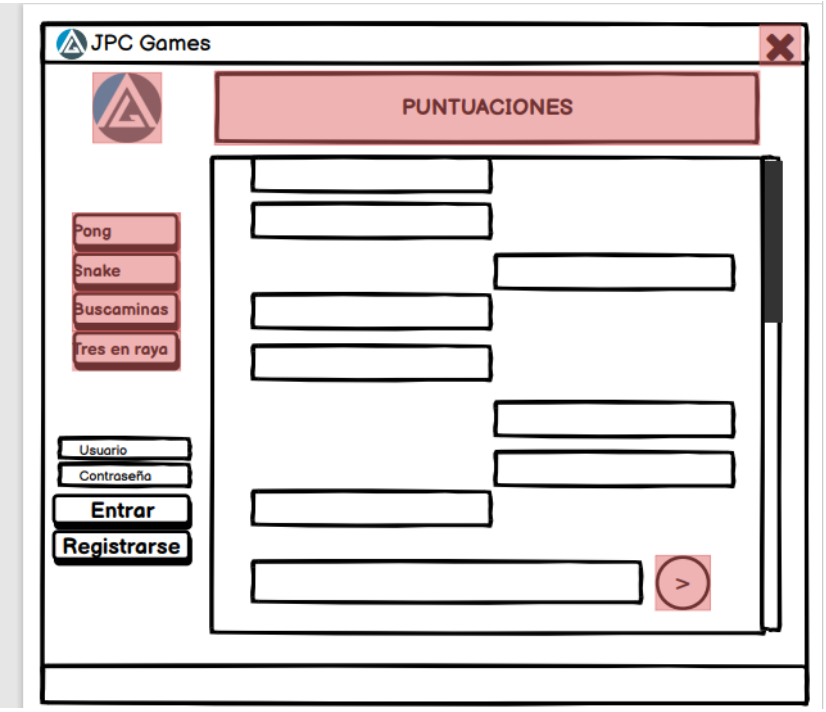
- ## Problemáticas

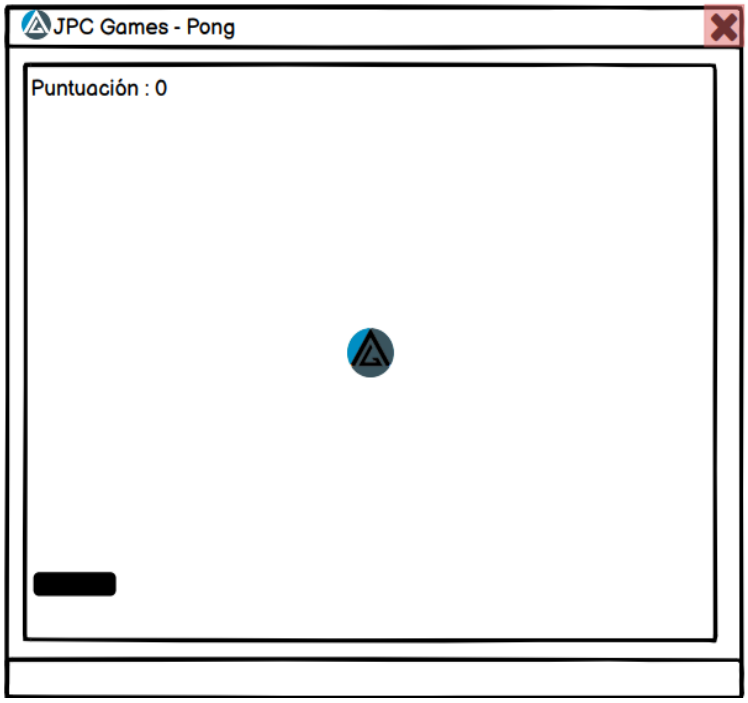
  La problemática que intenta solucionar es la accesibilidad y recuperación de juegos antiguos o clásico y modernizarlos de forma que la gente se vuelva a divertir con ellos.
  También resuelve la necesidad de encontrar juegos clásicos que ya no se encuentran disponibles o ya no son accesibles actualmente.

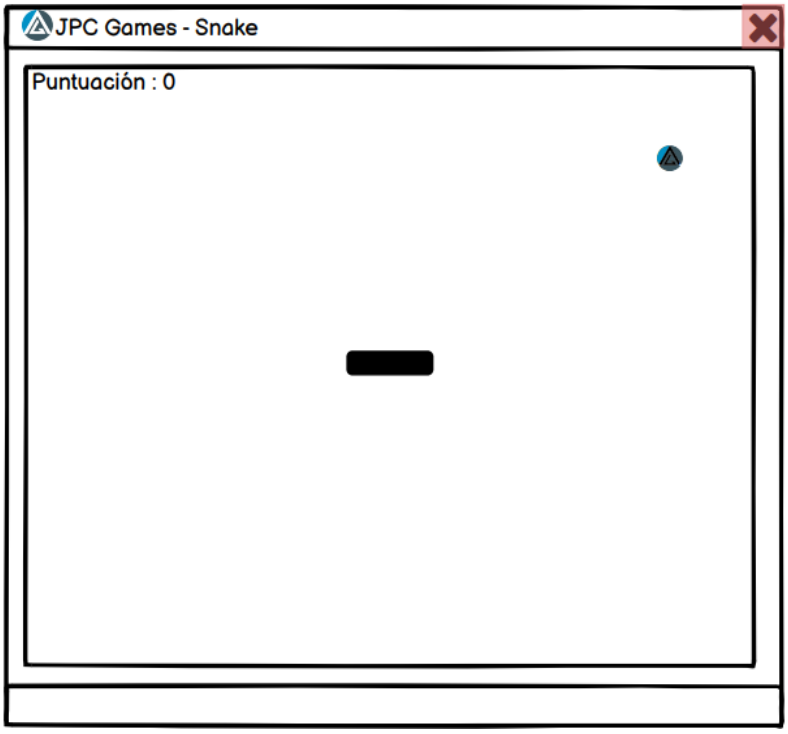
3. Desarrollo y Evolución

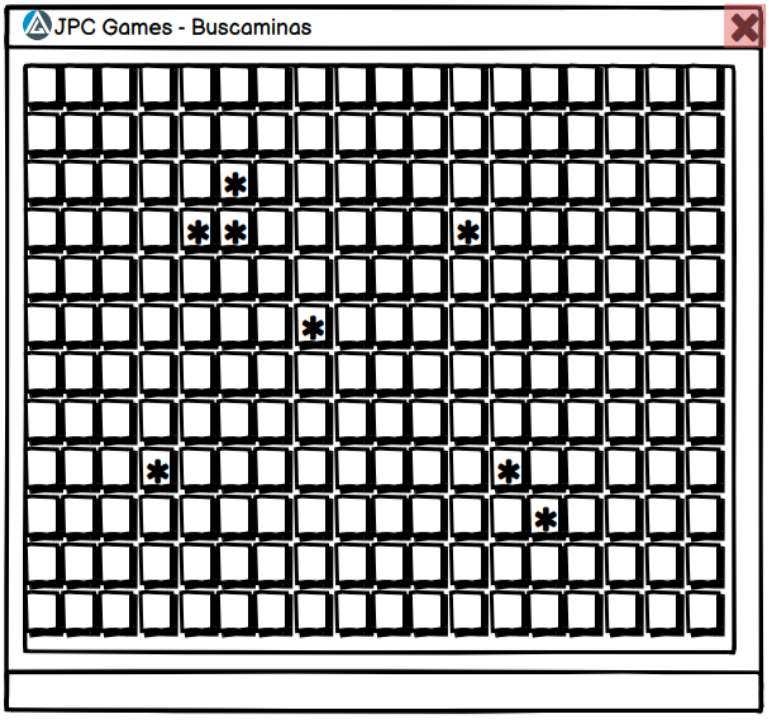
3.1 Creación del mockup

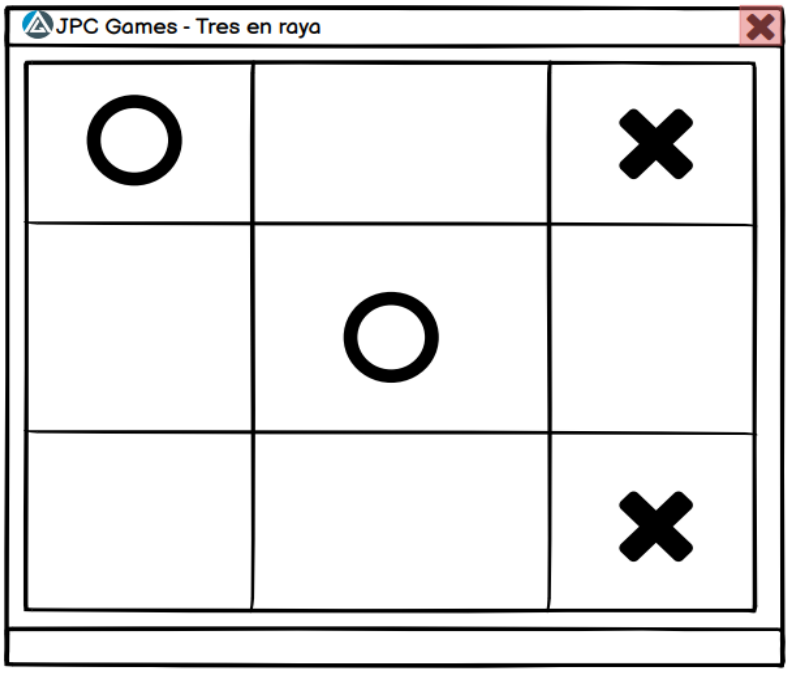
En el primer sprint de la semana comenzamos a hacer el mockup de la aplicación para tener una base y saber cómo nos gustaría que quedara la aplicación.

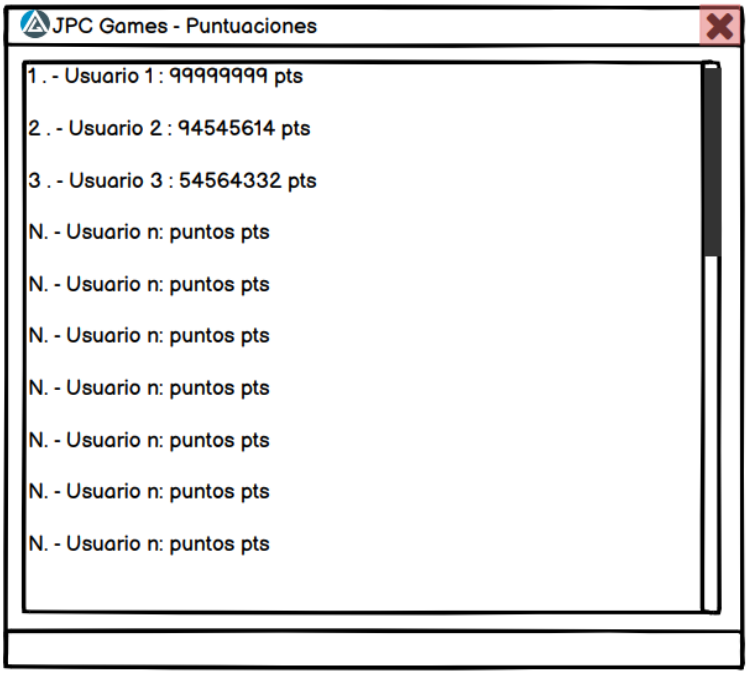






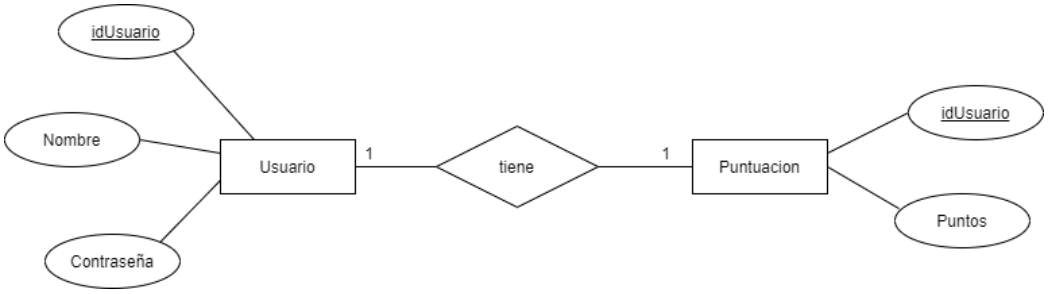



JPC Games - Tres en raya


JPC Games - Puntuaciones
1 . - Usuario 1 : 99999999 pts
2 . - Usuario 2 : 94545614 pts
3 . - Usuario 3 : 54564332 pts
N. - Usuario n: puntos pts
N. - Usuario n: puntos pts
N. - Usuario n: puntos pts
N. - Usuario n: puntos pts
N. - Usuario n: puntos pts
N. - Usuario n: puntos pts
N. - Usuario n: puntos pts

3.2 Creación de la base de datos

Una vez teníamos la base hecha, empezamos a hacer la base de datos de la aplicación. En este caso era muy sencilla, ya que solo recogíamos a los usuarios registrados y la puntuación total de todos los juegos.
Para empezar a hacer la base de datos primero hemos hecho la entidad-relación para tener la base.



Después de tener la base, ya hicimos el modelo relacional.

-Usuario(idUsuario, Nombre, Contraseña)

CP{idUsuario}

VNN{Nombre}

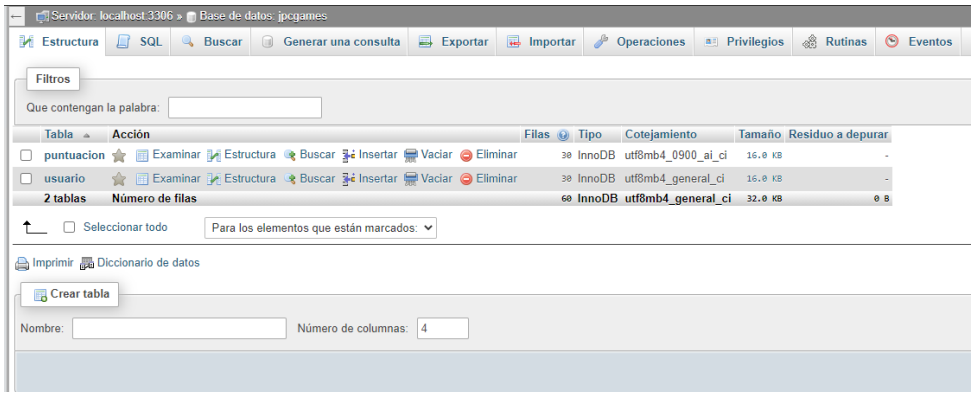
VNN{Contraseña}

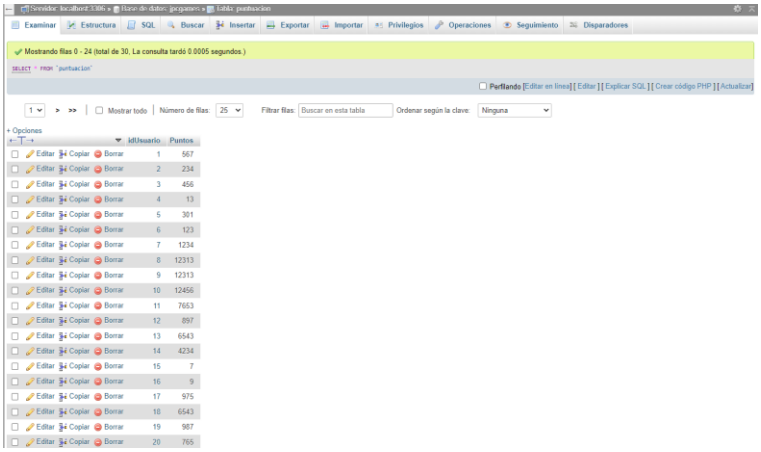
-Puntuacion(idUsuario, Puntos)

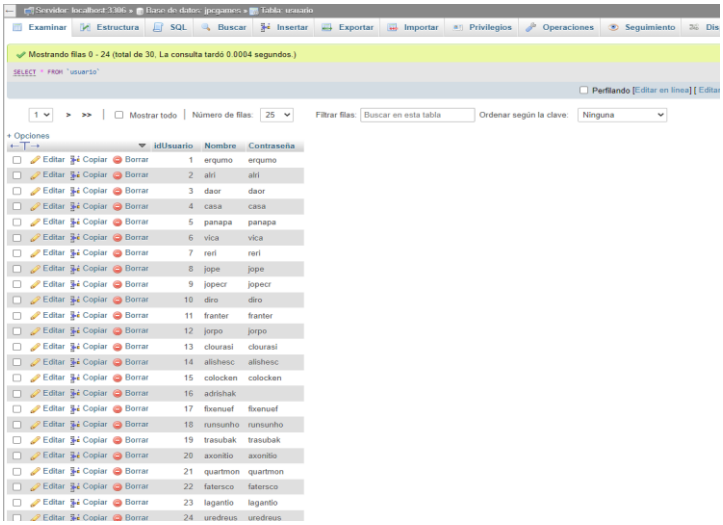
CP{idUsuario}

CA{idUsuario = Usuario}

Por último, implementamos la base de datos de nuestra aplicación con el phpmyadmin.



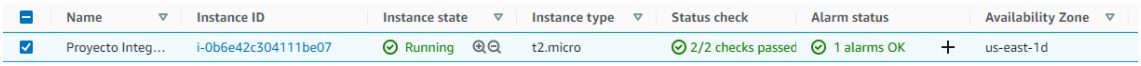


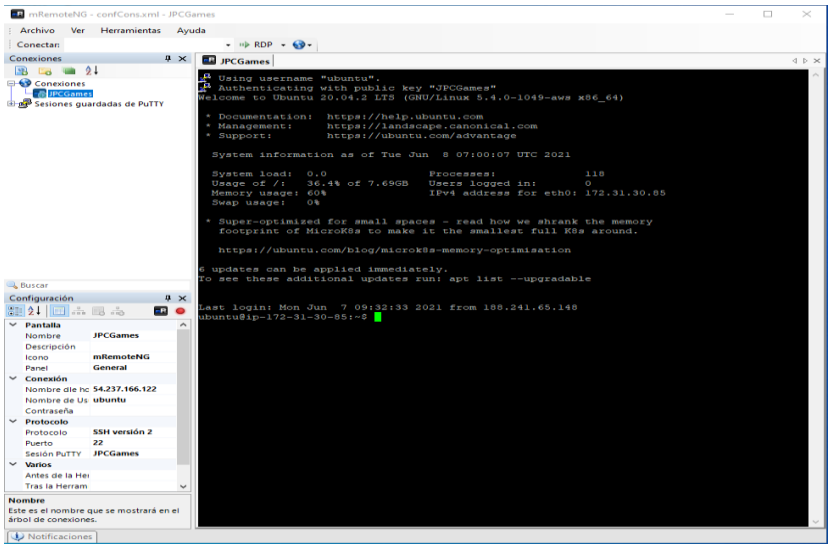


3.3 Infraestructura

- Cuando acabamos de hacer la base de datos, pensamos que había que alojarla en un algún sitio junto a la página web.
  En este caso, pensamos en crear una instancia EC2 de AWS e instalar en ella la pila LAMP para poder añadir la base de datos que estaba en nuestro localhost y la página web.

  Para poder instalar la pila LAMP, lo que hicimos es crear una máquina con Ubuntu Server y empezar a instalar el Apache, el MYSQL y el phpmyadmin.

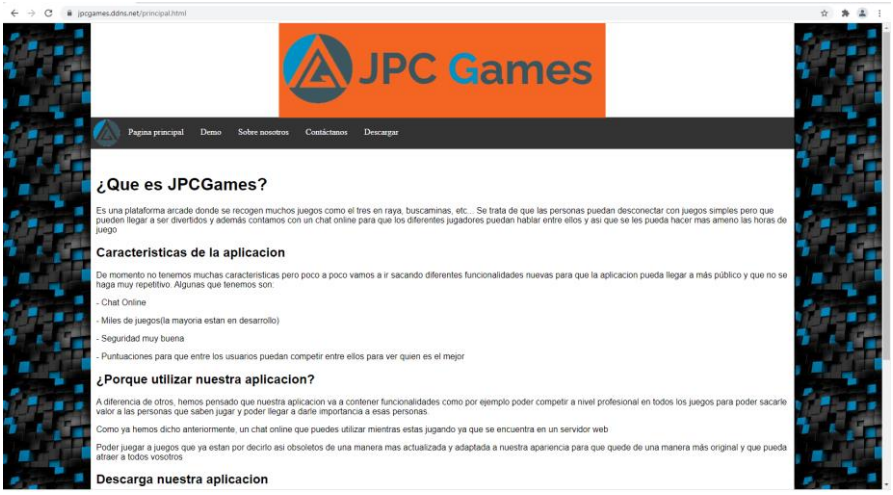




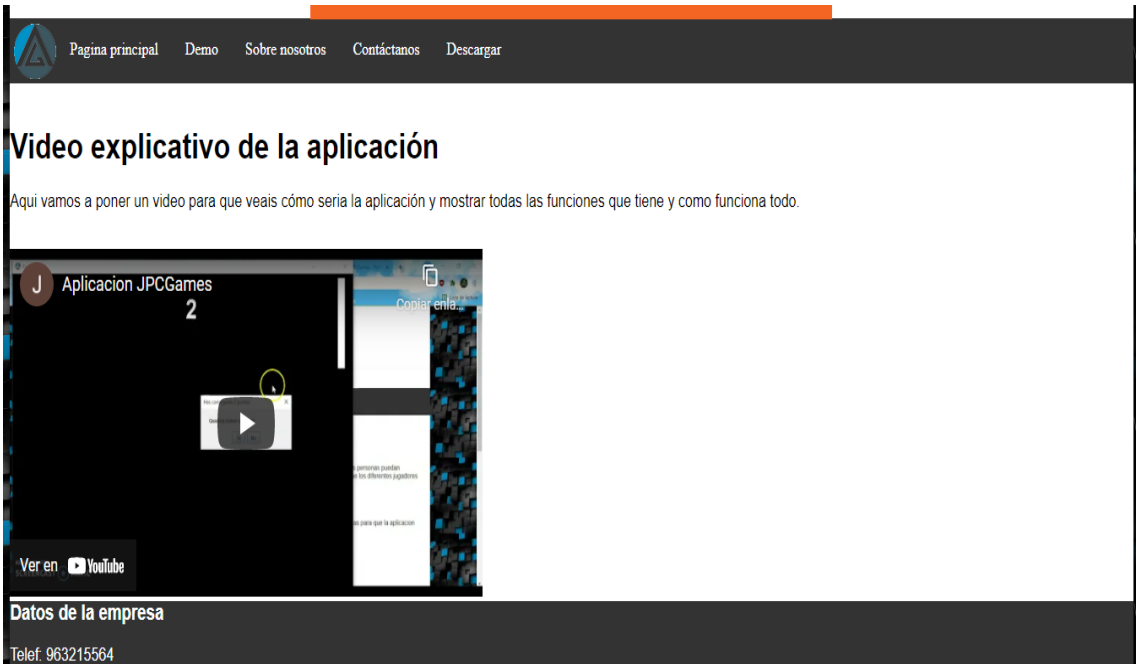
3.4 Página web

Pensamos que para que la aplicación pudiera llegar a todo el público había que hacer algo para darle publicidad.
Llegamos a la conclusión de que había que hacer una página web.
Nuestra página web consta de 5 partes:

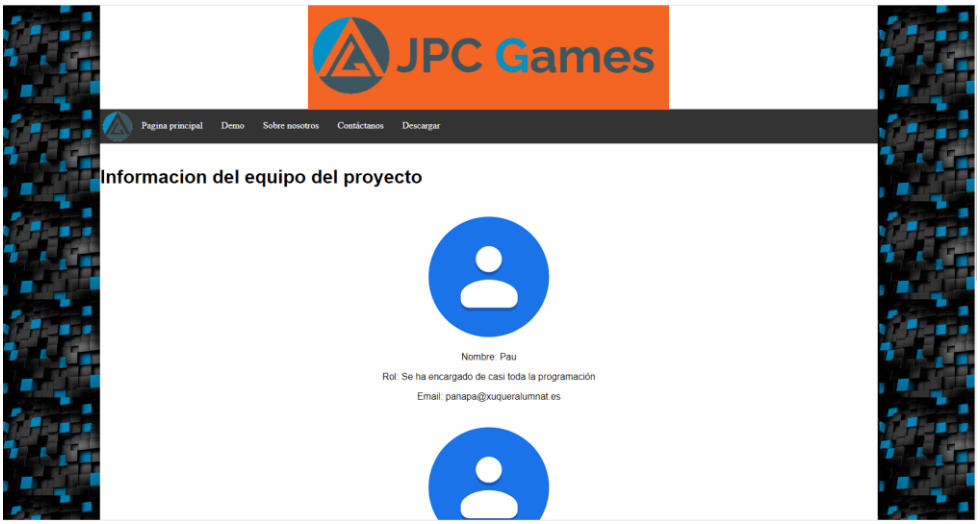
1. **La principal:** Donde se encuentra una ligera información de la aplicación.



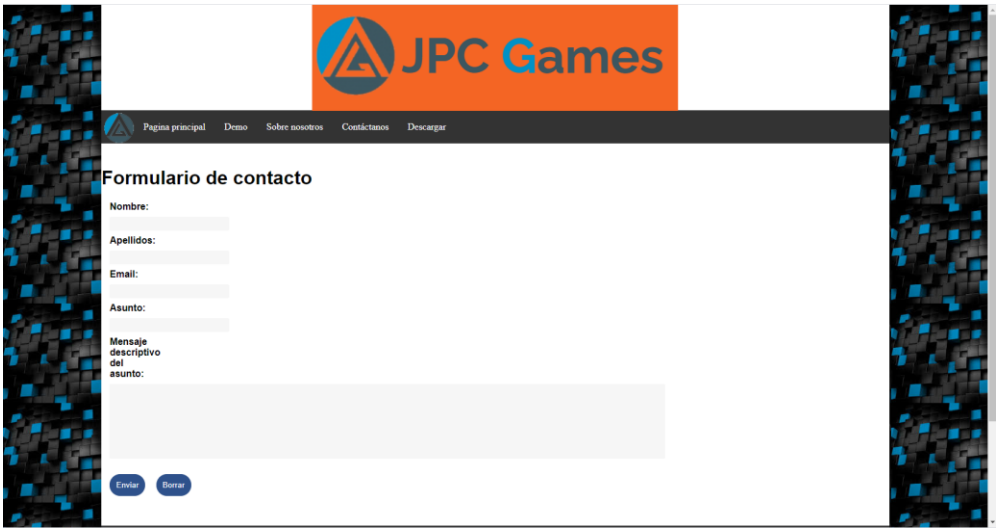
2. **Demo:** Donde se encontrara el video tutorial



3. **Sobre nosotros:** Donde está la información de los principales jefes del proyecto.



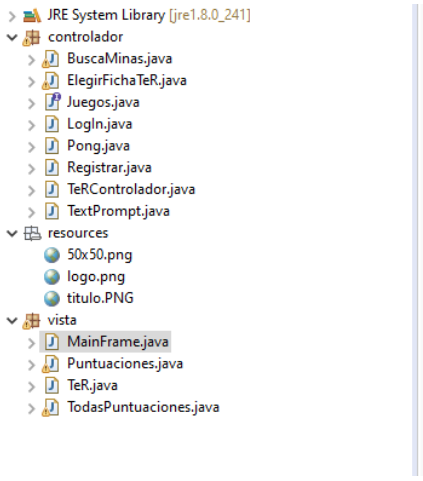
4. **Contáctanos:** Es el apartado donde nos podrán comunicar tanto por negocios o por errores que han encontrado.



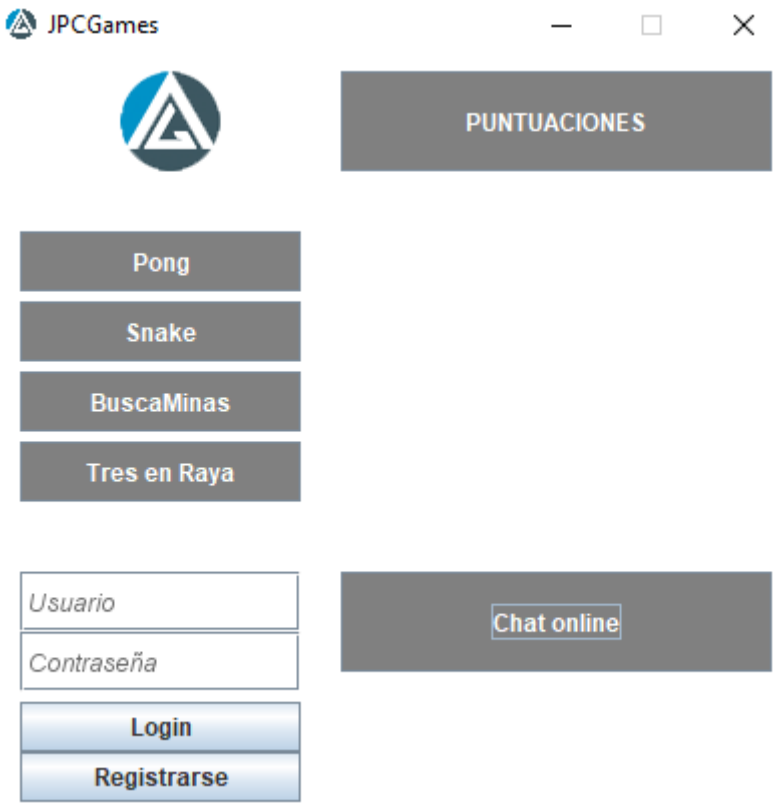
5. **Descargas:** donde al pulsar te descargaras la aplicación.

3.5 Creación de la aplicación, documentación y pruebas.

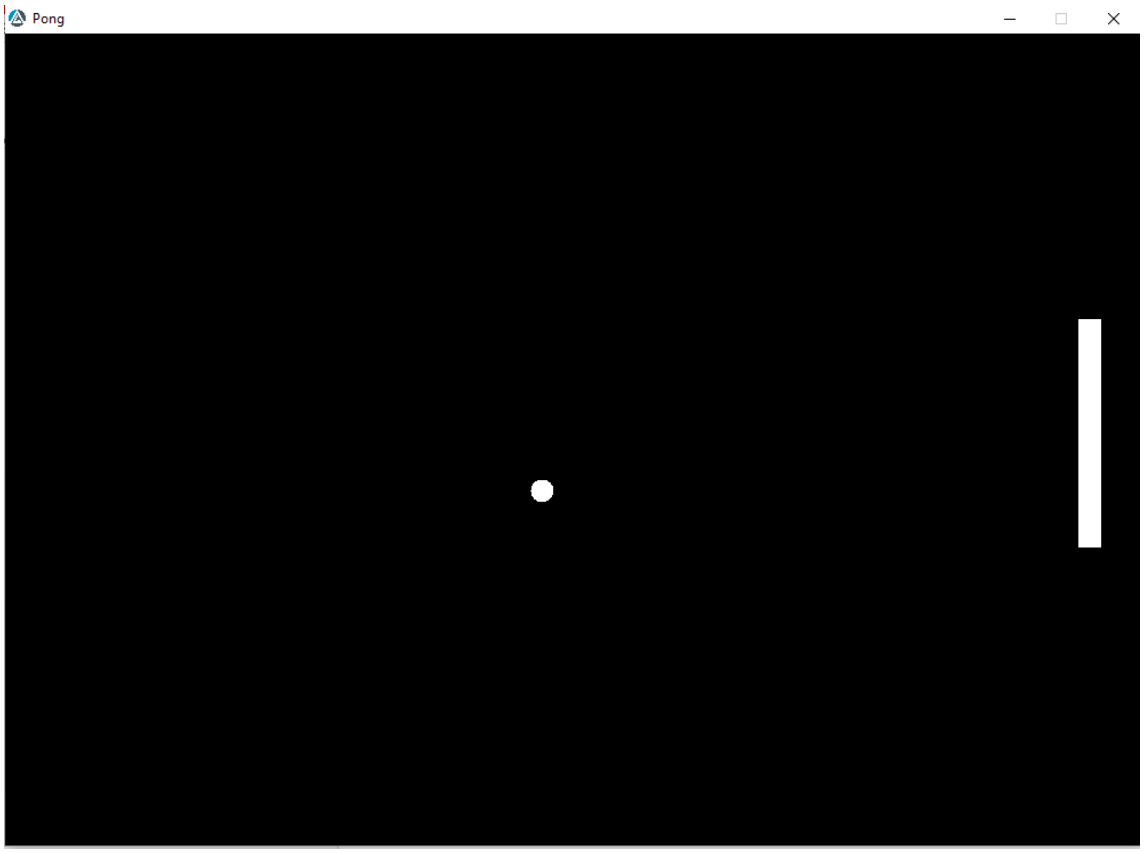
Una vez teníamos todo lo anterior hecho, había que empezar a hacer la aplicación.
Para hacerlo, decidimos utilizar el lenguaje Java.
Para la interfaz gráfica hemos utilizado las librerías de Java Swing
Todo esto lo hemos implementado en el IDE Eclipse.
Para que no estuviera todo junto, dividimos las diferentes clases en packages. En este caso, esta el package controlador, que es donde se encuentran todas las clases que crean la funcionalidad de los diferentes juegos o registros de usuarios.
Por otra parte, también está el package de vista, que es donde se lanza la interfaz gráfica de la aplicación.



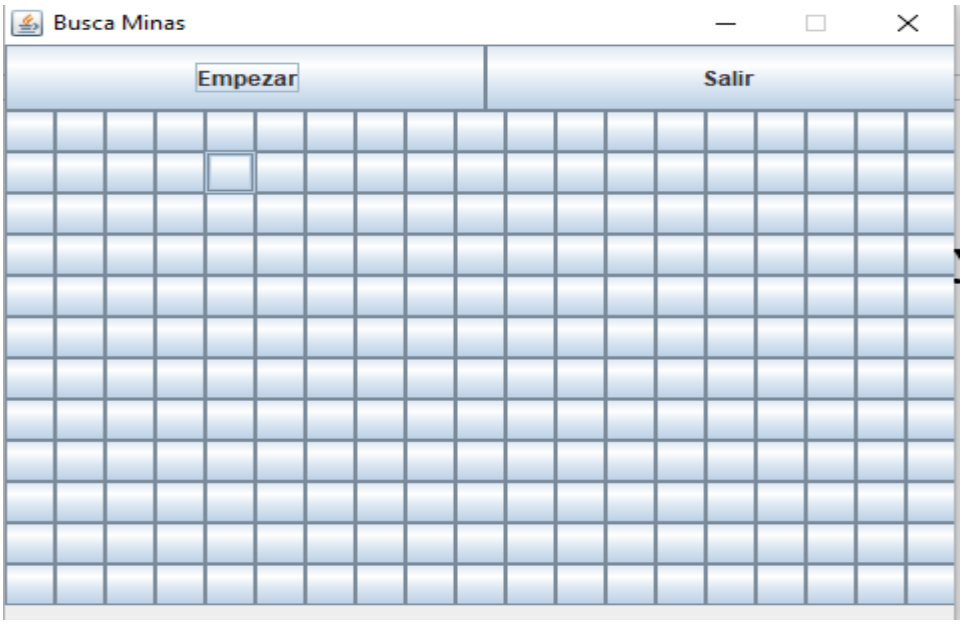
Aquí os dejamos como ha quedado una vez hemos programado la primera versión de nuestra aplicación.
- Esta es la pestaña principal donde se podrá acceder a los diferentes juegos como el Pong, Busca minas, Tres en Raya (“El Snake para futuras versiones”) también se podrá acceder a la página web tanto a las puntuaciones y chat online por último te puedes tanto como iniciar sesión como registrarte.



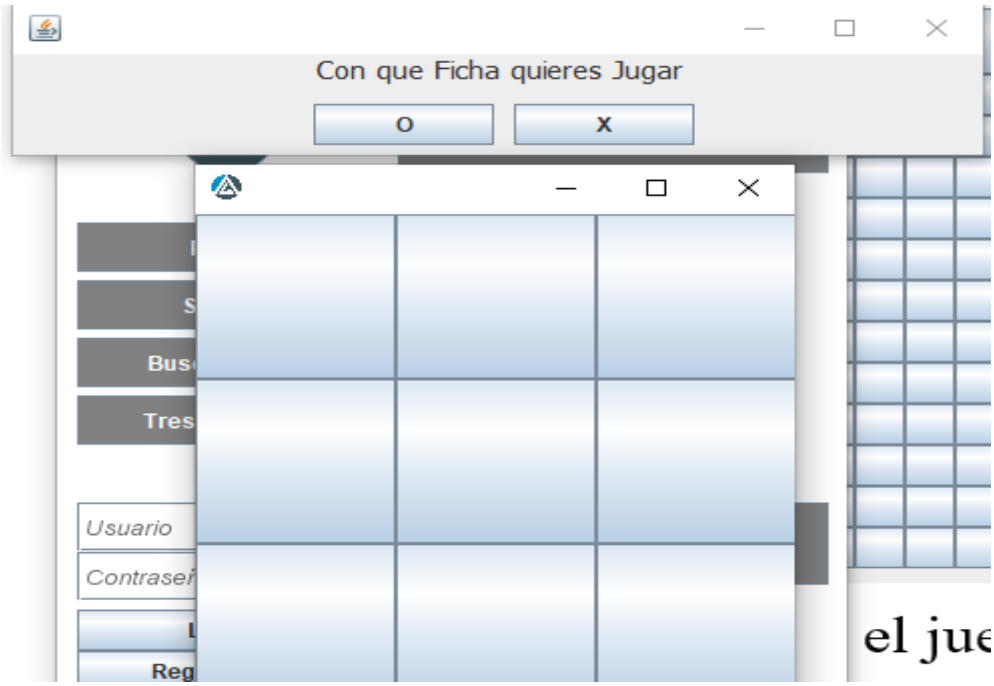
Este es el resultado final del pong.
El objetivo de este juego es evitar que la pelota golpee en la parte derecha de la ventana, intentando acumular la mayor cantidad de puntos.



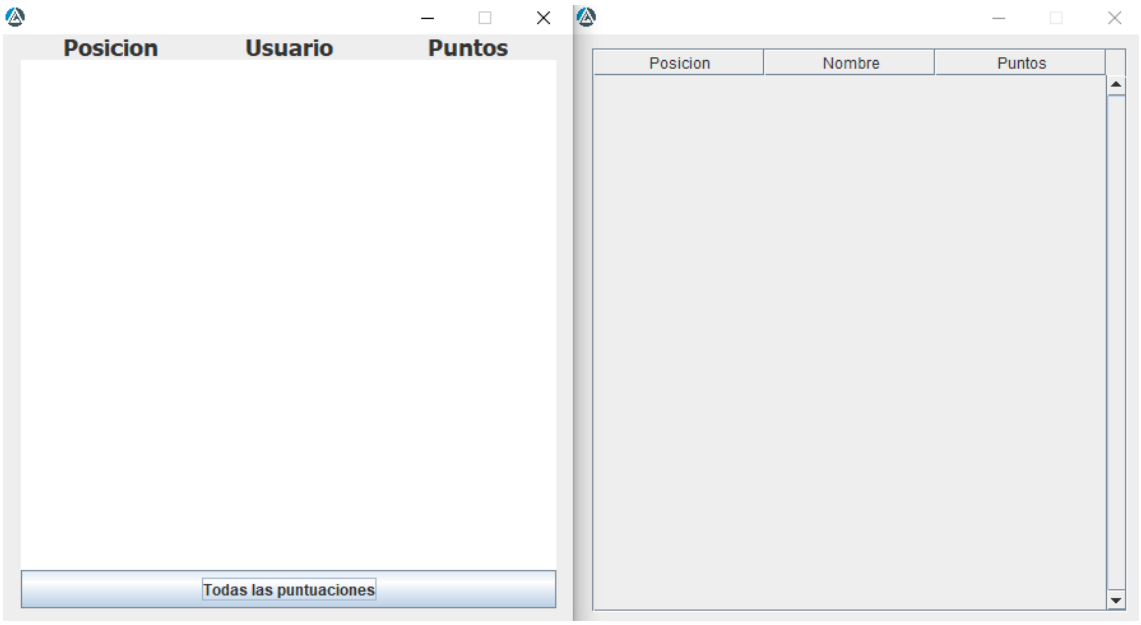
A partir de aquí empezamos el segundo sprint donde solo quedava acabar la aplicación y crear el plan de prevención, el organigrama y el Flyer.
Este es el resultado final del buscaminas.
El objetivo de este juego es intentar completar la tabla sin pulsar una mina guiándote por los números que te dicen cuántas minas puede haber alrededor de esa casilla.



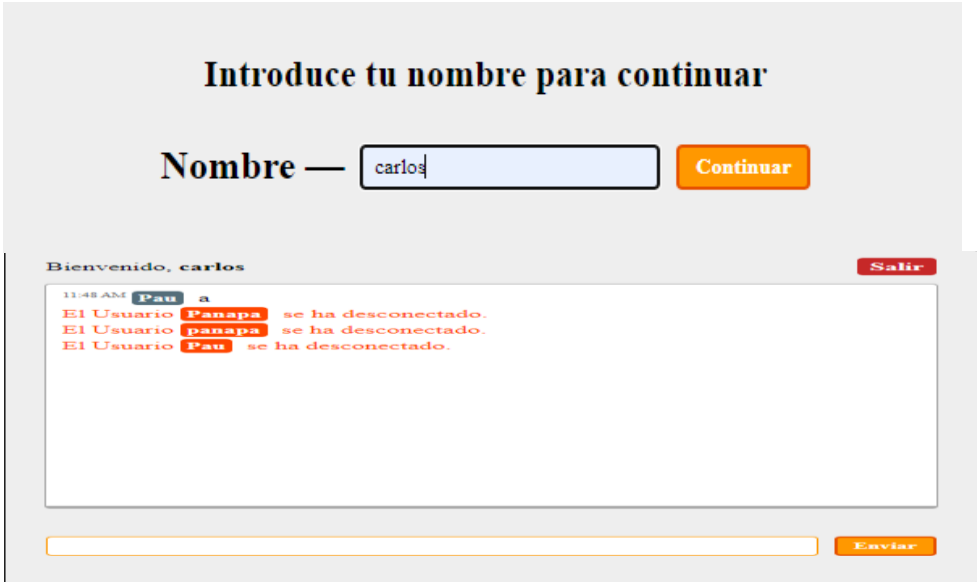
En el tres en raya eliges ficha primero para poder jugar contra la máquina, las reglas son intentar alinear tres de la misma ficha ya sea de derecha a izquierda, de arriba abajo o en diagonal.



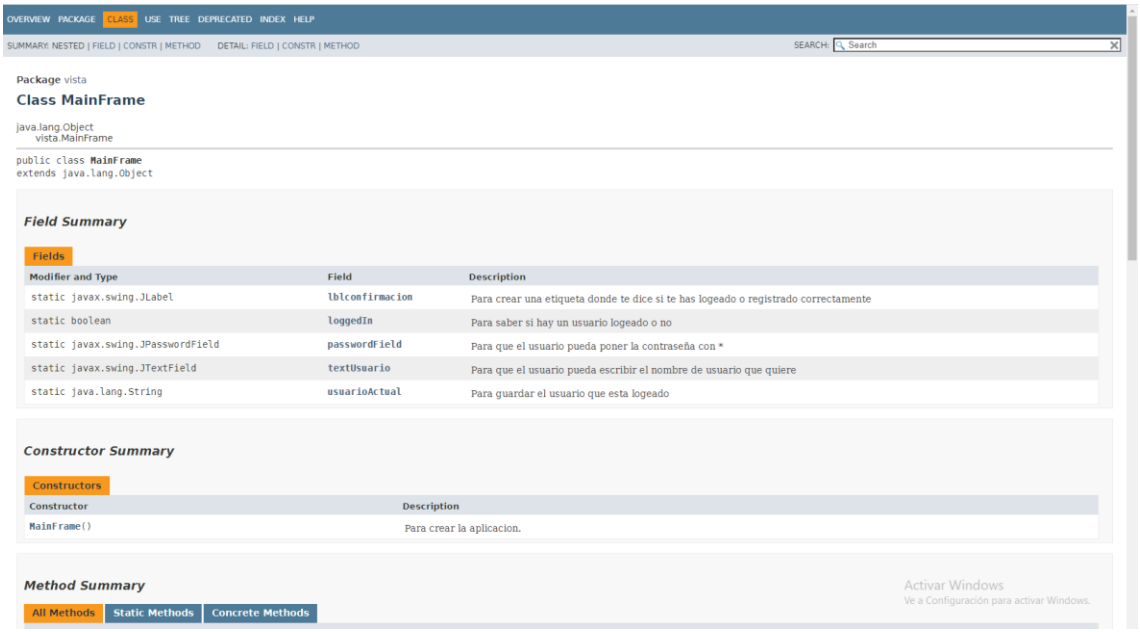
Aquí se muestran las 25 primeras puntuaciones que es el panel de la derecha y para poder ver todas las puntuaciones hay que pulsar el botón de debajo que es el panel de la derecha de la imagen.



Este es el chat online que lo hemos hecho mediante php.
En el momento que pulsas el botón se te abre una página web que es donde se encuentra el chat. Simplemente pones el nombre de usuario y ya directamente entras en el chat que es donde ya puedes hablar con todas las personas que estén conectadas en el momento.



Después de haber terminado la aplicación empezamos a documentar la aplicación para poder generar el Javadoc.



OVERVIEW PACKAGE CLASS USE TREE DEPRECATED INDEX HELP

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD SEARCH: Search

Package vista

Class MainFrame

java.lang.Object
vista.MainFrame

public class **MainFrame**
extends java.lang.Object

Field Summary

Fields

Modifier and Type	Field	Description
static javax.swing.JLabel	lblconfirmacion	Para crear una etiqueta donde te dice si te has logeado o registrado correctamente
static boolean	loggedIn	Para saber si hay un usuario logeado o no
static javax.swing.JPasswordField	passwordField	Para que el usuario pueda poner la contraseña con *
static javax.swing.JTextField	textUsuario	Para que el usuario pueda escribir el nombre de usuario que quiere
static java.lang.String	usuarioActual	Para guardar el usuario que esta logeado

Constructor Summary

Constructors

Constructor	Description
MainFrame()	Para crear la aplicacion.

Method Summary

All Methods Static Methods Concrete Methods

Activar Windows
Ve a Configuración para activar Windows.

Una vez hecho esto hacer las pruebas necesarias con el JUnit para que la aplicación funcionase correctamente.

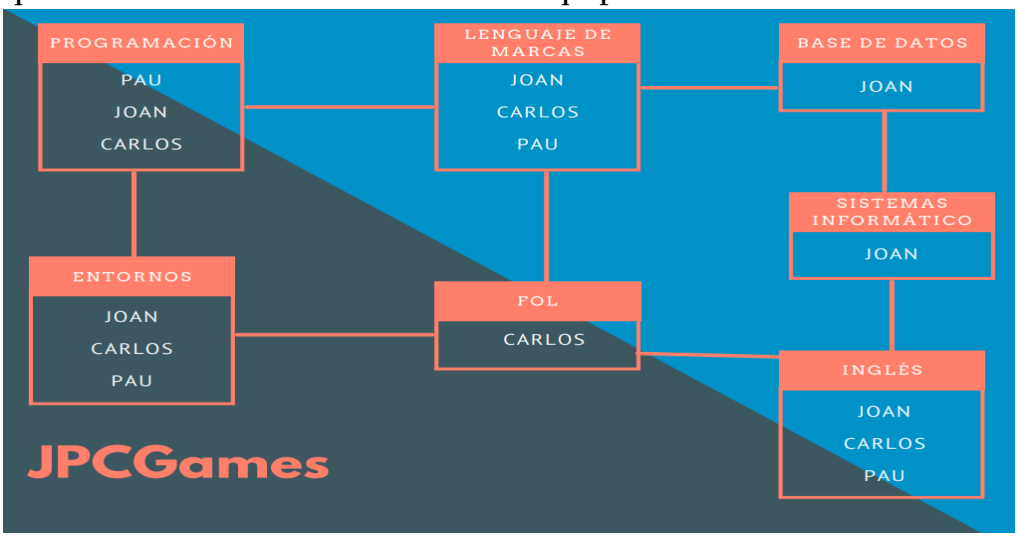
```
1  import static org.junit.jupiter.api.Assertions.*;
7
8  class Prueba {
9
10     @Test
11
12     void testTresenRayaUsuario() {
13
14         TeR ter = new TeR();
15
16         ter.btnArrIzq.setText("X");
17         ter.btnArrMed.setText("X");
18         ter.btnArrDer.setText("X");
19
20
21         boolean res = TeRControlador.chkTeRUsu("X");
22
23         assertTrue(res);
24
25         ter.btnArrIzq.setText("O");
26         ter.btnArrMed.setText("O");
27         ter.btnArrDer.setText("O");
28
29
30         res = TeRControlador.chkTeRUsu("O");
31
32         assertTrue(res);
33
34
35         ter.btnMedIzq.setText("X");
36         ter.btnMedMed.setText("X");
37         ter.btnMedDer.setText("X");
38
39
40         res = TeRControlador.chkTeRUsu("X");
41
42         assertTrue(res);
43
44
45         ter.btnMedIzq.setText("O");
46         ter.btnMedMed.setText("O");
47         ter.btnMedDer.setText("O");
```

Una vez acabada la aplicación, llegamos a la conclusión de que habían varios errores. Para ello hemos decidido empezar a hacer una segunda versión donde añadiremos puntuaciones de cada juego, un perfil donde los usuarios puedan ver sus diferentes rankings y varios juegos que están en desarrollo.

3.6 Marketing y prevenciones

Una vez realizada la aplicación, empezamos a crear el plan de prevenciones de riesgos que hay que tener en cuenta para evitar accidentes en el trabajo.

Seguidamente hicimos un organigrama donde se muestra el trabajo que ha realizado cada miembro del equipo.



Por último para terminar todo el proyecto hicimos un Flyer donde mostramos lo más importante de la aplicación en una simple portada.

4. URL de los sitios que hemos utilizado

- URL del Github:
  https://github.com/PauNavarro/JPCGames

- URL del Trello:
  https://trello.com/b/6iG91SCd/jpcgames

- URL de la Página web (Aquí se encuentra el video tutorial):
  https://jpcgames.ddns.net/

- URL de páginas que han sido de ayuda:
  https://docs.oracle.com/javase/7/docs/api/

  https://www.w3schools.com/html/