

# Manual Técnico Arkanoid

Realizado por:

—

Jose Roberto Nasser Sanchez 00245818

Roberto Carlos Orellana Chávez 00163618

Sebastian Ernesto Magaña Ramírez 00186118

Rodrigo Arevalo Najarro 00303318

Ricardo Daniel Henriquez Majano 00033218

## CONTENIDOS:

1. Objetivo de el documento.
2. Descripción general del proyecto.
3. Software utilizado para la creación del proyecto.
4. UML del proyecto.
5. Diagrama Entidad-Relación extendido(referente a la base de datos).
6. Diagrama relacional.
7. Conceptos tecnicos y distintos tipos de error.
8. Nomenclaturas
9. Eventos y excepciones.

## Objetivo del documento

El objetivo de este documento tiene la finalidad de orientar y dar un mayor rango de entendimiento acerca del diseño del software, explicando con diferentes herramientas la finalidad de este proyecto para así poder estar abiertos y preparados para futuras versiones.

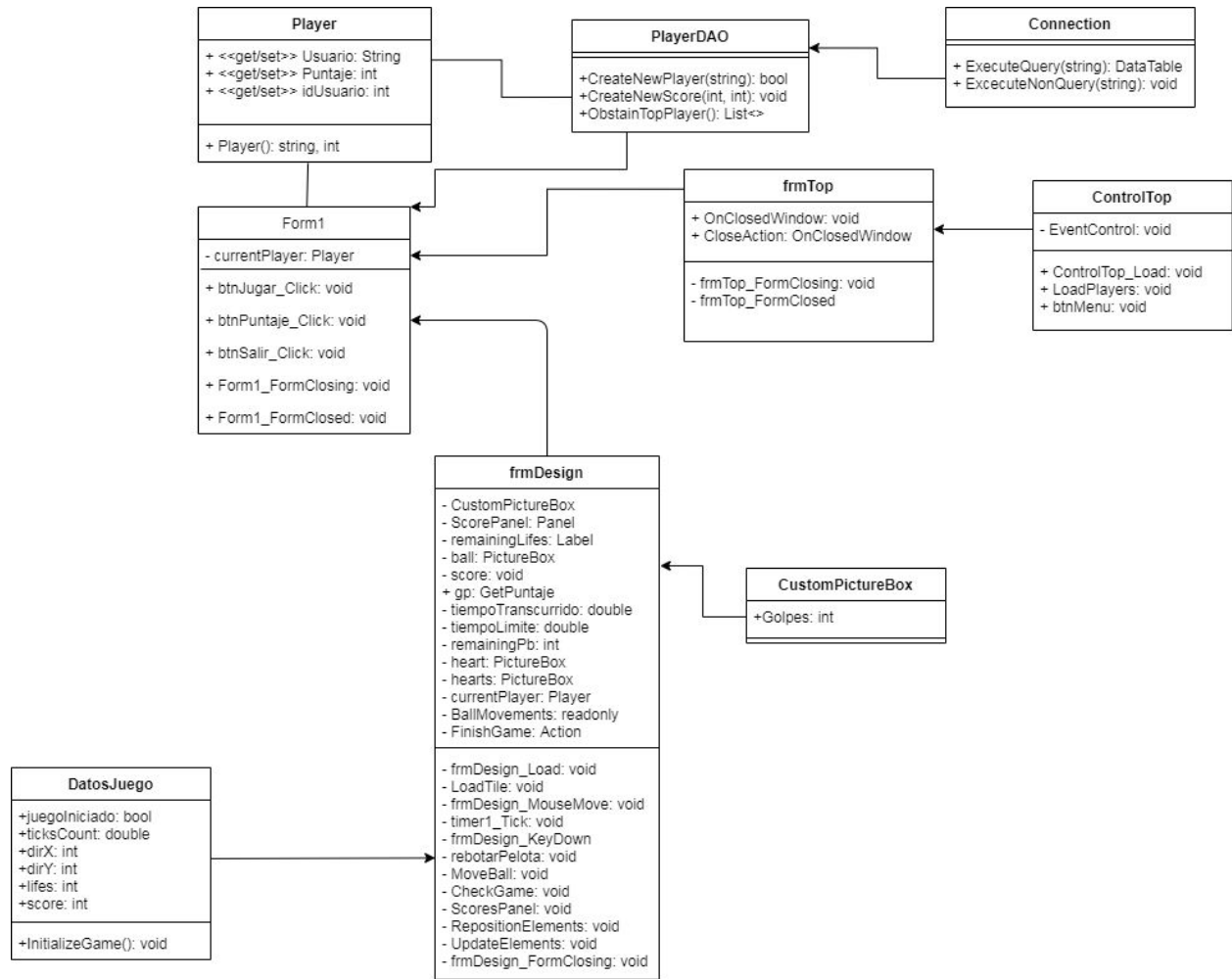
## Descripción general del proyecto

Para la realización de todo el programa fue necesario basarse en el paradigma orientado a objetos visto en el transcurso de todo el ciclo, además para trabajar controlar el programa fue requerido implementar el modelo vista-controlador(MVC). La principal funcionalidad del programa es implementar un juego capaz de tener las suficientes interacciones con los usuarios.

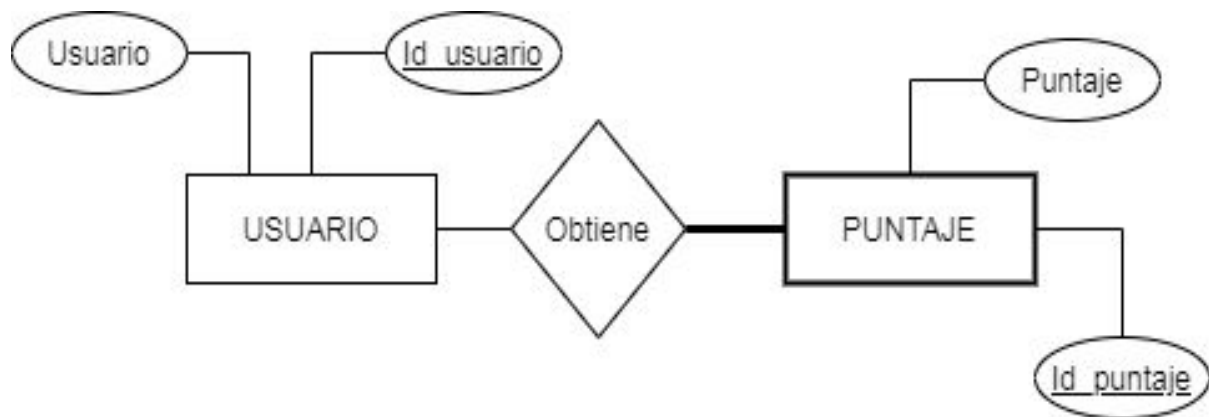
## Software utilizado para la creación del proyecto

Para la realización de este proyecto fue considerado por todos los miembros del equipo usar JetBrains Rider y para la creación de la base de datos pgAdmin 4v4. Complementos necesarios para poder mejorar el rendimientos del programa dentro de JetBrains Rider fue Npgsql.

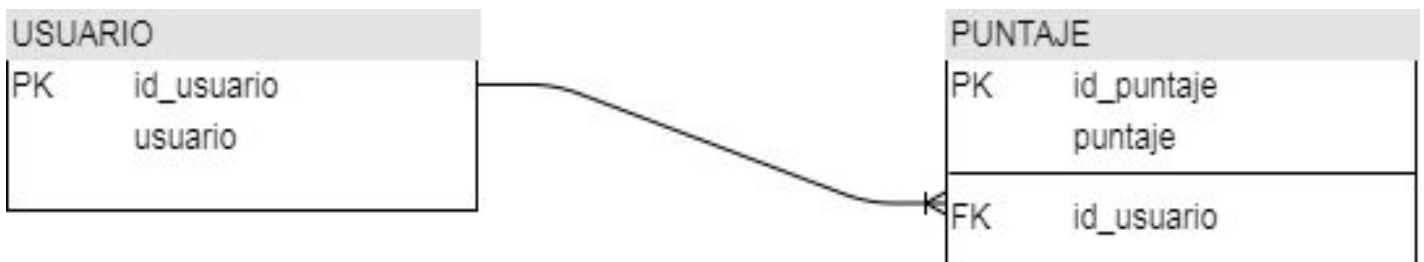
## UML del proyecto



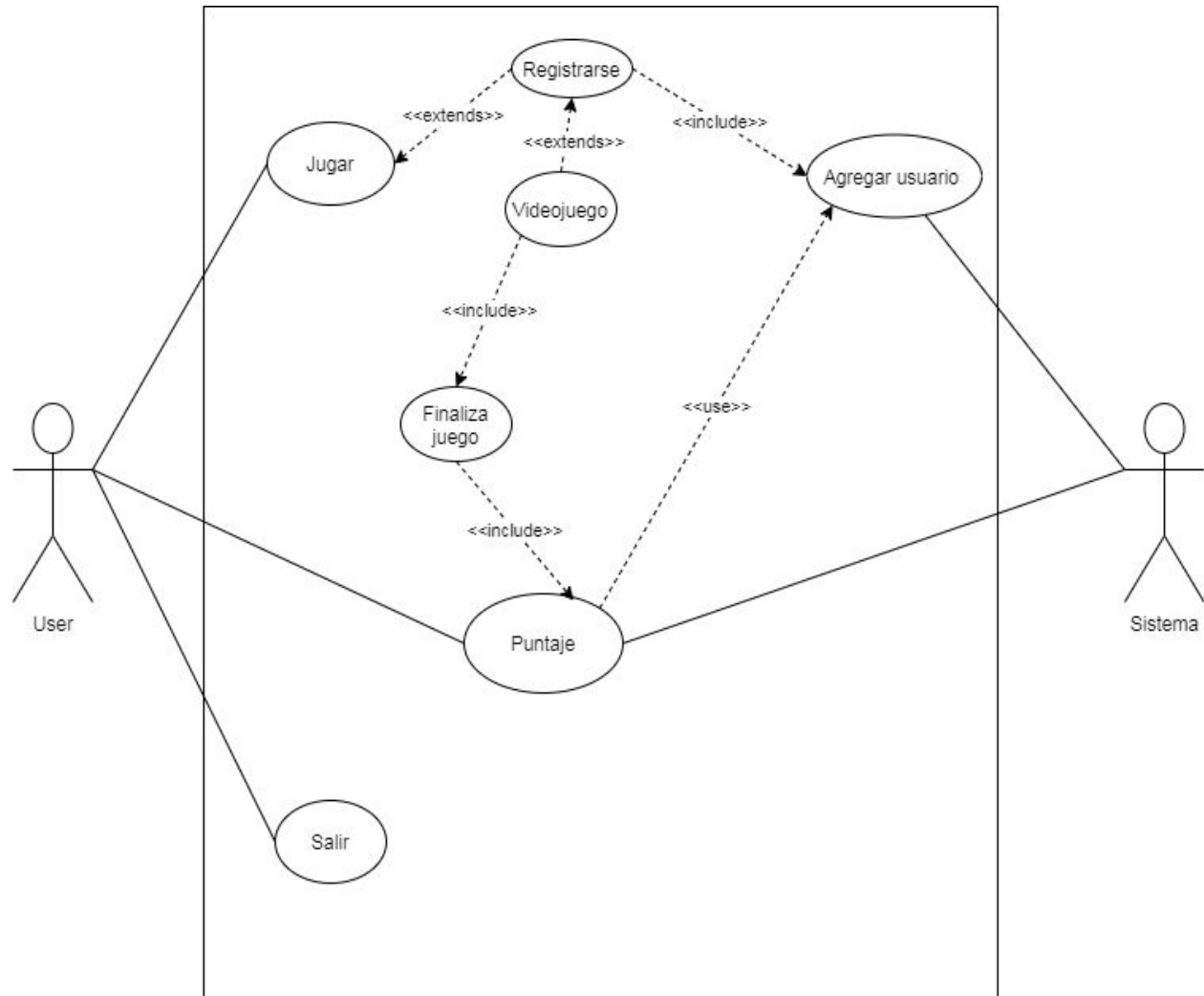
## Diagrama entidad-relación acerca de la base de datos.



## Diagrama relacional.



## UML Caso de usos.



## Conceptos tecnicos y distintos tipos de error.

La interfaz gráfica de nuestro programa consiste en varios formularios conectados y un control de usuario compuestos de varios botones y diferentes herramientas, cada formulario representa una parte diferente del menú del juego. Los formularios y el control de usuario implementados son:

Formularios	Control de usuario
Form1.cs	ControlTop.cs
frmDesign.cs	
frmRegister.cs	
frmTop.cs	

Para el manejo fundamental de todas las opciones del programa se ocuparon la siguiente lista de clases.

Connection.cs // Clase requerida para establecer conexión con base de datos.
CustomPictureBox.cs // Clase para instanciar las funciones del custom picturebox.
DatosJuego.cs // Clase para crear las variables a utilizar en el juego.
Player.cs // Clase para instanciar características del usuario.
PlayerDAO.cs // Clase para tener acceso a las variables del jugador.

## Nomenclaturas.

Para el entorno de el programa tanto para el funcionamiento como para la interfaz.

Frame	frm
-------	-----

Button	btn
Data Access Object	DAO
PictureBox	pic
Timer	tm

## Eventos y lista de excepciones.

Eventos:

ControlTop.cs

- ❑ ControlTop\_Load: Muestra los elementos gráficos para el programa para el control del usuario.
- ❑ button1\_Click: Se encarga de esconder la ventana.

Form1.cs

- ❑ btnJugar: Botón que despliega la ventana para poder registrarte dentro del juego.
- ❑ btnPuntaje: Botón que despliega la ventana del menú a un lado de la ventana del menú.
- ❑ btnSalir: La única finalidad de este botón es la de salir del programa.
- ❑ Form1\_FormClosed: Se encarga de cerrar todos los procesos y terminar todos los procesos abiertos por la aplicación.
- ❑ Form1\_FormClosing: Mensaje de confirmación en caso que el usuario quiera cerrar el programa.

frmDesign.cs

- ❑ frmDesing\_Load: Inicializa y establece la posición de todos los elementos creados en código.
- ❑ frmDesing\_MouseMove: Registra los movimientos del mouse de acuerdo a la voluntad del usuario.
- ❑ timer1\_Click: Evento que realiza las verificaciones si el juego ha comenzado o finalizado.
- ❑ frmDesign\_KeyDown: Registra las pulsacion del teclado, en este trabajo únicamente la de la barra espaciadora.



### frmRegister.cs

- ❑ btnRegister: Botón que verifica si el registro del usuario cumple con todas los requisitos para ser guardado en la base de datos.
- ❑ frmRegister\_FormClosing: Evento que solicita al usuario confirmar si desea cerrar el programa.

Las excepciones debido a sus nombres son autoexplicativas.

NoRemainingLifesException.cs
OutOfBoundsException.cs
WrongKeyException.cs
EmptyNicknameException.cs
ExceededMaxCharacterException.cs