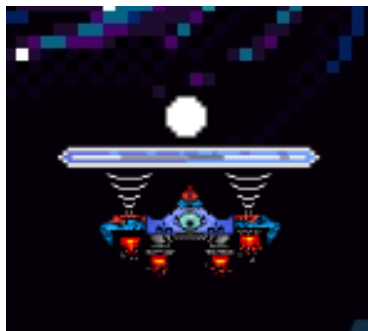


# Manual técnico

## Arkanoïd

### Los Reinicia Routers 2.0



Realizado por:

Evarist Alexander Ardón Arévalo, 00361518

Manuel de Jesús Hernández Benítez, 00094119

Mario Josué Rodríguez Miguel, 00131819

# Contenido

Manual técnico Arkanoid .....	1
Aspectos generales .....	3
-Objetivo del documento	
-Descripción general	
-Software utilizado	
Modelos utilizados .....	4
- UML Diagrama de clases	
-Diagrama entidad relación extendido	
-Diagrama relacional normalizado de base de datos utilizada	
Conceptos técnicos .....	8
-Implementación de interfaz gráfica	
-Manejo de clases en modelo	
-Manejo de clases en controlador	
Nomenclaturas .....	10
-Abreviatura	
Excepciones y eventos .....	11
-Excepciones	
-Eventos	

# Aspectos generales

## *-Objetivo del documento:*

Por medio de este documento, el objetivo es dar a conocer y explicar el software que se ha creado al igual que las herramientas utilizadas.

## *-Descripción general:*

Para poder llevar a cabo la creación de dicho software, se hizo uso del modelo - vista - controlador, sus siglas MVC. La principal funcionalidad del programa es llevar a cabo un juego en el que si el usuario gana se registra su puntuación y si pierde no se registra su puntuación.

## *-Software utilizado:*

Para poder llevar a cabo la creación del programa se utilizó JetBrains Rider 2019, en conjunto con PostgreSQL 12 para poder crear la base de datos. También un complemento adicional para JetBrains Rider ha sido Npgsql para la conexión con la base de datos.

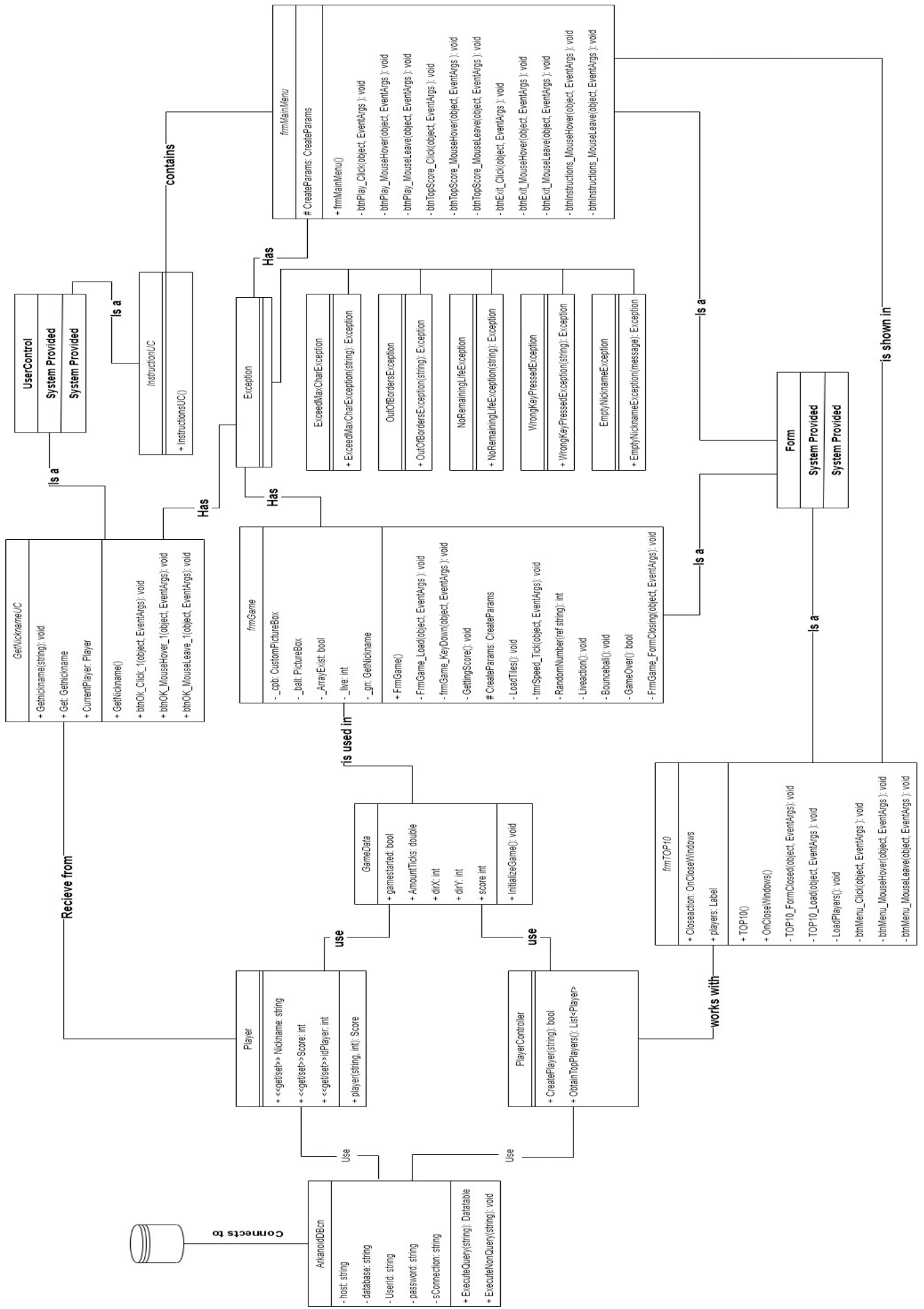
# Modelos utilizados

*-UML Diagrama de clases:*

El diseño del código está basado en el siguiente diagrama de clases siguiente (ver página siguiente):

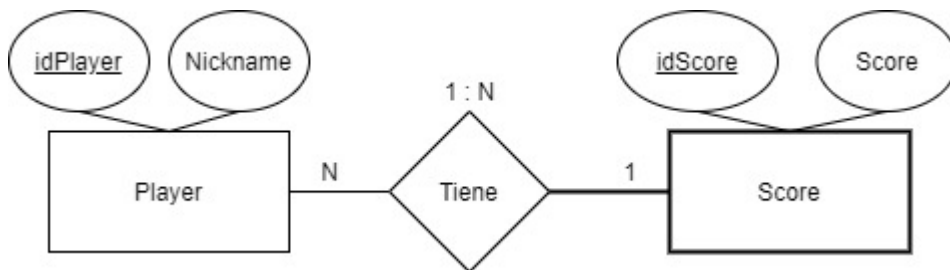
Enlace para una mejor visualización:

<https://drive.google.com/file/d/1-pE8-pSS8QS9vQed2ipEJcdYVzSf8V46/view?usp=sharing>



*-Diagrama entidad relación extendido:*

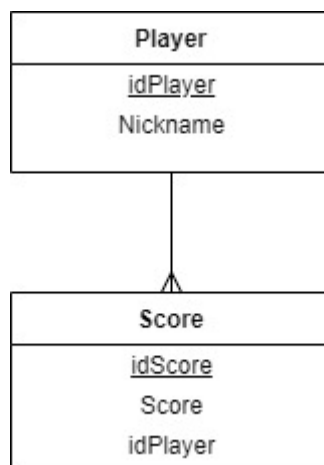
El esquema inicial de la base de datos se desglosa de la siguiente manera:



*-Diagrama relacional normalizado de base de datos*

*Utilizada:*

El esquema modificado de la base de datos se desglosa de la siguiente manera:



# Conceptos técnicos

## *-Implementación de interfaz gráfica:*

La interfaz gráfica del programa se divide en diferentes de la siguiente manera:

Contamos con tres ventanas globales o formularios, los cuales dos se componen por distintos botones los cuales son la ventana del menú en la cual también se carga un control de usuario (User Control en inglés) de instrucciones para el jugador y la ventana del TOP 10 de jugadores.

El restante que es la ventana del juego, cuenta con diferentes pictures box para poder cargar los bloques del juego, las vidas, la pelota y la plataforma del juego y también con un control de usuario al cargarse la ventana del juego, los controles de usuario existentes en nuestro programa son:

InstructionsUserControl.cs

GetNicknameUserControl.cs

### *-Manejo de clases en modelo:*

Para manejar la parte fundamental del modelo del programa, se cuenta con las siguientes clases:

#### CustomPictureBox.cs

Por medio de esta clase podemos llenar en la ventana del juego una matriz con los bloques necesarios para jugar.

#### Player.cs

Por medio de esta clase podemos obtener y manejar los datos del jugador como son su nickname, su id y su puntuación.

### *-Manejo de clases en controlador:*

Para manejar la parte fundamental del controlador del programa, se cuenta con las siguientes clases:

#### ArkanoidDB.cs

Por medio de esta clase podemos hacer la conexión a la base de datos.

#### GameData.cs

Por medio de esta clase controlamos la velocidad de la pelota, si el juego ha empezado o no y la puntuación que va sumando el jugador mientras no pierda y poder mostrarla en pantalla.



PlayerController.cs

Por medio de esta clase podemos hacer las diferentes consultas o acciones a la base de datos.

## Nomenclaturas

*-Abreviaturas:*

Para los elementos utilizados en la parte gráfica se implementaron las siguientes abreviaturas:

Label	lbl
Picture Box	pb
Text Box	txt
Button	btn
DateTimePicker	dtp

# Excepciones y eventos

## *-Excepciones:*

Las excepciones implementadas tienen un nombre autoexplicativo para una mejor comprensión de esta.

Estas constan de un constructor que recibe un string con el mensaje del error. Nuestras excepciones son las siguientes:

`EmptyNickNameException.cs`

Excepción para no dejar el campo donde pide el nickname del jugador vacío.

`ExceedMaxCharException.cs`

Excepción para controlar que el jugador no exceda el máximo de caracteres en su nickname puesto en la base de datos.

`NoRemainingLivesException.cs`

Excepción para controlar cuando ya no queden vidas disponibles en el juego.

`OutOfBordersException.cs`

Excepción para controlar cuando la bola se pueda llegar a salir de los bordes establecidos para jugar.

`WrongKeyPressedException.cs`

Excepción que detecta si el jugador está apretando otra tecla que no sea barra espaciadora para comenzar el juego.

*-Eventos:*

ClickEventArgs.cs

Detecta cuando le demos click a algún botón y pase algo.

MouseHoverEventArgs.cs

Evento que hace que un botón cambie de color o muestre algo al poner el mouse encima de dicho botón.

MouseLeaveEventArgs.cs

Evento para que cuando se quite el mouse de encima de un botón vuelva a estar como estaba.

KeyEventArgs.cs

Evento para detectar que se presiona una tecla.

LoadEventArgs.cs

Evento para que cuando cargue la ventana pase algo.

TickEventArgs.cs

Evento para saber los ticks y poder sacar la puntuación el juego.

FormClosingEventArgs.cs

Evento para que cuando le demos a la x (cerrar ventana) vuelva al menú.

FormClosedEventArgs.cs

Evento para cambiar entre ventanas cuando le demos a la x (cerrar ventana).