

Manual técnico Arkandoid



REALIZADO POR:

Andrea Alejandra Ayala Deras 00109319

Daniela Irene Huevo López 00167518


Fernando Miguel Rodríguez González 00211919





CONTENIDO:

Aspectos generales.....	4
Objetivo del documento	4
Descripción general.....	4
Software utilizado	4
Modelos utilizados.....	5
UML Diagrama de clases.....	5
Diagrama relacional normalizado de base de datos utilizada	8
Conceptos Tecnicos.....	9
Implementación de interfaz gráfica.....	9
Distintos tipos de errores	10
Manejo de clases en modelo.....	10
ConnectionDB.cs	10
CustomPictureBox.cs	10
GameData.cs	10
User.cs	10
Plataforma base.....	11
Abreviaturas	11
Excepciones.....	12
Excepciones.....	12





ASPECTOS GENERALES

Objetivo del documento

El presente documento tiene el fin de informar sobre el software creado, así como las herramientas utilizadas para el desarrollo del mismo.

Descripción general

Para la creación del software se ha utilizado el modelo – vista – controlador y el paradigma orientado a objetos. El programa consiste en un juego arcade que tiene como título Arkanoid que fue desarrollado aplicando los conocimientos obtenidos a lo largo de la materia.

Software utilizado

El desarrollo del proyecto se hizo en Visual Studio Community 2019, junto con PostgreSQL 12 para la creación de la base de datos. Para lograr la conexión del programa con la base de datos se hizo uso del complemento Npqsql.



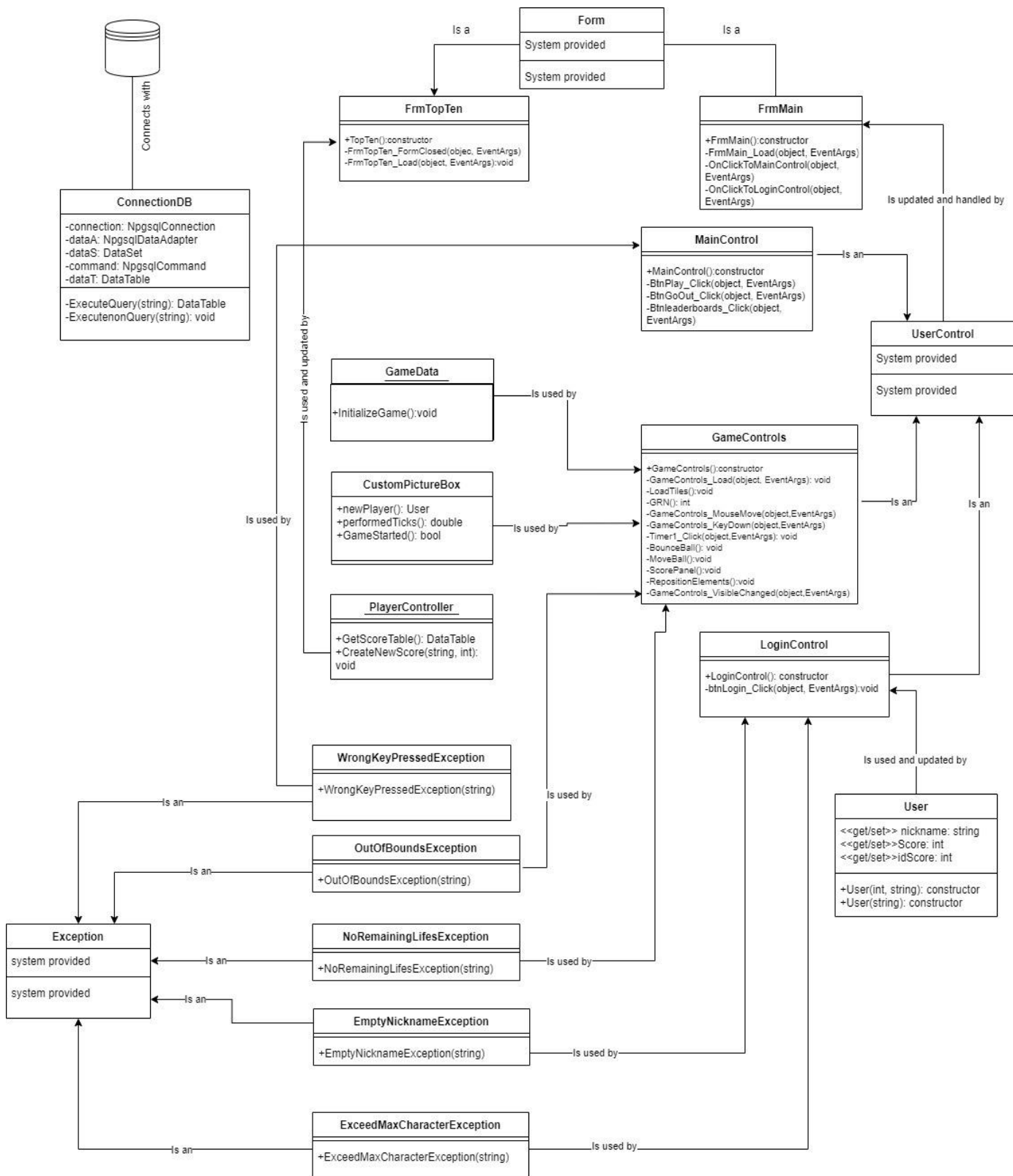


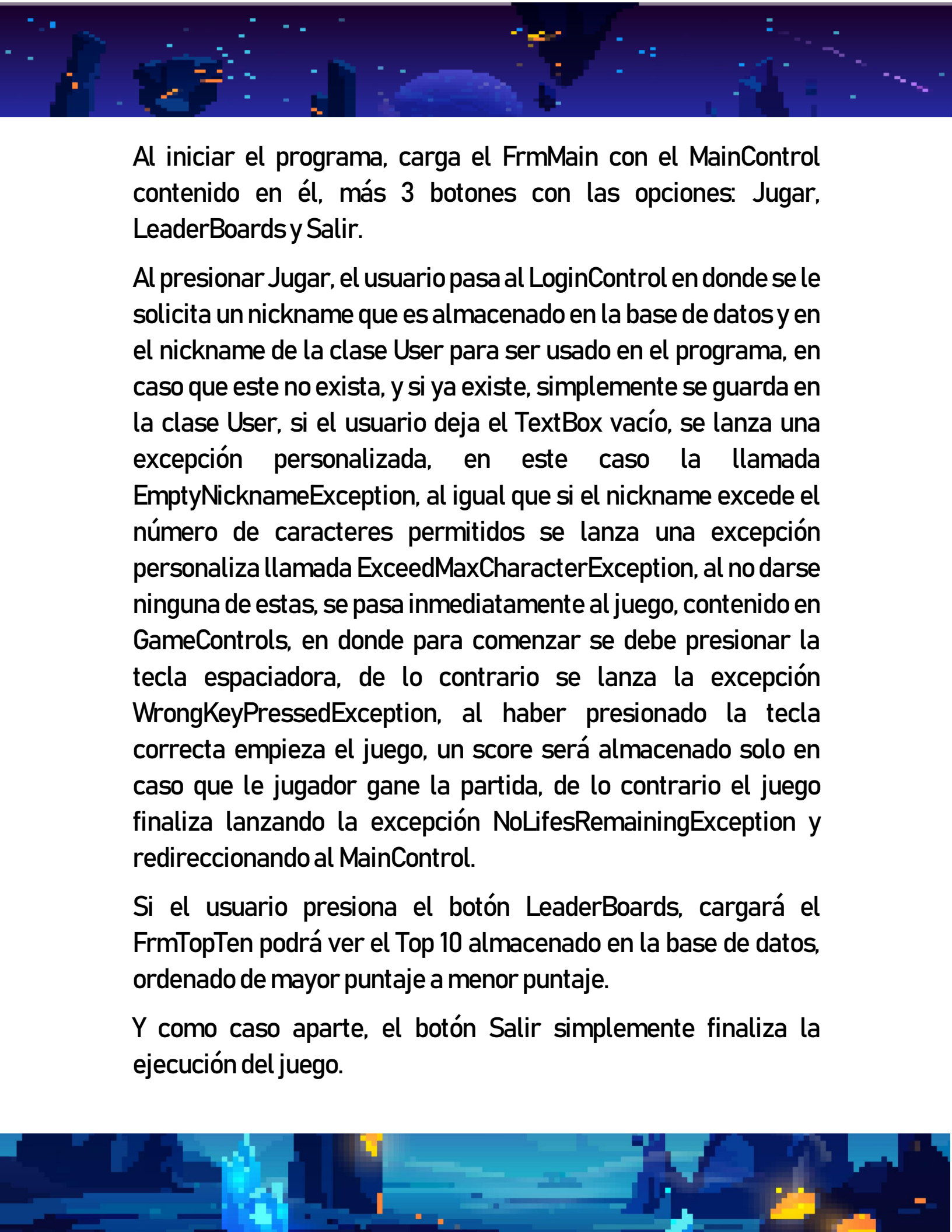
MODELOS UTILIZADOS

UML Diagrama de clases

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







Al iniciar el programa, carga el FrmMain con el MainControl contenido en él, más 3 botones con las opciones: Jugar, LeaderBoards y Salir.

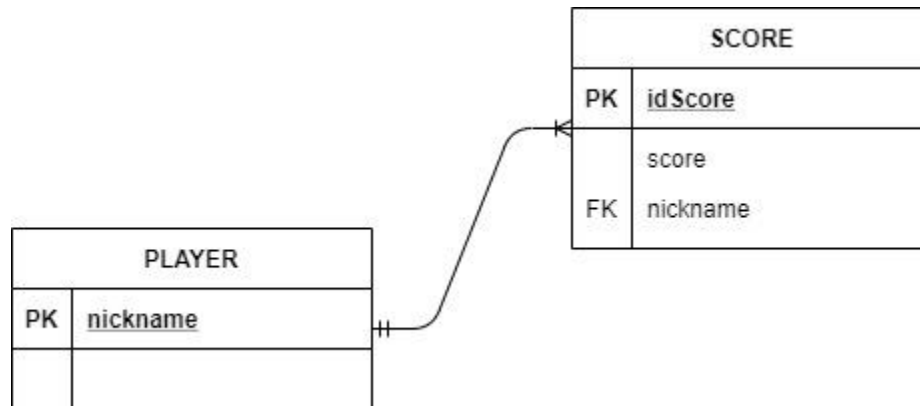
Al presionar Jugar, el usuario pasa al LoginControl en donde se le solicita un nickname que es almacenado en la base de datos y en el nickname de la clase User para ser usado en el programa, en caso que este no exista, y si ya existe, simplemente se guarda en la clase User, si el usuario deja el TextBox vacío, se lanza una excepción personalizada, en este caso la llamada EmptyNicknameException, al igual que si el nickname excede el número de caracteres permitidos se lanza una excepción personalizada llamada ExceedMaxCharacterException, al no darse ninguna de estas, se pasa inmediatamente al juego, contenido en GameControls, en donde para comenzar se debe presionar la tecla espaciadora, de lo contrario se lanza la excepción WrongKeyPressedException, al haber presionado la tecla correcta empieza el juego, un score será almacenado solo en caso que le jugador gane la partida, de lo contrario el juego finaliza lanzando la excepción NoLivesRemainingException y redireccionando al MainControl.

Si el usuario presiona el botón LeaderBoards, cargará el FrmTopTen podrá ver el Top 10 almacenado en la base de datos, ordenado de mayor puntaje a menor puntaje.

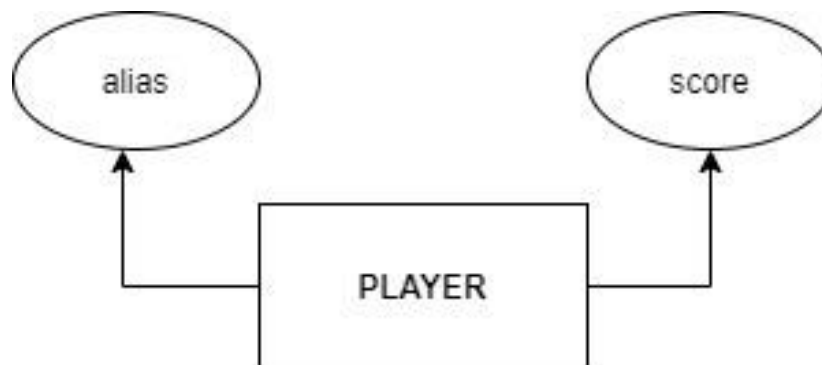
Y como caso aparte, el botón Salir simplemente finaliza la ejecución del juego.



Diagrama relacional normalizado de base de datos utilizada



En el diagrama podemos observar la relación que existe entre Player y Score, donde cada jugador puede tener mas de un Score asociado a él. Al igual que para explicar y entender mejor la relación se realizó un diagrama de relación extendida





CONCEPTOS TECNICOS

Implementación de interfaz gráfica

La interfaz gráfica del programa consiste en dos formularios o ventanas globales, la ventana principal, en donde cargan los distintos controles de usuario o mejor conocido por su término en inglés UserControl y una ventana secundaria para el Top 10, que se ha creado en un Form.

La manera en la que se ha trabajado la interfaz gráfica es la siguiente:

FrmMain.cs tiene el trabajo de cargar a los demás user control y el FrmTopTen.

FrmMain.cs contiene el método FrmMain_Load donde se le setean las propiedades de los User Control y se hace la conexión con MainControl.cs, GameControls.cs y LoginControl.cs

FrmTopTen.cs es una ventana que muestra al usuario el top 10 de los mejores puntajes dentro de toda la base de datos del juego.

LoginControl.cs se encarga de registrar al usuario en caso de que este no exista.

GamesControls.cs se encarga de todo el funcionamiento del juego, cargan los bloques, cargar la barra del jugador, las físicas de la pelota, el manejo de las vidas y registra los puntos obtenidos en la base de datos.

MainControl.cs Menú principal del programa, el cual se encarga de mostrar al usuario las diferentes opciones disponibles por medio de botones (jugar, ver top diez, salir). Para lograr la “comunicación” de este control con los demás se hizo uso de delegates esto se hizo para mejorar el flujo del programa.





Distintos tipos de errores

La experiencia del usuario puede verse afectada por los errores del programa, a continuación, se enlistan los posibles errores que un usuario puede experimentar:

- Dejar el espacio de usuario sin ningún carácter.
- Nombre de usuario tenga más de quince caracteres.
- Fallos ocasionados con la conexión a la base de datos.

Cabe mencionar que se han implementado excepciones personalizadas para manejar los distintos errores, las cuales se enlistan en posteriormente en el documento.

Manejo de clases en modelo

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

ConnectionDB.cs: Clase encargada de conectar con la base de datos y hacer las consultas necesarias a lo largo del programa.

CustomPictureBox.cs: Clase hija de PictureBox que es la que se utiliza para crear los obstáculos del juego permitiendo que algunos de estos tengan más de una vida.

GameData.cs: Clase estática que provee los datos necesarios para el juego: puntajes, vidas, juego iniciado, velocidades de la pelota.

User.cs: Esta clase almacena los valores del usuario (nickname, score), para luego insertarlos en la base de datos.





Plataforma base

Sistema operativo	Windows 10
Tecnologías	Visual Studio Community 2019
Lenguaje	C# 8.0
Gestor de DB	PostgreSQL 12


NOMENCLATURAS

Abreviaturas

Para el nombramiento de los controles de la interfaz gráfica se utilizó la siguiente normativa de nombramiento: <Abreviatura de tipo>_nombre/descripción

Las abreviaturas de presentan a continuación:

Label	lbl
TextBox	txt
PictureBox	pic
Button	btn
Data grid view	dgv
Panel	pnl
Form	frm
UserControl	Control





EXCEPCIONES

Excepciones

Las excepciones utilizadas tienen un nombre autoexplicativo. Cada una cuenta con un constructor que recibe un string el cual contiene el mensaje a mostrar al usuario. Se tienen las siguientes:

`OutOfBoundsException.cs`

`EmptyNicknameException.cs`

`NoRemainingLivesException.cs`

`WrongKeyPressedException.cs`

`ExceededMaxCharacterException.cs`

