



Manual técnico Arkanoïd



Realizado por:

Mario Ernesto Mayén Castro, 00220618

Carlos Misael Pérez Pérez, 00202118





CONTENIDO

Manual técnico Arkanoid	1
Aspectos generales	3
Objetivo del documento	3
Descripción general.....	3
Software utilizado	3
Modelos utilizados	4
UML Diagrama de clases	4
Diagrama relacional normalizado de base de datos utilizada...	7
Conceptos técnicos	9
Implementación de interfaz gráfica	9
Manejo de clases en modelo	10
Plataforma base	10
Nomenclaturas	12
Abreviaturas	12
Eventos y Excepciones.....	13
Eventos.....	13
Excepciones	14





Aspectos generales

Objetivo del documento

El objetivo de este documento pretende orientar y explicar el diseño del software creado, explicando las herramientas utilizadas y a utilizar para futuras versiones.

Descripción general

Para la creación del software se hizo uso del modelo – vista – controlador, sus siglas MVC. El programa está basado un juego interactivo el cual hace referencia al famoso juego Arkanoid de la consola SNES.

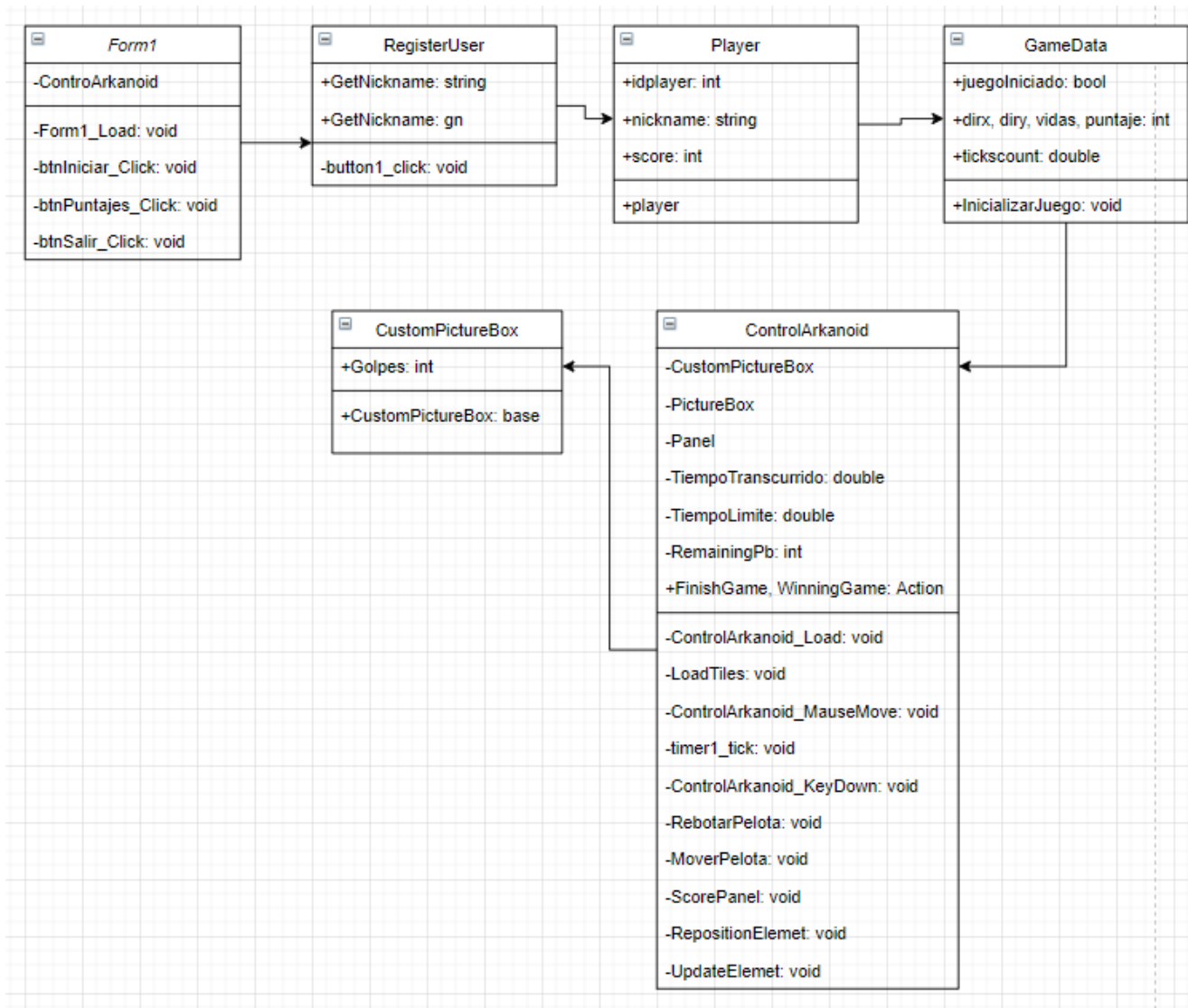
Software utilizado

Para la creación del programa se utilizó JetBrains Rider 2019 2019, en conjunto con PostgreSQL 12 (PgAdmin 4) para la creación de la base de datos. Complementos adicionales para JetBrains Rider utilizados han sido Npgsql para la conexión hacia la base de datos.

Modelos utilizados

UML Diagrama de clases

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





Explicando el diagrama de clases (UML), se tiene un usuario por la ejecución de todo el programa.

Form1 cuenta con el evento de cargar la pantalla de inicio del juego, así como los eventos click de los botones que redirigen hacia la opción de botón seleccionada.

El registerUser se encarga, como su nombre lo indica, de registrar los nuevos jugadores que desean entrar al juego haciendo uso de un nickname el cual debe ser único para llevar un registro único del puntaje del jugador en cuestión.

Así mismo esta clase está ligada a la clase player que lo que hace es almacenar tanto un id, que es único, así como el nickname y el puntaje de cada jugador que termina el juego.

La clase GameData implementa la verificación de si el juego esta iniciado o no, así como también es la encargada de los movimientos de la pelota, y también tanto como de las vidas y el puntaje del jugador.

El userControl llamado controlArkanoid se encarga de recibir datos de tipo objeto para poder realizar las acciones de la pelota, los puntajes, las vidas y las acciones de la nave.

A la vez esta se puede decir que esta es la clase madre ya que es la encargada de todos los movimientos del juego, tanto como de la





nave, así como el de los bloques y la pelota.

Esta misma clase tiene de hija la clase llamada CustomPictureBox, que se encarga de cargar ciertas animaciones y eventos del UserControl para el movimiento y animación de los recursos.

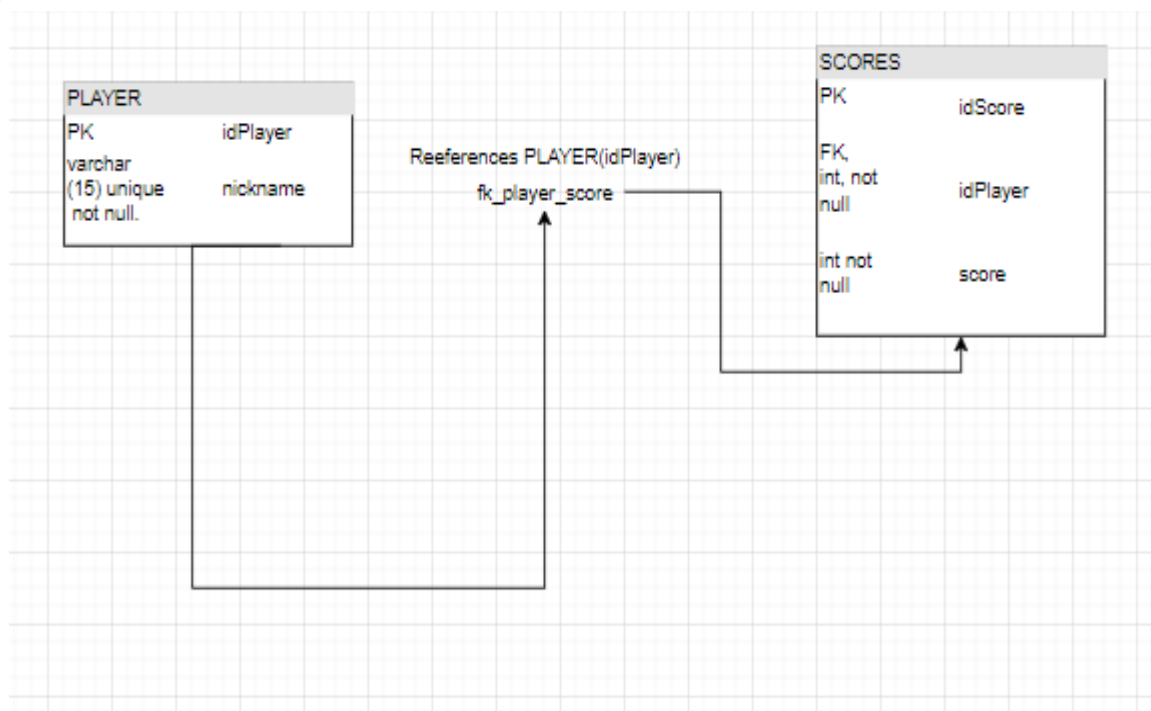


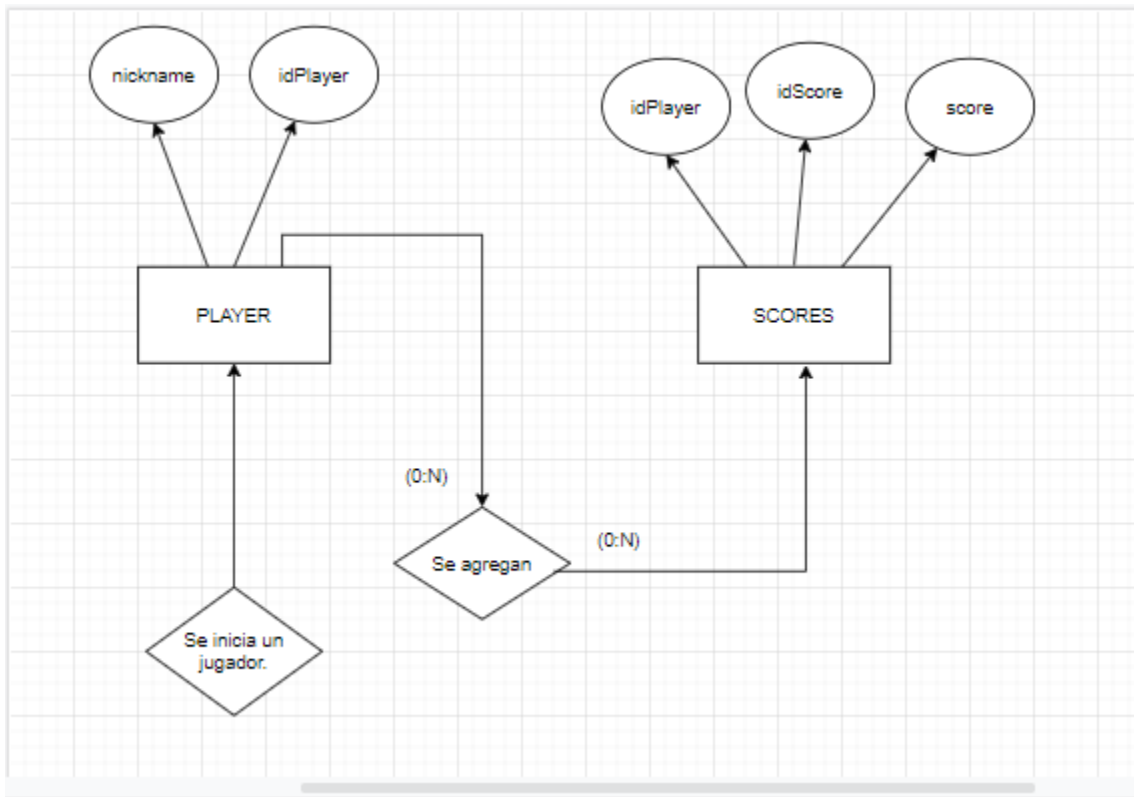
Diagramas de base de datos utilizada

Los diagramas utilizados constan de dos tablas relacionadas entre sí, cuyas tienen el nombre de PLAYER y SCORES, las cuales almacenan los resultados enviados por el juego cuando se crea un jugador al iniciar.

El jugador es guardado a partir de un nickname único, luego al finalizar se le asigna el puntaje correspondiente a la sesión del juego que se finalizó.

Se crearon dos diagramas, Diagrama Entidad Relación Extendido y el Diagrama Relacional, a continuación se presentan ambos diagramas:





En los siguientes diagramas se puede observar la relación entre las tablas.

Esta base de datos se comunica con nuestra Form TOP en el proyecto, en el cual se especifica por medio de una sentencia, que la base de datos retorne los 10 valores más altos almacenados, luego son mostrados en el Form en un listado de descendente de los jugadores almacenados con su nickname y el puntaje al correspondiente al lado.



Conceptos técnicos

Implementación de interfaz gráfica

La interfaz gráfica del programa consiste únicamente de dos ventanas (Form1.cs, Top.cs), compuesta de distintos botones que se encargan de iniciar el juego, mostrar el top o salir del juego.

El **control de usuario (User Control en inglés)**, el control de usuario representa una opción propia del código que debe ser implementada, control de usuario existente:

-ControlArkanoid.cs



Manejo de clases en modelo

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

-DataBaseController.cs

-CustomPictureBox.cs

-DatosJuego.cs

-Form1.cs

-Player.cs

-PlayerController.cs

-Program.cs

-RegisterUser.cs

-Top.cs

-ControlArkanoid.cs



Plataforma base

*Sistema operativo	Windows
*Tecnologías	JetBrains Rider 2019
*Lenguaje	C#
*Gestor de DB	PostgreSQL 12(PgAdmin 4)





Nomenclaturas

Abreviaturas

Para los elementos del entorno gráfico se implementa la siguiente normativa de nombramiento:

Label	lbl
Panel	panel
TextBox	Textbox
Picture Box	PictureBox
Button	btn



Eventos y Excepciones

Eventos

Para poder pasar una acción reflejada en un control de usuario al formulario principal fue necesario la implementación de eventos. Los cuatro eventos implementados fueron

- `Form1_Load`
Se encarga de iniciar el Panell y el fondo.
- `btnIniciar_Click`
Se inicia el juego y se controla el evento `WinningGame` y `FinishGame`.
- `btnPuntajes_Click`
Registra un click y se encarga de mostrar el Top de los puntajes almacenados en el juego.
- `btnSalir_Click`
Sale del juego y termina la sesión.



Excepciones

Las excepciones, debido a sus nombres, son autoexplicativas. Constan de un constructor que recibe un string con el mensaje de error. Se cuentan con las siguientes:

`EmptyNicknameException.cs`

`ExceededMaxCharactersException.cs`

`NoRemainingLivesException.cs`

`OutOfBoundsException.cs`

`WrongKeyPressedException.cs`

