

Manual técnico “Arkanoïd Clásico”



Realizado por:

- **Diego Andrés Portillo Gonzales, 00210719**
- **Diego Eduardo Abrego Cornejo, 00044919**
- **Gerardo Stanley Flores Roque, 00204919**

CONTENIDO

Manual técnico Arkanoid Clásico.....	1
Aspectos Generales.....	3
Objetivos del documento.....	3
Descripción General.....	3
Software utilizado.....	3
Modelos utilizados.....	4
UML Diagrama de clases.....	4
Diagrama relacional normalizado de bases de datos.....	6
Conceptos técnicos	7
Implementación de interfaz gráfica.....	7
Manejo de clases en modelo.....	7
Plataforma Base.....	7
Nomenclaturas.....	8
Abreviaturas.....	8
Eventos y Excepciones.....	9
Eventos.....	9
Excepciones.....	11

Aspectos generales

Objetivos del documento:

El objetivo de este documento pretende la orientación y explicación del diseño de software realizado, y a su vez realizar una explicación de las herramientas utilizadas en el desarrollo de dicho software.

Descripción general:

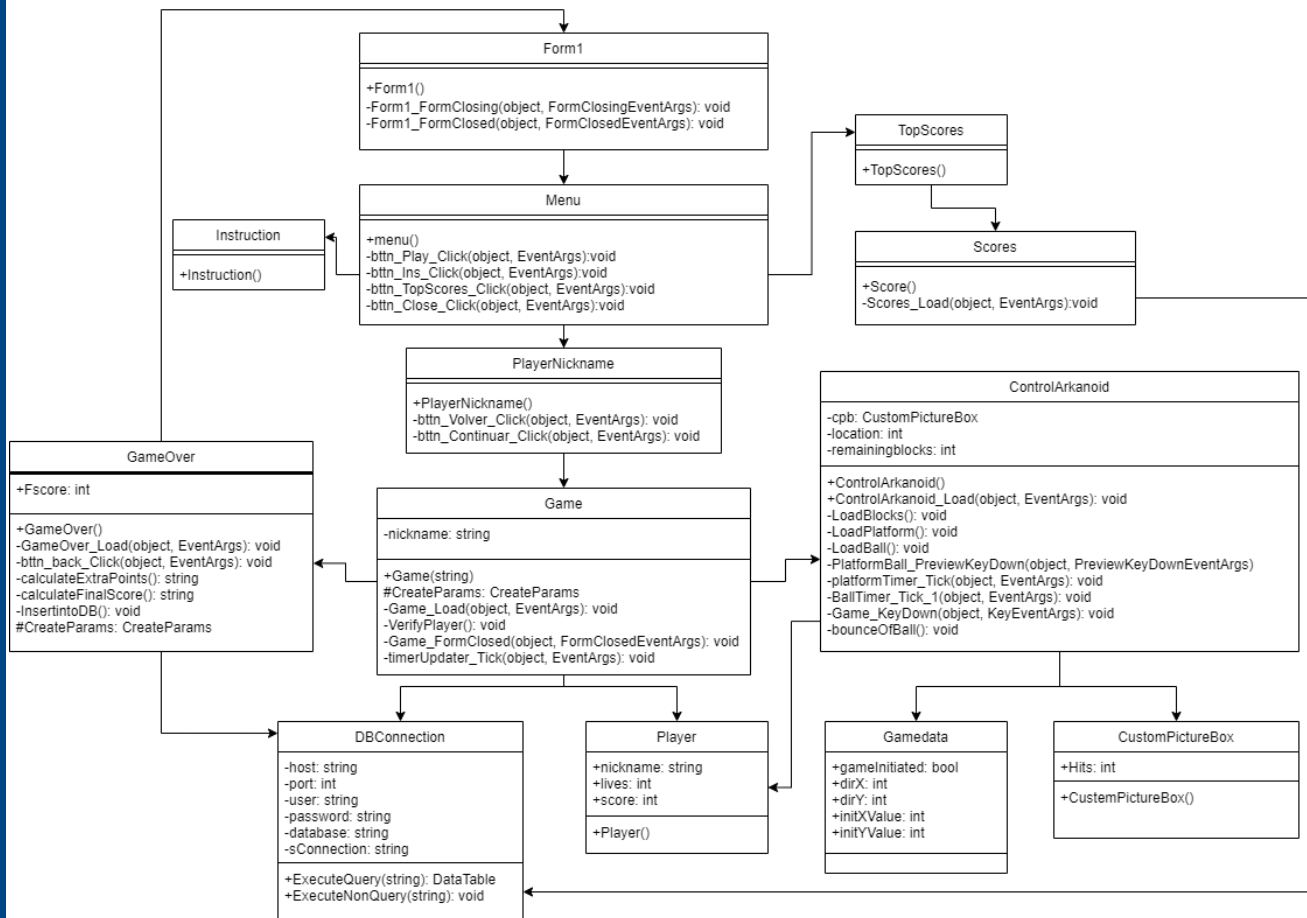
Para la creación del software se hizo uso del modelo - vista – controlador, cuyas siglas son MVC. El programa presente como uso principal la recreación del videojuego clásico “Arkanoïd” para el entretenimiento del usuario.

Software utilizado:

Para la creación del programa se utilizó JetBrains Rider 2020.1.3, en conjunto de PostgreSQL para crear la base de datos. Algunos complementos utilizados en Rider y Visual han sido utilizadas la herramienta adicional de Npgsql para la conexión.

Modelos utilizados

UML Diagrama de clases



Redacción:

Explicando el diagrama anteriormente mostrado: el programa inicia con un Form1 el cual despliega el user control llamado “menú”. Este user control posee botones que acceden a otros formularios, además de un botón para cerrar el programa. Estos botones nos dejan acceder al formulario de instrucciones y el de top scores; el cual accede a un user control llamado scores, que se conecta a la base de datos para obtener los diez mejores puntajes registrados en la base de datos.

El user control “menú” puede desplegar otro formulario llamado “PlayerNickname” el cual se utiliza para obtener el nombre del jugador.

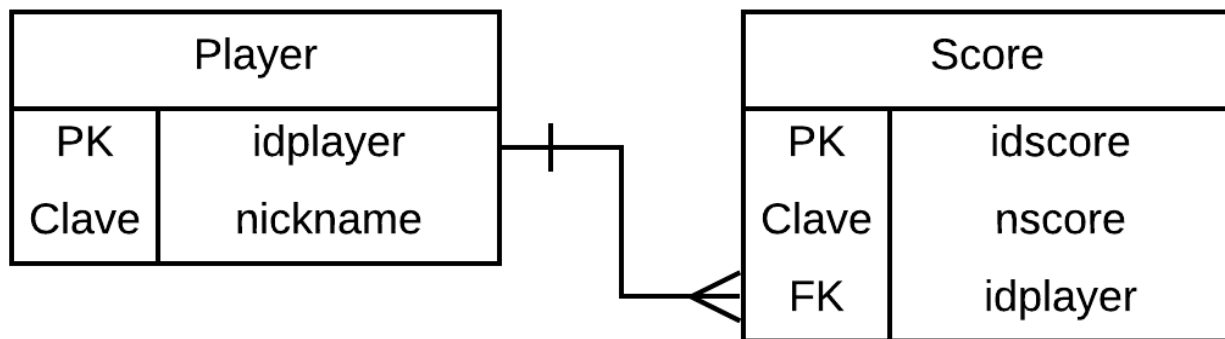
Consecuentemente este formulario conlleva a otro principal llamado “Game” en el cual se mostrará el juego en cuestión. Cuando este formulario carga, recibe un parámetro llamado “nickname”, parámetro que se compara mediante la clase estática “DBConnection” verificando si existe o no el base de datos. Si se da el caso de no existir, el formulario lo agrega mediante la clase estática ya mencionada; aunque en este formulario es donde se puede ver el juego, no está creado en este allí.

Para la creación del juego se utiliza el user control llamado “ControlArkanoid”, el cual se conecta a las clases estáticas llamadas “Gamedata” y “CustomPictureBox”. Gamedata se utiliza para los datos importantes del juego tales como si el juego esta iniciado o no, las direcciones en pixeles, además de sus valores iniciales. Por otro lado, CustomPictureBox se utiliza para crear los bloques del juego. Esta clase hereda de la clase “PictureBox”; Además, existe otra clase estática llamada “Player”, la cual es accedida por Game y ControlArkanoid.

La clase Player tiene la utilidad de mantener todos los datos importantes con respecto al jugador, tales como su nombre, sus vidas restantes y el puntaje que este posee. En el user control ControlArkanoid, existen los parámetros para que el juego termine, y que el formulario Game se cierre. Al cerrarse, este lanza uno nuevo llamado “GameOver”, quien se encarga de mostrar los datos de la clase Player, mostrando el nickname, sus vidas restantes y cuál es su puntaje inicial; adicionalmente este formulario se encarga de dar puntos extras que dependen de las vidas que le quedaran al jugador. Consecuentemente, también es el encargado de mandar los datos a la base de datos mediante una conexión a la clase DBConnection. Este formulario, además, posee un botón que nos sirve para regresar al menú principal, creando así un nuevo Form1, desde el cual se puede volver a acceder a las demás funcionalidades del programa.

Diagrama relacional normalizado de bases de datos utilizadas:

Complementariamente el programa utiliza una base de datos para guardar los usuarios y los puntajes de estos, esta base de datos posee un diagrama, mostrado a continuación:



Conceptos técnicos

Implementación de interfaz gráfica:

La interfaz gráfica del programa consiste de una ventana formulario, compuesta por distintos botones encargados de llevarnos a los distintos controles de usuario y formularios encargados de desplegar toda la información de nuestro proyecto. Los controles de usuarios utilizados en nuestro programa fueron tres los cuales son:

- ControlArkanoid.cs
- Menu.cs
- Scores.cs

Manejo de clases en modelo:

Para manejar una de las partes principales del modelo de nuestro programa, fue necesaria la utilización de las siguientes clases:

- CustomPictureBox.cs
- DBConnection.cs
- Gamedata.cs
- Player.cs

Plataforma Base:

Sistema operativo	Multiplataforma
Tecnologías	Jetbrains Rider 2020.1.3
Lenguaje	C#
Gestor de DB	PostgreSQL

Nomenclaturas

Abreviaturas:

Para los elementos del entorno grafico de nuestro proyecto se implementa la siguiente normativa de nombramiento:

Tipo ----- id-Correlativo

Label	lbl
Text Box	txt
Picture Box	pic
Button	bttm

Eventos y Excepciones

Eventos:

Durante el desarrollo del programa fueron necesarios la utilización de los siguientes eventos reflejados en los controles de usuario, dichos eventos implementados son:

- ControlArkanoid_Load

Al desplegar la pantalla y cargarla se despliegan todos los objetos dentro de ella como la plataforma, los bloques, la pelota, las vidas entre otras.

- btn_Play_Click

Registra un clic en el control de usuario menú, y envía el formulario para ingresar el apodo del usuario que ocupara en el juego.

- btn_Ins_Click

Registra un clic en el control de usuario menú, y envía un formulario para mostrar las instrucciones con los controles y algunas reglas del juego.

- btn_TopScores_Click

Registra un clic en el control de usuario menú, y envía un formulario que muestra las diez mejores puntuajes almacenados en la base de datos.

- btn_Close_Click

Registra un clic en el control de usuario menú, ejecutando un método el cual cierra la aplicación.

- platformBall_PreviewKeyDown

Registra la activación de las teclas flecha izquierda y derecha activadas desde teclado para poder mover la plataforma en el juego.

- BallTimer_Tick_1

Comienza el tiempo para el cual la pelota se mueve una vez se detecte que se a presionado la barra espaciadora para iniciar el juego.

- Game_KeyDown

Detecta la presión de la barra espaciadora devolviendo un booleano en verdadero para iniciar el movimiento de la pelota.

- BounceOfBall

Detecta cuando la pelota llega a los límites de la pantalla haciendo rebotar la pelota en las paredes y bloques.

- Scores_Load

Al desplegarse este control de usuario y terminar de cargar se accede a la base de datos desplegando los 10 mejores puntajes almacenados en dicha base de datos por medio de una lista y varios label.

Excepciones:

Debido a sus nombres las excepciones utilizadas son auto explicativas. Constan de un constructor que recibe un string con el mensaje de error. Se cuentan con las siguientes excepciones:

- `invalidcharacterfornickname.cs`
- `invalidnullnickname.cs`
- `dberrorexception.cs`