2020

Manual Técnico

ARKANOID

Realizado por:
Rodrigo David Cortez Martinez, 00045719
Carlos Rodrigo López López, 0008317
Jaime Samuel Portan Rivas, 00046119
Daniel Alfredo Quesada Cortez, 00147619

Contenido

Aspectos Generales

Objetivo del Documento

El propósito de este documento es explicar el funcionamiento del programa "ARKANOID", detallando las herramientas utilizadas para crear este programa; además de mostrar el Diagrama de clases UML utilizado para este programa, también detallaremos los Diagramas de nuestra base de datos implementada para este juego. De igual manera mostraremos y explicaremos algunos conceptos técnicos de nuestro programa, así como las nomenclaturas utilizadas y eventos y excepciones.

Descripción General

Para la creación del software se hizo el uso de un diagrama de clases UML. La función principal del programa es permitir que el usuario pueda jugar una recreación del juego "ARKANOID" de 1986.

Software Utilizado

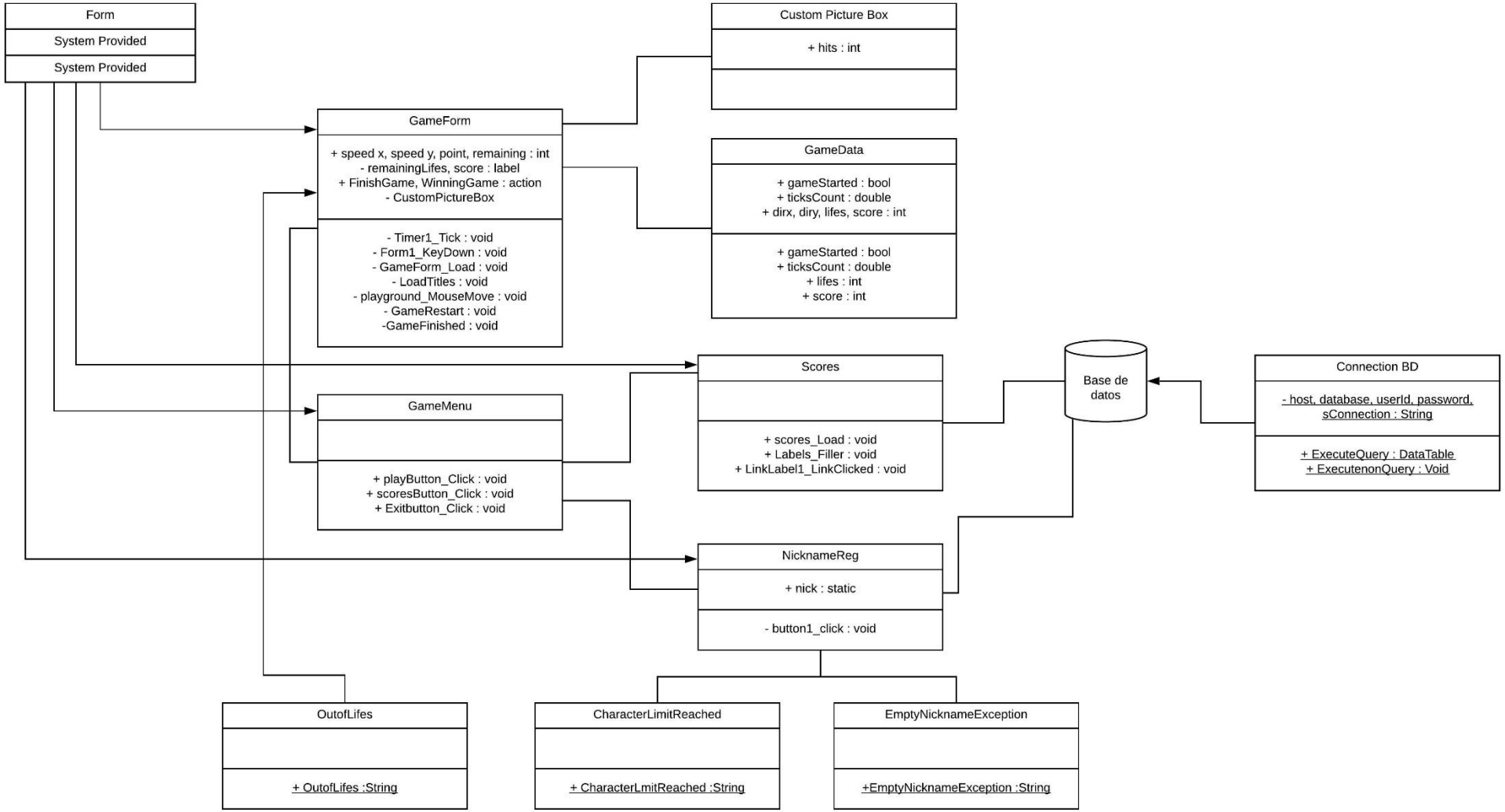
Para crear el programa se hizo uso del software JetBrains Rider 2020.1.3, en conjunto de PostgreSQL 12 para la creación de la base de datos. También se hizo uso del complemento Npgsql en Jetbrains Rider para realizar la conexión hacia la base de datos.

Modelos Utilizados

UML Diagrama de Clases

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

Se anexa un hipervínculo para una vista más detallada del diagrama:
https://drive.google.com/file/d/1gJkM1C5fOlA_jma-Xr3rUpk8mUmbMv6o/view?usp=sharing


Form
System Provided
System Provided
Custom Picture Box
+ hits : int
GameForm
+ speed x, speed y, point, remaining : int
- remainingLifes, score : label
+ FinishGame, WinningGame : action
- CustomPictureBox
- Timer1_Tick : void
- Form1_KeyDown : void
- GameForm_Load : void
- LoadTitles : void
- playground_MouseMove : void
- GameRestart : void
-GameFinished : void
GameData
+ gameStarted : bool
+ ticksCount : double
+ dirx, diry, lifes, score : int
+ gameStarted : bool
+ ticksCount : double
+ lifes : int
+ score : int
Scores
+ scores_Load : void
+ Labels_Filler : void
+ LinkLabel1_LinkClicked : void
Base de datos
Connection BD
- host, database, userId, password, sConnection : String
+ ExecuteQuery : DataTable
+ ExecutenonQuery : Void
GameMenu
+ playButton_Click : void
+ scoresButton_Click : void
+ Exitbutton_Click : void
NicknameReg
+ nick : static
- button1_click : void
OutofLifes
+ OutofLifes :String
CharacterLimitReached
+ CharacterLmitReached :String
EmptyNicknameException
+EmptyNicknameException :String

Explicando el diagrama de clases, se tiene un usuario por la ejecución del programa.
GameMenu es el Form principal donde se inicia el juego, este posee dos botones, el primero permite avanzar a otro Form el cual es NicknameReg que sirve para agregar un jugador y posteriormente iniciar el juego; el segundo conduce a un Form el cual es Scores y este permite visualizar el Top 10 de los mejores jugadores del juego; cabe mencionar que estos Forms (NicknameReg y Scores) están enlazados a nuestra de base de datos para que los datos se agreguen en las tablas correspondientes.

Como se mencionaba anteriormente el juego inicia cuando el jugador ingresa un nuevo usuario, esto nos lleva al Form que tiene como nombre GameForm en este se incluye toda la funcionalidad del juego, y GameData y CustomPictureBox son hijos de GameForm; GameData nos permite llevar iniciar el juego, llevar el conteo de vidas, el conteo de puntajes, y CustomPictureBox nos permitió realizar el golpe de la pelota a los bloques.

Diagrama Entidad Relación Extendido

Diagrama Relacional Normalizado

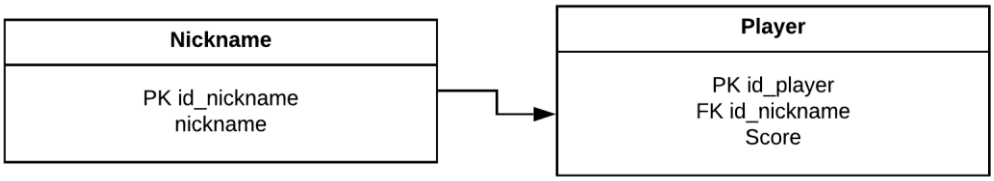
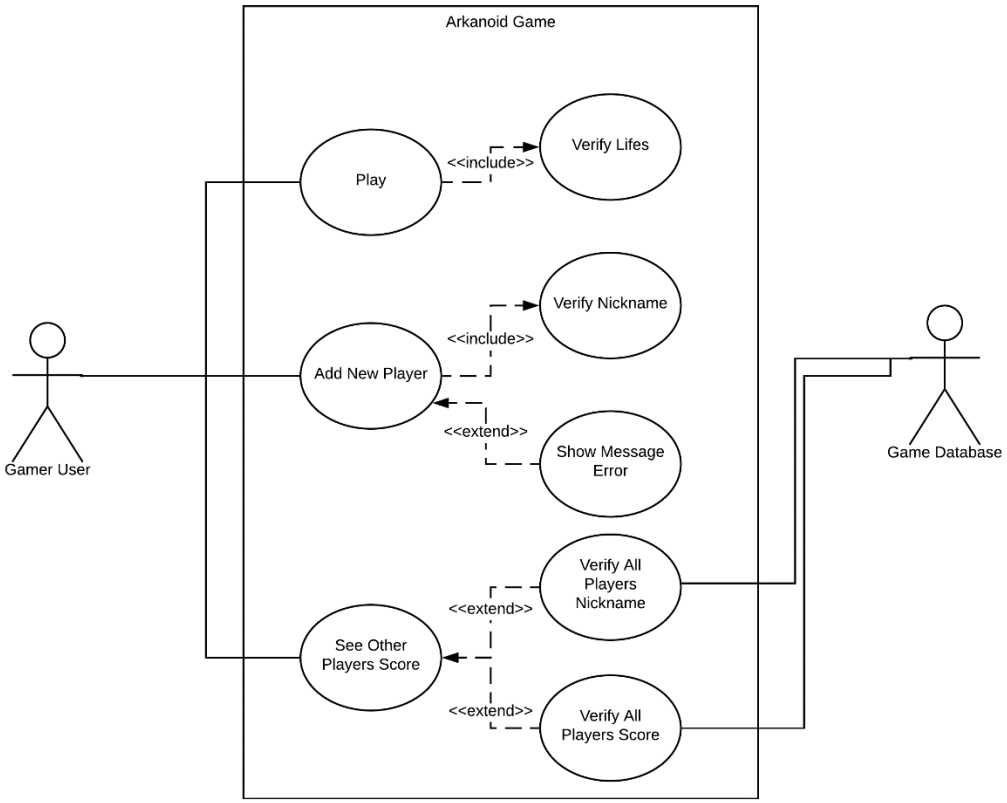


Diagrama de Casos de Uso UML



Conceptos Técnicos

La interfaz gráfica de nuestro programa está compuesta por una ventana formulario la cual posee el nombre de GameMenu, compuesta de distintos botones y varios Forms que se encargan de cargar una opción propia del programa, la lista de los Forms utilizados son:

- GameMenu
- Scores
- NicknameReg
- GameForm

Manejo de Clases

Para manejar el programa de una manera más eficaz se hizo la implementación de clases; las cuales son las siguientes:

- CustomPictureBox.cs
- GameData.cs

Plataforma Base

Sistema operativo	Multiplataforma
Tecnologías	JetBrains Rider 2020.1.3
Lenguaje	C#
Gestor de BD	PostgreSQL 12

Nomenclaturas

Abreviaturas

Las abreviaturas utilizadas fueron las siguientes:

Label	lbl
TextBox	txt
PictureBox	pic
Button	bttn

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:

- CharacterLimitReached.cs
- EmptyNicknameException.cs
- OutofLifes.cs