

**UNIVERSIDAD CENTROAMERICANA JOSE SIMEON CAÑAS UCA**

**Manual Técnico Proyecto Programación Orientada Objetos**

**Realizado por:**

**Rigoberto Campos Vega, 00081511**

**Contenido.**

1. **Aspectos generales**
2. **UML**
3. **Diagrama Entidad Relación Extendido y Diagrama Relacional**
4. **Conceptos técnicos y distintos tipos de error**
5. **Nomenclaturas.**
6. **Eventos y excepciones. Lista de eventos y excepciones con breve descripción**
7. **Aspectos Generales**

**Objetivo del documento.**

Esté documento tiene como objetivo mostrar el proceso de creación del juego 2d arkanaid, desde el proceso de análisis y diseño hasta el desarrollo final del mismo.

**Descripción general.**

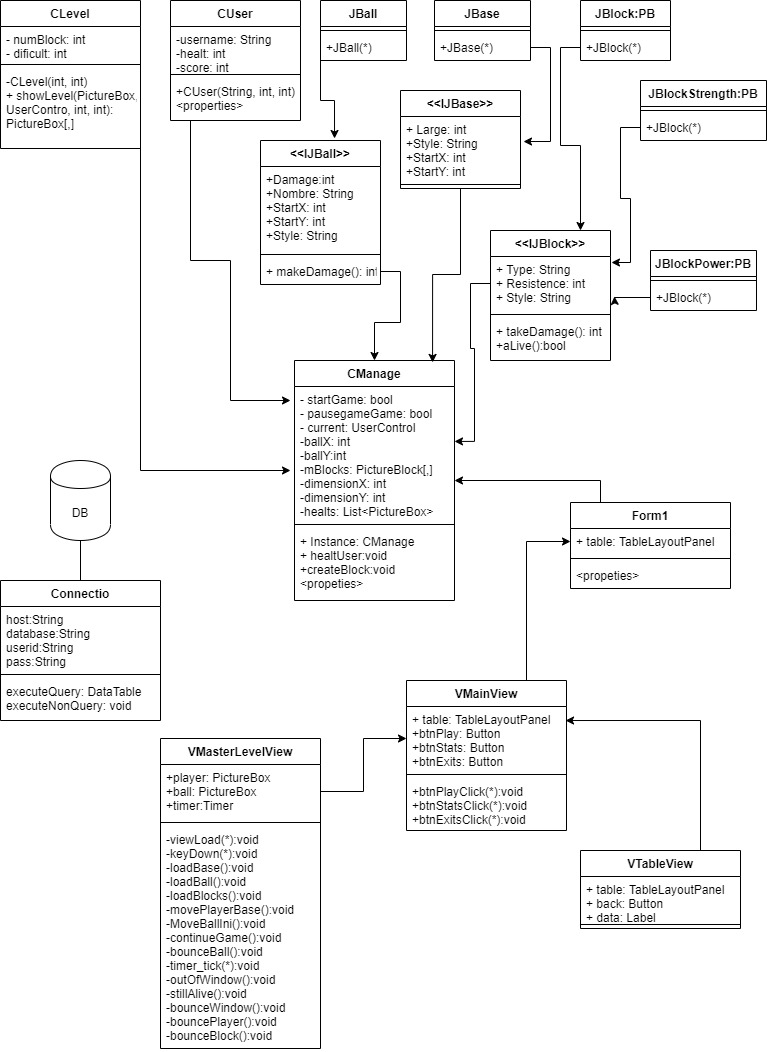
Para el desarrollo del proyecto se utilizaron las características de POO, se hizo uso de características como Polimorfismo, creación de clases, Herencia, interfaces, entre otros.

**Softwares utilizados.**

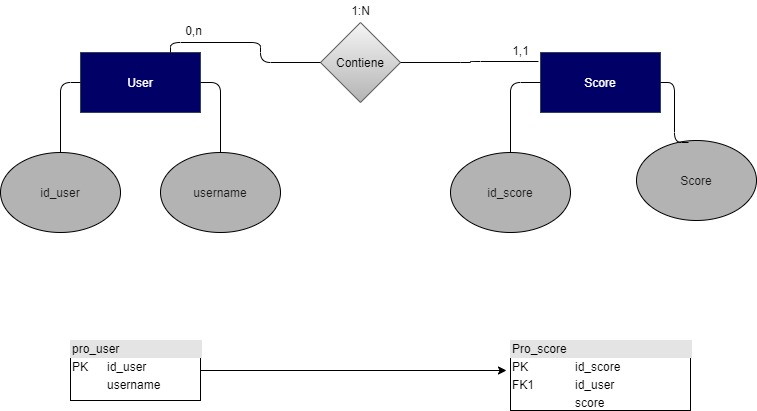
El análisis y diseño se elaboró en un tablero de Trello con una descripción general de los pasos a seguir, detallando algunas características principales del desarrollo.

El desarrollo del juego se hizo únicamente con C# y su librería gráfica (las imágenes se descargaron de internet).

1. **UML**



1. **Diagrama Entidad Relación Extendido y Diagrama Relacional**



1. **Conceptos técnicos y distintos tipos de error**

* **Interfaces.**

**IJBlock:** Representa el esqueleto de los bloques, contiene las propiedades necesarias para la creación de estos. Sus métodos son takeDamage y aLive. El primero hace referencia al manejo que hará el bloque del daño recibido y el segundo simplemente verifica si el bloque aun puede resistir.

**IJBall:** Esqueleto de las pelotas de la aplicación, contiene las propiedades necesarias para la creación de pelotas. Su método Hacer daño es la representación de cuanto daño hará la pelota en un impacto.

**IJBase:** Esqueleto de la base. Contiene las propiedades necesarias para la creación de todas las bases.

* **Clases.**

**CLevel:** Clase que se utiliza para crear los niveles del juego.

**CUser:** Clase que se encarga de la creación de usuarios, contiene el puntaje y el nombre de usuario.

**JBlock:** Clase que se encarga de la creación de bloques que resisten 1 golpe solamente.

**JBlocPower:** Clase que crea bloques mas poderosos capaces de resistir el daño recibido menos 2.

**JBlockStrenth:** Clase que genera los bloques mas resistentes del juego, capaces de resistir el daño recibido entre 2.

**JBall:** Clase que crea la pelota con menos daño del juego, realiza 4 de daño por impacto.

**JBallPower:** Clase que crea la pelota con mas daño del juego, realiza 8 de daño.

**JBase:** Clase que crea la base mas pequeña del juego.

**JBasePower:** Clase que crea la base mas larga del juego.

**CManage:** Clase que se encarga de la lógica del juego, conecta las pantallas y contiene atributos de las interfaces de la aplicación. Singlento.

**Constant:** Clase encargada de las constantes de la aplicación.

1. **Eventos**

Cada botón cuenta con un evento al ser pulsados, que cambian entre los UserControls de la aplicación.

La pantalla del juego contiene eventos de presionado del teclado, activado con las letra “a” y “d”.