
Universidad de los Andes

Facultad de Ingeniería

Mérida - Venezuela

# Ronda

(Desarrollo de software: Videojuego “de mesa” por turnos).

Semestre: A-2018

Prof.: Solazver Solé

**Integrantes:**

Juan Romay V-23.715.879

César Barrios V-21.226.140

**Introducción**

**Descripción del Producto**

Es un videojuego (software) llamado Ronda, para dos jugadores, por turnos. La temática del juego se centra en “aplastar células”. Un jugador tiene chance de aplastar de 1 a 3 células por turno

Las células, que son de color verde al ser aplastadas realizan las siguientes acciones:

.- Puede desaparecer al tocarla

.- Puede dividirse en otras dos células verdes

.- Cambiar a color Rojo. Cuando solo quedan células rojas acaba el juego

El jugador, como estrategia, puede pasar su turno. Al final el jugador que retire la última célula verde gana.

**Requisitos funcionales**

**.-** Diseñar un videojuego de tipo “mesa” sencillo, multijugador, por turnos, usando la plataforma Unity.

**Requisitos no funcionales**

.- Hacer uso de la implementación de la Ingeniería de Software y Base de Datos

.- Aprovechar la plataforma Unity para el diseño de la estructura del videojuego y que su desempeño sea óptimo

.- Mostrar el modelo de actores que describe los procesos que hacen los mismos

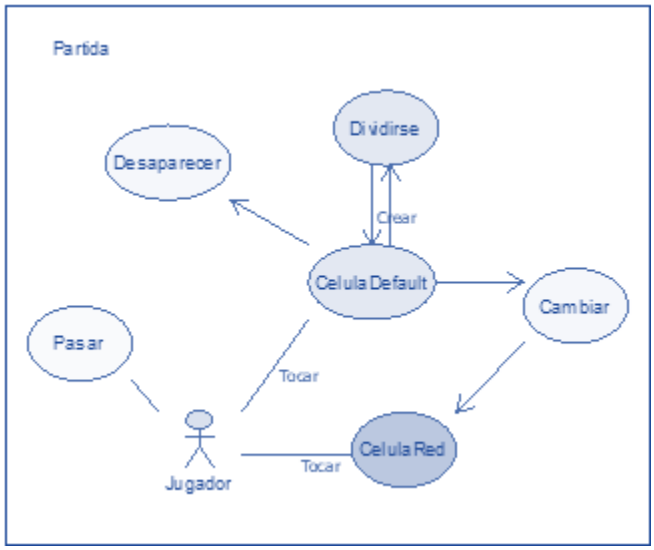
.- Diseñar la interfaz de usuario del videojuego y los gráficos de los componentes que aparecen al ejecutarlo para que sean amigables a la vista del usuario

*.- Implementar una base de datos y conectarla a Unity*

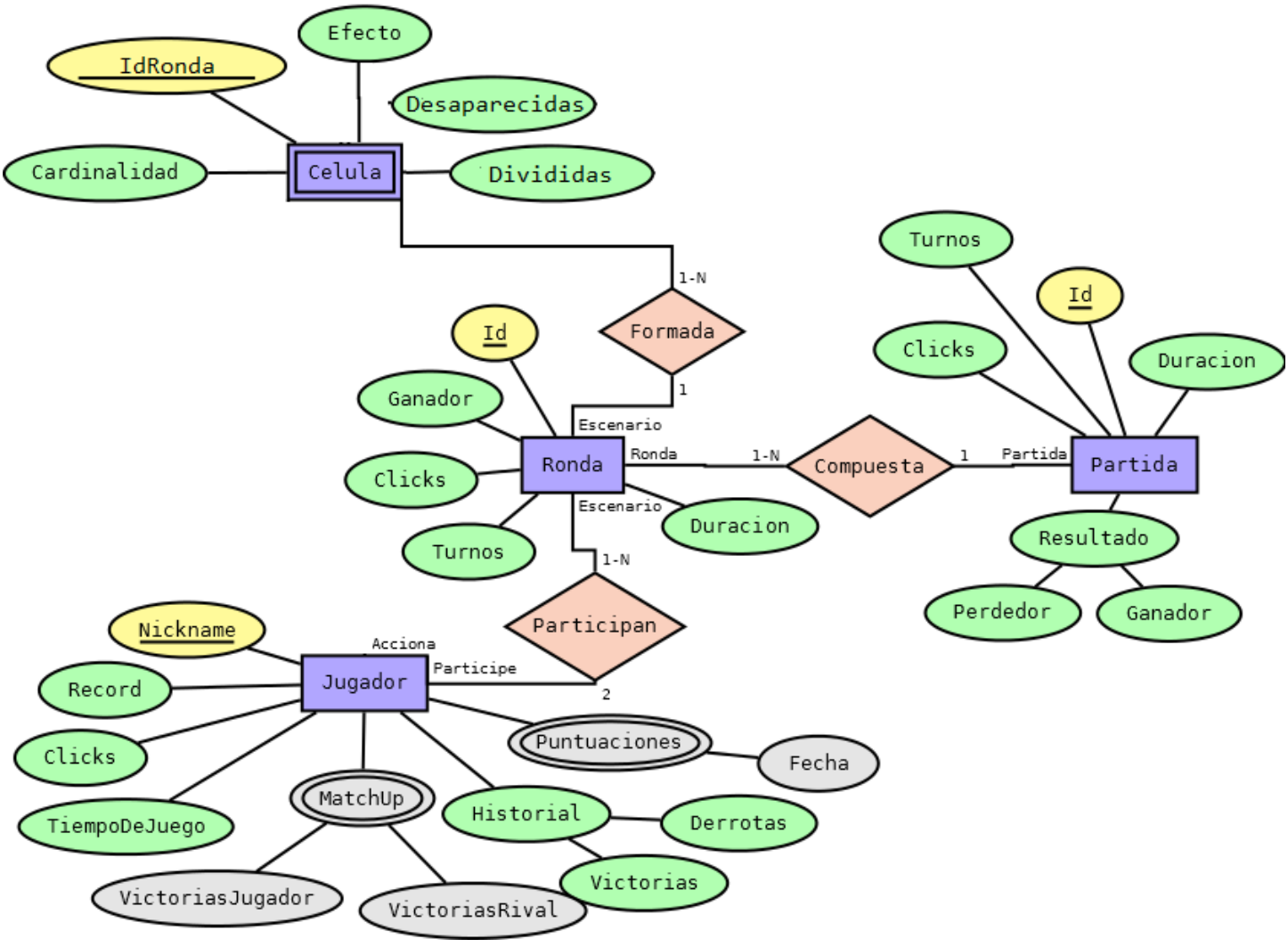
**Diagrama de Despliegue**

Este diagrama, representa donde irán alojados físicamente cada uno de los componentes del sistema. Dado que la arquitectura que presenta el videojuego es Stand-alone, el único componente que se representa o utiliza es el dispositivo del cliente (WindowsPC o Android). Se debe tener presente que el videojuego no requiere que ningún sistema aparte o el uso de una base de datos.

**Funcionamiento general de Ronda**




**Jugador:** El jugador actual que juega el turno. **CelulaDefault:** La célula principal (verde). **CelulaRed:** Célula sin acciones ni cambios (solo puede ser tocada por error). **Cambiar:** Mutación de la célula de verde a roja al hacer clic. **Desaparecer:** Destrucción de la instancia de una célula verde al hacer clic. **Dividirse:** Eliminación de la instancia actual de la célula verde y creación de dos nuevas instancias. **Pasar:** Si aún quedan chances de los 3 disponibles, pero no se quiere tocar la célula se puede pasar.


Efecto
IdRonda
Desaparecidas
Cardinalidad
Celula
Divididas
1-N
Formada
1
Id
Ganador
Escenario
Clicks
Ronda
Ronda
1-N
Compuesta
1
Partida
Partida
Turnos
Id
Clicks
Duracion
Escenario
Duracion
Turnos
1-N
Resultado
Participan
Perdedor
Ganador
Nickname
Acciona
Record
Jugador
Participe
2
Clicks
Puntuaciones
Fecha
MatchUp
Historial
TiempoDeJuego
Derrotas
Victorias
VictoriasJugador
VictoriasRival

**Modelo EER**

**Modelo Relacional**

Celula(Desparecidas, Rojas, Divididas, Cardinalidad, *IdRonda*)

Partida(Id, Duracion, Clicks, Turnos, Perdedor, Ganador, Fecha)

Ronda(Id, Ganador, Clicks, Turnos, Duracion, *idPartida*)

Participan(*IdRonda*, *NicknameJugador*)

Jugador(Nickname, Record, Clicks, TiempoDeJuego, Victorias, Derrotas)

PuntuacionJugador(*NicknameJugador*, Fecha, Puntuacion)

MatchUp(*NicknameJugador*, *Rival*, VictoriasJugador, VictoriasRival)

**Restricciones**

IdPartida en Ronda es clave foranea de Partida(Id)

IdRonda en Participan y Celula es clave foranea de Ronda(Id)

NicknameJugador en Participan es clave foranea de Jugador(Nickname)

NicknameJugador en MatchUp es clave foranea de Jugador(Nickname)

Rival en MatchUp es clave foranea de Jugador(Nickname)

En la tabla MatchUp:

NicknameJugador != Rival

**Significado de palabras claves**

**Gameplay.** Se refiere a todas las experiencias de un jugador durante la interacción con sistemas de juegos. Es un término empleado en el diseño y análisis de juegos que describe la calidad del juego en términos de sus reglas de funcionamiento y de su diseño como juego.

**Célula.** Unidad anatómica fundamental de todos los organismos vivos, generalmente microscópica, formada por citoplasma, uno o más núcleos y una membrana que la rodea.

**Stand-Alone**. Que puede funcionar offline

SQLite

```
CREATE TABLE `Celula` (
	`IdRonda`	INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
	`Desaparecidas`	INTEGER DEFAULT 0,
	`Rojas`	INTEGER DEFAULT 0,
	`Divididas`	INTEGER DEFAULT 0,
	`Cardinalidad`	INTEGER DEFAULT 0,
	FOREIGN KEY(`IdRonda`) REFERENCES `Ronda`(`Id`) ON DELETE CASCADE
);
CREATE TABLE `Jugador` (
	`Nickname`	VARCHAR ( 10 ) ,
	`Record`	INTEGER DEFAULT 0,
	`Clicks`	INTEGER DEFAULT 0,
	`TiempoDeJuego`	TIME ( 6 ) DEFAULT 0,
	`Victorias`	INTEGER DEFAULT 0,
	`Derrotas`	INTEGER DEFAULT 0,
	CHECK(Record>=0),
	CHECK(Victorias>=0),
	CHECK(Derrotas>=0),
	CHECK(Clicks>=0),
	CHECK(Nickname GLOB '*[a-zA-Z]*' AND Length ( Nickname ) >= 3),
	PRIMARY KEY(`Nickname`)
);
```

```
CREATE TABLE `MatchUp` (

	`NicknameJugador`	VARCHAR ( 10 ) NOT NULL CHECK(NicknameJugador GLOB '*[a-zA-Z]*' AND Length ( NicknameJugador ) >= 3),

	`Rival`	VARCHAR ( 10 ) NOT NULL CHECK(NicknameJugador GLOB '*[a-zA-Z]*' AND Length ( NicknameJugador ) >= 3),

	`VictoriasJugador`SMALLINT DEFAULT 0,

	`VictoriasRival`	SMALLINT DEFAULT 0,

	CHECK(NicknameJugador!=Rival),

	FOREIGN KEY(`Rival`) REFERENCES `Jugador`(`Nickname`) ON DELETE CASCADE,

	PRIMARY KEY(`NicknameJugador`,`Rival`),

	FOREIGN KEY(`NicknameJugador`) REFERENCES `Jugador`(`Nickname`) ON DELETE CASCADE

);

CREATE TABLE `Participa` (

	`NicknameJugador`	VARCHAR ( 10 ) NOT NULL CHECK(NicknameJugador GLOB '*[a-zA-Z]*' AND Length ( NicknameJugador ) >= 3),

	`IdPartida`	INTEGER NOT NULL,

	FOREIGN KEY(`IdPartida`) REFERENCES `Partida`(`Id`) ON DELETE CASCADE,

	FOREIGN KEY(`NicknameJugador`) REFERENCES `Jugador`(`Nickname`) ON DELETE CASCADE

);

CREATE TABLE `Partida` (

	`Id`	INTEGER PRIMARY KEY AUTOINCREMENT,

	`Duracion`	TIME ( 6 ) DEFAULT 0,

	`Clicks`	INTEGER DEFAULT 0,
```

```
	`Turnos`	SMALLINT DEFAULT 0,

	`Perdedor`	VARCHAR ( 10 ) NOT NULL CHECK(Ganador GLOB '*[a-zA-Z]*' AND Length ( Ganador ) >= 3),

	`Ganador`	VARCHAR ( 10 ) NOT NULL CHECK(Perdedor GLOB '*[a-zA-Z]*' AND Length ( Perdedor ) >= 3),

	`Fecha`	VARCHAR ( 20 ) DEFAULT 0,

	CHECK(Clicks>=0),

	CHECK(Turnos>=0),

	CHECK(Ganador!=Perdedor)

);

CREATE TABLE `PuntuacionJugador` (

	`NicknameJugador`	VARCHAR ( 10 ) NOT NULL CHECK(NicknameJugador GLOB '*[a-zA-Z]*' AND Length ( NicknameJugador ) >= 3),

	`Fecha`	DATE DEFAULT 0,

	`Puntuacion`	REAL DEFAULT 0,

	FOREIGN KEY(`NicknameJugador`) REFERENCES `Jugador`(`Nickname`) ON DELETE CASCADE,

	CHECK(Puntuacion>=0)

);

CREATE TABLE `Ronda` (

	`Id`	INTEGER PRIMARY KEY AUTOINCREMENT,

	`Duracion`	TIME ( 6 ) DEFAULT 0,

	`Clicks`	INTEGER DEFAULT 0,

	`Turnos`	SMALLINT DEFAULT 0,
```

**`Ganador`  VARCHAR ( 10 ) NOT NULL CHECK(Ganador GLOB '*[a-zA-Z]*' AND Length ( Ganador ) >= 3),**

**`IdPartida`  VARCHAR ( 10 ) DEFAULT 0,**

**FOREIGN KEY(`IdPartida`) REFERENCES `Partida`(`Id`) ON DELETE CASCADE,**

**CHECK(Clicks>=0),**

**CHECK(Turnos>=0)**

**);Historial de Versiones Ronda**

| Versión | Cambios en la versión |
|---|---|
| 1.0 | Estado inicial, Primera versión funcional del juego, solo con la posibilidad de las fichas (células) de estar en la pantalla |
| 1.1 | Experimentando con el borrado de las fichas, se desechó rápido esta versión |
| 1.2 | Agregadas las variables de conteo de las células |
| 1.3 | Interfaz de Usuario lista |
| 1.4 | Puntaje de los jugadores en el Canvas listo. |
| 1.5 | Agrega una escena llamada Menu correspondiente al menú principal del juego con opciones como jugar y salir, también en la escena juego ya cumple con total funcionalidad los tres botones correspondientes al turno y pausa del juego donde se enlazan las dos escenas. |
| 1.6 | Solución de un error en el Script ControlPause que generaba un mal puntaje al salir al menú principal y iniciar nuevamente el juego |

**Historial de Versiones Ronda Post-BaseDeDatos**

| Versión | Cambios en la Versión |
|---|---|
| **1.7** | **Creación del Script DBconnection y la escena de Login de Ronda** |

| | |
|---|---|
| **2.0** | **Conexión de la base de datos SQLite con Unity. Creación del registro de jugadores en la base de datos** |
| **2.1** | **Codificación de la auto inserción y actualización de las variables globales de Ronda dentro de la Base de Datos en el Script UpdateDB** |
| **2.2** | **Detalles de UI. Inserción de Datos de ejemplo.** |
| **2.3** | **Solución de un problema en el que al eliminar la partida empatada, volvía a introducir otra partida con datos Nulos** |
| **2.4** | **Cambios de UI. Introducido un CHECK para que el nombre de los jugadores este restringido a ciertos caracteres y tamaño** |

**Referencias**

.- Jugabilidad. (2017, 2 de diciembre). *Wikipedia, La enciclopedia libre*. Fecha de consulta: 05:02, diciembre 3, 2017 desde https://es.wikipedia.org/w/index.php?title=Jugabilidad&oldid=103877199.

.- Célula. (2017, 29 de septiembre). *Wikipedia, La enciclopedia libre*. Fecha de consulta: 05:02, diciembre 3, 2017 desde https://es.wikipedia.org/w/index.php?title=C%C3%A9lula&oldid=102218144.

.- Temario/Un ejemplo de la creación de un videojuego/Análisis (2017, 27 de julio). *Oficina del Software Libre.* Sacado de http://osl2.uca.es/wikijuegos/w/index.php?title=Temario/Un_ejemplo_de_la_creación_de_un_videojuego/Análisis&oldid=811

.- Veracierta G., Guzmán K., Ortiz M. (2012, 14 de agosto). *Videojuegos: Una alternativa para el aprendizaje de la ciencias en educación básica. Caso: funcionamiento sistema inmunológico.* Espacios. Vol. 34 (1) 2013. Pág. 8. Sacado de http://www.revistaespacios.com/a13v34n01/13340108.html#uno

.- Standalone software. (2017, July 28). In *Wikipedia, The Free Encyclopedia*. Retrieved 05:56, December 8, 2017, from https://en.wikipedia.org/w/index.php?title=Standalone_software&oldid=792726039

**Unity:**

-Technologies, U. (2015). *Unity - Manual: Unity Manual*. [online] Docs.unity3d.com. Available at: http://docs.unity3d.com/Manual/index.html [Accessed 25 June 2018].

-Technologies, U. (2015). *Unity - Scripting API:*. [online] Docs.unity3d.com. Available at: http://docs.unity3d.com/ScriptReference/index.html [Accessed 25 June 2018].