

# ENTREGA FINAL DEL PROYECTO

## PROGRAMACIÓN ORIENTADA A OBJETOS

# Contenido del documento

INTRODUCCIÓN.....	3
REQUERIMIENTOS.....	4
REQUERIMIENTOS FUNCIONALES Y NO FUNCIONALES.....	4
Diagrama de casos de uso.....	5
Casos de uso con flujos alternativos.....	5
Resumen de la evolución de los requerimientos y número total de requerimientos.....	7
DISEÑO .....	9
MAPEO DE REQUERIMIENTOS .....	9
DIAGRAMA DE CLASES .....	13
DIAGRAMA DE PAQUETES .....	13
EVIDENCIA DE LA ARQUITECTURA MVC.....	14
IMPLEMENTACIÓN .....	15
URL DEL REPOSITORIO.....	15
ESTÁNDAR DE CODIFICACIÓN.....	15
DOCUMENTACIÓN DEL CÓDIGO.....	16
REPORTE DE CONTRIBUCIÓN INDIVIDUAL DEL REPOSITORIO .....	16
PROCESO DE DESARROLLO .....	17
DESCRIPCIÓN DE LOS ROLES.....	17
PROCESO DE MONITOREO.....	17
DESCRIPCIÓN DE LAS MÉTRICAS DE CONTRIBUCIÓN INDIVIDUAL.....	18
REPORTE DE CONTRIBUCIÓN INDIVIDUAL .....	19
DESCRIPCIÓN DEL PROCESO PARA LA ADMINISTRACIÓN DEL PROYECTO .....	20
ELEMENTOS ADICIONALES.....	20

## INTRODUCCIÓN

La idea del desarrollo del proyecto surgió por una iniciativa de controlar un juego con el Kinect v1.

La propuesta del proyecto era manejar en un mismo proyecto los movimientos del usuario para controlar el juego mediante una serie de gestos pregerminados bastante genéricos y fáciles de hacer, la finalidad es el mero entretenimiento de los jugadores y la interacción con los videojuegos desde otro punto de vista.

Este documento cuenta con las siguientes especificaciones:

- Requerimientos del sistema
- Diseño
- Implementación
- Proceso de desarrollo
- Elementos adicionales

Integrantes del equipo:

Magdiel Pech Menéndez

# REQUERIMIENTOS

## REQUERIMIENTOS FUNCIONALES Y NO FUNCIONALES

Requerimientos funcionales:

El proyecto será controlado mediante un Kinect.

El proyecto contará con un menú principal.

El proyecto contará con un splash de carga para gestionar la conexión del sensor Kinect.

El proyecto contará con una pantalla de puntajes.

El proyecto contará con una pantalla para guardar su puntaje obtenido en la partida ingresando las iniciales de su nombre.

El proyecto contará con una base de datos para administrar los puntajes de los jugadores.

El proyecto contará con una selección de mano antes de iniciar la partida con la cuál el jugador decidirá con qué mano controlará los movimientos de la serpiente.

Requerimientos no funcionales:

Se redireccionarán las vistas de cada pantalla mediante los botones.

Se mostrará el esqueleto del usuario representado mediante un stickman en la pantalla de juego.

El puntaje será mostrado en la parte superior de la pantalla de juego.

El usuario verá el control del snake mediante una serie de gesticulaciones en la pantalla de juego.

Se mostrará un cursor para identificar la posición de la mano derecha del usuario.

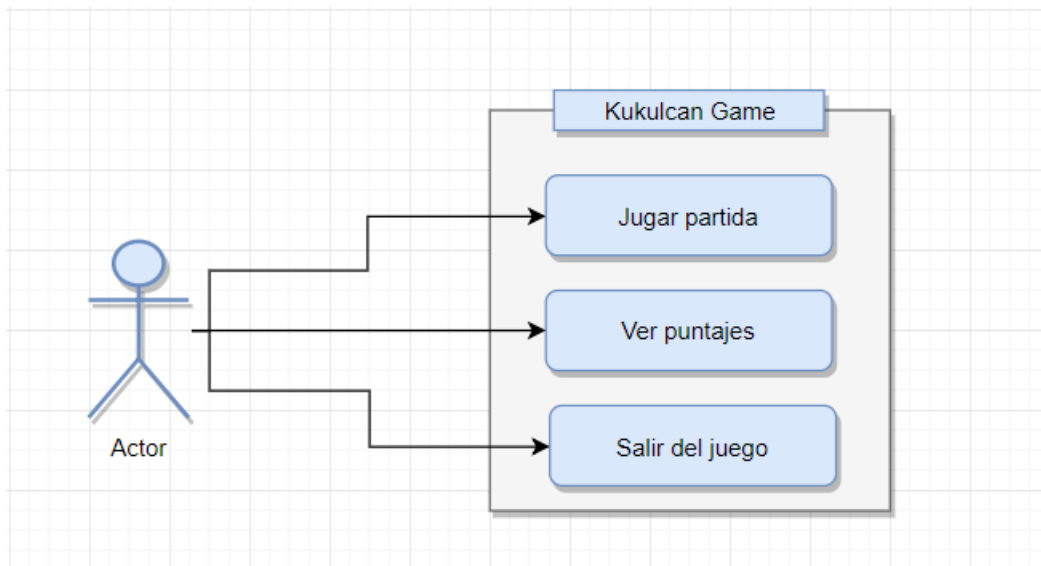
## Diagrama de casos de uso

Casos de uso del proyecto:

Los casos de uso los representaremos de la siguiente manera:

- Jugar partida: Caso de uso mediante el cual el actor podrá jugar una partida del juego.
- Ver puntajes: Caso de uso mediante el cual el actor podrá ver los mejores 5 puntajes del juego registrados en la base de datos.
- Salir del juego: Caso de uso mediante el cual el actor podrá salir de la aplicación.

Representados mediante el siguiente diagrama:



## Casos de uso con flujos alternativos

Flujos alternos de los casos de uso:

- Error al conectar la base de datos: Flujo alternativo que se genera al no conectar con la base de datos. Diagrama 1
- Kinect no conectado: Flujo alternativo que ocurre cuando no existe una referencia del Kinect v1. Diagrama 2

Los flujos alternos están representados por los siguientes diagramas:



Diagrama 1

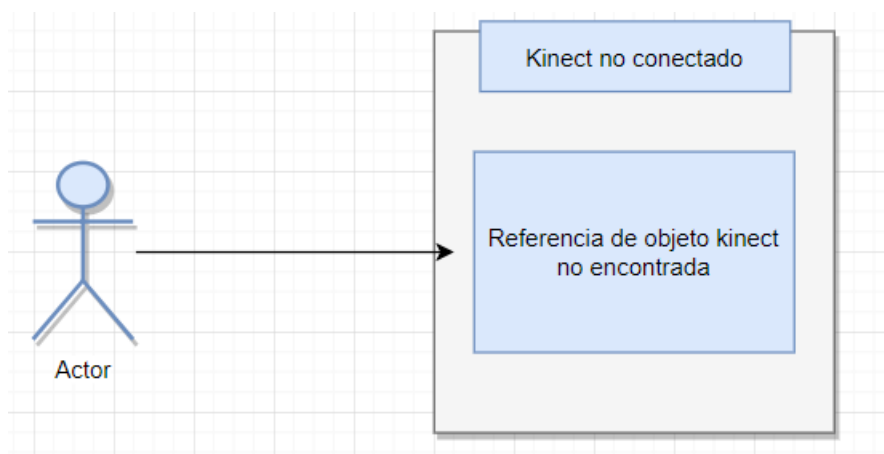


Diagrama 2

## Resumen de la evolución de los requerimientos y número total de requerimientos

Requerimientos funcionales.	Porcentaje de avance de la segunda entrega.	Porcentaje de avance de la entrega final.	Periodo de entrega establecido.
El proyecto será controlado mediante un Kinect.	70%	100%	Primera entrega
El proyecto contará con un splash de carga para gestionar la conexión del sensor Kinect.	100%	100%	Primera entrega
El proyecto contará con un menú principal.	100%	100%	Primera entrega
El proyecto contará con una pantalla de puntajes.	70%	100%	Primera entrega
El proyecto contará con una pantalla para guardar su puntaje obtenido en la partida ingresando las iniciales de su nombre.	0%	100%	Tercera entrega
El proyecto contará con una base de datos para administrar los puntajes de los jugadores.	0%	100%	Tercera entrega
El proyecto contará con una selección de mano antes de iniciar la partida con la cual el jugador decidirá con qué mano controlará los movimientos de la serpiente.	0%	100%	Tercera entrega
Número total: 7			

Requerimientos no funcionales.	Porcentaje de avance de la segunda entrega.	Porcentaje de avance de la entrega final.	Periodo de entrega establecido.
Se redireccionarán las vistas de cada pantalla mediante los botones.	50%	100%	Segunda entrega
Se mostrará el esqueleto del usuario representado mediante un stickman en la pantalla de juego.	100%	100%	Segunda entrega
El puntaje será mostrado en la parte superior de la pantalla de juego.	0%	100%	Tercera entrega
El usuario verá el control del snake mediante una serie de gesticulaciones en la pantalla de juego con una caja de colores.	0%	100%	Tercera entrega
Se mostrará un cursor para identificar la posición de la mano derecha del usuario.	100%	100%	Primera entrega
Número total: 5			



## DISEÑO

### MAPEO DE REQUERIMIENTOS

El mapeo de la funcionalidad de los requerimientos a base del diseño:

El proyecto será controlado mediante un Kinect, requerimiento cumplido al 100%.



El proyecto contará con un menú principal, requerimiento cumplido al 100%.



El proyecto contará con un splash de carga para gestionar la conexión del sensor Kinect, requerimiento cumplido al 100%.



El proyecto contará con una pantalla de puntajes, requerimiento cumplido al 100%.



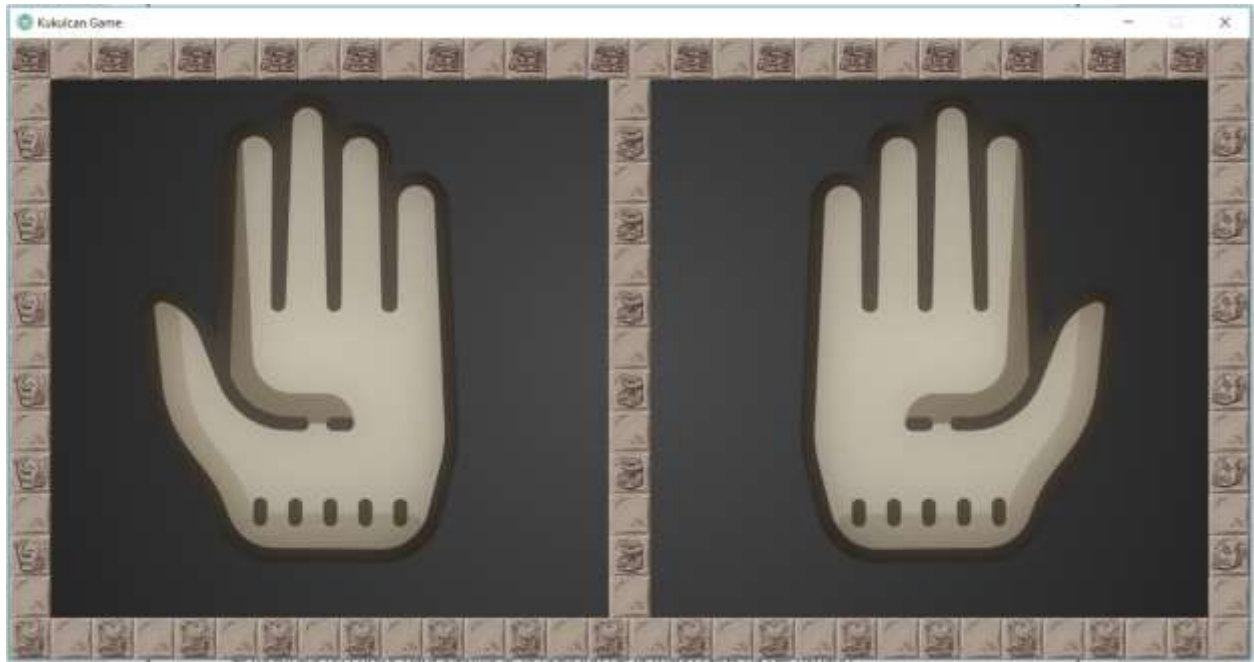
El proyecto contará con una pantalla para guardar su puntaje obtenido en la partida ingresando las iniciales de su nombre, requerimiento cumplido al 100%.



El proyecto contará con una base de datos para administrar los puntajes de los jugadores, requerimiento cumplido al 100%.

bdkukulcan puntajes	
🔑	id_puntaje : int(11)
📄	nombre : varchar(10)
#	puntaje : int(11)

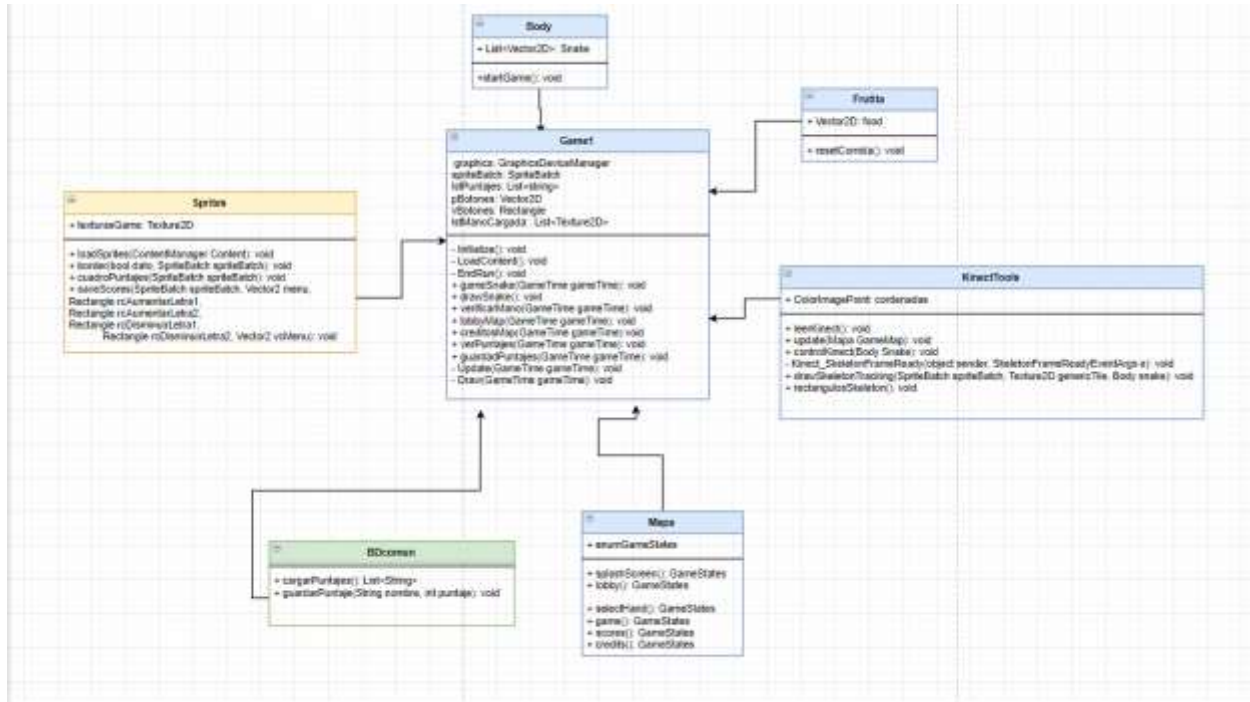
El proyecto contará con una selección de mano antes de iniciar la partida con la cual el jugador decidirá con qué mano controlará los movimientos de la serpiente, requerimiento cumplido al 100%.



De esta forma podemos ver que a base del mapeo de requerimientos quedan cubiertos al 100% en su funcionalidad.

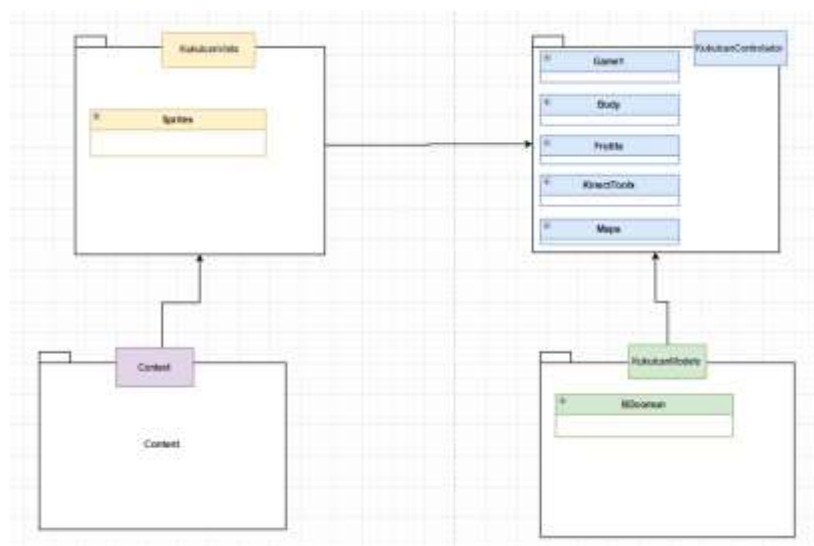
## DIAGRAMA DE CLASES

Diagrama de clases generado para el proyecto:



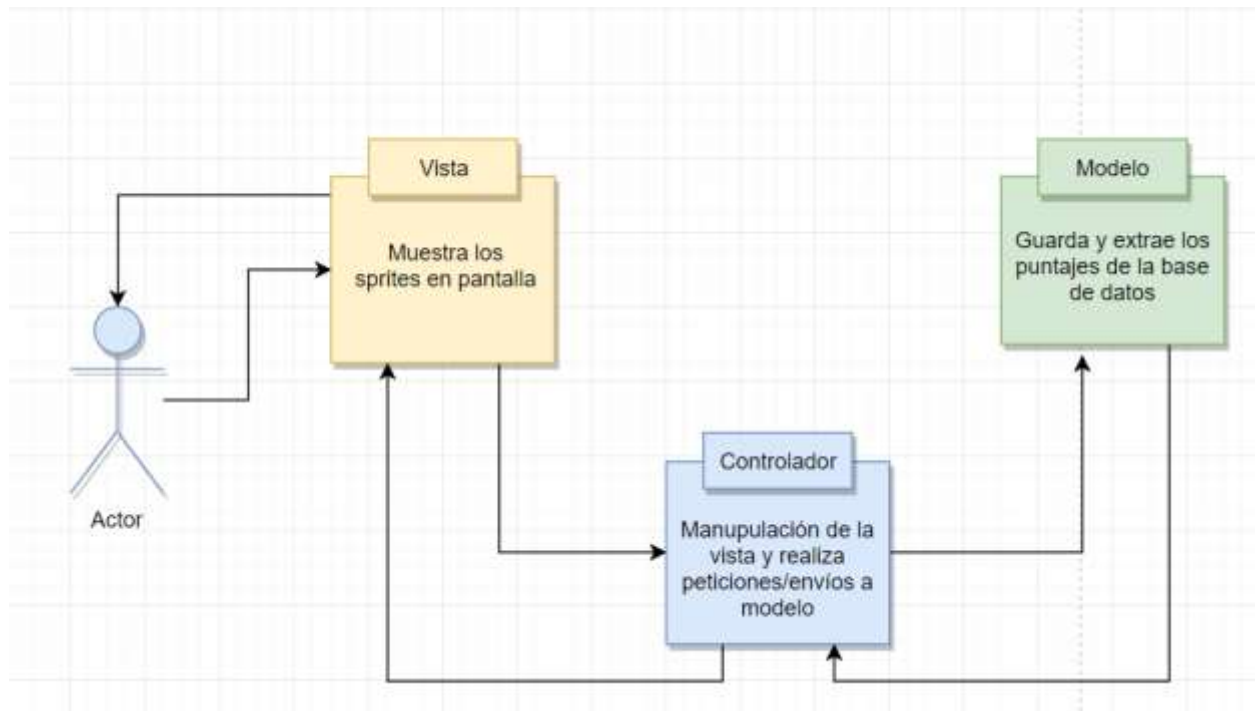
## DIAGRAMA DE PAQUETES

Este es el diagrama de paquetes en base al Modelo Vista Controlador:



## EVIDENCIA DE LA ARQUITECTURA MVC

Este diagrama evidencia el uso del MVC en el proyecto Kukulcan, la verdad fue un poco complicado aplicar esta estructura en la parte de videojuegos por la parte de los loops, pero aquí se demuestra que sí se pudo:



## IMPLEMENTACIÓN

### URL DEL REPOSITORIO

Este es el enlace del repositorio utilizado para el proyecto Kukulcán, debido a la desintegración de mi equipo, opté por migrar a un repositorio creado en mi cuenta y aquí se encuentra el código fuente del proyecto: <https://github.com/MagdielPm/KukulcanGame>

### ESTÁNDAR DE CODIFICACIÓN

El estándar de codificación utilizado está especificado de la siguiente manera:

- Nombramiento de clases:
  - Siempre el nombre de cada clase debe comenzar con una letra mayúscula y describir en su nombre la funcionalidad que tendrá.
- Nombramiento de variables y métodos:
  - Se deberán nombrar todas las variables de un bloque al inicio de este.
  - El nombre de cada variable o función debe representar, de manera clara, para qué va a servir en el programa.
  - La primera palabra de cada variable o método será escrita con letras minúsculas en su totalidad. De ser necesarias más palabras, cada una a partir de la segunda será escrita con su primera letra en mayúscula y las siguientes en minúscula.
  - No se permite el uso de signos de puntuación (guiones bajos, acentos) ni la letra ñ.
  - Las constantes empiezan con mayúscula.
- Contadores
  - Los contadores que no sirvan más que para llevar la cuenta de ciclos, hacer operaciones aritméticas o definir la posición de un vector deben ser escritos con las letras i, j, k, [...], donde 'i' siempre va a representar el primer contador que aparece en el bloque, 'j' el segundo, y así sucesivamente.
- Comentario general:
  - Va a estar ubicado en las primeras líneas del código, antes de cada método.
  - Debe incluir los siguientes datos referentes al código:
    - Descripción general del archivo.
    - Entradas.
    - Salidas.
- Comentario de cada método:
  - Estará ubicado antes del método.
  - Debe incluir los siguientes datos:
    - Descripción del método o comentario.
    - `<param>` (tipo y nombre de cada parámetro, separados por comas) `</param>`.
    - `<return>` (nombre del valor de retorno) `</return>`.

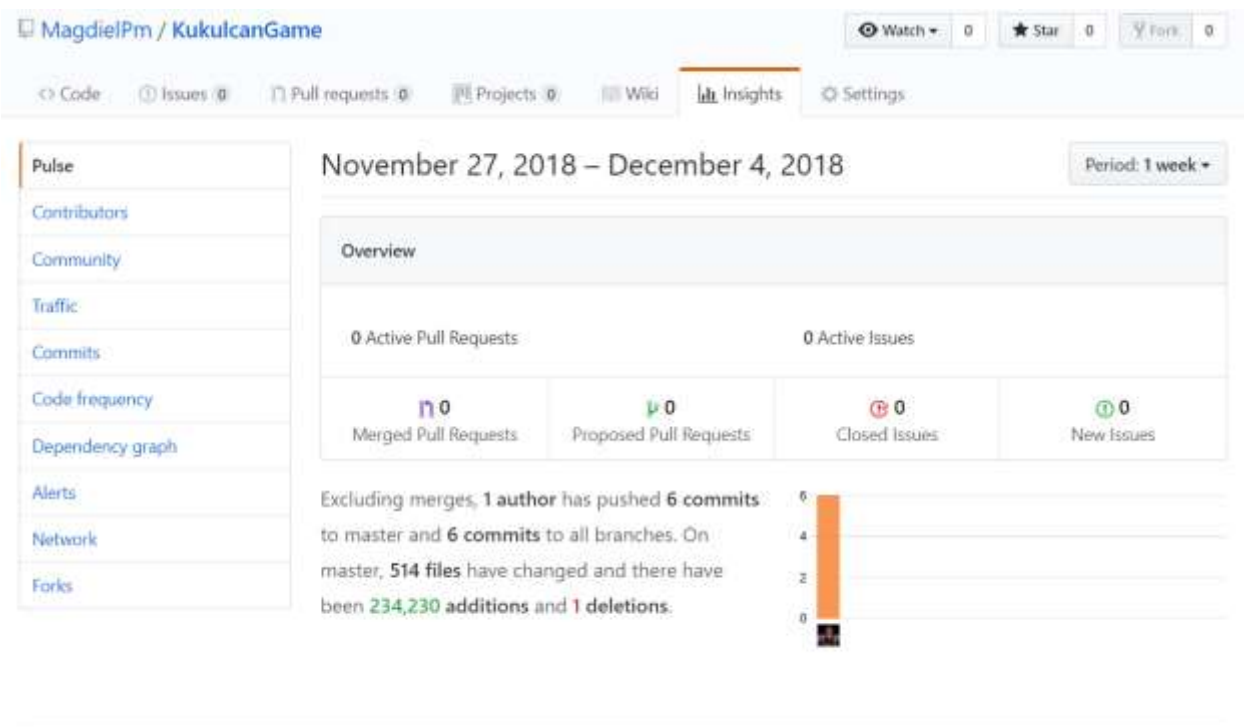
## DOCUMENTACIÓN DEL CÓDIGO

La documentación del código se encuentra en esta ruta de carpeta, partiendo de la raíz que en este caso es KukulcanGame, de todos modos, la incluiré en la carpeta de entrega final, es un archivo XML con la documentación del proyecto.

KukulcanGame\Kukulcan\KukulcanGame\bin\Windows\x86\Debug\KukulcanGame.XML

## REPORTE DE CONTRIBUCIÓN INDIVIDUAL DEL REPOSITORIO

Debido a que me quedé sin equipo en este proyecto, antes estábamos manejando el repositorio en la cuenta de uno de los integrantes del equipo, sin embargo, el abandono de todo mi obligó a crear un nuevo repositorio para esta entrega, aquí una evidencia del manejo del repositorio.





## PROCESO DE DESARROLLO

### DESCRIPCIÓN DE LOS ROLES

Actividades del líder del proyecto: Magdiel Pech Menéndez

- Organizar al equipo de trabajo para la producción y trabajo del proyecto
- Organización de los canales de comunicación y administrador de tareas
- Desarrollar el juego Kukulcán

Los entregables que se generaron por mi parte es el proyecto Kukulcan Game, las presentaciones de cada entrega, junto con la documentación y video de la entrega final.

### PROCESO DE MONITOREO

Se manejó la comunicación mediante un grupo de WhatsApp, en el cual únicamente se daban avisos rápidos, programaban citas o comunicados de ubicación para las juntas de trabajo. La otra aplicación para la gestión del monitoreo fue Slack, mediante el canal Game.



## DESCRIPCIÓN DE LAS MÉTRICAS DE CONTRIBUCIÓN INDIVIDUAL

Como métrica de contribución individual se determinó asignar a cada miembro del equipo un juego y administrar su proceso de desarrollo mediante la herramienta Trello, que se ligó a Slack para que cualquier movimiento o actividad se notificara en el canal de #game para mantener un informe constante del proceso de desarrollo por integrante.

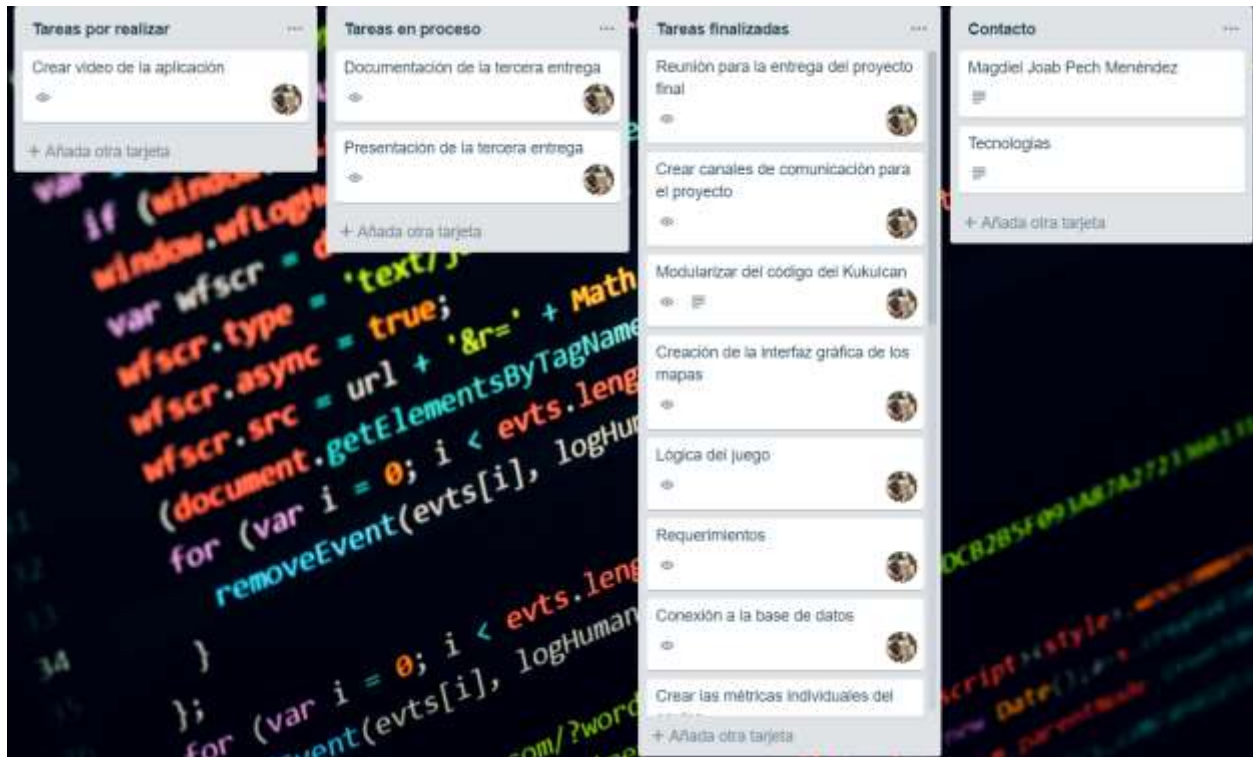
Como descripción de las métricas realizadas, se realizó un resumen de cada actividad asignándole un porcentaje a cada actividad, de esta forma se puede medir el rendimiento de cada integrante, obteniendo una suma de 100% en los procesos de métricas individuales.

Nombre del integrante	Tarea realizada	Porcentaje por tarea
Magdiel Pech Menéndez	Generar documentación del código	05%
Magdiel Pech Menéndez	Generar documentación de la entrega final del proyecto	10%
Magdiel Pech Menéndez	Desarrollo de juego Kukulcán	50%
Magdiel Pech Menéndez	Gestionar el proceso del avance del equipo de trabajo	10%
Magdiel Pech Menéndez	Creación del nuevo repositorio y canales de comunicación para el equipo de trabajo	05%
Magdiel Pech Menéndez	Creación del video de la última entrega	10%
Magdiel Pech Menéndez	Creación de la documentación de la segunda entrega	05%
Magdiel Pech Menéndez	Creación de la documentación de la primera entrega	05%

## REPORTE DE CONTRIBUCIÓN INDIVIDUAL

Como reporte de contribución individual, se manejó la aplicación de Trello para gestionar los avances de cada integrante del equipo.

Aquí adjunto el enlace para entrar al tablero, es público: <https://trello.com/b/27m7W26i/developing-game>

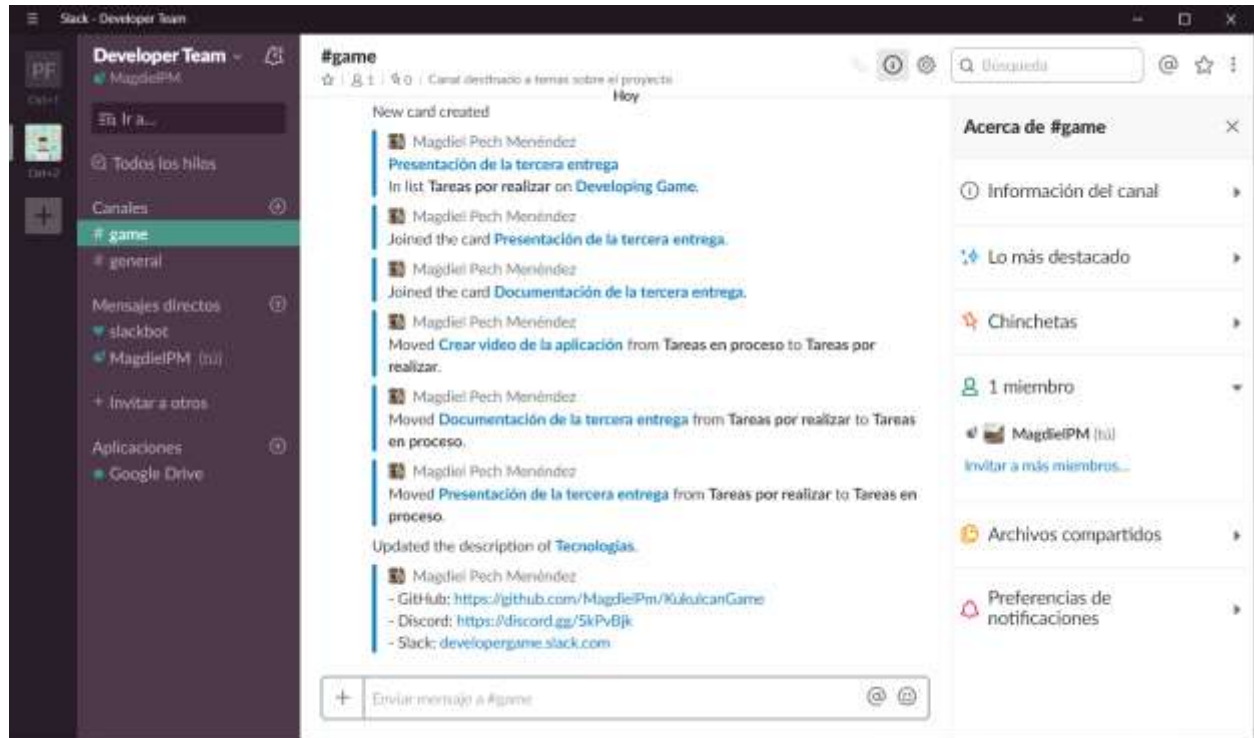


## DESCRIPCIÓN DEL PROCESO PARA LA ADMINISTRACIÓN DEL PROYECTO

La administración del proyecto se basó en el método ágil Kanban, se determinaron las tareas pendientes y las movíamos dependiendo del estado del proceso, adjunto una evidencia del canal de Slack que se creó para la comunicación y evidencias de trabajo en el tablero de Trello.

También adjunto el enlace para el acceso en el canal del grupo de desarrollo del proyecto.

<https://developergame.slack.com>



## ELEMENTOS ADICIONALES

Como elementos adicionales a la asignatura se aplicó el manejo del lenguaje de programación orientado a objetos C#, con un framework conocido como Monogame, también se utilizó una base de datos MySQL para gestionar los puntajes de los participantes, como último recurso adicional, está el manejo de la tecnología del sensor Kinect V1 para Xbox 360, mediante el cuál se realizó el control de todo el proyecto.