

# Plan de investigación

“Piñata de barrio” equipo 4.

1ºP. LIDTS



[Repositorio Github](#)

## Miembros:

Rodolfo Ramírez Díaz

Wendy Lizeth Robles Jose

Luis Ángel Serrano Catalá

Williams de Jesus Velasco Segura

Christian Jovanny Trejo Méndez

## Marco teórico

La gente ya hacía observaciones sobre como los cuerpos caían en forma de arco o con una curvatura, pero no sabían el por qué. Esto continuó hasta que Aristóteles afirmó que un “objeto permanece en reposo o se mueve en línea recta hacia el centro de la tierra a menos que se someta a una fuerza exterior” teniendo que interferir en esto y haga que se empiece a mover. A través de la información recopilada de sus diversos estudios empíricos hechos entre 1600 y 1660, Galileo Galilei pudo plantear nuevas ideas contrarias a las creencias de los Aristotélicos, quienes afirmaban “que un cuerpo de 10 veces más pesado que otro tardaba en caer 10 veces menos”. Realizando un experimento que trataba de utilizar su pulso para medir el tiempo de caída de dos cuerpos que relentizó utilizando planos inclinados, pudo afirmar que “todos los objetos caían a  $9.8 \text{ m/s}^2$ ”.

También realizó pruebas donde impulsó un objeto horizontalmente y otro que dejó caer desde el borde de una mesa, y encontró que ambos llegaban al piso al mismo tiempo. De esa observación, afirmó que “la componente vertical del movimiento de un objeto que cae es independiente de cualquier movimiento horizontal que lo acompañe”. Con eso se establece lo que llamamos *principio de superposición*, que dictamina que un movimiento consta de dos componentes que actúan al mismo tiempo pero, cuando se estudian, puede suponerse que primero ocurre uno y luego el otro.

Con investigaciones posteriores, los experimentos y afirmaciones de Galileo fueron comprobados y/o mejorados, siendo lo que al día de hoy conocemos como un *set* de fórmulas para calcular la trayectoria de cuerpos.

$$V_f = V_o \pm g \cdot t$$

$$h = V_o \cdot t \pm \frac{g \cdot t^2}{2}$$

$$h = \frac{1}{2} g \cdot t^2$$

$$V_f^2 = V_o^2 \pm 2 g \cdot h$$

Hernández, Landa (2004). Lecciones de Matemáticas 3. Colegio de Ciencias y Humanidades Naucalpan.

El término *motor de juego* (*game engine* en inglés) surgió a mediados de los años 90 con la aparición del famosísimo juego de acción en primera persona Doom, desarrollado por la compañía *id Software* bajo la dirección de *John Carmack*. Esta afirmación se sustenta sobre el hecho de que Doom fue diseñado con una arquitectura orientada a la reutilización mediante una separación adecuada en distintos módulos de los componentes fundamentales, como por ejemplo el sistema de renderizado gráfico, el sistema de detección de colisiones o el sistema de audio, y los elementos más artísticos, como por ejemplo los escenarios virtuales o las reglas que gobernaban al propio juego.

Este enfoque ha ido evolucionando y se ha expandido, desde la generación de *mods* por desarrolladores independientes o *amateurs* hasta la creación de una gran variedad de herramientas, bibliotecas e incluso lenguajes que facilitan el desarrollo de videojuegos. A día de hoy, una gran parte de compañías de desarrollo de videojuego utilizan motores o herramientas pertenecientes a terceras partes, debido a que les resulta más rentable económicamente y obtienen, generalmente, resultados espectaculares. Por otra

parte, esta evolución también ha permitido que los desarrolladores de un juego se planteen licenciar parte de su propio motor de juego, decisión que también forma parte de su política de trabajo.

Como conclusión final, resulta relevante destacar la evolución relativa a la generalidad de los motores de juego, ya que poco a poco están haciendo posible su utilización para diversos tipos de juegos. Sin embargo, el compromiso entre generalidad y optimalidad aún está presente. En otras palabras, a la hora de desarrollar un juego utilizando un determinado motor es bastante común personalizar dicho motor para adaptarlo a las necesidades concretas del juego a desarrollar.

J. Vives I. (2020). *Desarrollo de Videojuegos: Un Enfoque Práctico*. 3ra edición. Universidad de Castilla-La Mancha.

## Objetivos

- Simular correctamente la trayectoria parabólica según las variables insertadas

## Específicos

- Crear un juego atractivo que demuestre el uso de la fórmula de tiro parabólico en una situación común.
- Aprender con el desarrollo del juego
- Compartir todo el progreso por medio de una plataforma de código abierto.

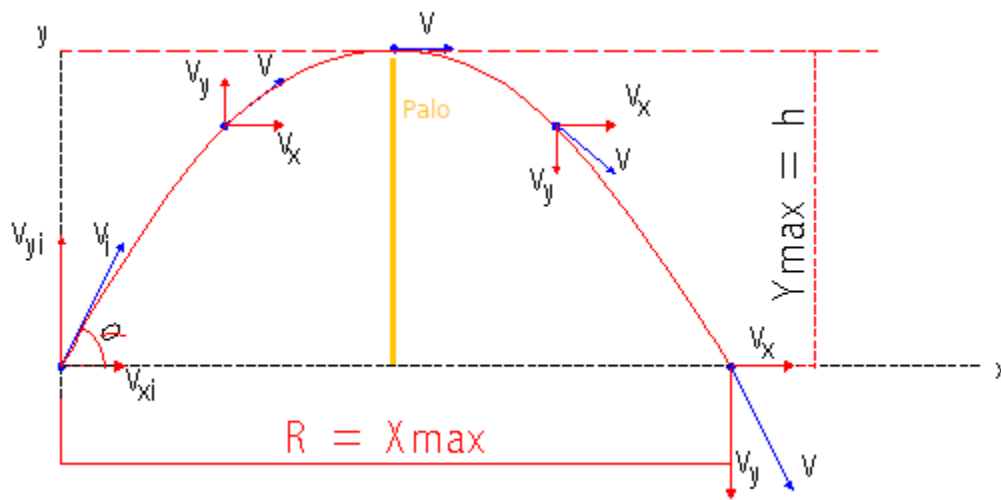
## Planteamiento del problema

La física soportada por las matemáticas son de gran valor pues permiten resolver problemáticas por más pequeñas, facilitando tareas y ahorrando tiempo.

¿Cómo la física puede ayudarnos en problemas? Con el uso de diversos modelos físico-matemáticos que nos permiten predecir eventos con poco conocimiento previo.

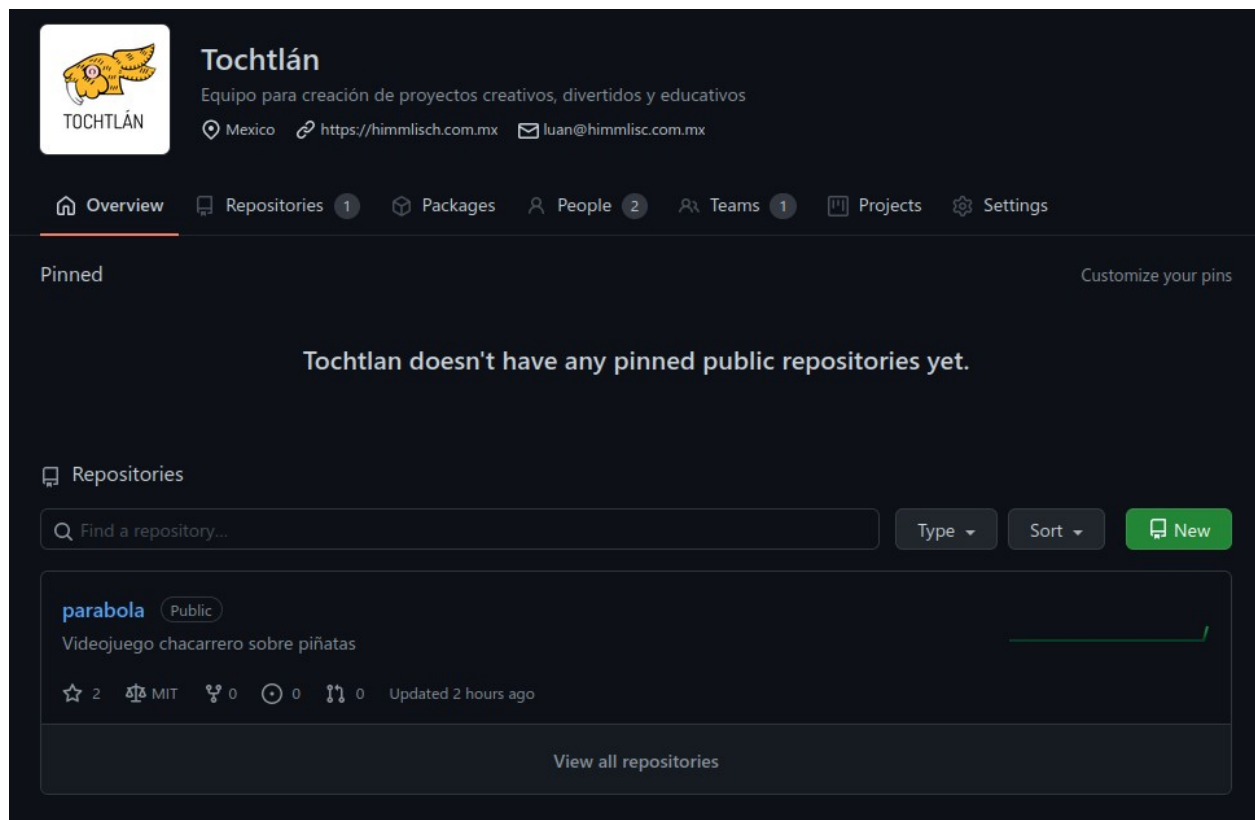
En el mundo diversos sistemas existen gracias a predicciones exactas, pero aún falta llevar este conocimiento a una escala menor. A las personas comunes les es indiferente el uso y beneficios de estos modelos y cómo es que formulas, como la del tiro parabólico, podría beneficiarlos.

Así pues, el objetivo del proyecto es enseñar los beneficios de los modelos físicos-matemáticos y más en específico del de la fórmula del tiro parabólico utilizándolo en un escenario real: se tendrá que colgar una piñata y para ello tendremos que calcular la trayectoria de una cuerda. Gracias a la correcta predicción de la trayectoria la cuerda se podrá posicionar donde se debe y colgar nuestra piñata.



## Estrategia de solución

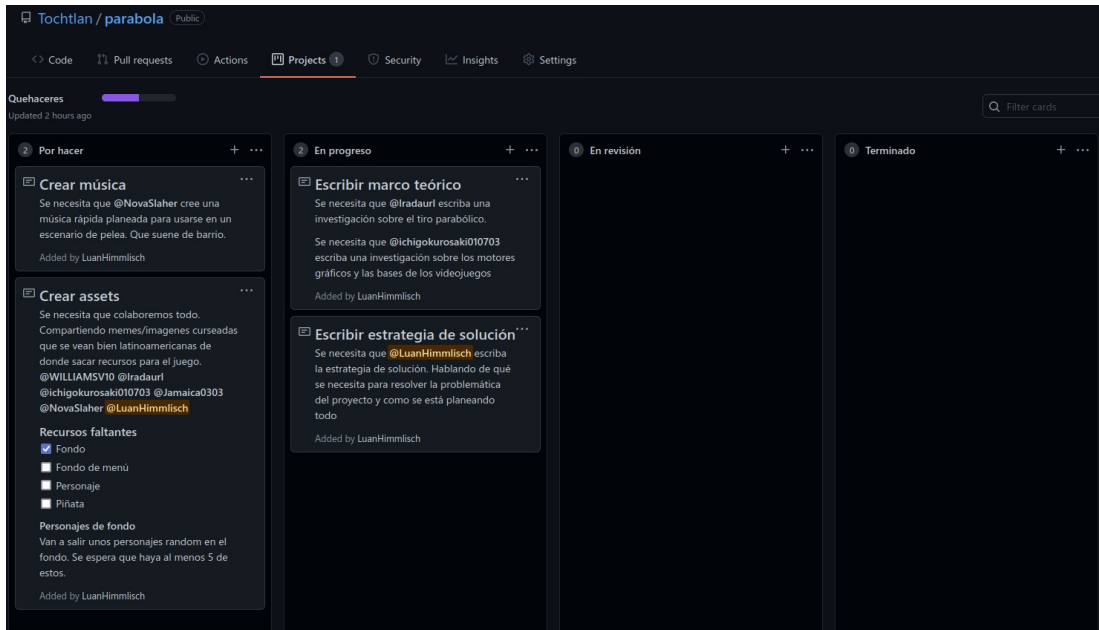
La creación de videojuegos es un proceso que involucra todas las artes e involucra habilidades específicas únicas en el campo: música, efectos de sonido, gráficos, experiencia del usuario, diseño de niveles, etc. Así llegamos a la conclusión que, más que en cualquier otro proyecto, es necesario planear el nuestro. Primero, teniendo que establecer un espacio de colaboración efectivo.



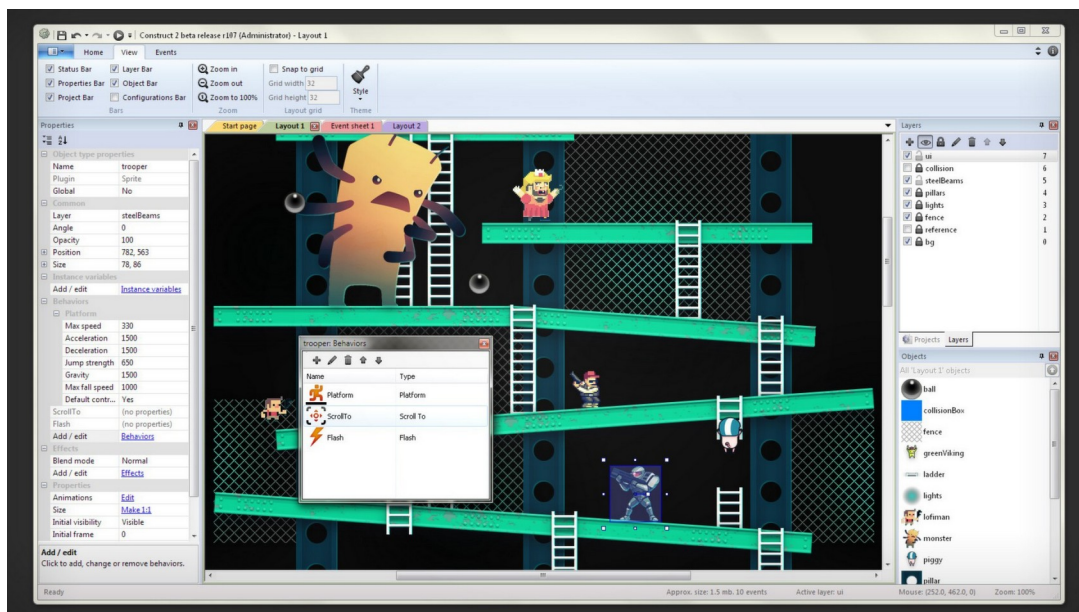
Para esto, hemos creado la organización Tochtlán, en la plataforma de Github, donde se planea subir nuestros proyectos desde ahora al futuro.

Además que Github nos ayuda con el objetivo de hacer que el proyecto sea de ayuda a futuros estudiantes. Haciendo que el repositorio donde estarán todos los archivos del proyecto, libres bajo la licencia del MIT.

Hacemos uso del organizador Kanban de Github para mantener todo lo planeado en un solo lugar.



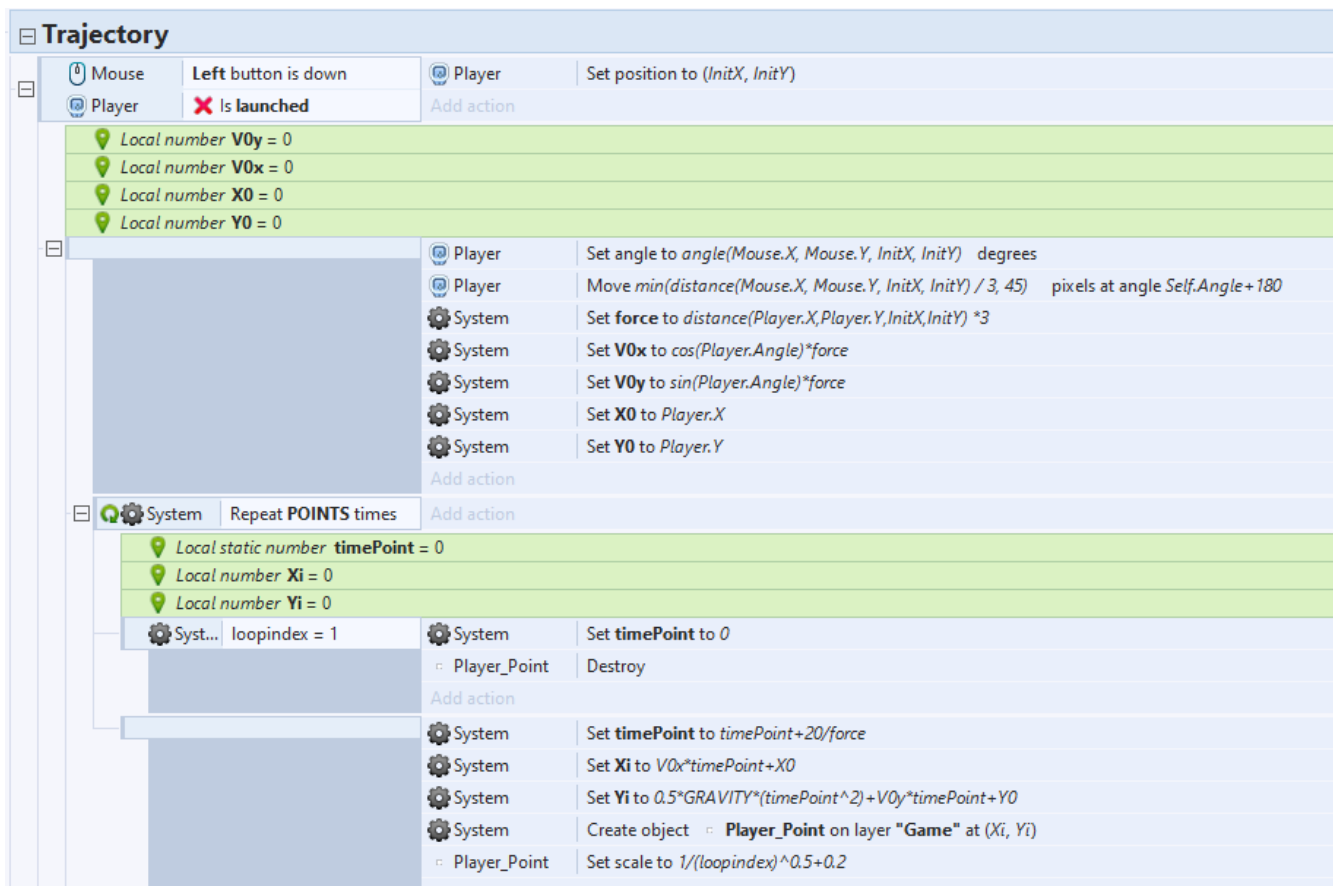
Una vez teniendo esto, pasamos con la búsqueda de un motor de videojuegos que se alinee con nuestros objetivos y requerimientos.





Existen diversos motores gráficos y herramientas para diversos tipos de juego. Ya teniendo en claro nuestro plan de acción, podemos fácilmente elegir cuál nos conviene.

En nuestro caso, utilizaríamos el motor para la creación de videojuegos 2D, Construct 2. Por su facilidad de uso y su programación visual perfecta para estudiantes recién iniciados como nosotros.



Con todas estas herramientas una vez en lugar, podremos dedicarnos a crear los recursos del juego, para esto hemos decidido un estilo de gráficos realistas que se apeguen al ambiente urbano-cotidiano de la persona común.

Se tiene que hacer una investigación a fondo en motores de búsqueda para encontrar imágenes con licencia apropiadas o hacer modificaciones respectivas para que su uso sea considerado como [Uso justo](#), pues al final los videojuegos no son más que una expresión artística más.

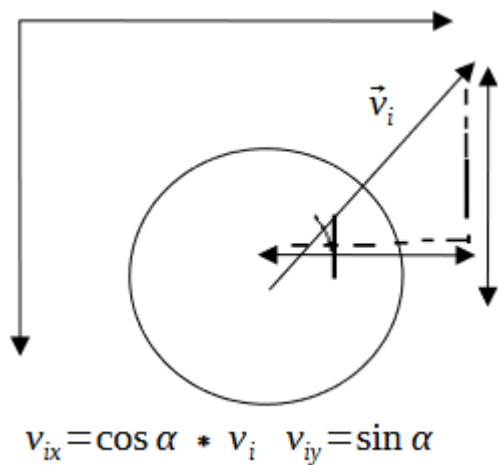
Para la creación de la música se hará uso del software FL Studio. Un software propietario para la composición de música y efectos de sonido.



## Fórmula de tiro parabólico

Para despejar nuestra fórmula es necesario determinar qué datos hacen falta. En nuestro caso, sabemos que la gravedad es  $g=9.81\text{ ms}^{-2}$  y que el ángulo y velocidad inicial (fuerza) serán datos que dependerán del usuario.

Para despejar la fórmula debemos encontrar las componentes X y Y de nuestra velocidad inicial. La siguiente fórmula es utilizada.



Después, hacemos uso de la 2da ley de Newton. Habla de que la suma de todas las fuerzas que se ejercen sobre un cuerpo, es igual a su masa y aceleración.

$$\begin{aligned} \sum F &= m * \vec{a} \leftrightarrow \vec{P} = m * \vec{g} \\ &\leftrightarrow m * \vec{g} = m * \vec{a} \end{aligned}$$

Una vez sustituido, tendremos que descomponer la fórmula a una con los datos que utilizaremos.

$$\begin{cases} x=0=\max \\ y=mg=may \end{cases}$$

Se aíslan las aceleraciones, sabiendo que la gravedad siempre siempre será una aceleración vertical presente:

$$\leftrightarrow \begin{cases} ax=0 \\ ay=g \end{cases}$$

E iniciamos a integrar nuestras operaciones:

$$\begin{cases} \int ax * dt = \int 0 * dt \\ \int ay * dt = \int g * dt \end{cases} \rightarrow \begin{cases} vx=v_0x \\ vy=+v_0y \end{cases}$$

De la integración de la aceleración sabremos la velocidad y al integrarla nuevamente obtenemos como resultado la posición del proyectil:

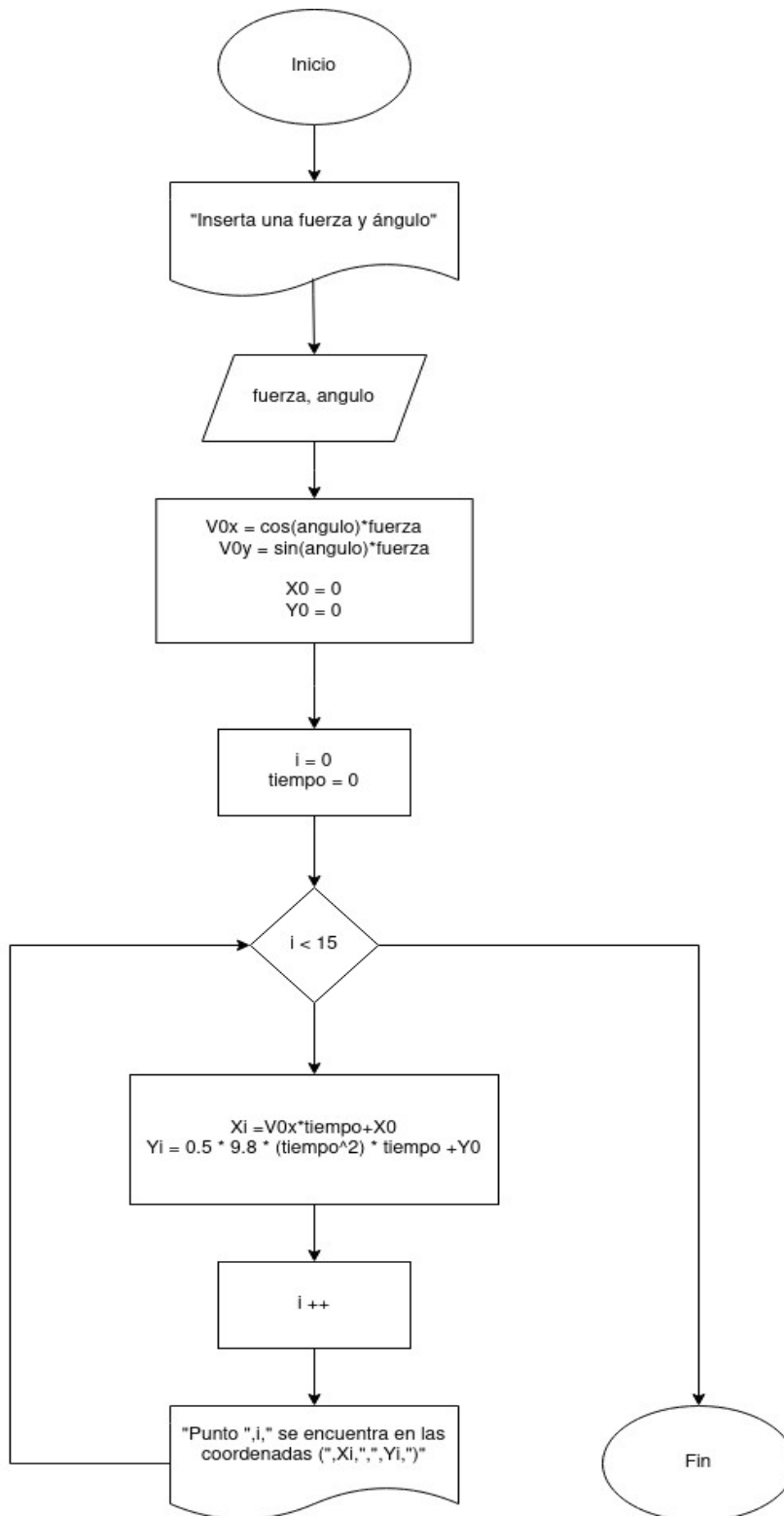
$$\rightarrow \begin{cases} \int v_x * dt = \int v_{0x} * dt \\ \int v_y * dt = \int +v_{0y} * dt \end{cases} \rightarrow \begin{cases} x = v_{0x} * t + x_0 \\ y = \frac{1}{2} g t^2 + v_{0y} * t + y_0 \end{cases}$$

Al finalizar, como resultado obtendremos las fórmulas necesarias para predecir la posición del proyectil dependiendo de un tiempo dado.

$$x = v_{0x} * t + x_0$$

$$y = \frac{1}{2} g t^2 + v_{0y} * t + y_0$$

## Diagrama



## Pseudo-código

Si click izquierdo es presionado y jugador no ha sido lanzado:

variables  $V_0x$ ,  $V_0y$ ,  $X_0$ ,  $Y_0$

voltear jugador a  $(\text{puntero.x}, \text{puntero.y})$

mover jugador a  $\min(\text{distancia}(\text{puntero.x}, \text{puntero.y}, \text{jugador.inicioX}, \text{jugador.inicioY})/3, 45)$

$\text{fuerza} = \text{distancia}(\text{jugador.x}, \text{jugador.y}, \text{jugador.inicioX}, \text{jugador.inicioY})$

$V_0x = \cos(\text{jugador.angulo}) * \text{fuerza}$

$V_0y = \sin(\text{jugador.angulo}) * \text{fuerza}$

$X_0 = \text{jugador.x}$

$Y_0 = \text{jugador.y}$

repetir 15 veces:

variables tiempo,  $X_i$ ,  $Y_i$

si iterador es 1:

eliminar puntos

$\text{tiempo} = \text{tiempo} + 20 / \text{fuerza}$

$X_i = V_0x * \text{tiempo} + X_0$

$Y_i = 0.5 * \text{gravedad} * \text{tiempo}^2 + V_0y * \text{tiempo} + Y_0$

dibujar punto en  $(X_i, Y_i)$

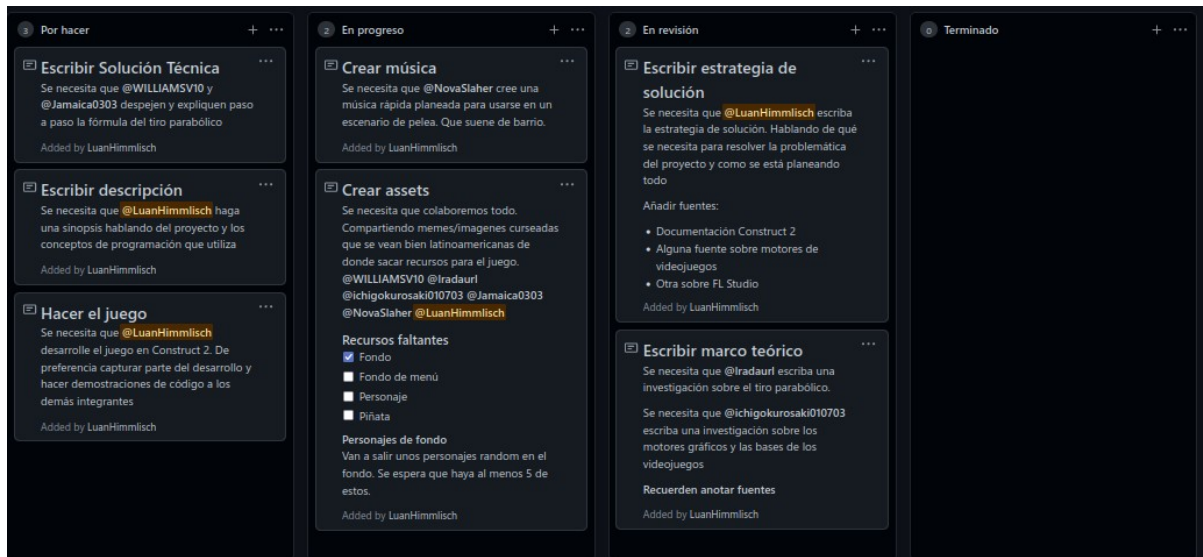
cambiar escala de punto a  $1/\text{iterador}^{0.5+0.2}$

## **Conclusión**

El desarrollo de un proyecto académico como lo es Piñata de Barrio conlleva un esfuerzo completo en diferentes áreas, no tan solo en la programación, si no que también en unas que muchas veces ignoramos: planeamiento, organización, investigación teórica, diseño, composición de música y muchas otras. Por lo que campos como programación estructurada nos son de gran ayuda para estos retos y nos dan invaluable conocimientos para nuestra futura carrera profesional.

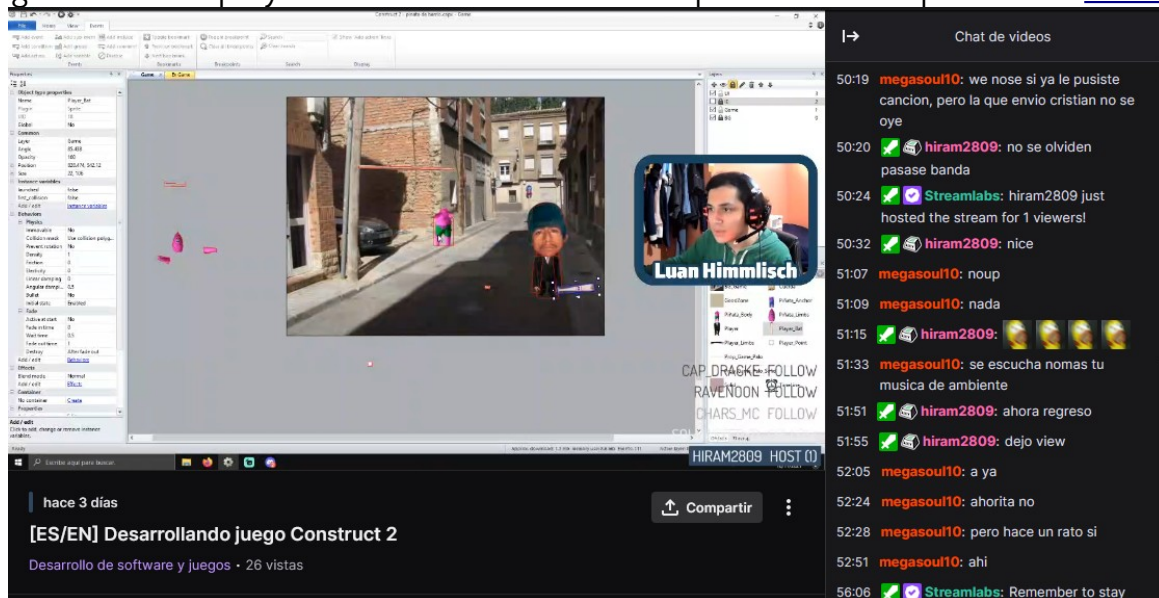


# Cronograma de actividades



Puedes encontrar el muro de actividades en vivo en [github](#).

La programación del proyecto ha sido realizada en vivo por medio de la plataforma [Twitch](#) y



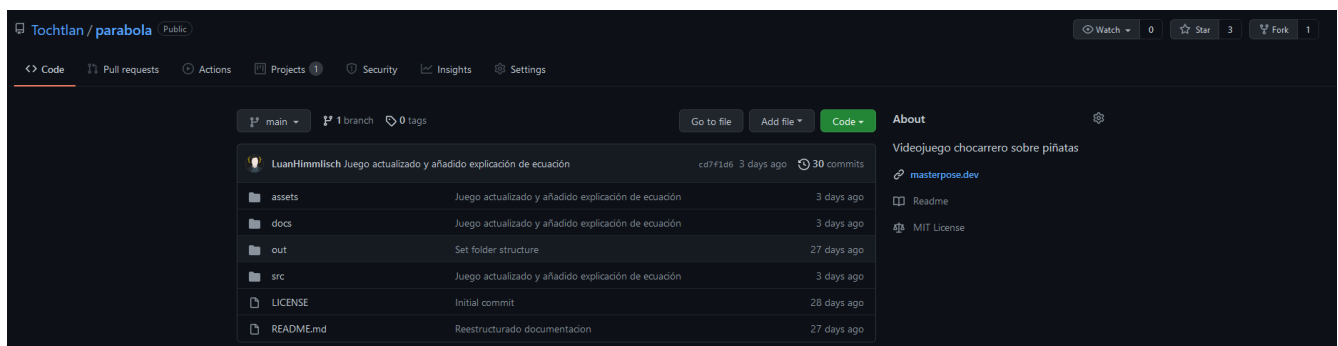
guardada en los [archivos del canal](#).

## Código fuente

El juego y sus ejecutables están disponibles en su [página de itch.io](#).



El código fuente está hosteado en el [repositorio de Github](#).



## Bibliografía

- Hernández, Landa, et.al (2004). *Lecciones de Matemáticas 3*. Colegio de Ciencias y Humanidades Naucalpan.
- J. L. Alvarez G. y Y. Posadas V (2002). *La obra de Galileo y la conformación del experimento en la física*. Revista Mexicana de Física.
- J. Vives I. (2020). *Desarrollo de Videojuegos: Un Enfoque Práctico*. 3ra edición. Universidad de Castilla-La Mancha.
- A. Gullen y T. Gullen (2017). *Online Manual & Documentation*. Scirra LTD. Recuperado de: <https://www.construct.net/en/make-games/manuals/>