

**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA  
FACULTAD DE INGENIERÍA, ARQUITECTURA Y DISEÑO**

**INGENIERÍA EN SOFTWARE Y TECNOLOGÍAS EMERGENTES.**

# **MANUAL TÉCNICO**

## **COSMIC CHAOS**

**Desarrolladores:**

Francisco Cornejo Pacheco  
Diego Alonso Ibarra Galindo

**Materia:** Programación Estructurada

**Profesor:** Pedro Nuñez Yepiz

**10 de Junio de 2024**

# ÍNDICE

<b>INTRODUCCIÓN.....</b>	<b>3</b>
<b>1. DESCRIPCIÓN GENERAL.....</b>	<b>3</b>
2.1 ALCANCE.....	3
2.2 OBJETIVOS.....	4
<b>2. CARACTERÍSTICAS DEL JUEGO:.....</b>	<b>4</b>
2.1 OBJETIVOS DEL JUEGO.....	5
2.2 JUGABILIDAD.....	5
2.3 OBJETOS COLECCIONABLES.....	5
<b>3. TÉRMINOS Y DEFINICIONES.....</b>	<b>6</b>
<b>4. MODELO DE ANÁLISIS.....</b>	<b>7</b>
4.1 COMPONENTES DEL SOFTWARE.....	8
4.2 MODELO DE DISEÑO.....	9
<b>5. ASPECTOS TECNICOS.....</b>	<b>10</b>
5.1. INSTRUCCIONES DE INSTALACIÓN.....	10
Windows.....	10
Linux, Mac.....	10
5.2 HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO.....	10
5.3 ESTRUCTURA DE ARCHIVOS DEL PROYECTO:.....	11
5.4 MODIFICACIÓN LOCAL.....	11
<b>6. ARQUITECTURA DEL SISTEMA.....</b>	<b>13</b>
6.1 REQUISITOS DEL SISTEMA.....	13
BIBLIOGRAFÍA.....	14

# INTRODUCCIÓN

En este manual, encontrarás información detallada sobre la estructura, funcionamiento y componentes del juego, así como instrucciones para su desarrollo, modificación y mantenimiento.

A lo largo de este manual, explicaremos los componentes clave del juego, desde la mecánica hasta la integración educativa y la arquitectura técnica subyacente.

## 1. DESCRIPCIÓN GENERAL

Cosmic Chaos es un videojuego en 2D que ofrece una experiencia única que combina acción, aprendizaje interactivo y desafíos estratégicos en el espacio exterior. Desde el control de una nave hasta la integración de preguntas educativas sobre el cosmos, este juego ha sido diseñado para ofrecer una experiencia inmersiva y enriquecedora.

La principal tarea del jugador es evitar colisiones con los meteoritos que caen del espacio exterior, salvaguardando así la integridad de la nave y su vida en el juego.

Durante el transcurso del juego, los jugadores tienen la oportunidad de recolectar varios recursos, incluyendo vidas adicionales, monedas de oro y power-ups. Estos power-ups se obtienen como recompensa por responder correctamente a preguntas relacionadas con aspectos del cosmos, tales como planetas, estrellas y fenómenos astronómicos.

La implementación de preguntas educativas, que funcionan como llaves para desbloquear recompensas dentro del juego, es una metodología efectiva para mejorar la retención de conocimientos y despertar la curiosidad científica. Esta técnica pedagógica, integrada de manera fluida en la mecánica del juego, no solo beneficia la experiencia lúdica, sino que también fortalece la comprensión de los jugadores sobre temas astronómicos, fomentando así un aprendizaje significativo y duradero.

### 2.1 ALCANCE

Cosmic Chaos se presenta como una experiencia de juego innovadora que va más allá del simple entretenimiento, fusionando elementos de acción, aprendizaje interactivo y desafíos estratégicos en un entorno espacial envolvente. A continuación, se detallan los alcances del juego en sus diferentes dimensiones:

### 1. Jugabilidad:

- **Mecánica principal:** El jugador controla una nave espacial, esquivando meteoritos que caen del espacio para preservar su integridad y continuar su viaje.
- **Desafíos:** La dificultad aumenta progresivamente, introduciendo meteoritos más veloces y en mayor cantidad, así como obstáculos adicionales.
- **Recompensas:** La recolección de recursos como vidas adicionales, monedas de oro y power-ups incentiva al jugador y añade un elemento estratégico a la experiencia.

### 2. Aprendizaje interactivo:

- **Preguntas educativas:** Integradas de forma natural en la dinámica del juego, estas preguntas evalúan el conocimiento del jugador sobre temas astronómicos.
- **Retención de conocimientos:** La asociación de preguntas correctas con recompensas en el juego refuerza el aprendizaje y la memorización de conceptos.
- **Curiosidad científica:** El juego despierta el interés por la astronomía, motivando a los jugadores a explorar y aprender más sobre el cosmos.

### 3. Aspectos pedagógicos:

- **Aprendizaje significativo:** La metodología de gamificación, que combina juego y aprendizaje, promueve una comprensión profunda y duradera de los temas.
- **Motivación intrínseca:** La satisfacción de superar desafíos y obtener recompensas impulsa a los jugadores a continuar aprendiendo.
- **Escenario atractivo:** El entorno espacial cautivador y la narrativa del juego generan un contexto propicio para el aprendizaje.

## 2.2 OBJETIVOS

Proporcionar una experiencia interactiva que estimule la curiosidad de los jugadores sobre el espacio exterior y el sistema solar, utilizando preguntas relacionadas con estos temas para aumentar el interés y la motivación por aprender.

## 2. CARACTERÍSTICAS DEL JUEGO:

Esta sección proporciona una visión detallada de los objetivos, la jugabilidad y los objetos coleccionables del juego. Entender estos elementos es esencial para maximizar la experiencia de juego y mejorar tus habilidades.

## 2.1 OBJETIVOS DEL JUEGO

- Sobrevivir el mayor tiempo posible esquivando los meteoros y recolectando monedas, vidas y power-ups.
- Acumular la mayor cantidad de puntos posible al destruir meteoros y recolectar monedas de oro.
- Alcanzar nuevos niveles de dificultad a medida que acumulas puntos en la partida.
- Demostrar y ampliar conocimientos sobre el sistema solar al responder preguntas sobre ello, para obtener los power-ups.

## 2.2 JUGABILIDAD

- El jugador controla una nave, la cual puede mover hacia arriba, abajo, izquierda y derecha, debe esquivar los meteoros que caen e intentar sobrevivir.
- Al iniciar una partida:
  - Se te brindan 5 vidas, cada colisión con un meteorito te resta una vida.
  - Se te brindan 10 misiles, los cuales puedes usar para destruir y protegerte de los meteoros, al destruir un meteorito se te brindan 5 puntos.
- Puedes recolectar diferentes objetos que te brindan powerups como escudos y munición.
- Cuando tomas un Power-Up, se te muestra una pregunta sobre el sistema solar, al responder correctamente se te brinda el PowerUp correspondiente.
- Si respondes correctamente 3 preguntas seguidas, se te brindan 2 vidas adicionales.
- El juego cuenta con 3 niveles de dificultad, los cuales vas alcanzando conforme subes puntos en la partida actual.
- La dificultad del juego va aumentando conforme subes de nivel, ya que van apareciendo más obstáculos
- En cada sesión del juego, se guardan tus mejores estadísticas obtenidas y puedes consultarlas cuando desees.

## 2.3 OBJETOS COLECCIONABLES

Durante el juego, van cayendo diferentes objetos con diferentes características, que puedes recolectar. A continuación, la descripción de cada objeto:

- **Caja de munición:** Te muestra una pregunta sobre el sistema solar, si respondes correctamente, se te otorga munición adicional.
- **Escudo:** Te muestra una pregunta sobre el sistema solar, si respondes correctamente, se te otorgan dos escudos. (cada escudo otorga inmunidad a un meteorito).
- **Corazón:** Te otorga una vida adicional.

- **Moneda de oro:** Te otorga 10 puntos.
- **Meteoro gris y café:** Al colisionar con uno de ellos, se te resta una vida, acaba el juego si pierdes todas las vidas.

### 3. TÉRMINOS Y DEFINICIONES

A continuación, encontrarás un glosario con definiciones de términos técnicos y específicos utilizados en Cosmic Chaos, para proporcionar una comprensión más clara.

**1. Juego lúdico:** Un juego lúdico es una actividad recreativa diseñada principalmente para el entretenimiento y la diversión.

**2. Hardware:** Definición: Parte física y tangible de un sistema informático o dispositivo electrónico.

**3. Software:** Definición: Parte lógica e intangible de un sistema informático, compuesta por programas y datos.

**4. Power-Up:** Es un elemento o efecto que mejora las habilidades o capacidades del jugador o del personaje que controla. Estos power ups pueden tomar diversas formas y tener una variedad de efectos, dependiendo del juego específico en cuestión.

**5. Colisión:** En un videojuego, una colisión ocurre cuando dos objetos en el mundo virtual interactúan de alguna manera. Puede ser cuando un personaje choca contra una pared, dos personajes se encuentran, o un proyectil impacta contra un objetivo.

**6. Objetos coleccionables:** Los objetos coleccionables en videojuegos son ítems, elementos, o artefactos que los jugadores pueden encontrar, recolectar y a menudo utilizar dentro del juego. Estos objetos suelen estar distribuidos a lo largo del mundo del juego y pueden tener diversos propósitos, como mejorar la experiencia de juego, desbloquear contenido adicional, o simplemente servir como metas opcionales para los jugadores más dedicados.

## 4. MODELO DE ANÁLISIS

En esta sección se hará la representación sistemática y detallada del sistema de software, para entender mejor sus componentes, interacciones y comportamiento.

1. **Jugador:** Representa al usuario dentro del juego, controlando una nave espacial que debe moverse y realizar diversas acciones.

**Acciones:**

- **Moverse:** Desplazarse horizontalmente para evitar colisiones con meteoritos.
- **Interactuar con Objetos:** Recoger vidas adicionales, monedas de oro y power-ups.
- **Responder Preguntas:** Contestar preguntas sobre el cosmos para obtener power-ups.

2. **Objetos Interactivos:** Elementos en el juego con los que el jugador puede interactuar para obtener beneficios o avanzar en el juego.

- **Meteoritos:** Obstáculos que el jugador debe evitar.
- **Vidas Adicionales:** Permiten al jugador continuar jugando después de una colisión.
- **Monedas de Oro:** Recursos coleccionables que pueden incrementar la puntuación o desbloquear nuevas características.
- **Power-ups:** Mejoras temporales que se obtienen al responder correctamente preguntas educativas sobre el cosmos.

3. **Preguntas Educativas:** Elementos interactivos que presentan desafíos de conocimiento al jugador.

4. **Recursos Gráficos:** Las texturas e imágenes que conforman los elementos visuales del juego.

- **Nave Espacial:** Diseño gráfico de la nave controlada por el jugador.
- **Meteoritos:** Texturas y animaciones de los meteoritos que el jugador debe esquivar.
- **Entorno Espacial:** Fondos y escenarios que representan el espacio exterior.
- **Elementos de UI:** Iconos y menús para vidas, monedas, preguntas, y power-ups.
- **Animaciones:** Movimientos y efectos visuales que añaden dinamismo al juego.

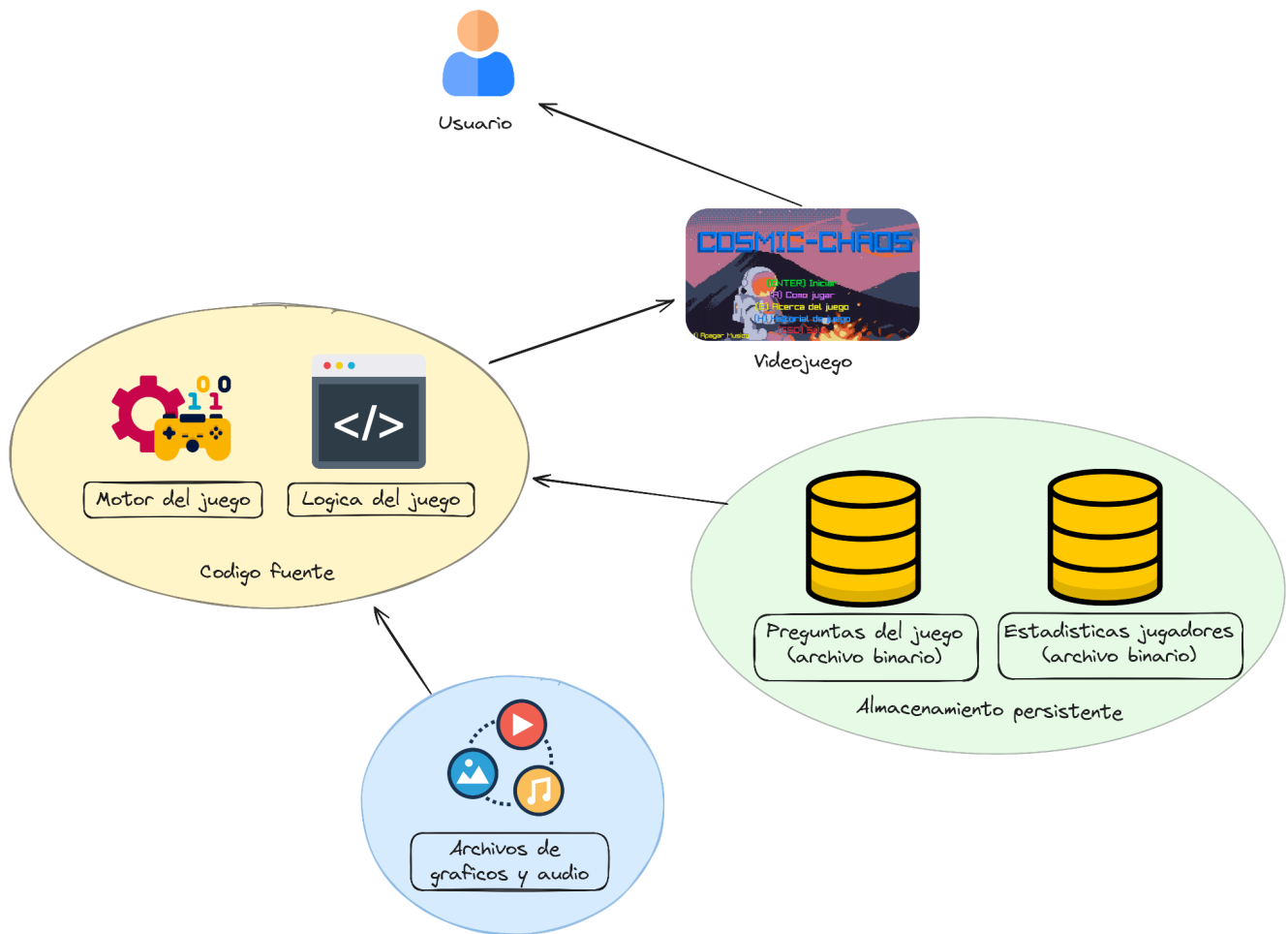
5- **Ajustes del Juego**

Opciones configurables que permiten personalizar la experiencia del usuario.

**Pantalla Completa:** Alternar entre modos de pantalla completa y ventana.

**Música:** Activar o desactivar la música de fondo del juego.

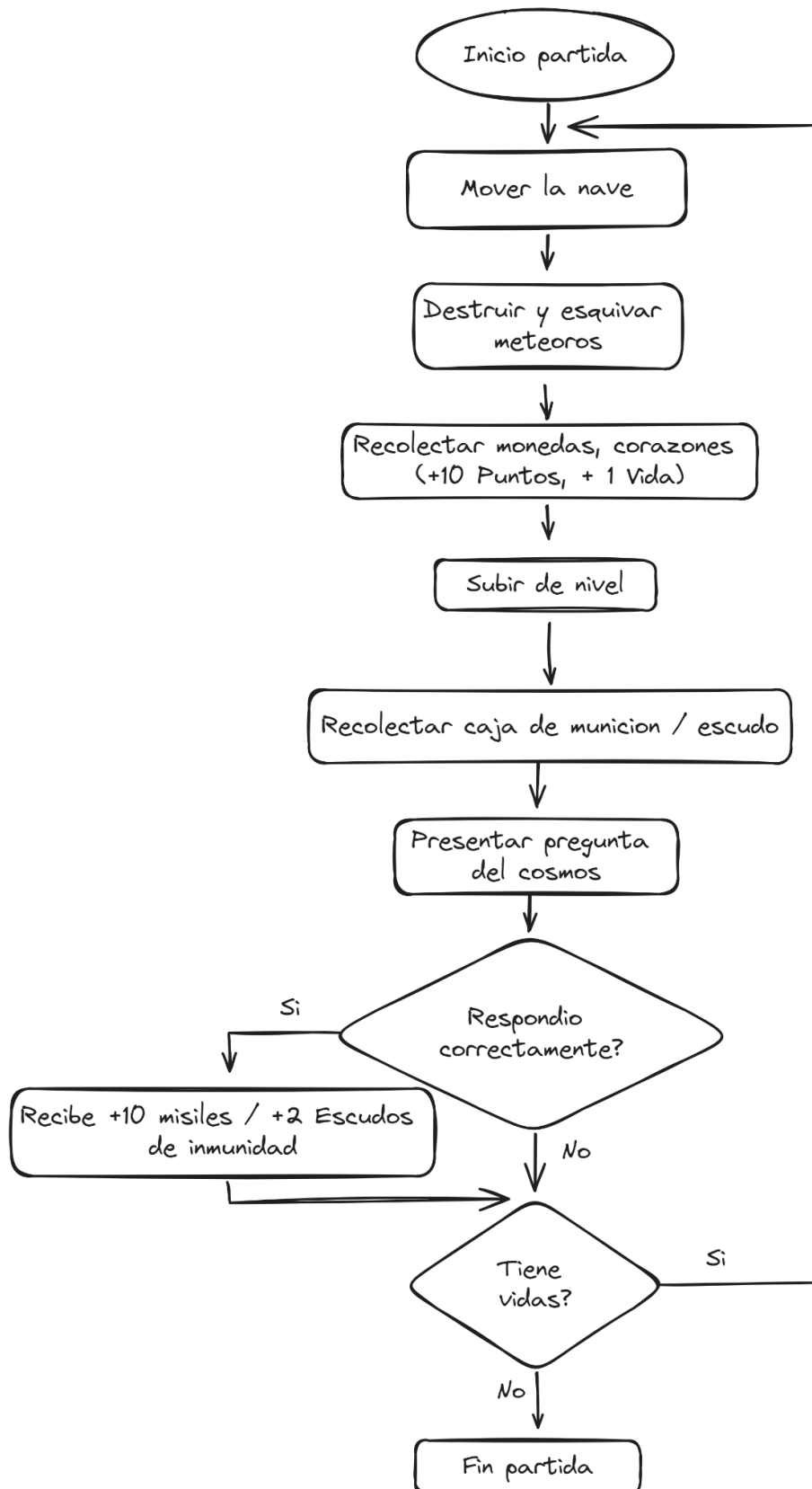
## 4.1 COMPONENTES DEL SOFTWARE





## 4.2 MODELO DE DISEÑO

El siguiente diagrama de flujo tiene como objetivo presentar de manera general el algoritmo de jugabilidad principal.



## 5. ASPECTOS TECNICOS

En esta sección, se proporcionará a los usuarios, desarrolladores y administradores información detallada sobre la arquitectura, los requisitos y el funcionamiento interno del software.

### 5.1. INSTRUCCIONES DE INSTALACIÓN

#### Windows.

1. Descargar instalador desde la pagina de releases del juego en github:  
<https://github.com/Dekstro999/COSMIC-CHAOS/releases>
2. Ejecutar el instalador y seguir instrucciones.
3. Ejecutar acceso directo generado en el escritorio.

#### Linux, Mac.

1. Seguir pasos de la sección 5.4

### 5.2 HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO

**Lenguaje de Programación:** El juego se desarrolló con lenguaje C. C es un lenguaje de programación de propósito general. Es uno de los lenguajes de programación más influyentes y ampliamente utilizados en la historia de la informática.

**Librería:** Se utilizó la librería Raylib para los gráficos del juego. Raylib es una biblioteca de desarrollo de videojuegos de código abierto y multiplataforma escrita en lenguaje C. Está diseñada para ser simple y fácil de usar, mientras proporciona un conjunto completo de características para el desarrollo de juegos 2D y 3D.

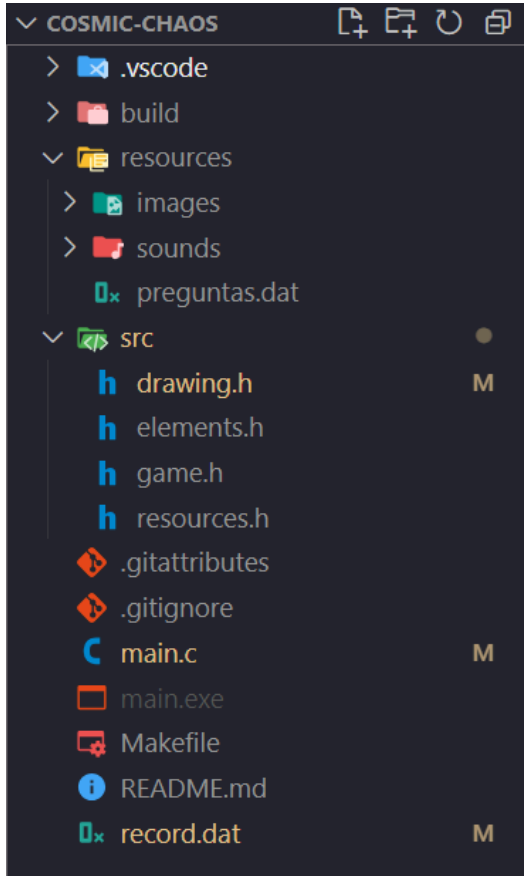
**Editor de código:** Se utilizó el editor Visual Studio Code para codificar el juego. Visual Studio Code es un editor de código fuente desarrollado por Microsoft que es muy popular entre los desarrolladores de software.

**Gráficos:** Utilizamos GIMP y PhotoShop para realizar algunas texturas y objetos del juego, y los fondos de niveles y menus los obtuvimos de la plataforma web ‘shutterstock’.

**Sonidos y Música:** Los sonidos y musica del juego, fueron obtenidos de la plataforma ‘fesliyanstudios’. La cual es una plataforma en línea que ofrece una amplia variedad de recursos de audio gratuitos y de alta calidad.

**Desarrollo colaborativo:** Se utilizó el sistema de control de versiones git, y la plataforma GitHub. Fue esencial para el manejo del código fuente y la colaboración entre los desarrolladores.

### 5.3 ESTRUCTURA DE ARCHIVOS DEL PROYECTO:



**build:** Archivos binarios necesarios para la compilación.  
**resources:** Imágenes y sonidos del juego, además de un archivo .dat que contiene las preguntas que se realizarán en las partidas.

**src:** archivos de encabezado:

- *drawing.h*: funciones de dibujo de interfaces y gráficos.
- *elements.h*: structs, enum y algunas macro.
- *game.h*: Lógica y motor del juego.
- *resources.h*: carga de texturas y sonidos.

**main.c:** archivo principal del juego.

**Makefile:** Serie de reglas y directivas utilizadas por la herramienta de compilación Make para construir y gestionar el proyecto.

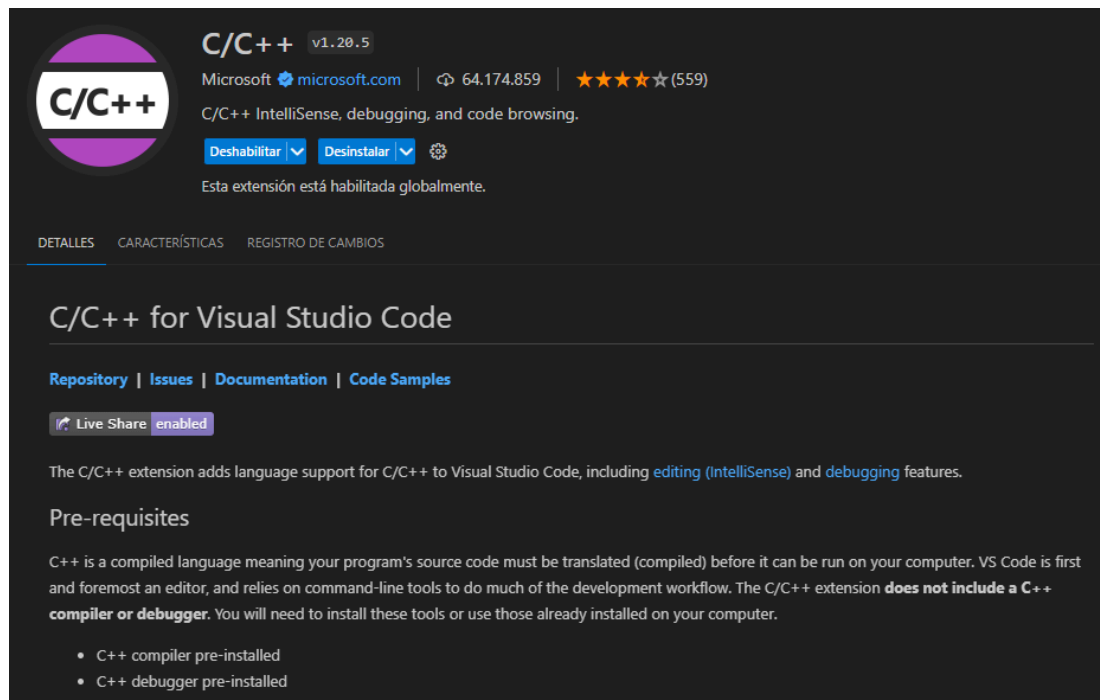
**record.dat:** Almacena historial de estadísticas de los jugadores.

### 5.4 MODIFICACIÓN LOCAL

Para modificar localmente el software, se necesitarán seguir las siguientes instrucciones:

- 1. Instalar Editor de código:** Necesitará un IDE o editor de código, el recomendado por nosotros es Visual Studio Code, puede descargarse de su pagina oficial: <https://code.visualstudio.com/download>
- 2. Instalar libreria Raylib:** Necesitará instalar la libreria con la que fue desarrollado el juego, puede descargar e instalar raylib desde su pagina oficial: <https://www.raylib.com/>.
- 3. Instalar compilador de C:** Para ejecutar el código necesita tener un compilador de C. Una opción para windows es Mingw64, para instalar, entrar al siguiente enlace y seguir las instrucciones: <https://www.msys2.org/>.
- 4. Instalar extension C/C++ para Visual Studio Code:**

Esta extensión proporciona herramientas de desarrollo indispensables para los lenguajes C/C++. Instalarla desde el marketplace de VSCode.



5. **Clonar o descargar repositorio:** Enlace al repositorio:  
<https://github.com/Dekstro999/COSMIC-CHAOS>
6. **Abrir carpeta del repositorio con Visual Studio Code:** En VSCode, ir a File > Open Folder > COSMIC-CHAOS.
7. **Compilar:** Abrir el archivo main.c, y ejecutar tarea de compilación, con el atajo ctrl + shift + b, o manualmente Terminal > Run Build Task.
8. **Ejecutar el juego:** Pulsar la combinación de teclas ctrl + f5 para ejecutar, o manualmente Run > Run without Debugging, o dando doble click al archivo ejecutable generado en el directorio.

## 6. ARQUITECTURA DEL SISTEMA

La arquitectura del sistema de Cosmic Chaos está diseñada para garantizar un rendimiento eficiente, escalabilidad y una experiencia de usuario fluida. La arquitectura se puede dividir en varias capas y componentes, cada una con funciones específicas y bien definidas.

### Frontend (Capa de Presentación):

- **Motor de Juego:** Controla la lógica de renderización de gráficos.
- **Interfaz de Usuario (UI):** Gestiona la visualización de elementos como menús y HUD.

### Backend (Capa de Lógica de Negocio):

- **Gestor de Eventos:** Maneja eventos como colisiones y respuestas a preguntas entre otros.
- **Módulo Educativo:** Genera y valida preguntas educativas.

### Persistencia de Datos:

- **Archivos binarios:** Almacena el historial de estadísticas de los jugadores y las preguntas a realizar en el juego.

### Integración y Recursos:

- **Recursos Gráficos y de Sonido:** Imágenes, sonidos y preguntas del juego.
- **Archivo Principal del Juego:** Coordinación de inicialización de componentes.
- **Herramientas de Compilación:** Makefile para compilar y construir el proyecto.

## 6.1 REQUISITOS DEL SISTEMA

### Hardware:

- **Memoria RAM:** 1GB.
- **Procesador:** Intel o AMD Gen 2, Dual-core.
- **Almacenamiento:** 30 MB de espacio en disco disponible.
- **Controles del juego:** Teclado estándar o Joystick.
- **Pantalla:** Resolución mínima de 1600x900 pixeles.

### Software:

- **Sistema operativo:** Windows, Linux o Mac.

# BIBLIOGRAFÍA

## Library

- Raylib in github : <https://github.com/raysan5/raylib>

## Software

- Pressman, Roger S. "Software Engineering: A Practitioner's Approach." McGraw-Hill Education, 2019.
- Sommerville, Ian. "Software Engineering." Pearson Education Limited, 2015.
- Pfleeger, Shari Lawrence, and Joanne M. Atlee. "Software Engineering: Theory and Practice." Pearson, 2013.

## Videogames

- Rollings, Andrew, y Ernest Adams. "Fundamentals of Game Design." Pearson Education, 2006.
- Adams, Ernest. "Break Into the Game Industry: How to Get a Job Making Video Games." McGraw-Hill Education, 2003.
- Schell, Jesse. "The Art of Game Design: A Book of Lenses." CRC Press, 2008.