Release 1: Estructura base y mecánicas principales

- Implementar las clases principales: **Jugador**, **Enemigo**, **Meteoro**, **Proyectil** y **PowerUp**.
- Programar métodos básicos como mover(), disparar(), y colisionar().
- Crear un bucle principal que permita el movimiento del jugador, la generación de meteoros y enemigos, y la detección de colisiones.
- Simular una partida básica sin interfaz gráfica avanzada, mostrando resultados por consola o en pantalla básica.

Release 2: Incorporación de habilidades y mejoras

- Desarrollar la lógica para **PowerUps** (como escudos, disparos múltiples, o velocidad extra).
- Implementar el sistema de **aura** (escudo temporal) y su duración.
- Agregar la lógica para el disparo múltiple (cañones adicionales) y su activación mediante la tienda.
- Integrar un sistema de puntuación y estadísticas básicas (meteoritos destruidos, esquivados, etc.).

Release 3: Escenario y ambientación

- Crear un fondo animado o estático para el espacio.
- Implementar la generación aleatoria de meteoros y enemigos con diferentes tamaños y velocidades.
- Asociar un escenario al juego, con lógica de ambientación (fondo, música, efectos de sonido).
- Agregar efectos visuales básicos, como explosiones al destruir meteoros o enemigos.

Release 4: Interfaz de usuario básica

- Implementar una interfaz que muestre:
    - **Barra de vidas** del jugador.
    - **Puntuación** y estadísticas (meteoritos destruidos, esquivados, tiempo jugado).

    - **Tesoros recolectados** y estado de habilidades activas (como el aura o el disparo múltiple).
- Mostrar mensajes en pantalla para eventos importantes (como "¡Aura activada!" o "¡Game Over!").

Release 5: Comportamiento de los enemigos

- Programar métodos para que los enemigos puedan:
    - **Moverse** de forma aleatoria o siguiendo patrones.
    - **Disparar** proyectiles hacia el jugador.
    - **Defenderse** o esquivar proyectiles del jugador (si aplica).
- Incorporar decisiones básicas de respuesta automática o aleatoria durante el combate.

Release 6: Cálculos de colisiones y finalización del juego

- Implementar la lógica para detectar colisiones entre:
    - **Proyectiles del jugador** y **enemigos/meteoritos**.
    - **Meteoros/enemigos** y el **jugador**.
- Ajustar las reglas para la pérdida de vidas y la activación de estados como invulnerabilidad temporal.
- Implementar la pantalla de **Game Over** con opciones para reiniciar o salir del juego.
- Agregar la lógica para determinar el final del juego y mostrar estadísticas finales.

Release 7: Pruebas y documentación

- Documentar todas las clases y métodos del juego.
- Realizar pruebas funcionales y unitarias para garantizar que las mecánicas principales funcionen correctamente.
- Preparar el entorno para despliegue o presentación del producto.
