

# Juego de Disparos

**Deler Santiago Mendez Mendez**

Ingeniería Electrónica y Telecomunicaciones, Popayán, Colombia, delermendez@unicauca.edu.co

## Resumen

Se realizó un juego para arduino de dos formas diferentes, ambas usan archivos .h donde están los sprites que son usados para visualizar los personajes que son parte del juego.

Las formas fueron:

- **Versión estructurada:** Toda la lógica del juego se encuentra en un único archivo.
- **Versión orientada a objetos (POO):** La lógica se divide en clases, lo que permite mayor escalabilidad por lo que está dividido en varios archivos.

Se logró de forma exitosa el funcionamiento del juego usando las dos formas. La programación orientada a objetos permite agregar de manera más eficiente nuevas funcionalidades por lo que es más útil si se necesitara añadir nuevas cosas al juego, el main unificado permite una lectura más sencilla pero se queda corto si necesita meter más cosas.

## 1. MARCO TEÓRICO

La Programación Orientada a Objetos (POO) es un paradigma de programación, es decir, un modelo o un estilo de programación que nos da unas guías sobre cómo trabajar con él. Se basa en el concepto de clases y objetos.

El uso de archivos .h para definir sprites es una práctica frecuente en el desarrollo de videojuegos en entornos limitados como Arduino. Estos archivos contienen arreglos de bits o matrices que representan visualmente a los personajes o elementos del juego, y se integran con bibliotecas gráficas como Adafruit GFX para ser mostrados en pantallas TFT.

## 2. DESARROLLO

### Diseño del juego

Se desarrollaron dos versiones del juego: una utilizando un enfoque estructurado y otra aplicando programación orientada a objetos. La primera tiene toda la lógica del juego en un único archivo principal, facilitando una visión del flujo de ejecución, pero limitando la escalabilidad. La segunda versión divide el sistema en clases, permitiendo una mayor claridad de funciones específicas, como el movimiento del jugador, la generación de enemigos y el manejo de colisiones.

### ***Ventajas de cada versión***

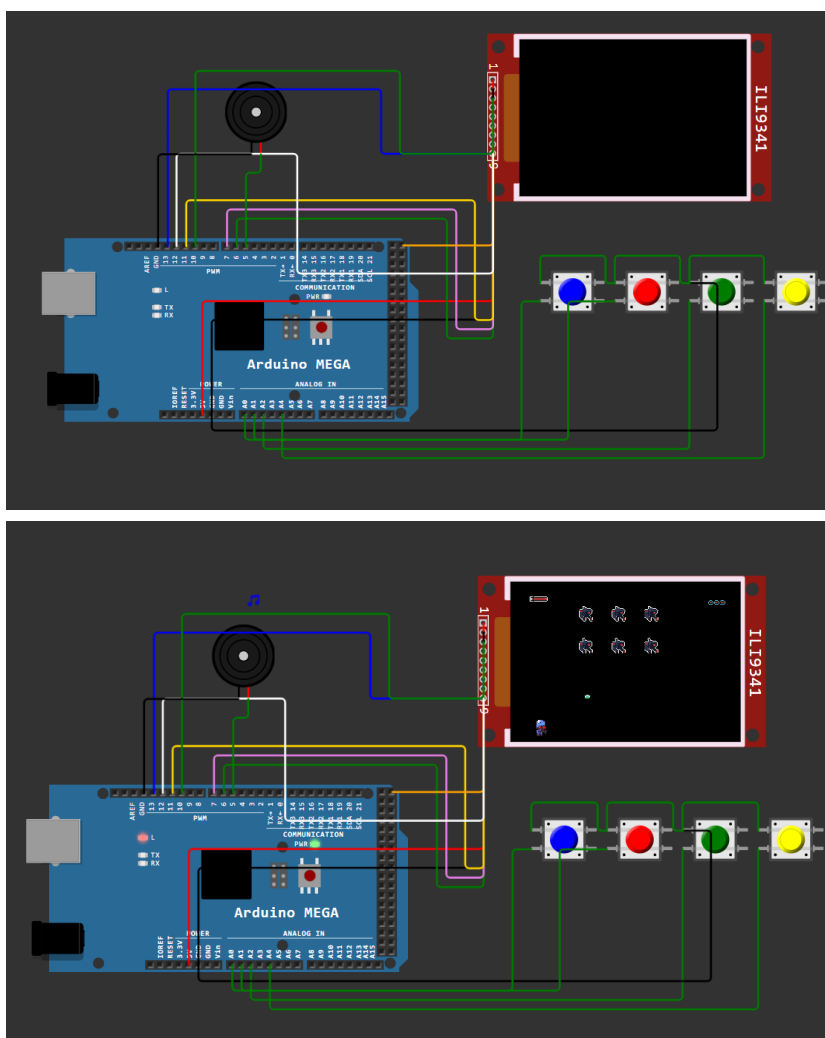
La versión estructurada permite una rápida implementación y comprensión. Sin embargo, se vuelve difícil de mantener en proyectos extensos. La versión orientada a objetos, requiere una planificación inicial más detallada, facilita la gestión del código conforme se agregan nuevas funcionalidades.

### **Implementación**

Se implementó un videojuego para Arduino utilizando una pantalla TFT ILI9341, un buzzer y botones para la interacción. El juego incluye personajes visualizados mediante sprites definidos en archivos .h, además se tuvo en cuenta las dos versiones para observar su funcionamiento.

Ambas versiones fueron programadas en Vs code usando librerías de arduino. Se utilizó la biblioteca Adafruit GFX para el manejo gráfico, y los sprites fueron diseñados en una herramienta llamada “LibreSprite” y cargados como arreglos. La interacción se logró mediante la lectura de botones.

Ambas versiones ofrecen un desempeño fluido en la pantalla. En las siguientes imágenes se muestra el circuito en Wokwi e igualmente se muestra el funcionamiento del juego.



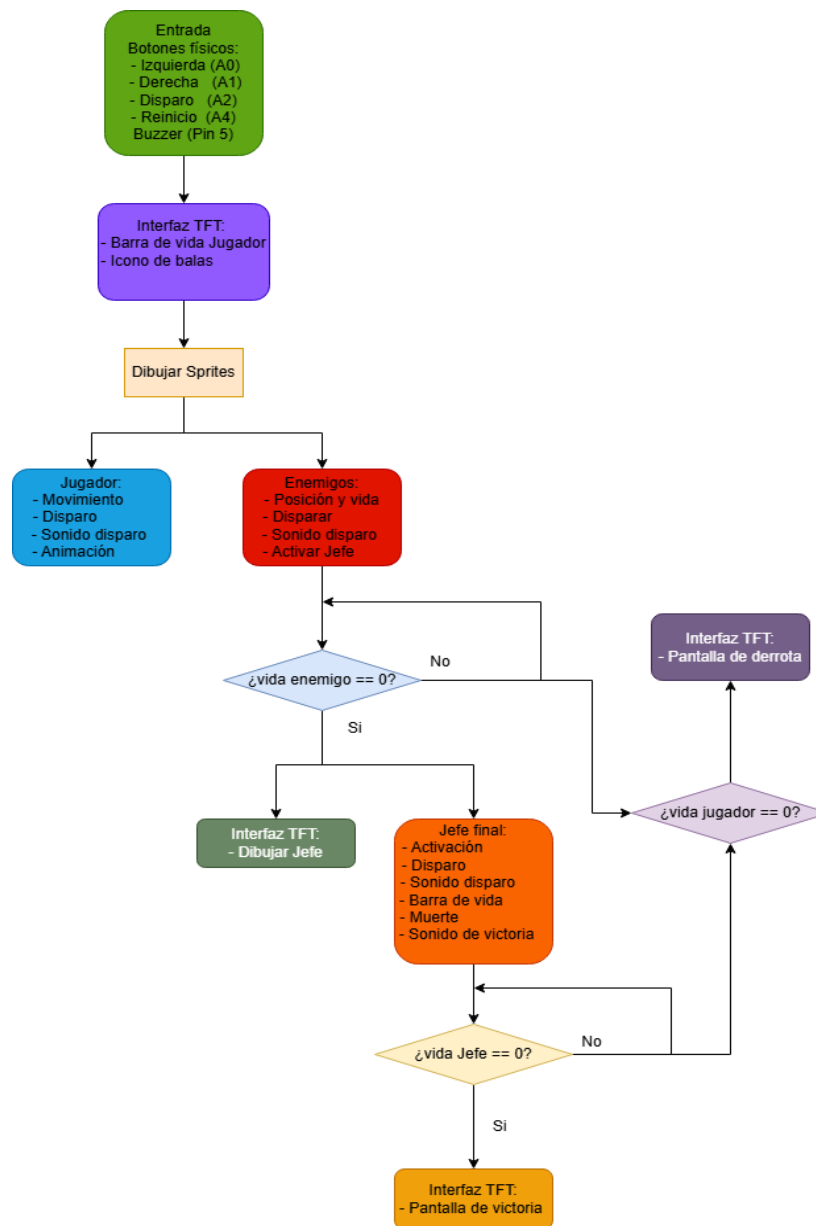


Figura 1. Diagrama en un solo main

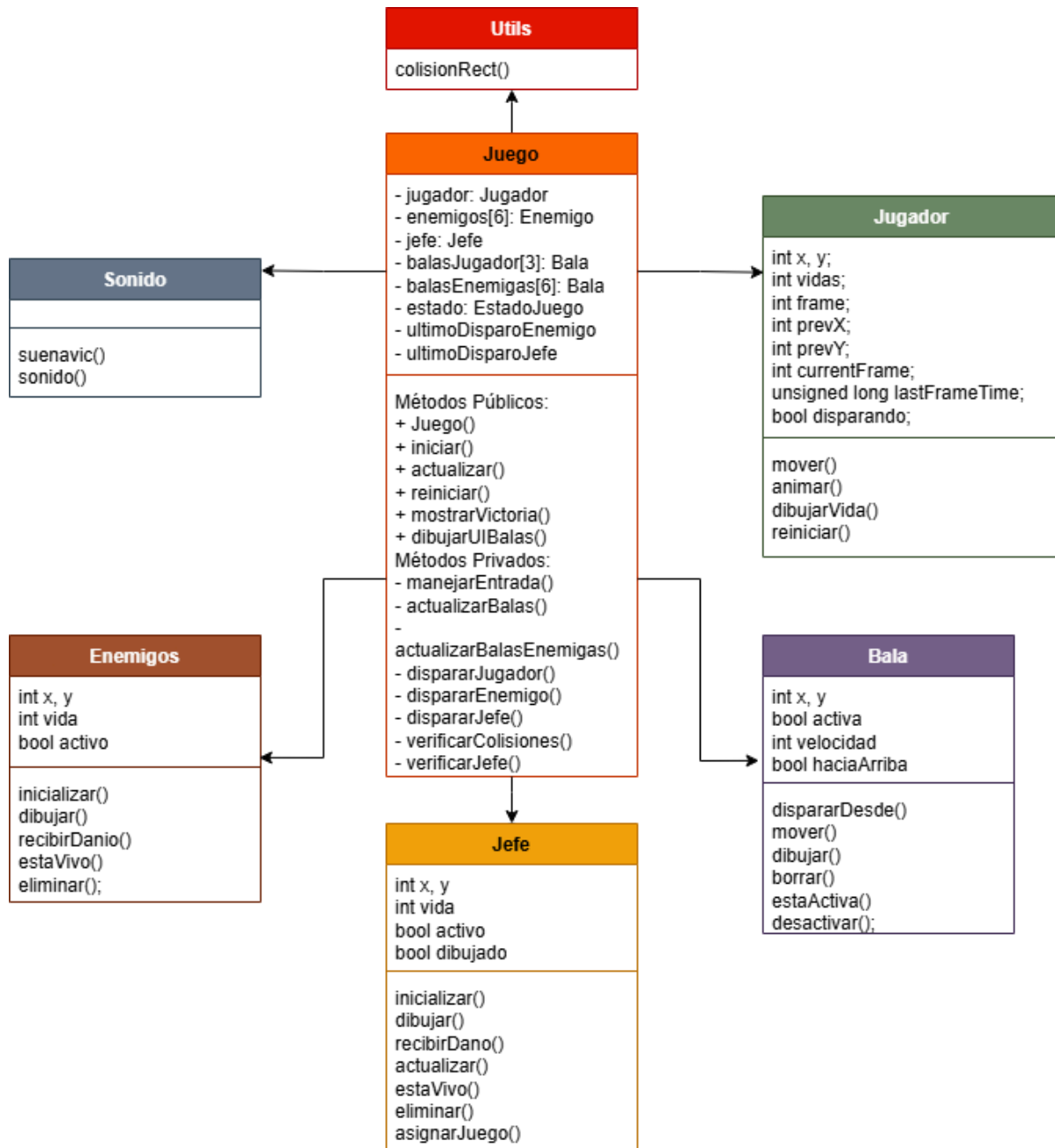


Figura 2. Diagrama en POO

### 3. CONCLUSIONES

Ambas versiones de desarrollo permitieron cumplir con el objetivo del proyecto que era hacer un juego funcional. La versión estructurada es ideal para proyectos pequeños que no necesiten de mucha escalabilidad, mientras que la orientada a objetos es más recomendable en proyectos complejos que requieran crecimiento a largo plazo, además, la creación de clases permite que se pueda escalar de forma más sencilla los objetos que utilizaremos en el juego.

### REFERENCIAS

Phillip Burgess. (2013). *Adafruit GFX Graphics Library*. Adafruit.

González, A. E. C. (2004). Programación orientada a objetos. *Centro de Investigación y de Estudios Avanzados*.