

La Evolución de un Clásico: Un Proyecto de Pong en Tres Partes

De la Consola a los Gráficos en C y C++

Javier Iparraguirre

Universidad Tecnológica Nacional
11 de abril 461, Bahía Blanca, Argentina
jiparraguirre@frbb.utn.edu.ar
<http://www.frbb.utn.edu.ar/>

14 de octubre de 2025

Pong

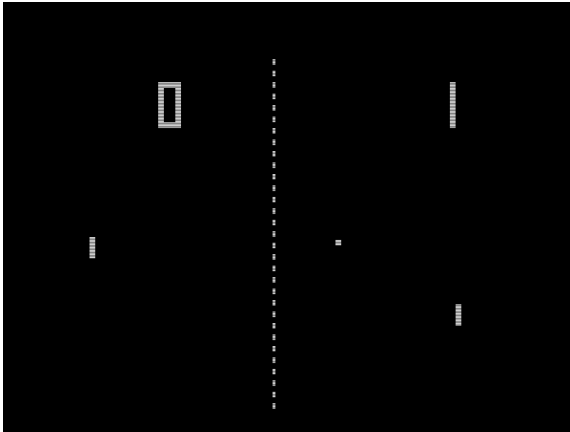


Figura: Juego clásico de Pong

Pong



Resumen del Proyecto

- Esta presentación explora tres implementaciones del clásico juego Pong [Wik23], cada una demostrando un nivel diferente de complejidad en la programación.
- Viajaremos desde una simple aplicación de consola multiplataforma, pasando por una versión con una estructura de proyecto profesional, hasta llegar a un juego gráfico completo usando la librería SDL3.

Implementación 1: La Base - Pong en Consola

- **Tecnología:** Escrito completamente en C estándar [lpa23a].
- **Multiplataforma:** Compatible con Windows, Linux, y macOS [lpa23a].
- **Gráficos:** Utiliza caracteres ASCII y códigos de escape ANSI para renderizar en la terminal [lpa23a].
- **Jugabilidad:** Un juego funcional de Jugador vs. IA de Pong.

Implementación 1: Fragmento de Código

```
1 // Dibuja todo el juego
2 void draw_game(void)
3 {
4     int i, j;
5
6     gotoxy(0, 0);
7
8     // Dibuja el borde superior
9     for (i = 0; i < FIELD_WIDTH; i++)
10     {
11         printf("#");
12     }
13     printf("\n");
14
15     // ... (resto de la logica de dibujo)
16 }
```

Implementación 2: Estructura Profesional

- **Mejoras:** Construido sobre la primera implementación.
- **Entorno de Desarrollo:** Incluye un proyecto completo de Visual Studio (`.vcxproj`, `.sln`) [lpa23d, lpa23c].
- **Código Base:** El mismo código C multiplataforma, pero ahora estructurado para un IDE profesional [lpa23b].

Implementación 2: ¿Por qué Visual Studio?

- **Depuración:** Herramientas de depuración avanzadas para recorrer el código e inspeccionar variables.
- **Gestión de Proyectos:** Simplifica la gestión de archivos fuente, dependencias y configuraciones de compilación.
- **IntelliSense:** Completado de código y características de navegación mejoradas.

Implementación 3: El Salto a Gráficos - SDL3

- **Lenguaje:** Transición de C a C++.
- **Librería Gráfica:** Utiliza la potente librería SDL3 para renderizado 2D.
- **Visuales:** Gráficos fluidos y acelerados por hardware reemplazan la interfaz ASCII.
- **Características Avanzadas:** Introduce conceptos más complejos como sistemas de tiempo real y una IA mejorada.

Implementación 3: Avances Técnicos

- **Principios Orientados a Objetos:** El juego se estructura en torno a las clases `GameObject` y `GameState` [lpa23e].
- **Programación Orientada a Eventos:** El bucle del juego ahora es impulsado por eventos de SDL para la entrada y gestión de la ventana [lpa23e].
- **Renderizado:** El renderizador de SDL se utiliza para dibujar formas y gestionar la ventana del juego [lpa23e].
- **IA:** La IA es más receptiva e inteligente, con su propia lógica de actualización [lpa23e].

```
1 // Esta funcion se ejecuta una vez por frame ,
2 // y es el corazon del programa.
3 SDL_AppResult SDL_AppIterate(void *appstate)
4 {
5     // ... (logica del juego y renderizado)
6 }
```

Comparando las Implementaciones

Característica	01-simple	02-simple-vs	03-simple-sdl
Lenguaje	C	C	C++
Gráficos	ASCII	ASCII	SDL3
Plataforma	Multi-plataforma	Multi-plataforma	Windows (Config.)
Desarrollo	Editor Básico	Visual Studio	Visual Studio
Complejidad	Baja	Media	Alta

Cuadro: Una comparación lado a lado

Progresión del Aprendizaje

- 01-simple: Domina los fundamentos de la programación en C y el desarrollo multiplataforma.
- 02-simple-visual-studio: Aprende prácticas de desarrollo profesional y los beneficios de un IDE.
- 03-simple-visual-studio-sdl: Sumérgete en el mundo de la programación gráfica, C++ y el desarrollo de juegos moderno.

¡Gracias!

¿Preguntas?

`jiparraguirre@frbb.utn.edu.ar`

Referencias I

- [lpa23a] Javier Iparraguirre.
Cross-platform pong game in c, 2023.
Source Code: [javierip/structured-programming-c/structured-programming-C-134cf6fa8d65f66039f07aac2d0003c7a80ad316/project-game-pong/01-simple/main.c](https://github.com/javierip/structured-programming-c/structured-programming-C-134cf6fa8d65f66039f07aac2d0003c7a80ad316/project-game-pong/01-simple/main.c).
- [lpa23b] Javier Iparraguirre.
main.c for visual studio, 2023.
Source Code: [javierip/structured-programming-c/structured-programming-C-134cf6fa8d65f66039f07aac2d0003c7a80ad316/project-game-pong/02-simple-visual-studio/main.c](https://github.com/javierip/structured-programming-c/structured-programming-C-134cf6fa8d65f66039f07aac2d0003c7a80ad316/project-game-pong/02-simple-visual-studio/main.c).

Referencias II

- [Ipa23c] Javier Iparraguirre.
Ponggame.sln, 2023.
Source Code: [javierip/structured-programming-c/structured-programming-C-134cf6fa8d65f66039f07aac2d0003c7a80ad316/project-game-pong/02-simple-visual-studio/PongGame.sln](https://github.com/javierip/structured-programming-c/structured-programming-C-134cf6fa8d65f66039f07aac2d0003c7a80ad316/project-game-pong/02-simple-visual-studio/PongGame.sln).
- [Ipa23d] Javier Iparraguirre.
Ponggame.vcxproj, 2023.
Source Code: [javierip/structured-programming-c/structured-programming-C-134cf6fa8d65f66039f07aac2d0003c7a80ad316/project-game-pong/02-simple-visual-studio/PongGame.vcxproj](https://github.com/javierip/structured-programming-c/structured-programming-C-134cf6fa8d65f66039f07aac2d0003c7a80ad316/project-game-pong/02-simple-visual-studio/PongGame.vcxproj).

Referencias III

- [Ipa23e] Javier Iparraguirre.
Simple pong sdl, 2023.
Source Code: [javierip/structured-programming-c/structured-programming-C-134cf6fa8d65f66039f07aac2d0003c7a80ad316/project-game-pong/03-simple-visual-studio-sdl/simplePongSDL/main.cpp](https://github.com/javierip/structured-programming-c/structured-programming-C-134cf6fa8d65f66039f07aac2d0003c7a80ad316/project-game-pong/03-simple-visual-studio-sdl/simplePongSDL/main.cpp).
- [Wik23] Wikipedia.
Pong — Wikipedia, la enciclopedia libre, 2023.
[Online; consultado 14-octubre-2025].