



Universidad Nacional
Autónoma de México
Facultad de Ingeniería



Proyecto Final:

Juego PONJ.



Nombre: Ordiales Caballero, Iñaky.

Cuenta: 317559897.

Asignatura: Estructura y Programación de
Computadoras (1503).

Grupo: 02.

Semestre: 2021 – 2.

Profesor: M.I. Luis Sergio Durán Arenas.

Fecha de entrega:

16 de agosto del 2021.

Introducción

El proyecto tiene como objetivo el demostrar la aplicación de los conocimientos práctico-teóricos de la programación en lenguaje ensamblador, los cuales fueron adquiridos a lo largo del curso. Para esto el profesor decidió que su desarrollo consistiría en la elaboración de un videojuego que replicara el primer videojuego de todos, Pong™.

Para la elaboración del problema se nos solicitó que cumpliera con las siguientes características. El programa debía ser codificado en lenguaje ensamblador y para la arquitectura Intel x86. Su funcionamiento general debía consistir de la impresión de una interfaz gráfica en pantalla con la cual el usuario pudiera interactuar mediante el ratón y el teclado. La interfaz tenía que mostrar en pantalla la información básica del juego Pong™, es decir:

- Los marcadores correspondientes para cada jugador.
- Un temporizador que muestre el tiempo restante en formato M:SS donde M son los minutos y SS los segundos (0 – 59). El temporizado arrancará en 2:00 e irá en reversa hasta llegar a 0:00.
- Dos barras controladas a través del teclado, representando a cada uno de los jugadores. Cada barra se moverá mediante dos teclas específicas, una para subir y la otra para bajar.
- Una pelota que rebotara dentro del área delimitada para el juego y según las condiciones que encuentre.
- Botones de inicio y terminación o reinicio del juego que serán pulsados mediante el uso del ratón. Éste último debe estar delimitado fuera del área de juego, en ningún momento interfiriendo con la vista del juego.
- Para finalizar el programa, el usuario debe hacerlo a través de la interfaz gráfica.

Mediante las características antes mencionadas, el usuario deberá poder jugar una simulación del juego Pong™. El juego puede verse como una simulación del tenis, pero con bordes laterales. Se tiene una pelota en un área rectangular, la pelota puede rebotar en los límites horizontales que son cerrados. Pero los límites verticales están abiertos y deben ser protegidos por los jugadores para evitar que la pelota salga por éstos, su “portería”. Cada jugador tiene una barra vertical que podrán mover a lo alto y contra las cuales la pelota rebotará cambiando su recorrido hacia la “portería” del otro jugador. Si la pelota rebasa horizontalmente a cualquiera de las barras de los jugadores, será un punto para el jugador que no fue rebasado. El ganador será quién haya acumulado más puntos al final del tiempo.

Desarrollo

Para la solución del problema planteado, lo primero que supe que quería hacer desde un principio era tener varias interfaces gráficas decoradas con las cuales el usuario pudiera seleccionar las diferentes opciones que pensaba agregarle al programa. Esto me daba la sensación de que al final se vería un programa más real, a que si sólo se implementara el juego.

El proceso que se siguió a lo largo del desarrollo fue el siguiente:

Lo primero que hice fue analizar el archivo base que se nos fue proporcionado por el profesor. Tras revisar línea por línea pude obtener una muy buena idea de lo que se podía hacer para toda la parte gráfica. Luego revisé algunos ejemplos también proporcionados por el profesor para entender cómo podría implementar otros funcionamientos como el cronometro o la lectura del ratón. Ya que tenía toda esta información, decidí empezar a desarrollar mi programa por partes. Creaba una parte en un archivo auxiliar y luego la agregaba al proyecto completo. Primero hice la interfaz y el funcionamiento para seleccionar entre individual o dos jugadores, luego la interfaz y el funcionamiento para elegir entre por tiempo o por puntos. Y finalmente la interfaz del juego y el funcionamiento de sus botones.

Ya que tenía toda la parte externa a los movimientos del juego, ahora tuve que empezar a pelearme con lo más complicado. Uno de mis mayores problemas fue el de hacer que el programa siguiera corriendo, es decir el cronometro avanzara, mientras estaba esperando una tecla para mover a los jugadores. Esto me costó mucho averiguarlo, pero finalmente lo pude hacer al checar si el buffer del teclado estaba vacío antes de checar por una tecla. Al resolver esto sólo fue establecer las teclas y las lógicas de los movimientos limitado de los jugadores. Después tuve que trabajar en la pelota y los rebotes. Decidí que dependiendo del punto de la barra del jugador contra el que chocara la pelota, el rebote que tendría. Entonces la pelota tenía varios tipos de movimiento, con inclinaciones, sentidos y direcciones diferentes. Al establecer todo esto, ya lo que me faltaba era programar los goles, el fin del juego y algo muy importante que fue la velocidad de movimiento de jugadores y de pelota. Estos últimos no debían depender entre ellos, es decir la velocidad de movimiento de un jugador y la de la pelota eran diferentes.

Al ya tener todo integrado, me aparecieron algunos errores con los rebotes de la pelota y los movimientos de los jugadores. A veces la pelota rebotaba y se salía del área de juego, o atravesaba y borraba parte de los jugadores. Al checar mis condiciones para el movimiento de la pelota y de los jugadores, lo pude solucionar con un poco de tiempo de trabajo. Y de esta manera quedó realizado el programa, con varias interfaces y tipos de juego.

Además de las características requeridas en las especificaciones del proyecto, decidí agregar algunas de las funcionalidades extras sugeridas en el documento. Pero también agregué algunas que se me ocurrieron por mi cuenta. Todas estas funcionalidades añadidas son:

➤ **Sugeridas:**

- ✓ Modificar la pendiente y la velocidad de rebote de la pelota dependiendo la zona de rebote en un jugador.
- ✓ Una pantalla que muestre el resultado cuando finalice el juego, indicando al ganador y con la opción de volver a la pantalla de inicio e iniciar un juego nuevo.
- ✓ Que uno de los jugadores sea controlado por la CPU (sin que sea imposible ganarle).

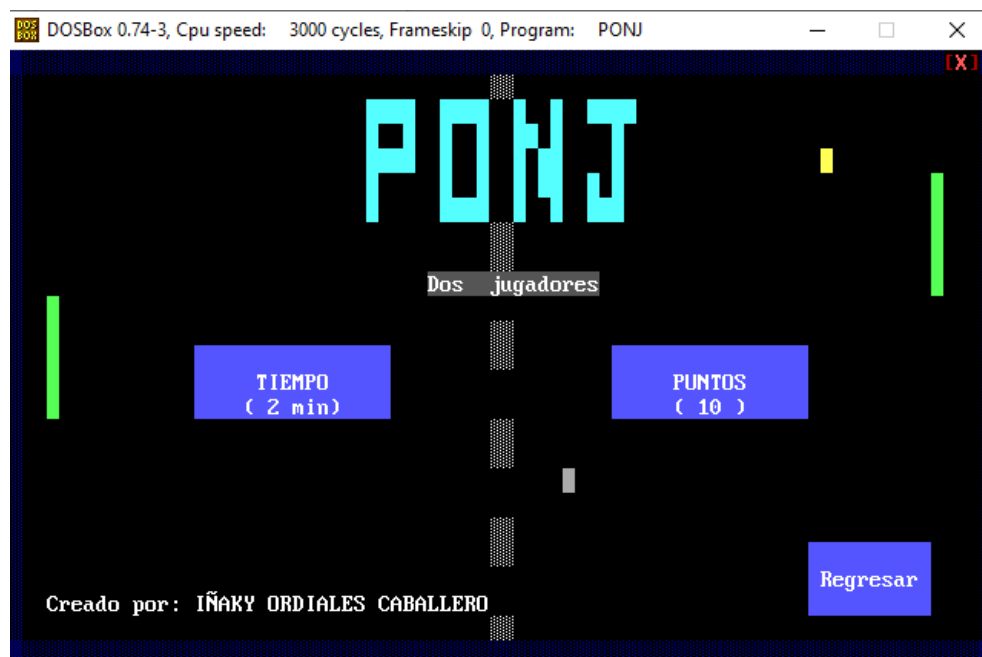
➤ **Propias:**

- ✓ Modo de juego por puntos
- ✓ Varias interfaces para la selección de opciones.

En un principio pensé en realizar el programa en modo gráfico, pero al analizarlo me di cuenta de que la dificultad se incrementaba demasiado. Por esto al final escogí hacerlo en modo texto con una resolución de 80x25 pixeles. Conociendo esto, las diferentes interfaces gráficas las fui diseñando con ayuda de hojas de cálculo, donde les daba la forma de rectángulos chicos parados a las celdas y las agrupaba en 80x25. Esto me permitió poder visualizar y calcular mejor todo lo gráfico del proyecto, tanto para la decoración como para los límites del juego mismo.

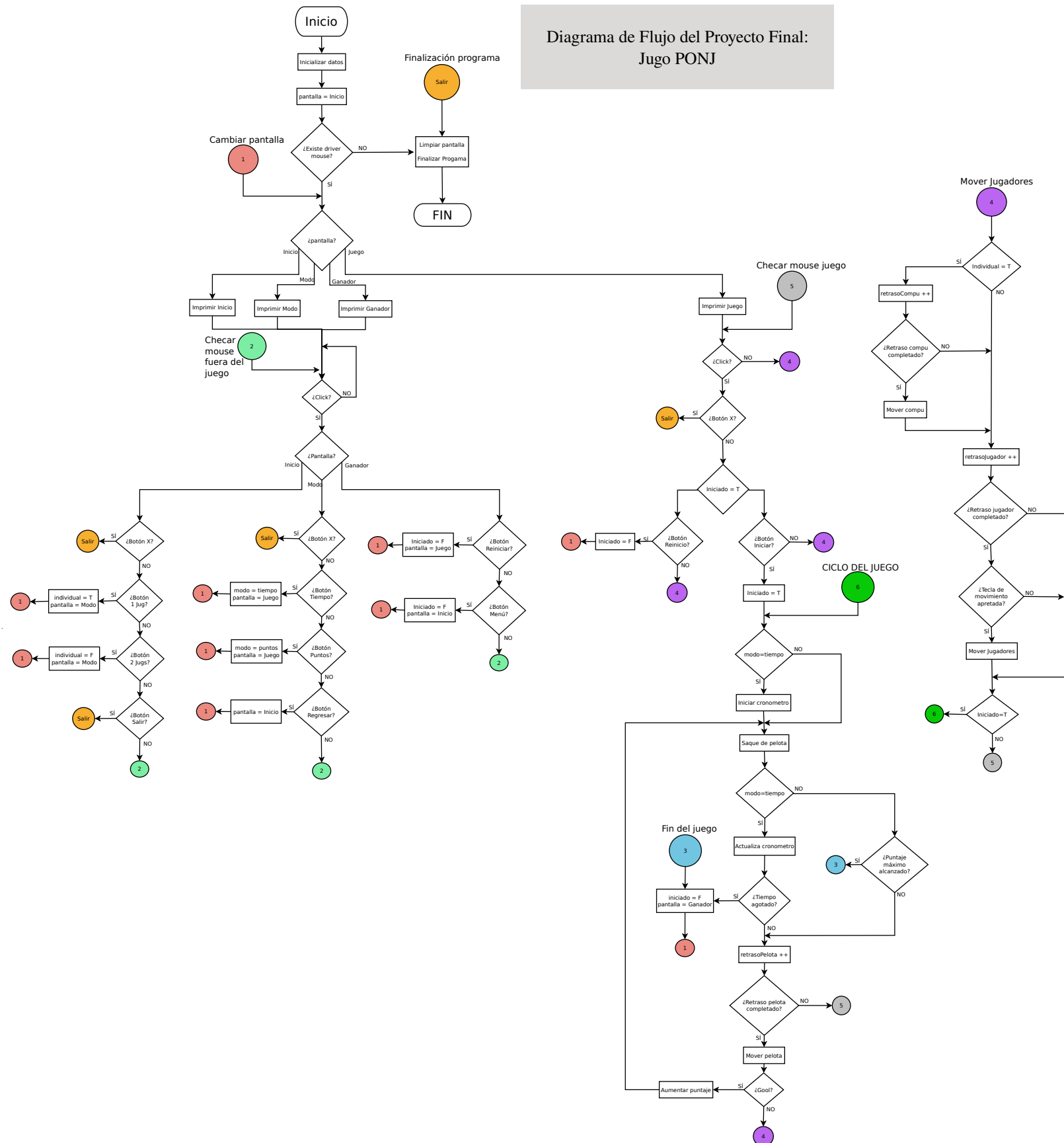
A continuación, se muestran las principales interfaces del programa.





En la siguiente página podemos ver un diagrama de flujo general sobre el funcionamiento del programa solución.

Diagrama de Flujo del Proyecto Final: Jugo PONJ



Conclusión

La realización del programa fue elaborada de manera exitosa. Y a través de ella se lograron cumplir los objetivos del proyecto, ya que efectivamente pude poner a prueba y demostrar mis conocimientos de desarrollo en lenguaje ensamblador para arquitectura Intel x86.

A lo largo del trabajo que implicó la creación de un programa que simulara el juego Pong™, hubo muchas dificultades y problemas que tuve que resolver. Creo que el haberlo empezado con bastante tiempo de anticipación ayudó a no sentirme presionado y poder trabajar con más gusto. Sin embargo, sí hubo momentos en los que ya llevaba un par de horas trabajando en la misma parte y sentía nunca iba a encontrar la solución. Pero siempre acabé encontrándola. Algo que me gustó mucho y me parece relevante de mencionar es que para poder hacer muchas de las cosas solicitadas, tuve que investigar por mi cuenta diferentes interrupciones e instrucciones. Esto me hizo aprender mucho en el proceso. Si puedo decir que cualquier cosa que no me hubiera quedado del todo claro en las clases de teoría, al realizar este programa las pude entender bien e implementar. Además, que el pensar que se está trabajando con el lenguaje más cercano a código binario (objeto) da un sentido de satisfacción y mayor comprensión de las computadoras.

Creo que el programa que realicé cumple y rebasa las expectativas del proyecto. Funciona bien y sin errores, además de que lo considero atractivo visualmente. Por lo que me siento satisfecho con haber hecho un buen trabajo final.