*Instituto Tecnológico de Costa Rica*

*Unidad de Computación*

***Gato: Tres en línea.***

*Samantha Acuña Montero*

*Katherine Amador González*

*Joshua Esteban Sancho Burgos*

*Sede San Carlos*

*18 de junio del 2021*

Indices

[Introducción 2](#_Toc74931584)

[Análisis del problema 4](#_Toc74931585)

[Solución del problema 5](#_Toc74931586)

[Análisis de resultados 8](#_Toc74931587)

[Conclusiones 10](#_Toc74931588)

[Recomendaciones 11](#_Toc74931589)

[Bibliografía 12](#_Toc74931590)

# Introducción

En este proyecto se propone elaborar un juego de estrategia, en este caso el juego llamado “Gato” o “Tres en línea”, el cual consiste en un enfrentamiento de jugador a jugador en el que se turnan en una matriz de 3x3 para colocar un símbolo, ya sea una “X” o un octágono. La selección de la posición de este signo es una decisión propia del jugador, este puede digitarlo donde sea, mientras el campo seleccionado no esté ocupado por el otro jugador o se encuentre fuera de la matriz. El jugador que logre conectar una línea horizontal, vertical o diagonal, gana la ronda, y para que el jugador gane la partida, debe de ganar un mínimo de tres partidas u obtener 50 puntos o más. Este proyecto se desarrollará por medio del lenguaje Ensamblador que es un lenguaje de programación de bajo nivel, por ende, el proyecto se realizará con el fin de poner en práctica el conocimiento adquirido en clases, además de desarrollar el aprendizaje y la lógica de programar con el lenguaje Ensamblador por medio del programa “emu8086”. También, se realizará un refuerzo de estos conocimientos, por medio de investigaciones sobre el tema, efectuadas en libros o artículos. Por lo tanto, es importante recalcar que el lenguaje Ensamblador, es un lenguaje muy antiguo y como se mencionó anteriormente de bajo nivel, como lo indica Ocaña (2020):

El lenguaje que utilizaban los primeros programadores era el binario o lenguaje de máquina, pero a medida que iba avanzando la tecnología en aquellos tiempos, se necesitaban programas más fáciles de entender y programar, de ahí nace el lenguaje Ensamblador. (p.325)

Por consiguiente, se utilizará el lenguaje Ensamblador para la elaboración del proyecto, no solo por finalidades del curso, sino que también, porque este es un complemento importante para el desarrollo de un programador, permitiendo que este conozca mejor el lenguaje máquina, como lo indica Ocaña (2020): “El lenguaje Ensamblador convierte el programa en Lenguaje de Máquina, radica en asociar a los códigos de operación denominado Optocodes, en palabras claves que faciliten su uso por parte del programador”. (p.325) Este documento está organizado de la siguiente forma: en la página 2, se encuentra la introducción, en la página 4 se ubica el análisis del problema , donde se presenta la estructura analizada del problema presentado para la elaboración del proyecto, en la página 5 se presenta la solución del problema, donde se explica el desarrollo de la solución para el problema analizado anteriormente, en la página 8 se presenta el análisis de resultados, donde se indica el estado del proyecto y sus observaciones, en la página 10 se presentan las conclusiones, dando a conocer los resultados obtenidos en el proyecto, en la página 11 se encuentran las recomendaciones, donde se da un punto de vista útil basado en la experiencia de trabajar en el proyecto y en la página 12 se encuentra la bibliografía, donde se recopilan las fuentes de las cuáles se recabó información.

# Análisis del problema

En este proyecto se desarrollará un juego llamado “Gato” o “Tres en línea”. Para poder desarrollarlo correctamente, al iniciar el juego, este debe de imprimir el nombre del juego y los créditos de los desarrolladores, seguido de la impresión de las reglas y, por último, los jugadores deben de introducir un “nickname” de tres caracteres para ser representados en el juego. Además, para que este sea jugable, es necesaria la participación de dos jugadores, donde cada uno tendrá un turno de por medio para poder colocar el signo correspondiente en el lugar que lo desee, sin embargo, si alguno de los dos jugadores coloca el signo en una posición ocupada o que este fuera de la matriz jugable no se colocará el signo, sonará un sonido de advertencia y automáticamente se le restarán los puntos correspondientes su puntaje total (1 si oprimió una zona fuera de la matriz o 10 si oprimió un área ocupada). Por otro lado, si alguno de los jugadores se equivoca y da un click derecho, se le restarán 5 puntos de su puntaje total. Se considera que para que algún jugador gane una ronda, este deberá de formar una línea vertical, horizontal o diagonal, dentro de la matriz cuadriculada del juego. Si el jugador realiza alguna de estas, se le sumarán 20 puntos a su puntaje total y automáticamente la matriz se reiniciará para comenzar otra partida, el juego termina cuando alguno de los jugadores llega a obtener 50 puntos o más. Además, este presentará tres botones con las opciones respectivas de rendirse, reiniciar y salir, y los jugadores podrán presionar cualquiera de estos durante el desarrollo de la partida. Se requiere el uso del emulador “emu8086” para la correcta ejecución del juego, y también es necesario el uso correcto y específico de los registros, operadores e interrupciones, con tal de garantizar el correcto desarrollo del mismo.

# Solución del problema

El problema se solucionó, por medio de la programación de instrucciones en el “emu8086”, donde primeramente se desarrollaron las variables que dan forma al logo del juego. Este se realizó por medio de la impresión de valores ascii en pantalla, formando así, mediante “ascii art” el logo deseado. Luego se programaron las variables que contienen los valores de los créditos de los autores del juego y las instrucciones. Este es el menú de inicio del juego, donde los jugadores pueden pasar de la impresión del logo y créditos a las instrucciones y luego al juego como tal, oprimiendo la tecla “enter”. Es importante mencionar que antes de comenzar el juego como tal, los jugadores deben introducir un “nickname” de tres caracteres con el cuál deseen ser representados en el juego, esto fue realizado por medio de la interrupción 21h con servicio 01h, que permite la entrada de caracteres con eco. Además, se realizaron macros para realizar saltos de línea, imprimir mensajes, imprimir textos especiales (Como el logo y las reglas), imprimir símbolos, etc. Esto se realizó para que la estructura del código fuera más ordenada y fácil de manipular a la hora de colocar o imprimir estructuras específicas en pantalla. De la misma forma, para realizar la estructura gráfica del juego, se utilizó la interrupción 10h con servicio 12h para entrar al modo de video con resolución de 640x480px y 16 colores disponibles, se dibujó la matriz con cuatro columnas y cuatro filas, dando así nueve espacios para que los jugadores puedan colocar su respectivo símbolo en la matriz, se crearon casillas para los jugadores al lado derecho e izquierdo de la pantalla, donde se muestra el nombre del jugador, su signo en el juego y su respectivo puntaje y, por último, se agregaron los botones correspondientes para que los jugadores puedan reiniciar el juego, rendirse o salir del mismo. La interrupción 33h con los servicios 00h (Activación de mouse), 01h (Mostrar puntero de mouse en pantalla) y 03h (Registro de botón y posición en (X, Y)) fue utilizada con tal de habilitar el uso del mouse de la computadora para así lograr que los jugadores puedan hacer uso de las funciones especificadas en pantalla.

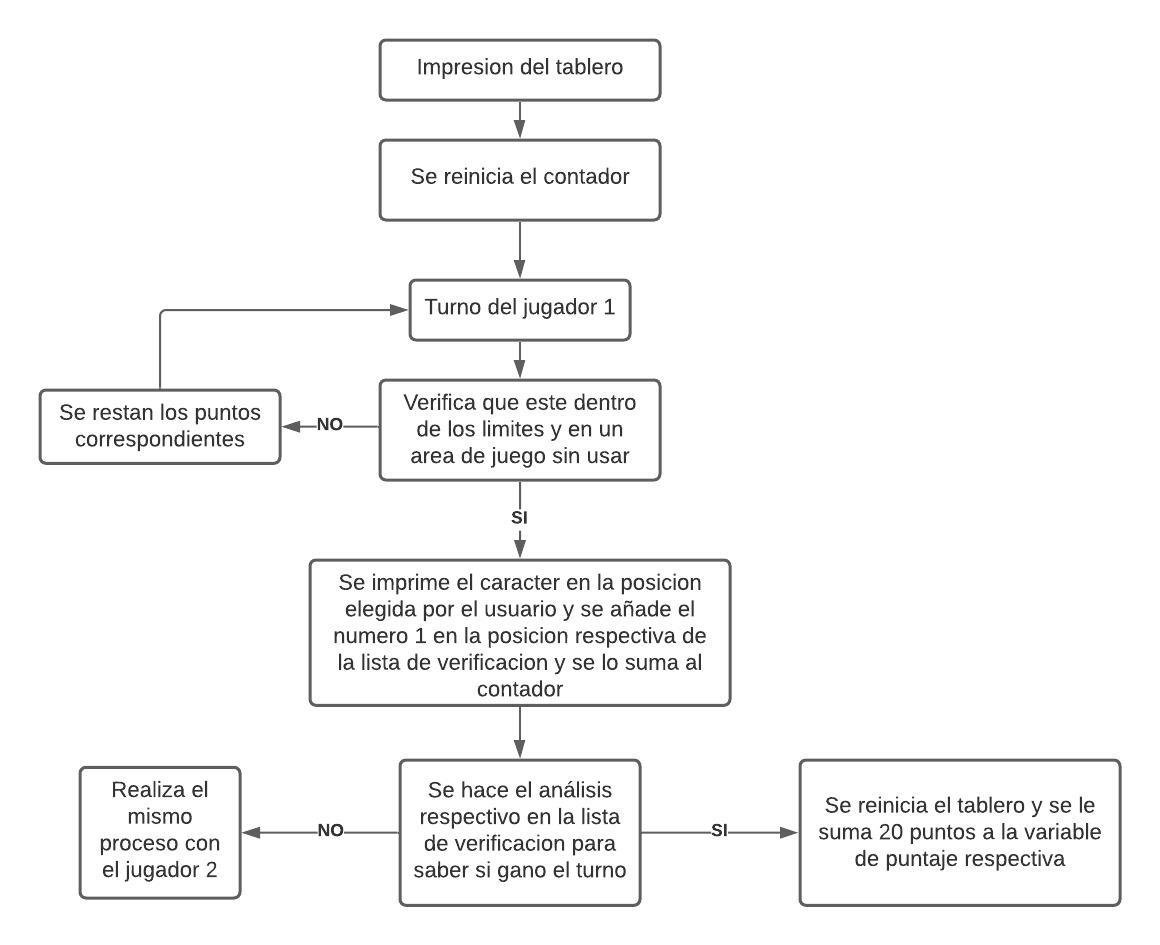


Figura 1: Diagrama de flujo

**Tabla

Descripción generada automáticamente**

**Imagen digital de una jaula

Descripción generada automáticamente con confianza baja** Figura 2: Cálculos necesarios para impresión de símbolo OFigura 3: Prototipo de interfaz de juego

# Análisis de resultados

Como resultado final, se presenta que la impresión de la matriz, los símbolos en pantalla, los botones y las casillas de nombre, símbolo y puntaje de cada jugador se imprimen de manera correcta y eficiente, con todos los parámetros establecidos en el código y en las necesidades del programa como tal, como lo son los diferentes colores, las formas y los caracteres en pantalla, así como también se presenta toda la información necesaria durante el desarrollo de la partida en pantalla. Los botones están delimitados correctamente y cada uno realiza su función respectiva, además de que en cada una de las posiciones en la matriz se imprime el respectivo carácter según el usuario que la oprime en la posición especificada y no presentan problema alguno. La ejecución del juego como tal es fluida, la impresión de los puntajes se da inmediatamente luego de la suma o resta de puntos correspondientes y luego de que alguno gana 50 puntos o más, se imprime el ganador en pantalla y el juego se reinicia para iniciar una nueva partida.

Cuadro 1: Análisis de resultados con respecto a los requerimientos.

|  |  |  |
| --- | --- | --- |
| *Tarea/Requerimiento* | *Estado* | *Observaciones* |
| *Dibujo de la matriz.* | *Completo 100%* | *El dibujo de la matriz se imprime correctamente en pantalla.* |
| *Elección de carácter con el que se va a jugar e ingreso de “nickname” de tres caracteres.* | *Completo 100%* | *Los jugadores pueden elegir su símbolo al escribir su “nickname” de tres caracteres al inicio del juego.* |
| *Incremento y decremento de los puntajes.* | *Completo 100%* | *Los puntajes se incrementan y decrementan dependiendo del caso específico.* |
| *El color de los signos de los jugadores difiere.* | *Completo 100%* | *El color del símbolo “X” es cian brillante y el color del símbolo del octágono es rojo brillante.* |
| *Turno de por medio entre los jugadores.* | *Completo 100%* | *Los jugadores tienen un turno de por medio cada uno para seleccionar la posición que quieren jugar. Siempre empieza el jugador con el símbolo “X”.* |
| *Implementación del mouse.* | *Completo 100%* | *Las delimitaciones del mouse funcionan correctamente y realizan las funciones especificadas.* |
| *Implementación de sonido.* | *Completo 100%* | *El juego realiza un sonido cada vez que algún jugador oprime una zona prohibida.* |
| *Funcionamiento de botones de rendirse, reiniciar y salir.* | *Completo 100%* | *Los 3 botones realizan las funciones especificadas correctamente.* |

# Conclusiones

1. Se logró determinar y conocer el proceso que realizaban las interrupciones con sus respectivos servicios del mouse, teclado y DOS, llegando a implementarlos de forma correcta en el proceso de desarrollo de las instrucciones.
2. Se determinó la importancia de diseñar los programas previamente y realizar los procesos de obtención de las coordenadas en “X” e “Y” en pantalla con el fin de ubicar las zonas específicas de impresión de los píxeles y delimitar el funcionamiento del ratón dentro de los rangos establecidos en la pantalla.
3. Se remarcó la importancia de crear etiquetas y macros con el fin de escribir un código ordenado donde no se repitan grandes cantidades de código innecesario y a la vez se indagó en el correcto uso de estos para sus diferentes requerimientos.
4. Se investigó el uso correcto uso de los registros, para que cada una de las instrucciones dadas al emulador se completaran sin afectar los resultados requeridos o el funcionamiento del programa como tal.
5. Se implementó el uso de técnicas externas a las vistas en clase, como el diseño de interfaces, diagramas de flujo e implementación de algunas técnicas como la implementación de sonido.

# Recomendaciones

1. Es importante conocer y estudiar la sintaxis correcta del lenguaje Ensamblador, para lograr un correcto funcionamiento de los programas y sus respectivas instrucciones.
2. El comentar zonas de código innecesarias para algunas pruebas puede ser útil, ya que, a partir de cierta cantidad de líneas, el código se vuelve pesado y difícil de emular, lo que lleva a perder mucho tiempo si es necesaria la comprobación de un problema en varias ocasiones.
3. Se debe de ejecutar el código paso a paso en zonas donde hay posibles errores o bugs del código para así localizar el problema de la forma más rápida posible y sin perder el tiempo.
4. Es preferible la búsqueda de información en libros de años anteriores al 2000, ya que el lenguaje Ensamblador x86 es antiguo y se facilita más su búsqueda en estos que en la propia internet, donde casi siempre la información que se encuentra es en foros, siendo esta de casos muy específicos y hasta difícil de entender en muchas ocasiones.
5. Algunas áreas o campos de estudio relacionados al proyecto son: la psicopedagogía, como forma de enseñar el proceso elaboración de algoritmos de juegos y fomentar la educación relacionada a la tecnología, y el campo del desarrollo de videojuegos como tal, en el que se tiene como base en muchas ocasiones juegos tan simples como “Gato” para crear juegos nuevos y con un funcionamiento aún más complejo.

# Bibliografía

Daniel-gomez. (2015, octubre 8). LENGUAJE ENSAMBLADOR Y PROGRAMACIÓN PARA IBM® PC Y COMPATIBLES. Dokumen.tips; Unknow. https://dokumen.tips/documents/lenguaje-ensamblador-y-programacion-para-ibm-pc-y-compatibles-5616b5dc931b5.html

Galeano Gil G. Gómez Puildo J A. (s/f). Tabla de interrupciones. Tripod.com. Recuperado el 17 de junio de 2021, de http://ebadillo\_computacion.tripod.com/ensamblador/8086\_int.pdf

Ocaña Velásquez, J. D. (2020). ARQUITECTURA DE COMPUTADORAS. http://repositorio.uladech.edu.pe/handle/123456789/18020