



## CI-0119 Proyecto Integrador de Arquitectura y Ensamblador Enunciado ejemplo del proyecto

Profesores: Francisco Arroyo Mora ([rafael.arroyo@ucr.ac.cr](mailto:rafael.arroyo@ucr.ac.cr))

Alexander Quesada Quesada C16131

Alejandro Jiménez Corea B84032

Carlos Antonio Sánchez Blanco Blanco C17226

Diego Gonzales Mata C03406

Emanuel González Chaves C03361

### Tema

Este proyecto consiste en un **Tic-Tac-Toe 4x4**. Este juego requerirá una función de arreglos que mantenga el rastro de los tiempos y jugadas de ambos jugadores y una arquitectura lógica por debajo de la interfaz humana que la sustituya.

La función del software en este tipo de juegos no complejos está basada en descarte de posibilidades usando funciones lógicas que actúen como sustento lógico, igual que en juegos antiguos como Tetris o Buscaminas, son meras representaciones matemáticas conectadas desde un hardware representado en compuertas lógicas a una interfaz humana capaz de interactuar con los usuarios, o jugadores dado al ser un juego, a través de un dispositivo periférico que reciba los comandos.

### Tic-Tac-Toe

El juego consiste en 16 cuadros ordenados en una matriz 4x4, originalmente, en una 3x3, pero para este proyecto se expande una fila y columna más, y se conserva la regla de “3 en raya gana” lo cual implica que si cualquier jugador (de los 2 jugadores) coloca su marca 3 veces seguidas en diagonal, horizontal o vertical, gana, adicionalmente, al contrario del juego original, se añade un temporizador de 5 jugadas por marca, es decir, una marca no se queda fija en todo el juego sin parar, sino que, se desvanece 5 jugadas después de que fue colocada.





## Requerimientos funcionales

1. El juego deberá representar en pantalla las jugadas solicitadas individualmente por cada jugador, siendo 0, no jugada, “1”, jugada por el jugador 1 y “2” jugada por el jugador 2.
2. En la primera jugada, las casillas centrales están bloqueadas.
3. Las jugadas que se desvanecen se verán reflejadas en pantalla cambiando a 0.
4. Cuando se desvanece una jugada, cualquier jugador puede reutilizar esa casilla.
5. El juego deberá terminar y avisar cuando un jugador gane, indicando el ganador.
6. Bloquea las casillas que no sean 0 para cualquier jugador.
7. Se hará una primera versión en lenguaje de alto nivel que muestre el diseño de la matriz de casillas con caracteres para ser representada durante la partida a los jugadores.

## Requerimientos de arquitectura

1. Para el diseño de la lógica a nivel de hardware se creará utilizando un diagrama tipo UML y se pondrá en práctica utilizando un lenguaje de alto nivel (C, C++) y Ensamblador.
2. La interacción con los jugadores, el dibujo de la matriz y el envío de información al ensamblador se programará en un lenguaje de alto nivel (C, C++). Para esto se utiliza métodos que utilizan los retornos de las funciones en el lenguaje ensamblador para generar una interfaz gráfica que represente la situación del tablero, y métodos que envían las jugadas al sustento lógico-matemático de x86.
3. El lenguaje ensamblador cumple con las siguientes funciones:
  - **Llevar el registro de jugadas:** Función que devuelve 16 enteros cada turno que representan las casillas del tablero. Esto se hace con un vector, todos sus valores inicializados en 0. Se coloca 1 en la casilla es seleccionada por el jugador 1, y 2 si es seleccionada por el jugador 2.
  - **Contar las jugadas:** Función que devuelve un entero que representa el turno a jugar, inicialmente en 0, suma 1 por cada jugada.





- **Bloquear las casillas centrales al comienzo:** Función que no permite jugar las casillas centrales cuando el contador de jugadas es 1
- **Desvanecer casillas:** Para esto se usa un arreglo de 16 enteros, inicializados en 0, se coloca un 5 en la posición del vector correspondiente a la casilla jugada, y se elimina 1 a todos los valores del vector que sean distintos de 0 por cada jugada nueva, cuando un valor en este vector pasa de ser 1 a 0, la casilla que representa en el registro de jugadas, pasa a ser 0 (Al inicio de cada nuevo turno).
- **Verificar 3 en raya:** Función que devuelve un 1 o 2 cuando en diagonal, horizontal o vertical se tienen casillas jugadas por el mismo jugador.

