

**Asignatura:**

[Código, Grupo y Nombre de la Asignatura]

**Trabajo de Curso:**

**Tic-Tac-Toe**

**Autores:**

[Matrícula y Nombre del Estudiante 1]

[Matrícula y Nombre del Estudiante 2]

[Matrícula y Nombre del Estudiante 3]

**Tutor:**

[Título o Grado Científico y Nombre del Tutor/Profesor]

**Entregado en Fecha:**

[Fecha de Entrega]

**PONTIFICIA UNIVERSIDAD CATÓLICA MADRE Y MAESTRA – PUCMM  
FACULTAD DE CIENCIAS DE LAS INGENIERÍAS  
Departamento de Ingeniería en Sistemas y Computación – ISC**

# Resumen / Abstracto del Problema

Tic Tac Toe es un juego de estrategia en el cual existe un tablero de dimensión 3x3 o 5x5 en el que juegan dos jugadores (X u O). Un jugador gana si consigue tener una línea de 3 de sus símbolos: la línea puede ser horizontal, vertical o diagonal. Cada jugador solo debe colocar su símbolo una vez por turno y no debe ser sobre una casilla ya jugada. En caso de que el jugador haga trampa el ganador será el otro. Se debe conseguir realizar una línea recta o diagonal por símbolo.

Tic Tac Toe is a strategy game in which there is a board of dimension 3x3 in which two-player play (X or O). One player wins and tries to obtain a line of 3 symbols: the line could be horizontal, vertical or diagonal. Each player can only put one symbol at time for each turn and cannot put the symbol over a cell that already has been played. In case the player cheats in the game the other player will be the winner. There should be a straight line or a diagonal line for each symbol.

Tabla de contenido

[Resumen / Abstracto del Problema 1](#_Toc409432871)

[Análisis 2](#_Toc409432872)

[Cuerpo del Informe 2](#_Toc409432873)

[Código Fuente 2](#_Toc409432874)

[Condiciones de Error 2](#_Toc409432875)

[Notas al Profesor 2](#_Toc409432876)

[Conclusiones 2](#_Toc409432877)

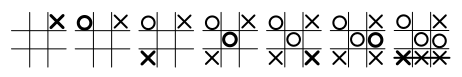
[Recomendaciones 2](#_Toc409432878)

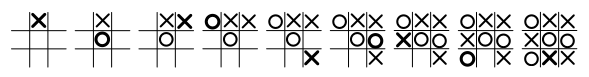
[Referencias Bibliográficas 2](#_Toc409432879)

# Análisis

Este juego será desarrollado en Unity 3D, el cual es un motor de plataforma cruzada utilizado para desarrollar video juegos para dispositivos tales como móviles, PC, consolas de video juego, y sitios web.

El juego Tic-Tac-Toe es un juego muy simple, en donde dos jugadores tratan de completar una fila, columna o diagonal con tres X o tres O en un tablero 3x3. Al empezar el juego, este empieza con X y luego la O. Al juego ser bien simple los jugadores no tardan en darse cuenta que el juego siempre termina empatado sin importar con que símbolo este empiece porque al tablero ser pequeño hay una jugada óptima para ambos jugadores y esto hace que el juego siempre quede empatado en el momento que ambos jugadores notan la mejor jugada. Más abajo se puede ver ejemplos de jugadas en las imágenes que dan una idea de cómo es el juego.





# Cuerpo del Informe

El cuerpo del informe se estructura en dos capítulos:

- Capítulo 1 – Descripción del Negocio

Aquí usted expondrá todo lo referente a la investigación realizada sobre el negocio. Deben incluirse las reglas de negocio que permitan validar la aplicación.

Ejemplos de algunas reglas de negocio:

* Una asignatura tiene uno o varios profesores.
* Todas las personas tienen un local asociado.

- Capítulo 2 – Solución Propuesta

Este capítulo tiene la finalidad de describir la solución propuesta ante la problemática planteada. Incluir:

* Diagramas de flujo, pseudo-códigos o diagramas de clases, casos de uso, actividades, de estado, de secuencia o cualquier otra herramienta que explique cuáles algoritmos, librerías, funciones o componentes se utilizaron para la resolución del problema.
* Organización del proyecto en capas lógicas, indicando los paquetes o librerías implementados en cada capa (nombre y descripción del contenido).
* Descripción de los patrones utilizados en la modelación. Si se han implementado algunos patrones de diseño en sus diagramas, utilice este espacio para citar acerca de los mismos sus nombres, descripción general y finalidad de su uso en esta solución.
* Capturas de pantalla o ejemplos de uso. Muestre el proceso a seguir para los principales lugares en su proyecto, así como datos de prueba y flujos de trabajo. Puede auxiliarse de capturas de pantalla, ejemplos de salida o vídeos que ilustren su objetivo.

Debe describir de una manera natural, concisa y clara por qué prefirió un algoritmo y no otro. No "corra" el programa. Es muy contraproducente describir instrucción por instrucción todo el código; se deben agrupar las sentencias que guarden relación y explicar el motivo de las mismas como un todo. Trate de no mencionar cosas sobreentendidas.

Es la sección más oportuna para hacer un comentario de por qué hizo algo de una manera específica. Si su código tiene muchas funciones, módulos o librerías, describa los más relevantes por separado..

# Código Fuente

Bajo este título se incluye el código fuente del programa a entregar. Deberá siempre anexar al documento el o los archivo(s) fuente para poderlos compilar de manera adecuada. Puede adjuntar en documento Zip toda la solución, en caso de que sea necesario. En caso de que su proyecto tenga múltiples archivos de código fuente, o resulte impráctico colocarlo dentro de esta sección, bastará con enviar el código junto con este reporte sin incluirlo en esta sección.

# Condiciones de Error

Por lo general, hay motivos que pueden causar que el usuario de un programa obtenga una anomalía o que el algoritmo usado tenga ciertas restricciones de uso. Ejemplo de esto es cuando a un programa que opera con números positivos se le digitan negativos. Ocurre también con funciones que no son capaces de operar con un número muy grande (por ejemplo, el factorial).

Estas anomalías deben enumerarse en esta sección del trabajo. Deberá incluir ejemplos de corrida con errores en esta sección donde se contemple el mismo claramente y se pueda visualizar cualquier mensaje de error desplegado. Tenga cuidado al decir que su programa no tiene condiciones de error. Quedará luego a consideración del profesor valorar las condiciones de error.

# Notas al Profesor

En caso de que desee hacer un señalamiento al profesor acerca de su trabajo, incluirlo bajo este título al final. Puede incluir cualquier otro aspecto que el equipo considere necesario exponer o explicar.

Por ejemplo: aspectos tenidos en cuenta para el diseño de la interfaz, filosofía de trabajo en la aplicación, tratamiento de errores, ayudas, aspectos novedosos tenidos en cuenta en la implementación, organización del trabajo en equipo, cronogramas de trabajo (plan y real), entrevistas (planificadas, reales, plantillas utilizadas, tipos de usuarios/clientes entrevistados) , estándar de código utilizado, ente otros..

# Conclusiones

Aquí se abordan las conclusiones del trabajo realizado, no de la asignatura, ni sobre los beneficios que le reporta a un estudiante la realización de este proyecto de curso. Deben redactarse teniendo en cuenta:

* Estado de cumplimiento de los objetivos planteados en la introducción.
* Observaciones, mediciones, estilos, etc., puestos en práctica durante la ejecución del proyecto. Por ejemplo: conclusiones, que consideren de interés a partir de sus propias experiencias, respecto a la organización del trabajo en equipo.

# Recomendaciones

Se redactan teniendo en cuenta aspectos que quedaron pendientes, mejoras que en el futuro se pueden realizar a la aplicación, posibilidades de implantación, entre otros.

# Referencias Bibliográficas

Especifique las referencias bibliográficas utilizadas durante la elaboración del proyecto.