

INSTITUTO TECNOLÓGICO AUTÓNOMO DE MÉXICO



**JUGADOR ARTIFICIAL DE DOMINÓ BASADO EN ÁRBOL  
DE BÚSQUEDA MONTE CARLO**

**TESIS**

**QUE PARA OBTENER EL TÍTULO DE  
INGENIERO EN COMPUTACIÓN**

**P R E S E N T A**

*Andrés Cruz y Vera*

**ASESOR: Dr. Marco Antonio Morales Aguirre**

**TABLA DE CONTENIDO**

**1. INTRODUCCIÓN**

1.1 Contexto

1.2 Identificación del problema

1.3 Objetivo

1.4 Metodología

1.5 Organización del documento

**2. ANÁLISIS DE REQUISITOS DE SOFTWARE**

2.1 Requerimientos funcionales

2.2 Restricciones

2.3 Trabajos relacionados

**3. REGLAS DEL DOMINÓ**

**4. JUEGOS ESTOCÁSTICOS CON INFORMACIÓN IMPERFECTA**

**5. ÁRBOL DE BUSQUEDA MONTE CARLO**

**6. DISEÑO DEL PROGRAMA**

**7. IMPLEMENTACIÓN**

**8. VALIDACIÓN**

**9. CONCLUSIONES**

**REFERENCIAS**

**PLAN DE TRABAJO**

## **Introducción**

En el primer capítulo se presentará el problema que aborda el presente trabajo. Se dará el contexto histórico de la relación entre los videojuegos y los jugadores artificiales para luego identificar el problema y definir tanto los objetivos como la metodología a seguir.

### **1.1 Contexto**

La historia de los juegos por computadora inicia desde la década de 1950 en el ámbito académico y en los años setenta y ochenta gana popularidad para el público en general. Los videojuegos han tenido un gran impacto en la cultura popular, así como en grandes figuras de la computación que tuvieron su primer acercamiento a los ordenadores por medio de estos y del lenguaje BASIC

Asimismo, los juegos de mesa han tenido un papel importante en el desarrollo del área de inteligencia artificial siendo una área muy fructífera de investigación como en el caso del ajedrez y la famosa contienda entre Deep Blue y Garry Kasparov

### **1.2 Identificación del problema**

El desarrollo de videojuegos es un ámbito multidisciplinario en donde se utilizan técnicas de inteligencia artificial para complementar la experiencia de juego del usuario. Los jugadores artificiales (o *bots*) cumplen un papel importante como contrincantes o personajes secundarios dentro del juego.

Con miras a desarrollar una versión online del juego de dominó con un modelo de monetización basado en anuncios, se tiene como uno de los objetivos maximizar el número de impresiones de los anuncios en el usuario. Así, es necesario proveer una experiencia atractiva que tenga como efecto que el usuario pase un largo tiempo activo en la página.

El lanzamiento a mercado de un juego multijugador online presenta distintos retos. Entre ellos, existe la necesidad de crear una base mínima de usuarios que permita tener un tiempo razonable de espera para poder encontrar una partida a la cual unirse. Una forma de solventar parcialmente este obstáculo, particularmente en las primeras fases del lanzamiento, es contar con jugadores artificiales que suplementen la falta de contrincantes humanos.

Así, es deseable contar con un jugador artificial que permita a los usuarios iniciar una partida aun en las circunstancias en que no cuenten con suficientes personas para completar los equipos. Al momento en que se realiza este escrito, no se ha encontrado una implementación de código abierto de un jugador artificial para el juego de dominó (con las reglas que se usan en latinoamérica) que cuente tanto con una licencia que permita su uso comercial así como una interfaz de programación diseñada para su integración a un juego de tiempo real con usuarios humanos.

### 1.3 Objetivos

Implementar un programa de computadora que sea capaz de jugar en una partida de dominó como parte de un equipo de dos participantes que compiten con dos contrincantes.

### 1.4 Metodología

Para la implementación del bot, se decidió utilizar la metodología de cascada debido a que el alcance y la funcionalidad del proyecto es relativamente pequeña.

### 1.5 Organización del documento

1. Introducción
2. Análisis de requisitos del software
3. Las reglas del dominó
4. Juegos estocásticos con información imperfecta
5. El método de Árbol de Búsqueda Montecarlo
6. Diseño del programa
7. Implementación
8. Validación
9. Conclusiones

## **2. Análisis**

Una vez que se ha identificado el problema, se formulará una primer aproximamiento a la solución por medio de los requerimientos funcionales que debe cumplir, así como por las restricciones que debe satisfacer. También se mostrarán trabajos relacionados.

### **2.1 Requerimientos funcionales**

El programa generará una jugada a partir del estado actual del juego. Es decir, el programa recibirá como entrada una representación de sus fichas asignadas así como de las fichas tiradas por los otros participantes y como salida indicará cual de sus fichas debe jugarse.

También será posible elegir distintos niveles de juego para el programa

### **2.2 Restricciones**

El programa debe generar las jugadas en un tiempo razonable, siempre terminar la ejecución antes de un lapso predeterminado para poder utilizarse en un juego de tiempo real contra contrincantes humanos y no debe poseer información sobre las manos de sus contrincantes ni de su pareja de equipo.

En segundo lugar, el desempeño del programa tiene que ser mejor que la estrategia más sencilla posible: de entre las fichas que se pueden bajar siempre elegir la de mayor puntaje. Así, un equipo de los jugadores aquí implementados debe vencer a un equipo de jugadores débiles un  $X\%$  de las partidas. Por otro lado, es posible cuantificar el desempeño de la solución poniéndola a prueba contra un equipo que conozca todas las fichas del juego. Así, la solución deberá ganar un  $Y\%$  de partidas contra un equipo de jugadores fuertes.

Por último, debe exponer una API sencilla para ser integrado a distintas interfaces, tanto aplicaciones web como móviles.

### **2.3 Trabajos relacionados**

Uno trabajo importante en el ámbito de algoritmos para juegos de información imperfecta lo realiza Ginsberg (2001). Con esta metodología logra implementar un jugador de Bridge de nivel experto.

Por otra parte, un jugador artificial en un contexto de incertidumbre puede estudiarse desde la perspectiva de procesos de decisión de Markov como en la disciplina de aprendizaje por refuerzo. En este campo es importante el trabajo de Mnih et al. (2013) que es uno de los primeros en integrar aprendizaje profundo a los algoritmos de aprendizaje por refuerzo para la creación de agentes en el juego de Atari.

Long et al. (2010) realizaron un trabajo en donde , a partir de árboles de juego sintéticos, definen indicadores estadísticos que les permiten identificar propiedades importantes de juegos de información imperfecta en los que el método de Arbol de Búsqueda Monte Carlo (MCTS por sus siglas en ingles) se puede adaptar exitosamente. Dicho trabajo extiende la línea de investigación sobre las limitaciones de MCTS en el contexto de información imperfecta que inician Frank y Basin (1998)

Asimismo, se recuperó de la web un proyecto de licenciatura sobre un jugador artificial para dominó (en el texto se le refiere como Latin-American dominoes) desarrollado por Angeris y Li (2016) de la universidad de Stanford. El proyecto consiste en simulaciones para contrastar distintos algoritmos pero no tiene la finalidad de ser consumido como una API.

## **Referencias**

Angeris, G., & Li, L. (2016). CS 221 Project Final : DominAI. Recuperado de <https://web.stanford.edu/~guillea/papers/dominai.pdf>

Frank, I., and Basin, D. 1998. Search in games with in- complete information: A case study using bridge card play. *Artificial Intelligence* 87–123.

Ginsberg, M. L. (2001). GIB: Imperfect Information in a Computationally Challenging Game. *Journal of Artificial Intelligence Research*, 14, 303-358. <https://doi.org/10.1613/jair.820>

Long, Jeffrey Richard et al. "Understanding the Success of Perfect Information Monte Carlo Sampling in Game Tree Search." *AAAI* (2010).

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D. & Riedmiller, M. (2013). Playing Atari with Deep Reinforcement Learning. , .