



# rubik-cube

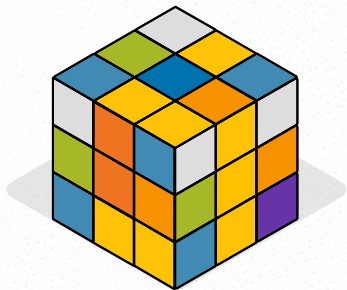
*Resolución de cubos Rubik mediante  
Inteligencia Artificial*

# Cubo Rubik

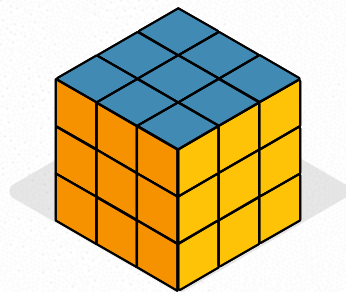
## *Definición del Problema*

# Cubo Rubik

Estado Inicial



Solución



- ▶ 26 piezas tridimensionales
- ▶ 12 acciones/rotaciones posibles
- ▶ Medida de fitness: cantidad de piezas colocadas correctamente

- ▶ 43 trillones de estados posibles
- ▶ 1 estado objetivo

# Algoritmos utilizados

*para resolver el problema*

# Algoritmo Genético

- ▶ Operadores de Selección Natural: **selección, entrecruzamiento, mutación.**
- ▶ Cada **ejecución** busca **resolver** un cubo en un **estado inicial específico.**
- ▶ **Población inicial:** mezclas aleatorias.
- ▶ **Selección de padres:** probabilidad según fitness.
- ▶ **Entrecruzamiento:** single-point crossover.
- ▶ **Mutación:** con alta probabilidad en 3 puntos.
- ▶ **Selección:** individuos más aptos entre padres e hijos.



# Deep Reinforcement Learning

- ▶ Reinforcement Learning + Deep Learning
- ▶ **Modelo:** Deep Q Network
- ▶ **Agente:** Train DQN Algorithm
- ▶ **Entorno:** Cubo Rubik
- ▶ **Objetivo:** maximizar la **suma de recompensas** tomando **acciones** a través de **episodios**

# Algoritmo Completamente Aleatorio

- ▶ Implementado para **medir** la eficiencia de los otros métodos.
- ▶ **Funcionamiento:** aplicar **20 acciones aleatorias distintas** en **n** cubos iguales y registrar el **mejor resultado**.

# Métodos de prueba

*de los algoritmos*



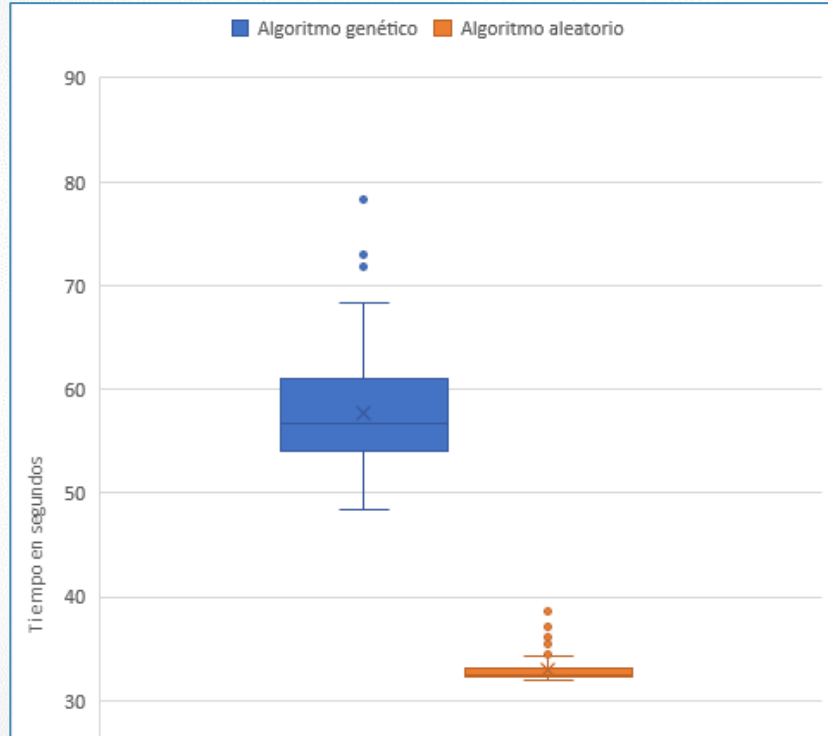
# Métodos de prueba

- ▶ **Deep Reinforcement Learning:** no fue considerado. Realiza siempre la misma acción.
- ▶ **Algoritmo Genético:**
  - 7 distintas combinaciones de parámetros
  - 30 distintos estados iniciales
- ▶ **Algoritmo Aleatorio:**
  - 30 distintos estados iniciales
- ▶ **Método de medición:** ejecutar el algoritmo genético y aleatorio sobre los mismos 135 estados iniciales.

# Resultados

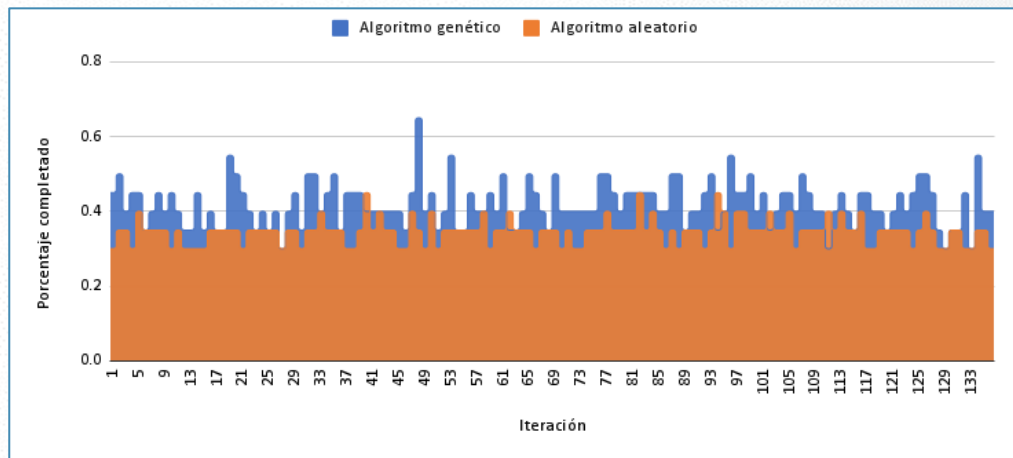
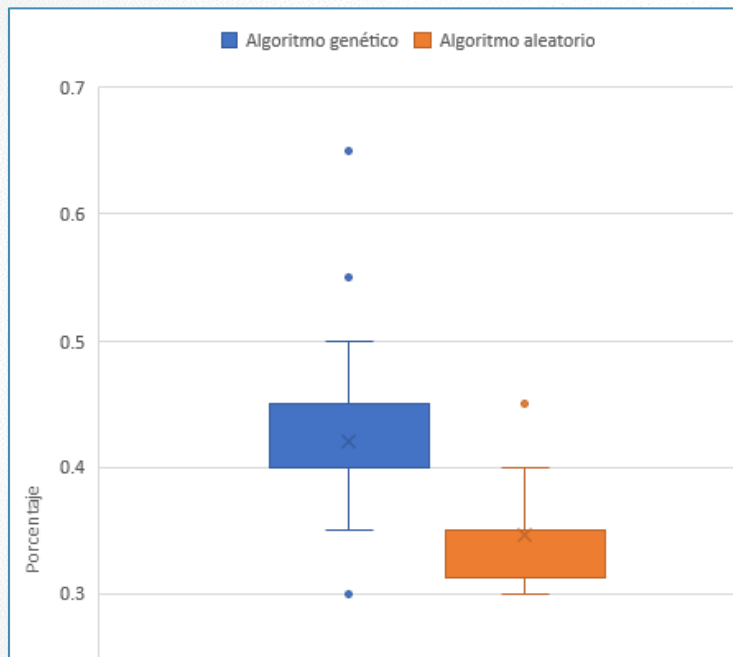


# Tiempos de ejecución



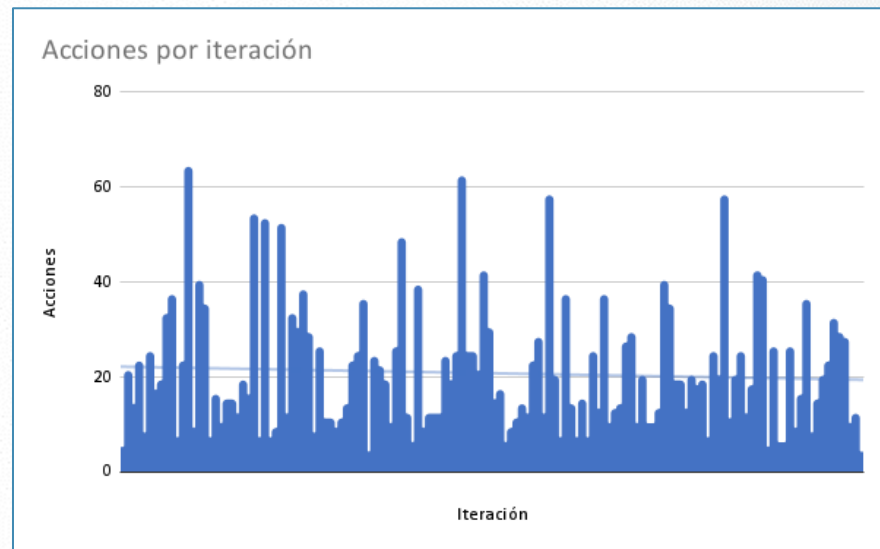
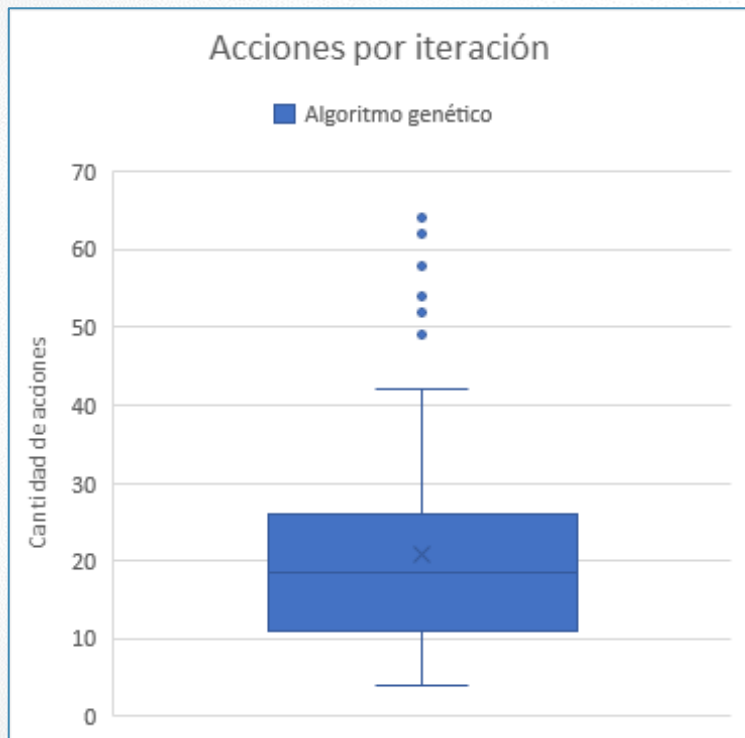
Algoritmo Aleatorio **57%** más rápido que Algoritmo Genético

# Porcentaje completado



Algoritmo Genético **8%** más eficaz que Algoritmo Aleatorio

# Acciones por iteración (Algoritmo Genético)



Promedio: 21 acciones



# Conclusiones

*e ideas futuras*

# Conclusiones

- ▶ Ningún algoritmo llega al estado objetivo.
- ▶ El Algoritmo Genético obtiene 8% mejores resultados que el Algoritmo Aleatorio.
- ▶ El Algoritmo Aleatorio tiene 57% menor costo temporal que el Algoritmo Genético.
- ▶ El Algoritmo Genético genera soluciones con 21 acciones en promedio.
- ▶ El Algoritmo de DRL no logra ningún resultado superior al 10% de progreso.

## Ideas Futuras

- ▶ Realizar mejoras en el **Algoritmo Aleatorio** para mejorar su **efectividad**.
- ▶ Aplicar restricciones basadas en **teoría de grupos** a las **secuencias** de acciones aleatorias.
- ▶ **Reimplementar** el algoritmo de **Deep Reinforcement Learning**.

A decorative background featuring a network diagram with nodes and connecting lines. The nodes are colored in blue, orange, teal, and purple. Some nodes are highlighted with dashed circles. The diagram is positioned on the left and right sides of the slide, framing the central text.

# ¿Preguntas?

GRACIAS