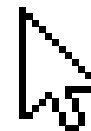


# (72.27) Sistemas de Inteligencia Artificial

## Trabajo Práctico N°2

# Algoritmos Genéticos



Alejo Flores Lucey	62622
Andrés Carro Wetzel	61655
Ian Franco Tognetti	61215
Matías Daniel Della Torre	61016

# 01 Introducción

Presentación del problema a resolver

# 02 Implementación

Definiciones sobre la implementación del algoritmo genético

# 03 Elección de hiperparámetros

Ejercicio 1.1

## 03.01 Análisis de resultados

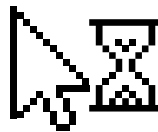
Mejor fitness, tiempo de ejecución, cantidad de mutaciones

# 04 ¿Qué pasa al dejarlo correr por más tiempo?

Ejercicio 1.2

# Introducción

Hemos implementado un **motor de algoritmos genéticos** para optimizar la creación de personajes en un **juego de rol**. Se busca maximizar el desempeño de los personajes dentro de un tiempo limitado.





# 02

## Implementación

Veamos cómo definimos el **genotipo** de los personajes, **detalles de la implementación** y las **convenciones** utilizadas en el estudio.

# Jugadores disponibles



## Warrior

En el frente de batalla junto a los defensores.



## Guardian

Deberá interponerse entre los ataques y sus colegas.



## Archer

Se ubica al final, atacando desde la distancia.



## Wizard

Posee hechizos ofensivos, también auxilia a sus tropas

# Definición del genotipo

Height	Strength	Skill
Intelligence		
Courage	Physique	



# Métodos genéticos implementados

## Selección

- Elite
- Roulette
- Universal
- Boltzmann
- Ranking
- Torneo determinístico
- Torneo probabilístico

## Cruza

- Cruza de un punto (One Point)
- Cruza de dos puntos (Two Points)
- Cruza anular
- Cruza uniforme (Uniform)

## Mutación

- Gen
- Multigen uniforme

# Detalles de implementación

Toda la configuración se recibe en un JSON. Se desarrolló un **JSON-Schema** que valida que el input sea válido.

Se puede seleccionar la población con **N métodos de selección distintos**, cada uno con distintos hiperparámetros.

Se hace uso del patrón **Singleton** para servir el generador de números pseudo-aleatorios, con una semilla predefinida.

Se pueden proveer **muchas condiciones de corte**, y el motor parará cuando alguna de ellas se cumpla.

Para el estudio de resultados, **cada conjunto de configuraciones se ejecuta 10 veces**, para evitar sacar conclusiones de procesos estocásticos.





# 03

## Selección de los hiperparámetros

Al ser tantas opciones, se **define un método** para conseguir la mejor configuración.

## 01. Selección del hiperparámetro

Se elige un hiperparámetro a estudiar.  
El resto se fija arbitrariamente

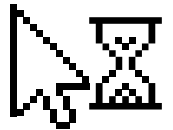
## 02. Variación del hiperparámetro

Se va variando el hiperparámetro en los disponibles.  
Si tiene opciones extra, estas también se van iterando

## 03. Análisis de los resultados

Se analizan los resultados obtenidos. Se utilizará el  
mejor hiperparámetro para los futuros estudios.

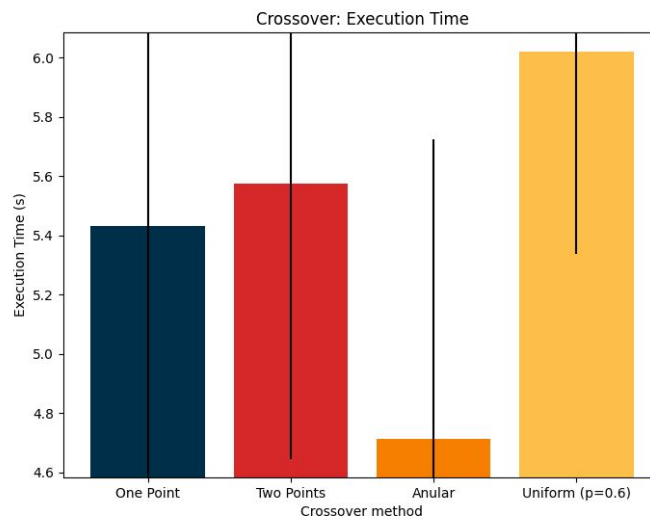
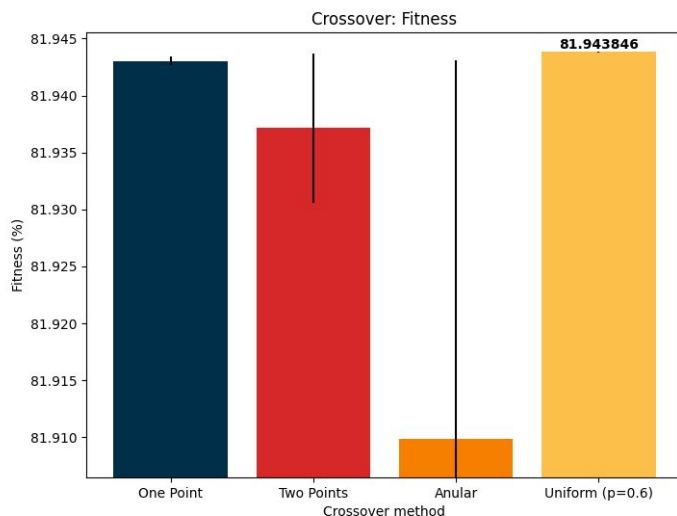
# Métodos de Cruza



# Método de cruza

OnePoint(), TwoPoints(), Anular(), Uniform(p=0.6)

- Player: **WIZARD**
- Selection: **[ELITE()]**
- Mutation: **GEN(p=0.5)**
- Pick: **TRADITIONAL - [ELITE()]**
- Population Size: **1000**
- Mates per Generation: **100**
- Repeticiones: **10**

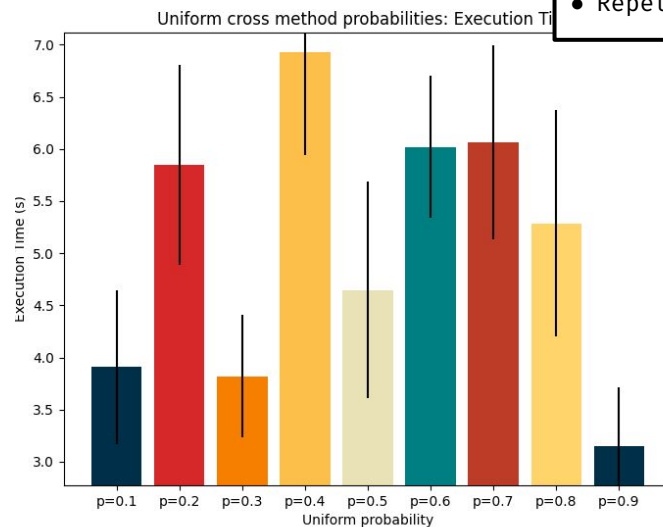
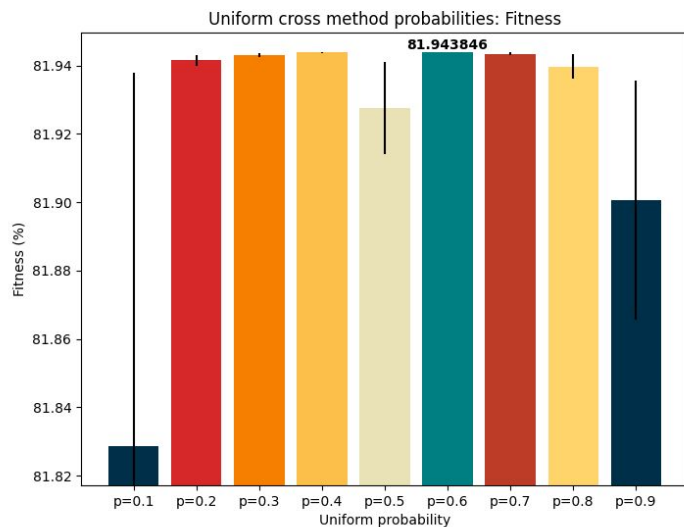


Equilibrando entre Fitness y Execution Time: **One Point**

# Cruza - Uniforme

Estudiamos la mejor probabilidad de intercambio

- Player: **WIZARD**
- Selection: **[ELITE()]**
- Mutation: **GEN(p=0.5)**
- Crossover: **UNIFORM(p=X)**
- Pick: **TRADITIONAL - [ELITE()]**
- Population Size: **1000**
- Mates per Generation: **100**
- Repeticiones: **10**

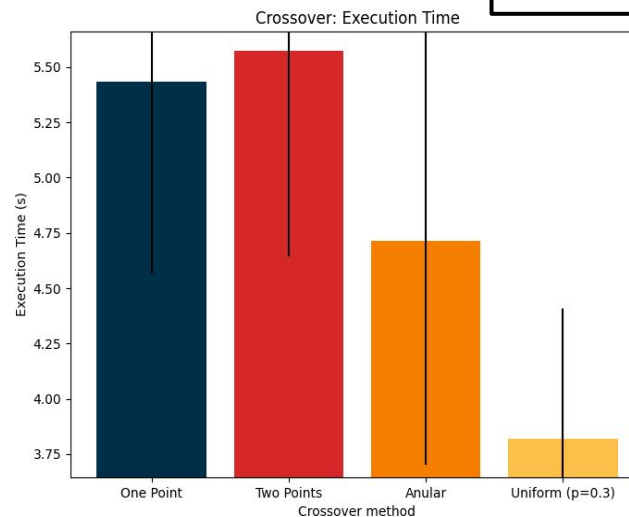
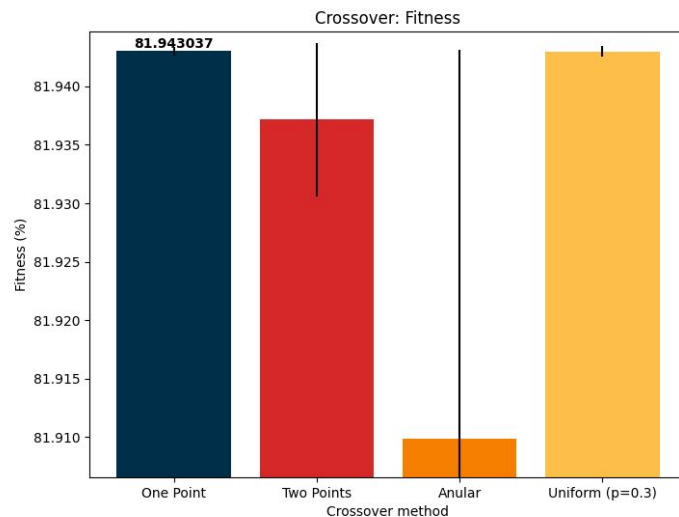


Equilibrando entre Fitness y Execution Time: **p=0.3**

# One Point vs Uniforme (0.3)

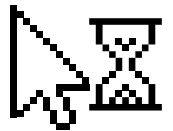
Comparando el tiempo de ejecución entre los 2 mejores fitness, el mejor resultado es...

- Player: **WIZARD**
- Selection: **[ELITE()]**
- Mutation: **GEN(p=0.5)**
- Crossover: **X**
- Pick: **TRADITIONAL - [ELITE()]**
- Population Size: **1000**
- Mates per Generation: **100**
- Repeticiones: **10**



Mejor resultado final: **Uniform(p=0.3)**

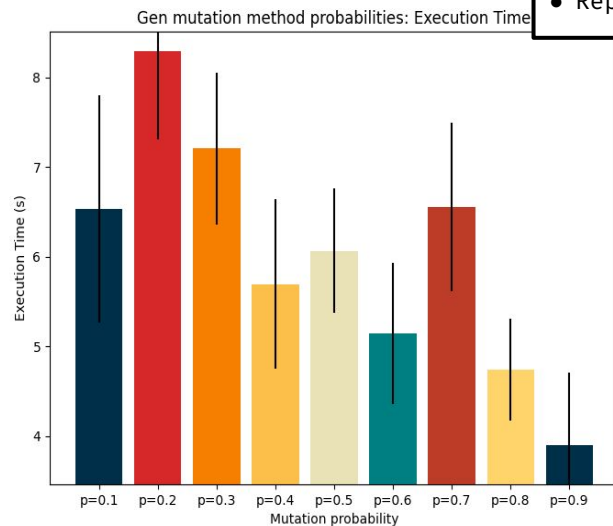
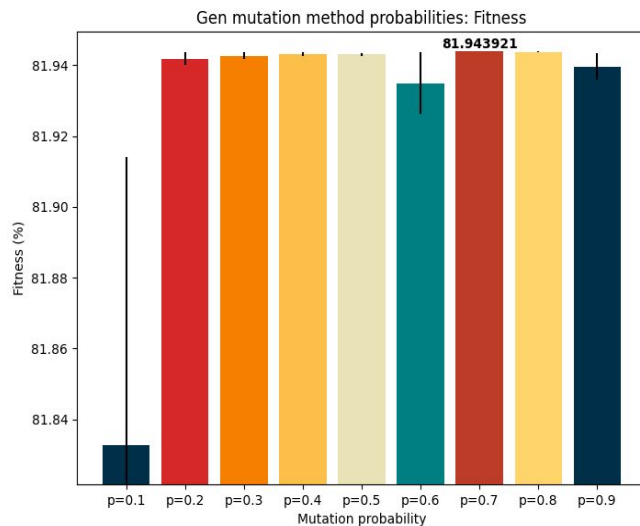
# Métodos de Mutación



# Mutación - Gen

Estudiamos la mutación manteniendo el hiperparámetro anterior, Uniform( $p=0.3$ )

- Player: **WIZARD**
- Selection: **[ELITE()]**
- Mutation: **GEN( $p=X$ )**
- Crossover: **UNIFORM( $p=0.3$ )**
- Pick: **TRADITIONAL - [ELITE()]**
- Population Size: **1000**
- Mates per Generation: **100**
- Repeticiones: **10**



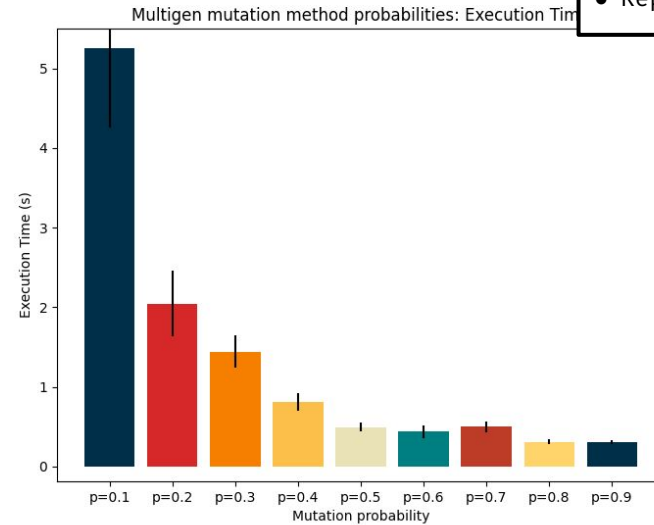
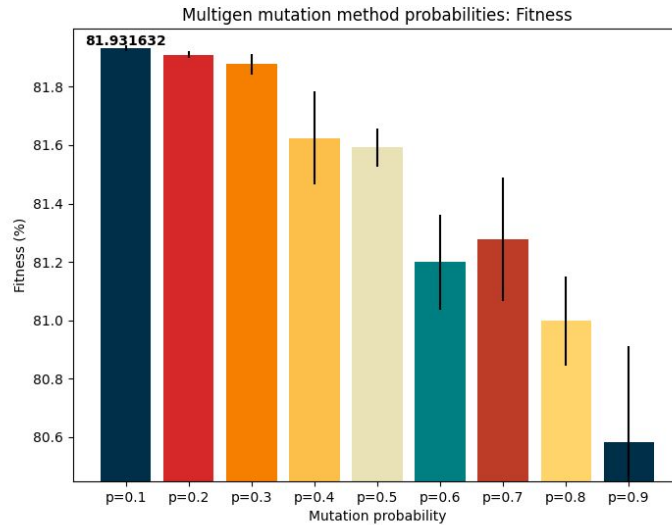
Equilibrando entre Fitness y Execution Time:  **$p=0.8$**



# Mutación - Multigen

Estudiamos la mutación manteniendo el hiperparámetro anterior, Uniform( $p=0.3$ )

- Player: **WIZARD**
- Selection: **[ELITE()]**
- Mutation: **MULTIGEN( $p=X$ )**
- Crossover: **UNIFORM( $p=0.3$ )**
- Pick: **TRADITIONAL - [ELITE()]**
- Population Size: **1000**
- Mates per Generation: **100**
- Repeticiones: **10**

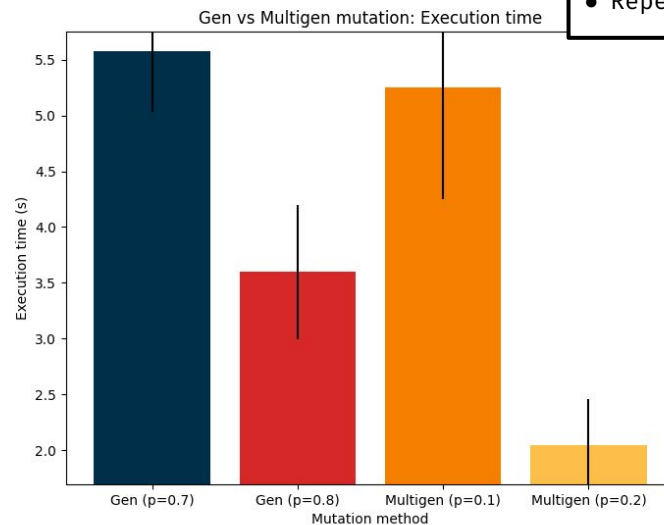
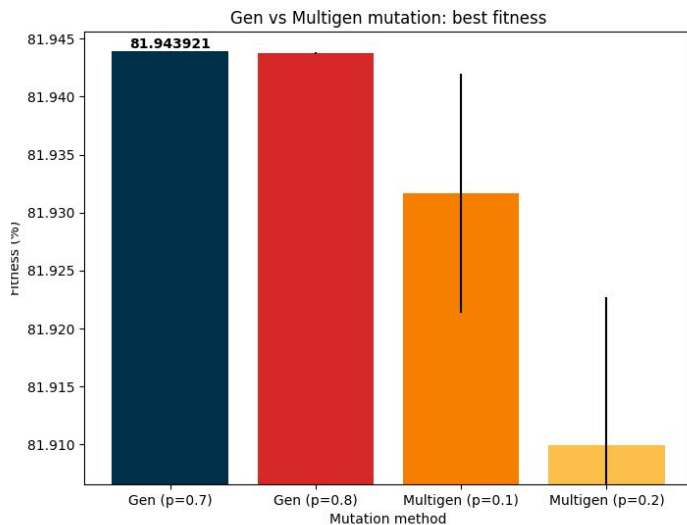


Equilibrando entre Fitness y Execution Time: **p=0.2**

# Gen vs Multigen

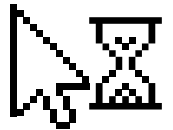
Comparando el tiempo de ejecución entre los 2 mejores fitness de Gen y Multigen, el mejor resultado es...

- Player: **WIZARD**
- Selection: **[ELITE()]**
- Mutation: **X**
- Crossover: **UNIFORM(p=0.3)**
- Pick: **TRADITIONAL - [ELITE()]**
- Population Size: **1000**
- Mates per Generation: **100**
- Repeticiones: **10**



Mejor resultado final: **Gen(p=0.8)**

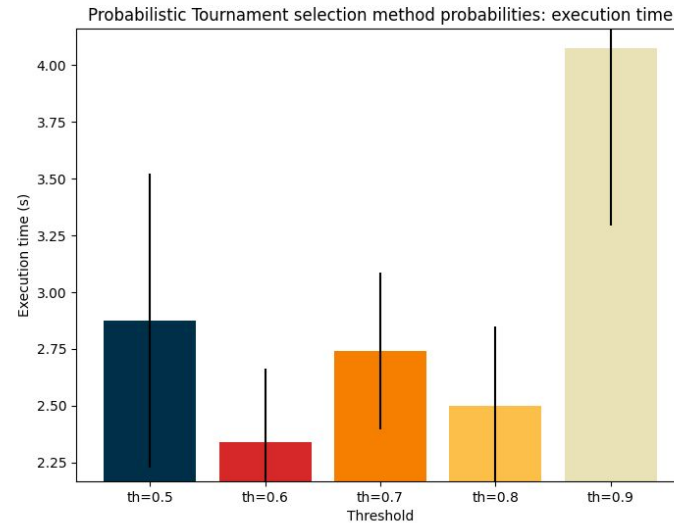
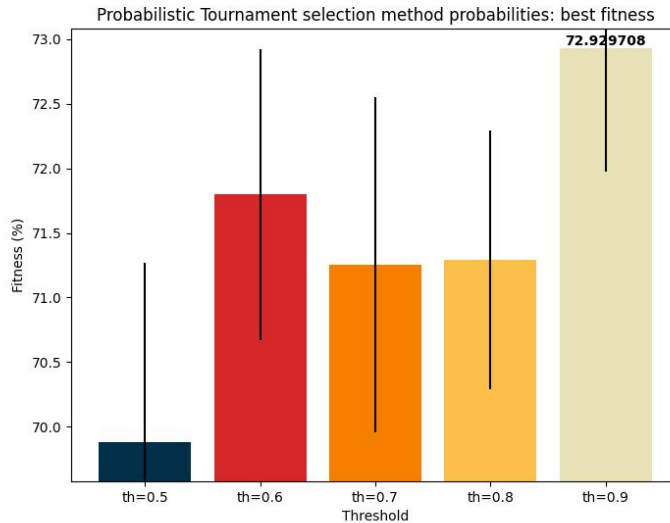
# Métodos de Selección



# Selección - ProbTournament

Variando el threshold del método de selección Probabilistic Tournament, el mejor resultado es...

- Player: **WIZARD**
- Selection: **[PROBABILISTIC-TOURNAMENT(th=X)]**
- Mutation: **GEN(p=0.8)**
- Crossover: **UNIFORM(p=0.3)**
- Pick: **TRADITIONAL - [ELITE()]**
- ...

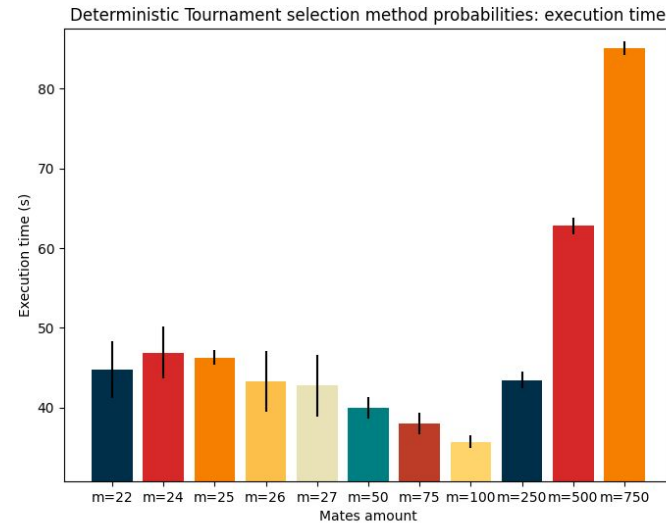
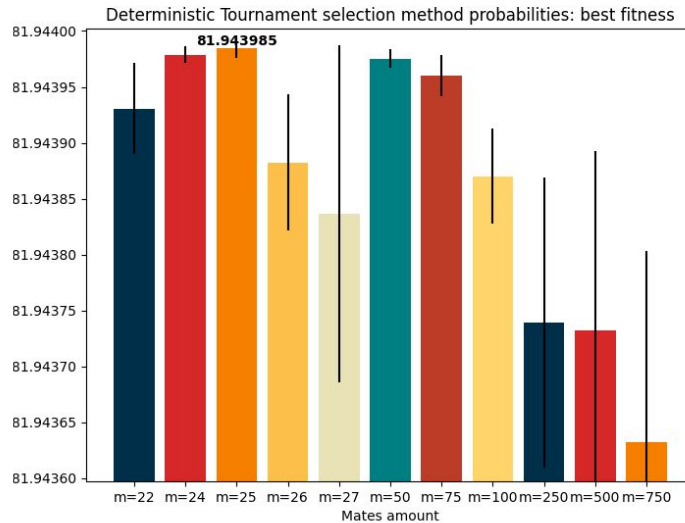


Equilibrando entre Fitness y Execution Time: **th=0.6**

# Selección - DetTournament

Variando el  $m$  del método de selección Deterministic Tournament, el mejor resultado es...

- Player: **WIZARD**
- Selection: **[DETERMINISTIC-TOURNAMENT( $m=X$ )]**
- Mutation: **GEN( $p=0.8$ )**
- Crossover: **UNIFORM( $p=0.3$ )**
- Pick: **TRADITIONAL - [ELITE()]**
- ...

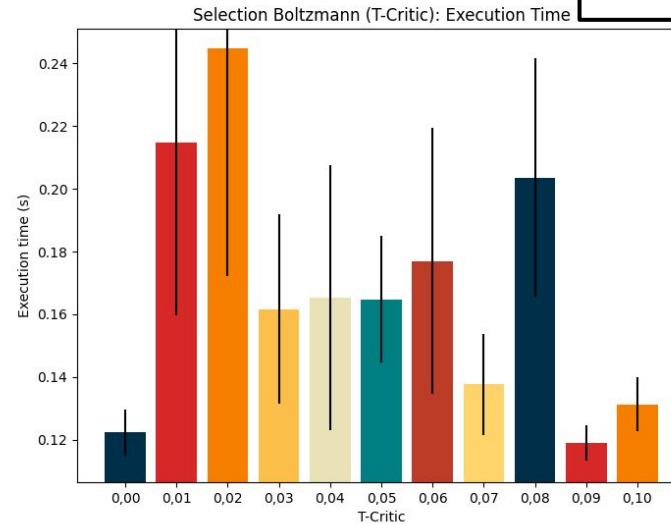
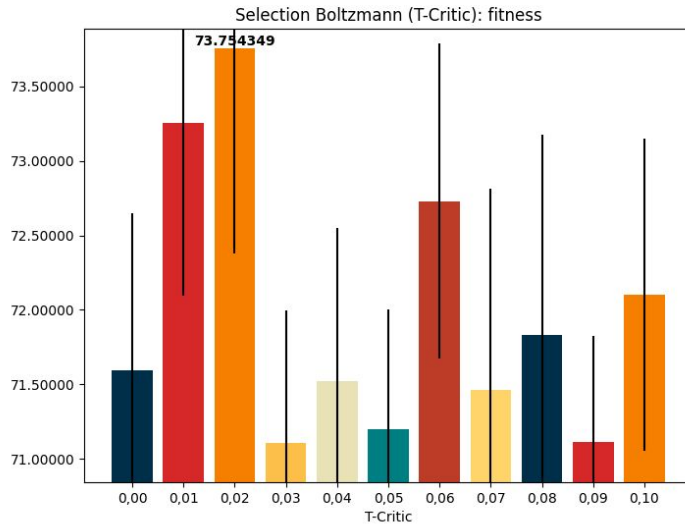


Equilibrando entre Fitness y Execution Time: **m=25**

# Seleccion - Boltzmann

Variando los parámetros del método de selección Boltzmann, el mejor resultado es...

- Player: **WIZARD**
- Selection: **[BOLTZMANN(tCritic=X, t0=50, k=0.05)]**
- Mutation: **GEN(p=0.8)**
- Crossover: **UNIFORM(p=0.3)**
- Pick: **TRADITIONAL - [ELITE()]**
- ...

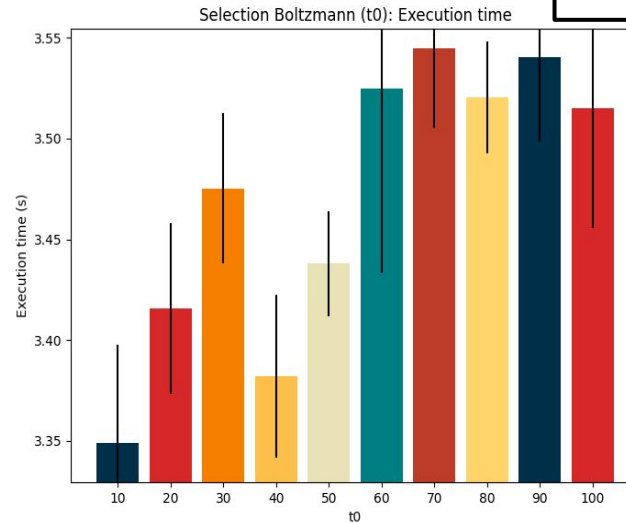
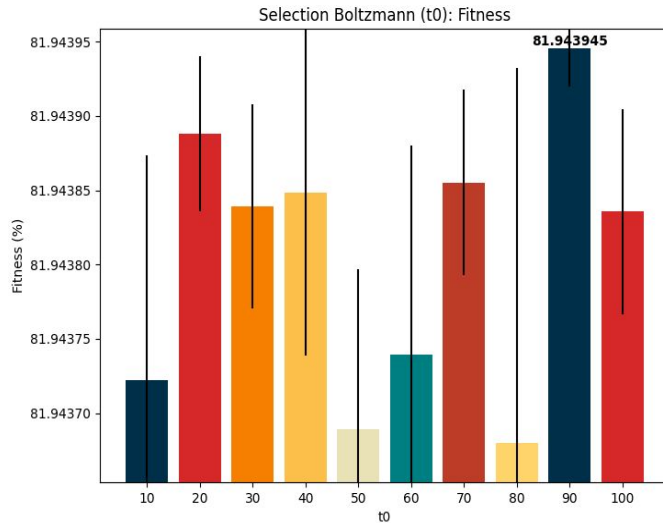


Según Fitness: **tCritic=0.02**

# Seleccion - Boltzmann

Variando los parámetros del método de selección Boltzmann, el mejor resultado es...

- Player: **WIZARD**
- Selection: **[BOLTZMANN( $t_{Critic}=0.02$ ,  $t_0=X$ ,  $k=0.05$ )]**
- Mutation: **GEN( $p=0.8$ )**
- Crossover: **UNIFORM( $p=0.3$ )**
- Pick: **TRADITIONAL - [ELITE()]**
- ...

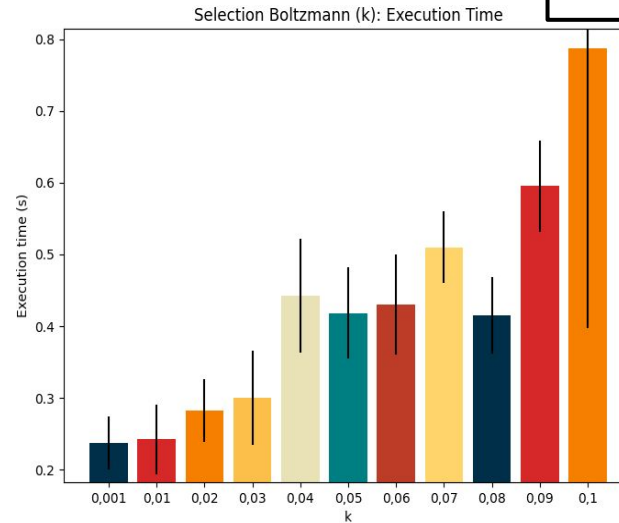
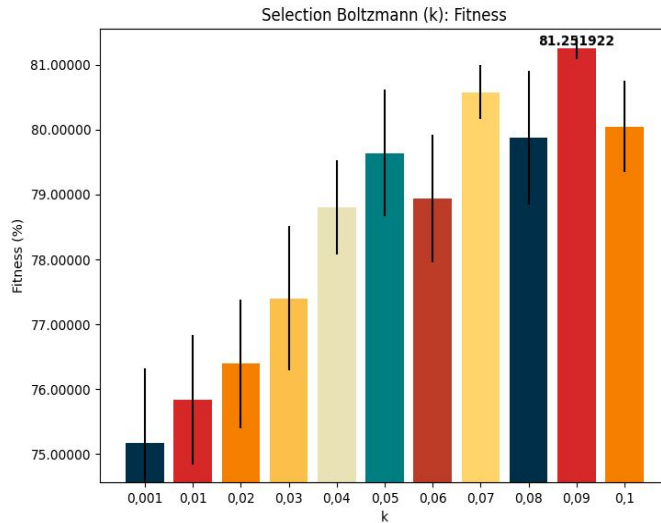


Equilibrando entre Fitness y Execution Time:  **$t_0=20$**

# Seleccion - Boltzmann

Variando los parámetros del método de selección Boltzmann, el mejor resultado es...

- Player: **WIZARD**
- Selection: **[BOLTZMANN( $t_{Critic}=0.05$ ,  $t_0=20$ ,  $k=X$ )]**
- Mutation: **GEN( $p=0.8$ )**
- Crossover: **UNIFORM( $p=0.3$ )**
- Pick: **TRADITIONAL - [ELITE()]**
- ...



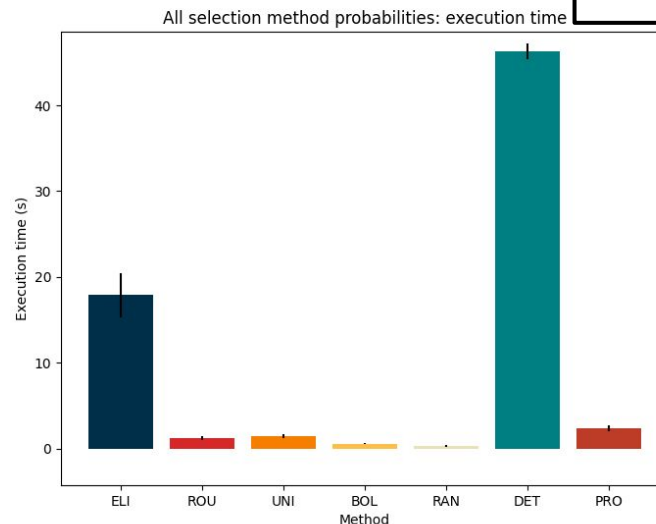
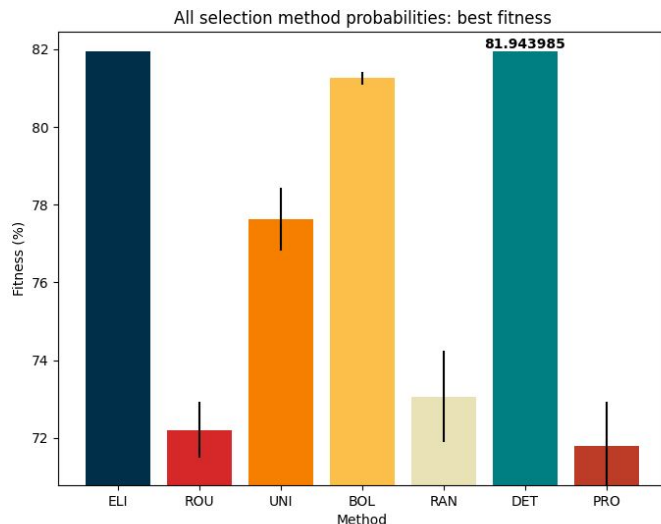
Equilibrando entre Fitness y Execution Time: **k=0.09**



# Método de selección

Elite(), Roulette(), Universal(), Boltzmann(tc=0.02, t0=20, k=0.9), Ranking(), DetTournament(m=25), ProbTournament(th=0.6)

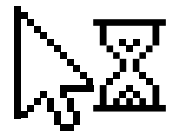
- Player: **WIZARD**
- Selection: **X**
- Mutation: **GEN(p=0.8)**
- Crossover: **UNIFORM(p=0.3)**
- Pick: **TRADITIONAL - [ELITE()]**
- Population Size: **1000**
- Mates per Generation: **100**
- Repeticiones: **10**



Equilibrando entre Fitness y Execution Time: **Elite**

# Métodos de Selección

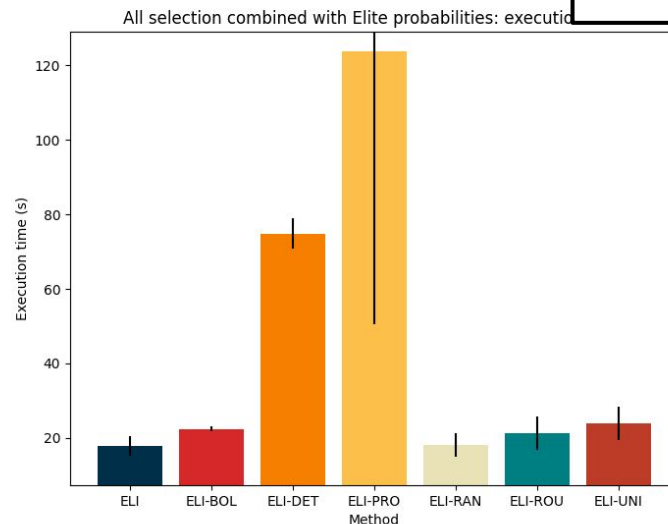
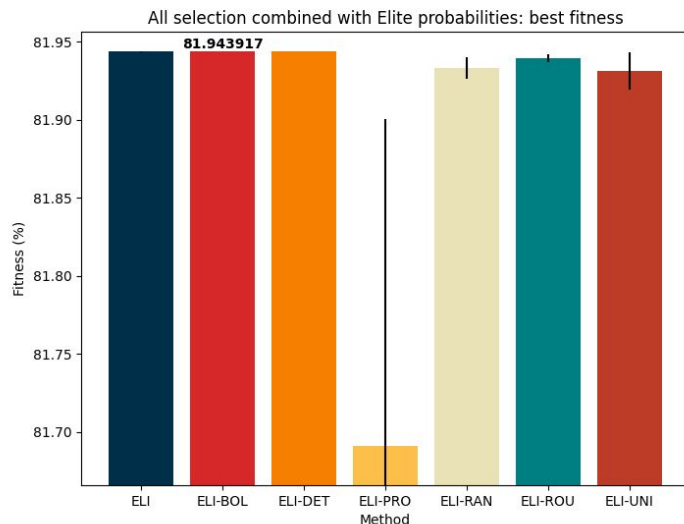
¿Mejora el fitness con más de un método de selección?



# Método de selección

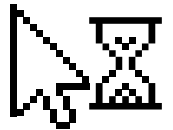
Se propone un análisis de fitness y tiempo de ejecución entre el método Elite y los demás...

- Player: **WIZARD**
- Selection: **[ELITE() - X]**
- Mutation: **GEN(p=0.8)**
- Crossover: **UNIFORM(p=0.3)**
- Pick: **TRADITIONAL - [ELITE()]**
- Population Size: **1000**
- Mates per Generation: **100**
- Repeticiones: **10**



Equilibrando entre Fitness y Execution Time: **Elite**

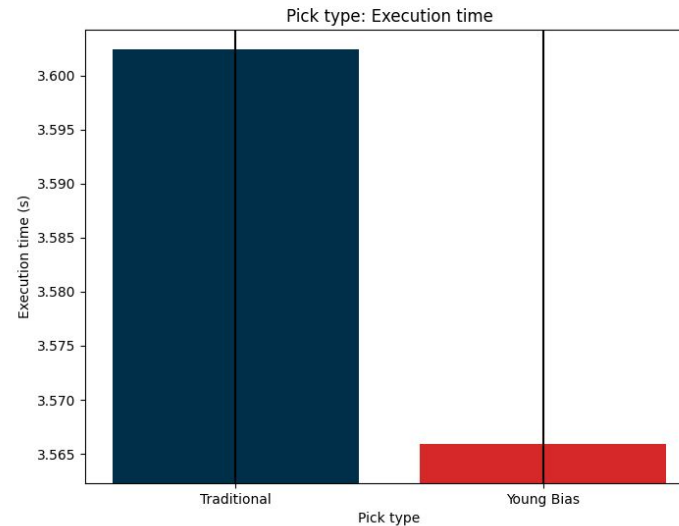
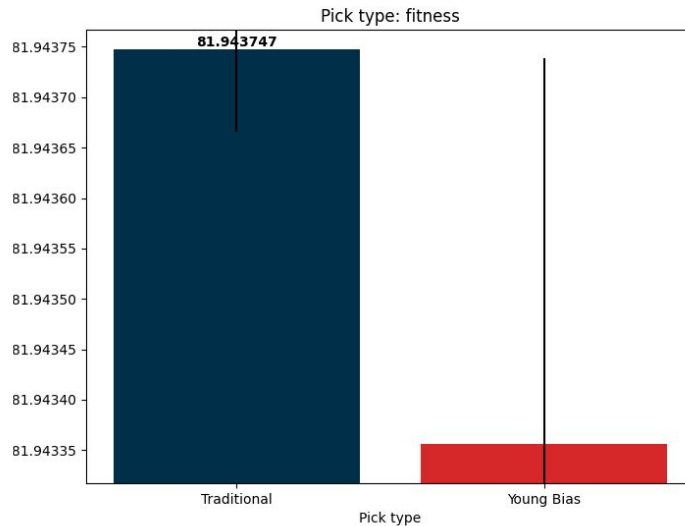
# Métodos de Reemplazo



# Método de reemplazo

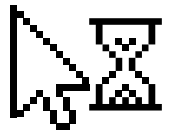
Tradicional, Sesgo de Juventud

- Player: **WIZARD**
- Selection: **[ELITE()]**
- Mutation: **GEN(p=0.8)**
- Crossover: **UNIFORM(p=0.3)**
- Pick: **X - [ELITE()]**
- Population Size: **1000**
- Mates per Generation: **100**
- Repeticiones: **10**



Mejor fitness: **Traditional**

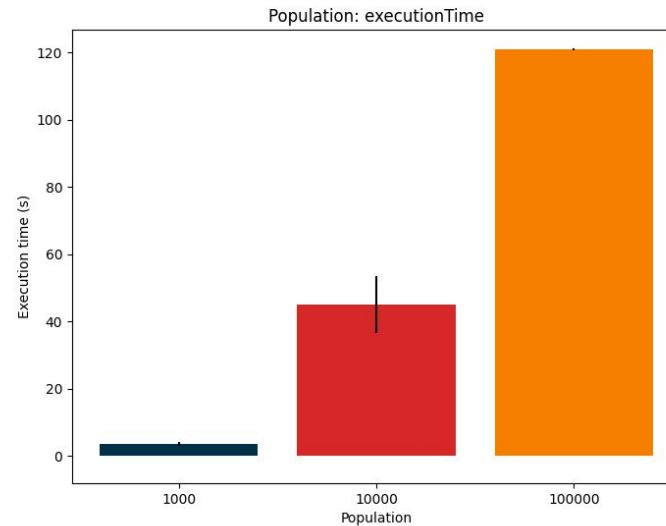
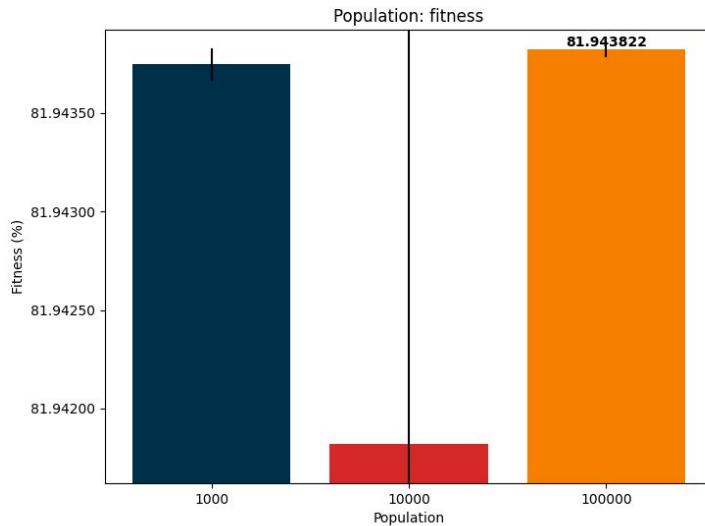
# Tamaño de la población



# Tamaño de la población

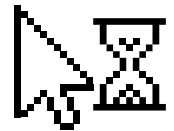
Se varía la cantidad de individuos, en órdenes de magnitud. La cantidad de padres es un porcentaje de N.

- Player: **WIZARD**
- Selection: **[ELITE()]**
- Mutation: **GEN(p=0.8)**
- Crossover: **UNIFORM(p=0.3)**
- Pick: **TRADITIONAL-[ELITE()]**
- Population Size: **X**
- Mates per Generation: **10%X**
- Repeticiones: **10**



Equilibrando entre Fitness y Execution Time: **1000**

Cantidad de  
padres por  
generación

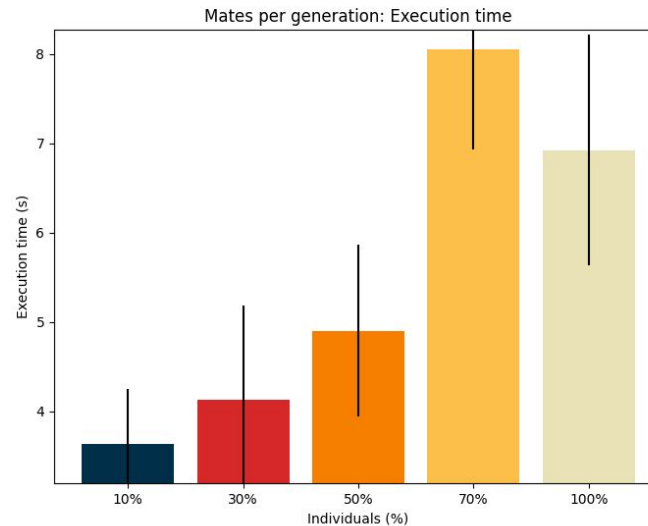
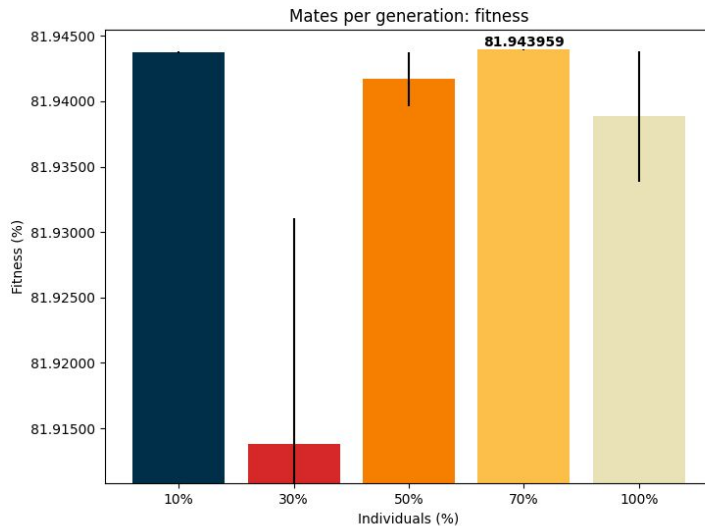




# Cantidad de padres

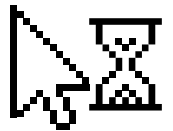
Variamos el porcentaje de la población que se aparearán en cada generación.

- Player: **WIZARD**
- Selection: **[ELITE()]**
- Mutation: **GEN(p=0.8)**
- Crossover: **UNIFORM(p=0.3)**
- Pick: **TRADITIONAL-[ELITE()]**
- Population Size: **1000**
- Mates per Generation: **X**
- Repeticiones: **10**



Equilibrando entre Fitness y Execution Time: **10%**

# Conclusiones



# Los mejores hiperparámetros son:

- Método de selección:
  - **Elite, 100% de la población.**
- Método de cruce:
  - **Uniforme,  $p=0.3$**
- Método de mutación:
  - **Gen,  $p=0.8$**
- Método de reemplazo:
  - **Tradicional - Elite, 100%**
- Tamaño de la población:
  - **1000 individuos**
- Cantidad de padres por generación:
  - **10% de la población**

# Los jugadores para ITBUM ONLINE

[ repartiendo 200 puntos ]



## Warrior



1.915 mts



91



81



28



0



0



61.45752



## Archer



1.915 mts



91



81



28



0



0



92.18625



## Guardian



1.30 mts



0



0



28



81



91



86.75761



## Wizard



1.915 mts



91



81



28



0



0

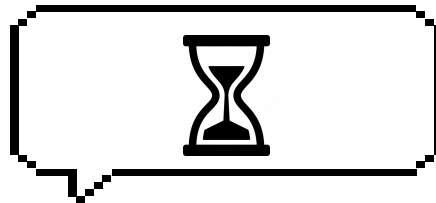


81.94336

**Estos resultados son así por la fórmula de fitness.**

El Warrior, Archer & Wizard le dan mucha importancia al ATAQUE

El Guardian le da mucha importancia a la DEFENSA



# 04

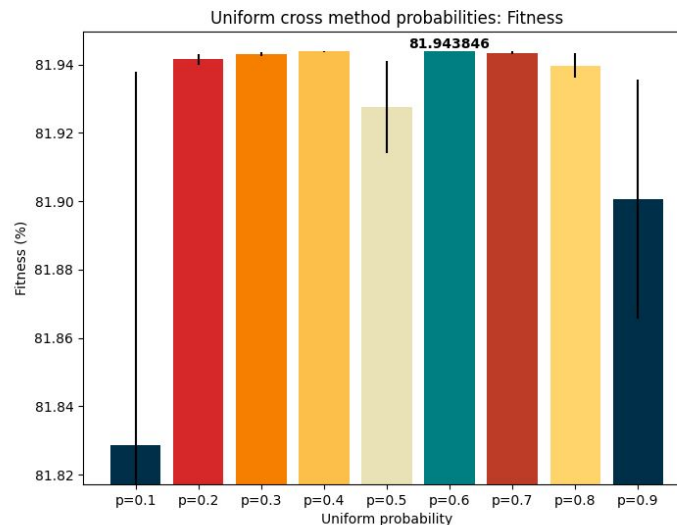
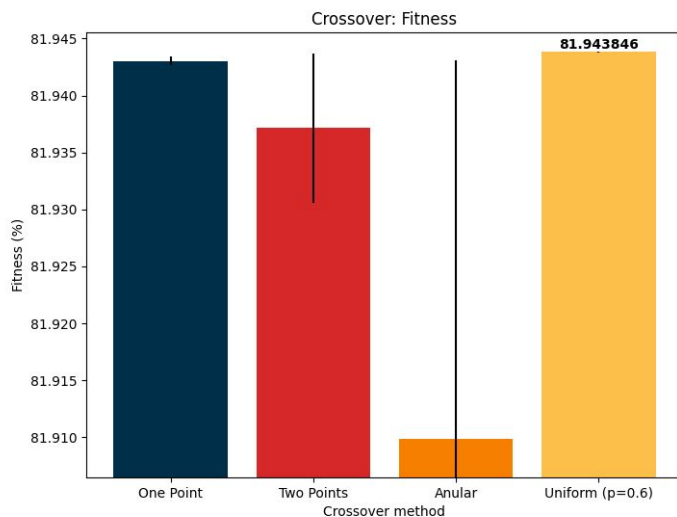
## ¿Y si tenemos mucho más tiempo?

El siguiente análisis **no tiene en cuenta** el tiempo de ejecución: se relaja la restricción del tiempo y tenemos 30 minutos para correr el motor.

# Método de cruza

OnePoint(), TwoPoints(), Anular(), Uniform(p=0.6)

- Player: **WIZARD**
- Selection: **[ELITE()]**
- Mutation: **GEN(p=0.5)**
- Pick: **TRADITIONAL - [ELITE()]**
- Population Size: **1000**
- Mates per Generation: **100**
- Repeticiones: **10**

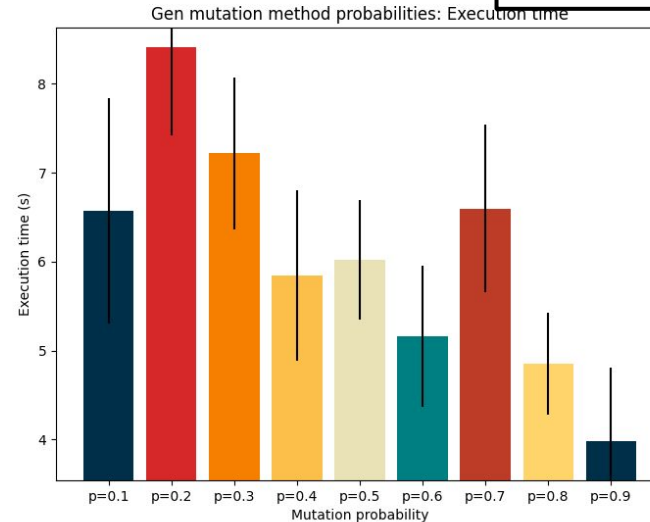
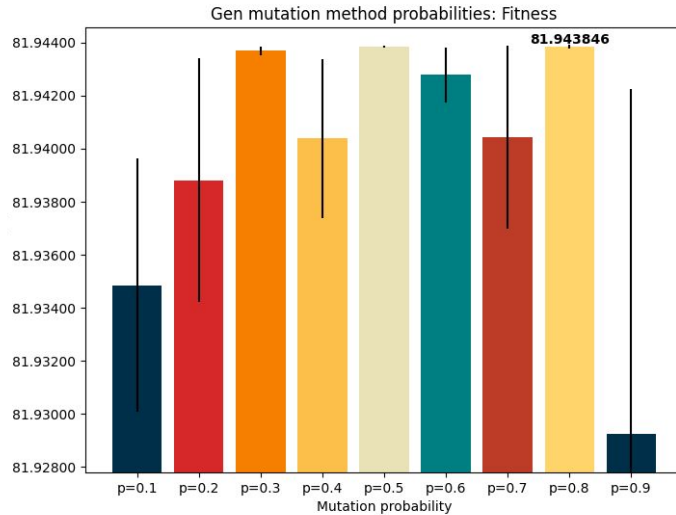


Mejor fitness: **Uniform(p=0.6)**

# Mutación - Gen

Estudiamos la mutación manteniendo el hiperparámetro anterior, Uniform( $p=0.6$ )

- Player: **WIZARD**
- Selection: **[ELITE()]**
- Mutation: **GEN( $p=X$ )**
- Crossover: **UNIFORM( $p=0.6$ )**
- Pick: **TRADITIONAL - [ELITE()]**
- Population Size: **1000**
- Mates per Generation: **100**
- Repeticiones: **10**

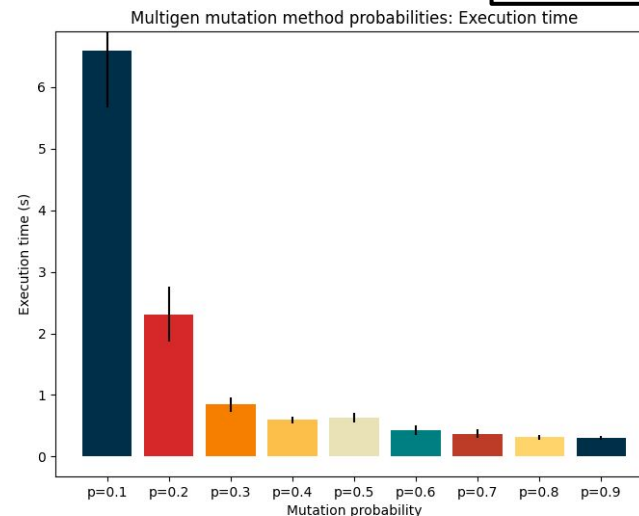
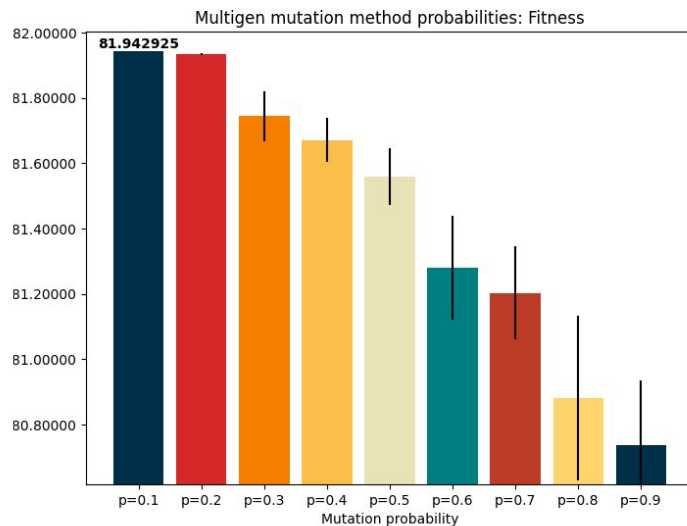


Mejor Fitness: **p=0.8**

# Mutación - Multigen

Estudiamos la mutación manteniendo el hiperparámetro anterior, Uniform( $p=0.6$ )

- Player: **WIZARD**
- Selection: **[ELITE()]**
- Mutation: **MULTIGEN( $p=X$ )**
- Crossover: **UNIFORM( $p=0.6$ )**
- Pick: **TRADITIONAL - [ELITE()]**
- Population Size: **1000**
- Mates per Generation: **100**
- Repeticiones: **10**



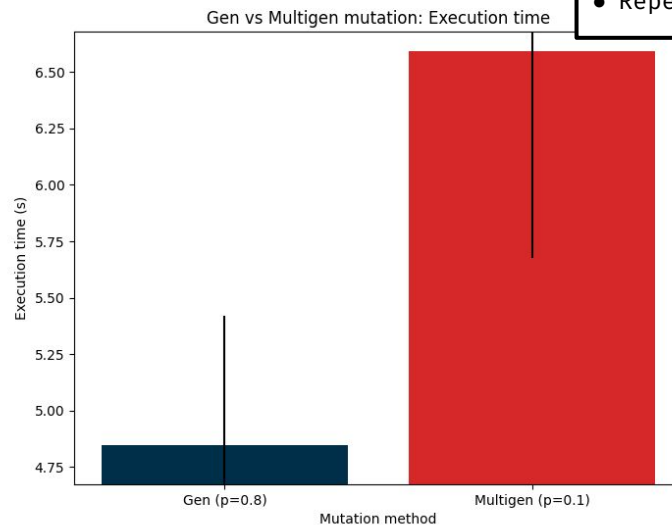
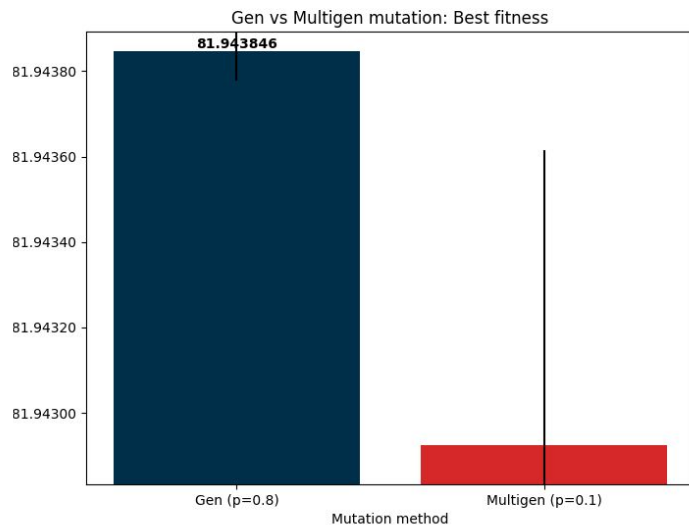
Mejor Fitness: **p=0.1**



# Gen vs Multigen

Comparando el tiempo de ejecución entre los mejores fitness de Gen y Multigen, el mejor resultado es...

- Player: **WIZARD**
- Selection: **[ELITE()]**
- Mutation: **X**
- Crossover: **UNIFORM(p=0.6)**
- Pick: **TRADITIONAL - [ELITE()]**
- Population Size: **1000**
- Mates per Generation: **100**
- Repeticiones: **10**

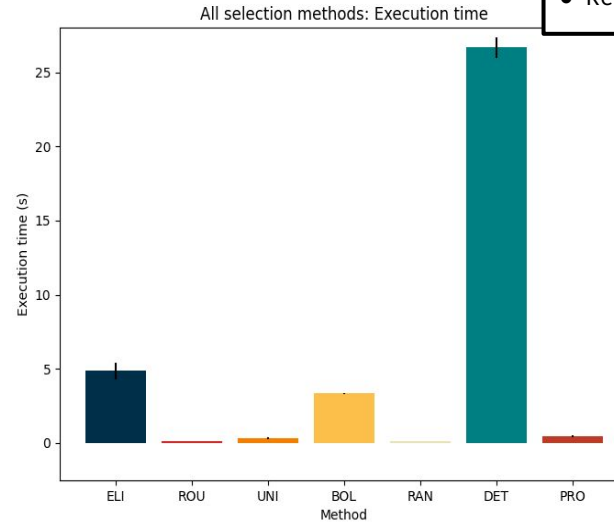
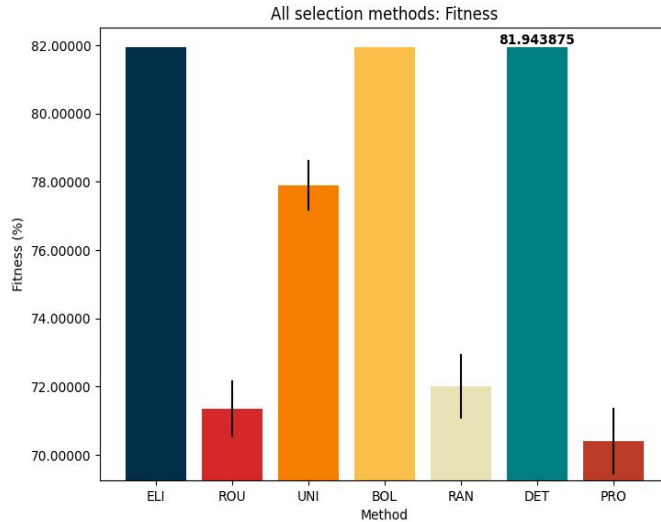


Mejor resultado final: **Gen(p=0.8)**

# Método de selección

Elite(), Roulette(), Universal(), Boltzmann(tc=0.02, t0=20, k=0.9), Ranking(), DetTournament(m=25), ProbTournament(th=0.6)

- Player: **WIZARD**
- Selection: **X**
- Mutation: **GEN(p=0.8)**
- Crossover: **UNIFORM(p=0.6)**
- Pick: **TRADITIONAL - [ELITE()]**
- Population Size: **1000**
- Mates per Generation: **100**
- Repeticiones: **10**

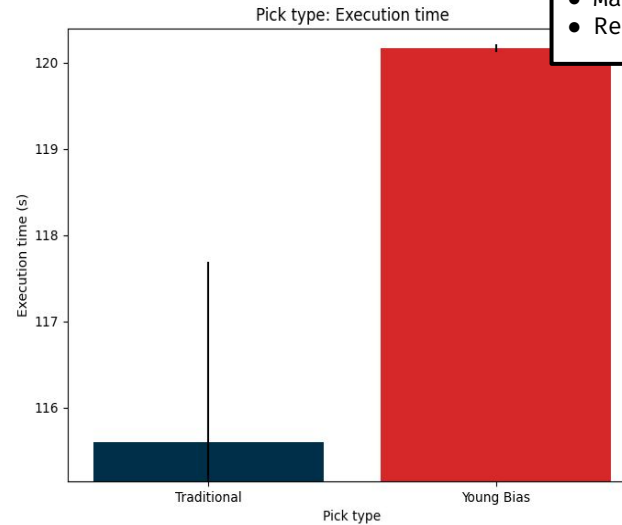
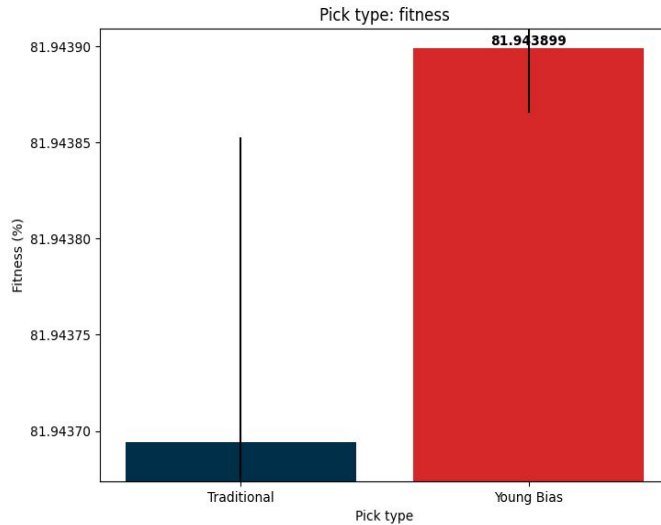


Mejor Fitness: **Deterministic Tournament**

# Método de reemplazo

Tradicional, Sesgo de Juventud

- Player: **WIZARD**
- Selection: **[DETERMINISTIC TOURNAMENT()]**
- Mutation: **GEN(p=0.8)**
- Crossover: **UNIFORM(p=0.6)**
- Pick: **X [DETERMINISTIC TOURNAMENT()]**
- Population Size: **1000**
- Mates per Generation: **100**
- Repeticiones: **10**

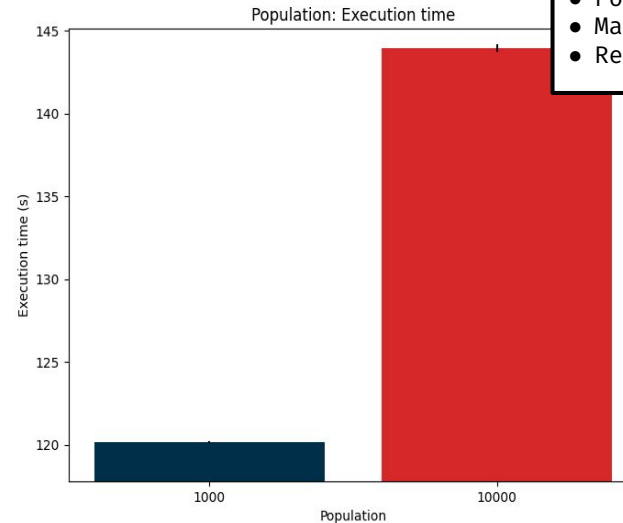
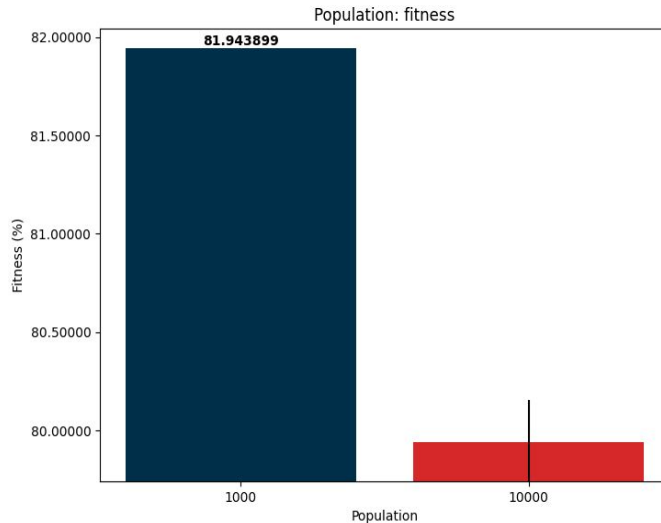


Mejor fitness: **Young Biased**

# Tamaño de la población

Se varía la cantidad de individuos, en órdenes de magnitud. La cantidad de padres es un porcentaje de N.

- Player: **WIZARD**
- Selection: **[DETERMINISTIC TOURNAMENT()]**
- Mutation: **GEN(p=0.8)**
- Crossover: **UNIFORM(p=0.6)**
- Pick: **YOUNG BIASED-[DETERMINISTIC TOURNAMENT()]**
- Population Size: **X**
- Mates per Generation: **10%**
- Repeticiones: **10**

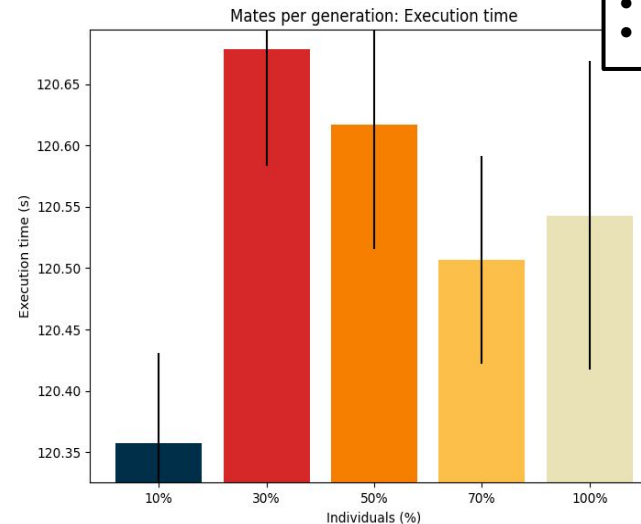
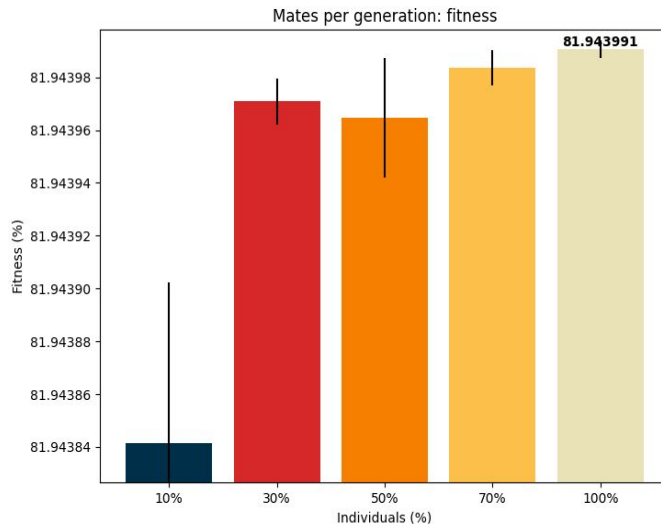


Mejor Fitness: **1000**

# Cantidad de padres

Variamos el porcentaje de la población que se apareará en cada generación.

- Player: **WIZARD**
- Selection: **[DETERMINISTIC TOURNAMENT()]**
- Mutation: **GEN(p=0.8)**
- Crossover: **UNIFORM(p=0.6)**
- Pick: **YOUNG BIASED-DETERMINISTIC TOURNAMENT()**
- Population Size: **1000**
- Mates per Generation: **X%**
- Repeticiones: **10**



Mejor Fitness: **100%**

# Los mejores hiperparámetros son:

[ repartiendo 200 puntos ]

- Método de selección:
  - **Deterministic Tournament, 100%**
- Método de cruza:
  - **Uniforme,  $p=0.6$**
- Método de mutación:
  - **Gen,  $p=0.8$**
- Método de reemplazo:
  - **Young Biased - Deterministic Tournament, 100%**
- Tamaño de la población:
  - **1000 individuos**
- Cantidad de padres por generación:
  - **100% de la población**



## Warrior



61.45752

Variación: 0 %



## Archer



92.18628

Variación: +3e<sup>-5</sup> %



## Guardian



86.75761

Variación: 0 %



## Wizard



81.94336

Variación: 0 %

# ¡Muchas gracias por su atención!

CREDITS: This presentation template was created by  
**Slidesgo**, and includes icons by **Flaticon**, and infographics  
& images by **Freepik**

# Enlaces Útiles

- <https://github.com/alejofl/sia>  
Repositorio del proyecto.
- <https://numpy.org/doc/stable/reference/random/index.html>  
Documentación de Numpy para generador de números aleatorios.
- <https://tour.json-schema.org/>  
Documentación interactiva de JSON Schema.