

# Optimización de composición de equipo en League of Legends utilizando Algoritmo Genético

Daiki Benedek Rueda Tokuhara  
Universidad Nacional Autónoma de México  
Facultad de Ciencias  
daikibenedek@ciencias.unam.mx

**Abstract**—En los últimos años, la escena competitiva de los esports ha crecido exponencialmente y League of Legends se ha consolidado como uno de los más jugados y vistos a nivel mundial. En este juego se enfrentan dos equipos de cinco personas con el objetivo de destruir la base del enemigo. Cada persona tiene un personaje o campeón específico con diferentes estadísticas, habilidades y sinergias con otros personajes y la selección de estos es la pieza fundamental de la competencia.

Este programa optimiza la composición de equipo en League of Legends, combinando los atributos de los campeones y las estrategias de juego a través de un Algoritmo Genético. Su objetivo es generar equipos a partir del conjunto total de opciones disponibles en el juego para encontrar la mejor contraestrategia ante un oponente con un enfoque propuesto en tres estrategias de juego.

## I. GLOSARIO

- Draft:** Selección de los personajes.
- Campeones:** Personajes.
- Ban:** Campeones no utilizables.
- Engage:** Valor para iniciar peleas.
- Poke:** Valor para daño de lejos.
- Team fight:** Valor para pelea prolongada.

## II. INTRODUCCIÓN

En los últimos años, los esports han experimentado un crecimiento exponencial, consolidándose como una de las industrias de entretenimiento más dinámicas del mundo. Entre ellos, League of Legends (LoL) se ha destacado como un referente global, atrayendo a millones de jugadores y espectadores con 6.9 millones solo en la final del mundial del 2024. Una de las tareas más complejas en el competitivo es el draft de campeones, donde los equipos seleccionan y bloquean personajes determinando gran parte del éxito en la partida. Los jugadores y coaches son responsables de poder elegir estos campeones y múltiples veces la partida se gana desde este momento.

El draft no solo implica seleccionar campeones fuertes, sino también prever las elecciones del equipo contrario, contrarrestar estrategias y crear sinergias. Esta toma de decisiones es compleja debido al número de combinaciones posibles el cuál podemos ver como un problema de optimización, lo que la convierte en un problema ideal para abordar con algoritmos genéticos.

En este proyecto, empleamos un algoritmo genético para optimizar el draft competitivo de LoL, maximizando tres métricas de equipo centrales que son "Engage", "Poke" y "Team Fight." Estas tres métricas toma en cuenta todas las estadísticas de los campeones elegidos para la composición y las compara con las del equipo enemigo para encontrar la mejor solución posible.

## III. TRABAJO RELACIONADO

Dentro del staff de equipos profesionales están los coaches y analistas que utilizando la API de League of Legends crean sus propios algoritmos para optimización de distintas cosas y existen muy pocas publicaciones al respecto ya que estos programas son lo que les da el prestigio a estos analistas del competitivo. De igual forma si los otros equipos saben qué utilizan los demás equipos es más fácil crear contraestrategias antes de la competencia ya que contarían con información interna del equipo. De las publicaciones hechas nos centramos en dos, una que utiliza un algoritmo de MinMax y regresión lineal [1] y otro que implementa un algoritmo genético para encontrar una composición óptima pero sin compararla con un oponente [2].

El trabajo de *Oliveira, Placides, Baffa y Machado* [1] propone un modelo computacional utilizando el algoritmo MinMax y regresión lineal para una composición de campeones. Este modelo automáticamente le sugiere al jugador una composición de equipo, campeón por campeón que aumenta la ventaja de nuestro equipo y al mismo tiempo disminuye la del otro. El MinMax es usado para generar las posibles composiciones y la regresión lineal es utilizada para evaluar la composición para saber cuál maximiza el porcentaje de victoria.

*Costa, C. Souza y M. Souza* [2] implementan un algoritmo genético para la generación de una composición de equipo que maximice las tres métricas generales de todo equipo, que son engage, poke y team fight sumando estadísticas individuales de campeón a cada métrica para maximizar las tres. Este algoritmo solo crea composiciones óptimas de cualquier tipo.

Nuestro enfoque es hacia la competencia, el cuál recibe una composición de equipo del oponente e implementa un algoritmo genético guiada por una función fitness que

maximiza las métricas específicas que necesitamos para tener el mejor porcentaje de victoria dependiendo de qué estrategia es la más adecuada para ganar.

#### IV. DESCRIPCIÓN DEL ENFOQUE

Nuestro programa cuenta con una base de datos de los campeones que se utilizaron principalmente en el último torneo mundial de 2024 obtenida de gol.gg [3] que nos da un total de 87 campeones de 168 totales extendida en el archivo *Individuo.java* que toma estadísticas promedio obtenidas de Statcheck.lol [5] que cuenta con toda la información requerida para los campeones. A esto le sumamos una composición generada por el usuario o aleatoriamente y serán nuestros datos de entrada para nuestro algoritmo genético y nos dará una composición con el mejor fitness en contra del oponente. Este proceso está visualizado en la Fig. 1.

Para las métricas individuales de los campeones tomamos las siguientes estadísticas de la Tabla I y asignamos valores máximos para cada una, esto para asignar un porcentaje específico a cada estadística para la función fitness de equipo total. De esta forma nosotros tenemos control de qué estadísticas estamos buscando que haga que mejore o empeore la composición.

Tabla I: Estadísticas individuales

Attributes		Maximum Value	Strategies
Attack damage (AD)		300	Engage, Poke, Team Fight
Ability Power (AP)		500	Engage, Poke, Team Fight
Movement (MS)	Speed	450	Engage
Attack range (R)		1200	Poke
Armor Resist (AR)		300	Team Fight
Magic Resist (MR)		300	Team Fight
Health points (HP)		4000	Team Fight
Attack speed (AS)		2.5	Poke, Team Fight
Crowd Control (CC)		5	Engage

##### A. Función Fitness

Para nuestras funciones fitness, tomaremos las métricas descritas de Engage, Poke y Team Fight junto con las estadísticas individuales y las multiplicaremos por el porcentaje máximo que queremos que represente en la función y la dividiremos por la suma de los valores máximos que pueden tener.



Fig. 1: Proceso del Algoritmo Genético

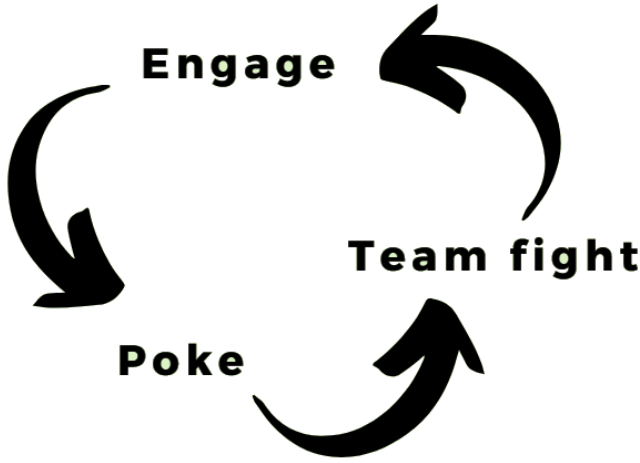


Fig. 2: Triángulo de debilidades

Haremos este proceso cinco veces, una por cada campeón e iremos sumando a la función fitness final. Estas se calcularán con las siguientes funciones:

$$\text{Engage} = \frac{(\text{AD} + \text{AP}) \cdot 40}{800} + \frac{(\text{MS}) \cdot 20}{450} + \frac{(\text{CC}) \cdot 40}{5}$$

$$\text{Poke} = \frac{(\text{AD} + \text{AP}) \cdot 40}{800} + \frac{(\text{R}) \cdot 40}{1200} + \frac{(\text{AS}) \cdot 20}{2.5}$$

$$\text{Team Fight} = \frac{(\text{AD} + \text{AP}) \cdot 30}{800} + \frac{(\text{total AS}) \cdot 10}{2.5} +$$

$$\frac{(\text{AR} + \text{MR}) \cdot 30}{600} + \frac{(\text{HP}) \cdot 30}{4000}$$

Ya sumando las estadísticas de los 5 campeones calcularemos la función fitness total, que nos dice si una solución es mejor que otra comparándola con el oponente. Tomaremos a **Self** como nuestra composición y a **Op** como la del oponente:

$$E = \text{Engage} \mid P = \text{Poke} \mid TF = \text{Team Fight}$$

$$\text{Total} = (\text{Self}(E) - \text{Op}(P)) + (\text{Self}(P) - \text{Op}(TF)) + (\text{Self}(TF) - \text{Op}(E))$$

Esto ya que para las métricas de equipo formamos un triángulo de "debilidades" en contra de la estrategia que tenga un equipo en específico descrito en la Fig. 2.

6	45	67	90	83
				
9	34	66	94	76

Fig. 3: Representación de una solución interna

### B. Representación de las soluciones

En el programa, cada individuo se representa como un arreglo de enteros de tamaño 5, donde cada entero corresponde a un campeón específico. Estos números no se repiten en la solución, y están asociados a los datos de los campeones almacenados en un HashMap para facilitar su acceso y manipulación. La representación interna de cada solución se ilustra en la Figura 3, mientras que su interpretación gráfica se muestra en la Figura 4.

Para incorporar las restricciones del proceso de draft, cada iteración considera a los campeones baneados (prohibidos), que incluyen 10 personajes divididos entre ambos equipos. Esto asegura que los campeones seleccionados cumplan con las reglas del draft. La secuencia de selección de campeones sigue el orden visualizado en la Figura 5, donde los iconos pequeños representan los campeones baneados y los iconos grandes, los seleccionados.

### C. Operador de Cruza

El operador de cruce utilizado es una variante de cruce uniforme. En cada iteración, los genes (campeones) de dos individuos "padres" se combinan con una probabilidad del 50%. La modificación introducida en este operador garantiza que ningún campeón se repita dentro de una misma composición. La Fig. 6 muestra un ejemplo de cómo se aplica esta cruce en la generación de nuevos individuos.

### D. Método de selección

Para la selección de padres, se emplea un método de selección aleatoria uniforme, en el cual todos los individuos de la población tienen la misma probabilidad de ser seleccionados. Este enfoque asegura que cada individuo, independientemente de su fitness, tenga una oportunidad justa de contribuir a la diversidad genética de la población.

### E. Reemplazo

El esquema de reemplazo utilizado es  $(\mu + \lambda)$ , donde tanto los padres como los hijos compiten por un lugar en la nueva generación. De esta manera, solo los individuos con el mayor

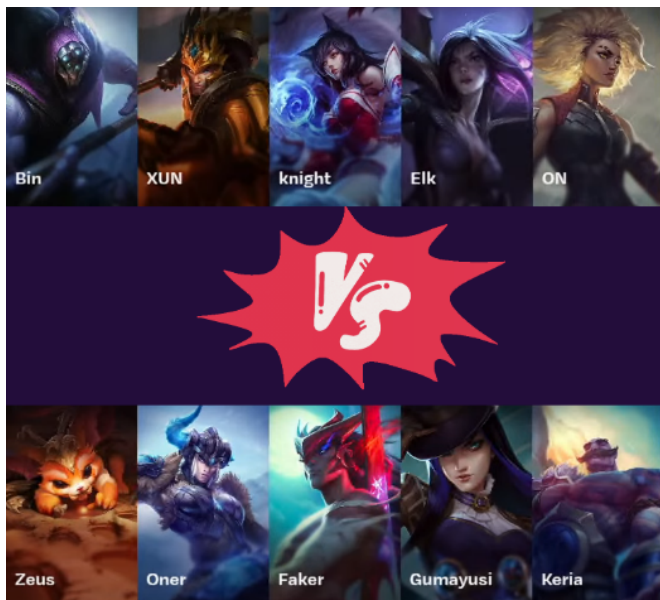


Fig. 4: Representación de solución gráfica

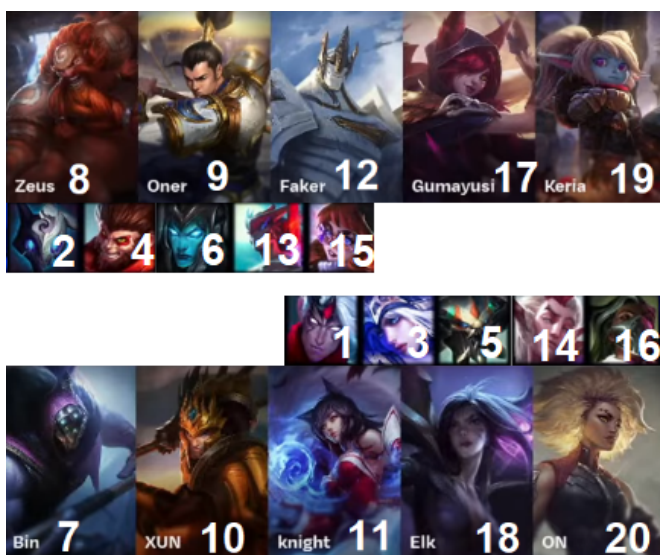


Fig. 5: Orden del draft

fitness se preservan. Este método combina lo mejor de la generación actual y la anterior, acelerando la convergencia hacia soluciones óptimas.

#### F. Mutación

El operador de mutación modifica una solución seleccionando aleatoriamente uno de los genes (campeones) y reemplazándolo por otro que no esté presente en la composición actual con una probabilidad del 10% para escapar de óptimos locales. La Fig. 7 ilustra el proceso de mutación en un individuo.



Fig. 6: Operador de cruce uniforme modificado

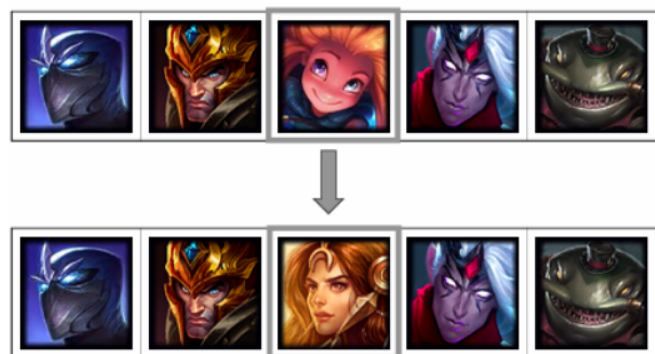


Fig. 7: Mutación

## V. EXPERIMENTACIÓN

Para la experimentación primero mostraremos como se ve una solución gráficamente en nuestro programa después de correr 1 iteración del algoritmo genético (Fig. 8 y Fig. 9).

Para validar la efectividad del algoritmo genético desarrollado, se llevaron a cabo una serie de experimentos enfocados en analizar el impacto de diferentes configuraciones de población y generaciones sobre la calidad de las soluciones obtenidas y el tiempo de convergencia.

#### A. Visualización de las Soluciones

Como punto de partida, se generó una representación gráfica de una solución tras ejecutar una iteración del algoritmo genético (Figuras 8 y 9). Esta representación permite evaluar de manera visual la composición generada y compararla con las restricciones y métricas impuestas en el diseño del problema.



```

Generación: 150 | Mejor Fitness: 96.46 | Peor Fitness: 96.46 | Promedio Fitness: 96.46
=== Resumen Final ===
Total de Generaciones: 150
Número de Individuos por Generación: 100
Tiempo de Ejecución: 0.4140 segundos
La población convergió en la generación: 86
Campeones Baneados:
- Bardo
- Zyra
- Senna
- Jayce
- Camille
- Ziggs
- Caitlyn
- Sejuani
- Nidalee
- Twitch

Oponente | Sett | Lee Sin | Seraphine | Kalista | Renata Glasc |
Poke: 221.07999 Engage: 196.0 Team Fight: 208.04

Mejor al final: | Gragas | Nocturne | Sylas | Ezreal | Lux |
Poke: 275.72 Engage: 229.0 Team Fight: 208.86
Fitness Total: 96.46002

Peor al final: | Gragas | Nocturne | Sylas | Ezreal | Lux |
Poke: 275.72 Engage: 229.0 Team Fight: 208.86
Fitness Total: 96.46002

Mejor de todas las generaciones: | Gragas | Nocturne | Sylas | Ezreal | Lux |
Poke: 275.72 Engage: 229.0 Team Fight: 208.86
Fitness Total: 96.46002

Peor de todas las generaciones: | Urgot | Nunu | Kassadin | Jinx | Alistar |
Poke: 199.63998 Engage: 185.0 Team Fight: 203.31999
Fitness Total: -29.160004

```

Fig. 8: Datos de una iteración del Algoritmo Genético

### B. Configuración Inicial: Alta Población y Pocas Generaciones

En el primer conjunto de experimentos, se utilizó un tamaño de población elevado y un número reducido de generaciones. Los resultados mostraron que, aunque la diversidad inicial de soluciones era alta, la población no convergía eficientemente hacia soluciones óptimas, generando composiciones de menor calidad (Figuras 10 a 13).

### C. Configuración Alternativa: Altas Generaciones y Baja Población

El segundo experimento empleó una configuración opuesta: un tamaño reducido de población con un mayor número de generaciones. En este caso, se observó una rápida convergencia hacia soluciones estables y de alta calidad, aunque con una menor exploración del espacio de búsqueda (Figuras 14 a 17).

### D. Evaluación Final: Relación Óptima entre Individuos y Generaciones

En el último conjunto de pruebas, se fijaron tanto la composición del oponente como los campeones prohibidos (bans), permitiendo comparar directamente el impacto de la relación entre individuos y generaciones. Los resultados indican que una configuración equilibrada, con un tamaño de población moderado y un número de generaciones ligeramente mayor, garantiza una adecuada exploración del espacio de soluciones mientras se mantiene un tiempo de ejecución razonable (Figuras 18 a 25).

Con estas últimas dos vemos que para nuestra implementación de nuestro algoritmo genético nos es más conveniente mantener una relación de individuos y genera-

ciones relativamente cercana, con un ligero incremento en las generaciones para siempre asegurar que nuestro algoritmo encuentra la solución más óptima al oponente.

## VI. CONCLUSIÓN

El objetivo principal de este proyecto era la aplicación real para el trabajo de análisis competitivo de League of Legends, en dónde no hay mucha información pública de programas o algoritmos para mantener la competitividad e información interna de los equipos.

Es complejo el poder comparar nuestras soluciones con la de otros artículos publicados ya que no hay muchos en existencia y todos utilizan un enfoque distinto, no obstante, el proyecto logró integrar datos relevantes de los campeones, las métricas específicas y las restricciones que tiene el competitivo de League of Legends generando soluciones óptimas para contrarrestar las composiciones enemigas y tener una ventaja en la competencia.

En el desarrollo, el mantener un buen equilibrio entre individuos y generaciones es crucial para la optimización del tiempo del algoritmo así como la implementación de nuestro operador de cruza uniforme, la mutación dirigida a un solo gen o campeón y el reemplazo basado en aptitud de selección ( $\mu + \lambda$ ) fue bien aplicada y nuestro algoritmo encuentra soluciones óptimas al hacerle frente a oponentes.

## REFERENCES

- [1] V. d. C. Oliveira, B. J. Placides, M. d. F. O. Baffa, & A. F. d. V. Machada, "A hybrid approach to build automatic team composition in league of legends," in Proceedings of the 16th Brazilian Symposium on Computer Games and Digital Entertainment, ser. SBGames'17, 2017, pp. 630–633.
- [2] Costa, L. M., Souza, A. C. C., & Souza, F. C. M. (2019). An Approach for Team Composition in League of Legends using Genetic Algorithm. En Proceedings of SBGames 2019 - Computing Track – Full Papers (pp. 414-423). XVIII SBGames, Río de Janeiro, Brasil, 28-31 de octubre de 2019.
- [3] gol.gg. (2024). Worlds Main Event 2024 - Tournament Stats. Recuperado el 28 de noviembre de 2024, de <https://gol.gg/tournament/tournament-stats/Worlds>
- [4] op.gg. (n.d.). Champions - Statistics and Analytics. Recuperado el 28 de noviembre de 2024, de <https://www.op.gg/champions>
- [5] Statcheck.lol. (n.d.). League of Legends Statistics and Match Analysis. Recuperado el 28 de noviembre de 2024, de <https://statcheck.lol/>

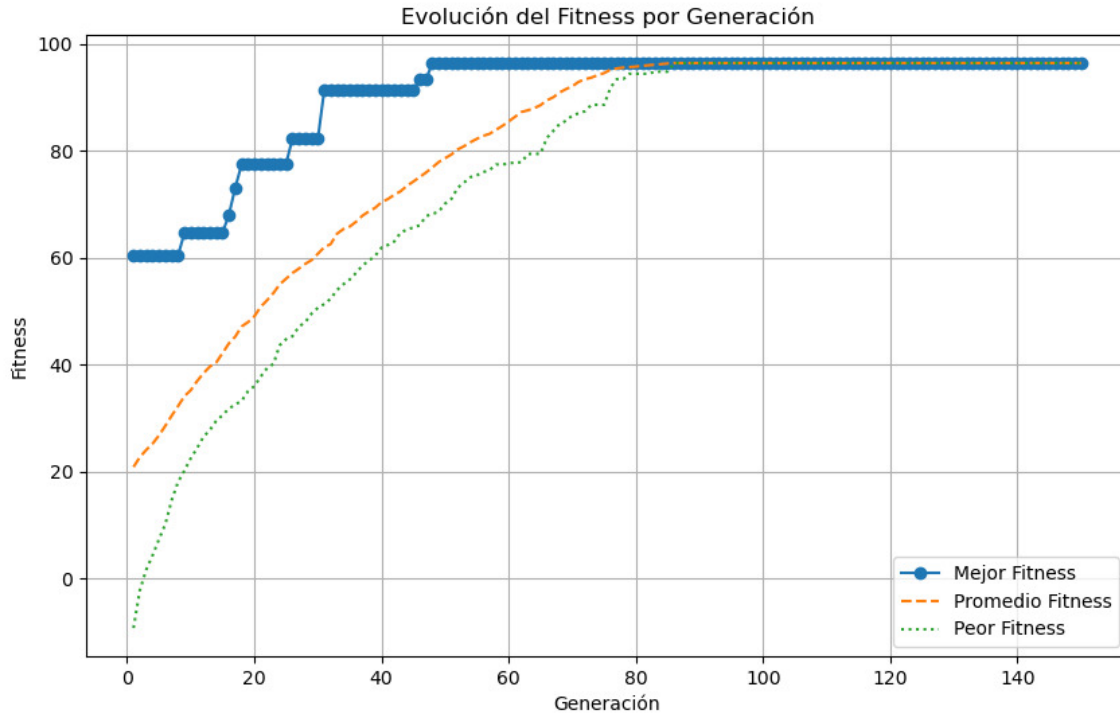


Fig. 9: Gráfica del Algoritmo Genético

```

Generación: 96 | Mejor Fitness: 25.82 | Peor Fitness: -12.90 | Promedio Fitness: -2.26
Generación: 97 | Mejor Fitness: 25.82 | Peor Fitness: -12.64 | Promedio Fitness: -2.06
Generación: 98 | Mejor Fitness: 25.82 | Peor Fitness: -12.24 | Promedio Fitness: -1.79
Generación: 99 | Mejor Fitness: 27.82 | Peor Fitness: -12.18 | Promedio Fitness: -1.53
Generación: 100 | Mejor Fitness: 27.82 | Peor Fitness: -11.88 | Promedio Fitness: -1.21

=== Resumen Final ===
Total de Generaciones: 100
Número de Individuos por Generación: 500
Tiempo de Ejecución: 0.4760 segundos
La población convergió en la generación: 0
Campeones Baneados:
- Draven
- Sylas
- Jax
- Kennen
- Nocturne
- Neeko
- Amumu
- Ezreal
- Sett
- Alistar

Oponente | K'sante | Zyra | Orianna | Ashe | Taric |
Poke: 220.59999 Engage: 242.0 Team Fight: 200.3

Mejor al final: | Gragas | Nidalee | Ryze | Zeri | Lux |
Poke: 264.144 Engage: 227.0 Team Fight: 199.57199
Fitness Total: 27.81601

Peor al final: | Sion | Lee Sin | Cassiopeia | Miss Fortune | Lux |
Poke: 233.68 Engage: 207.0 Team Fight: 210.34
Fitness Total: -11.880005

```

Fig. 10: Generaciones: 100, Individuos: 500

```

Generación: 96 | Mejor Fitness: 69.08 | Peor Fitness: 24.16 | Promedio Fitness: 36.69
Generación: 97 | Mejor Fitness: 69.08 | Peor Fitness: 24.48 | Promedio Fitness: 37.00
Generación: 98 | Mejor Fitness: 69.08 | Peor Fitness: 24.72 | Promedio Fitness: 37.33
Generación: 99 | Mejor Fitness: 69.08 | Peor Fitness: 24.90 | Promedio Fitness: 37.52
Generación: 100 | Mejor Fitness: 69.08 | Peor Fitness: 25.22 | Promedio Fitness: 37.72

=== Resumen Final ===
Total de Generaciones: 100
Número de Individuos por Generación: 500
Tiempo de Ejecución: 0.5200 segundos
La población convergió en la generación: 0
Campeones Baneados:
- Brand
- Bard
- LeeBlanc
- Yasuo
- Neeko
- Olaf
- Syndra
- Fiddlesticks
- Urgot
- Zeri

Oponente | Rumble | Maokai | Smolder | Draven | Nautilus |
Poke: 229.4 Engage: 202.0 Team Fight: 200.7

Mejor al final: | Kennen | Amumu | Sylas | Ezreal | Lulu |
Poke: 271.12 Engage: 217.0 Team Fight: 213.06
Fitness Total: 69.08

Peor al final: | Gnar | Lee Sin | Galio | Ziggs | Senna |
Poke: 236.88 Engage: 214.0 Team Fight: 206.44
Fitness Total: 25.220016

```

Fig. 11: Generaciones: 100, Individuos: 500

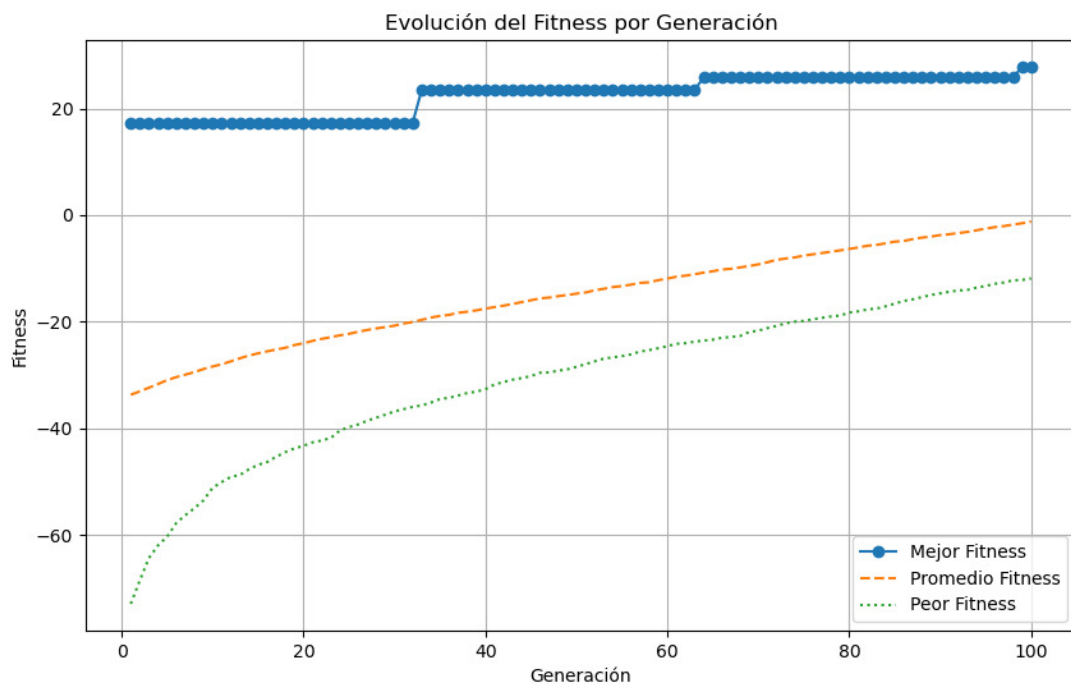


Fig. 12: Tabla de Fig. 10

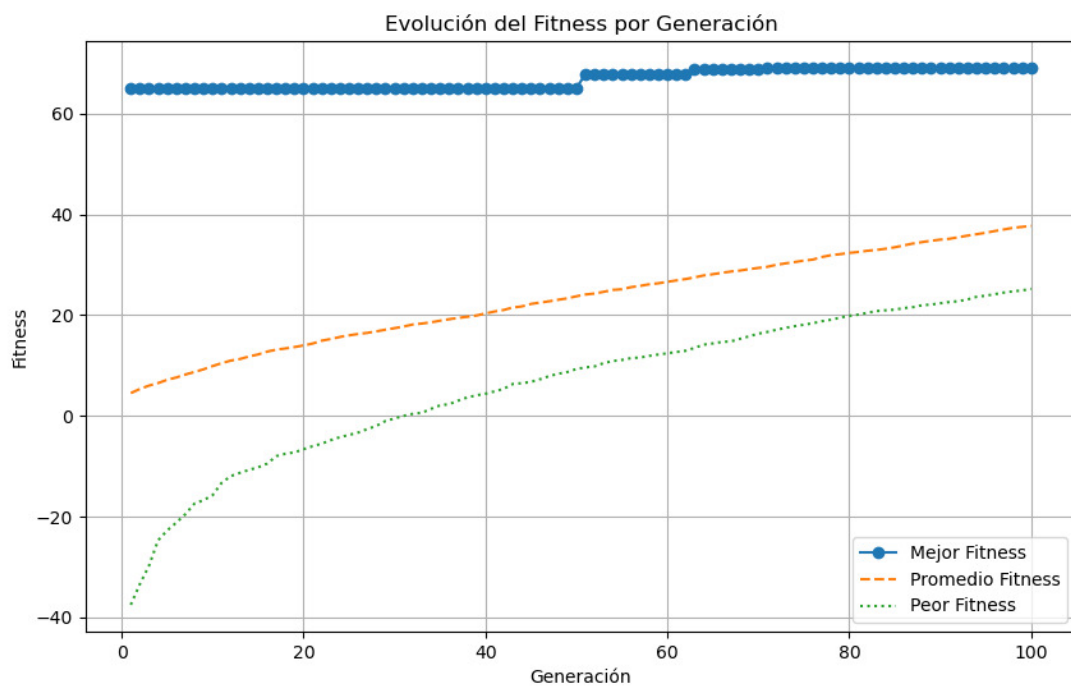


Fig. 13: Tabla de Fig. 11

```

Generación: 500 | Mejor Fitness: 81.62 | Peor Fitness: 81.62 | Promedio Fitness: 81.62

=== Resumen Final ===
Total de Generaciones: 500
Número de Individuos por Generación: 100
Tiempo de Ejecución: 0.4290 segundos
La población convergió en la generación: 92
Campeones Baneados:
- Hwei
- Neeko
- Nidalee
- Darius
- Akali
- Bard
- Lux
- Kennen
- Braum
- Ezreal

Oponente | Rumble | Kindred | Galio | Miss Fortune | Taric |
Poke: 212.23999 Engage: 206.0 Team Fight: 208.12

Mejor al final: | Gragas | Nocturne | Sylas | Zeri | Lulu |
Poke: 266.32 Engage: 234.0 Team Fight: 207.66
Fitness Total: 81.620026

```

Fig. 14: Generaciones: 500, Individuos: 100

```

Generación: 500 | Mejor Fitness: 103.22 | Peor Fitness: 103.22 | Promedio Fitness: 103.22

=== Resumen Final ===
Total de Generaciones: 500
Número de Individuos por Generación: 100
Tiempo de Ejecución: 0.4600 segundos
La población convergió en la generación: 78
Campeones Baneados:
- Zeri
- Sejuani
- Brand
- Morgana
- Lillia
- Tristana
- Nidalee
- Fiddlesticks
- Gragas
- Gnar

Oponente | Jayce | Kindred | Galio | Jinx | Braum |
Poke: 204.23999 Engage: 205.0 Team Fight: 198.12

Mejor al final: | Kennen | Nocturne | Sylas | Ezreal | Lux |
Poke: 277.72 Engage: 223.0 Team Fight: 209.86
Fitness Total: 103.22002

```

Fig. 15: Generaciones: 500, Individuos: 100



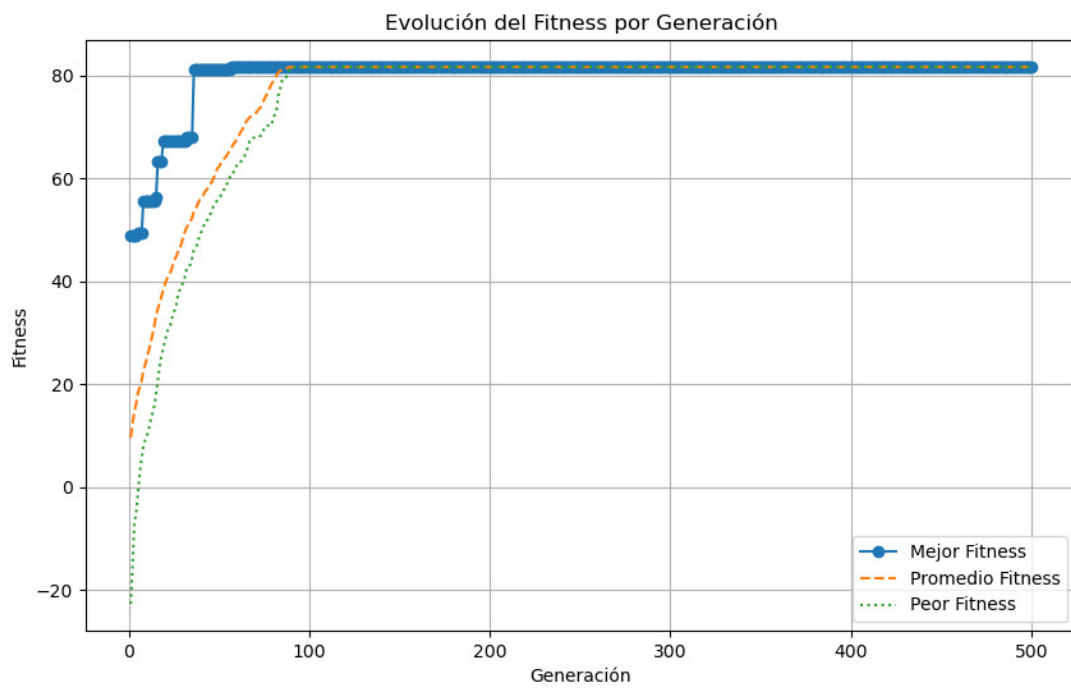


Fig. 16: Tabla de Fig. 14

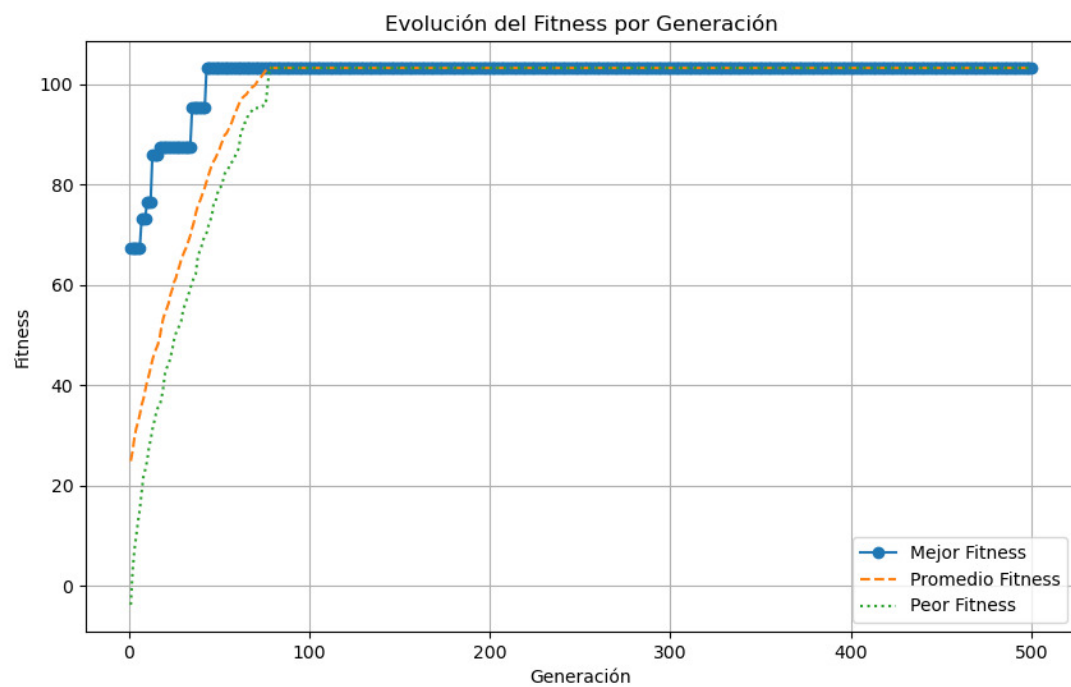


Fig. 17: Tabla de Fig. 15

```

Generación: 96 | Mejor Fitness: 35.40 | Peor Fitness: -7.36 | Promedio Fitness: 4.00
Generación: 97 | Mejor Fitness: 35.40 | Peor Fitness: -7.04 | Promedio Fitness: 4.26
Generación: 98 | Mejor Fitness: 35.40 | Peor Fitness: -6.84 | Promedio Fitness: 4.48
Generación: 99 | Mejor Fitness: 35.40 | Peor Fitness: -6.72 | Promedio Fitness: 4.60
Generación: 100 | Mejor Fitness: 35.40 | Peor Fitness: -6.40 | Promedio Fitness: 4.94

=== Resumen Final ===
Total de Generaciones: 100
Número de Individuos por Generación: 500
Tiempo de Ejecución: 0.6760 segundos
La población convergió en la generación: 0
Campeones Baneados:
- Bardo
- Zyra
- Senna
- Jayce
- Camille
- Ziggs
- Caitlyn
- Sejuani
- Nidalee
- Twitch

Oponente | Sett | Lee Sin | Seraphine | Kalista | Renata Glasc |
Poke: 250.12 Engage: 208.0 Team Fight: 208.56

Mejor al final: | Gnar | Amumu | Sylas | Ezreal | Lulu |
Poke: 274.72 Engage: 217.0 Team Fight: 210.36
Fitness Total: 35.40001

Peor al final: | Gnar | Amumu | Syndra | Zeri | Tahm Kench |
Poke: 243.51999 Engage: 213.0 Team Fight: 203.76
Fitness Total: -6.400009

```

Fig. 18: Solución fija - Generaciones: 100, Individuos: 500

```

Generación: 150 | Mejor Fitness: 96.46 | Peor Fitness: 96.46 | Promedio Fitness: 96.46

=== Resumen Final ===
Total de Generaciones: 150
Número de Individuos por Generación: 100
Tiempo de Ejecución: 0.4140 segundos
La población convergió en la generación: 86
Campeones Baneados:
- Bardo
- Zyra
- Senna
- Jayce
- Camille
- Ziggs
- Caitlyn
- Sejuani
- Nidalee
- Twitch

Oponente | Sett | Lee Sin | Seraphine | Kalista | Renata Glasc |
Poke: 221.07999 Engage: 196.0 Team Fight: 200.04

Mejor al final: | Gragas | Nocturne | Sylas | Ezreal | Lux |
Poke: 275.72 Engage: 229.0 Team Fight: 208.86
Fitness Total: 96.46002

Peor al final: | Gragas | Nocturne | Sylas | Ezreal | Lux |
Poke: 275.72 Engage: 229.0 Team Fight: 208.86
Fitness Total: 96.46002

Mejor de todas las generaciones: | Gragas | Nocturne | Sylas | Ezreal | Lux |
Poke: 275.72 Engage: 229.0 Team Fight: 208.86
Fitness Total: 96.46002

Peor de todas las generaciones: | Urgot | Nunu | Kassadin | Jinx | Alistar |
Poke: 199.63998 Engage: 185.0 Team Fight: 203.31999
Fitness Total: -29.160004

```

Fig. 20: Solución fija - Generaciones: 150, Individuos: 100

```

Generación: 500 | Mejor Fitness: 77.42 | Peor Fitness: 77.42 | Promedio Fitness: 77.42

=== Resumen Final ===
Total de Generaciones: 500
Número de Individuos por Generación: 100
Tiempo de Ejecución: 0.7380 segundos
La población convergió en la generación: 98
Campeones Baneados:
- Bardo
- Zyra
- Senna
- Jayce
- Camille
- Ziggs
- Caitlyn
- Sejuani
- Nidalee
- Twitch

Oponente | Sett | Lee Sin | Seraphine | Kalista | Renata Glasc |
Poke: 223.44 Engage: 213.0 Team Fight: 199.72

Mejor al final: | Gragas | Nocturne | Sylas | Ezreal | Lux |
Poke: 275.72 Engage: 229.0 Team Fight: 208.86
Fitness Total: 77.42

```

Fig. 19: Solución fija - Generaciones: 500, Individuos: 100

```

Generación: 2000 | Mejor Fitness: 82.86 | Peor Fitness: 82.86 | Promedio Fitness: 82.86

=== Resumen Final ===
Total de Generaciones: 2000
Número de Individuos por Generación: 1500
Tiempo de Ejecución: 97.7540 segundos
La población convergió en la generación: 1358
Campeones Baneados:
- Bardo
- Zyra
- Senna
- Jayce
- Camille
- Ziggs
- Caitlyn
- Sejuani
- Nidalee
- Twitch

Oponente | Sett | Lee Sin | Seraphine | Kalista | Renata Glasc |
Poke: 233.48 Engage: 195.0 Team Fight: 202.24

Mejor al final: | Gragas | Nocturne | Sylas | Ezreal | Lux |
Poke: 275.72 Engage: 229.0 Team Fight: 208.86
Fitness Total: 82.86

```

Fig. 21: Solución fija - Generaciones: 2000, Individuos: 1500

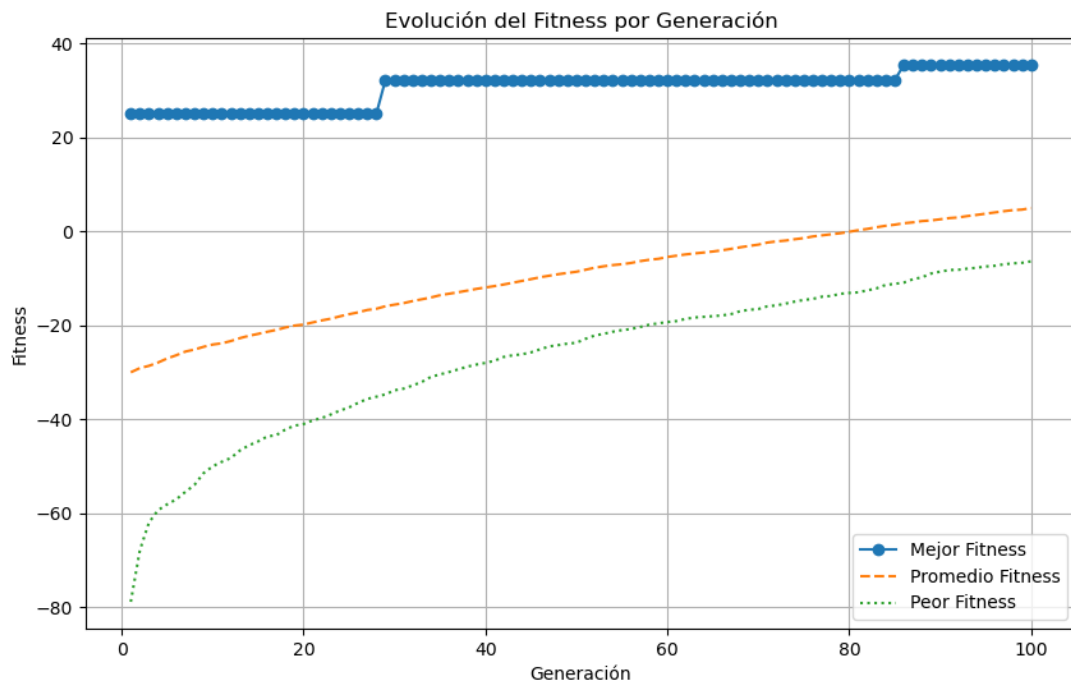


Fig. 22: Tabla de Fig. 18

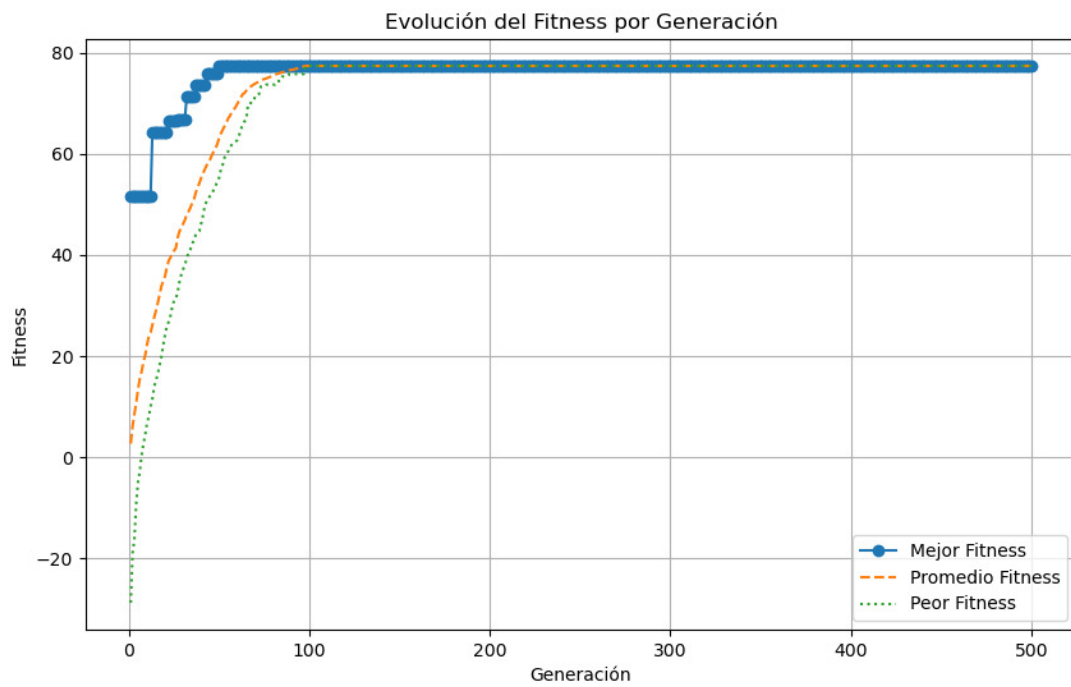


Fig. 23: Tabla de Fig. 19

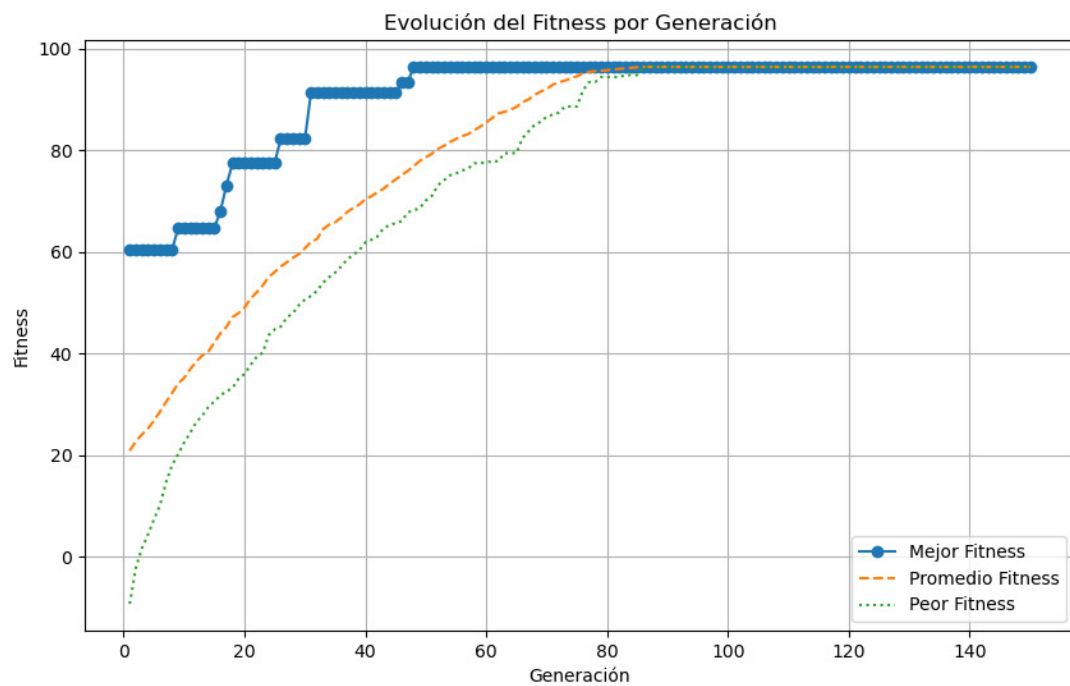


Fig. 24: Tabla de Fig. 20

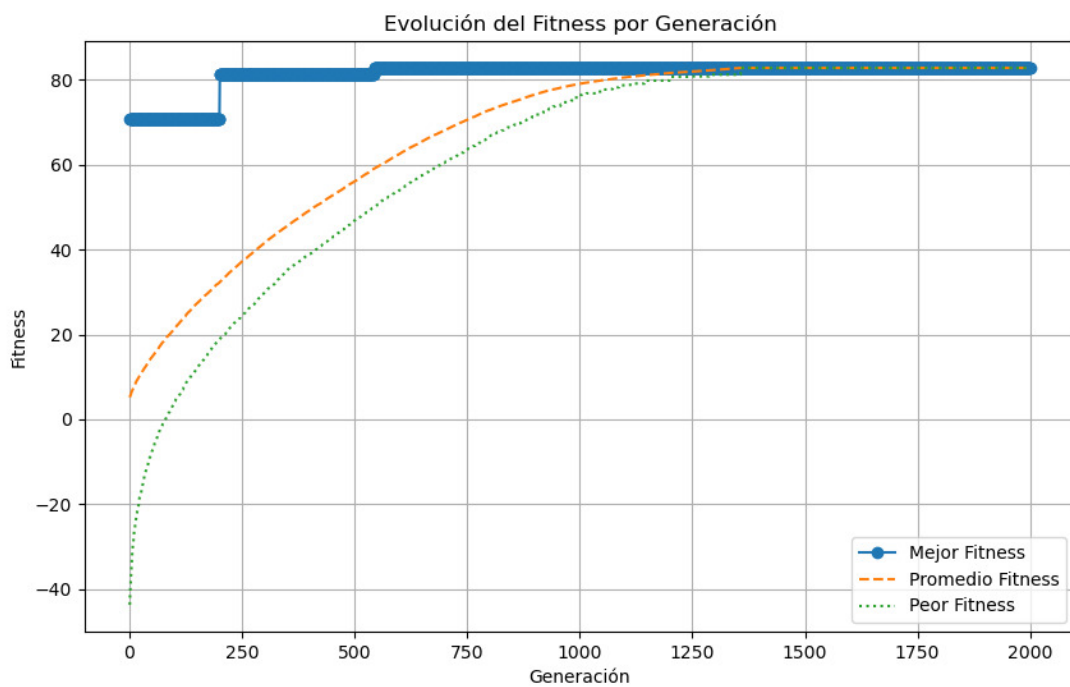


Fig. 25: Tabla de Fig. 21