

## Información General

Curso : Cómputo Evolutivo  
Semestre : 2025 - 2  
Profesores : Katya Rodríguez Vázquez  
: Augusto César Poot Hernández  
Entrega : Marzo 18 de 2025  
Alumno : Pablo Uriel Benítez Ramírez, 418003561

## Objetivo

El objetivo de esta tarea es la comparación entre las codificaciones: Binaria, Real y Entera.

## Comparación de métodos de codificación

**Definición 0.1.** La *codificación binaria* representa soluciones usando cadenas de bits (0's y 1's).

- *Ventajas*
  - Eficiencia en términos de memoria y operaciones lógicas.
  - Permite aplicar operadores genéticos simples (cruce y mutación).
  - Compatible con muchas técnicas clásicas de optimización.
- *Desventajas*
  - Puede requerir una decodificación antes de evaluar la función objetivo.
  - La resolución está limitada por el número de bits utilizados.
  - La representación binaria puede llevar a la pérdida de precisión en problemas con soluciones continuas.

**Definición 0.2.** La *codificación real* representa directamente los valores de la solución como números reales.

- *Ventajas*
  - Evita la necesidad de decodificación.
  - Representa soluciones con mayor precisión.
- *Desventajas*
  - Puede ser más propensa a la convergencia prematura.
  - La manipulación de valores reales requiere más procesamiento que la manipulación de bits.

**Definición 0.3.** La *codificación entera* representa soluciones con valores enteros dentro de un rango específico.

- *Ventajas*
  - Adecuada para problemas donde las variables son inherentemente discretas.
  - Se pueden usar operadores genéticos específicos como permutaciones en problemas combinatorios.
- *Desventajas*
  - Dependiendo del problema, algunos operadores genéticos pueden ser ineficientes.
  - No es adecuada para problemas de optimización continua.

<b>Característica</b>	<b>Binaria</b>	<b>Real</b>	<b>Entera</b>
<b>Espacio de búsqueda</b>	Discreto	Continuo	Discreto
<b>Precisión</b>	Depende del número de bits	Alta	Depende del rango
<b>Uso común</b>	Problemas lógicos, combinatorios	Optimización continua	Problemas combinatorios
<b>Facilidad de implementación</b>	Alta	Media	Media
<b>Eficiencia computacional</b>	Alta en términos de memoria	Menos eficiente (operaciones con flotantes)	Similar a binaria

Cuadro 1: Comparación entre codificaciones Binaria, Real y Entera

## Implementación

Se consideran las siguientes características:

- Se considera fija la probabilidad de cruce y el número de individuos.
- Criterio de paro
  - 50 generaciones sin cambiar
  - llegar al umbral esperado
  - llegar al máximo de generaciones
- Se utiliza la selección por Torneo binario.
- Elitismo al 20 %

## Esfera

$$f(\mathbf{x}) = \sum_{i=1}^n x_i^2 \quad -10 \leq x \leq 10$$

$$f(x_1, \dots, x_n) = f(0, \dots, 0) = 0$$

## Codificación binaria

### Parámetros

```
m = 15                # longitud del genotipo
pob_size = 50          # tamaño de la población
a = -10                # valor mínimo de x
b = 10                 # valor máximo de x
decimales=3           # número de decimales
n = 2                  # variables a considerar
t = m * n              # total de bits
epsilon = 1e-6         # cota de selección
proba_cruza = 0.8       # probabilidad de cruza
proba_muta = 1/t        # probabilidad de mutación
generaciones = 100     # número de generaciones
etapas = 5             # número de etapas
umbral = 1e-8          # umbral de paro
max_sin_mejora = 50     # max de generaciones sin mejora
```

$n = 2$

Promedio de convergencia: 33.7 generaciones

La iteracion que tiene la minima evaluacion es: 1 con 20 generaciones.

Iteracion 1:

Total de generaciones: 20

Mejor genotipo: [0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1]

Mejor fenotipo(x1,...,xn): [-0. -0.]

Mejor evaluacion (x1,...,xn): 0.0

$n = 5$

Promedio de convergencia: 115.6 generaciones

La iteracion que tiene la minima evaluacion es: 5 con 102 generaciones.

Iteracion 5:

Total de generaciones: 102

Mejor genotipo: [1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1  
1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0]

Mejor fenotipo(x1,...,xn): [ 0. 0. -0. 0. 0.]

Mejor evaluacion (x1,...,xn): 0.0

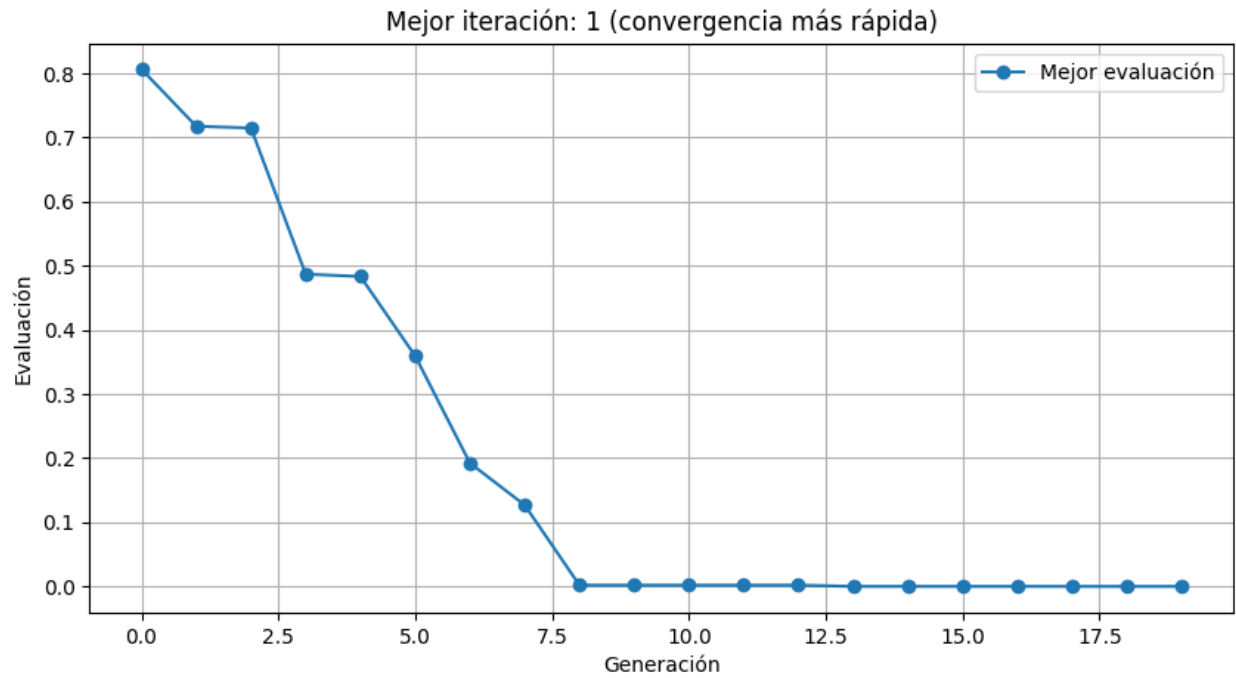


Figura 1: Codificación binaria Esfera  $n = 2$

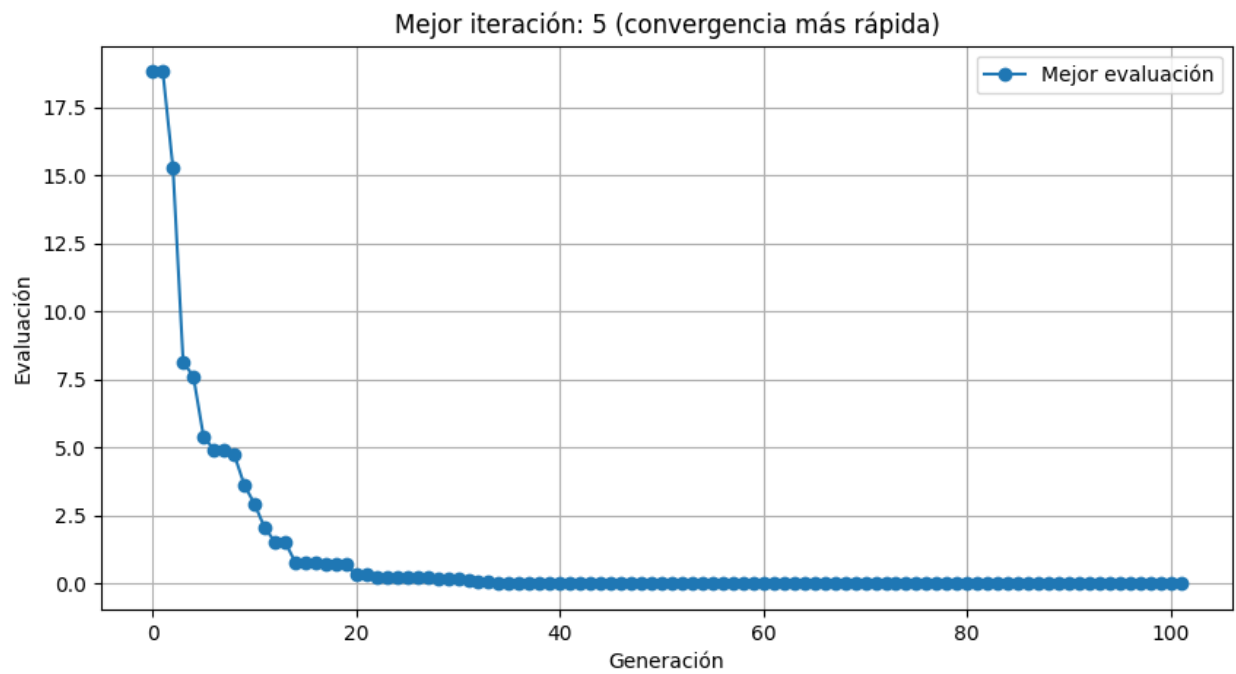


Figura 2: Codificación binaria Esfera  $n = 5$

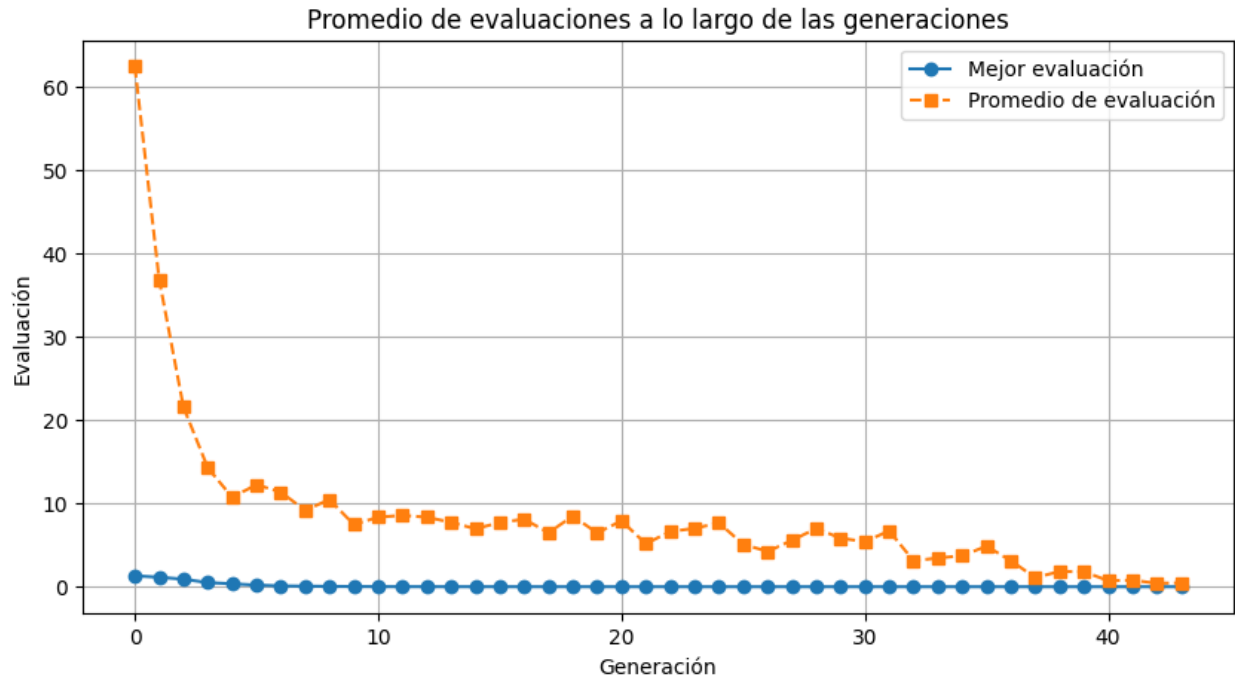


Figura 3: Promedio codificación binaria Esfera  $n = 2$

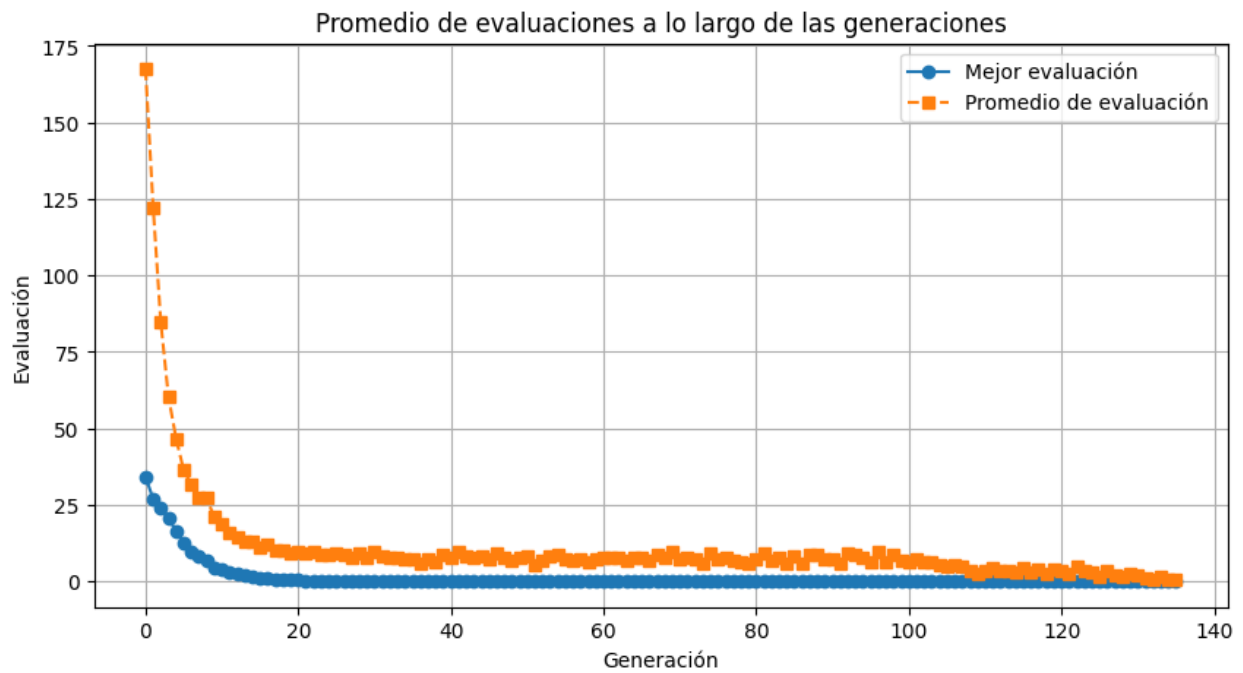



Figura 4: Promedio codificación binaria Esfera  $n = 5$

**Código:**  *Nombre de archivo:* Cod binaria.ipynb, con la función *Esfera* activada. Además de respetar los valores de  $n = 2$  o  $n = 5$  dependiendo del caso.

## Codificación real

### Parámetros

```
pob_size = 50      # tamaño de la población
a = -10            # valor mínimo de x
b = 10             # valor máximo de x
decimales = 3      # número de decimales
n = 2              # variables a considerar
epsilon = 1e-6     # cota de selección
proba_cruza = 0.8   # probabilidad de cruza
proba_muta = 0.2    # probabilidad de mutación
generaciones = 100 # número de generaciones
etapas = 5          # número de etapas
umbral = 1e-8       # umbral de paro
max_sin_mejora = 50 # max de generaciones sin mejora
```

$n = 2$

Promedio de convergencia: 67.2 generaciones

La iteracion que tiene la minima evaluacion es: 1 con 92 generaciones.

Iteracion 1:

Total de generaciones: 92

Mejor fenotipo (x1,...,xn): [ 0.001 -0.002]

Mejor evaluacion f(x1,...,xn): 0.0

$n = 5$

Promedio de convergencia: 114.1 generaciones

La iteracion que tiene la minima evaluacion es: 1 con 98 generaciones.

Iteracion 1:

Total de generaciones: 98

Mejor fenotipo (x1,...,xn): [ 0.018 0.003 -0.026 0.034 -0.002]

Mejor evaluacion f(x1,...,xn): 0.002



Figura 5: Codificación real Esfera  $n = 2$

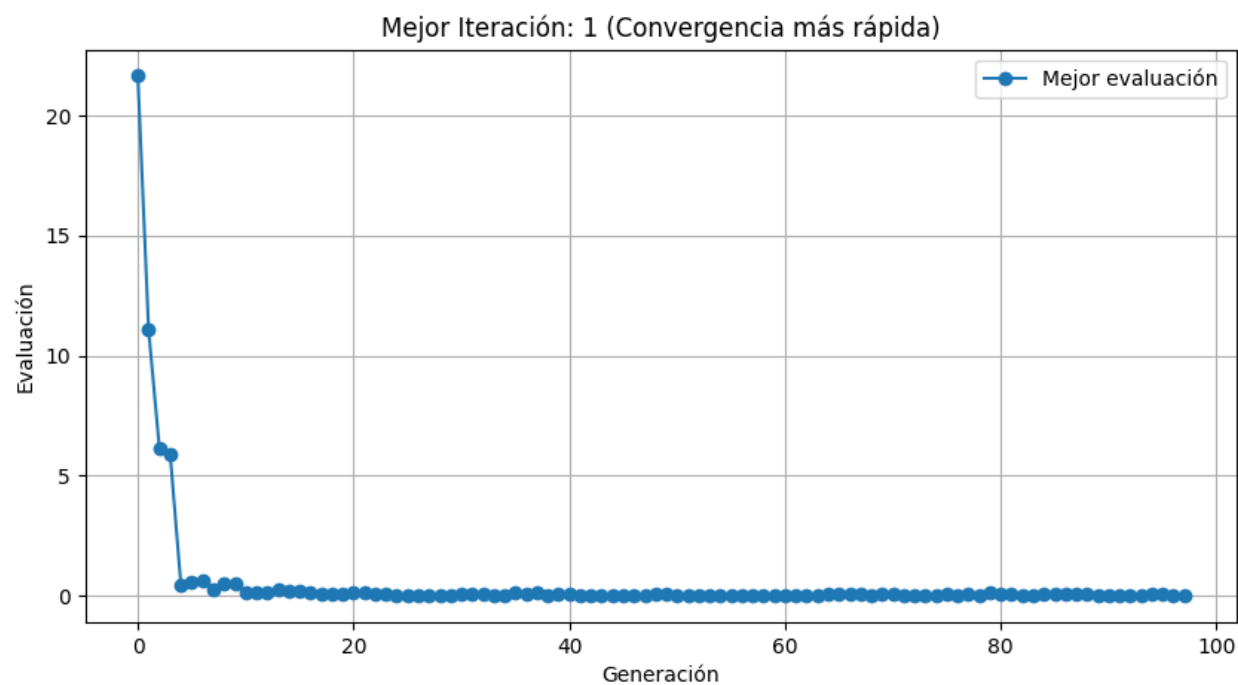


Figura 6: Codificación real Esfera  $n = 5$

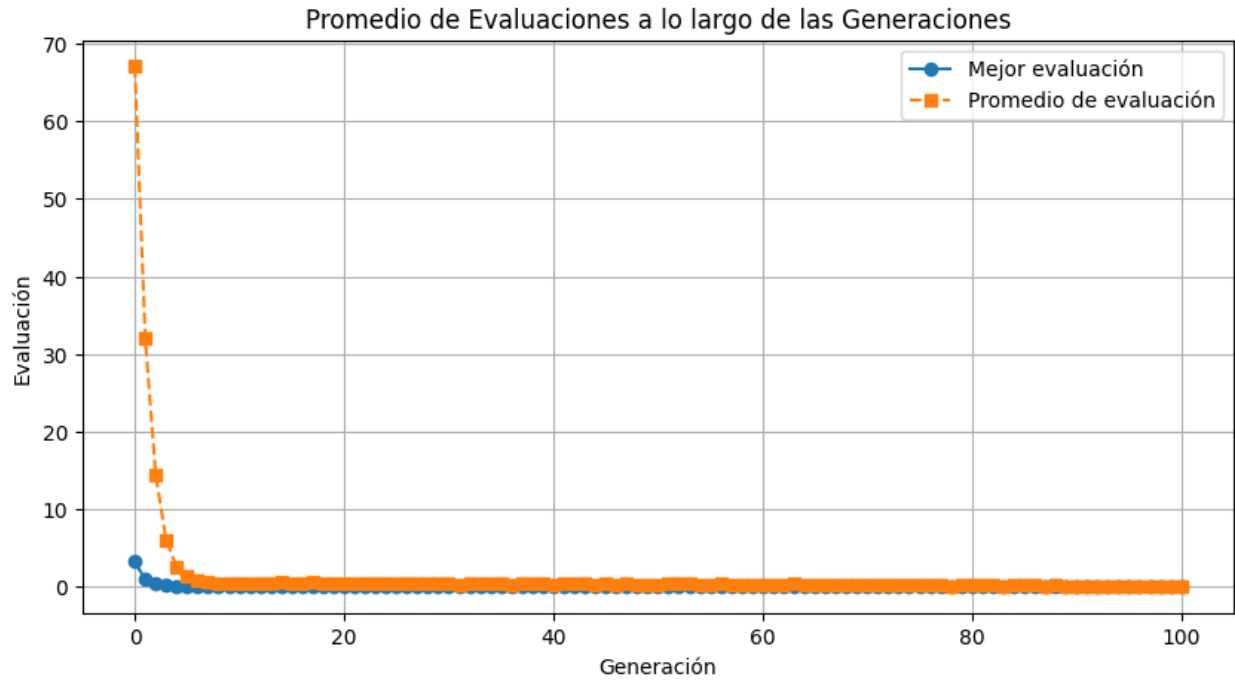


Figura 7: Promedio codificación real Esfera  $n = 2$

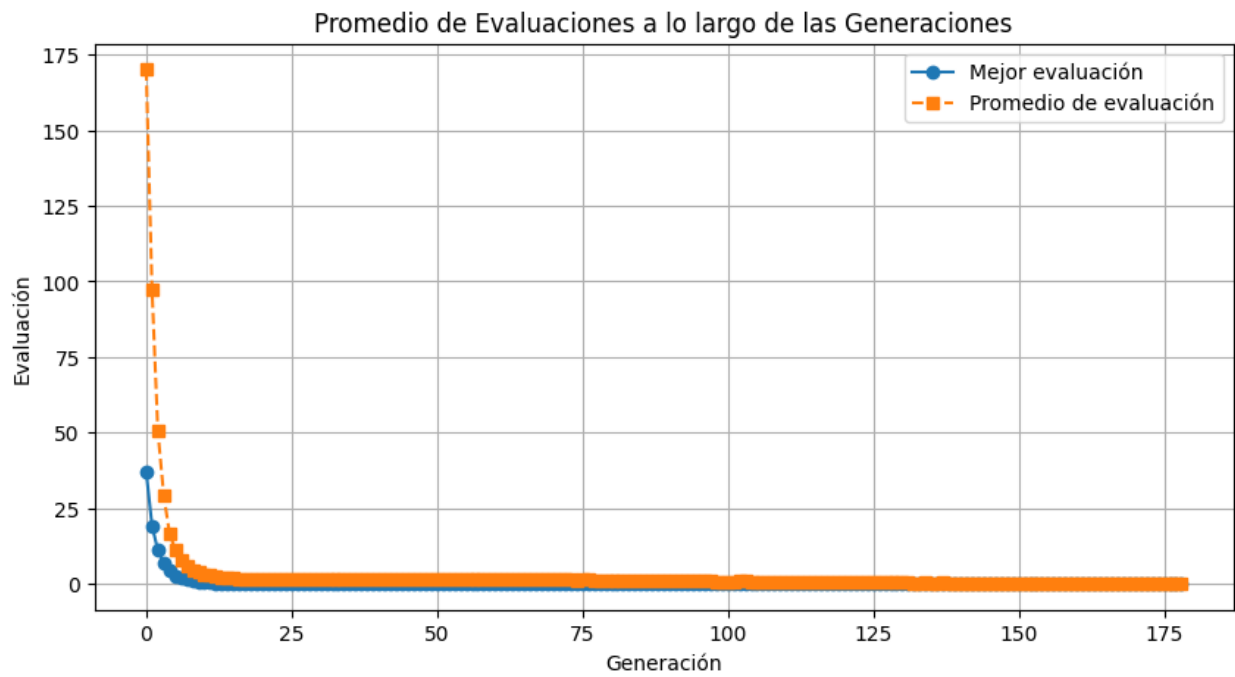



Figura 8: Promedio codificación real Esfera  $n = 5$

**Código:**  Nombre de archivo: `Cod real.ipynb`, con la función *Esfera* activada. Además de respetar los valores de  $n = 2$  o  $n = 5$  dependiendo del caso.



## Rosenbrok

$$f(\mathbf{x}) = \sum_{i=1}^{n-1} \left[ 100 (x_{i+1} - x_i^2)^2 + (1 - x_i)^2 \right]$$
$$-10 \leq x \leq 10$$

$$\text{Min} = \begin{cases} n = 2 & \rightarrow f(1, 1) = 0, \\ n = 3 & \rightarrow f(1, 1, 1) = 0, \\ n > 3 & \rightarrow f(\underbrace{1, \dots, 1}_{n \text{ times}}) = 0 \end{cases}$$

## Codificación binaria

### Parámetros

```
m = 15                # longitud del genotipo
pob_size = 50          # tamaño de la población
a = -10                # valor mínimo de x
b = 10                 # valor máximo de x
decimales = 3          # número de decimales
n = 2                  # variables a considerar
t = m * n              # total de bits
epsilon = 1e-6         # cota de selección
proba_cruza = 0.8      # probabilidad de cruza
proba_muta = 1/t       # probabilidad de mutación
generaciones = 100     # número de generaciones
etapas = 5             # número de etapas
umbral = 1e-8          # umbral de paro
max_sin_mejora = 50    # max de generaciones sin mejora
```

$n = 2$

Promedio de convergencia: 132.5 generaciones

La iteracion que tiene la minima evaluacion es: 1 con 77 generaciones.

Iteracion 1:

Total de generaciones: 77

Mejor genotipo: [1 0 0 0 1 1 0 0 1 1 0 1 1 1 0 1 0 0 0 1 1 0 0 1 1 1 0 1 1 0]

Mejor fenotipo(x1,...,xn): [1.005 1.01 ]

Mejor evaluacion (x1,...,xn): 0.0

$n = 5$

Promedio de convergencia: 165.3 generaciones

La iteracion que tiene la minima evaluacion es: 7 con 205 generaciones.

Iteracion 7:

Total de generaciones: 205

Mejor genotipo: [1 0 0 0 1 1 0 1 1 0 0 0 1 0 1 1 0 0 0 1 1 1 0 0 1 0 1 0 0 1 1 0 0 0 1 1 1  
1 1 1 1 1 1 1 1 1 0 0 1 0 1 0 0 0 0 1 1 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0]

Mejor fenotipo(x1,...,xn): [1.058 1.119 1.25 1.579 2.5 ]

Mejor evaluacion (x1,...,xn): 0.448

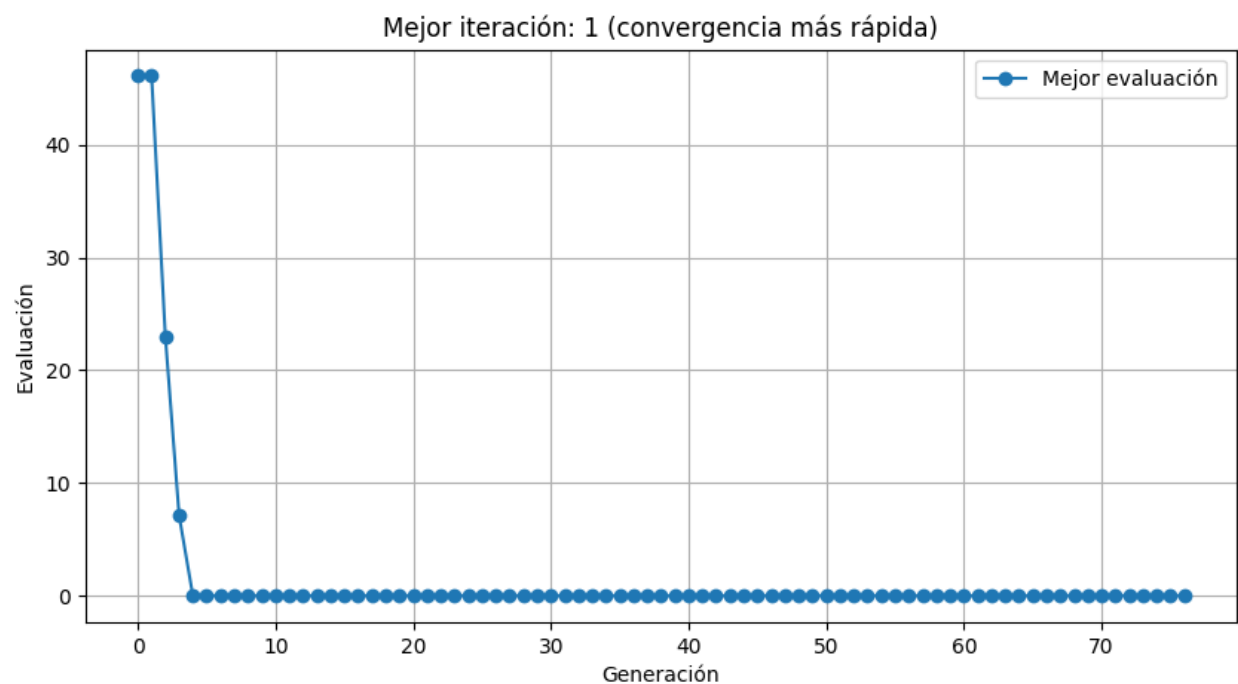


Figura 9: Codificación binaria Rosenbrok  $n = 2$

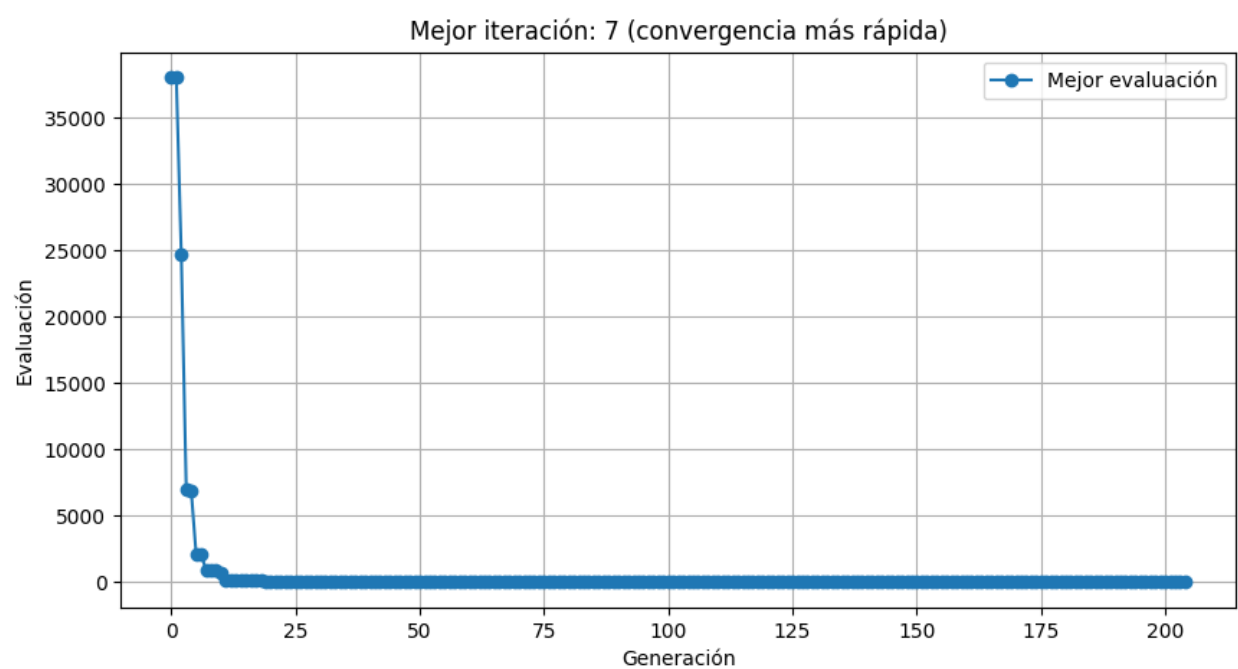


Figura 10: Codificación binaria Rosenbrok  $n = 5$

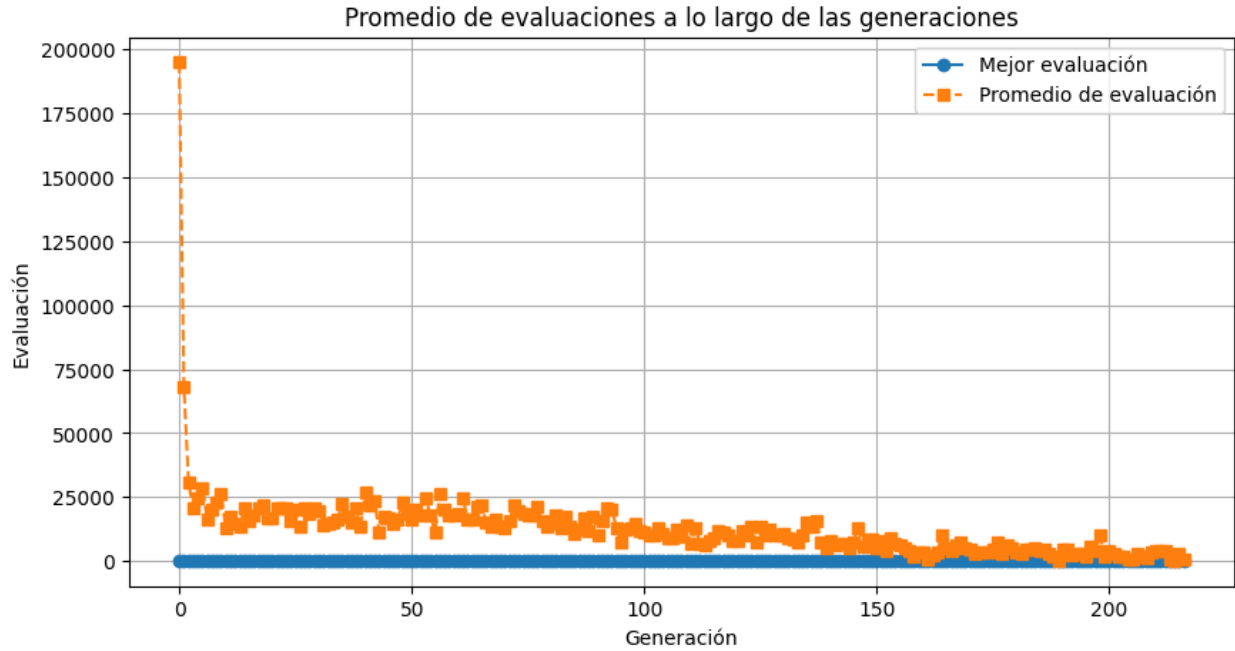


Figura 11: Promedio codificación binaria Rosenbrok  $n = 2$

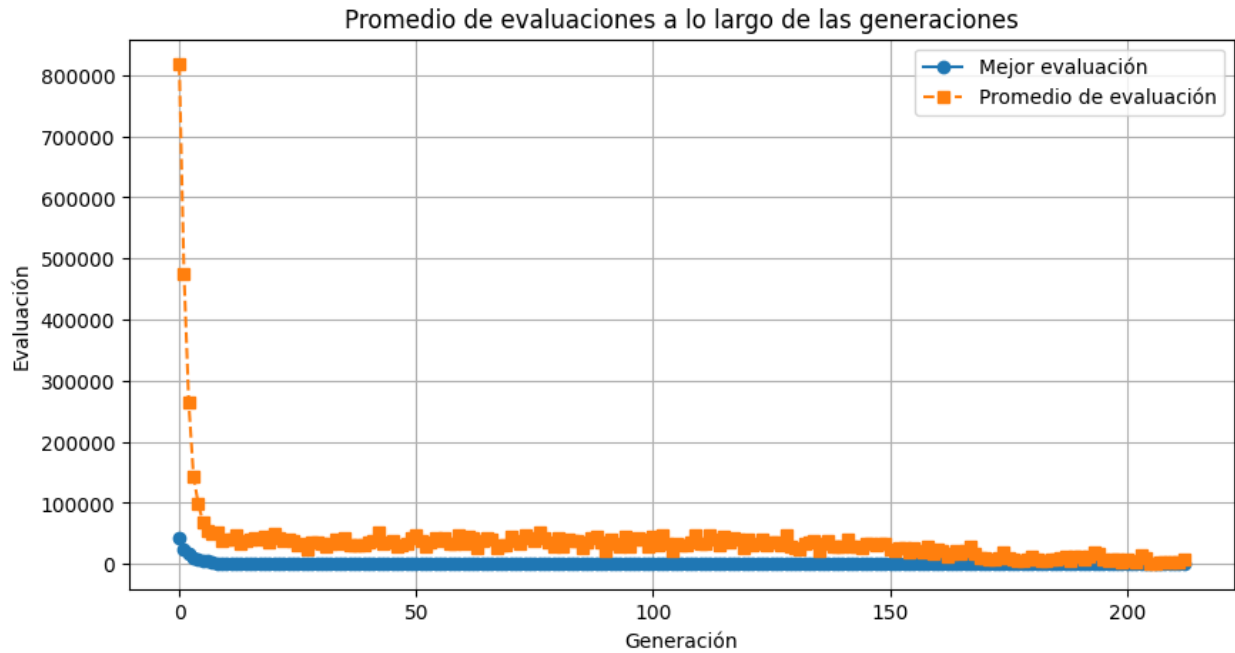



Figura 12: Promedio codificación binaria Rosenbrok  $n = 5$

**Código:**  *Nombre de archivo:* `Cod binaria.ipynb`, con la función *Rosenbrok* activada. Además de respetar los valores de  $n = 2$  o  $n = 5$  dependiendo del caso.

## Codificación real

### Parámetros

```
pob_size = 50      # tamaño de la población
a = -10            # valor mínimo de x
b = 10             # valor máximo de x
decimales = 3      # número de decimales
n = 5              # variables a considerar
epsilon = 1e-6     # cota de selección
proba_cruza = 0.8   # probabilidad de cruza
proba_muta = 0.2    # probabilidad de mutación
generaciones = 100 # número de generaciones
etapas = 5          # número de etapas
umbral = 1e-8       # umbral de paro
max_sin_mejora = 50 # max de generaciones sin mejora
```

$n = 2$

Promedio de convergencia: 73.8 generaciones

La iteracion que tiene la minima evaluacion es: 8 con 59 generaciones.

Iteracion 8:

Total de generaciones: 59

Mejor fenotipo (x1,...,xn): [0.984 0.97 ]

Mejor evaluacion f(x1,...,xn): 0.001

$n = 5$

Promedio de convergencia: 105.2 generaciones

La iteracion que tiene la minima evaluacion es: 7 con 63 generaciones.

Iteracion 7:

Total de generaciones: 63

Mejor fenotipo (x1,...,xn): [ 0.644 0.404 0.165 0.027 -0.018]

Mejor evaluacion f(x1,...,xn): 2.173

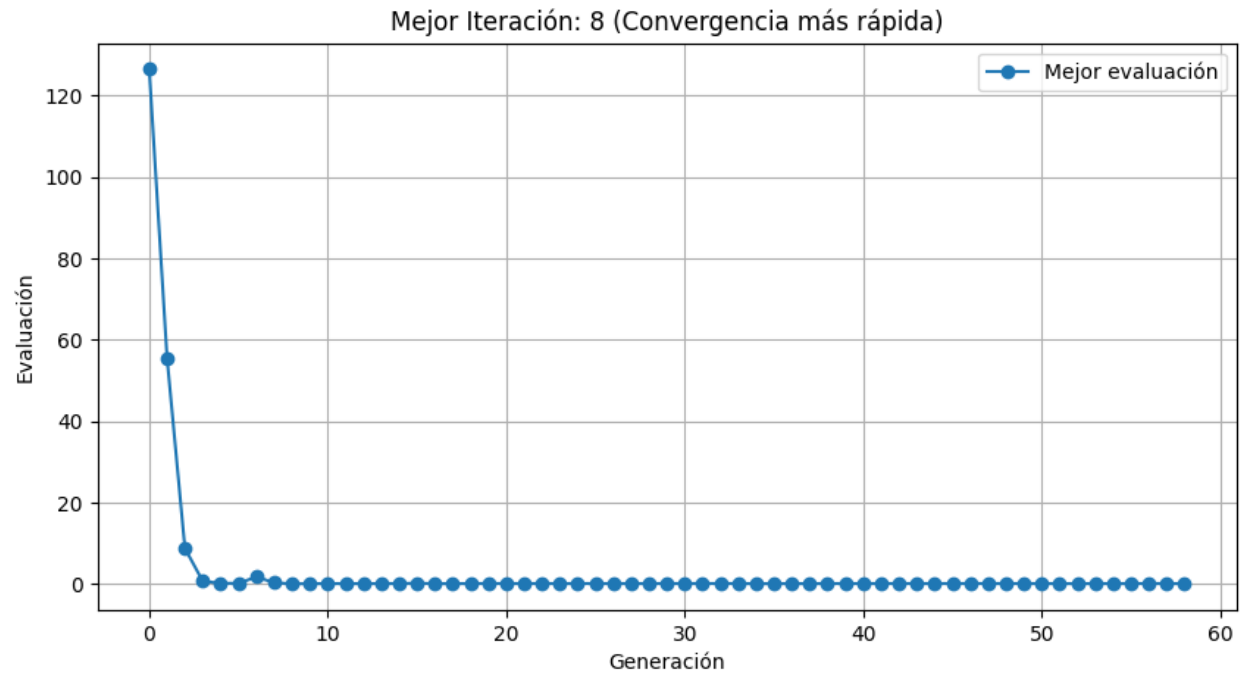


Figura 13: Codificación real Rosenbrok  $n = 2$



Figura 14: Codificación real Rosenbrok  $n = 5$

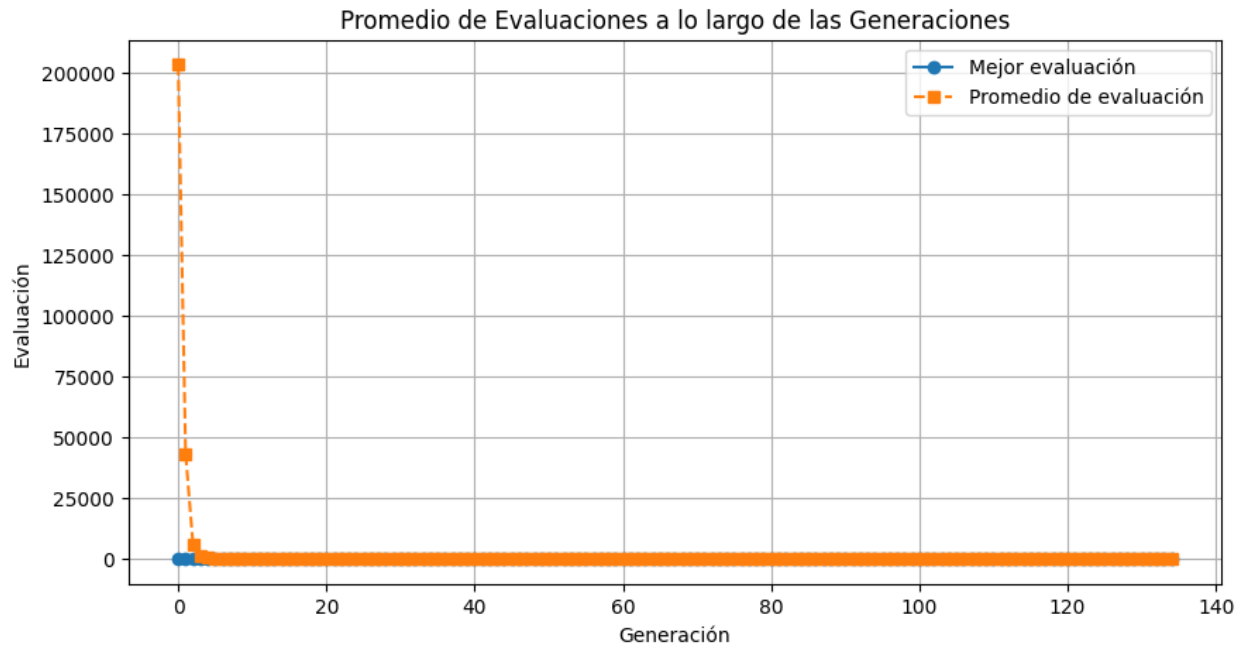


Figura 15: Promedio codificación real Rosenbrok  $n = 2$

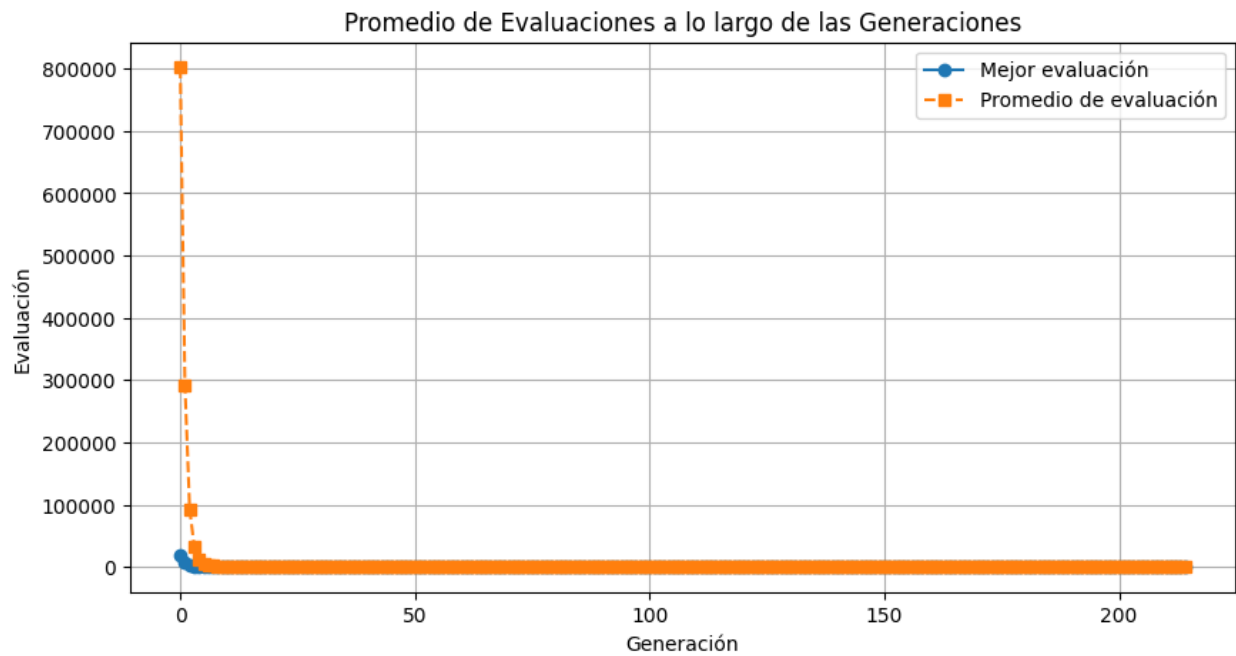



Figura 16: Promedio codificación real Rosenbrok  $n = 5$

**Código:**  *Nombre de archivo:* `Cod_real.ipynb`, con la función *Rosenbrok* activada. Además de respetar los valores de  $n = 2$  o  $n = 5$  dependiendo del caso.

## Himmelblau

$$f(x, y) = (x^2 + y - 11)^2 + (x + y^2 - 7)^2.$$
$$-5 \leq x, y \leq 5$$

$$\text{Min} = \begin{cases} f(3,0,2,0) & = 0,0 \\ f(-2,805118,3,131312) & = 0,0 \\ f(-3,779310,-3,283186) & = 0,0 \\ f(3,584428,-1,848126) & = 0,0 \end{cases}$$

## Codificación binaria

### Parámetros

```
m = 14           # longitud del genotipo
pob_size = 50     # tamaño de la población
a = -5           # valor mínimo de x
b = 5            # valor máximo de x
decimales = 3     # número de decimales
n = 2            # variables a considerar
t = m * n        # total de bits
epsilon = 1e-6    # cota de selección
proba_cruza = 0.8 # probabilidad de cruza
proba_muta = 1/t  # probabilidad de mutación
generaciones = 100 # número de generaciones
etapas = 5        # número de etapas
umbral = 1e-8     # umbral de paro
max_sin_mejora = 50 # max de generaciones sin mejora
```

Promedio de convergencia: 83.5 generaciones

La iteracion que tiene la minima evaluacion es: 1 con 87 generaciones.

Iteracion 1:

```
Total de generaciones: 87
Mejor genotipo: [0 0 1 1 1 0 0 0 0 1 0 1 1 1 0 1 0 0 0 0 0 1 0 0 1]
Mejor fenotipo(x1,...,xn): [-2.806  3.131]
Mejor evaluacion (x1,...,xn): 0.0
```



Figura 17: Codificación binaria Himmelblau

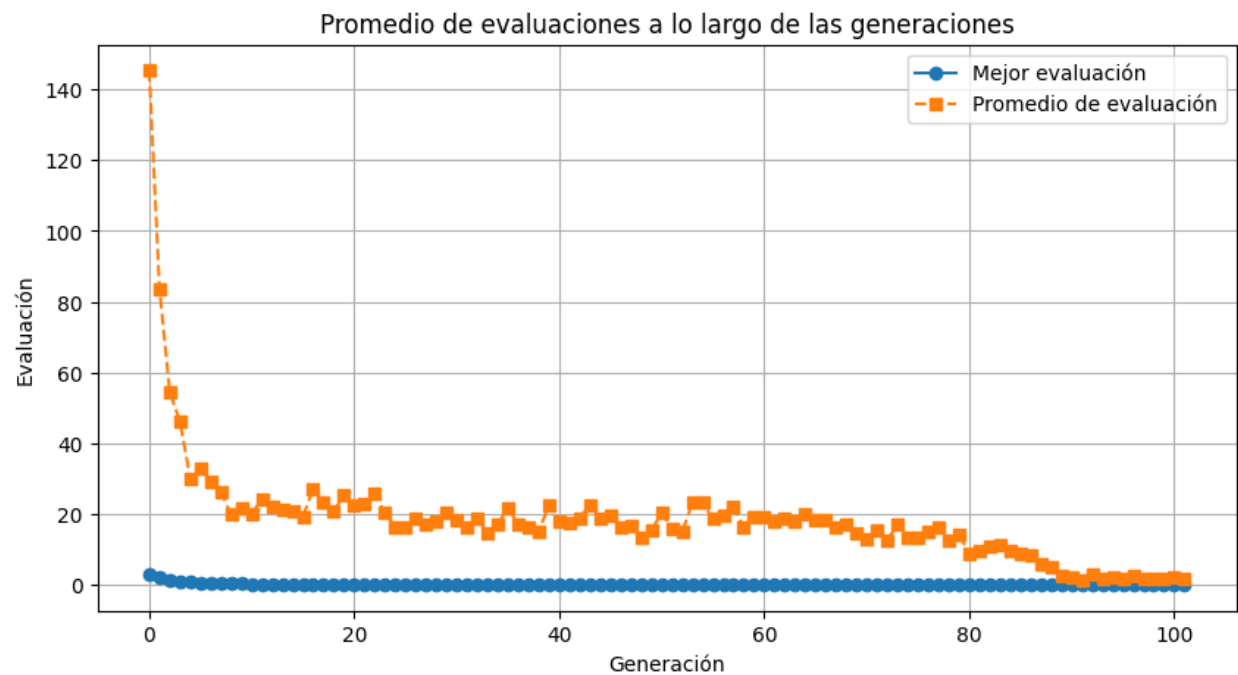



Figura 18: Promedio codificación binaria Himmelblau

Código:  Nombre de archivo: Cod binaria.ipynb, con la función *Himmelblau* activada.



## Codificación real

### Parámetros

```
pob_size = 50      # tamaño de la población
a = -5             # valor mínimo de x
b = 5              # valor máximo de x
decimales = 3      # número de decimales
n = 2              # variables a considerar
epsilon = 1e-6     # cota de selección
proba_cruza = 0.8   # probabilidad de cruza
proba_muta = 0.2    # probabilidad de mutación
generaciones = 100 # número de generaciones
etapas = 5         # número de etapas
umbral = -100      # umbral de paro
max_sin_mejora = 50 # max de generaciones sin mejora
```

Promedio de convergencia: 134.3 generaciones

La iteración que tiene la mínima evaluación es: 1 con 113 generaciones.

Iteración 1:

Total de generaciones: 113

Mejor fenotipo (x1,...,xn): [-2.805 3.132]

Mejor evaluación f(x1,...,xn): 0.0

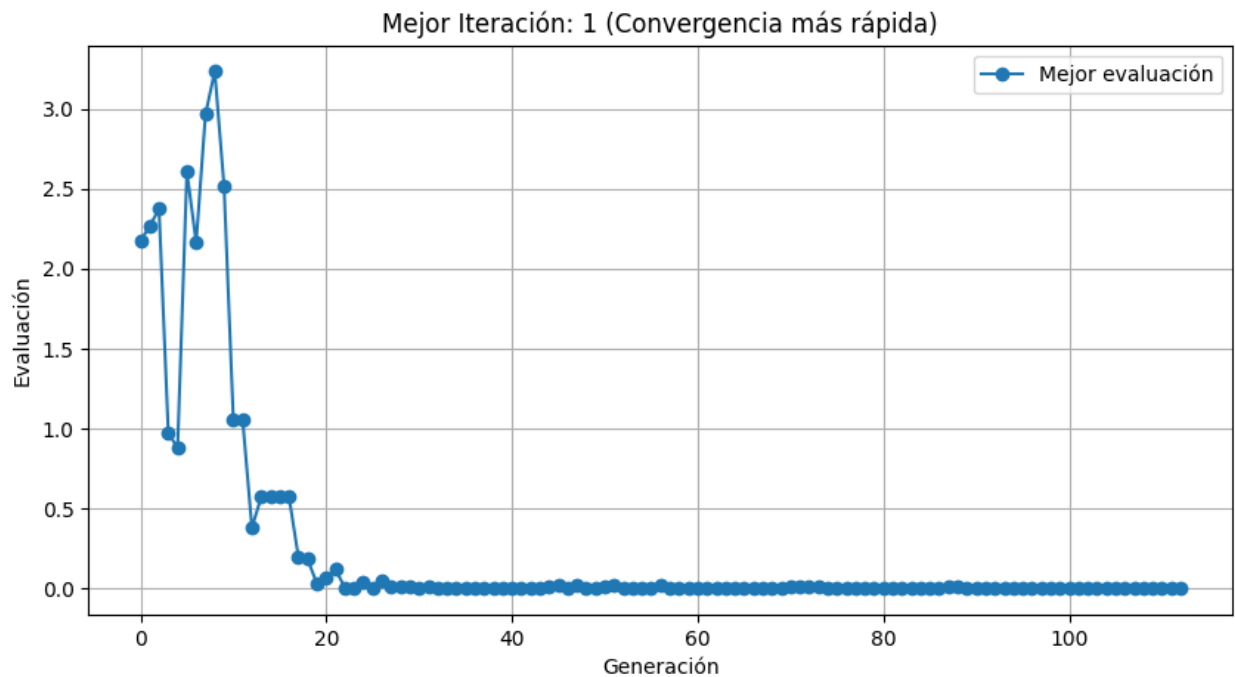


Figura 19: Codificación real Himmelblau

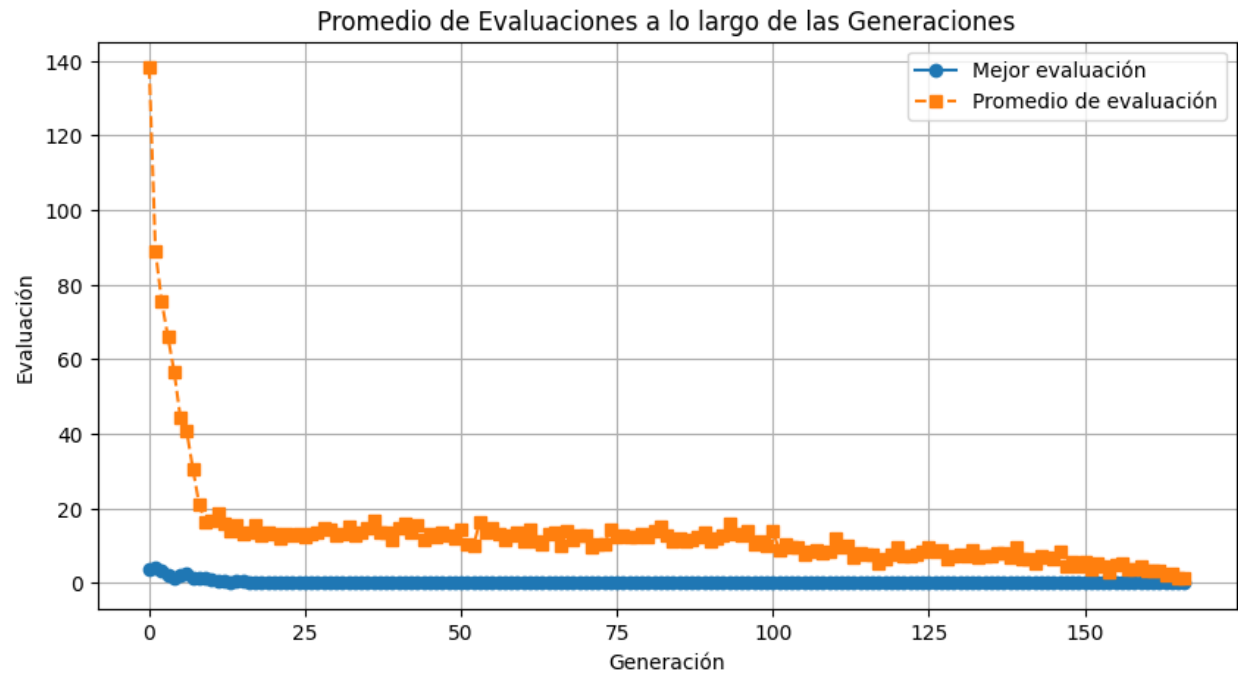



Figura 20: Promedio codificación real Himmelblau

Código:  Nombre de archivo: `Cod_real.ipynb`, con la función *Himmelblau* activada.

## Eggholder

$$f(x, y) = -(y + 47) \sin \sqrt{\left| \frac{x}{2} + (y + 47) \right|} - x \sin \sqrt{|x - (y + 47)|}$$
$$-512 \leq x, y \leq 512$$

$$f(512, 404, 2319) = -959,6407$$

## Codificación binaria

### Parámetros

```
m = 20                # longitud del genotipo
pob_size = 50         # tamaño de la población
a = -512              # valor mínimo de x
b = 512               # valor máximo de x
decimales=3          # número de decimales
n = 2                 # variables a considerar
t = m * n             # total de bits
epsilon = 1e-6        # cota de selección
proba_cruza = 0.8     # probabilidad de cruza
proba_muta = 1/t      # probabilidad de mutación
generaciones = 100    # número de generaciones
etapas = 5            # número de etapas
umbral = -1000        # umbral de paro
max_sin_mejora = 50   # max de generaciones sin mejora
```

Promedio de convergencia: 157.0 generaciones

La iteracion que tiene la minima evaluacion es: 5 con 89 generaciones.

Iteracion 5:

Total de generaciones: 89

Mejor genotipo: [1 0 0 1 0 0 1 1 1 1 1 1 1 1 1 1 1]

Mejor fenotipo(x1,...,xn): [512. 404.]

Mejor evaluacion (x1,...,xn): -959.58

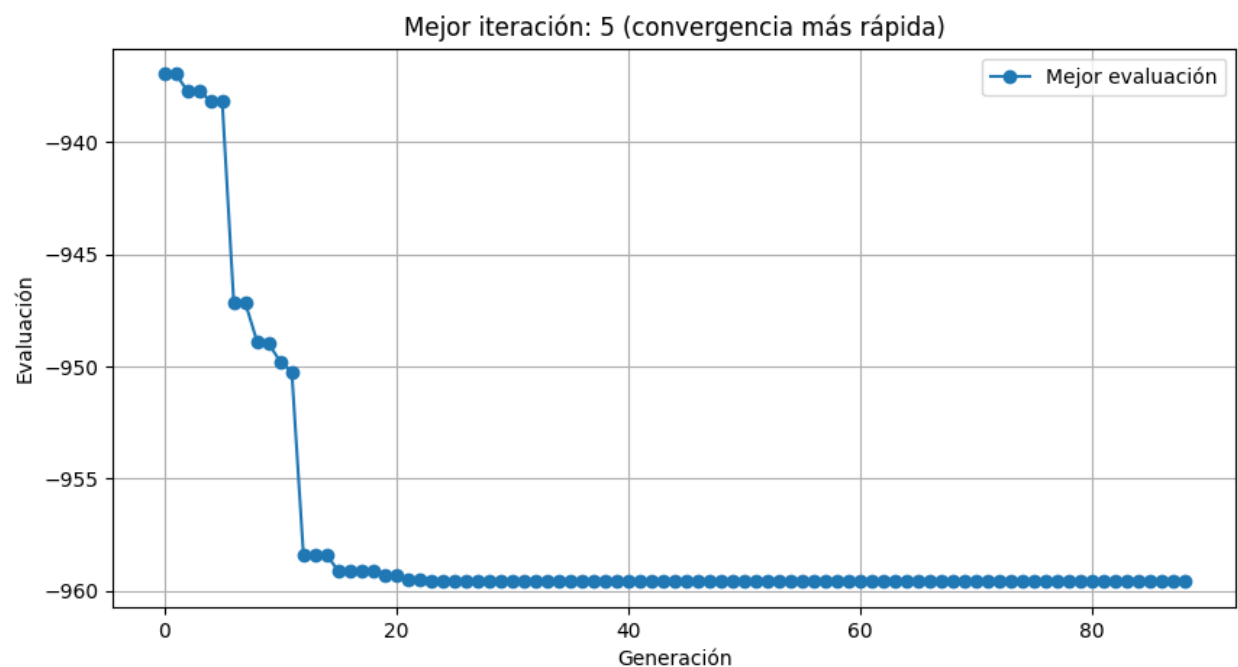


Figura 21: Codificación binaria Eggholder

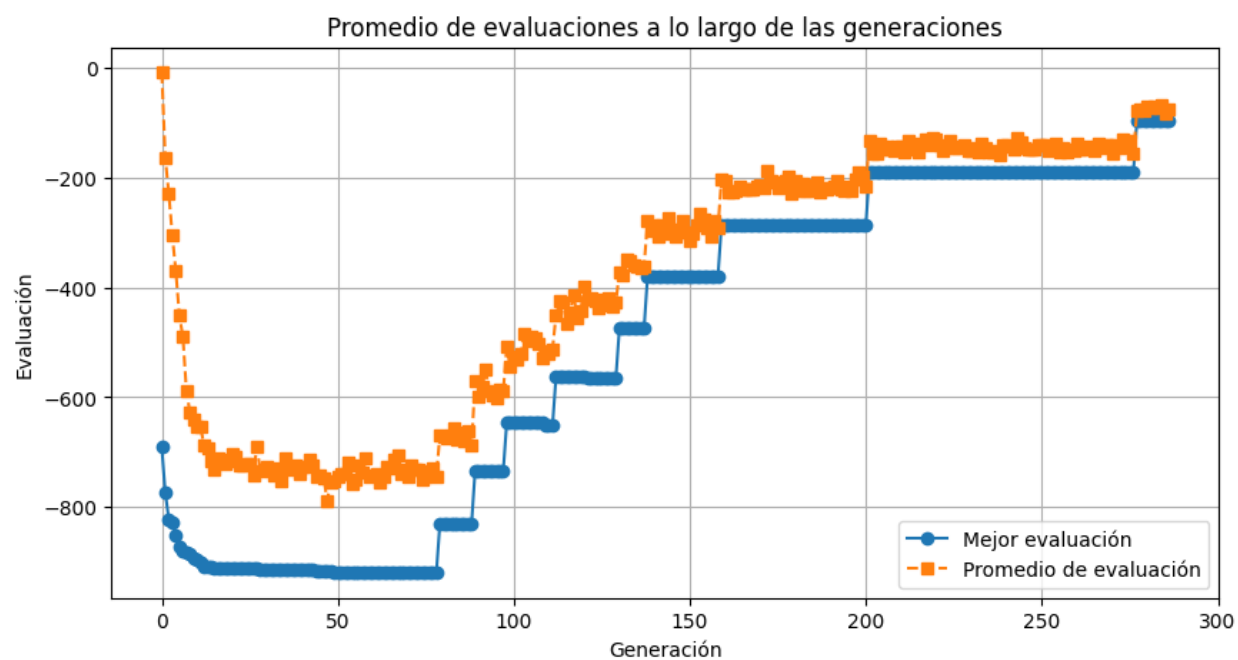



Figura 22: Promedio codificación binaria Eggholder

Código:  Nombre de archivo: Cod binaria.ipynb, con la función *Eggholder* activada.

## Codificación real

### Parámetros

```
pob_size = 50      # tamaño de la población
a = -512           # valor mínimo de x
b = 512            # valor máximo de x
decimales=3        # número de decimales
n = 2              # variables a considerar
epsilon = 1e-6     # cota de selección
proba_cruza = 0.8  # probabilidad de cruza
generaciones = 100 # número de generaciones
etapas = 5         # número de etapas
umbral = -1000     # umbral de paro
max_sin_mejora = 50 # max de generaciones sin mejora
```

Promedio de convergencia: 146.9 generaciones

La iteración que tiene la mínima evaluación es: 3 con 77 generaciones.

Iteracion 3:

Total de generaciones: 77

Mejor fenotipo (x1,...,xn): [512. 404.232]

Mejor evaluacion f(x1,...,xn): -959.641

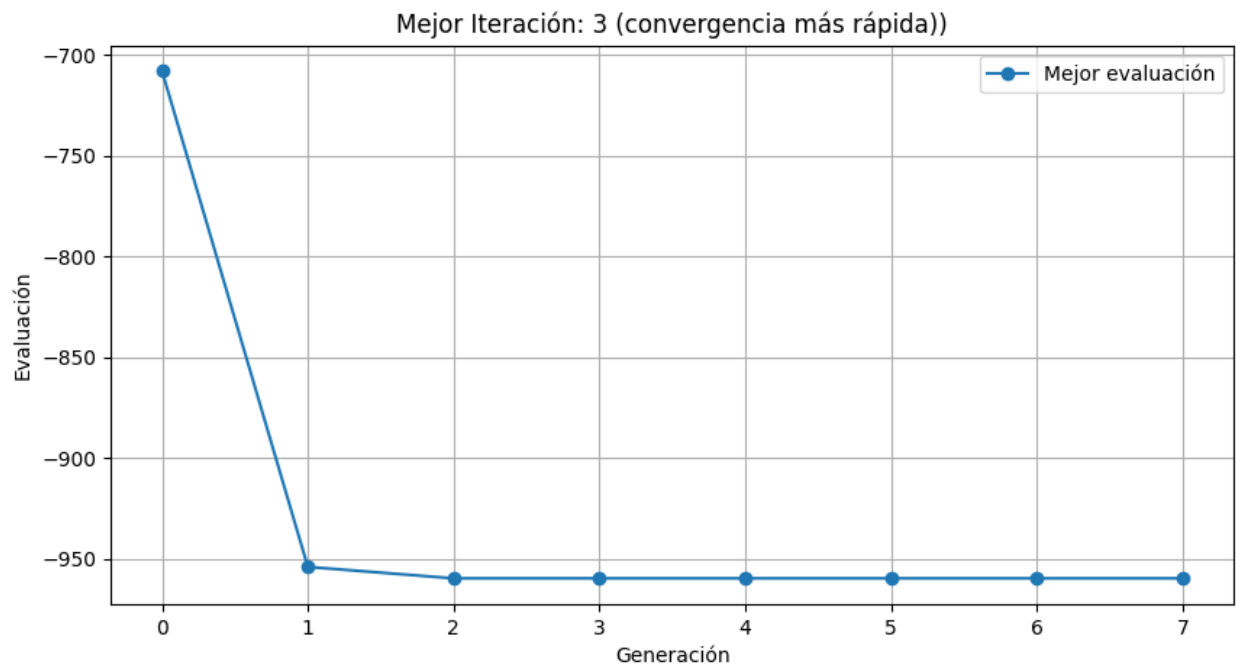


Figura 23: Codificación real Eggholder

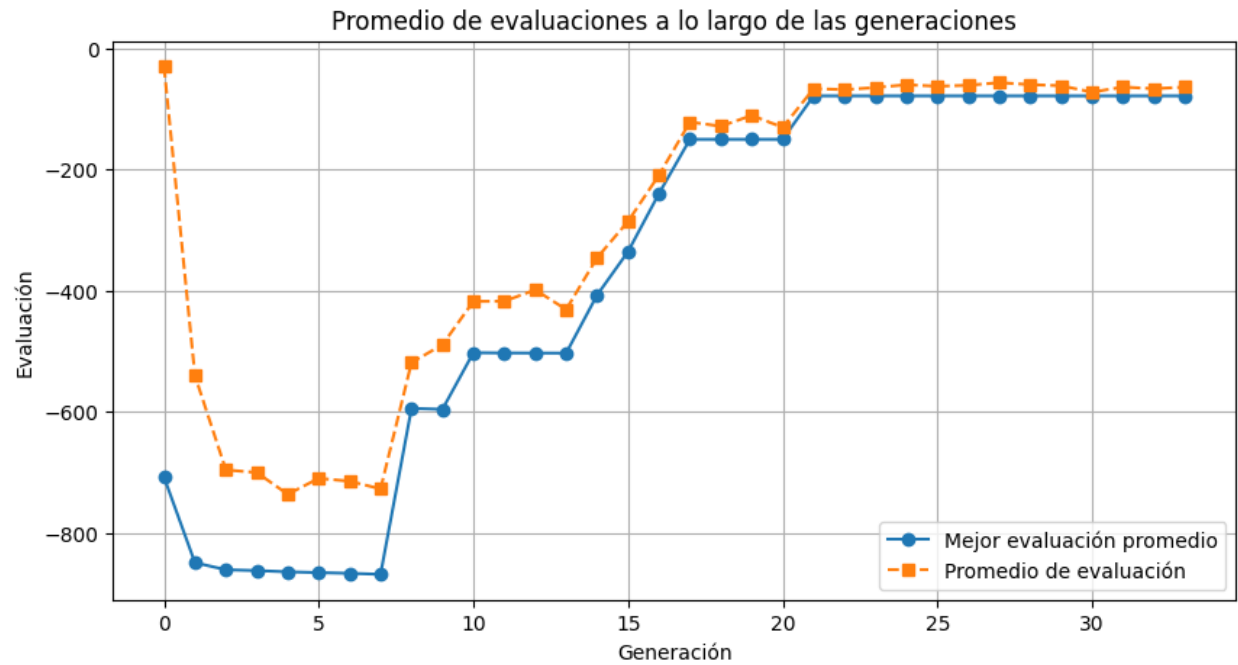




Figura 24: Promedio codificación real Eggholder

Código:  Nombre de archivo: `Cod_real.ipynb`, con la función *Eggholder* activada.

## N-Reinas

Para N-Reinas, la mejor representación es un vector de tamaño  $n$  donde cada valor indica la fila de la reina en la columna correspondiente. La función de aptitud es el número de pares de reinas no atacándose.

**Código:**  *Nombre de archivo:* `Reina.py`, especificando que codificación será usada, en el parámetro `codificacion` es necesario escoger entre `{'real', 'binario' o 'entero'}`.

### Codificación Binaria

#### Parámetros

```
N = 14                                # Número de reinas
pob_size = 100                        # Tamaño de la población
proba_muta = 0.2                      # Tasa de mutación
codificacion = 'binario'              # 'entero', 'real' o 'binario'
E = 10                                # Número de experimentos
GEN_MAX = 5000                        # Límite de generaciones por experimento
```

Experimento 1 finalizado: No se encontró solución en 5000 generaciones

Experimento 2 finalizado: No se encontró solución en 5000 generaciones

Experimento 3 finalizado: No se encontró solución en 5000 generaciones

Experimento 4 finalizado: No se encontró solución en 5000 generaciones

Experimento 5 finalizado: No se encontró solución en 5000 generaciones

Experimento 6 finalizado: No se encontró solución en 5000 generaciones

Experimento 7 finalizado: No se encontró solución en 5000 generaciones

Experimento 8 finalizado: No se encontró solución en 5000 generaciones

Experimento 9 finalizado: No se encontró solución en 5000 generaciones

Experimento 10 finalizado: No se encontró solución en 5000 generaciones

No converge dada la complejidad de  $2^{56}$

## Codificación Real

### Parámetros

```
N = 14                                # Número de reinas
pob_size = 100                        # Tamaño de la población
proba_muta = 0.2                      # Tasa de mutación
codificacion = 'real'                 # 'entero', 'real' o 'binario'
E = 10                                # Número de experimentos
GEN_MAX = 5000                        # Límite de generaciones por experimento
```

Experimento 1 finalizado: No se encontró solución en 5000 generaciones

Experimento 2 finalizado: No se encontró solución en 5000 generaciones

Experimento 3 finalizado: Solución encontrada en generación 79

Cromosoma decodificado: [ 2 5 11 4 12 10 3 7 0 8 13 1 6 9]

Experimento 4 finalizado: No se encontró solución en 5000 generaciones

Experimento 5 finalizado: No se encontró solución en 5000 generaciones

Experimento 6 finalizado: No se encontró solución en 5000 generaciones

Experimento 7 finalizado: No se encontró solución en 5000 generaciones

Experimento 8 finalizado: No se encontró solución en 5000 generaciones

Experimento 9 finalizado: No se encontró solución en 5000 generaciones

Experimento 10 finalizado: Solución encontrada en generación 287

Cromosoma decodificado: [ 3 7 13 1 9 6 10 2 4 11 0 8 12 5]

Representación del experimento 3 (codificación real): Cromosoma decodificado: [ 2 5 11 4 12 10 3 7 0 8 13 1 6 9]

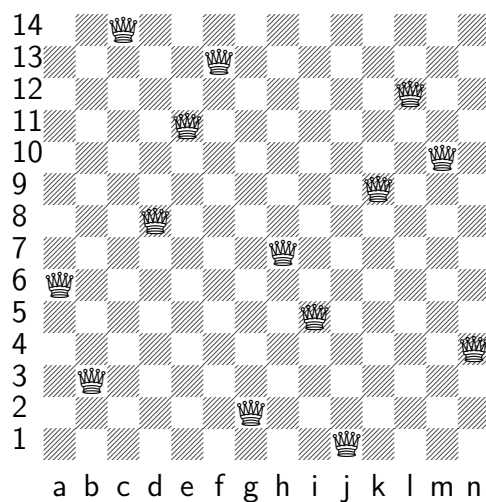


Figura 25: Tablero 14-Reinas



## Codificación Entera

### Parámetros

```
N = 14                                # Número de reinas
pob_size = 100                        # Tamaño de la población
proba_muta = 0.2                      # Tasa de mutación
codificacion = 'entero'               # 'entero', 'real' o 'binario'
E = 10                                # Número de experimentos
GEN_MAX = 5000                        # Límite de generaciones por experimento
```

Experimento 1 finalizado: No se encontró solución en 5000 generaciones

Experimento 2 finalizado: No se encontró solución en 5000 generaciones

Experimento 3 finalizado: Solución encontrada en generación 133  
Cromosoma decodificado: [ 8 6 4 9 5 0 13 10 12 7 1 3 11 2]

Experimento 4 finalizado: No se encontró solución en 5000 generaciones

Experimento 5 finalizado: Solución encontrada en generación 2831  
Cromosoma decodificado: [ 9 1 8 2 11 3 7 10 13 5 12 0 4 6]

Experimento 6 finalizado: No se encontró solución en 5000 generaciones

Experimento 7 finalizado: Solución encontrada en generación 972  
Cromosoma decodificado: [ 5 12 1 6 13 2 8 11 0 3 10 4 7 9]

Experimento 8 finalizado: No se encontró solución en 5000 generaciones

Experimento 9 finalizado: No se encontró solución en 5000 generaciones

Experimento 10 finalizado: No se encontró solución en 5000 generaciones

Representación del experimento 3 (codificación entera): Cromosoma decodificado: [ 8 6 4 9 5 0 13 10 12 7 1 3 11 2]

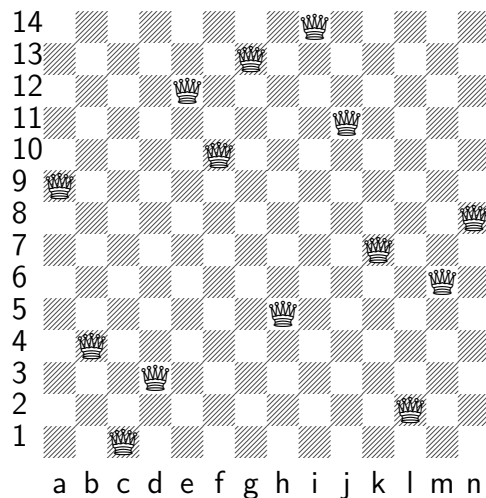



Figura 26: Tablero 14-Reinas

# TSP

*Dada una lista de ciudades y las distancias entre cada par de ciudades, ¿cuál es la ruta más corta posible que visita cada ciudad exactamente una vez y regresa a la ciudad de origen?*

**Código:**  *Nombre de archivo:* TSP.ipynb, especificando que codificación será usada, en el parámetro CODING es necesario escoger entre {'real', 'binario' o 'entero'}.

## Codificación Binaria

### Parámetros

```
POP_SIZE = 100
GENERATIONS = 500
TOURNAMENT_SIZE = 3
MUT_RATE = 0.2
MAX_NO_IMPROVE = 50
CODING = 'binario' # 'entero', 'real', 'binario'
EXPERIMENTS = 10
```

```
Exp 1: Máximo de generaciones alcanzado | Mejor costo: 2293.0
Exp 2: Máximo de generaciones alcanzado | Mejor costo: 2171.0
Exp 3: Máximo de generaciones alcanzado | Mejor costo: 2173.0
Exp 4: Máximo de generaciones alcanzado | Mejor costo: 2516.0
Exp 5: Máximo de generaciones alcanzado | Mejor costo: 2195.0
Exp 6: Máximo de generaciones alcanzado | Mejor costo: 2098.0
Exp 7: Máximo de generaciones alcanzado | Mejor costo: 2231.0
Exp 8: Máximo de generaciones alcanzado | Mejor costo: 2155.0
Exp 9: Máximo de generaciones alcanzado | Mejor costo: 2158.0
Exp 10: Máximo de generaciones alcanzado | Mejor costo: 2226.0
```

## Codificación Real

### Parámetros

```
POP_SIZE = 100
GENERATIONS = 500
TOURNAMENT_SIZE = 3
MUT_RATE = 0.2
MAX_NO_IMPROVE = 50
CODING = 'real' # 'entero', 'real', 'binario'
EXPERIMENTS = 10
```

```
Exp 1: Máximo de generaciones alcanzado | Mejor costo: 2120.0
Exp 2: Máximo de generaciones alcanzado | Mejor costo: 2088.0
Exp 3: Máximo de generaciones alcanzado | Mejor costo: 2167.0
Exp 4: Máximo de generaciones alcanzado | Mejor costo: 2095.0
Exp 5: Máximo de generaciones alcanzado | Mejor costo: 2157.0
Exp 6: Máximo de generaciones alcanzado | Mejor costo: 2158.0
Exp 7: Máximo de generaciones alcanzado | Mejor costo: 2090.0
Exp 8: Máximo de generaciones alcanzado | Mejor costo: 2085.0
Exp 9: Máximo de generaciones alcanzado | Mejor costo: 2195.0
Exp 10: Máximo de generaciones alcanzado | Mejor costo: 2085.0
```

## Codificación Entera

### Parámetros

```
POP_SIZE = 100
GENERATIONS = 500
TOURNAMENT_SIZE = 3
MUT_RATE = 0.2
MAX_NO_IMPROVE = 50
CODING = 'entero' # 'entero', 'real', 'binario'
EXPERIMENTS = 10
```

```
Exp 1: Máximo de generaciones alcanzado | Mejor costo: 2085.0
Exp 2: Máximo de generaciones alcanzado | Mejor costo: 2090.0
Exp 3: Máximo de generaciones alcanzado | Mejor costo: 2158.0
Exp 4: Máximo de generaciones alcanzado | Mejor costo: 2103.0
Exp 5: Máximo de generaciones alcanzado | Mejor costo: 2210.0
Exp 6: Máximo de generaciones alcanzado | Mejor costo: 2095.0
Exp 7: Máximo de generaciones alcanzado | Mejor costo: 2155.0
Exp 8: Máximo de generaciones alcanzado | Mejor costo: 2103.0
Exp 9: Máximo de generaciones alcanzado | Mejor costo: 2090.0
Exp 10: Máximo de generaciones alcanzado | Mejor costo: 2167.0
```