

Memoria Práctica 2

Programación Evolutiva

Grupo 14

*Pablo Jiménez Cruz
Pedro Sánchez Escribano*

● **Análisis de las gráficas y parámetros del algoritmo genético:**

Se va a realizar una descripción de la gráfica y de sus parámetros:

- La gráfica muestra en color rojo el peor absoluto durante la ejecución.
- La gráfica muestra en color azul el mejor absoluto durante la ejecución.
En el caso de que haya elitismo esta no será visible.
- En color verde se muestra el mejor de cada generación.
- En color amarillo se muestra la media de cada generación.

Debajo de la gráfica se muestra la mejor y peor combinación junto con su fitness de todas las generaciones de la ejecución. También se muestran el total de mutaciones, selecciones y cruces.

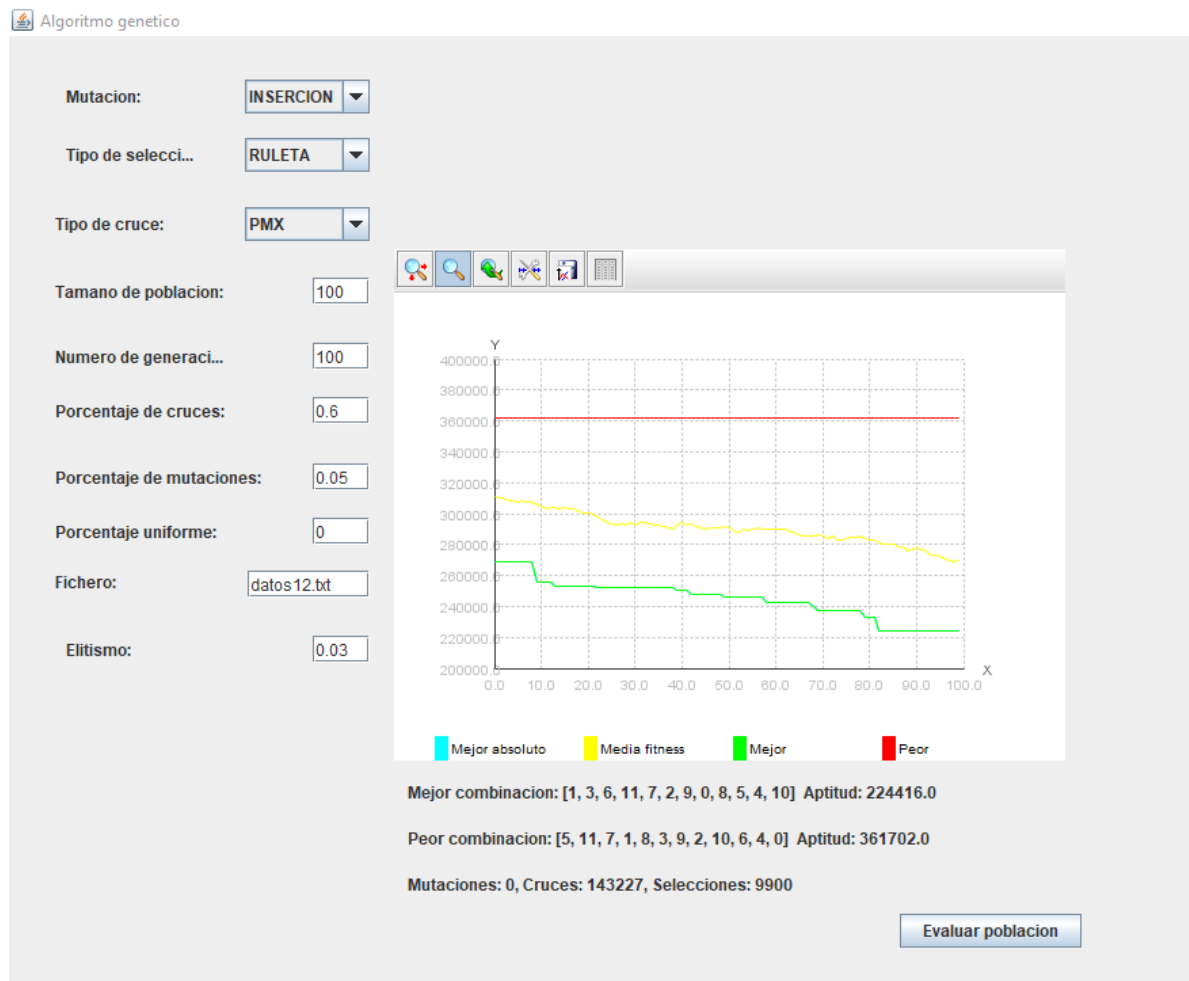
Los parámetros que necesita el algoritmo genético son varios:

- El tipo de mutación a elegir entre las mutaciones implementadas.
- El tipo de mutación a elegir entre las mutaciones implementadas.
- El tipo de mutación a elegir entre las mutaciones implementadas.
- Un entero que codifica el tamaño de la población.
- Un entero que codifica el número de generaciones.
- La probabilidad de cruce(double).
- La probabilidad mutación(double).
- El fichero que se va a coger para realizar la evaluación (ajuste.txt, datos12.txt, datos15.txt, datos30.txt).
- El porcentaje de elitismo.

● **Resultados Obtenidos:**

A continuación, se mostrará una tabla que muestra un resumen de 20 ejecuciones, incluyendo para cada ejecución la aptitud y el fenotipo del mejor individuo de la población. En la parte inferior de la tabla se mostrará un resumen de las 20 ejecuciones que muestre la mejor solución de esas 20 ejecuciones, la media y la peor solución.

– Ejemplo 1: *datos12.txt* con coste óptimo



Como se puede ver en la imagen se muestra:

- El mejor individuo y su actitud
- El peor individuo y su actitud
- El número de mutaciones, cruce y selecciones

Número Generación	Mejor Solución	Aptitud Mejor Solución
1	1,3,6,11,7,2,9,0,8,5,4,10	224416
2	10,7,3,0,8,5,9,4,6,11,2,1	234788
3	9,2,10,11,5,4,6,3,8,7,0,1	242180
4	10,11,3,0,5,9,4,8,7,1,2,6	235468
5	10,1,3,0,4,7,8,11,9,5,2,6	239912
6	6,2,4,0,1,3,10,8,5,7,11,9	243252
7	6,3,1,11,7,2,9,0,8,5,4,10	229982
8	1,3,6,11,7,2,9,0,8,5,4,10	224416
9	4,8,6,0,1,2,5,11,10,3,9,7	244890
10	7,3,1,5,8,10,9,0,6,11,4,2	238490
11	7,3,8,11,1,2,9,0,6,4,5,10	229092
12	9,3,10,11,5,2,1,0,8,7,4,6	239480
13	9,4,5,11,1,2,10,0,8,7,3,6	242854
14	10,7,6,0,3,5,9,4,8,2,11,1	237560
15	10,11,3,0,9,1,4,8,5,7,2,6	235870
16	9,3,10,11,5,2,1,0,8,7,4,6	239480
17	7,3,5,8,1,2,11,0,4,6,9,10	243612
18	4,11,9,0,1,8,3,5,10,7,2,6	235972
19	4,11,9,0,1,8,3,5,10,7,2,6	235972
20	6,9,11,3,0,4,7,5,2,1,8,10	239818

Mejor solución de las 20 ejecuciones:

1,3,6,11,7,2,9,0,8,5,4,10	224416
---------------------------	--------

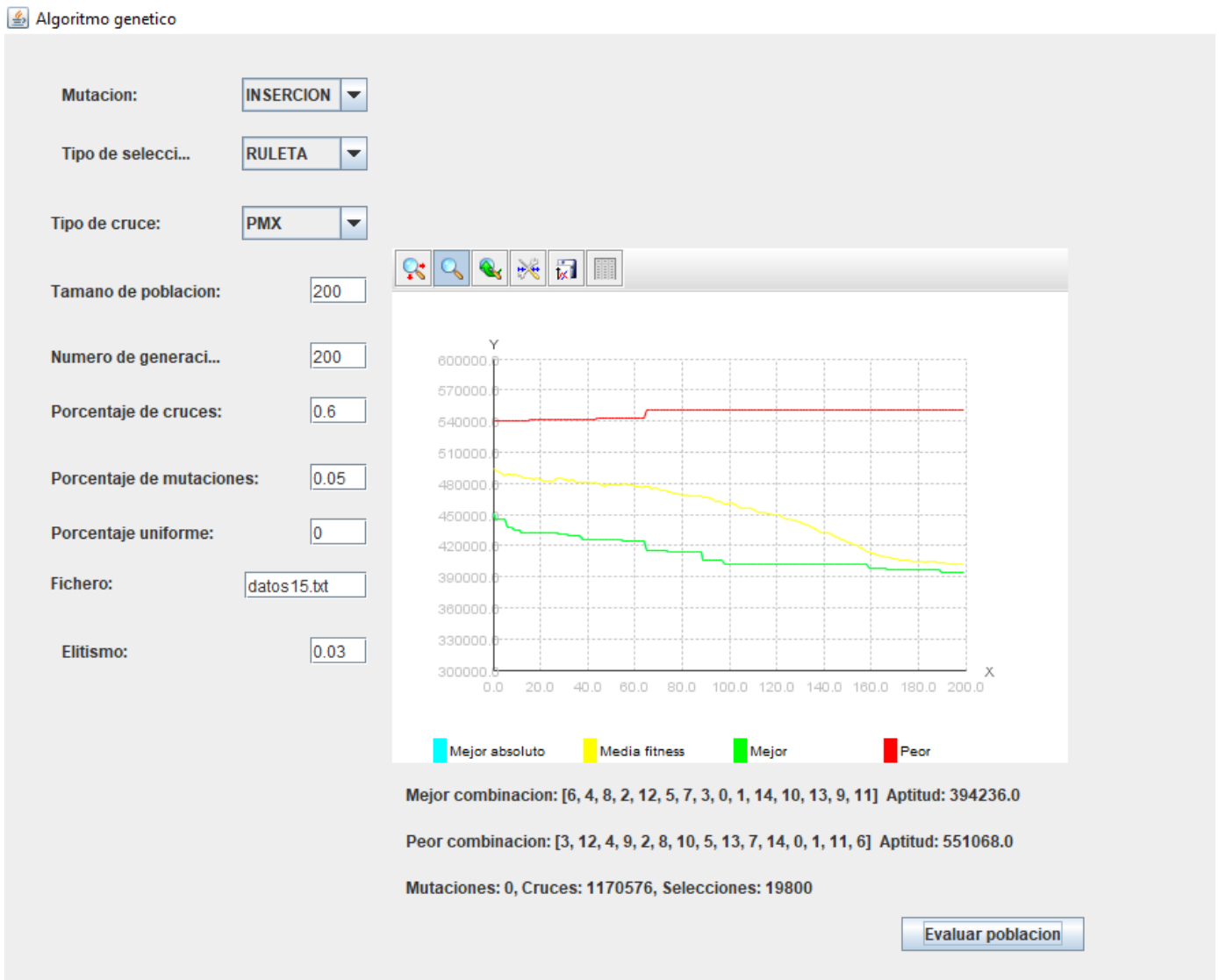
Peor solución de las 20 ejecuciones:

4,8,6,0,1,2,5,11,10,3,9,7	244890
---------------------------	--------

Media después de las 20 ejecuciones:

236696

– Ejemplo 2: *datos15.txt*



Como se puede ver en la imagen se muestra:

- El mejor individuo y su actitud
- El peor individuo y su actitud
- El número de mutaciones, cruce y selecciones

Número Generación	Mejor Solución	Aptitud Mejor Solución
1	6,4,8,2,12,5,7,3,0,1,14,10,13,9,11	394236
2	4,1,5,0,13,10,3,8,7,11,2,14,6,12,9	394692
3	4,9,12,6,5,0,1,10,14,2 13,7,3,8,11	397354
4	10,1,0,3,4,7,8,2,13,5,12,9,6,11,14	393086
5	6,10,11,2,0,13,1,7,3,12,9,8,4,14,5	398332
6	8,2,7,9,3,14,6,1,5,13,12,10,0,11,4	410264
7	10,2,12,0,14,4,6,1,5,7,3,13,11,9,8	392712
8	7,8,5,4,13,0,1,10,12,6,2,14,3,9,11	411782
9	14,10,13,2,0,11,1,7,6,12,9,8,4,3,5	407458
10	6,11,14,9,2,10,0,12,13,1,5,7,4,3,8	414122
11	9,5,8,2,12,7,6,11,1,4,0,10,3,13,14	399748
12	10,14,12,0,2,4,6,1,5,13,3,7,11,9,8	396018
13	1,8,13,11,12,3,10,2,6,9,4,5,0,7,14	394626
14	1,11,13,5,8,2,6,13,3,10,12,0,9,7,4	409346
15	10,4,7,11,12,1,5,2,6,9,3,8,0,13,14	398714
16	3,0,13,11,12,4,6,7,8,9,14,5,10,1,2	394658
17	8,2,7,0,14,1,5,4,6,11,3,10,9,13,12	398420
18	11,5,1,3,4,10,0,14,13,12,8,6,9,7,2	400324
19	9,8,5,0,13,4,13,7,6,2,11,14,12,10	405030
20	0,11,10,8,3,2,6,9,14,13,1,5,4,7,12	399032

Mejor solución de las 20 ejecuciones:

10,2,12,0,14,4,6,1,5,7,3,13,11,9,8	392712
------------------------------------	--------

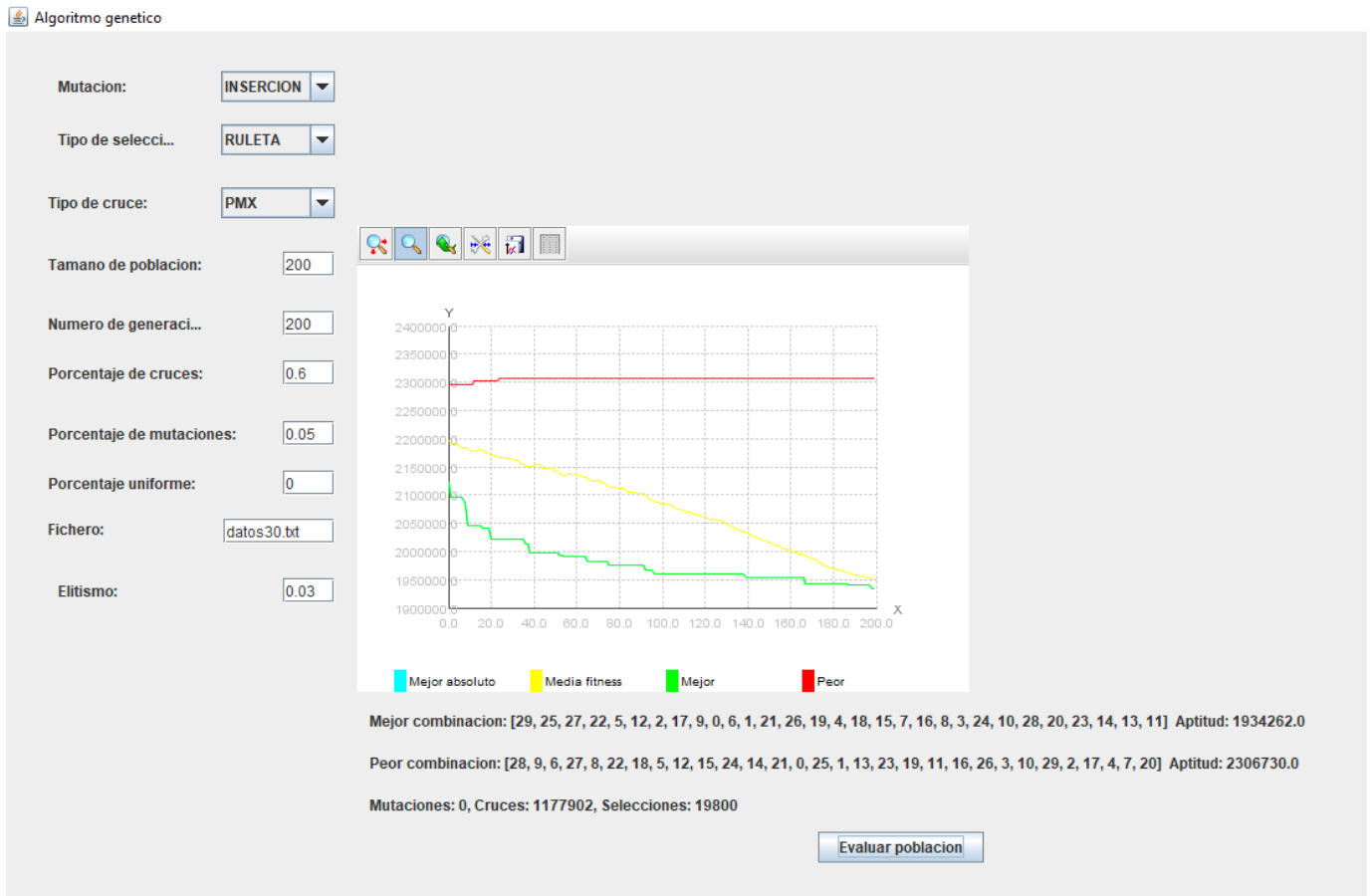
Peor solución de las 20 ejecuciones:

6,11,14,9,2,10,0,12,13,1,5,7,4,3,8	414122
------------------------------------	--------

Media después de las 20 ejecuciones:

400494,7

– Ejemplo 3: *datos30.txt*



Como se puede ver en la imagen se muestra:

- El mejor individuo y su actitud
- El peor individuo y su actitud
- El número de mutaciones, cruce y selecciones

– Ejemplo 3: *datos30.txt*

Mejor combinacion: [29, 25, 27, 22, 5, 12, 2, 17, 9, 0, 6, 1, 21, 26, 19, 4, 18, 15, 7, 16, 8, 3, 24, 10, 28, 20, 23, 14, 13, 11]	Aptitud: 1934262.0
Mejor combinacion: [15, 5, 7, 20, 25, 11, 10, 13, 0, 24, 19, 2, 18, 9, 16, 26, 8, 6, 14, 17, 1, 28, 29, 27, 3, 23, 22, 4, 12, 21]	Aptitud: 1958224.0
Mejor combinacion: [9, 19, 16, 4, 20, 3, 14, 22, 0, 1, 5, 6, 2, 28, 26, 17, 29, 27, 11, 10, 15, 12, 21, 8, 24, 25, 13, 18, 7, 23]	Aptitud: 1947488.0
Mejor combinacion: [10, 19, 27, 17, 6, 21, 8, 29, 7, 11, 28, 26, 15, 5, 13, 12, 18, 24, 25, 22, 0, 4, 14, 23, 3, 1, 20, 9, 16, 2]	Aptitud: 1966804.0
Mejor combinacion: [17, 10, 13, 26, 15, 9, 5, 0, 25, 1, 6, 22, 21, 28, 27, 8, 23, 3, 29, 14, 2, 18, 12, 16, 20, 11, 19, 7, 4, 24]	Aptitud: 1947158.0
Mejor combinacion: [13, 5, 2, 6, 24, 25, 7, 23, 14, 1, 19, 21, 17, 8, 16, 4, 11, 20, 26, 0, 29, 12, 10, 15, 22, 9, 27, 3, 18, 28]	Aptitud: 1965472.0
Mejor combinacion: [11, 22, 7, 5, 25, 12, 24, 1, 16, 13, 4, 3, 17, 10, 23, 28, 0, 26, 9, 18, 19, 8, 14, 20, 29, 2, 27, 15, 6, 21]	Aptitud: 1946370.0
Mejor combinacion: [1, 25, 18, 22, 9, 15, 6, 11, 26, 27, 23, 21, 8, 19, 0, 28, 5, 29, 3, 20, 10, 16, 14, 24, 12, 13, 17, 7, 2, 4]	Aptitud: 1948504.0
Mejor combinacion: [28, 2, 11, 21, 26, 18, 17, 3, 10, 7, 8, 0, 29, 23, 20, 15, 19, 14, 6, 1, 24, 9, 27, 22, 4, 12, 16, 13, 25, 5]	Aptitud: 1950734.0
Mejor combinacion: [18, 22, 27, 26, 4, 23, 6, 0, 21, 12, 1, 20, 8, 15, 25, 29, 28, 16, 5, 17, 3, 14, 19, 2, 7, 10, 11, 24, 13, 9]	Aptitud: 1952740.0
Mejor combinacion: [1, 28, 23, 21, 11, 20, 29, 24, 19, 6, 16, 15, 26, 13, 27, 25, 10, 22, 12, 7, 14, 18, 8, 4, 3, 17, 2, 9, 0, 5]	Aptitud: 1951070.0
Mejor combinacion: [10, 25, 20, 5, 19, 9, 13, 26, 2, 11, 14, 23, 12, 28, 18, 22, 16, 24, 17, 27, 1, 29, 0, 21, 8, 15, 4, 7, 3, 6]	Aptitud: 1906424.0
Mejor combinacion: [17, 20, 0, 14, 23, 10, 4, 21, 7, 16, 22, 13, 5, 12, 29, 19, 18, 24, 26, 9, 8, 11, 25, 1, 6, 15, 3, 28, 27, 2]	Aptitud: 1956834.0
Mejor combinacion: [13, 19, 24, 16, 3, 25, 27, 10, 9, 26, 28, 0, 14, 8, 20, 11, 1, 18, 2, 12, 17, 5, 23, 29, 21, 22, 7, 6, 15, 4]	Aptitud: 1937098.0
Mejor combinacion: [15, 10, 1, 2, 7, 27, 22, 16, 5, 25, 8, 12, 6, 9, 29, 3, 0, 28, 14, 17, 21, 24, 13, 11, 20, 26, 18, 4, 19, 23]	Aptitud: 1946326.0
Mejor combinacion: [15, 10, 1, 2, 7, 27, 22, 16, 5, 25, 8, 12, 6, 9, 29, 3, 0, 28, 14, 17, 21, 24, 13, 11, 20, 26, 18, 4, 19, 23]	Aptitud: 1946326.0
Mejor combinacion: [13, 9, 15, 8, 2, 23, 7, 17, 22, 10, 16, 18, 24, 6, 3, 4, 14, 0, 5, 11, 19, 20, 12, 26, 21, 29, 27, 25, 1, 28]	Aptitud: 1958510.0
Mejor combinacion: [16, 8, 13, 24, 25, 15, 0, 27, 18, 3, 26, 12, 17, 14, 5, 19, 22, 21, 9, 10, 7, 28, 20, 11, 2, 6, 1, 4, 23, 29]	Aptitud: 1950884.0
Mejor combinacion: [16, 7, 22, 4, 27, 1, 6, 10, 8, 15, 13, 25, 11, 29, 24, 26, 14, 12, 28, 23, 18, 3, 20, 19, 2, 5, 17, 9, 0, 21]	Aptitud: 1953238.0
Mejor combinacion: [24, 6, 8, 27, 29, 14, 11, 13, 22, 20, 2, 21, 23, 5, 16, 26, 1, 10, 3, 0, 18, 4, 15, 19, 12, 17, 25, 9, 7, 28]	Aptitud: 1938542.0

Mejor solución de las 20 ejecuciones:

10,25,30,5,19,9,13,26,2,11,14,23, 12,18,28,22,16,24,17,27,1,29,0,2 1,8,15,4,7,3,6	1906424
---	---------

Peor solución de las 20 ejecuciones:

10,18,27,17,6,21,8,29,7,11,2,8,26 ,15,5,13,12,18,24,25,22,0,4,14,2 3,3,1,20,9,16,2	1966804
--	---------

Media después de las 20 ejecuciones:

1948150

● Funciones Propias:

- Selección (SelPropia.java):

Utilizamos un tipo de selección a favor de los mejores candidatos en el cual generamos un número aleatorio(n) entre 0 y 10. Este será el numero de seleccionados y el valor devuelto por la función principal de la clase SelPropia. Mediante la función ordenarPoblacion iremos situando los n mejores individuos al principio de la población.

Una vez terminada la ordenación los mejores individuos se sitúan al principio de la población, se duplican los n primeros individuos de la población en las n últimas posiciones de la misma población, eliminando así los peores individuos de la población que se situaban en las últimas posiciones de la población.

- Mutación (MutacionPropia.java):

Utilizamos un tipo de mutación que decide si mutar o no generando un entero aleatorio entre 0 y el tamaño de la población. Si este valor es menor que la probabilidad mutación * 100 entonces se aplicará la mutación.

Generamos otro entero aleatorio, esta vez entre 0 y la longitud del Cromosoma que determinará la posición a cambiar en cada cromosoma de la población.

En dicha posición se igualará el valor de la siguiente posición de cada cromosoma de la población.

- **Cruce (CrucePropio.java):**

Utilizamos un tipo de cruce definiendo tres listas para cada cromosoma.

```
//linkedList1      3 4 5 2 1
//linkedListFinal1 1 2 3 4 5
//linkedValues1    2 3 4 1 0
```

LinkedList: Lista con valores de 0 a size consecutivos

LinkedValues: Indica en que posición de la linkedListFinal se encuentra el valor de la linkedList1

Finalmente, después de rellenar las listas asignaremos al cromosoma hijo 1 los valores del mismo cromosoma (cromosoma padre 1) en las posiciones pares y los valores del otro cromosoma (cromosoma padre 2) en las posiciones impares. También asignaremos al cromosoma hijo 2 los valores del mismo cromosoma (cromosoma padre 2) en las posiciones pares y los valores del otro cromosoma (cromosoma padre 1) en las posiciones impares.