

EXAMEN FINAL

PAUL MURILLO^{*}, KRITZAN SULLCA^{*}

PROGRAMACIÓN CONCURRENTE

CONTENTS

1	Introducción	
2	Marco Teórico	
2.1	Algoritmos Genéticos	
2.2	Población	
2.3	Generaciones	
2.4	Crossover	
2.5	Mutación	
2.6	Fitness	
3	Metodología	
3.1	Definición de parámetros	
3.2	Diseño de la arquitectura	
3.3	Implementación del servidor	
3.4	Implementación de los trabajadores	
3.5	Prueba en un clúster	
3.6	Desarrollo del programa	
4	Resultados y Discusiones	

ABSTRACT

En este informe se presenta un estudio detallado sobre la implementación de algoritmos genéticos mediante concurrencia con hilos, paralelismo y distribución. Se define una serie de parámetros, como el tamaño de la población, el número de generaciones, el porcentaje de crossover, mutación y elite, y la función de fitness basada en la ecuación descrita en el documento brindado por el docente. Se evalúa el tiempo de ejecución utilizando los parámetros iniciales y se realiza una comparación con los resultados esperados. El enfoque propuesto utiliza un servidor en **LP1** (Java) y **LP2** (Python) para enviar tareas a los trabajadores en **LP1** (Java) y **LP2** (Python), y se realiza una prueba en imágenes de docker que simulan clústers. Se describe en detalle la arquitectura diseñada, que incluye la interacción entre el servidor y los trabajadores, y se presenta un diagrama de protocolo que muestra las etapas del proceso. Además, se proporciona una explicación exhaustiva del desarrollo del programa. Los resultados obtenidos se presentan y se discuten, analizando el tiempo de ejecución y comparándolo con los resultados esperados, mientras que se exploran las ventajas y desventajas de utilizar concurrencia con hilos, paralelismo y distribución en la implementación de algoritmos genéticos.

^{*} Facultad de Ciencias, Universidad Nacional de Ingeniería, Lima, Perú

1 INTRODUCCIÓN

Los algoritmos genéticos son técnicas de optimización inspiradas en la evolución biológica que han demostrado su eficacia en la resolución de problemas complejos. En este artículo, se presenta un estudio exhaustivo sobre la implementación de algoritmos genéticos utilizando concurrencia con hilos, paralelismo y distribución, con el objetivo de mejorar la eficiencia y el tiempo de ejecución de estos algoritmos. El uso de concurrencia y paralelismo permite realizar cálculos simultáneos en diferentes hilos y procesos, mientras que la distribución permite aprovechar los recursos de un clúster para acelerar el proceso de optimización.

En este estudio, se definen varios parámetros cruciales para la implementación de los algoritmos genéticos, como el tamaño de la población, el número de generaciones, el porcentaje de crossover, mutación y elite. Además, se utiliza una función de fitness.

La implementación propuesta utiliza un enfoque cliente-servidor, donde un servidor en LP1 (Java) envía tareas a los trabajadores, que pueden estar implementados en LP1 (Java) o LP2 (Python). Se realiza una prueba en un clúster para evaluar el rendimiento y la escalabilidad del sistema.

Se proporciona una descripción detallada de los algoritmos genéticos, explicando conceptos fundamentales como la población, las generaciones, el crossover, la mutación y la función de fitness. Además, se discute la relevancia y aplicabilidad de los algoritmos genéticos en la resolución de problemas de optimización.

Además incluye el diseño de una arquitectura específica para la implementación de los algoritmos genéticos en un entorno concurrente y distribuido. Se describe la interacción entre el servidor y los trabajadores, y se presenta un diagrama de protocolo que muestra las etapas del proceso, desde el envío de tareas hasta la consolidación de los resultados.

Finalmente, se presentan los resultados obtenidos a través de la ejecución del programa implementado. Se analiza el tiempo de ejecución utilizando los parámetros iniciales y se compara con los resultados esperados. Además, se exploran las ventajas y desventajas de utilizar concurrencia con hilos, paralelismo y distribución en la implementación de algoritmos genéticos, y se discuten posibles mejoras y áreas de investigación futuras, y se incluye el código fuente utilizado en la implementación, en el anexo. Esto proporciona una visión completa del proceso de implementación y los resultados obtenidos.

2 MARCO TEÓRICO

En este apartado se presenta el marco teórico necesario para comprender los fundamentos y conceptos clave relacionados con la implementación de algoritmos genéticos mediante concurrencia con hilos, paralelismo y distribución. A continuación, se describen los principales elementos teóricos:

2.1 Algoritmos Genéticos

Los algoritmos genéticos son técnicas de optimización inspiradas en la evolución biológica. Estos algoritmos simulan la evolución de una población de posibles soluciones a través de mecanismos de selección, reproducción, crossover y mutación. A través de iteraciones sucesivas, los algoritmos genéticos convergen hacia soluciones óptimas o subóptimas del problema en cuestión.

2.2 Población

La población en un algoritmo genético consiste en un conjunto de individuos, donde cada individuo representa una posible solución al problema. La diversidad y el tamaño de la población son factores importantes que influyen en la calidad de las soluciones obtenidas.

2.3 Generaciones

Las generaciones en un algoritmo genético representan las etapas sucesivas de evolución de la población. En cada generación, se aplican los operadores genéticos (crossover y mutación) para crear una nueva generación de individuos, que reemplaza a la generación anterior.

2.4 Crossover

El operador de crossover implica la combinación de información genética de dos individuos seleccionados de la población para crear nuevos individuos. Este proceso imita la reproducción sexual y busca intercambiar y combinar características favorables de los padres para producir descendencia potencialmente mejorada.

2.5 Mutación

La mutación es un operador genético que introduce cambios aleatorios en los individuos de la población. Este proceso permite la exploración del espacio de búsqueda y evita que el algoritmo genético quede atrapado en óptimos locales.

2.6 Fitness

La función de fitness es una medida utilizada para evaluar la calidad de un individuo en términos de su capacidad para resolver el problema objetivo. Cuanto mayor sea el valor de fitness, mejor será la solución representada por el individuo.

El pseudocódigo para la implementación de algoritmos genéticos mediante concurrencia con hilos, paralelismo y distribución se presenta a continuación:

```
Función AlgoritmoGenético(P, G, Crossover, Mutación, Elite):
    // Inicialización de la población
    población = InicializarPoblación(P)
    // Evaluación inicial de la población
    EvaluarPoblación(población)

    Para cada generación g en 1 hasta G:
        // Operadores genéticos
        nuevaGeneración = SeleccionarElite(población, Elite)

    Mientras la nuevaGeneración no esté completa:
```

```

padre1 = SeleccionarPadre(población)
padre2 = SeleccionarPadre(población)
descendiente = AplicarCrossover(padre1, padre2, Crossover)
AplicarMutación(descendiente, Mutación)
nuevaGeneración.AgregarIndividuo(descendiente)

población = nuevaGeneración

// Evaluación de la población
EvaluarPoblación(población)

// Devolver la mejor solución encontrada
mejorSolución = ObtenerMejorSolución(población)
Devolver mejorSolución

```

En este pseudocódigo se muestra la estructura general de un algoritmo genético, incluyendo la inicialización de la población, los operadores genéticos (crossover y mutación), la evaluación de la población y la selección de la mejor solución encontrada. El pseudocódigo puede ser adaptado y complementado según las necesidades específicas de la implementación en un entorno concurrente y distribuido.

Este marco teórico proporciona los conceptos y fundamentos necesarios para comprender la implementación de algoritmos genéticos mediante concurrencia con hilos, paralelismo y distribución. Estos elementos teóricos serán aplicados en la metodología presentada en el siguiente apartado.

3 METODOLOGÍA

3.1 Definición de parámetros

En esta etapa, se realiza una cuidadosa definición de los parámetros necesarios para la implementación de los algoritmos genéticos. Se determina el tamaño de la población (P), el número de generaciones (G) y se ajusta el porcentaje de crossover, mutación y elite de acuerdo con las características del problema y los requisitos de optimización. Además, se especifica la función de fitness utilizada para evaluar la calidad de las soluciones generadas. Por ejemplo, se puede emplear la función de fitness basada en la ecuación:

$$\begin{aligned}
 &1.23 \cdot \sin \frac{x^2 + y^2}{1.0563} \left(\left(\frac{0.4}{1 + 0.02 \cdot ((x + 20)^2 + (y + 20)^2)} \right) \right. \\
 &\quad + \left(\frac{0.2}{1 + 0.05 \cdot ((x + 5)^2 + (y + 25)^2)} \right) \\
 &\quad + \left(\frac{0.7}{1 + 0.01 \cdot ((x)^2 + (y - 30)^2)} \right) \\
 &\quad + \left(\frac{1}{1 + 2 \cdot ((x + 30)^2 + (y)^2)} \right) \\
 &\quad \left. + \left(\frac{0.05}{1 + 0.1 \cdot ((x - 30)^2 + (y + 30)^2)} \right) \right)
 \end{aligned}$$

para evaluar la calidad de las soluciones generadas por los algoritmos genéticos.

donde x e y representan las variables a optimizar. La elección adecuada de estos parámetros es fundamental para lograr un equilibrio entre la exploración y explotación del espacio de búsqueda.

3.2 Diseño de la arquitectura

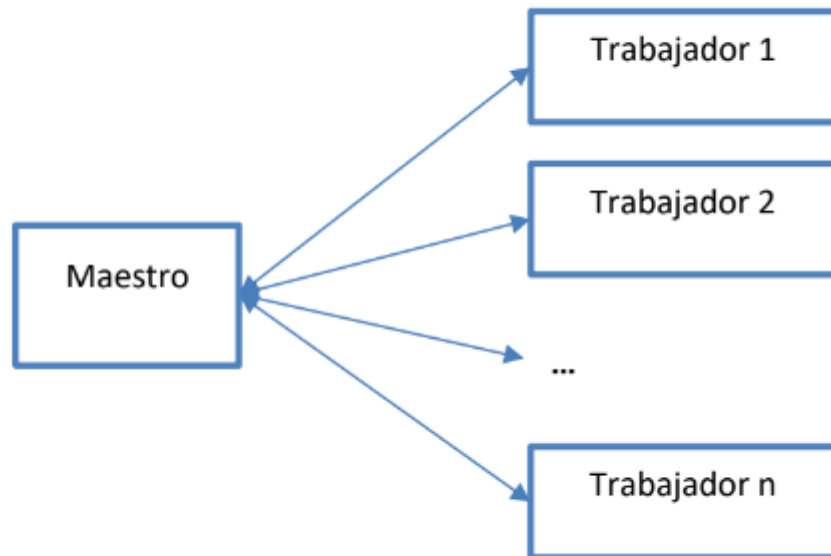
En esta fase, se diseña una arquitectura cliente-servidor que permita la implementación de los algoritmos genéticos en un entorno concurrente y distribuido. El servidor, implementado en LP1 (Java), actúa como punto central de control y gestiona la comunicación entre los clientes y los trabajadores. Se establecen canales de comunicación utilizando sockets u otros mecanismos adecuados para el intercambio de información entre las diferentes partes. Además, se considera la escalabilidad del sistema, permitiendo que nuevos trabajadores se unan al proceso de optimización de manera dinámica.

3.3 Implementación del servidor

En esta etapa, se desarrolla el servidor en LP1 (Java) con el objetivo de gestionar las tareas de los algoritmos genéticos. El servidor recibe las tareas generadas por los clientes, distribuye estas tareas entre los clientes y recopila los resultados obtenidos. Se emplea la concurrencia con hilos para gestionar múltiples solicitudes de tareas simultáneamente y asegurar una adecuada utilización de los recursos del sistema. La implementación del servidor se realiza de manera modular y escalable, permitiendo la expansión de la capacidad de procesamiento de acuerdo con las necesidades.

3.4 Implementación de los trabajadores

En esta etapa, se implementan los trabajadores, los cuales pueden estar desarrollados en LP1 (Java) o LP2 (Python). Los trabajadores son responsables de ejecutar los algoritmos genéticos asignados por el servidor. Utilizan la concurrencia y el paralelismo para ejecutar los cálculos en múltiples hilos o procesos, aprovechando al máximo los recursos disponibles. Cada trabajador recibe una tarea específica del servidor, procesa los cálculos necesarios y devuelve los resultados al servidor. Se aplican técnicas de programación concurrente y paralela para optimizar el rendimiento y la eficiencia de los cálculos realizados por los trabajadores.



3.5 Prueba en un clúster

Para evaluar el rendimiento y la escalabilidad del sistema implementado, se realiza una prueba en un clúster. El clúster consta de un conjunto de máquinas interconectadas que se utilizan para distribuir y acelerar el proceso de optimización. Se analiza el rendimiento en función del tamaño del clúster y la carga de trabajo asignada a cada máquina. Se recopilan métricas de rendimiento, como el tiempo de ejecución y la eficiencia del sistema, para evaluar la capacidad de escalabilidad y la utilización efectiva de los recursos del clúster.

3.6 Desarrollo del programa

En esta etapa, se explica en detalle el desarrollo del programa basándose en las explicaciones y el código proporcionado en clase. Se describen los pasos necesarios para implementar la lógica de los algoritmos genéticos utilizando las capacidades de concurrencia, paralelismo y distribución según lo definido en la arquitectura diseñada. Se pueden utilizar librerías específicas, como la Java Concurrency API o herramientas de programación paralela en Python, para facilitar la implementación y la gestión de la concurrencia y el paralelismo en el programa. Se realizan pruebas exhaustivas y se ajustan los parámetros y la arquitectura según sea necesario para optimizar el rendimiento y la eficacia del sistema.

4 RESULTADOS Y DISCUSIONES

La implementación de algoritmos genéticos mediante concurrencia con hilos, paralelismo y distribución representa un enfoque prometedor para mejorar la eficiencia y el tiempo de ejecución de estos algoritmos. La metodología presentada en este artículo brinda una guía detallada para implementar de manera efectiva estos algoritmos en entornos concurrentes y distribuidos. Sin embargo, se debe tener en cuenta que cada problema puede requerir ajustes y optimizaciones específicas, por lo que se recomienda un análisis exhaustivo y una adaptación adecuada para obtener los mejores resultados en cada caso.

- Para 50 clientes, 25 en java y 25 en python con dos generaciones se obtiene lo siguiente

```

[SERVER] Nuevo cliente conectado (Clientes conectados: 50)

Generación número 8

[SERVER] Mandando población: 19.620278/1.27339/0.00000,8.79665/14.76955/0.00000,6.209419/8.579563/0.00000,19.392597/4.278238/0.00000,16.815888/4.729739/0.00000,15.73546
[CLIENTE 0] 17.204291/6.086532/0.00000,17.564037/3.279652/0.00000,8.517510/14.061989/0.00000,15.802252/16.484829/0.00000,4.873019/13.851951/0.00000,6.801255/2.41498/0.0000
[CLIENTE 1] 15.837333/2.776639/0.00000,6.801255/2.41498/0.00000,6.538538/7.952427/0.00000,14.31939/7.799238/0.00000,2.887547/1.48895/0.00000,4.44768/6.567024/0.00000,6
[CLIENTE 2] 10.262292/5.377892/0.00000,10.262292/5.377892/0.00000,10.584911/16.49589/0.00000,8.238414/13.354121/0.00000,5.056704/13.883781/0.00000,8.815259/6.397991/0.0000
[CLIENTE 3] 11.55585/8.472238/0.00000,2.887547/1.48895/0.00000,10.473895/13.731999/0.00000,17.204291/8.866532/0.00000,16.494139/7.953318/0.00000,11.27679/7.764372/0.0000
[CLIENTE 4] 10.801431/6.513786/0.00000,7.559354/7.116832/0.00000,6.894782/6.442333/0.00000,5.847463/4.472893/0.00000,4.501040/7.104744/0.00000,6.533158/11.38531/0.00000,11
[CLIENTE 5] 12.801883/6.428981/0.00000,8.445749/4.926702/0.00000,16.275887/3.595402/0.00000,10.346159/7.813242/0.00000,12.26864/6.525468/0.00000,13.959452/12.17211/0.0000
[CLIENTE 6] 11.18185/14.237604/0.00000,13.868146/8.783444/0.00000,3.512024/8.322993/0.00000,4.895417/11.418849/0.00000,8.897763/5.158977/0.00000,9.918851/6.944277/0.00000
[CLIENTE 7] 16.543832/14.569942/0.00000,8.894354/6.957448/0.00000,11.583288/6.555333/0.00000,9.905851/10.258462/0.00000,8.863352/6.898883/0.00000,10.262292/5.377892/0.00000
[CLIENTE 8] 5.512024/8.322993/0.00000,6.538538/7.952427/0.00000,6.190959/6.300789/0.00000,8.814429/8.064342/0.00000,5.997163/0.219713/0.00000,11.659187/3.488199/0.00000,7.5
[CLIENTE 9] 10.346159/7.813242/0.00000,10.597731/12.527883/0.00000,8.713882/6.739537/0.00000,6.801255/2.41498/0.00000,10.264238/6.593789/0.00000,17.564037/3.279652/0.00000
[CLIENTE 10] 9.46824/7.31288/0.00000,12.98982/9.431909/0.00000,9.451861/11.240667/0.00000,15.735486/2.281045/0.00000,6.544858/4.729564/0.00000,5.462014/6.568161/0.00000
[CLIENTE 11] 1.31749/6.52109/0.00000,8.897343/8.150097/0.00000,12.001803/6.42801/0.00000,10.597731/12.527883/0.00000,11.397891/1.588395/0.00000,12.520145/5.957807/0.00000
[CLIENTE 12] 9.885226/9.177495/0.00000,7.741658/9.98238/0.00000,7.279584/12.06985/0.00000,9.964143/6.70881/0.00000,8.713882/6.739537/0.00000,15.805374/2.27843/0.00000,6
[CLIENTE 13] 15.432386/10.965140/0.00000,3.847463/4.472893/0.00000,12.441257/11.155781/0.00000,13.959452/12.517111/0.00000,3.678149/7.450136/0.00000,11.848893/2.884442/0.000
[CLIENTE 14] 14.31939/7.799238/0.00000,12.257348/12.086496/0.00000,4.114321/4.47824/0.00000,11.136442/2.578891/0.00000,5.05716/10.45195/0.00000,6.538538/7.952427/0.0000
[CLIENTE 15] 2.819222/8.294484/0.00000,10.18947/9.652594/0.00000,9.279584/12.069885/0.00000,7.218975/2.170572/0.00000,14.873648/12.183499/0.00000,10.972453/5.438314/0.0000
[CLIENTE 16] 11.132883/2.984586/0.00000,11.326491/14.320965/0.00000,4.853124/12.661795/0.00000,6.501040/7.104744/0.00000,14.31939/7.799238/0.00000,8.815259/6.397991/0.0000
[CLIENTE 17] 8.896354/6.557448/0.00000,10.972453/5.438314/0.00000,10.44464/6.686441/0.00000,11.435755/15.54386/0.00000,5.62715/5.798353/0.00000,7.838451/7.82469/0.00000
[CLIENTE 18] 7.473884/14.654287/0.00000,8.781188/1.074692/0.00000,4.526522/10.710182/0.00000,7.593354/7.116832/0.00000,13.959452/12.517111/0.00000,13.455755/13.54386/0.0000
[CLIENTE 19] 3.489847/8.664448/0.00000,12.441257/11.155781/0.00000,8.877354/1.287775/0.00000,8.172881/5.515208/0.00000,14.364857/7.731153/0.00000,6.209419/8.579563/0.00000
[CLIENTE 20] 9.46824/7.31288/0.00000,2.183874/5.98839/0.00000,9.885226/9.177495/0.00000,7.462721/5.461978/0.00000,6.821664/9.278831/0.00000,4.628322/14.395291/0.00000,4
[CLIENTE 21] 12.884247/9.74977/0.00000,7.727814/9.940893/0.00000,11.748983/3.001789/0.00000,6.637335/11.843485/0.00000,12.548323/8.288742/0.00000,6.821664/9.278831/0.00000
[CLIENTE 22] 12.548323/8.288742/0.00000,8.79665/14.76955/0.00000,7.727814/9.940893/0.00000,6.69574/16.829194/0.00000,10.28838/3.895822/0.00000,4.92842/10.82579/0.0000
[CLIENTE 23] 5.713882/6.739537/0.00000,7.564897/5.128432/0.00000,15.565227/3.452180/0.00000,17.204291/8.866532/0.00000,3.489847/8.664448/0.00000,15.128192/9.133666/0.00000
[CLIENTE 24] 4.114321/4.47824/0.00000,8.278587/5.17354/0.00000,4.982646/5.82344/0.00000,4.449768/0.587034/0.00000,12.886247/9.74977/0.00000,10.18947/9.652594/0.00000,1
  
```

```

[CLIENTE 26] 12.257348/12.086496/0.00000,4.526522/10.710182/0.00000,10.776696/7.3857715/0.00000,8.216895/7.7910845/0.00000,4.349227/13.687424499999999/0.00000,10.262292/5.377892/0.00000,5.05716/10.45189
[CLIENTE 27] 4.882421499999999/11.429118/0.00000,10.3461585/7.8110415/0.00000,15.735486/2.281045/0.00000,11.7088865/6.7384215/0.00000,11.459384999999999/3.488199999999999/0.00000,11.888888/15.4699405/0.00000,13.9
[CLIENTE 28] 5.7483353/4.670045000000001/0.00000,15.879826999999999/9.784375/0.00000,8.778842/14.193484999999999/0.00000,7.2239145/10.946842/0.00000,15.8022515/16.484829/0.00000,12.441257/11.155781000000001/0.00000
[CLIENTE 29] 11.88984/7.5868845/0.00000,14.315165/2.176521/0.00000,15.8022515/16.484829/0.00000,14.319395/7.79923/0.00000,12.548323/8.288742/0.00000,7.216895/7.7910845/0.00000,8.89255/6.858546/0.00000,12.801883/6.428981/0.00000
[CLIENTE 30] 7.475315/9.9668715/0.00000,9.2317925/2.7874385/0.00000,15.4879375/4.4867855/0.00000,15.825258000000001/2.227843/0.00000,16.161975/11.9579485/0.00000,14.6775885/7.582919/0.00000,10.161975/11.9579485/0.00000
[CLIENTE 31] 6.7124525/8.487788/0.00000,8.63784/9.3782338/0.00000,8.89255/6.858546/0.00000,7.475315/9.9668715/0.00000,5.512024/8.322992500000001/0.00000,11.713882/6.739537/0.00000,10.95716/10.45189/0.00000,6.641770800
[CLIENTE 32] 6.7124525/8.487788/0.00000,8.8152585/6.397990500000001/0.00000,13.136419999999999/2.578891/0.00000,8.79665/14.76955/0.00000,7.757111/10.788555/0.00000,8.877354/1.2877745/0.00000,15.128192/9.133666/0.00000
[CLIENTE 33] 5.646177000000001/5.0115865/0.00000,7.49413/7.9531/0.00000,16.5438315/14.509841999999999/0.00000,7.559355000000001/7.116832499999999/0.00000,7.2239145/10.946842/0.00000,9.88522/9.177495/0.00000,7.223
[CLIENTE 34] 5.7483353/4.670045000000001/0.00000,7.588102/11.674789/0.00000,6.778842/14.193484999999999/0.00000,11.831948899999914/13.28761481826548/0.00000,16.983774999999999/7.782961999999999/0.00000,15.859
[CLIENTE 35] 14.894584/6.5264045/0.00000,14.319395/7.79923/0.00000,4.783405/11.759594999999999/0.00000,12.441257/11.155781000000001/0.00000,10.262292/5.377892/0.00000,16.4843
[CLIENTE 36] 10.597731/12.527883000000001/0.00000,15.8022515/16.484829/0.00000,10.262292/5.377892/0.00000,2.107936/6.540165/0.00000,9.743862/18.8991285/0.00000,14.319395/7.79923/0.00000,15.642014999999999/6.56601
[CLIENTE 37] 1.416925/4.188363000000001/0.00000,8.781188/1.0746915/0.00000,13.455755000000001/13.54386/0.00000,11.88984/7.5868845/0.00000,10.801431999999999/6.513786/0.00000,10.95716/10.451899999999999/6.898883/0.00000
[CLIENTE 38] 8.1671935/1.234551000000002/0.00000,6.894782/6.442333499999999/0.00000,17.5848365/3.2796515/0.00000,10.3461585/7.8110415/0.00000,10.28588/3.895200000000003/0.00000,16.983774999999999/7.7829614
[CLIENTE 39] 11.214835/8.399652/0.00000,5.632714999999999/5.798353000000005/0.00000,1.2396310000000001/7.157086/0.00000,7.46958/3.801319/0.00000,3.489847499999999/7.864448499999999/0.00000,9.570431499999999
[CLIENTE 40] 7.275855/12.069885/0.00000,7.564897/5.128432000000005/0.00000,11.888888/15.4699405/0.00000,14.33347/10.959341999999999/0.00000,5.571857/7.4872835/0.00000,7.564897/5.128432000000005/0.00000,4.85312
[CLIENTE 41] 6.821664999999999/9.2788315/0.00000,8.517959/14.861988/0.00000,10.884808499999999/17.203757/0.00000,8.8152585/6.397990500000001/0.00000,10.1842375/4.5039885/0.00000,13.888345/5.7836435/0.00000,8.796
[CLIENTE 42] 6.821664999999999/9.2788315/0.00000,8.216895/7.7910845/0.00000,15.8373325/2.7768385/0.00000,11.512635/6.654451/0.00000,14.873648/12.183499/0.00000,9.8355185/2.628963/0.00000,10.26888/9.895200000000003/0.00000
[CLIENTE 43] 4.853129999999999/12.661475/0.00000,10.1842375/4.5039885/0.00000,13.815580499999999/8.8161795/0.00000,5.128010000000001/6.8243025/0.00000,11.512635/6.654451/0.00000,10.884808499999999/17.203757
[CLIENTE 44] 14.368495/8.557893/0.00000,10.182878/1.273839/0.00000,5.05716/10.45189/0.00000,8.863252499999999/6.898883/0.00000,16.161975/11.9579485/0.00000,13.67288/17.462671999999999/0.00000,16.67478
[CLIENTE 45] 7.838459999999999/7.824994999999999/0.00000,6.837335/11.843485/0.00000,6.778842/14.193484999999999/0.00000,11.7088865/6.7384215/0.00000,7.5483353/4.670045000000001/0.00000,10.26888/9.895200000000003/0.00000
[CLIENTE 46] 6.35865/8.784315/0.00000,9.5465475/13.8763635/0.00000,5.662014999999999/6.568161/0.00000,10.597731/12.527883000000001/0.00000,7.3876425/8.8976145/0.00000,17.204291/8.866532/0.00000,13.447280000000000
[CLIENTE 47] 6.1989585/6.3007875/0.00000,9.933405/7.8564935/0.00000,4.9573395/10.793441/0.00000,6.53725890250559/7.9328526388551/0.00000,8.238414/13.354121/0.00000,13.9594515/12.517105000000001/0.00000,10.31199
[CLIENTE 48] 12.776696/7.3857715/0.00000,16.5438315/14.509841999999999/0.00000,13.672988/17.462671999999999/0.00000,7.172881/5.515195/0.00000,11.397891/1.588395/0.00000,7.475315/9.9668715/0.00000,9.9353185/1.620
[CLIENTE 49] 8.8152585/6.397990500000001/0.00000,9.468246000000001/7.31398/0.00000,11.276789/7.764371/0.00000,12.486215/1.77423/0.00000,8.8152585/6.397990500000001/0.00000,9.934485/7.8564935/0.00000,1.416925/4.
  
```

```

[SERVER] Mandando población: 8.815259/6.397991/118.643865,9.488246/7.313988/102.99858,11.27679/7.764371/187.453438,12.488622/1.724238/159.18858,8.815259/6.397991/118.643865
[CLIENTE 0] 12.45826/6.448318/0.00000,11.151446/11.934977/0.00000,4.79789/4.571467/0.00000,12.91128/8.498883/0.00000,7.812586/6.498875/0.00000,17.204291/8.866532/0.00000
[CLIENTE 1] 7.918862/7.33289/0.00000,7.882537/4.486458/0.00000,9.976957/6.794292/0.00000,7.938322/8.778727/0.00000,8.128252/5.521382/0.00000,9.580789/4.788417/0.00000,12.3
[CLIENTE 2] 11.859024/9.48876/0.00000,12.97740/11.79331/0.00000,10.847885/15.48891/0.00000,10.463941/10.437444/0.00000,9.258353/9.345607/0.00000,6.739524/12.427714/0.0000
[CLIENTE 3] 6.452518/3.42582/0.00000,10.185572/4.95115/0.00000,12.76854/10.133657/0.00000,7.544957/10.43518/0.00000,11.416338/1.18385/0.00000,4.853124/12.661475/0.0000
[CLIENTE 4] 14.411172/10.37877/0.00000,8.120191/7.483385/0.00000,8.278249/5.218378/0.00000,12.658847/10.435678/0.00000,5.995974/9.574701/0.00000,11.144597/12.461475/0.0000
[CLIENTE 5] 12.533247/4.77181/0.00000,10.39488/9.36228/0.00000,12.368716/4.216852/0.00000,12.224284/10.518301/0.00000,8.868428/8.498293/0.00000,12.857271/4.45549/0.00000
[CLIENTE 6] 10.701993/10.86883/0.00000,9.278234/12.83488/0.00000,9.278234/12.83488/0.00000,4.574629/11.385244/0.00000,9.884387/8.547812/0.00000,9.484678/7.315495/0.00000
[CLIENTE 7] 9.43848/9.29185/0.00000,12.48362/9.943487/0.00000,12.90948/13.28986/0.00000,10.168388/9.888498/0.00000,8.984537/8.174743/0.00000,9.942891/10.490489/0.00000
[CLIENTE 8] 11.475972/10.497815/0.00000,10.223978/8.448468/0.00000,6.838338/7.952427/0.00000,11.492186/6.388571/0.00000,11.411377/157170/0.00000,11.412296/16.06616/0.0000
[CLIENTE 9] 7.2864/11.78829/0.00000,12.002819/10.91191/0.00000,4.370117/8.81363/0.00000,9.36881/4.412612/0.00000,11.884891/10.11577/0.00000,11.788371/9.8384/0.00000
[CLIENTE 10] 5.107514/6.08945/0.00000,12.88767/6.48919/0.00000,9.85173/10.49245/0.00000,8.85831/12.481379/0.00000,7.133958/4.488418/0.00000,8.85831/12.481379/0.00000,11.412296/16.06616/0.0000
[CLIENTE 11] 7.421710/3.45982/0.00000,12.561964/5.198242/0.00000,12.28461/5.17631/0.00000,10.807238/7.538434/0.00000,12.801883/6.428981/0.00000,10.308954/7.855182/0.0000
[CLIENTE 12] 9.162985/13.21284/0.00000,3.488483/4.78478/0.00000,8.38193/11.20218/0.00000,10.89552/6.87137/0.00000,5.448878/6.323928/0.00000,5.949838/10.58890/0.00000
[CLIENTE 13] 7.82498/7.74863/0.00000,9.85623/5.74457/0.00000,13.832272/11.03046/0.00000,5.912771/11.826749/0.00000,7.917361/4.555332/0.00000,14.834149/12.085145/0.0000
[CLIENTE 14] 5.94928/7.74229/0.00000,11.11974/9.187264/0.00000,9.216886/6.113627/0.00000,9.93843/10.914642/0.00000,9.848458/5.654223/0.00000,15.735486/2.281045/0.00000
[CLIENTE 15] 12.592821/12.058462/0.00000,15.93815/7.08648/0.00000,12.83898/5.15275/0.00000,8.618283/4.342179/0.00000,16.388485/3.827845/0.00000,12.1898
```

```

Run Server Client-Handler
[CLIENTE 30] 11.99119475/10.1702335/0,12.880224249999999/9.8913240808080801/0,11.77559/5.2483425/0,10.29079425/10.30095/0,9.916882999999999/9.70404225/0,8.847256/5.843640000
[CLIENTE 31] 9.18845275/8.4954025/0,10.45200475/8.95034675/0,8.138985808080801/5.92984/0,7.1420705808080801/7.19341/0,13.4930885/10.2357675/0,8.8273575/0,9.3387305/
[CLIENTE 32] 12.859417999999999/8.890475999999999/0,3.9457137500000004/5.28918/0,8.9409125/11.973675/0,8.3172325/5.998164999999999/0,7.793650249999999/10.2957435/0,11.5512679999
[CLIENTE 33] 16.5438315/14.509041999999999/0,9.88344/9.2256670000000005/0,10.83632475/10.66707225/0,5.45642975/10.14957875/0,10.7943915/9.8489975/0,11.886117499999999/12.0954567
[CLIENTE 34] 17.7634855/8.47921425/0,6.62568275/8.17237475/0,7.275647999999994/7.04612675/0,8.115544/10.753738249999998/0,17.7634855/8.47921425/0,6.52354025/11.55603825/0,9.9925
[CLIENTE 35] 8.516439999999999/9.6691785/0,15.294329999999999/8.73867825/0,9.9263335/9.149483/0,5.736549999999999/7.377665749999999/0,12.441257/11.155781808080801/0,14.35417325/1
[CLIENTE 36] 12.3817215/8.23833525/0,11.13208325/9.2292135/0,10.5351765/12.61801175/0,10.27382675/7.42934025/0,7.76122325/10.118481/0,9.8434755/5.3522875808080801/0,11.670402/10.
[CLIENTE 37] 11.8883635/8.989167499999999/0,12.484204000000002/10.098855/0,10.98488725/8.4847675/0,9.832342/6.3823345/0,7.47177475/5.4852135/0,8.4695885808080801/0,12.45296/0,9.44
[CLIENTE 38] 14.208115499999999/8.21888625/0,13.734945000000001/6.7691265/0,7.559894249999999/9.839186/0,8.88471775/6.64849225/0,12.755284096653834/10.58655849751017/0,11.500179
[CLIENTE 39] 15.28148225/6.59538025/0,12.680453499999999/6.3414485/0,12.320797249999998/11.33036725/0,9.594673749999998/6.61069258080801/0,8.56765525/9.6162125/0,10.3315425/0.
[CLIENTE 40] 8.5251175/13.481487/0,9.780804/10.4668205/0,9.57266675/5.47275475/0,12.6529475/0.194837999999999/0,6.87197/6.5812415/0,12.3740975/9.4396875/0,8.671985499999999/8.72
[CLIENTE 41] 11.418453499999999/8.926189/0,7.923325999999999/11.513595808080801/0,8.28127314319748/5.377999863568973/0,9.2114365/6.158678999999999/0,12.718543999999999/11.92247
[CLIENTE 42] 5.82643175/13.8113174999999/0,6.84562/6.53967475/0,8.41915875/0.4135615/0,12.05272025/7.44556875/0,7.17133375808080805/10.371559/0,10.488444758080802/6.9843155/0.
[CLIENTE 43] 10.99419075/7.1656895/0,9.68384925/8.78627675/0,10.709253/8.45662625/0,8.80331/8.923476808080801/0,7.99641225/12.814829749999999/0,7.858562499999999/14.932636/0,12.1
[CLIENTE 44] 9.993405/7.6556935/0,10.80473825/8.21542875/0,9.48782725/6.8832795/0,15.1281915/9.133664/0,9.463245/6.828634/0,5.6752375/8.673801/0,10.27382675/7.42934025/0,10.9495
[CLIENTE 45] 11.28645675/15.43916175/0,10.93085475/8.78564675/0,9.17253575/0.833139999999999/0,4.139928/6.275326499999999/0,14.4312985/12.73423625/0,12.75776825/10.9464525808080
[CLIENTE 46] 15.13218/8.39588775/0,3.5826395/4.42502175/0,14.4049385/11.517651/0,13.73745675/12.8356425/0,6.21010775/11.547586/0,12.347379499999999/11.58601475/0,8.84884/9.9135
[CLIENTE 47] 13.297775808080801/2.95030725/0,8.45398425/6.63826825/0,8.8868115/9.596546499999999/0,8.45398425/6.63826825/0,5.218744/9.631228249999999/0,8.255476/9.1170575/0,8.99
[CLIENTE 48] 15.18848975/15.986256999999999/0,14.83417775/5.81304075/0,9.5785295/10.1234745/0,10.447682/8.34178975/0,14.341071/9.7586275/0,12.49043975/9.521882499999999/0,15.9487
[CLIENTE 49] 12.458847499999999/10.4358675/0,12.877846/11.870261249999999/0,9.18865275/8.69854025/0,12.877846/11.870261249999999/0,11.38845675/5.558661/0,12.801803/6.4289085/0,1

El mejor individuo encontrado es:
x = 12.658847
y = 10.43587
Valor de la función: 269.137893274309
Tiempo Final:0.597

```

- Ahora tomaremos diferentes NumGeneraciones = 2,10,30,60,100,150,300 y mediremos los tiempos

```

Run Server Client-Handler
[CLIENTE 30] 11.99119475/10.1702335/0,12.880224249999999/9.8913240808080801/0,11.77559/5.2483425/0,10.29079425/10.30095/0,9.916882999999999/9.70404225/0,8.847256/5.843640000
[CLIENTE 31] 9.18845275/8.4954025/0,10.45200475/8.95034675/0,8.138985808080801/5.92984/0,7.1420705808080801/7.19341/0,13.4930885/10.2357675/0,8.8273575/0,9.3387305/
[CLIENTE 32] 12.859417999999999/8.890475999999999/0,3.9457137500000004/5.28918/0,8.9409125/11.973675/0,8.3172325/5.998164999999999/0,7.793650249999999/10.2957435/0,11.5512679999
[CLIENTE 33] 16.5438315/14.509041999999999/0,9.88344/9.2256670000000005/0,10.83632475/10.66707225/0,5.45642975/10.14957875/0,10.7943915/9.8489975/0,11.886117499999999/12.0954567
[CLIENTE 34] 17.7634855/8.47921425/0,6.62568275/8.17237475/0,7.275647999999994/7.04612675/0,8.115544/10.753738249999998/0,17.7634855/8.47921425/0,6.52354025/11.55603825/0,9.9925
[CLIENTE 35] 8.516439999999999/9.6691785/0,15.294329999999999/8.73867825/0,9.9263335/9.149483/0,5.736549999999999/7.377665749999999/0,12.441257/11.155781808080801/0,14.35417325/1
[CLIENTE 36] 12.3817215/8.23833525/0,11.13208325/9.2292135/0,10.5351765/12.61801175/0,10.27382675/7.42934025/0,7.76122325/10.118481/0,9.8434755/5.3522875808080801/0,11.670402/10.
[CLIENTE 37] 11.8883635/8.989167499999999/0,12.484204000000002/10.098855/0,10.98488725/8.4847675/0,9.832342/6.3823345/0,7.47177475/5.4852135/0,8.4695885808080801/0,12.45296/0,9.44
[CLIENTE 38] 14.208115499999999/8.21888625/0,13.734945000000001/6.7691265/0,7.559894249999999/9.839186/0,8.88471775/6.64849225/0,12.755284096653834/10.58655849751017/0,11.500179
[CLIENTE 39] 15.28148225/6.59538025/0,12.680453499999999/6.3414485/0,12.320797249999998/11.33036725/0,9.594673749999998/6.61069258080801/0,8.56765525/9.6162125/0,10.3315425/0.
[CLIENTE 40] 8.5251175/13.481487/0,9.780804/10.4668205/0,9.57266675/5.47275475/0,12.6529475/0.194837999999999/0,6.87197/6.5812415/0,12.3740975/9.4396875/0,8.671985499999999/8.72
[CLIENTE 41] 11.418453499999999/8.926189/0,7.923325999999999/11.513595808080801/0,8.28127314319748/5.377999863568973/0,9.2114365/6.158678999999999/0,12.718543999999999/11.92247
[CLIENTE 42] 5.82643175/13.8113174999999/0,6.84562/6.53967475/0,8.41915875/0.4135615/0,12.05272025/7.44556875/0,7.17133375808080805/10.371559/0,10.488444758080802/6.9843155/0.
[CLIENTE 43] 10.99419075/7.1656895/0,9.68384925/8.78627675/0,10.709253/8.45662625/0,8.80331/8.923476808080801/0,7.99641225/12.814829749999999/0,7.858562499999999/14.932636/0,12.1
[CLIENTE 44] 9.993405/7.6556935/0,10.80473825/8.21542875/0,9.48782725/6.8832795/0,15.1281915/9.133664/0,9.463245/6.828634/0,5.6752375/8.673801/0,10.27382675/7.42934025/0,10.9495
[CLIENTE 45] 11.28645675/15.43916175/0,10.93085475/8.78564675/0,9.17253575/0.833139999999999/0,4.139928/6.275326499999999/0,14.4312985/12.73423625/0,12.75776825/10.9464525808080
[CLIENTE 46] 15.13218/8.39588775/0,3.5826395/4.42502175/0,14.4049385/11.517651/0,13.73745675/12.8356425/0,6.21010775/11.547586/0,12.347379499999999/11.58601475/0,8.84884/9.9135
[CLIENTE 47] 13.297775808080801/2.95030725/0,8.45398425/6.63826825/0,8.8868115/9.596546499999999/0,8.45398425/6.63826825/0,5.218744/9.631228249999999/0,8.255476/9.1170575/0,8.99
[CLIENTE 48] 15.18848975/15.986256999999999/0,14.83417775/5.81304075/0,9.5785295/10.1234745/0,10.447682/8.34178975/0,14.341071/9.7586275/0,12.49043975/9.521882499999999/0,15.9487
[CLIENTE 49] 12.458847499999999/10.4358675/0,12.877846/11.870261249999999/0,9.18865275/8.69854025/0,12.877846/11.870261249999999/0,11.38845675/5.558661/0,12.801803/6.4289085/0,1

El mejor individuo encontrado es:
x = 12.658847
y = 10.43587
Valor de la función: 269.137893274309
Tiempo Final:0.597

```

```

Client.java Server.java Util.java ClientHandler.java Main.java
Download grammar and spelling checker for Asturian?
9 static final int POPULATION_SIZE = 20;
10 static final int NUM_GENERACIONES = 10;
11 static final double LOWER_BOUND = 0;

Run Server Client-Handler
[CLIENTE 38] 9.363185303710939/9.527014391601561/0,9.361108283922562/9.183683896868701/0,9.4465725209793/9.171580868835198/0,9.647888236957747
[CLIENTE 39] 9.79785664335374/8.830885496010816/0,9.871163876915728/8.873597183185185/0,8.631227469730426/8.748511967941436/0,9.89964122525261
[CLIENTE 40] 8.104974692246082/9.684345628018065/0,7.672655691737693/0.81948691024263/0,8.018348275089718/9.425578086482467/0,8.07147226807942
[CLIENTE 41] 7.7499903633093386/6.938436428760648/0,7.988963873844973/7.0748368074517015/0,7.880652285658572/6.778586369653762/0,7.83589120587
[CLIENTE 42] 6.389468565634919/7.97339154595625/0,6.586268773503581/8.069663339551223/0,6.167923071818986/7.83891567896969/0,6.4106029066779
[CLIENTE 43] 6.375102773024495/7.44884117251789/0,8.271389468789314/7.058916323111491/0,8.245834344556483/7.16229173102998/0,8.544539585125
[CLIENTE 44] 6.821632038079362/9.554976489044587/0,6.567922562496413/9.591140643342503/0,6.753877825189335/9.58642492796667/0,6.75079216698621
[CLIENTE 45] 9.336868698429688/8.821867404296875/0,9.374638087898624/8.1548787799804687/0,9.218451785156251/7.842383892177501/0,9.8335669353834
[CLIENTE 46] 8.13880614578098/6.1518782565764968/0,7.96253447134644/6.075963347225115/0,8.04832844535422/6.0178458964272245/0,8.0193712562968
[CLIENTE 47] 6.76338517143954/8.091625810390568/0,6.6845691169697785/8.098948544355441/0,6.785453278072352/8.355188883632757/0,6.7693507691335
[CLIENTE 48] 5.820918949479558/6.459963861456897/0,5.69380776296311/8.62093884632811/0,5.90290227598625/8.42595421875/0,5.92383859375/8.2
[CLIENTE 49] 10.681155141038873/8.788489534143879/0,10.522213989165513/8.596795262329788/0,10.79848777498861/8.657228324890767/0,11.223513897

El mejor individuo encontrado es:
x = 10.601155
y = 8.78841
Valor de la función: 188.22889286212502
Tiempo Final:2.591

```


RESULTADOS Y DISCUSIONES

```
Client.java x Server.java x Utils.java x ClientHandler.java x Main.java x
Download grammar and spelling checker for Asturian?

9      3 usages
      static final int POPULATION_SIZE = 20;
      2 usages
10     static final int NUM_GENERACIONES = 30;
      2 usages
11     static final double LOWER_BOUND = 0;
      2 usages

Run: Server x ClientHandler x
[CLIENTE 38] 10.438536592714586/10.605482759445717/0,10.437854735172236/10.606002234016369/0,10.438411713176517/10.60559633113
[CLIENTE 39] 9.555966299846247/10.831861244480955/0,9.555947224839695/10.831861526652814/0,9.555980192884647/10.83183083847785
[CLIENTE 40] 14.568209323794168/12.889795182642892/0,14.568257957349598/12.889898013805273/0,14.568206511965664/12.88984030886
[CLIENTE 41] 11.051063725405888/12.689788577601989/0,11.050992449949655/12.689683623367765/0,11.050931473888415/12.68973541597
[CLIENTE 42] 10.729747253462484/9.322351856464557/0,10.728322214136584/9.320593747678595/0,10.727884173950981/9.32028688357143
[CLIENTE 43] 10.797156196881033/10.368215578487374/0,10.78867521701247/10.367988334047616/0,10.796624574297695/10.354178348683
[CLIENTE 44] 11.748936328977576/12.57768576586217/0,11.748986658563179/12.57798637744516/0,11.74894998615412/12.57775066386844
[CLIENTE 45] 11.828114544743167/12.11787399683492/0,11.827974088025694/12.118422178048595/0,11.827888307174184/12.117438657510
[CLIENTE 46] 10.612362439861327/10.543434878103887/0,10.611823788367658/10.543956405440387/0,10.6121404270811/10.5433426471845
[CLIENTE 47] 12.320428869963316/10.84989592464939/0,12.320742684247936/10.847795673824843/0,12.320473417432103/10.848846766328
[CLIENTE 48] 9.08912845723749/10.720767578128951/0,9.089056775579124/10.720610966358057/0,9.0896208099220113/10.721958252986536
[CLIENTE 49] 12.08952920417657/9.18257085366863/0,12.084589774739863/9.18591087959826/0,12.091317402066618/9.181081223231328/

El mejor individuo encontrado es:
x = 12.089529
y = 9.18257
Valor de la función: 230.47630324674103
Tiempo final:7.563
```

```
Client.java x Server.java x Utils.java x ClientHandler.java x Main.java x
Download grammar and spelling checker for Asturian?

9      3 usages
      static final int POPULATION_SIZE = 20;
      2 usages
10     static final int NUM_GENERACIONES = 60;
      2 usages
11     static final double LOWER_BOUND = 0;
      2 usages

Run: Server x ClientHandler x
[CLIENTE 38] 7.9865133141948/7.2288696467886435/0,7.986589211482951/7.228981531063563/0,7.986430
[CLIENTE 39] 11.211974645160076/6.86099163978626/0,11.212288724665992/6.860784387976412/0,11.211
[CLIENTE 40] 10.487583964687612/7.232697609452311/0,10.487588350132409/7.232705963969879/0,10.48
[CLIENTE 41] 9.21949517595364/6.115470805618926/0,9.219464628712553/6.115472262235521/0,9.219577
[CLIENTE 42] 12.09270846560603/10.070482267092455/0,12.092715127987026/10.070483056642923/0,12.0
[CLIENTE 43] 11.08302482969243/7.091159651820842/0,11.076181961172352/7.077597664137089/0,11.076
[CLIENTE 44] 11.163840808987212/6.356914348400668/0,11.163000192062952/6.356571051685984/0,11.16
[CLIENTE 45] 11.179724393549636/8.014761116546977/0,11.179723356719071/8.014762309394172/0,11.17
[CLIENTE 46] 9.958063864692367/8.241405594067233/0,9.958144555543358/8.241563083354972/0,9.95749
[CLIENTE 47] 9.376086577922269/5.586818429162701/0,9.3760808991605708/5.586798731831399/0,9.37608
[CLIENTE 48] 12.479102559735514/5.881599971357681/0,12.479092018661301/5.88159553366124/0,12.479
[CLIENTE 49] 9.400554047357199/7.74116688177441/0,9.39999711209404/7.741653828271998/0,9.4000608

El mejor individuo encontrado es:
x = 9.400554
y = 7.741167
Valor de la función: 148.296082028805
Tiempo final:14.55
```

```
Client.java x Server.java x Utils.java x ClientHandler.java x Main.java x
Download grammar and spelling checker for Asturian?

9      3 usages
      static final int POPULATION_SIZE = 20;
      2 usages
10     static final int NUM_GENERACIONES = 100;
      2 usages
11     static final double LOWER_BOUND = 0;
      2 usages

Run: Server x ClientHandler x
[CLIENTE 38] 10.276359116832671/11.173829294085474/0,10.276366097406793/11.173855555469885/0,10.27
[CLIENTE 39] 10.823573698279736/11.986252786994864/0,10.823011286056423/11.986762212410801/0,10.82
[CLIENTE 40] 8.676688347201452/13.13875572788108/0,8.67668878278273/13.138756589572004/0,8.676903
[CLIENTE 41] 10.691948094568104/8.219323421534666/0,10.694938294382911/8.22071775443082/0,10.6949
[CLIENTE 42] 10.563650606076563/7.598748604784815/0,10.56197600315273/7.600404421217816/0,10.5627
[CLIENTE 43] 11.421275835132224/9.75592058446216/0,11.421298315253285/9.755874006351245/0,11.4212
[CLIENTE 44] 10.821319895005981/9.042962508379304/0,10.821372181322275/9.04290249961652/0,10.8213
[CLIENTE 45] 10.278903103463252/8.393780142543513/0,10.278877825593518/8.393798902604306/0,10.278
[CLIENTE 46] 10.781768713350758/9.680288826677337/0,10.781768699968005/9.680288800886432/0,10.7817
[CLIENTE 47] 9.004756912848645/11.5628498269952/0,8.998110096012827/11.565377179026965/0,8.995003
[CLIENTE 48] 9.908102119866617/9.400970014318768/0,9.905986923370307/9.401469542614336/0,9.912828
[CLIENTE 49] 11.122962964210043/11.031468613524776/0,11.122871418907101/11.031297823802007/0,11.12

El mejor individuo encontrado es:
x = 11.122963
y = 11.031469
Valor de la función: 245.41361419733
Tiempo final:24.229
```

```
Client.java x Server.java x Utils.java x ClientsHandler.java x Main.java x
Download grammar and spelling checker for Asturian?

9      3 usages
      static final int POPULATION_SIZE = 20;

10     2 usages
      static final int NUM_GENERACIONES = 150;

11     2 usages
      static final double LOWER_BOUND = 0;

Run: Server x ClientsHandler x
[CLIENTE 38] 8.453283793845109/13.744812476819053/0,8.453203671555377/13.744810153198717/0,8.453939150848026/10.8714723610809/9,
[CLIENTE 39] 9.067195965624737/14.613928536312981/0,9.063738317285584/14.6128586683886153/0,9.067203511459564/14.6128586683886153/0,
[CLIENTE 40] 9.469137546778183/12.53191860619635/0,9.467656396988696/12.53341854181645/0,9.468921801181287/12.53341854181645/0,
[CLIENTE 41] 9.111274637983342/10.486954574233564/0,9.118244636997235/10.486941626580777/0,9.118486818659602/10.486941626580777/0,
[CLIENTE 42] 8.52796215574669/11.8837802614948863/0,8.527773974824997/11.882604551378659/0,8.524977010430932/11.882604551378659/0,
[CLIENTE 43] 10.8714781185799/9.681504337989454/0,10.87146295699429/9.681505914915865/0,10.8714723610809/9.681505914915865/0,
[CLIENTE 44] 6.29544828502882/14.844912056494558/0,7.299075177584129/14.845646289564393/0,7.295868456579684/14.845646289564393/0,
[CLIENTE 45] 7.729488342645535/12.817671925568784/0,7.728942875274516/12.81932381090312/0,7.728948321821943/12.81932381090312/0,
[CLIENTE 46] 9.980617384078015/11.18904160327088/0,9.980637270178343/11.1890881909565/0,9.98051403479462/11.1890881909565/0,
[CLIENTE 47] 9.57877880878791/15.391202570772524/0,9.57882397028513/15.391131780020494/0,9.578746552931525/15.391131780020494/0,
[CLIENTE 48] 8.405217141390928/13.280894864749966/0,8.405219805936891/13.28085551051888/0,8.429991098952527/13.28085551051888/0,
[CLIENTE 49] 9.197418802785899/13.263986089145257/0,9.197418750859368/13.263986046627107/0,9.197418770054492/13.263986046627107/0,

El mejor individuo encontrado es:
x = 9.197419
y = 13.263986
Valor de la función: 260.52584086975696
Tiempo final:36.416
```

```
ClientJava x ServerJava x UtilJava x ClientHandlerJava x MainJava x
Download grammar and spelling checker for Asturian?

3 usages
static final int POPULATION_SIZE = 20;

2 usages
static final int NUM_GENERACIONES = 300;

2 usages
static final double LOWER_BOUND = 0;

2 usages

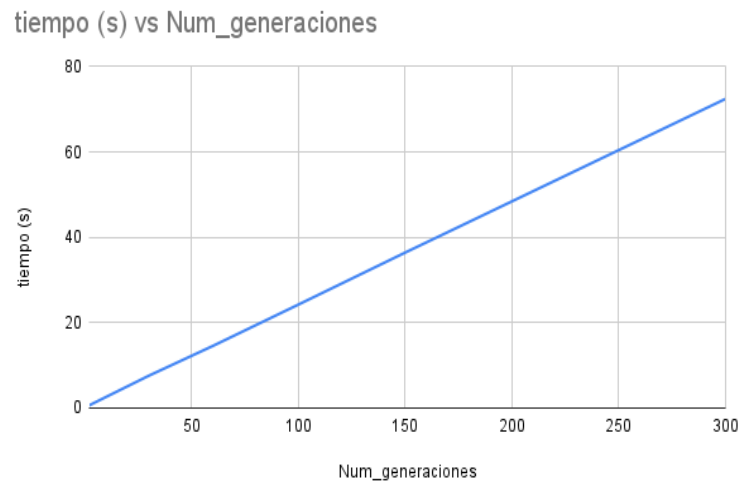
Run: Server x ClientsHandler x
[CIENTE] 38 | 6.5283208347346632/0, 6.9687185770708649/0, 6.5277265069446804/0, 6.969033730116179/0, 6.528736156424364/0,
[CIENTE] 39 | 6.955115238425488/10, 37830868564391/0, 6.951861053090044/10, 378241866683808/0, 6.95520415356362/0,
[CIENTE] 40 | 10.912915559318522/12, 203247973753953/0, 10.914164285557767/12, 204657705333612/0, 10.913598292771/0,
[CIENTE] 41 | 7.917419516677344/12, 578378991045467/0, 7.917435522112897/12, 578371199658658/0, 7.917419516677344/12,
[CIENTE] 42 | 9.7829338022696317/0, 051421226607864/0, 9.783208237881253/7, 051301503517912/0, 9.782975864489716/0,
[CIENTE] 43 | 7.383218199019375/9, 6.58085390159479/0, 7.383439686239669/9, 6.64982643025616/0, 7.3834523702715575/0,
[CIENTE] 44 | 9.926893469495186/9, 0.046394780072374/0, 9.928496174203826/9, 0.04643832338912/0, 9.926893469495186/9,
[CIENTE] 45 | 8.936423740582427/9, 2.30835383327586/0, 8.936386652443018/9, 2.30948937278573/0, 8.936417157479811/0,
[CIENTE] 46 | 8.44739089979812/12, 408489956393934/0, 8.441529054192216/12, 408450174081992/0, 8.448752241295349/0,
[CIENTE] 47 | 7.733975241162694/10, 383401335978288/0, 7.73394663187892/10, 38344692746754/0, 7.733952354646985/0,
[CIENTE] 48 | 8.96990368182688/9, 0.237410935752615/0, 8.9699887867173/9, 0.237138987717223/0, 8.978447578580804/9,
[CIENTE] 49 | 9.088944187299462/7, 0.504376228942295/0, 9.092345532772153/9, 0.503331166669042/9, 9.086691825903438/0,

El mejor individuo encontrado es:
x = 9.088944
y = 9.504376
Valor de la función: 172.942066184512
Tiempo final:72.444
```

- Tabla que recolecta NumGeneraciones y su tiempo

Num_generaciones	tiempo (s)
2	0.597
10	2.591
30	7.563
60	14.55
100	24.229
150	36.416
300	72.444

- Grafica obtenida



REFERENCES

[1] Códigos y teoría de las clases. Nuñez, Y. [2] El código se puede encontrar en [Enlace a GitHub](#)