

Técnicas, Entornos y Aplicaciones de Inteligencia Artificial

Práctica 4. ALGORITMOS GENÉTICOS

Objetivo:

utilizar Opt4J para diseñar, resolver y evaluar un problema de optimización mediante AG

Opt4J está disponible en:

Poliformat y en <http://opt4j.sourceforge.net/>

Opt4J. Entorno libre



A Modular Framework for Meta-heuristic Optimization

Disponible en: <http://opt4j.sourceforge.net/>

Formulación sencilla de problemas utilizando librerías implementadas en Java

Existe un boletín completo que explica su instalación y uso

Algoritmos Genéticos

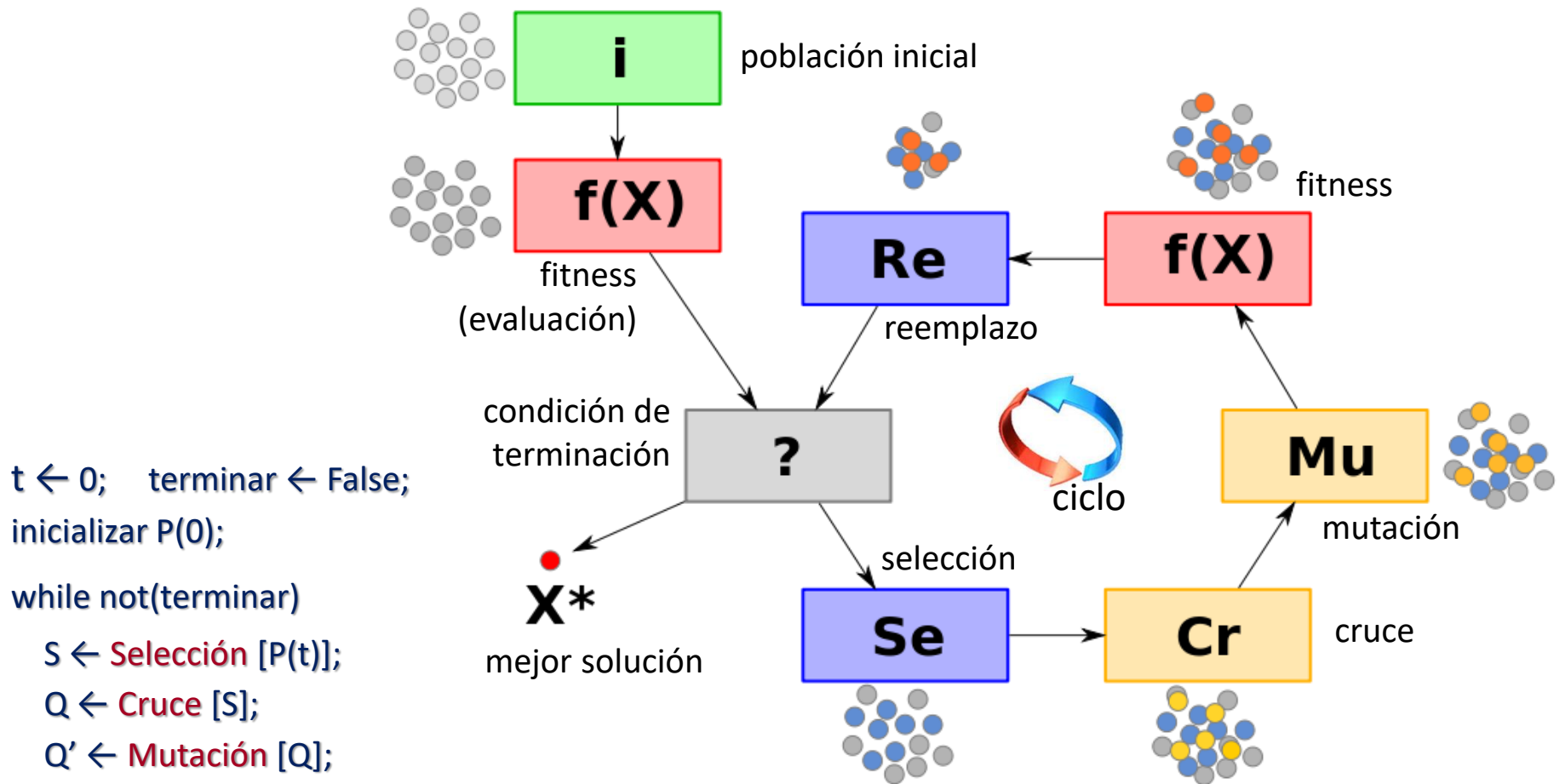
Diseño algoritmo genético

- Diseño del individuo. Codificación y decodificación.
- Función de evaluación (fitness)
- Generación población inicial.
- Selección. Cruce (individuos inválidos). Mutación. Reemplazo.

Evaluación algoritmo genético

- Criterios de evaluación: Fitness versus Soluciones generadas, Tiempo cómputo.
- Tamaños del problema
- Parámetros de evaluación: Población, Selección, Cruce, Mutación, etc.

Práctica 4: Opt4J



https://commons.wikimedia.org/wiki/File:Evolutionary_algorithm.svg

```
t ← 0;  terminar ← False;  
inicializar P(0);
```

```
while not(terminar)
```

```
    S ← Selección [P(t)];
```

```
    Q ← Cruce [S];
```

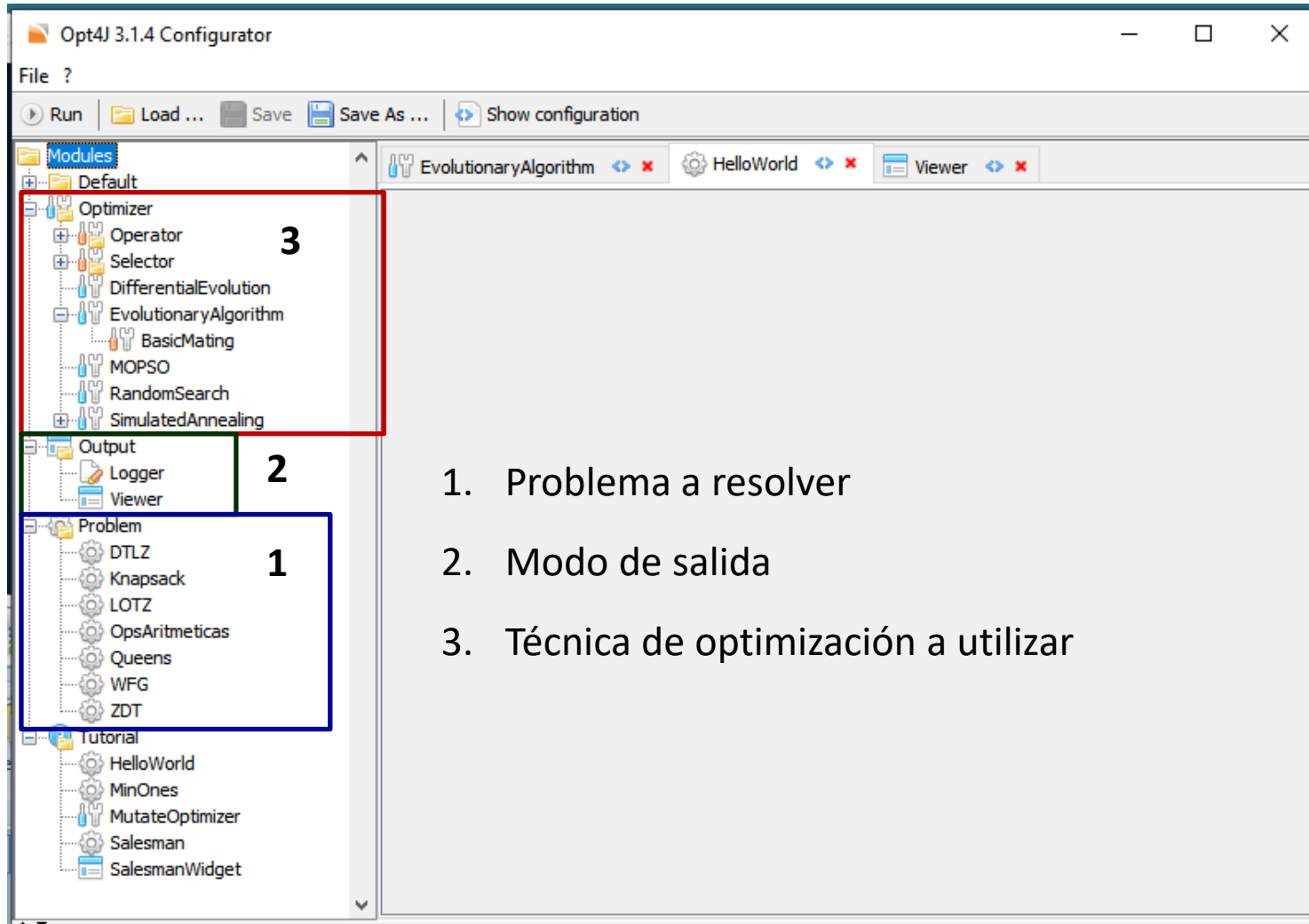
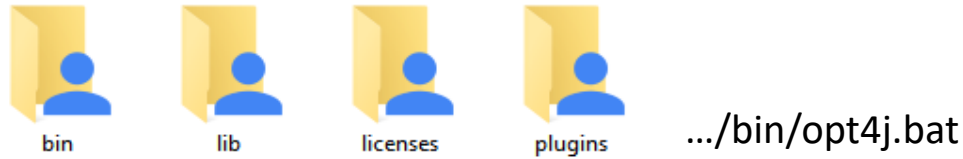
```
    Q' ← Mutación [Q];
```

```
    P(t+1) ← Reemplazo [P(t), S, Q'];
```

```
    t ← t+1
```

```
    terminar ← (Convergencia [P(t+1)]) OR (t>límite)
```

```
end_while
```



1. Problema a resolver
2. Modo de salida
3. Técnica de optimización a utilizar

Práctica 4: Opt4J

Parametrización del AG

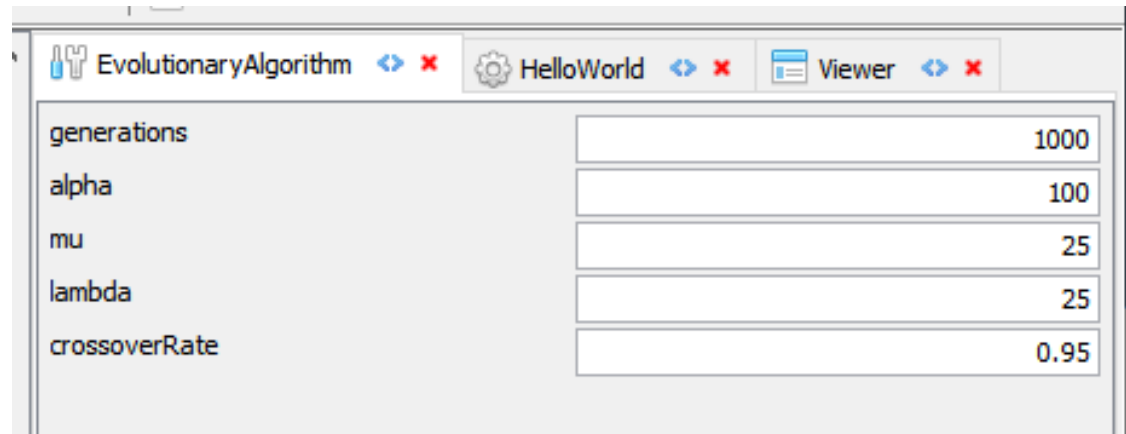
Nº Generaciones

Nº Población Inicial

Nº padres seleccionados / Iteración

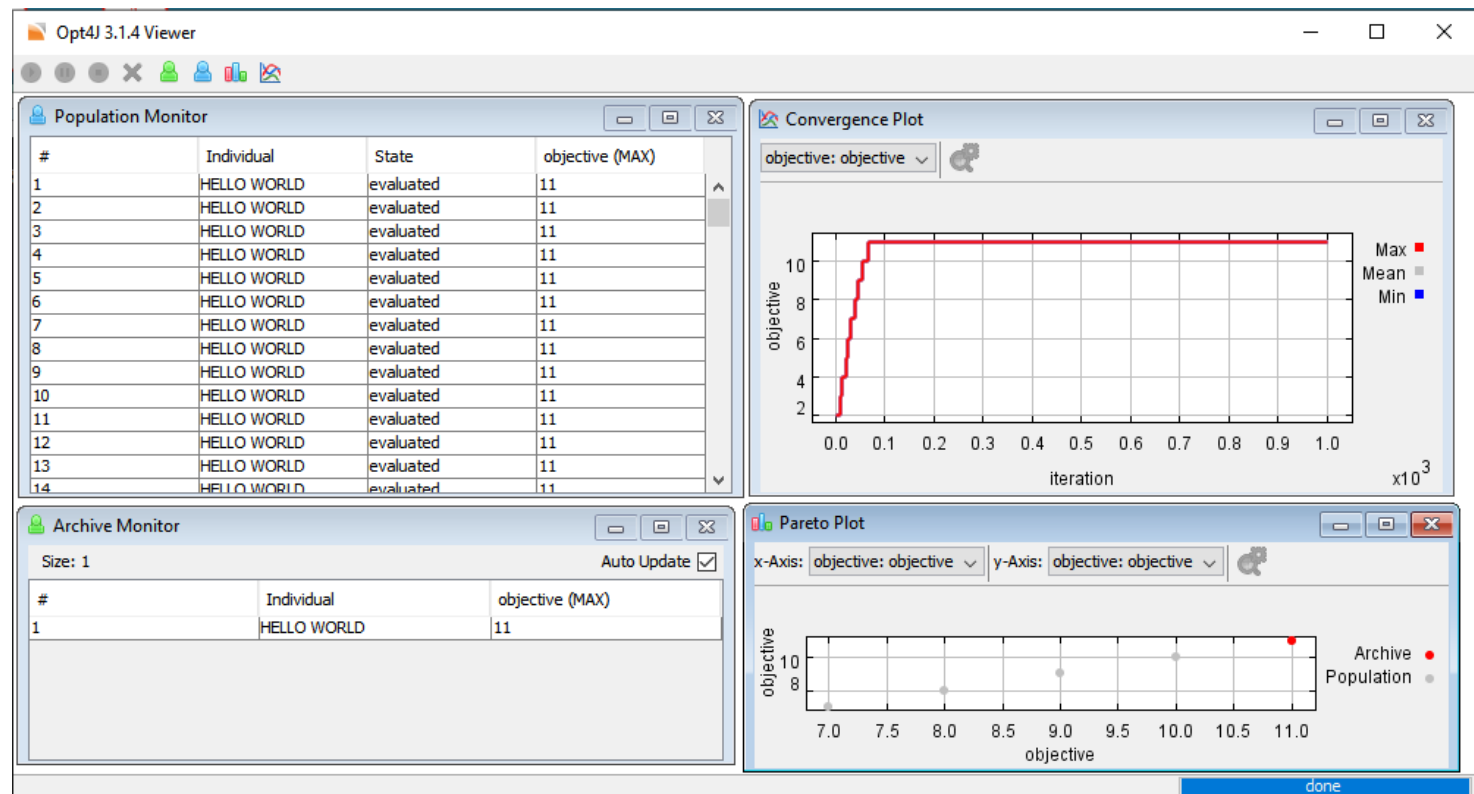
Nº Hijos generados / Iteración

Probabilidad cruce de dos padres



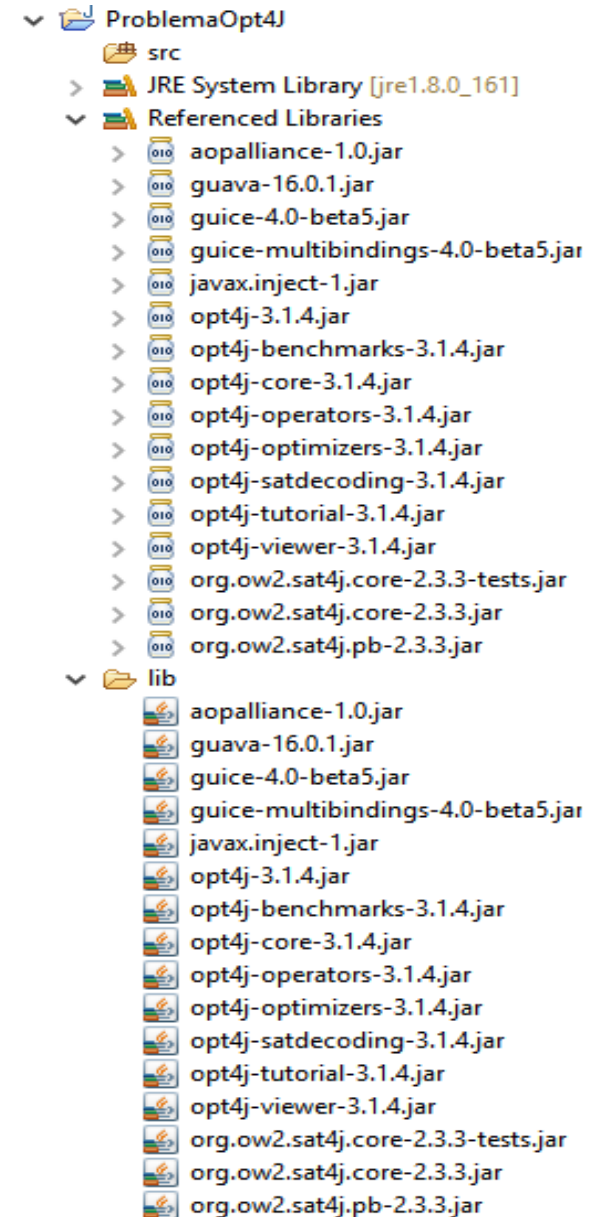
Parameter	Value
generations	1000
alpha	100
mu	25
lambda	25
crossoverRate	0.95

Resultados:

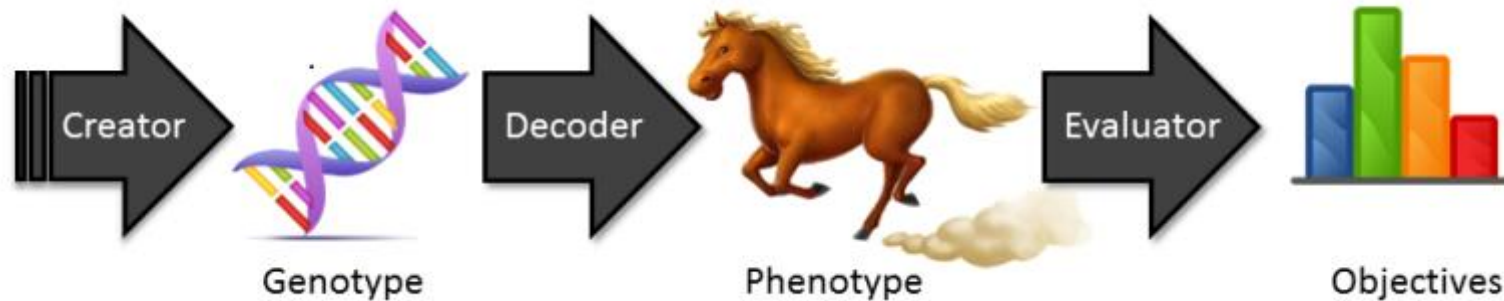


Práctica 4: Integración Opt4j en ECLIPSE

- Opt4J permite la importación y resolución de problemas previamente modelados en Java.
- Por simplicidad, utilizaremos el **entorno Eclipse**.
- **Configuración de ECLIPSE en Boletín (1.2, 1.3)**
- Modelado del problema en Java (Creator, Decoder y Evaluator)



Básicamente, hay que implementar tres clases



```
public class NombreClaseCreator implements Creator<GENOTIPO>
```

```
public class NombreClaseDecoder implements Decoder<GENOTIPO, FENOTIPO>
```

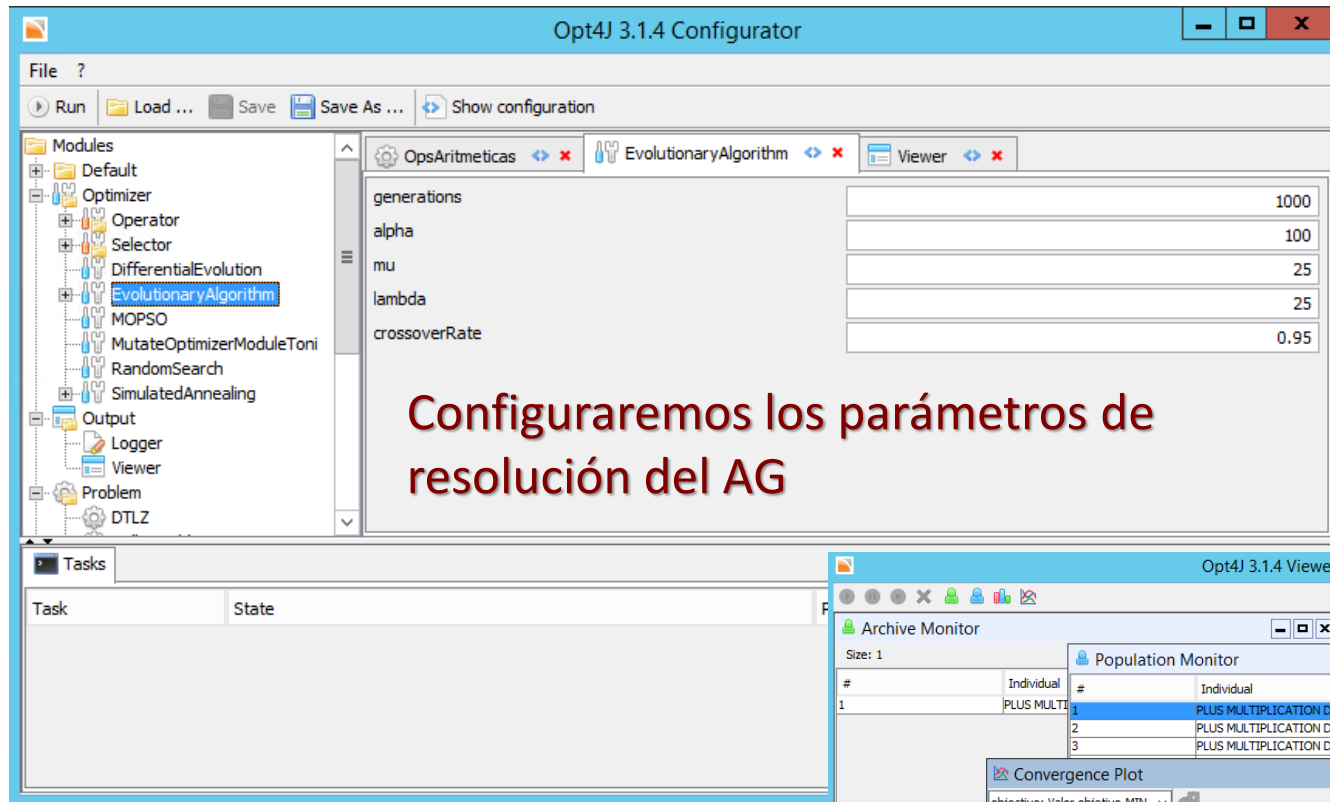
```
public class NombreClaseEvaluator implements Evaluator<FENOTIPO>
```

...más la clase Module que las referencia:

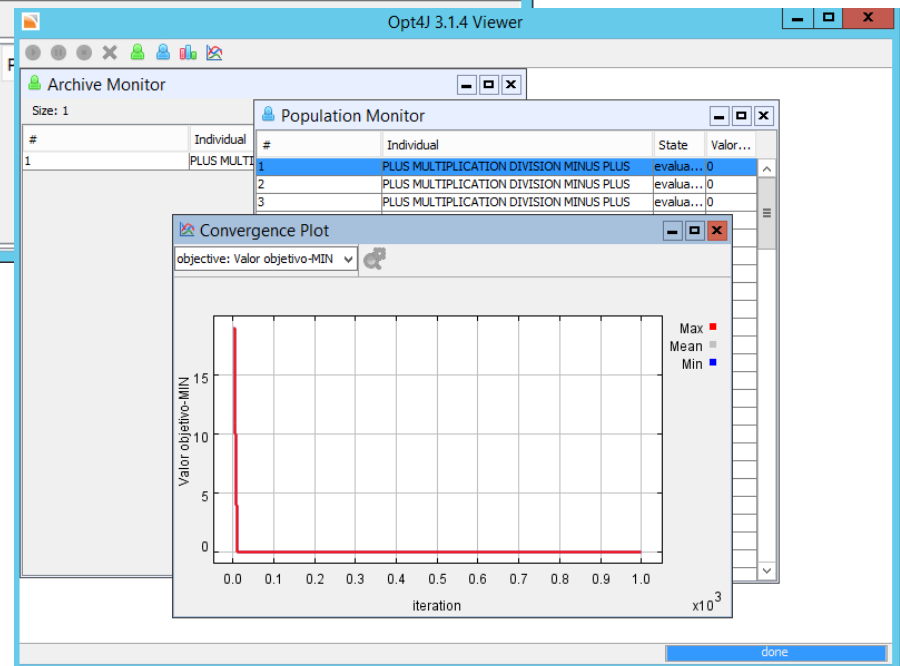
```
public class ClaseModule extends ProblemModule
```

Ver ejemplos en el Boletín!!

Práctica 4: resolviendo el problema en Opt4J



Y analizaremos los resultados



Evaluación:

- Realizar el ejercicio propuesto (se necesitará para el día de la evaluación, en el que se planteará una breve ampliación)

Calendario:

Sem	<u>LABORATORIO</u>	Evaluación
15-XII	Opt4J	
22-XII	Opt4J	
12-I		P4: <i>Aplicac. Opt4J</i>

Aplicación y evaluación de Algoritmos Genéticos (15%) P4