
PROGRAMACIÓN EVOLUTIVA

MEMORIA DE LA PRÁCTICA 1

ALEJANDRO BARRACHINA ARGUDO
ADRIÀ CARRERAS BAGUR

*Grado en Ingeniería informática
Facultad de Informática
Universidad Complutense de Madrid*

Índice general

1. Capturas de funcionamiento	5
2. Observaciones de la práctica, arquitectura del código y ejecución	11
2.1. Observaciones	11
2.2. Arquitectura	11
2.3. Ejecución y compilación de la práctica	11
3. Distribución de trabajo y conclusiones	13
3.1. Distribución	13
3.2. Conclusiones	13

1 | Capturas de funcionamiento

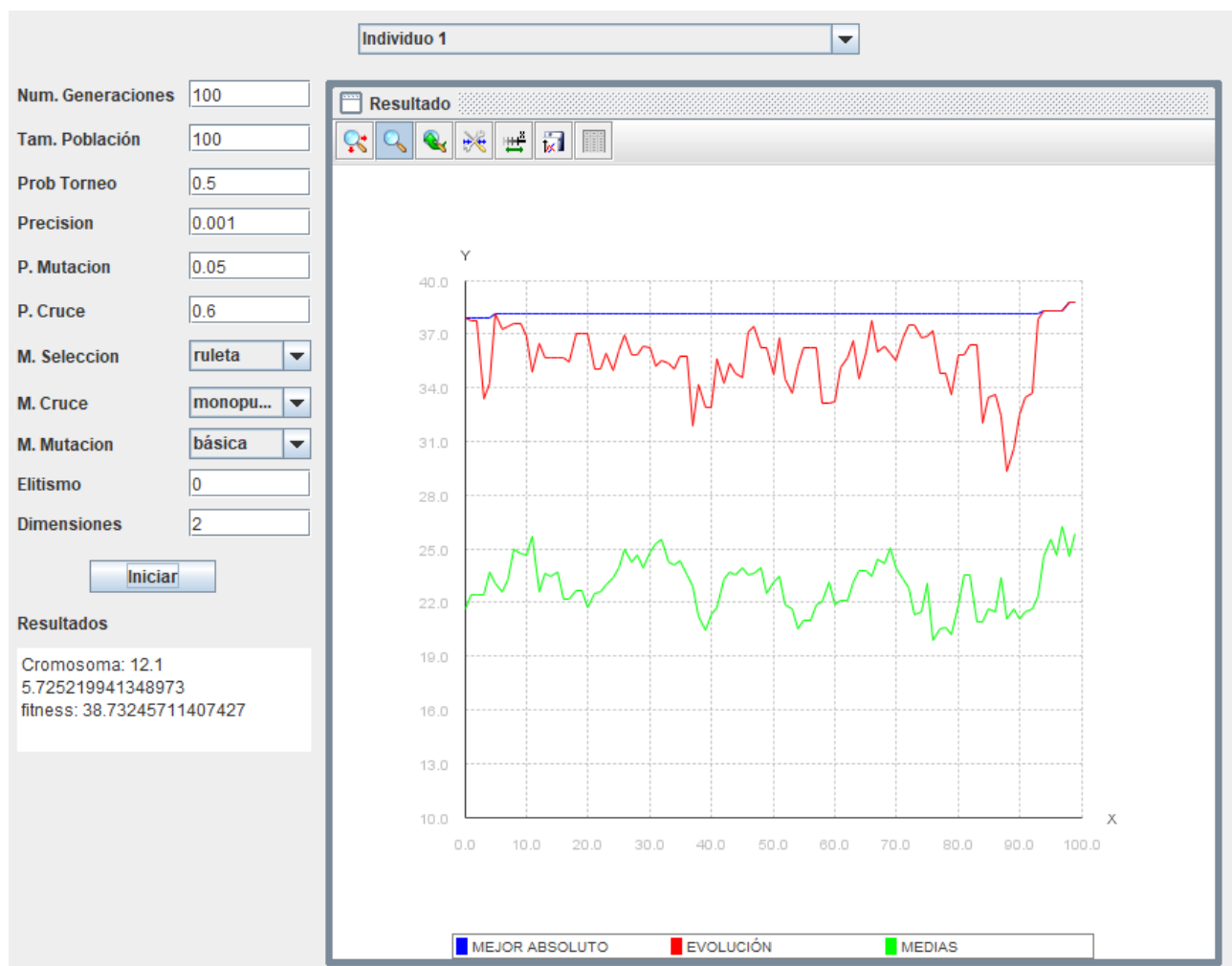


Figura 1.0.1: Individuo 1

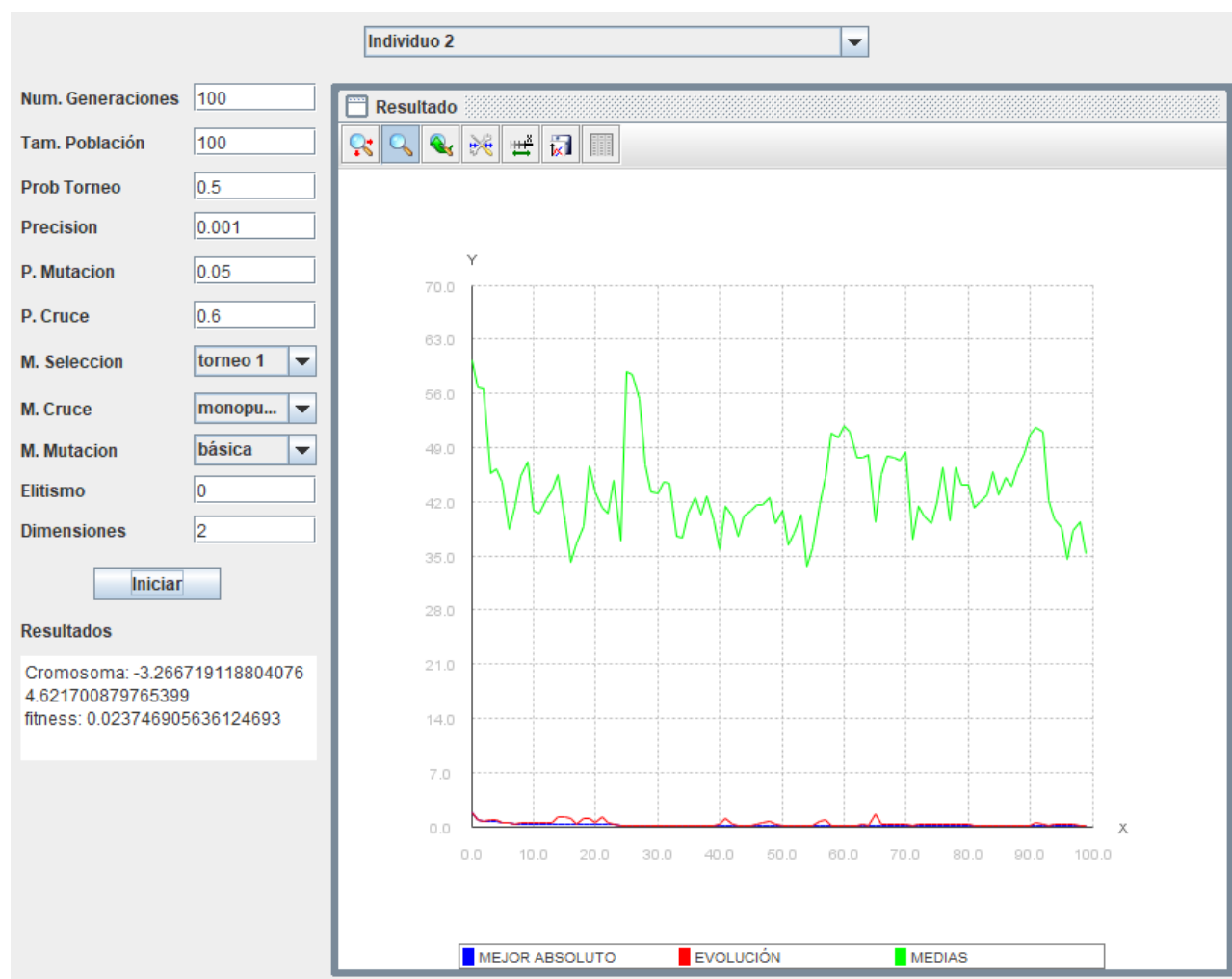


Figura 1.0.2: Individuo 2

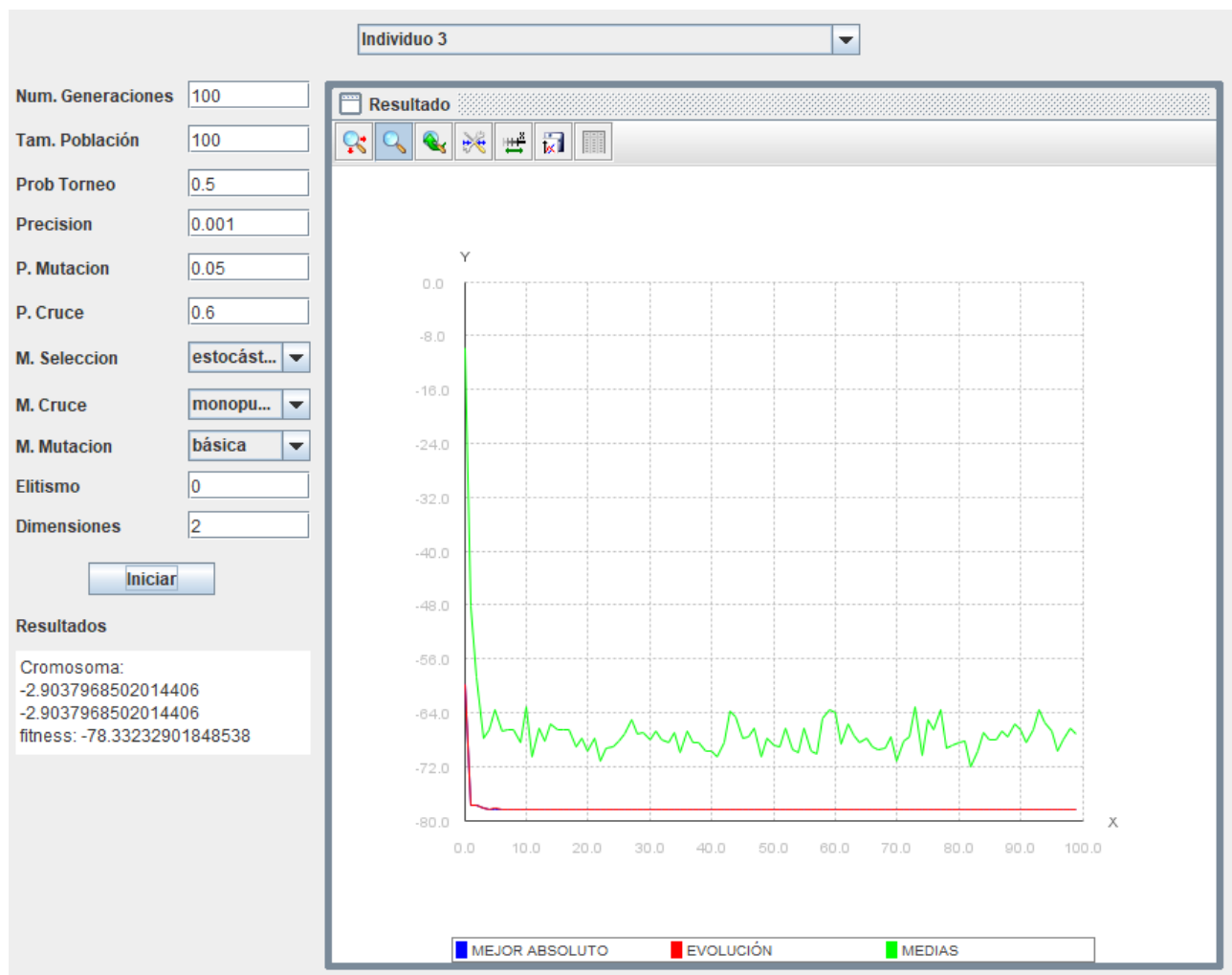


Figura 1.0.3: Individuo 3

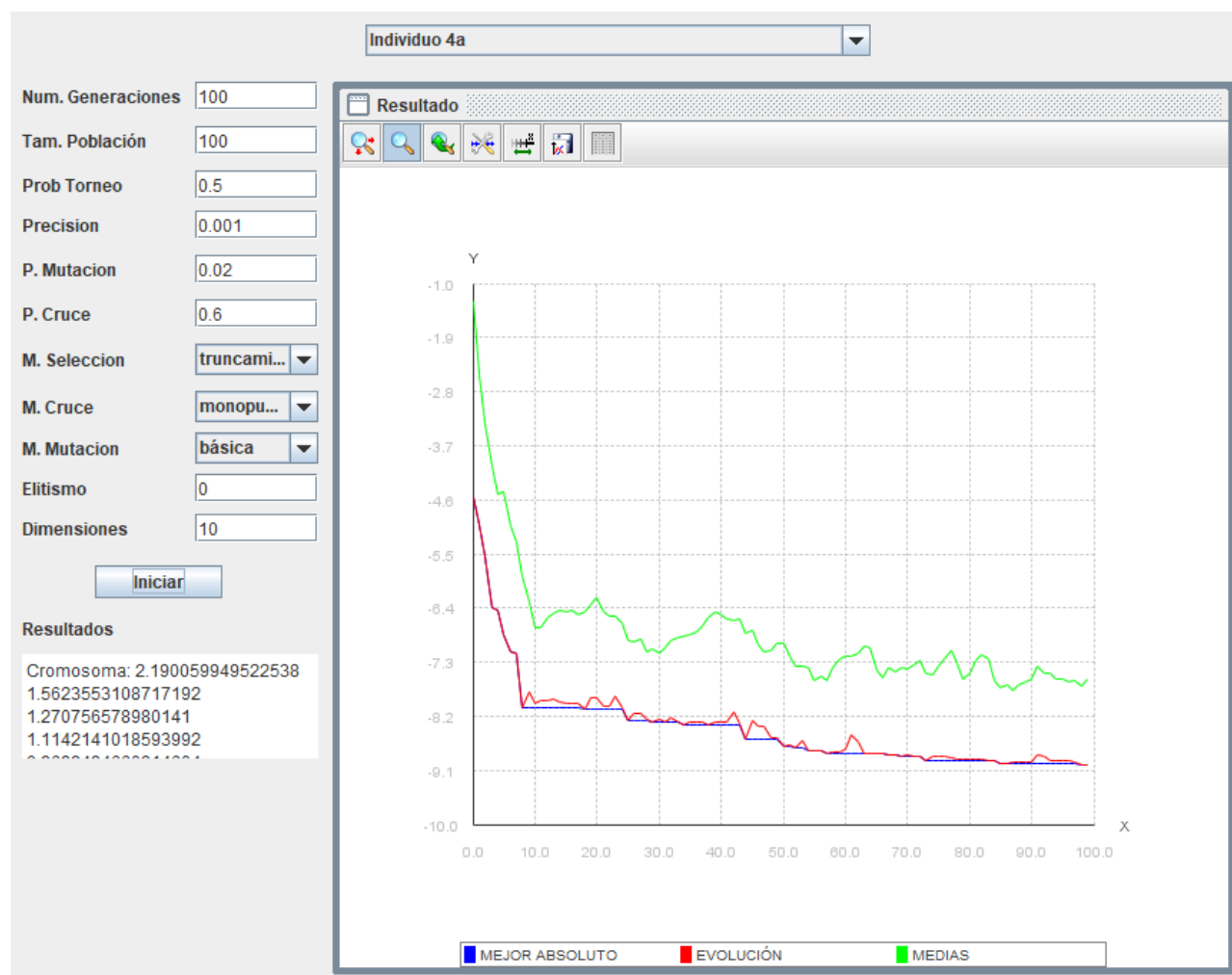


Figura 1.0.4: Individuo 4A

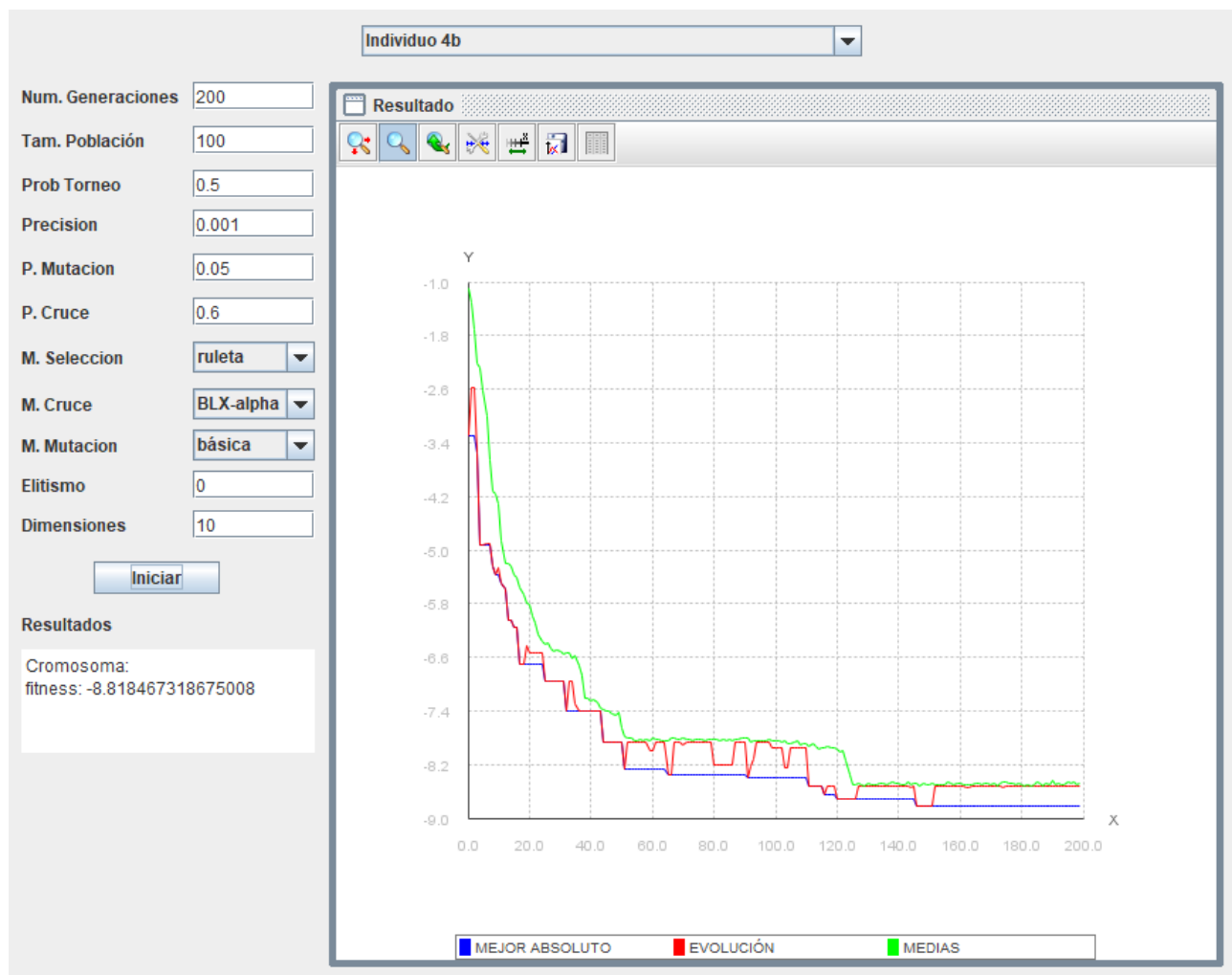


Figura 1.0.5: Individuo 4b

2 | Observaciones de la práctica, arquitectura del código y ejecución

2.1. Observaciones

El método de selección que mejores resultados nos ha dado ha sido la selección por truncamiento, ya que al escoger solo los mejores individuos, siempre permite evolucionar mejor a la población.

El elitismo hace que la convergencia de la curva sea total entre el mejor absoluto y la evolución. Con valores altos de mutación ($> 0,9$) da valores inconsistentes.

Podemos observar en la figura 1.0.4 que el método de truncamiento para la selección provoca una evolución consistente y bastante convergente con el mejor absoluto incluso sin elitismo.

La primera función, en la figura 1.0.1, nos sirve para ver como la mutación alta nos provoca unos cambios muy drásticos en la media de cada generación y cambios muy dispares en la línea de evolución.

Tanto en la figuras 1.0.1 y 1.0.5 podemos ver que, aunque hace su trabajo, la selección por ruleta es la menos eficiente de las que tenemos en cuanto a evolución del individuo.

2.2. Arquitectura

Para mantener todo el código organizado, se usa la siguiente estructura de paquetes:

- **go2:** paquete que engloba toda la práctica.
- **g02.cruces:** Paquete que reúne los cuatro cruces distintos bajo la interfaz Cruces.
- **g02.individual:** Paquete que contiene todos los individuos de las distintas funciones bajo la clase padre Individuo<T>.
- **g02.Selections:** Paquete para reunir las distintas selecciones usadas en la práctica bajo la clase padre Selection<T>, donde se encuentra las funciones para corregir el fitness.
- **g02.Ventana:** Paquete para la interfaz gráfica de la práctica y el método main.

Para una visión mas completa del proyecto, desde el navegador puede abrir el archivo `G02/target/site/index.html`

2.3. Ejecución y compilación de la práctica

Al ser un proyecto en maven se puede importar directamente desde eclipse con la opción de abrir proyectos del sistema de archivos.

Para compilar el proyecto solo hay que darle al botón de *run* y debería instalar las dependencias necesarias. Si no fuera así se puede ejecutar el comando `mvn install` para instalar todas las dependencias.

Así mismo, el proyecto se puede correr desde el propio eclipse usando el botón de *run*, usando la configuración de *aplicación java* y como clase principal `g02.Ventana.ventana`.

El archivo jar se encuentra en la carpeta `P01/target/`

3 | Distribución de trabajo y conclusiones

3.1. Distribución

Adrià ha hecho toda la GUI, parte de los cruces y parte de los individuos. Alejandro ha hecho la selección, parte de los cruces y parte de los individuos. La clase AlgoritmoGenetico se ha hecho entre los dos integrantes.

3.2. Conclusiones

En conclusión, el elitismo ayuda a un progreso más rápido de la evolución de una población siempre y cuando se use de forma moderada, ya que si no puede producir estancamientos. Los métodos de selección que mejor nos han funcionado han sido: torneos deterministas, truncamientos y estocástico. La mutación usada en valores exagerados nos ha causado problemas.