

**J. Ángel Velázquez Iturbide**

# AlgorEx: Manual de Usuario

Número 2020-05

**Serie de Informes Técnicos DLSI1-URJC**

**ISSN 1988-8074**

**Departamento de Lenguajes y Sistemas Informáticos I  
Universidad Rey Juan Carlos**



## Índice

|     |                                       |    |
|-----|---------------------------------------|----|
| 1   | Introducción .....                    | 1  |
| 2   | Visión General de la Aplicación ..... | 2  |
| 1.1 | Una Sesión de Usuario .....           | 2  |
| 1.2 | Instalación .....                     | 9  |
| 3   | Interfaz de Usuario.....              | 9  |
| 3.1 | Tabla de Datos.....                   | 10 |
| 3.2 | Tabla de Ejecuciones.....             | 11 |
| 3.3 | Tabla de Resumen Numérico .....       | 15 |
| 3.4 | Pestaña de Resumen Gráfico .....      | 17 |
| 4   | Funciones de AlgorEx.....             | 20 |
| 4.1 | Menú Archivo .....                    | 20 |
| 4.2 | Menú Experimentación .....            | 21 |
| 4.3 | Menú Tablas.....                      | 28 |
| 4.4 | Menú Configuración .....              | 28 |
| 4.5 | Menú Ayuda.....                       | 29 |
|     | Referencias.....                      | 30 |

# AlgorEx: Manual de Usuario

J. Ángel Velázquez Iturbide <sup>1</sup>

<sup>1</sup> Departamento de Informática y Estadística, Universidad Rey Juan Carlos,  
Móstoles, Madrid, España  
angel.velazquez@urjc.es

**Resumen.** AlgorEx es un sistema para la experimentación educativa con algoritmos, sobre todo con algoritmos de optimización. El sistema permite cargar una clase Java que contenga distintos algoritmos que resuelvan el mismo problema y ejecutarlos sobre muchos casos de prueba. En caso de seleccionar el criterio de optimalidad o de eficiencia en tiempo, los resultados o las medidas de tiempos tomadas para los distintos algoritmos se comparen entre sí. AlgorEx da soporte para las tareas auxiliares necesarias para un experimento: fijación de las condiciones de comparación, generación de los casos de prueba, almacenamiento y comparación de los resultados, y análisis estadístico.

**Palabras clave:** problemas, algoritmos, experimentación, optimalidad, tiempos de ejecución.

## 1 Introducción

OptimEx es un sistema para la experimentación con algoritmos de optimización con fines educativos [1][2]. El sistema permite cargar una clase Java que contenga distintos algoritmos que resuelven el mismo problema de optimización y comparar los resultados que los algoritmos calculan sobre muchos casos de prueba. OptimEx da soporte para las tareas auxiliares necesarias para un experimento: fijación de las condiciones de comparación, generación de los casos de prueba, almacenamiento y comparación de los resultados, y análisis estadístico. También incluye algunas facilidades para su uso educativo, como la exportación de tablas y diagramas a fichero o el cambio de idioma. La versión 1.4 de OptimEx también permite medir tiempos de ejecución siguiendo el mismo esquema experimental que para la optimalidad [3][4]. AlgorEx es la siguiente versión de OptimEx que, incluye un nuevo criterio de experimentación (el “criterio nulo”), con lo cual cambia de nombre porque el papel destacado del criterio de optimalidad se reduce aún más.

El informe s consta de tres apartados. El apartado 2 realiza una breve presentación de AlgorEx, mostrando una supuesta y breve sesión de usuario, y explicando su descarga e instalación. El apartado 3 presenta los distintos elementos de su interfaz de usuario, sobre todo sus tablas y diagramas. Por último, el apartado 4 presenta las funciones de AlgorEx con todo detalle. Algunas figuras se presentan varias veces, aunque con la misma numeración, para facilitar la lectura del manual.

## 2 Visión General de la Aplicación

En este apartado realizamos una introducción a AlgorEx. En primer lugar, presentamos una breve sesión de usuario. Se presenta el modo de trabajo con AlgorEx, sin entrar en detalle apenas en los elementos de la aplicación, que pueden encontrarse en los apartados 3 y 4. Después, se explica la descarga e instalación de AlgorEx.

### 1.1 Una Sesión de Usuario

Al abrir AlgorEx, se contempla una interfaz de usuario “vacía”, como en la Fig. 1.

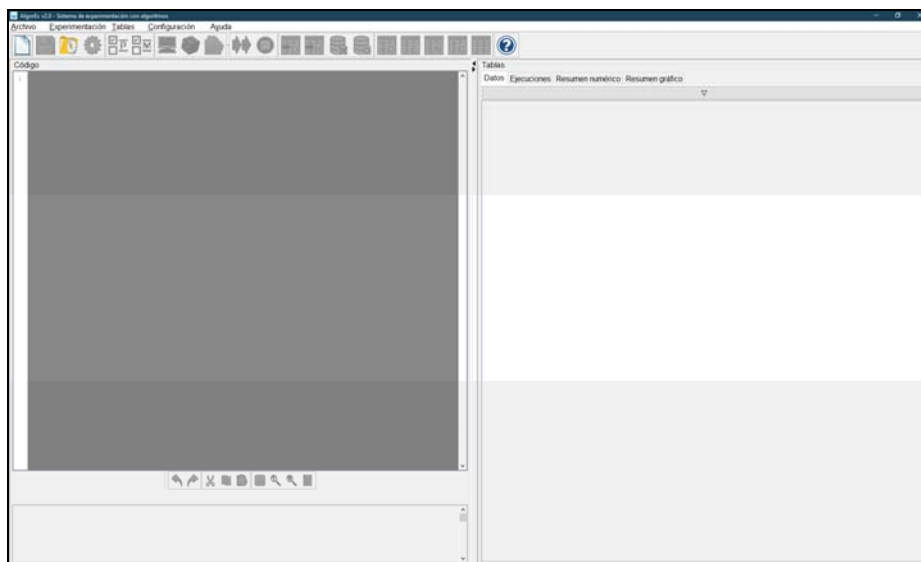


Fig. 1. Interfaz de usuario de AlgorEx al inicio de sesión.

Puede observarse que la ventana consta de un menú principal y una barra de iconos en la parte superior y de dos paneles verticales en el área central. El panel izquierdo alberga un editor de texto y una pequeña área de mensajes debajo. El panel derecho alberga cuatro pestañas para tablas y diagramas.

Veamos una sesión ideal. El primer paso consiste en cargar una clase Java con los algoritmos a comparar (opción “Cargar y compilar clase...”). Supongamos que cargamos una clase llamada `LlenadoCajas.java`, que contenga varios algoritmos para las dos versiones (maximización y minimización) del problema de llenado de cajas [5, págs. 534-536]. Con esta única operación, se compila y extrae información sobre los métodos públicos de la clase, que se usará posteriormente, y se carga la clase en el editor. Si la clase produjera errores de compilación, se mostrarían uno o más mensajes explicativos en la zona de mensajes situada bajo el editor.

En un segundo paso se deben indicar las condiciones en las cuales se va a realizar el experimento. Se selecciona la función “Seleccionar problema...”, que presenta varias opciones, como muestra la Fig. 2.

El usuario debe seleccionar, de arriba abajo: (a) la signatura del problema que resuelven los métodos a ejecutar, (b) el criterio a usar para seleccionar y comparar información resultante de ejecutar los métodos, y en caso de comparar su optimalidad, (c) si es un problema de maximización o de minimización, y (d) si los algoritmos se van a comparar usando medidas de desviación absolutas o relativas. En la Fig. 2, puede verse que se ha elegido la signatura de la versión de minimización del problema, y que la clase contiene tres algoritmos con dicha signatura. También puede verse que se va a comparar la optimalidad de los algoritmos con medidas relativas.



Fig. 2. Diálogo de selección del problema.

Entonces, el usuario debe crear los juegos de prueba. Lo más frecuente es generarlos aleatoriamente. Si seleccionamos la opción “Generar datos aleatorios...”, primero se abre un diálogo pidiéndonos el número de juegos de prueba a generar; supongamos que dejamos el valor por defecto, 100. También permite cargar automáticamente de ficheros: juegos de datos preparados y los rangos de valores a usar en la generación de datos. Véase el diálogo en la Fig. 3.

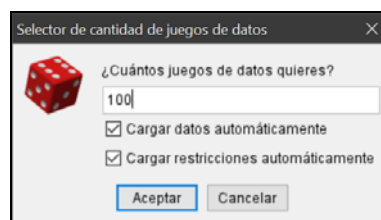


Fig. 3. Primer diálogo de la generación aleatoria de datos.

A continuación, se abre un diálogo particularizado para el problema seleccionado en el paso anterior. En nuestro ejemplo, la versión de minimización del problema de

llenado de cajas tiene dos parámetros (véase la Fig. 4): los pesos de los objetos a meter en cajas (un vector) y la capacidad de cada caja (un valor entero). Los rangos de valores mostrados son los cargados automáticamente tras el diálogo anterior.

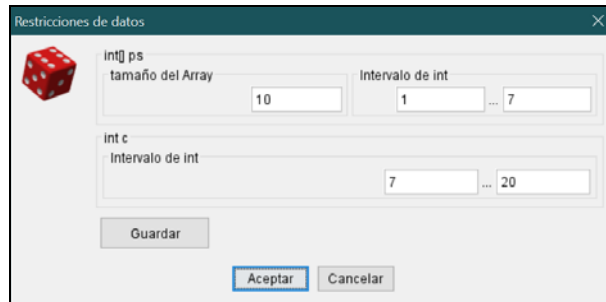


Fig. 4. Segundo diálogo de la generación aleatoria de datos.

Los juegos de prueba se cargan en la tabla de datos, como puede verse en la Fig. 5. La columna primera numera cada juego de datos, mientras que cada una de las demás columnas está asociada a un parámetro. Obsérvese que la tabla de datos de la figura contiene una primera fila con su celda izquierda en verde y otras filas, en azul. La primera fila corresponde a un juego de datos cargado automáticamente por activar la opción correspondiente en el primer diálogo de “Generar datos aleatorios...”, incluyendo tras su número el resultado óptimo del problema para estos datos de entrada. Las restantes 100 filas se han generado automáticamente y solamente contienen datos de entrada.

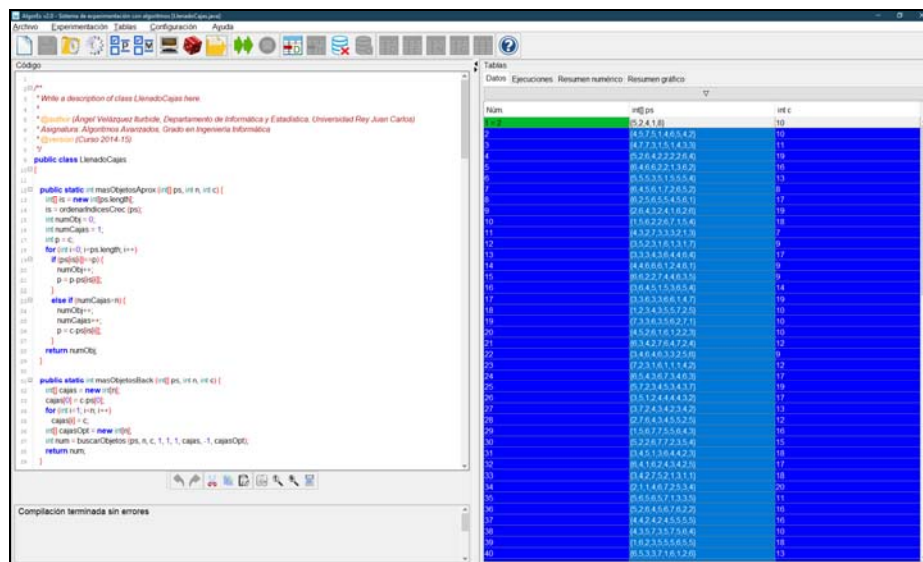


Fig. 5. Datos cargados y generados, almacenados en la tabla de datos.

El siguiente paso es seleccionar los juegos de datos a ejecutar. Tras una operación de generación aleatoria, quedan seleccionadas (en color azul) todas las filas correspondientes a juegos de datos cargados y generados para que los algoritmos puedan ejecutarlos inmediatamente, sin necesidad de seleccionar sus filas. En la Fig. 5, no aparece seleccionada la primera fila para que se aprecie el distinto formato de los juegos de datos cargados y los generados.

A continuación, deben ejecutarse los algoritmos usando la opción “Ejecución”. La Fig. 6 muestra el contenido de la tabla de ejecuciones tras ejecutar los 100 juegos de datos generados anteriormente con los tres algoritmos de minimización. Cada fila contiene los valores calculados por los algoritmos para el juego de prueba correspondiente en la tabla de datos. Cada columna contiene los valores calculados por un algoritmo. Las celdas coloreadas de verde oliva son las que tienen un valor óptimo dentro de cada fila, estando coloreadas de verde claro las celdas con valores subóptimas. Si un algoritmo tiene valores óptimos en todas las filas, los valores de su columna se colorean con fuente azulada, como es el caso del tercer algoritmo en la Fig. 6.

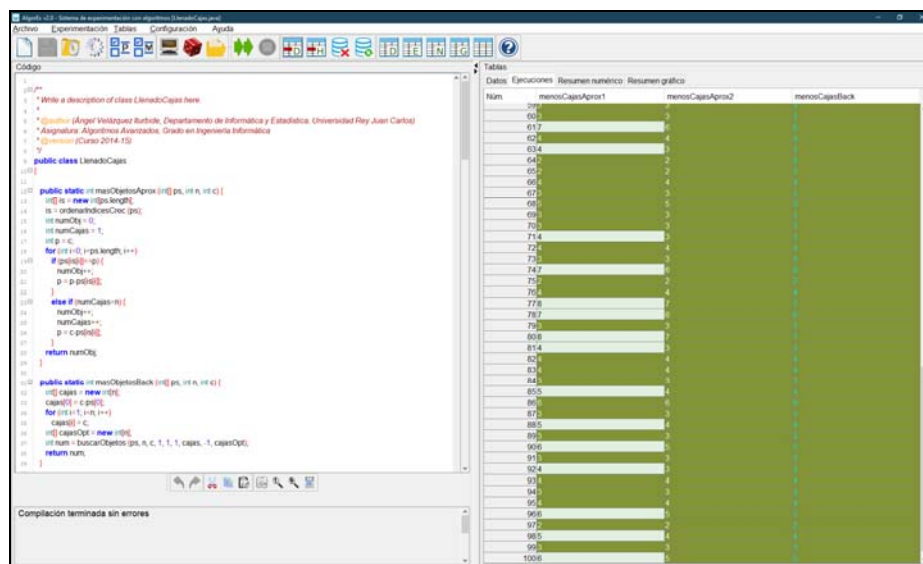


Fig. 6. Tabla de ejecuciones bajo el criterio de optimalidad, con los resultados calculados por los tres algoritmos.

Se tiene un resumen de los resultados del experimento en las dos pestañas restantes del panel derecho. El resumen numérico se muestra en la Fig. 7 y el gráfico, en la Fig. 8.



| Medidas                         | menosCajasAprox1 | menosCajasAprox2 | menosCajasBack |
|---------------------------------|------------------|------------------|----------------|
| Núm. ejecuciones                | 100              | 100              | 100            |
| Núm. ejec. válidas del método   | 100              | 100              | 100            |
| Núm. ejec. válidas en total     | 100              | 100              | 100            |
| % Soluciones subóptimas         | 42,00 %          | 1,00 %           | 0,00 %         |
| % Soluciones óptimas            | 58,00 %          | 99,00 %          | 100,00 %       |
| % Soluciones sobreóptimas       | 0,00 %           | 0,00 %           | 0,00 %         |
| % Diferencia media subóptima    | 125,25 %         | 133,33 %         | 0,00 %         |
| % Diferencia máxima subóptima   | 150,00 %         | 133,33 %         | 0,00 %         |
| % Diferencia media sobreóptima  | 0,00 %           | Ejec. nº 31, 55  | 0,00 %         |
| % Diferencia máxima sobreóptima | 0,00 %           | 0,00 %           | 0,00 %         |

Fig. 7. Tabla de resumen numérico del experimento con el criterio de optimalidad.

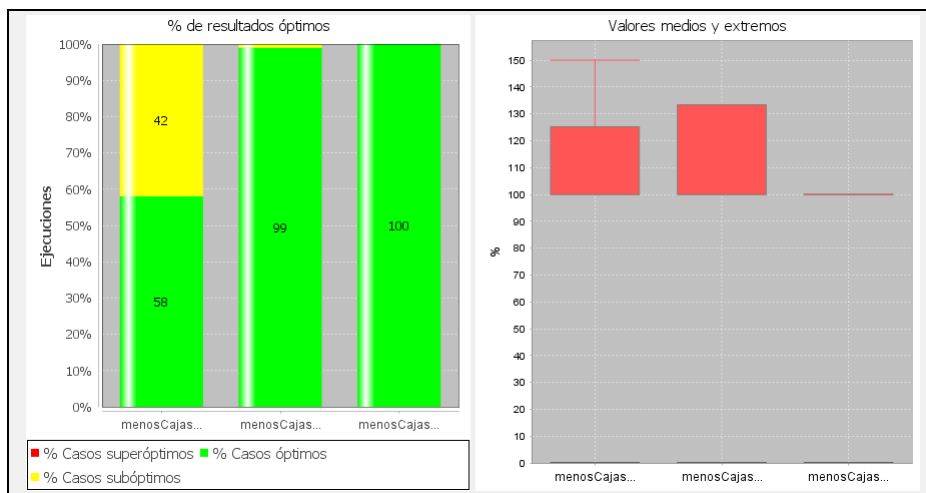


Fig. 8. Diagramas de resumen gráfico del experimento con el criterio de optimalidad.

Una vez obtenidos los resultados del experimento, los valores de todas las pestañas pueden exportarse a ficheros de distintos formatos (gráficos, Excel, texto, etc.).

En nuestra sesión, vamos a cambiar el criterio de la experimentación, de la optimalidad a la eficiencia en tiempo. Para ello, volvemos a utilizar la opción “Seleccionar problema...” para cambiar el criterio de la experimentación a eficiencia. Véase en la Fig. 9 el diálogo con la nueva opción seleccionada. (Obsérvese que existe un tercer criterio, “nulo”, que se explica con la función “Seleccionar problema...”, subapartado 4.2.)



Fig. 9. Diálogo de selección de problema, con el criterio modificado.

Como resultado, se borran los resultados calculados y almacenados en la tabla de ejecuciones, así como los resúmenes. Debemos ejecutar de nuevo los métodos sobre los juegos de prueba. La Fig. 10 muestra los nuevos contenidos de la tabla de resultados, donde aparecen tiempos de ejecución. Todas las celdas tienen el mismo color verde claro porque no se compara su contenido.

| Datos            | Ejecuciones      | Resumen numerico | Resumen grafico |
|------------------|------------------|------------------|-----------------|
| menosCajasAprox1 | menosCajasAprox2 | menosCajasBack   |                 |
| Núm.             | 396,493          | 3,934            | 3,263           |
|                  | 602,082          | 0,002            | 10,888          |
|                  | 0,0,641          | 0,003            | 0,485           |
|                  | 423,31           | 0,002            | 8,301           |
|                  | 033,261          | 0,002            | 8,291           |
|                  | 641,666          | 0,002            | 8,773           |
|                  | 651,882          | 0,002            | 10,357          |
|                  | 662,54           | 0,002            | 5,182           |
|                  | 671,232          | 0,002            | 9,264           |
|                  | 682,956          | 0,003            | 2,121           |
|                  | 692,176          | 0,002            | 7,076           |
|                  | 701,184          | 0,002            | 7,007           |
|                  | 712,707          | 0,002            | 4,067           |
|                  | 722,466          | 0,002            | 8,434           |
|                  | 732,389          | 0,002            | 7,613           |
|                  | 746,437          | 0,003            | 0,233           |
|                  | 751,139          | 0,003            | 8,567           |
|                  | 762,239          | 0,002            | 3,349           |
|                  | 774,696          | 0,002            | 0,031           |
|                  | 783,787          | 0,003            | 0,564           |
|                  | 792,486          | 0,002            | 9,133           |
|                  | 803,228          | 0,002            | 0,262           |
|                  | 812,783          | 0,002            | 9,219           |
|                  | 822,701          | 0,002            | 4,901           |
|                  | 832,481          | 0,002            | 4,575           |
|                  | 842,044          | 0,003            | 6,735           |
|                  | 853,059          | 0,002            | 4,074           |
|                  | 863,286          | 0,002            | 0,424           |
|                  | 871,817          | 0,003            | 8,781           |
|                  | 883,474          | 0,003            | 3,622           |
|                  | 892,615          | 0,002            | 9,325           |
|                  | 902,66           | 0,002            | 0,451           |
|                  | 911,906          | 0,003            | 7,615           |
|                  | 923,085          | 0,002            | 5,848           |
|                  | 932,507          | 0,002            | 4,063           |
|                  | 942,842          | 0,003            | 9,175           |
|                  | 953,305          | 0,002            | 5,567           |
|                  | 963,168          | 0,002            | 1,449           |
|                  | 971,792          | 0,002            | 9,93            |
|                  | 983,71           | 0,002            | 1,967           |
|                  | 992,263          | 0,002            | 9,111           |
|                  | 1002,937         | 0,002            | 0,839           |

Fig. 10. Tabla de ejecuciones con tiempos de ejecución de los tres algoritmos.

La Fig. 11 y la Fig. 12 muestran, respectivamente, los nuevos contenidos de la tabla de resumen numérico y los diagramas de resumen gráfico. El resumen numérico solamente incluye tiempos máximos, mínimos y medios. Asimismo, el diagrama muestra estos tiempos de forma proporcional.

| Medidas                       | menosCajasAprox1 | menosCajasAprox2 | menosCajasBack |
|-------------------------------|------------------|------------------|----------------|
| Núm. ejecuciones              | 100              | 100              | 100            |
| Núm. ejec. válidas del método | 100              | 100              | 100            |
| Núm. ejec. válidas en total   | 100              | 100              | 100            |
| Tiempo máximo                 | 9.474 ms         | 1.246 ms         | 13.225 ms      |
| Tiempo medio                  | 2.834 ms         | 0.002 ms         | 5.941 ms       |
| Tiempo mínimo                 | 1.133 ms         | 0.002 ms         | 0.013 ms       |

Fig. 11. Tabla de resumen numérico del experimento con el criterio de eficiencia.

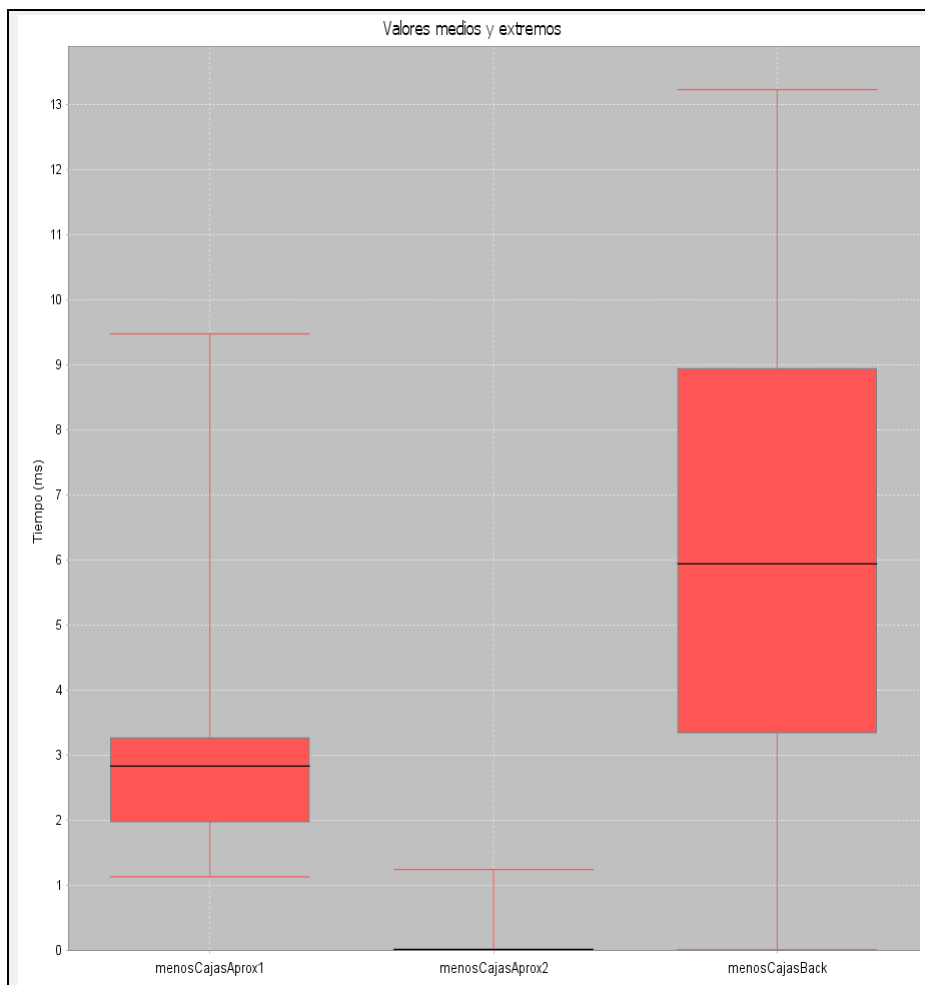


Fig. 12. Diagramas de resumen gráfico del experimento con el criterio de eficiencia.

## 1.2 Instalación

El sistema AlgorEx es distribuye como un fichero JAR. Se encuentra disponible en la siguiente URL:

<https://lite.etsii.urjc.es/tools/optimex>

Para ejecutar AlgorEx en Windows, basta con hacer doble clic en el fichero JAR. Se necesita tener instalado un JDK (*Java Development Kit*), versión 7 o posterior, para poder compilar las clases de Java y ejecutar sus métodos.

La primera vez que se ejecuta, AlgorEx busca automáticamente una JDK. Le sugiere una JDK al usuario, en un diálogo sencillo, dándole la posibilidad de navegar por el sistema de ficheros para elegir otra (véase Fig. 13).

Al acabar la primera sesión de AlgorEx, crea un fichero de configuración que incluye, entre otra información, el camino de la máquina virtual seleccionada. En consecuencia, este diálogo no se muestra en sesiones posteriores.

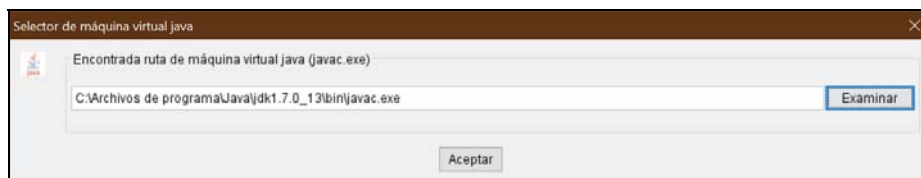


Fig. 13. Diálogo de selección de una máquina virtual de Java.

## 3 Interfaz de Usuario

La interfaz de usuario de AlgorEx consta de tres zonas:

- Parte superior. Contiene el menú principal, con submenús desplegables, que agrupan 28 funciones. Como en otras aplicaciones, un título de función que termina en puntos suspensivos significa que se abrirá un diálogo. Debajo del menú principal se encuentra la barra de iconos, con 21 iconos. AlgorEx es un sistema sencillo, por lo que la mayor parte de sus funciones se encuentran disponibles en la barra de iconos. Véase la Fig. 14.



Fig. 14. Parte superior izquierda de la interfaz de usuario.

- Panel de código. Se ubica en la mitad izquierda de la pantalla. Agrupa el editor en su parte superior y una pequeña zona de mensajes en su parte inferior. El editor es multicolor para facilitar la lectura del código y presenta las líneas numeradas a su izquierda (véase Fig. 15). En las cabeceras de los métodos hay, entre los números de línea y el código, un botón etiquetado “+” o “-” que permite contraer o expandir, respectivamente, su código,

El editor incluye 9 iconos en su parte inferior para operaciones de edición (véase Fig. 14). De izquierda a derecha, son: deshacer (atajo: Ctrl+Z), rehacer (Ctrl+V), cortar (Ctrl+X), copiar (Ctrl+C), pegar (Ctrl+V), ir a línea, buscar, buscar siguiente, y seleccionar todo (Ctrl+A).

El área de mensajes muestra los mensajes de error del compilador de Java. Cada mensaje se muestra descompuesto en tres partes para facilitar su comprensión. En la Fig. 15 se muestra un mensaje de error producido por la falta del signo de punto y coma en la línea 28.



Fig. 15. Parte inferior del panel de código.

- Panel de tablas. Se ubica en la mitad derecha de la pantalla. Agrupa cuatro pestañas, de las cuales tres contienen tablas y una, diagramas. Dada su mayor complejidad, se presentan en detalle en los subapartados siguientes.

### 3.1 Tabla de Datos

La tabla de datos alberga los datos de entrada a usar como juegos de prueba. Los valores de cada juego de prueba se almacenan en una fila distinta de la tabla. La columna primera contiene una numeración de los juegos de prueba, comenzando desde 1, y el resto de columnas, el valor de cada parámetro en los distintos juegos de prueba.

Los juegos de prueba preparados previamente constituyen un caso especial. Pueden cargarse en el primer diálogo de la función “Generar datos aleatorios...”. En la tabla de datos, se distinguen de los demás juegos de prueba en que su casilla de numeración está coloreada en color verde y además contiene su resultado optimal (véase Fig. 5). El resto de celdas de la tabla se producen con funciones del menú Archivo (véase apartado 4.1). Se presentan en color gris claro y sin resultado optimal asociado.

La operación de manipulación más frecuente sobre la tabla de datos consiste en seleccionar las filas que se van a usar como juegos de prueba en una operación de ejecución. Las filas seleccionadas se resaltan en color azul. La Fig. 5 muestra la tabla con todos los juegos de prueba seleccionados salvo el primero, que es un juego preparado.

Pueden realizarse tres operaciones de selección con el ratón:

- Pulsar en cualquier lugar de una fila. Queda seleccionada dicha fila.
- Pulsar en una fila y arrastrar hacia arriba o hacia abajo. Quedan seleccionadas todas las filas sobre las cuales se ha arrastrado el ratón.

- Pulsar en la cabecera de la tabla. Si no están seleccionadas todas las filas, produce la selección de todas. Si estuvieran seleccionadas todas, dejan de estarlo.

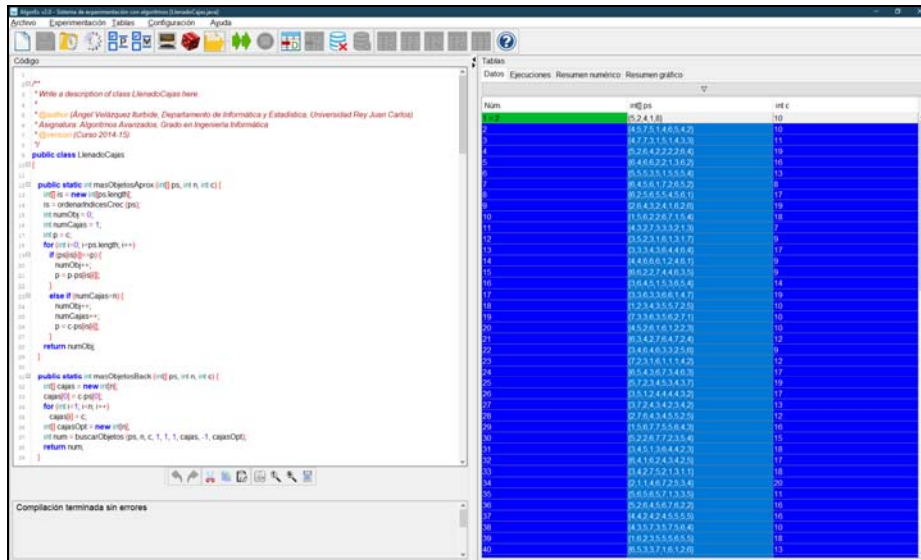


Fig. 5. Datos cargados y generados, almacenados en la tabla de datos.

### 3.2 Tabla de Ejecuciones

La tabla de ejecuciones alberga los resultados o los tiempos de ejecución de los algoritmos, tras ejecutarse con los juegos de prueba y según el criterio de experimentación seleccionados (véanse Fig. 6 y Fig. 10).

La columna primera tiene color gris oscuro y contiene la misma numeración que los juegos de prueba de la tabla de datos. El resto de columnas contienen los valores o los tiempos obtenidos al ejecutar cada algoritmo. Las celdas se colorean de forma distinta, dependiendo de que el criterio de la experimentación sea bien optimalidad bien eficiencia en tiempo o nulo.

Si el criterio de la experimentación es la optimalidad (véase Fig. 6), el valor calculado por cada algoritmo para cada juego de datos se almacena en la fila y columna correspondiente. Dichos valores se comparan dentro de cada fila según el criterio de optimización seleccionado (maximizar o minimizar) y se determina cuáles son optimales para el juego de prueba correspondiente. Las celdas de una fila que contengan valores optimales se colorean en verde oliva y los demás, en verde claro. Las columnas que tengan todas sus celdas de color verde oliva presentan sus valores en fuente azul claro para destacar que, según los resultados del experimento realizado, se trata de un algoritmo exacto.

En la Fig. 6, los dos primeros algoritmos han producido resultados suboptimales con algunos casos de prueba, mientras que el algoritmo tercero tiene un 100% de

casos optimales. Aunque los casos visibles en la tabla pueden dar la impresión de que el segundo algoritmo es exacto, la fuente blanca indica que hay casos suboptimales no visibles.

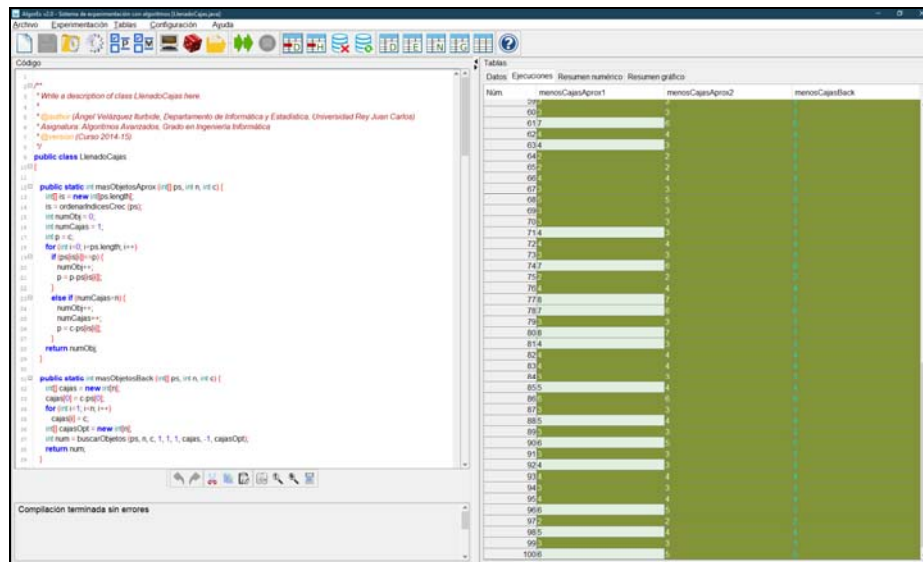


Fig. 6. Tabla de ejecuciones bajo el criterio de optimalidad, con los resultados calculados por los tres algoritmos.

Las celdas también pueden contener información de otros dos tipos. Por un lado, el usuario puede indicar con la función “Seleccionar método...” que un método dado es exacto. Obsérvese que en una situación como la descrita, en la que todos los valores de una fila se comparan entre sí para determinar el máximo en un problema de maximización (o el mínimo en un problema de minimización), solamente existe el valor optimal y, en todo caso, valores suboptimales. Sin embargo, al marcar un método como exacto, se considera que todos los valores que calcula son optimales. Por tanto, podría darse la situación en la que existan otros valores mayores (o menores en problemas de minimización) en la misma fila. Estas celdas corresponden a situaciones que no deberían darse, por lo cual se colorean rojizos. Se dará más información sobre estas situaciones en la tabla de resumen numérico (apartado 3.3).

Supongamos que, en nuestro ejemplo, se ha seleccionado el algoritmo inexacto *menosCajasAprox2* como exacto, lo cual produce que el algoritmo *menosCajasBack* genere un valor mayor en un caso. La Fig. 16 muestra la tabla de ejecuciones, con dicho caso anómalo coloreado rojizo.

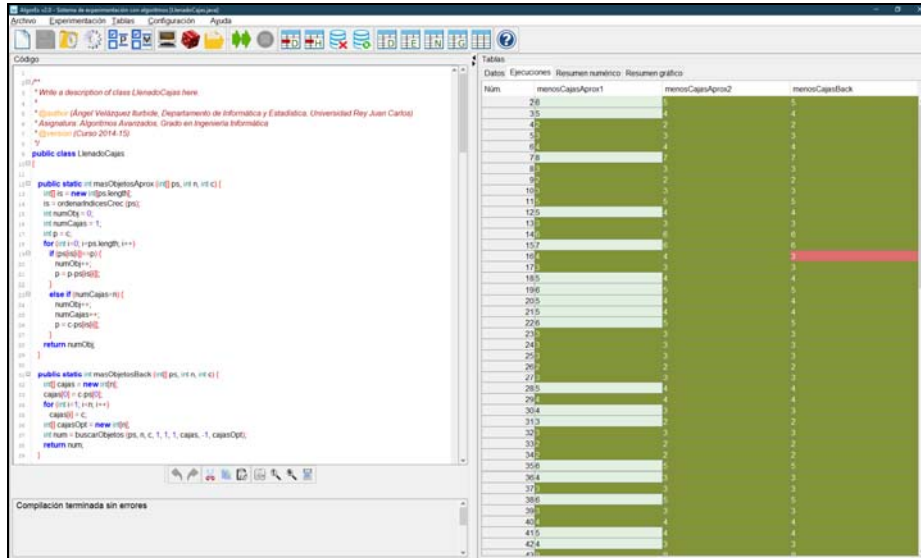


Fig. 16. Tabla de ejecuciones bajo el criterio de optimalidad, con los resultados calculados por los tres algoritmos y un resultado “sobreoóptimo”.

La Fig. 17 ilustra cómo se muestra en la tabla de resumen numérico esta situación anómala, con un valor “sobreoóptimo” en el método *menosCajasBack*.

| Medidas                          | menosCajasAprox1 | menosCajasAprox2 | menosCajasBack |
|----------------------------------|------------------|------------------|----------------|
| Núm. ejecuciones                 | 100              | 100              | 100            |
| Núm. ejec. válidas del método    | 100              | 100              | 100            |
| Núm. ejec. válidas en total      | 100              | 100              | 100            |
| % Soluciones subóptimas          | 41,00 %          | 0,00 %           | 0,00 %         |
| % Soluciones óptimas             | 59,00 %          | 100,00 %         | 99,00 %        |
| % Soluciones sobreoóptimas       | 0,00 %           | 0,00 %           | 1,00 %         |
| % Diferencia media subóptima     | 125,05 %         | 0,00 %           | 0,00 %         |
| % Diferencia máxima subóptima    | 150,00 %         | 0,00 %           | 0,00 %         |
| % Diferencia media sobreoóptima  | 0,00 %           | 0,00 %           | 75,00 %        |
| % Diferencia máxima sobreoóptima | 0,00 %           | 0,00 %           | 75,00 %        |

Fig. 17. Tabla de resumen numérico del experimento con el criterio de optimalidad y casos “sobreoóptimos”.

Asimismo, la Fig. 18 muestra el resumen gráfico. Para *menosCajasBack*, el diagrama de casos presenta en color rojo el 1% de porcentaje de casos “sobreoóptimos”, mientras que el diagrama de desviación muestra que calcula valores por debajo del 100%.



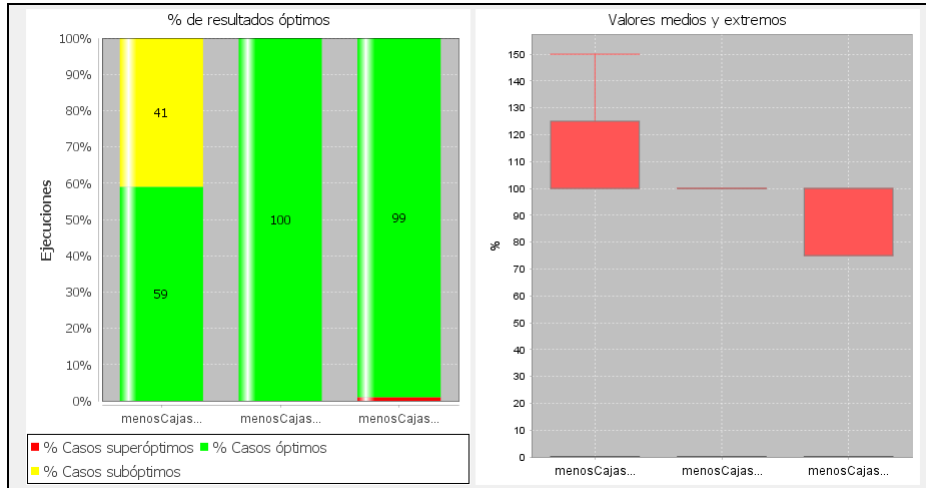


Fig. 18. Diagramas de resumen gráfico del experimento con el criterio de optimalidad, y con casos “sobreóptimos”.

Por último, si algún algoritmo produjera un error de ejecución para algún juego de datos, su celda correspondiente contendrá el mensaje “Tiempo excedido” y se coloreará de gris claro.

Si el criterio de la experimentación es la eficiencia (véase Fig. 10) o el criterio nulo, los valores mostrados no se comparan entre sí. Por tanto, todas las celdas se colorean del mismo color, verde claro. Con el criterio de eficiencia en tiempo, se muestran tiempos de ejecución medidos, expresados en milisegundos. Con el criterio nulo se muestran los resultados de los algoritmos, al igual que con el criterio de optimalidad pero sin comparar entre sí los resultados de cada fila.

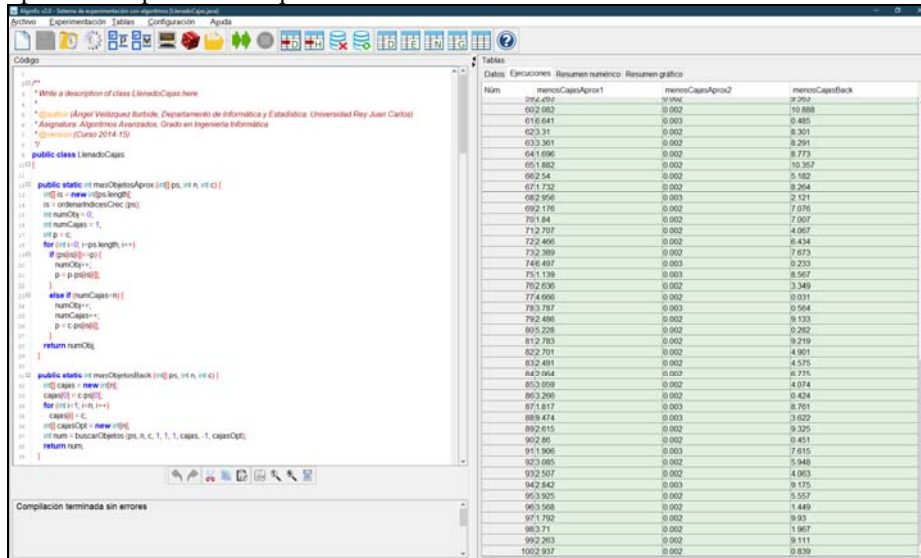


Fig. 10. Tabla de ejecuciones con tiempos de ejecución de los dos algoritmos.

### 3.3 Tabla de Resumen Numérico

La tabla de resumen numérico alberga un resumen estadístico de los resultados o los tiempos de ejecución medidos para los algoritmos, según el criterio de la experimentación seleccionado. La columna primera identifica la medida incluida en cada fila y el resto de columnas, los valores que toma dicha medida para cada algoritmo.

Cuando el criterio de experimentación es la optimalidad, se presentan 10 medidas (véase Fig. 7).

| Medidas                         | menosCajasAprox1 | menosCajasAprox2 | menosCajasBack |
|---------------------------------|------------------|------------------|----------------|
| Núm. ejecuciones                | 100              | 100              | 100            |
| Núm. ejec. válidas del método   | 100              | 100              | 100            |
| Núm. ejec. válidas en total     | 100              | 100              | 100            |
| % Soluciones subóptimas         | 42,00 %          | 1,00 %           | 0,00 %         |
| % Soluciones óptimas            | 58,00 %          | 99,00 %          | 100,00 %       |
| % Soluciones sobreóptimas       | 0,00 %           | 0,00 %           | 0,00 %         |
| % Diferencia media subóptima    | 125,25 %         | 133,33 %         | 0,00 %         |
| % Diferencia máxima subóptima   | 150,00 %         | 133,33 %         | 0,00 %         |
| % Diferencia media sobreóptima  | 0,00 %           | Ejecs. nº 31, 55 | 0,00 %         |
| % Diferencia máxima sobreóptima | 0,00 %           | 0,00 %           | 0,00 %         |

Fig. 7. Tabla de resumen numérico del experimento con el criterio de optimalidad.

Las medidas son de tres tipos:

- Números de ejecuciones:
  - Número de ejecuciones. Es el número de juegos de prueba sobre los que se han ejecutado todos los algoritmos seleccionados. Su valor es igual en todas las columnas.
  - Número de ejecuciones válidas del método. Para cada método, es el número de juegos de prueba que han terminado su ejecución normalmente, sin un error en tiempo de ejecución. Se considera un error en tiempo de ejecución que el algoritmo haya lanzado una excepción o que AlgorEx haya detenido una ejecución porque ha detectado que tardaba un tiempo excesivo.
  - Número de ejecuciones válidas en total. Es el número de juegos de prueba para los que todos los algoritmos han terminado su ejecución normalmente.

En la Fig. 7 no se ha producido ningún error en tiempo de ejecución, por lo cual los tres algoritmos presentan 100 ejecuciones en las tres filas.

- Porcentajes de soluciones válidas:
  - % soluciones subóptimas. Es el porcentaje de soluciones válidas de un algoritmo que han sido subóptimas. Una solución es subóptima en un problema de maximización si es menor que otra solución para el mismo juego de datos. Asimismo, es subóptima en un problema de minimización si es mayor (como sucede en el ejemplo mostrado).
  - % soluciones óptimas. Es el porcentaje de soluciones válidas de un algoritmo que han sido óptimas.

- % soluciones sobreóptimas. Damos el nombre de soluciones “sobreóptimas” a ciertas situaciones que no deberían suceder. Solamente aparecen casos “sobreóptimos” si el usuario marca con la función “Seleccionar métodos...” que un algoritmo es exacto. Entonces, en cada fila se toman como valores de referencia los valores calculados por dicho algoritmo. Obviamente, el algoritmo marcado tendrá un 100% de soluciones optimales. Sin embargo, puede suceder que algún otro algoritmo produzca en alguna fila un valor mayor que la solución “óptima” si el problema es de maximización o inferior si es de minimización. Este campo mide el porcentaje de soluciones válidas de cada algoritmo que corresponden a estos casos, es decir que son soluciones “sobreóptimas”.

En la Fig. 7 puede verse que el tercer algoritmo es exacto, con 100% de casos optimales, mientras que el primero y el segundo son inexactos, con 58% y 99%, respectivamente, de casos optimales y 42% y 1%, respectivamente, de casos suboptimales.

- Desviaciones respecto a las soluciones optimales:
  - % valor medio subóptimo. Se considera que la solución óptima de cada fila corresponde al 100%. A cada valor subóptimo o sobreóptimo situado en la misma fila le corresponde un porcentaje proporcional. Esta medida calcula el valor medio del porcentaje correspondiente a los casos válidos suboptimales. Obsérvese que, para problemas de maximización, será un valor inferior a 100% y para problemas de minimización, superior.
  - % valor extremo subóptimo. Al igual que en la medida anterior, se considera que la solución óptima de cada fila corresponde al 100%. Esta medida calcula el valor más alejado del 100% en los casos válidos suboptimales. De nuevo, para problemas de maximización, será un valor inferior a 100% y para problemas de minimización, superior.
  - % valor medio sobreóptimo. Se define se forma análoga al “% valor medio subóptimo” pero para los casos sobreóptimos.
  - % valor extremo sobreóptimo. Se define se forma análoga al “% valor extremo subóptimo” pero para los casos sobreóptimos.

En la Fig. 7 puede verse que los casos suboptimales del algoritmo primero alcanzan en media el 125,25% del valor óptimo, aunque en el caso más desfavorable asciende hasta el 150%. También puede observarse que, cuando el ratón se sitúa encima de alguna de las filas de valores extremos, AlgorEx muestra el número de los casos de prueba correspondientes (en la figura, los juegos de datos 31 y 55).

Cuando el criterio de experimentación es la eficiencia, se presentan 6 medidas (véase Fig. 11).

| Medidas                       | menosCajasAprox1 | menosCajasAprox2                           | menosCajasBack |
|-------------------------------|------------------|--|----------------|
| Núm. ejecuciones              | 100              | 100  | 100            |
| Núm. ejec. válidas del método | 100              | 100  | 100            |
| Núm. ejec. válidas en total   | 100              | 100  | 100            |
| Tiempo máximo                 | 9.474 ms         | 1.246 ms                                   | 13.225 ms      |
| Tiempo medio                  | 2.834 ms         | ID: 87 - Data: [[4,3,6,5,3,6,4,6,6,2], 17] | 5.941 ms       |
| Tiempo mínimo                 | 1.133 ms         | 0.002 ms                                   | 0.013 ms       |

**Fig. 11.** Tabla de resumen numérico del experimento con el criterio de eficiencia.

Las medidas son de dos tipos:

- Números de ejecuciones. Se definen igual que con el criterio de optimalidad:
  - Número de ejecuciones.
  - Número de ejecuciones válidas del método.
  - Número de ejecuciones válidas en total.
- Tiempos de ejecución:
  - Tiempo máximo. Es el tiempo máximo tardado por el algoritmo en alguna ejecución válida.
  - Tiempo medio. Es el tiempo medio tardado por el algoritmo en todas sus ejecuciones válidas.
  - Tiempo mínimo. Es el tiempo mínimo tardado por el algoritmo en alguna ejecución válida.

En la Fig. 11 puede verse que los tiempos de ejecución del algoritmo tercero (de vuelta atrás) son mayores que los otros dos (algoritmos aproximados). No obstante, el algoritmo primero también tiene un tiempo relativamente alto en el caso peor.

Cuando el criterio de experimentación es “nulo”, sólo se presentan los 3 números de ejecuciones también mostrados bajo los criterios anteriores (véase Fig. 19).

| Medidas                       | menosCajasAprox1 | menosCajasAprox2 | menosCajasBack |
|-------------------------------|------------------|------------------|----------------|
| Núm. ejecuciones              | 100              | 100              | 100            |
| Núm. ejec. válidas del método | 100              | 100              | 100            |
| Núm. ejec. válidas en total   | 100              | 100              | 100            |

**Fig. 19.** Tabla de resumen numérico del experimento con el criterio nulo.

### 3.4 Pestaña de Resumen Gráfico

La pestaña de resumen gráfico presenta en forma de diagramas los resultados o tiempos de ejecución presentados en la tabla de resumen numérico, salvo los números de ejecuciones (véanse Fig. 8 y Fig. 12). Solamente se muestra para los criterios de optimalidad o de eficiencia en tiempo, presentando uno o dos diagramas por algoritmo, respectivamente.

Cuando el criterio de experimentación es la optimalidad, pueden presentarse 2 diagramas por algoritmo (véase Fig. 7).

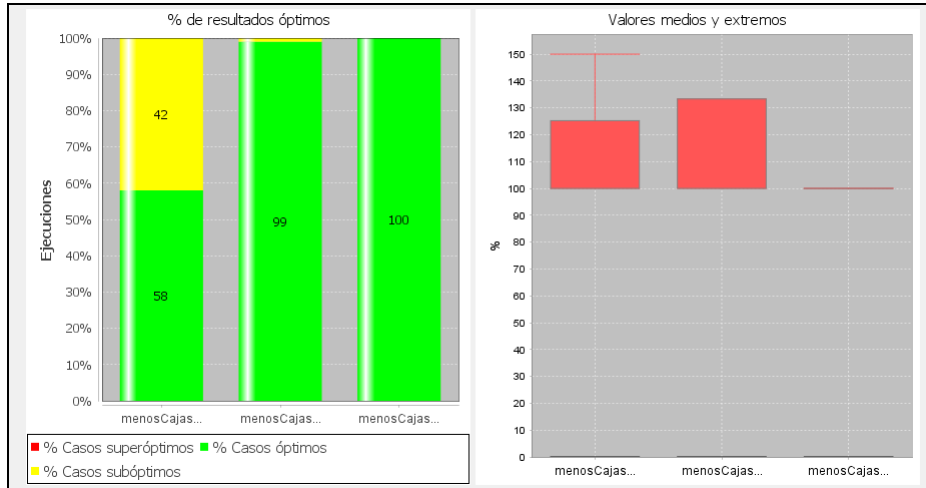


Fig. 8. Diagramas de resumen gráfico del experimento con el criterio de optimalidad.

Veamos cada diagrama:

- Diagrama de casos. Está situado a la izquierda de la pestaña. Recoge los porcentajes de soluciones válidas que son subóptimas, óptimas y sobreóptimas para cada algoritmo. Tiene el mismo formato que el diagrama de “columna 100% apilada” de Excel.

Cada algoritmo tiene asociada una barra cuya altura total suma el 100% de casos. La barra se parte en tres zonas, que corresponden a los porcentajes de casos subóptimos (en amarillo), óptimos (en verde) y sobreóptimos (en rojo). Si algún porcentaje es nulo, no se dibuja. Para mayor precisión, cada zona contiene también su porcentaje correspondiente, siempre que quepa en la misma.

Supongamos que un diagrama de casos presenta tres porcentajes. En un problema de maximización, se presentan, de arriba abajo, los casos sobreóptimos, óptimos y subóptimos. En un problema de minimización, los casos sobreóptimos y subóptimos aparecen invertidos, es decir, los casos subóptimos se presentan arriba (calculan costes mayores) y los sobreóptimos debajo (calculan costes menores).

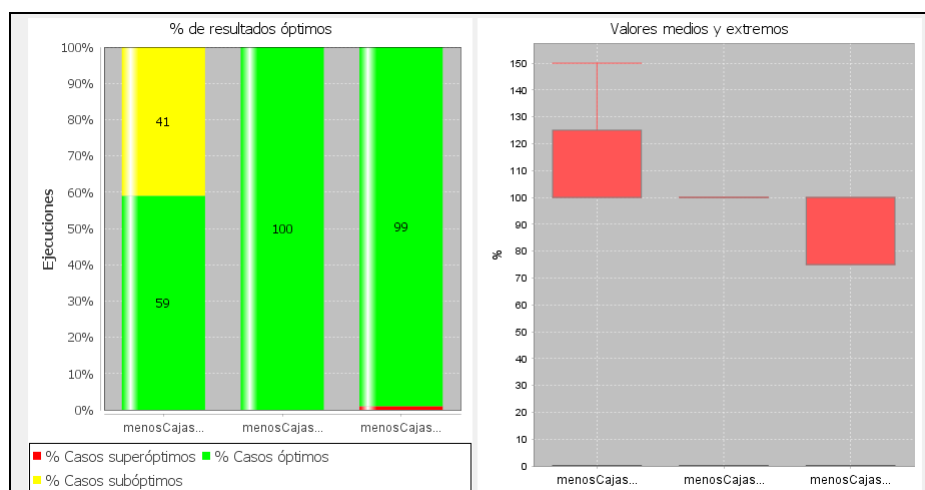
Véase la Fig. 8, con porcentajes subóptimos y óptimos solamente, y la Fig. 18, con porcentajes de los tres tipos.

- Diagrama de desviación. Está situado a la derecha de la pestaña. Solamente se muestra si hay algún algoritmo exacto. Es un diagrama basado en el diagrama estadístico de cajas o bigotes (*boxplot*).

Supongamos que es un problema de maximización y que el algoritmo tiene casos subóptimos. El diagrama de desviación presenta un rectángulo que baja desde el 100% al “% valor medio subóptimo”. Además, se prolonga una línea vertical hasta una línea horizontal situada en la posición correspondiente a “% valor extremo subóptimo”. De igual forma, si el algoritmo presenta valores sobreóptimos, se prolongará hacia arriba una representación de las dos medidas de dispersión correspondientes. Si el problema es de minimización,

como en el ejemplo, la representación es similar, pero los casos subóptimos se dibujan por encima del 100% y los sobreóptimos, por debajo.

Véase la Fig. 8, con casos subóptimos solamente, y la Fig. 18, con casos subóptimos y sobreóptimos. Recuérdese que solamente presentan desviación los algoritmos inexactos.



**Fig. 18.** Diagramas de resumen gráfico del experimento con el criterio de optimalidad, y con casos "sobreóptimos".

Cuando el criterio de experimentación es la eficiencia en tiempo, se presenta un solo diagrama por algoritmo (véase Fig. 12). De nuevo, el diagrama está basado en el diagrama estadístico de cajas.

El significado de los elementos del diagrama es el siguiente:

- Los tiempos máximo y mínimo corresponden con las líneas horizontales rojas situadas en las partes superior e inferior del diagrama.
- La caja sólida, roja, engloba el 50% de los tiempos medidos, en concreto los tiempos comprendidos entre los 25% tiempos menores y el 75%.
- Una línea horizontal negra representa el tiempo medio. Al corresponder al tiempo medio y no a la mediana de los tiempos medidos (que estaría situado en la posición 50%), puede aparecer fuera de la caja.

El resumen gráfico permite interactuar arrastrando el ratón sobre él, produciendo zoom. Su uso es frecuente porque el rango de tiempos medidos suele ser grande, resultando en un diagrama donde apenas se ven los tiempos de los algoritmos más rápidos.

Conviene avisar de que algunos tiempos medidos son desmesuradamente mayores que los tiempos medidos para otros datos del mismo tamaño. Esto se debe a peculiaridades de la máquina virtual de Java. Es el caso del primera algoritmo en nuestro ejemplo.

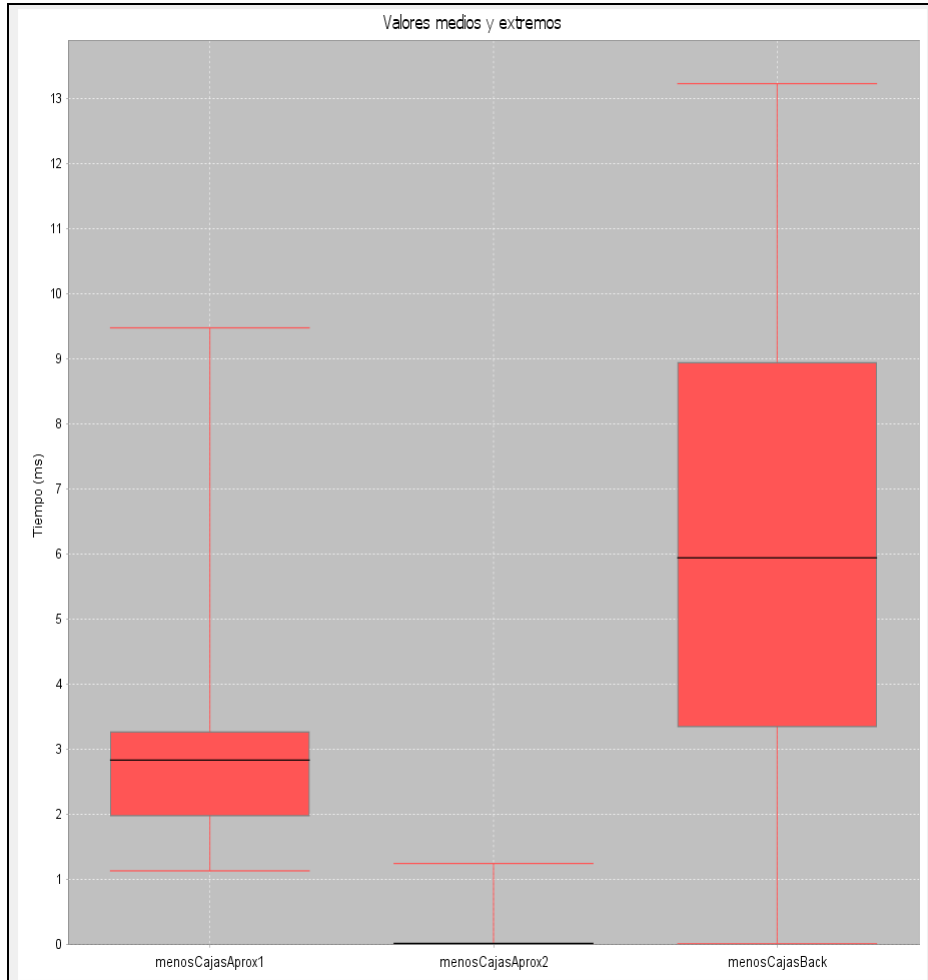


Fig. 12. Diagramas de resumen gráfico del experimento con el criterio de eficiencia.

## 4 Funciones de AlgorEx

Veamos las funciones de AlgorEx siguiendo la estructura del menú principal, que estructura 28 funciones en 5 opciones principales. No se explican apenas algunas funciones cuya etiqueta es autoexplicativa o que ya se han explicado suficientemente en los apartados anteriores.

### 4.1 Menú Archivo

El menú Archivo agrupa las funciones relacionadas con el manejo de ficheros o entrada de juegos de prueba. Incluye 10 opciones organizadas en 3 grupos:

- “Nueva clase”. Se crea una clase vacía para que el usuario la edite y guarde. En caso de que ya se haya realizado un experimento en esta sesión, AlgorEx pide confirmación al usuario porque se perdería el contenido de todo el panel de tablas.
- “Guardar clase...”. Guarda en disco la clase cargada en el editor.
- “Cargar y compilar clase...”. Se navega por el sistema de ficheros, se selecciona un fichero Java, se compila y se carga en el editor. También pide confirmación en caso de haber realizado un experimento.
- “Compilar clase”. Compila la clase cargada en el editor. También pide confirmación en caso de haber realizado un experimento.
- “Exportar datos...”. Exporta la cabecera y las filas de la tabla de datos en un fichero. Sin embargo, no se exporta la columna de numeración. El usuario puede elegir la exportación como un fichero de texto (TXT), de marcas (XML) o gráfico (GIF, JPG o PNG). Puede encontrarse más información sobre el formato XML en la ayuda interactiva de AlgorEx.
- “Exportar ejecuciones...”. Exporta la cabecera y las filas de la tabla de ejecuciones. Sin embargo, no se exporta la columna de numeración. El usuario puede elegir la exportación como un fichero Excel (XLS) o gráfico (GIF, JPG o PNG).
- “Exportar resumen numérico...”. Exporta la cabecera y las filas de la tabla de resumen numérico. El usuario puede elegir la exportación como un fichero Excel (XLS) o gráfico (GIF, JPG o PNG).
- “Exportar resumen gráfico...”. Exporta el contenido de la pestaña de resumen gráfico. El usuario puede elegir un formato gráfico (GIF, JPG o PNG).
- “Exportar todas las tablas...”. Se exporta el contenido de las cuatro pestañas del panel de tablas en cuatro hojas de un fichero Excel, llamadas Datos, Ejecuciones, ResumenNumerico y Graficas. En los tres primeros casos, se exporta tanto la cabecera de la tabla como sus filas, pero no se exporta la columna de numeración.
- “Salir”. Si se ha experimentado, pide una confirmación al usuario.

## 4.2 Menú Experimentación

El menú Experimentación agrupa las funciones relacionadas con la preparación y realización del experimento. Incluye 7 opciones organizadas en 3 grupos. Las dos primeras opciones permiten declarar las condiciones del experimento:

- “Seleccionar problema...”. El diálogo permite configurar la experimentación de forma adecuada (véase Fig. 2).



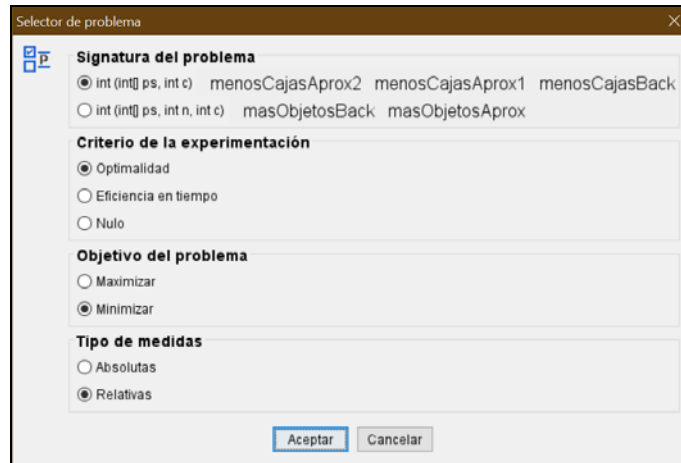


Fig. 2. Diálogo de selección del problema.

El diálogo consta de las siguientes partes:

- Signatura del problema. Presenta las firmas de los métodos públicos de la clase. Se presenta el tipo de datos del resultado, seguido de los tipos de datos de los parámetros, separados por blancos y entre paréntesis. También se muestra el identificador de los parámetros si son comunes entre los distintos métodos y se utiliza un JDK de la versión 8 o posterior. Para facilitar la selección, a la derecha de cada firma se muestra el identificador de los métodos que tienen dicha firma.

Si hubiera una única firma, esta selección está deshabilitada y la firma está preseleccionada. Si el diálogo se ha invocado tras haber experimentado y se modifica la firma, el sistema nos avisará de que se perderá todo el contenido del panel de tablas.

- Criterio de la experimentación. Permite elegir entre optimalidad, eficiencia en tiempo (de ejecución) y nulo. Este último criterio está concebido para permitir la realización de ejecuciones con muchos juegos de datos sin comparar los algoritmos entre sí. Puede resultar útil para consultar los resultados de algoritmos que son de optimización.

Si el diálogo se ha invocado tras haber experimentado y se modifica el criterio, el sistema nos avisará de que se perderán los resultados. En caso de confirmar la acción, se conserva el contenido de la tabla de datos, pero se borra el contenido del resto de tablas.

- Objetivo del problema. Esta opción sólo se habilita cuando el criterio de la experimentación es optimalidad, permitiendo indicar si la función objetivo del problema es de maximización o de minimización.
- “Seleccionar medidas...”. Esta opción sólo se habilita cuando el criterio de la experimentación es optimalidad, permitiendo elegir entre medidas de desviación absolutas y relativas. Las medidas de desviación absolutas miden la diferencia (real o entera) entre las soluciones optimales y las demás. Las medidas de desviación relativas miden la proporción entre las soluciones optimales y las demás, suponiendo que

una solución óptima corresponde al 100%. Por defecto, se toman medidas relativas.

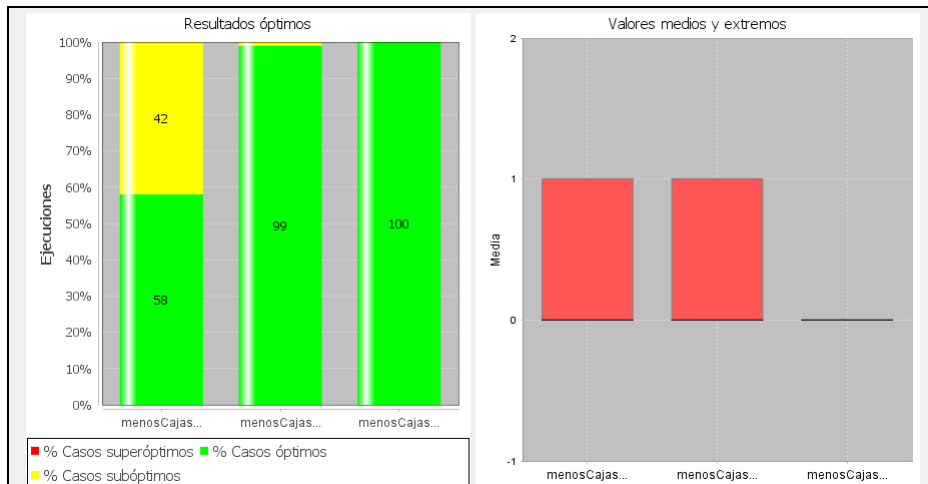
La selección de medidas absolutas afecta a los resúmenes numérico y gráfico. El resumen numérico utiliza medidas de desviación absolutas (véase Fig. 20). Su nombre y definición son parecidos a los usados para medidas relativas.

- Diferencia media subóptima.
- Diferencia extrema subóptima.
- Diferencia media sobreóptima.
- Diferencia extrema sobreóptima.

| Medidas                       | menosCajasAprox1 | menosCajasAprox2                           | menosCajasBack |
|-------------------------------|------------------|--|----------------|
| Núm. ejecuciones              | 100              | 100  | 100            |
| Núm. ejec. válidas del método | 100              | 100  | 100            |
| Núm. ejec. válidas en total   | 100              | 100  | 100            |
| % Soluciones subóptimas       | 41,00 %          | 0,00 %                                     | 0,00 %         |
| % Soluciones óptimas          | 59,00 %          | 100,00 %                                   | 99,00 %        |
| % Soluciones sobreóptimas     | 0,00 %           | 0,00 %                                     | 1,00 %         |
| Diferencia media subóptima    | 1,00             | 0,00                                       | 0,00           |
| Diferencia máxima subóptima   | 1,00             | 0,00                                       | 0,00           |
| Diferencia media sobreóptima  | 0,00             | Ejec. nº 2, 3, 7, 12, 15, 16, 18, 19, etc. | 1,00           |
| Diferencia máxima sobreóptima | 0,00             | 0,00                                       | 1,00           |

**Fig. 20.** Tabla de resumen numérico del experimento con el criterio de optimalidad y medidas absolutas.

El diagrama gráfico de desviación muestra las medidas de desviación absolutas de forma similar a las relativas, con la única diferencia de que la escala no mide porcentajes sino valores numéricos (véase Fig. 21).



**Fig. 21.** Diagramas de resumen gráfico del experimento con el criterio de optimalidad y medidas absolutas.

- “Seleccionar métodos...”. Una vez que se ha elegido un problema en AlgorEx, por defecto se usan todos los métodos públicos de la clase que tengan dicha signatura. Sin embargo, el usuario puede dejar fuera del experimento algunos métodos, por ejemplo, por estar inacabados o crear alguna clase de problema.

La Fig. 22 muestra el diálogo que se abre. Puede observarse que se presentan todos los métodos que tienen la signatura seleccionada, de arriba abajo. Para cada método, pueden marcarse dos opciones: si se quieren comparar sus resultados y si se quiere tomar el método como referencia para las comparaciones. Por defecto, todos los métodos están seleccionados y no se considera que ninguno sea óptimo, sino que deben compararse sus resultados para determinar qué valor es óptimo para cada juego de prueba.

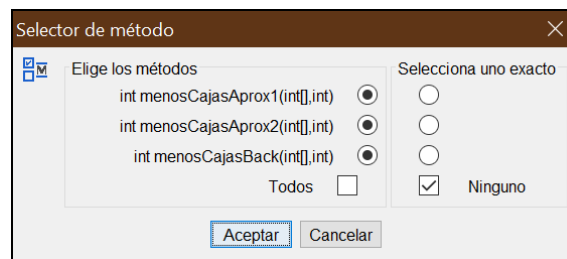


Fig. 22. Diálogo de selección de métodos.

El segundo grupo de tres opciones del menú Experimentación permite producir los juegos de datos para el experimento:

- “Introducir datos por teclado”. Se presenta un diálogo a medida para que el usuario introduzca un juego de prueba. Indica el tipo de datos de cada parámetro. La Fig. 23 muestra el diálogo para la versión de minimización del problema del llenado de cajas.

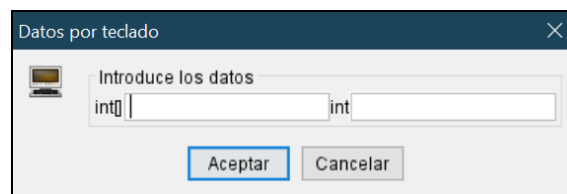


Fig. 23. Diálogo de introducción de datos por teclado.

- “Generar datos aleatorios...”. Se presentan sucesivamente dos diálogos. En el primer diálogo se pide el número de casos de prueba a generar. Por defecto, se generan 100. También ofrece la posibilidad de cargar automáticamente juegos de datos y rangos de valores predefinidos:
  - Cargar datos automáticamente. Permite cargar automáticamente juegos de datos preparados. Se ha concebido para cargar casos de prueba cuyo resultado óptimo se conoce, como ejemplos incluidos en el enunciado de un problema. En caso de marcar esta casilla, AlgorEx busca, en el mismo directorio donde se encontraba la clase cargada en el editor, un

fichero de texto con el mismo nombre que la clase y con extensión DATA, y carga los casos de prueba correspondientes al problema seleccionado en la tabla de datos (véase apartado 3.1).

El fichero puede contener casos de prueba para todas las firmas de métodos públicos de la clase. La Fig. 24 muestra un ejemplo para el problema del llenado de cajas, con casos de prueba para dos problemas. El fichero comienza con un caso de prueba para la versión de maximización del problema y sigue con otro caso para la versión de minimización, separados por una línea con un punto y coma.

```
int[] int int -> int
{2,3,4,5} 2 7 -> 4
;
int[] int -> int[]
{5,2,4,1,8} 10 -> 2
```

Fig. 24. Fichero de texto de carga automática, con juegos de datos preparados.

Para cada problema, debe escribirse su firma en una primera línea. Se escribirá el tipo de datos de cada parámetro, separados por blancos, seguido de una flecha (“->”) y el tipo de datos del resultado. Después, puede escribirse uno o más casos de prueba, cada uno en una línea distinta. Cada caso de prueba se escribe con una sintaxis similar a la de la firma: valores de los parámetros, separados por blancos, una flecha y valor óptimo del problema para dichos datos de entrada.

Aunque el fichero sea común para varios problemas, en caso de cambiar de problema, deberá leerse el fichero de nuevo, igual que deben generarse datos de nuevo.

- Cargar restricciones automáticamente. Permite cargar automáticamente los rangos de valores a usar para la generación aleatoria de juegos de prueba. Dado que el diálogo de generación de datos recuerda los últimos rangos usados, esta función está concebida para la experimentación con un mismo problema en varias sesiones. En caso de marcar esta casilla, AlgorEx busca, en el mismo directorio donde se encontraba la clase cargada en el editor, un fichero de texto con el mismo nombre que la clase y con extensión CNSTR.

El fichero puede contener restricciones para las distintas firmas de métodos públicos de la clase. La Fig. 25 muestra un ejemplo para el problema del llenado de cajas, con restricciones para dos problemas. El fichero comienza con las restricciones para la versión de maximización del problema y sigue con las restricciones de la versión de minimización, separados por una línea con un punto y coma.

Para cada problema, debe escribirse su firma en una primera línea. Se escribirá el tipo de datos de cada parámetro, separados por blancos, seguido de una flecha y el tipo de datos del resultado. Después, se escriben las restricciones de cada parámetro, cada uno en una línea distinta. Los valores a escribir para cada parámetro son los mismos que

aparecen en el diálogo de generación aleatoria de datos (compárense los valores de la segunda parte de la Fig. 25 con los valores de la Fig. 4).

```
int[] int int
10 1 7
1 10
7 20
;
int[] int
10 1 7
7 20
```

Fig. 25. Fichero de texto de carga automática, con restricciones preparadas.

Aunque el fichero sea común para varios problemas, en caso de cambiar de problema, deberá leerse el fichero de nuevo, igual que deben generarse datos de nuevo.

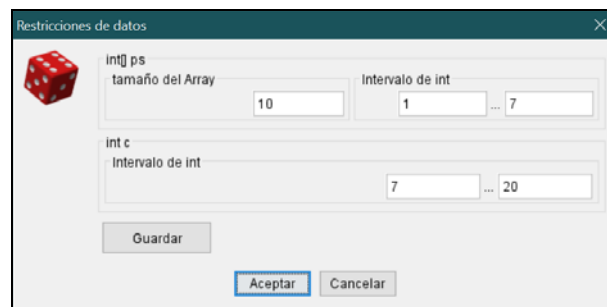


Fig. 4. Segundo diálogo de la generación aleatoria de datos.

El segundo diálogo de la función “Generar datos aleatorios...” incluye una opción Guardar (véase la Fig. 4) que permite que los valores de los campos del diálogo se almacenen como restricciones a usar en futuras sesiones. Un fichero de restricciones es un fichero de texto con extensión CNSTR y el formato mostrado en Fig. 25.

Este diálogo se presenta a medida para que se introduzca la longitud de los arrays y los rangos de valores de las variables simples a generar. Por defecto, la longitud de los arrays es 12 y los valores enteros varían entre 0 y 30 (véase Fig. 26).

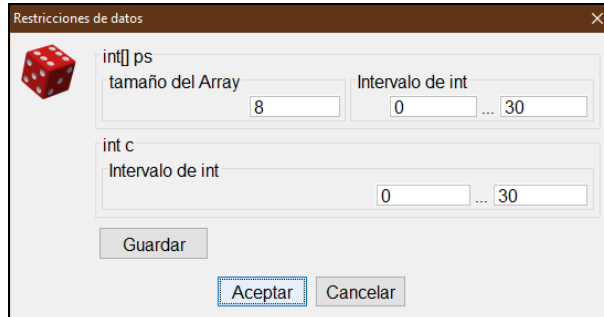


Fig. 26. Segundo diálogo de la generación aleatoria de datos con valores por defecto.

- “Cargar datos de fichero...”. Permite cargar juegos de prueba de un fichero TXT o XML, bien creados manualmente bien resultantes de exportar la tabla de datos. La Fig. 27 muestra el formato de los datos para el problema de ejemplo. Puede verse que consta de varias líneas, mostrando la primera los tipos de datos de los parámetros y las siguientes, los distintos juegos de prueba. En la Fig. 27 se muestra el comienzo de los datos mostrados en la Fig. 5. El primero es un juego de prueba de longitud 5 cargado automáticamente al seleccionar el problema, mientras que el segundo y posteriores son juegos de longitud 10 generados aleatoriamente.

```
int[] int
{5,2,4,1,8} 10
{6,1,6,7,2,4,4,4,7,2} 12
{1,6,2,6,4,3,5,3,5,7} 13
{5,7,5,2,3,5,7,5,4,7} 18
```

Fig. 27. Fichero de texto con datos de entrada.

Un tercer grupo de dos opciones del menú Experimentación permite controlar la ejecución de los algoritmos seleccionados:

- “Ejecución”. Produce la ejecución de los métodos sobre los juegos de prueba seleccionados. Aquellos juegos de datos que ya se ejecutaron no se ejecutan de nuevo. Según avanza la ejecución de los juegos de datos, se van actualizando las tablas, presentándose una barra informativa del número y porcentaje de juegos de datos ya ejecutados.

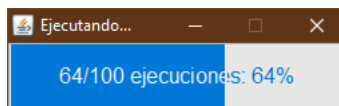


Fig. 28. Barra informativa del avance de las ejecuciones.

Si algún caso de prueba está tardando un tiempo excesivo en terminar, AlgorEx lo detecta y consulta al usuario si debe parar su ejecución. De esta forma, pueden evitarse esperas muy largas e incluso ejecuciones infinitas. Al

usuario se le ofrece las opciones de continuar y de parar, aprovechando para preguntar si su respuesta se aplicará a este caso solamente o también otros casos que puedan detectarse (véase Fig. 29).

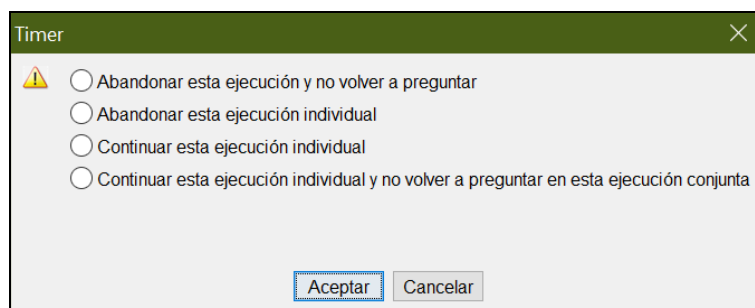


Fig. 29. Diálogo de consulta por tiempo excesivo de ejecución de un caso de prueba.

- “Parar ejecución”. Permite parar la ejecución en marcha. Está concebido para aquellos casos en los que la ejecución avanza muy lenta, bien porque son muchos casos de prueba bien porque algún algoritmo tarda mucho en cada ejecución individual. La parada se realiza de forma ordenada, de forma que se hayan medido todos los algoritmos para los casos de prueba ejecutados.

### 4.3 Menú Tablas

El menú Tablas agrupa 4 funciones de manejo (de borrado) de tablas:

- “Borrar fila en tabla de datos”. Borra la fila seleccionada de la tabla de datos. Si hay varias filas seleccionadas, borra la primera. El borrado de una fila se hace manteniendo la coherencia entre las tablas. Por tanto, también se borra la fila correspondiente de la tabla de ejecuciones y se actualizan los resúmenes numérico y gráfico.
- “Borrar fila en ejecuciones”. Borra la fila seleccionada de la tabla de ejecuciones. Sólo puede seleccionarse una fila. El borrado de una fila se hace manteniendo la coherencia entre las tablas, modificando de forma coherente la tabla de datos, la tabla de resumen numérico y la pestaña de resumen gráfico.
- “Borrar datos de sesión”. Se borra el contenido de todas las pestañas del panel de tablas.
- “Borrar resultados, mantener datos”. Se mantiene el contenido de la tabla de datos, pero se borra el contenido de las demás pestañas del panel de tablas.

### 4.4 Menú Configuración

El menú Configuración agrupa 4 funciones de configuración de la aplicación:

- “Idioma...”. Selecciona el idioma de los textos usados en la interfaz de usuario. Actualmente, puede elegirse entre español e inglés.

- “Máquina virtual Java...” Permite cambiar de máquina virtual navegando por el sistema de ficheros. Es el mismo diálogo que surge al ejecutar AlgorEx por primera vez (véase Fig. 13).
- “Actualización continua de resúmenes”. Es una opción que puede estar activada o no. Si está activada, cada vez que se completa la ejecución de una fila de la tabla de ejecuciones, se actualizan los resúmenes numérico y gráfico. Si está desactivada, solamente se actualizan los resúmenes una vez terminada la ejecución de todos los casos de prueba seleccionados. Esta segunda opción está concebida para acelerar el proceso experimental. Por defecto, la opción está activada.
- “Mostrar guía rápida al inicio”. Es una opción que puede estar activada o no. Si está activada, cuando el sistema arranca, se muestra una ventana que muestra la secuencia ideal de pasos a realizar en un experimento algorítmico (véase Fig. 31, en el siguiente subapartado). Por defecto, la opción está activada.

#### 4.5 Menú Ayuda

El menú Ayuda agrupa 3 funciones de ayuda e información sobre AlgorEx:

- “Ayuda sobre AlgorEx”. Contiene la ayuda interactiva de AlgorEx, organizada de varias formas con un índice completo (véase Fig. 30).

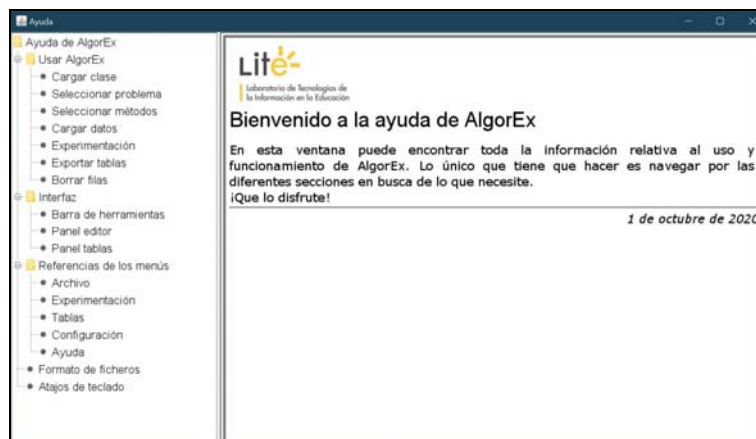


Fig. 30. Ayuda interactiva.

- “Sobre AlgorEx”. Contiene un reconocimiento a los desarrolladores de las distintas versiones de OptimEx y AlgorEx, así como la licencia de uso.
- ”Guía rápida de experimentación”. Resume la secuencia de pasos que formarían un experimento ideal (véase Fig. 31). Esta ventana se muestra la primera vez que se ejecuta AlgorEx para que sirva de orientación al usuario novel. Puede desactivarse para que no resulte molesta su aparición al usuario



cuando ya está familiarizado con el sistema, pero puede volverse a consultar con esta función de ayuda.

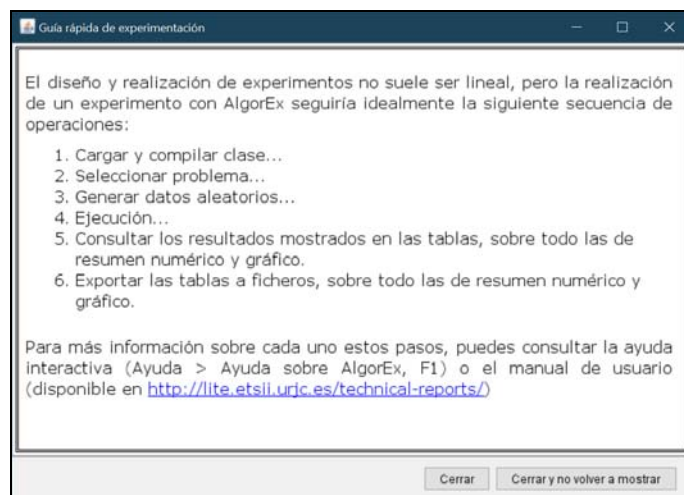


Fig. 31. Guía rápida de experimentación.

**Agradecimientos.** Este trabajo se ha financiado con los proyectos de investigación TIN2015-66731-C2-1-R del Ministerio de Economía y Competitividad y S2018/TCS-4307 de la Comunidad Autónoma de Madrid. Este último también está financiado con fondos estructurales (FSE y FEDER).

## Referencias

1. Velázquez Iturbide, J.Á.: Design and evaluation of OptimEx, an experimentation system for optimization algorithms. En: ICT in Education - Multiple and Inclusive Perspectives, M. J. Marcelino A. J. Mendes y C. Azevedo Gomes (eds.), Springer (2016) 51-68, DOI 10.1007/978-3-319-22900-3\_4
2. Velázquez-Iturbide, J.Á.: GreedEx and OptimEx: Two tools to experiment with optimization algorithms. International Journal of Engineering Education, 32, 3A (2016) 1.097-1.106
3. Velázquez Iturbide, J.Á., Escudero Angulo, J.J.: Extending instructional support in an algorithm benchmarking system. En: Proceedings of the 21st International Symposium on Computers in Education (SIIE 2019), Gonçalo Marques, C., Pereira, I., y Pérez, D. (eds.). IEEE Xplore, 2019, 6 págs., DOI [10.1109/SIIE48397.2019.8970134](https://doi.org/10.1109/SIIE48397.2019.8970134)
4. Velázquez Iturbide, J.Á.: OptimEx versión 1.4: Manual de uso. Serie de Informes Técnicos DLSII-URJC, no. 2020-03, Universidad Rey Juan Carlos, 2020. Disponible en: <http://lite.etsii.urjc.es/technical-reports/>
5. Brassard, G., Bratley, P.: Fundamentals of Algorithmics. Englewood Cliffs, NJ: Prentice-Hall, 1996