



## Programación de Sistemas y Concurrencia

Práctica Evaluable  
Curso 2022-2023

### Descripción del sistema

Queremos comparar el tiempo que tardan en ordenarse listas de  $N$  números (almacenadas en arrays del tipo  $a[0..N-1]$ ) utilizando el algoritmo de ordenación por selección y el algoritmo de la burbuja, que se describen a continuación:

---

#### Ordenación por selección

```
for (int i=0; i<N-1; i++){  
    //buscamos el menor elemento del array a[i..N-1]  
    //y lo intercambiamos con a[i]  
    int menor = i  
    for (int j=i+1; j<N; j++){  
        if (a[j]<a[menor]) menor = j;  
    }  
    int aux = a[i]  
    a[i] = a[menor]  
    a[menor] = aux  
    //Inv: a[0..i] tiene los i+1 menores elementos de a[0..N-1]  
    //Inv: a[0..i] está ordenado  
}
```

---

#### Método de ordenación Burbuja

```
for (int i=0; i<N-1; i++){  
    //recorremos el array a[i..N-1] desde la posición N-1 a la i.  
    //Comparamos los elementos dos a dos y los intercambiamos si  
    //están desordenados  
    for (int j=N-1; j>i; j--){  
        if (a[j]<a[j-1]){  
            int aux = a[j]  
            a[j] = a[j-1]  
            a[j-1] = aux  
        }  
    }  
    //Inv: a[0..i] tiene los i+1 menores elementos de a[0..N-1]  
    //Inv: a[0..i] está ordenado  
}
```

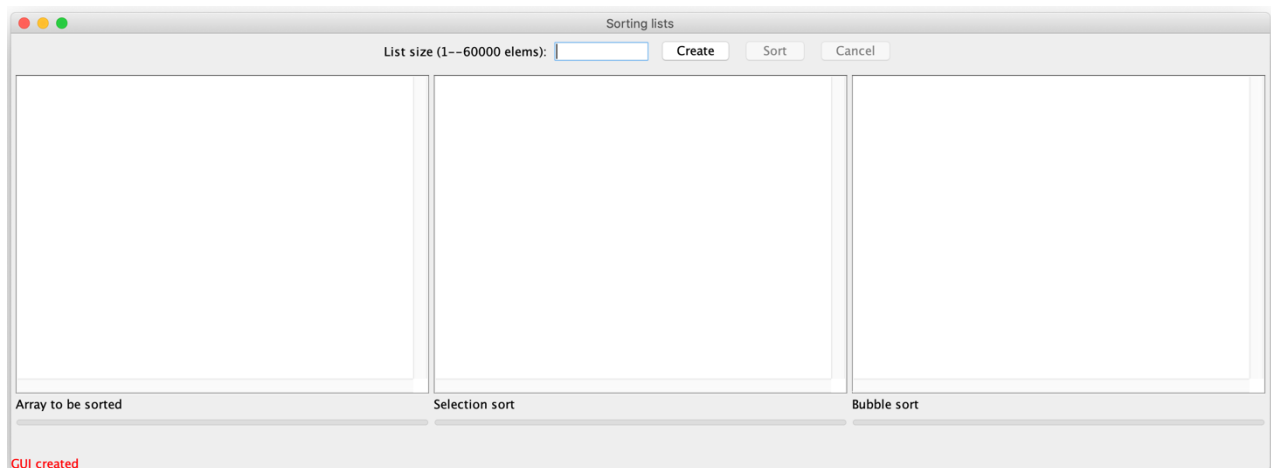
Este algoritmo se puede optimizar de varias formas. Una de ellas consiste en dejar de iterar el bucle más externo cuando se detecta que en una iteración no se ha intercambiado ningún par de elementos (lo que significa que la lista está ordenada).

---

Observa que en los dos algoritmos descritos, en cada iteración del bucle más externo la sublista  $a[0..i]$  se encuentra ya ordenada.

Se pide realizar una GUI que, dada una lista de números, generados de forma aleatoria, la muestre ordenada utilizando los dos algoritmos descritos arriba. En el campus virtual tienes la plantilla de la clase **Panel** que te servirá para empezar el proyecto. Hay que hacer tres versiones de este proyecto, y entregar cada una de ellas en un fichero comprimido.

La clase **Panel** proporcionada permite crear una interfaz con el aspecto que se muestra en la figura:



Los principales elementos de la interfaz son:

1. Un campo de texto (`TextField size`) en el que el usuario introduce el tamaño  $N$  de la lista que se va a ordenar.
2. Un botón (`Button createButton`) con el texto “Create” que, una vez pulsado, genera de forma aleatoria una lista de  $N$  números entre 0 y 99999 y muestra la lista en el área más a la izquierda.
3. Un área de texto (`TextArea area`) en donde se muestra la lista aleatoria generada tras pulsar el botón “Create”.
4. Un botón (`Button sortButton`) con el texto “Sort” que, una vez pulsado, ordena la lista creada utilizando los dos algoritmos (selección y burbuja).
5. Un botón (`Button cancelButton`) con el texto “Cancel” que, una vez pulsado, permite cancelar la ordenación de la lista.
6. Dos áreas de texto (`TextArea areaSelection`, `areaBubble`) en las que se muestran las listas ordenadas utilizando cada uno de los algoritmos.
7. Tres etiquetas (`Label messageArea`, `messageSelection`, `messageBubble`) debajo de las áreas de texto que se utilizan para mostrar el estado de las tareas que se están llevando a cabo.
8. Dos barras de progreso (`JProgressBar progressSelection`, `progressBubble`) debajo de las respectivas áreas de texto que se utilizan en la segunda parte del ejercicio para mostrar el progreso de cada ordenación.
9. Una etiqueta roja (`Label comment`) en la parte inferior de la GUI que permite dar información al usuario.

### Ejercicio 1 (0.5 puntos): GUI con SwingWorker con done

1. Incluye en tu proyecto la clase `Panel`, que presentará un aspecto similar al que se muestra en la figura.
2. Crea una clase `Controller` que implemente las interfaces necesarias para manejar los eventos de los botones “Create” y “Sort”.
3. Cuando el usuario pulsa el botón “Create” debe crearse una lista de números aleatorio del tamaño indicado en el campo de texto `size`. La lista se debe mostrar en el área de texto de la izquierda (llamada `area`), y debe actualizarse la etiqueta `message` con el mensaje “List created”. Si el tamaño introducido por el campo de texto no está en el rango `[1,60000]` debe borrarse el número e informar al usuario, a través de la etiqueta `comment`, de que el número introducido no es correcto. Cuando se crea una nueva lista, las áreas de texto `areaSelection` y `areaBubble` deben limpiarse.
4. Crea una clase `WorkerSelection` que dada una lista la ordene utilizando el algoritmo de ordenación por selección. La lista a ordenar se pasa como parámetro en el constructor y, en éste, **debe crearse una copia de la lista que es la que se ordena y se devuelve** (Nota: no debe ordenarse la lista de entrada).
5. Crea una clase `WorkerBubble` que dada una lista la ordene utilizando el algoritmo de la burbuja. La lista a ordenar se pasa como parámetro en el constructor y, en éste, **debe crearse una copia de la lista que es la que se ordena y se devuelve** (Nota: no debe ordenarse la lista de entrada).
6. Cuando el usuario pulsa el botón “Sort” la lista creada debe ser ordenada por los dos workers y posteriormente debe mostrarse el resultado en las áreas de texto correspondientes. Mientras que cada worker está ordenando la lista debe aparecer el mensaje “Sorting the list” debajo de cada área de texto. Cuando un worker ha terminado debe aparecer un mensaje del tipo “List sorted in xxx ms.” debajo de su área de texto. Para calcular el tiempo que se tarda en realizar la ordenación pueden utilizarse instrucciones del tipo:

```
long startTime = System.currentTimeMillis(); //inicio de la ordenación
.....
total= (System.currentTimeMillis()-startTime);//fin de la ordenación
```

7. Crea una clase principal que construya una hebra dispatcher desde la que se cree la GUI.

**Nota: En esta primera versión no es necesario controlar la barra de progreso, ni hacer uso del botón “Cancel”**

### Ejercicio 2 (0.3 puntos): GUI con SwingWorker con publish y barra de progreso.

En esta versión de la GUI se utiliza el hecho de que después de cada iteración del bucle más externo de cada uno de los algoritmos la lista `a[0..i]` ya está ordenada. Crea una nueva versión y rehúsa las clases `WorkerSelection` y `WorkerBubble` (recuerda que se entrega tres versiones, una por cada ejercicio).

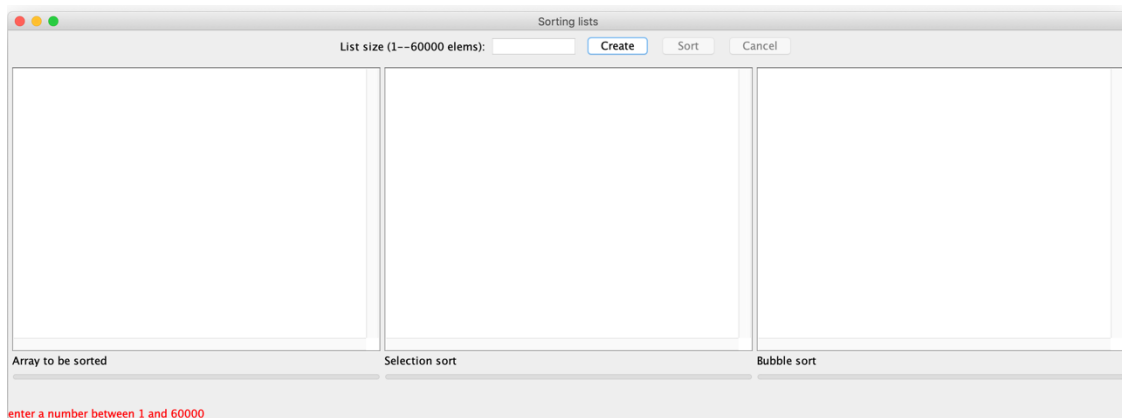
1. Modifica las clases `WorkerSelection` y `WorkerBubble` para que las listas ordenadas se vayan mostrando en las respectivas áreas de texto a medida que se van iterando los bucles más externos de los algoritmos. Además, los workers tienen que actualizar su progreso de manera que cada barra de progreso muestre el porcentaje de trabajo realizado por cada worker.
2. Modifica la clase `Controller` para que implemente las interfaces necesarias para manejar los eventos de cambio de progreso de los workers.

**Ejercicio 3 (0.2 puntos):** Realiza las modificaciones necesarias en la clase `Panel`, `Controller`, `WorkerSelection` y `WorkerBubble` para que

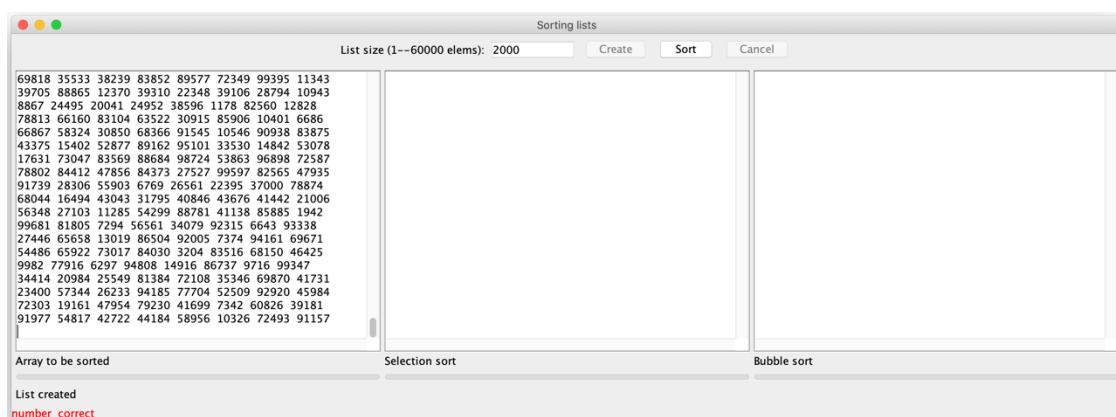
1. Sea posible cancelar la ordenación de los dos workers cuando el usuario pulsa el botón “Cancel”.
2. Sólo esté habilitado el botón “Create” si no se ha creado una lista.
3. Sólo esté habilitado el botón “Sort” si se ha creado la lista pero aún no se ha ordenado.
4. Sólo esté habilitado el botón “Cancel” mientras que las listas se están ordenando.
5. Después de haber ordenado la lista, o si se ha cancelado la ordenación, sólo debe estar habilitado el botón “Create”.

### Ejemplos de la vista en la primera parte del proyecto

1.- GUI después de pulsar el botón “Create” sin introducir ningún número



2.- GUI después de pulsar “Create” y de introducir el valor 2000



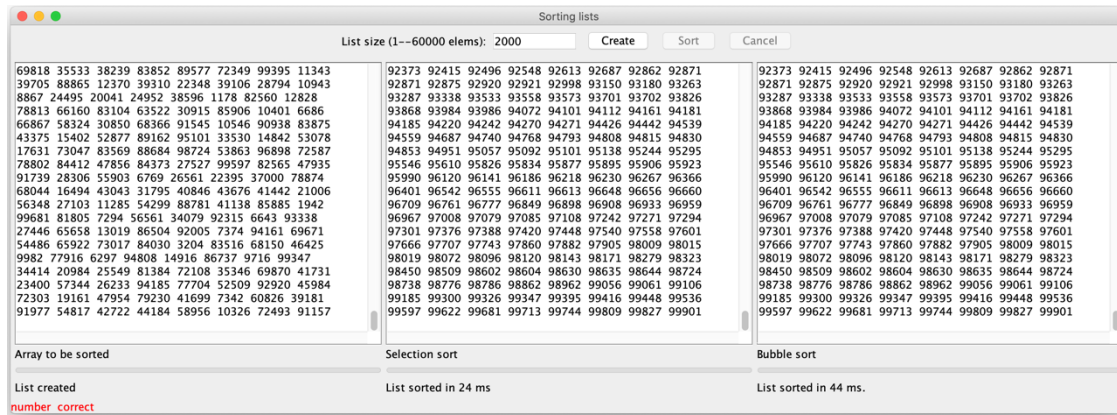


UNIVERSIDAD  
DE MÁLAGA



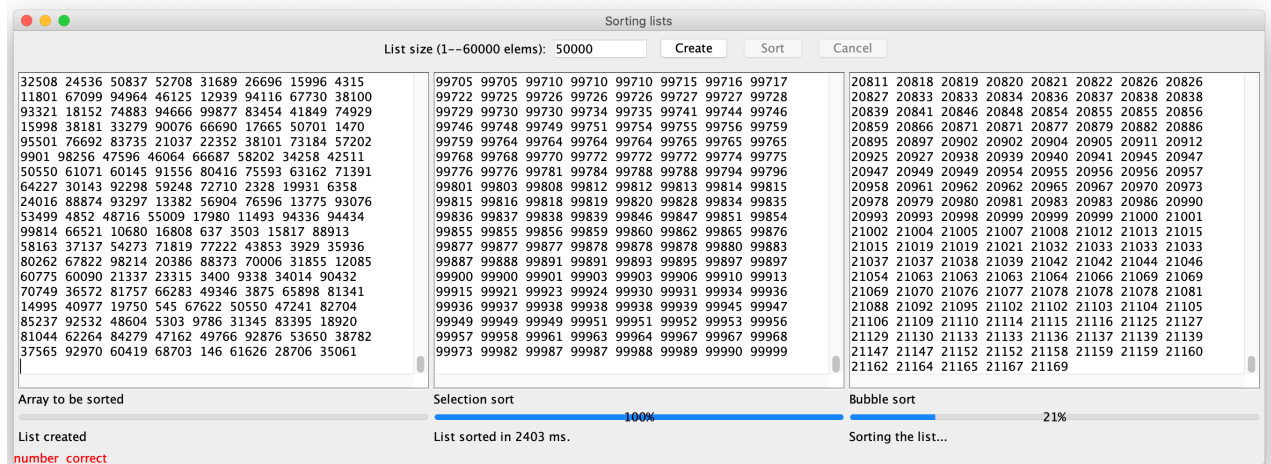
Dpto. Lenguajes y  
Ciencias de la Computación

### 3.- GUI después de pulsar “Sort”



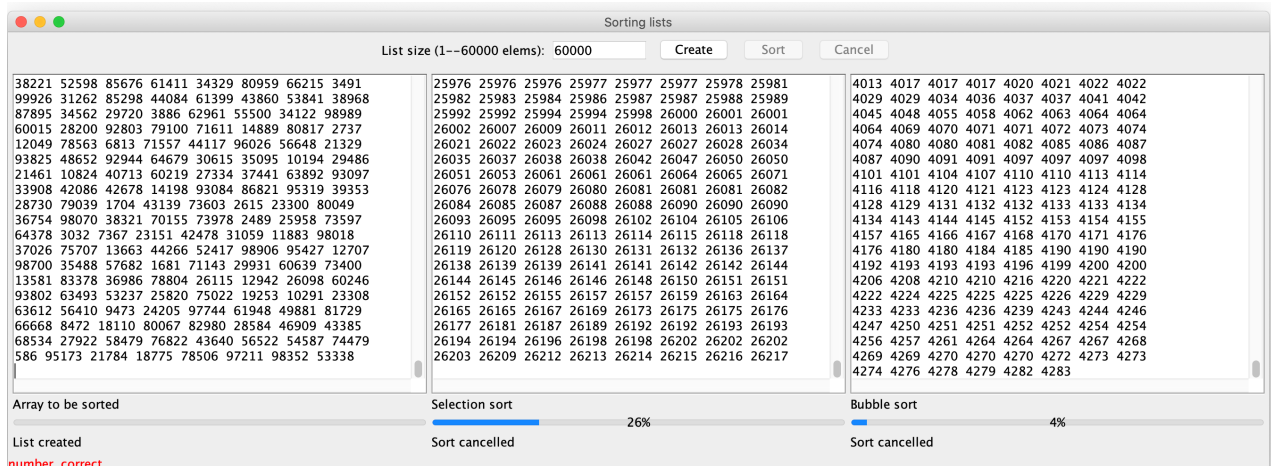
### Ejemplo de la vista en la segunda parte del proyecto:

#### 1.- GUI durante la ordenación de la lista



### Ejemplo de la vista en la tercera parte del proyecto:

#### 1. GUI después de haber cancelado la ordenación



Esta obra está sujeta a la licencia Reconocimiento-NoComercial-SinObraDerivada 4.0 Internacional de Creative Commons. Para ver una copia de esta licencia, visite <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

Autor: Profesores de la asignatura