

## Práctica 3. Java Swing. Paneles

**Asignatura: Diseño de Interfaces de Usuario**

### Resumen:

Diseño de una interfaz para mostrar los elementos de una matriz usando como elementos principales: *Slider*, *Text Area* y *Panels* proporcionados por la librería Swing.

Carlos Brito Pérez

Pablo Morales Gómez

Grado en: Ingeniería Informática. 4º Curso

## ÍNDICE

INTRODUCCIÓN .....	3
DESCRIPCIÓN.....	3
FOTO DE LA INTERFAZ .....	5
BIBLIOGRAFÍA .....	5

---

## INTRODUCCIÓN

En esta práctica utilizamos la IDE NetBeans para desarrollar una pequeña aplicación en la que podemos visualizar una matriz cuadrada (10x10) rellena de números generados aleatoriamente por un rango introducido por el usuario. La ventana hace uso de distintos elementos de la librería Swing para alcanzar el objetivo.

## DESCRIPCIÓN

La aplicación posee dos partes claramente delimitadas por dos *JPanels*, además cada una tiene un título explicativo que orienta al usuario sobre cual es el propósito de cada segmento. El primero se encuentra en la parte superior y es donde se visualiza la matriz en cuestión, está compuesta por un *JSlider* y un *JTextArea*. El segundo se usa para indicar cuál es el rango numérico entre los que se encuentran los números que están dentro del vector de dos dimensiones.

La barra deslizante tiene un papel protagonista dentro del funcionamiento del programa y por eso se encuentra situada en el centro de la ventana. Desplazándola de izquierda a derecha y viceversa se van mostrando u ocultando los valores de la matriz (tal como se indica en la práctica solo se muestran aquellos que sean mayores que el valor actual de la barra). Para mostrar cuál es el valor que posee el *JSlider* en todo momento hemos dispuesto un *JLabel* en el que se va mostrando constantemente (se aprovecha el evento que se lanza cada vez que cambia de estado). Por otro lado, también hemos añadido unos *Ticks* equiespaciados para que el cliente tenga una noción de a que altura se encuentra la barra. Decidimos no incluir el valor que representa cada *Tick*, porque ya aparece en la ya mencionada etiqueta y porque cuando el rango de los números es muy amplio tienden a desajustarse y superponerse.

El *JTextArea* es el elemento donde mostramos la matriz, cada valor está separado por barras verticales consiguiendo así facilitar la diferenciación entre las cifras. Aseguramos que las barras siempre queden alineadas ajustando el espacio entre ellas mediante el cálculo del número máximo de caracteres que puede ocupar un número. Gracias a que usamos una tipografía monoespaciada podemos calcular el número de espacios que se deben añadir para estandarizar el espacio ocupado por cada número. El cálculo antes mencionado es una simple diferencia entre el número máximo posible de caracteres y el número de caracteres que posee la cifra.

Para la comodidad de los usuarios hemos añadidos algunas características al *JTextArea*. No es editable (como se indica en el guion de prácticas) y las dimensiones están ajustadas para intentar reducir la necesidad de desplazamiento vertical y conseguir así que navegar por el área de texto sea menos engorroso para cualquier tipo de usuario (limitando el desplazamiento para que sea horizontal si el tamaño de las cifras así lo requiriera).

Para escoger el rango numérico disponemos de dos *JTextField*s que representan el límite superior e inferior, y así aparece indicado. Como el programa carece de botones cada vez que se introduce o se retira un carácter en cualquiera de ellos, se lanza un evento. Este se encarga primero, de comprobar que no se hayan introducido letras o signos no permitidos, y si así fuera se le indicaría al usuario con un mensaje de aviso. A continuación, comprobamos que el mínimo introducido no sea mayor que el máximo y viceversa, para evitar cualquier problema con la ejecución del código y para que exista una coherencia en la interacción del cibernauta con la interfaz. Si fuera mayor se le hará saber al usuario mediante otro mensaje de aviso como debe proceder. Cuando se han completado las comprobaciones se reajustará el rango de valores y se recalcularán un nuevo conjunto de datos para la matriz además de los pertinentes ajustes en la barra deslizante.

Por último, cada uno de los elementos principales mencionados posee un *ToolTipText*, con esto conseguimos que en caso de duda el cliente pueda mantener el ratón sobre cualquiera de los componentes y recibir indicaciones sobre cuál es su uso.

## FOTO DE LA INTERFAZ

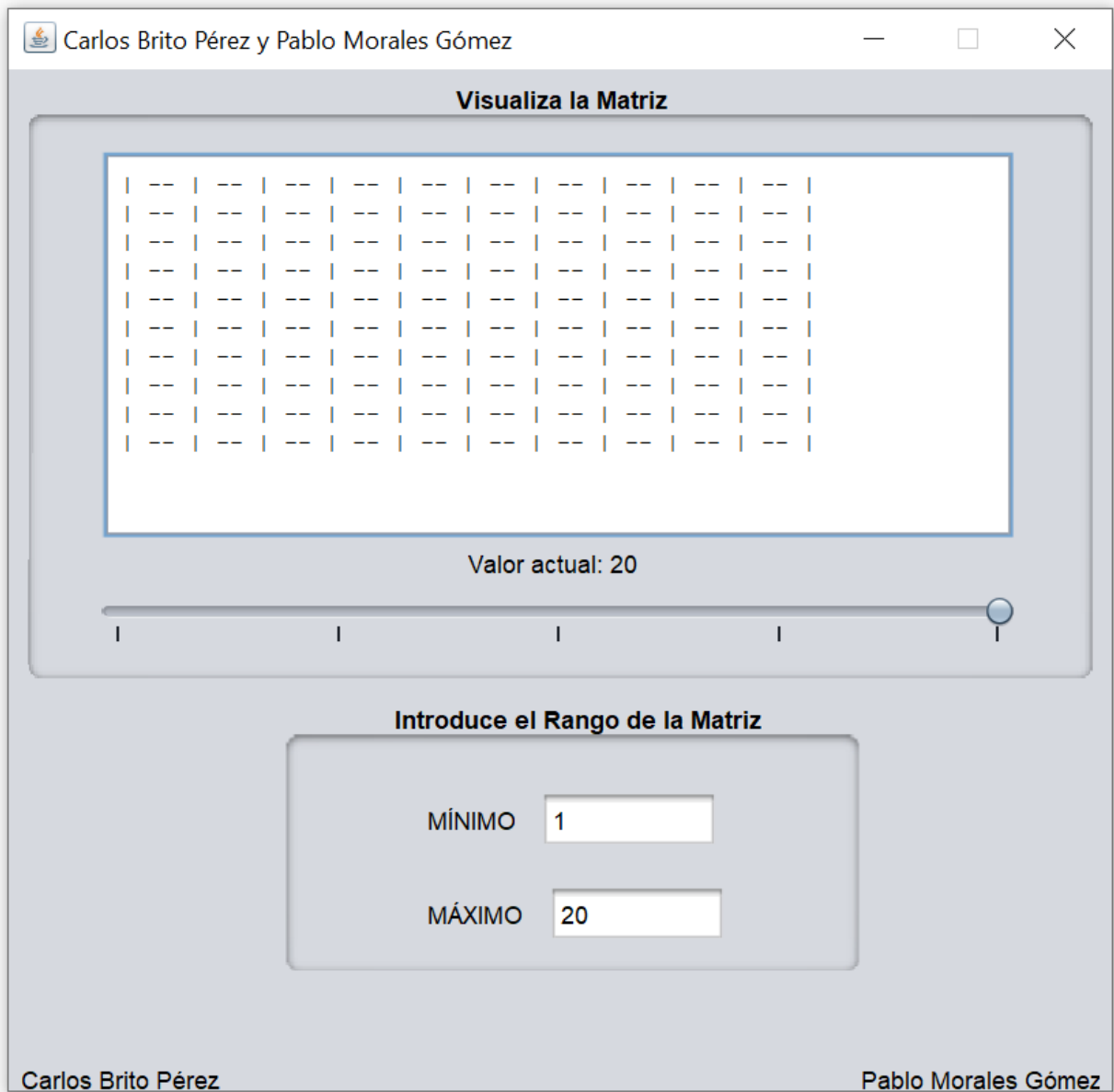


Foto de la interfaz descrita en la memoria.

## BIBLIOGRAFÍA

- Guion de Prácticas
- Vídeo Explicativo sobre los Elementos Swing para la Práctica 3