

Tarea 6

Interacción Persona - Computador

Álvaro Alcántara Ruiz de Garibay, María Isabel de la Osa Rocha,
Noé Ruano Gutiérrez, Pablo Sáez Alonso

Noviembre 2021

1. Crear nuevos Documentos, guardarlos y abrirlos.

A través del grupo de iconos situado en el extremo izquierdo de la barra de herramientas superior, las opciones del menú desplegable ‘Archivo’, o los atajos de teclado *Ctrl+N*, *Ctrl+S*, *Ctrl+Shift+S*, es posible crear nuevos documentos, guardarlos y abrirlos, respectivamente.

Para crear un nuevo documento, se abre un editor vacío sobre el que podrá trabajar el usuario.

Al guardar un documento, este recibe un nombre por defecto, y lo reescribe en los sucesivos guardados.

Al acceder a la utilidad de “Guardar como...”, se muestra un JFileChooser con el que el usuario podrá indicar al editor tanto la ruta como el nombre del documento, dentro del sistema de ficheros.

2. Cortar/Copiar/Pegar texto.

A través del grupo de iconos situado en segunda posición desde la izquierda, en la barra de herramientas superior, o bien a través de las opciones del menú desplegable ‘Archivo’, o los atajos de teclado *Ctrl+X*, *Ctrl+C*, *Ctrl+V* es posible cortar, copiar o pegar una selección de texto, respectivamente.

Las selecciones de texto que se realizan sobre el TextArea, se guardan en un string “clipboard”.

3. Alinear Texto

En el extremo derecho de la JToolBar de la parte superior, se encuentran los 4 botones correspondientes a los alineamientos de texto, siendo estos, Alinear a la izquierda, Centrar, Alinear a la derecha y Justificar.

Esto se consigue mediante el uso de SimpleAttributeSets, a los cuales se les asocia cada una de las alineaciones deseadas (StyleConstants.ALIGN_LEFT, StyleConstants.ALIGN_CENTER, StyleConstants.ALIGN_RIGHT, StyleConstants.ALIGN_JUSTIFIED).

Posteriormente, y para que el alineamiento se lleve a cabo se utiliza un setParagraphAttributes con el documento correspondiente al textArea.

4. Especificar tamaño, Color y Tipo de Fuente

En la JToolBar de la parte superior, se podrán editar distintos aspectos de un texto seleccionado, así como copiar, cortar y pegar.

Si deseamos cambiar el color del texto seleccionado, debemos darle al botón de la barra de herramientas que contiene un círculo con la escala cromática. Se

desplegará un objeto JColorChooser que nos permitirá seleccionar el color que el usuario desee.

Cambiar de tamaño y fuente se implementa de forma similar: un JComboBox desplegará las distintas opciones de tamaño y fuente y permitirá seleccionar la deseada, esta elección afectará solo al texto seleccionado.

5. Resaltar Texto en Negrita, Cursiva y Subrayado.

De manera similar a como ocurre con el alineamiento del texto, las tres formas de resaltar el texto también se basan en acceder a los estilos del texto definiendo distintos SimpleAttributeSet que lleven a cabo las modificaciones deseadas.

Se calcula el texto destacado y su longitud mediante

```
inicioSeleccion = textArea.getSelectionStart();
```

```
finSeleccion = textArea.getSelectionEnd();
```

```
longitudSeleccion = finSeleccion - inicioSeleccion;
```

Posteriormente se crea el atributo de estilo deseado (StyleConstants.setBold / StyleConstants.setItalic / StyleConstants.setUnderline) y se adjudica este atributo al texto seleccionado haciendo uso de doc.setCharacterAttributes.

A cada uno de estos atributos se le ha adjudicado un atajo de teclado (Ctrl+B, Ctrl+I, Ctrl+U, respectivamente).

Además, siguiendo las indicaciones sobre las mejoras en la entrevista realizada, se ha hecho una gestión de Resalto. Esto hace que cuando un texto ya se encuentre resaltado de alguna manera, al volver a seleccionar el resalto que le afecta, el resalto se elimina. (Un ejemplo es, si un texto está en Negrita y se selecciona Negrita de nuevo, el texto pierde la Negrita).

Además, también se han incluido botones en la JToolBar superior de manera que la edición del texto sea de fácil acceso en caso de no conocer los atajos de teclado.

6. Búsqueda y reemplazo

El usuario puede realizar una búsqueda en el text area introduciendo el conjunto de caracteres que quiere buscar en el text field para buscar y presionando la tecla enter. El conjunto de caracteres que el usuario desea buscar se encuentra en una variable de tipo string "busqueda". Si en el text area aparece esa palabra, la misma aparece en amarillo en el text area. Esto se ha conseguido mediante el uso de las clases Highlighter y HighlightPainter.

Por otro lado, cuando el usuario busca un conjunto de caracteres en el text field para buscar, puede luego introducir otro conjunto de caracteres en el text field para reemplazar y presionar enter. De esta forma, se produce un reemplazo de la palabra buscada a la nueva introducida en el text field para reemplazo. Esto se

ha conseguido metiendo el texto del text area en un string y haciendo uso del método replaceAll que aportan los strings.

7. Proceso de trabajo

Partimos de un boceto realizado en Paint. A partir de ahí, dividimos el trabajo en dos grandes tareas: diseño de la interfaz, y funcionalidades. Subdividimos esos dos grupos en subtareas que nos repartimos y realizamos individualmente. Una vez completados todos los desarrollos individuales, juntamos todas las partes.

Con el editor montado y ejecutando, realizamos la “entrevista con los clientes” y aplicamos sus sugerencias al producto final.

Porcentaje de distribución del trabajo

- Álvaro Alcántara: 25%
- Noé Ruano Gutiérrez: 25%
- María Isabel De La Osa: 25%
- Pablo Sáez Alonso: 25%