



Proceso de elaboración: Componente en Swing

SAMUEL SANCHEZ & ALAN MIKOLAJCZYK

16/11/2023

I.E.S. FRANCISCO DE QUEVEDO

INDICE

INTRODUCCIÓN	2
PROCESO DE ELABORACIÓN	3
DECISIONES DE DISEÑO	4
CONCLUSIÓN	5

INTRODUCCIÓN

El proceso de elaboración de componentes en Swing es esencial para entender cómo se estructuran y organizan los elementos visuales en una aplicación Java. Un componente en Swing puede ser cualquier objeto visual, desde botones y campos de texto hasta paneles y marcos completos. La construcción efectiva de estos componentes implica comprender la jerarquía de contenedores y componentes, así como la aplicación de propiedades y comportamientos específicos, lo que se verá con posterioridad.

En este trabajo examinaremos cómo se diseña y desarrolla un componente en Swing. En este caso, crearemos un “date picker”. Precisaremos los conceptos clave, la disposición del componente y la gestión de errores, fundamentales para la exitosa creación de componentes que sean atractivos para cualquier usuario de estos.

PROCESO DE ELABORACIÓN

La clase inicialmente contempló diversas opciones para determinar la estructura más apropiada para el componente 'Fecha', buscando una uniformidad en su implementación. Se consideró la posibilidad de consolidar todas las funcionalidades en una única clase. Sin embargo, tras un análisis detallado, se optó por dividir el desarrollo en diferentes carpetas, cada una con sus respectivas responsabilidades.

En el proceso de distribución de tareas, uno de los integrantes se encargó de participar activamente en las discusiones (aunque ha habido más participantes), contribuyendo a la concepción del componente y delineando sus características esenciales. Mientras tanto, el otro miembro asumió la responsabilidad de llevar a la práctica la visión acordada, implementando el componente y demostrando su funcionalidad en una presentación concreta. Al haber terminado el primer diseño, se ha puesto a prueba para posibles errores de excepción.

Posteriormente, el otro miembro continuó con la labor de documentar todo lo hecho y terminar el trabajo de la demo del componente (haciendo así las pruebas pertinentes), al igual que hacer las pruebas necesarias para verificar la correcta funcionalidad de la demo.

Esta decisión estratégica permitió una colaboración efectiva, como en el trabajo anterior, donde ambas partes desempeñaron roles complementarios para lograr un desarrollo integral y coherente del componente 'Fecha', al igual que la demo, implementando este componente. Este enfoque demostró ser eficaz al aprovechar las fortalezas individuales de los miembros del equipo, garantizando así la calidad y cohesión del trabajo final, así como la rapidez en el desarrollo del trabajo.

DECISIONES DE DISEÑO

Al determinar la metodología de diseño, elegimos una aproximación sencilla que consistió en utilizar dos cuadros desplegables (comboBox), uno para el día y otro para el mes, junto con un campo de texto (textField) para el año. Al tener el componente visual creado seguimos con la incorporación de la interfaz, que tendría dos funcionalidades:

- 1- `getDate()` -> Aquí devuelve un `LocalDate` y recoge los datos de los dos comboBox y del `TextField`, y estos elementos los transforma en un `LocalDate`.
- 2- `setDate()` -> Sirve para posibles funcionalidades de poner una fecha predeterminada.

A continuación, seguimos con la implementación de tres excepciones:

- 1- `FechaIncompletaException`: Excepción que se lanza cuando no hay definido un día, mes o año.
- 2- `FechaIncorrectaException`: Excepción que se lanza cuando una fecha es incorrecta, por ejemplo, "30 de febrero".
- 3- `FechaImposibleException`: Excepción que se lanza cuando es imposible la fecha, por ejemplo, "29 de febrero de 2023".

Después, entramos en la fase de prueba de funcionalidad (demo). Incorporamos dos instancias del componente de fecha en una demostración, acompañadas de un botón diseñado para validar el correcto uso de cada uno de los elementos. Posteriormente, comenzamos con la funcionalidad del botón. Primero, tuvimos en cuenta las posibles excepciones que nos podría dar el componente al haberlo hecho. Tras ello, continuamos con el control de errores de la funcionalidad de la demo, y, finalmente que nos dé el resultado.

Este enfoque minimalista no sólo simplificó la interfaz para el usuario, haciendo la interacción más intuitiva, sino que también facilitó la verificación de la eficacia y coherencia del componente. La inclusión de un botón de validación en la demostración permitió asegurarnos de que cada elemento del componente "fecha" estuviera siendo utilizado de manera adecuada, garantizando así la robustez y fiabilidad del diseño implementado.

CONCLUSIÓN

En definitiva, de nuestra exploración del proceso de elaboración del componente 'Fecha' en Swing destacamos la importancia de comprender la jerarquía de contenedores y componentes en el diseño de interfaces gráficas en Java. Optamos por una estructura dividida en carpetas, asignando responsabilidades específicas para lograr una implementación uniforme y coherente.

El trabajo colaborativo resultó ser clave, dividiendo tareas entre participantes, desde la concepción hasta la implementación práctica. La estrategia de dividir roles permitió aprovechar las fortalezas individuales y agilizar el desarrollo del componente 'Fecha' y su demostración.

En cuanto a las decisiones de diseño, elegimos una aproximación sencilla, utilizando comboBox para día y mes y un textField para el año. Integramos funcionalidades como getDate() y setDate(), junto con la implementación de excepciones para manejar situaciones específicas.

Finalmente, la fase de prueba reveló la eficacia del diseño minimalista, simplificando la interfaz para el usuario y facilitando la validación de la funcionalidad. La inclusión de un botón de validación en la demo garantizó el correcto uso de cada elemento del componente 'Fecha', asegurando robustez y fiabilidad en el diseño implementado.