
PROYECTO CACULADORA V2

by

Jonathan Carrero

Abstract. — Pasamos a una nueva versión del proyecto en el que se introducen operaciones con distintas bases y algunas funcionalidades adicionales. Esto hace que tengamos que introducir nueva lógica en el programa y modificar alguna de la que ya teníamos.

1. Funcionamiento

La calculadora debe permitir realizar operaciones con números negativos únicamente en base decimal. Para simplificar las cosas, haremos que sólo se realicen operaciones con números positivos para el resto de bases. Al comenzar, el usuario debe elegir la base en la que quiere operar, al igual que cuando resetea la calculadora.

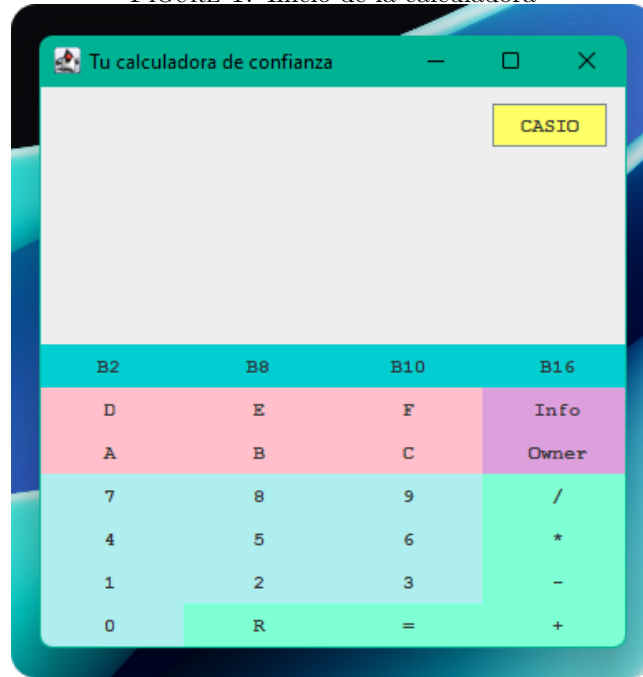
2. Detalles de implementación

La clase *Main* no cambia. En cuanto a la clase *Engine*, debes añadir algunos atributos adicionales tales como los nuevos botones, paneles necesarios para la nueva estructura, un atributo en el que se amanece la base numérica, nuevos tipos de botón en el `enum`, etc. Como comprenderás, todos estos atributos deben ser instanciados en la constructora, siguiendo la misma lógica que teníamos hasta ahora.

A continuación, vamos a echar un vistazo al resultado final de lo que debería ser la ventana (ver Figura 1) y la nueva estructura de paneles. El panel donde se muestra el display de las operaciones, ahora pasa a ser el panel central de la calculadora. Al mismo tiempo, se crea un nuevo panel norte que, a su vez, contiene dos sub-paneles (izquierdo y derecho). En el panel izquierdo aparece

la base con la que estamos operando, y en el panel derecho aparece la marca de la calculadora (ver Figura 2).

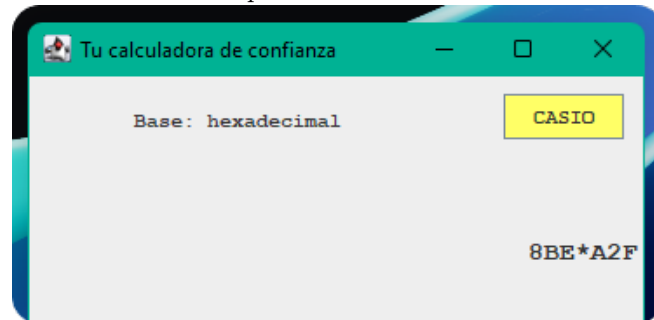
FIGURE 1. Inicio de la calculadora



Con el objetivo de que quede lo mejor posible y a tu gusto, tienes total libertad para poner los layouts que consideres en esta nueva distribución de paneles, siempre y cuando aquello se parezca a una calculadora y aparezcan los elementos que se piden.

Como puedes ver en las imágenes, cuando se inicia la calculadora no aparece ninguna base. Por lo que en esta segunda versión, lo primero que debemos hacer antes de realizar cualquier cálculo es elegir la base en la que queremos operar. Como ya se ha mencionado, en el lado izquierdo del panel norte aparecerá un mensaje de la base con la que estamos trabajando, que cambiará únicamente cuando pulsemos alguna otra base o se reinicie la calculadora. De hecho, cuando pulsamos la letra R (para lo cual ya se presupone que la calculadora está encendida), lo que aparece es un mensaje que nos pide que, de nuevo, antes de operar debemos elegir una base (ver Figura 3). Nótese que este mensaje aparece únicamente cuando la calculadora ya está encendida y esta ha sido reiniciada.

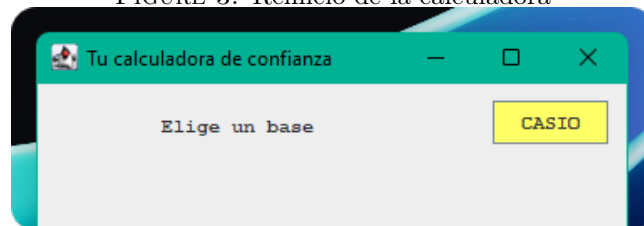
FIGURE 2. Operando en base hexadecimal



El recuadro donde aparece `CASIO` puede ser casi cualquier tipo de elemento, ya que existen diferentes implementaciones. En mi caso, es un botón. Debes buscar la manera de añadir un `ActionListener` para que cuando este botón sea pulsado, te lleve a la página web de las calculadoras de la marca `CASIO`, donde hay calculadoras realmente buenas que duran para toda la vida (aunque no tan buenas como la nuestra, claro).

Una funcionalidad que también conviene destacar son los pop-ups que aparecen cuando pulsamos sobre el botón `Info` y `Owner`. Esta funcionalidad debe implementarse en otra clase llamada `VentanaEmergente`, la cual recibe el mensaje que debe mostrar la nueva ventana. En el caso de la Figura 4 podemos observar qué ocurre cuando pulsamos sobre el botón `Owner`.

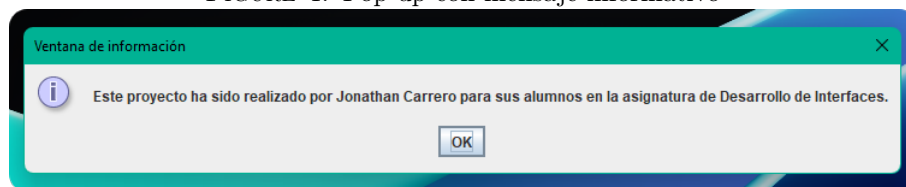
FIGURE 3. Reinicio de la calculadora



Pasemos a profundizar un poco más sobre la lógica de los cambios de base. Una idea que se nos podría ocurrir es la de implementar esta lógica en el método `operation()`. Por ejemplo, podríamos coger el valor de los atributos `this.num1` y `this.num2` y operarlos según la base en la que nos encontremos. El problema de esto es que se supone que `operation()` únicamente realiza las operaciones. Es decir: coge los atributos y realiza la operación asignada (sumar, restar, multiplicar o dividir) retornando un resultado y actualizando

el display con dicho resultado. Por esta razón, y si se supone que esta es la funcionalidad de `operation()`, no conviene implementar cambios de base en este método. ¡Debes respetar la lógica del programa! Eso quiere decir que si tu programa crece, debes seguir respetando lo que hacen los métodos y, si es necesario implementar una nueva lógica, pues lo más correcto sería implementar esa lógica en otro sitio (en otro método, por ejemplo). Tenemos que tener muy claro lo que hace cada método y no mezclar la lógica de cada uno de ellos, o nuestro programa rápidamente se convertirá en algo caótico, sin sentido y totalmente ineficiente. Por todo esto, vuelvo a recordar, debes implementar la lógica de los cambios de base en otro sitio y, cuando hayas hecho esto, entonces ya puedes llamar al método `operation()` tranquilamente para que realice el cálculo y actualice el resultado.

FIGURE 4. Pop-up con mensaje informativo



Por último, vamos a ver algunas consideraciones a tener en cuenta y errores que no debes controlar para hacer más sencilla la implementación.

- Vamos a establecer que únicamente se puedan hacer operaciones (sumar, restar, multiplicar o dividir) con números positivos para cualquier base que no sea la base decimal.
- No es necesario que controles comportamientos inadecuados por parte del usuario, como por ejemplo que en mitad de una operación cambie de base, o que si estás operando en base binaria de repente introduzca una letra B o un número 2, etc.
- A modo de consejo, cuando tengas las operaciones con bases, puedes buscar en Internet calculadoras de bases y comprobar si los resultados de tu calculadora son correctos (ver [1, 3, 2]).

References

- [1] MINIWEBTOOL, *Calculadora binaria*. <https://miniwebtool.com/es/binary-calculator>. [Online; accessed 31-October-2023].

- [2] ———, *Calculadora hexadecimal*. <https://miniwebtool.com/es/hex-calculator>. [Online; accessed 31-October-2023].
- [3] ———, *Calculadora octal*. <https://miniwebtool.com/es/octal-calculator>. [Online; accessed 31-October-2023].

November 27, 2023

J.C., Colegio Litterator • *E-mail* : jonathan.carrero@colegiolitterator.com