

Aplicación grafica para la ejecución de comandos

19/09/2022 | Versión 1.0

Joaquín Fernández | I.E.S Francisco de Quevedo 2º Desarrollo de interfaces 2022-2023

Índice

<i>Introducción</i>	2
<i>Requisitos que debería cumplir</i>	2
<i>Dibujo de la apariencia de la aplicación</i>	3
<i>Proceso de ejecución y uso</i>	4
<i>Plan de evaluación de la usabilidad.....</i>	5

Introducción

El objetivo de esta aplicación es una calculadora siguiendo el estilo de creación de scripts en PowerShell. El usuario introducirá una operación matemática que la aplicación calculará y mostrará tanto el proceso como el resultado para que el usuario pueda comprender todo el curso de la operación y se puedan corregir los errores que suceden cuando calculamos sobre papel.

A lo que aspira esta aplicación, es que los profesores usen esta aplicación para corregir los procesos de manera más cómoda o que incluso los alumnos puedan corregir sus ejercicios teniendo el proceso para saber dónde se equivocaron y no tener que adivinar con el único resultado final que la calculadora normal proporciona.

En los próximos capítulos voy a exponer cuales son los requisitos que la aplicación debería cumplir para que funcione de manera esperada, la apariencia que debería proporcionar la aplicación para que sea cómoda para el usuario y por último un pequeño resumen de cómo va a funcionar y un corto manual para su uso.

Requisitos que debería cumplir

La aplicación debería ser capaz de ejecutar cálculos matemáticos sencillos, deberá mostrar el resultado con el proceso en la parte inferior de la ventana y darle al usuario la opción de importar tanto el cálculo, mostrando solo el dato introducido y el resultado, o todo el proceso del cálculo.

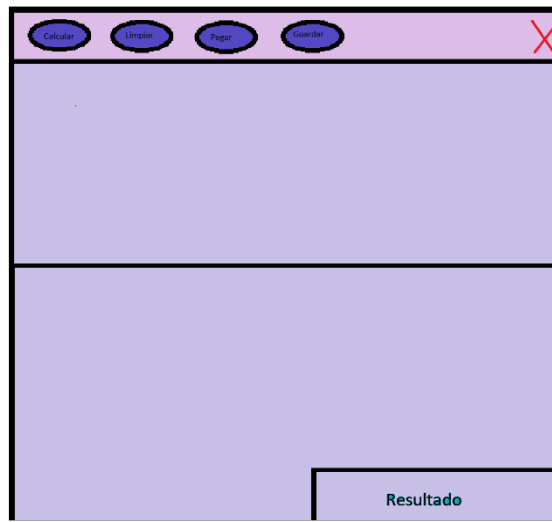
También debería tener diferentes botones para que el usuario pueda guardar ese resultado completo, copiarlo y limpiar la pantalla.

Siguiendo los principios de diseño, he juntado los más útiles e importantes a mi parecer para esta aplicación en concreto:

- Que la aplicación sea fácil de entender y usar para el usuario. Para esto he intentado usar la mínima cantidad de botones con el motivo de que la aplicación sea sencilla a la vista.
- Evitar la distracción del usuario. Siguiendo los pasos del punto anterior, la idea es tener un mínimo número de botones y funcionalidad para que el usuario no se sienta abrumado.
- Ayudar a recorrer la aplicación desde que se abre hasta que se cierra. En cada botón o caja de texto, se indica al usuario para qué sirve y qué hace cada uno.
- Permitir al usuario decidir cómo y cuándo interactuar con la interfaz, aportando descripciones junto a las interacciones. También juntando con el punto anterior, hay algunas cajas a las que el usuario no puede acceder hasta que ponga una operación, para ello, las que no pueda usar, aparecerán como bloqueadas.

Dibujo de la apariencia de la aplicación

Para el prototipado estático de esta aplicación quería usar colores sencillos y claros como un rosa apagado (#C8BFE7) para los fondos tanto de la introducción del usuario como el del resultado y el proceso, para la barra del menú he usado un color más rosa (#DDBDE7) y uno más morado para los botones (#55448C2). Quería que la aplicación tuviera un tono relajado y que ayudará a su uso con colores poco llamativos para que sea amable a la vista.



Esa imagen representa la idea principal de la interfaz de la aplicación con los colores deseados a usar, es una interfaz sencilla con cuatro cajas, la primera contiene todos los botones a los que el usuario tiene acceso para usar la aplicación, la segunda sirve para exponer la operación que se va a calcular, la tercera es en la que se muestra todo el proceso que ha hecho la calculadora para llegar hasta el resultado que se mostrará finalmente en la última caja.

Para el modelo dinámico comenzaremos a comprobar cómo funciona la aplicación y la corrección de errores. Y para el modelo robusto ya terminaremos de corregir los errores de la misma y tendremos una aplicación hasta cierto punto funcional.

Proceso de ejecución y uso

El funcionamiento se basa en leer la operación proporcionada por el usuario y seguir la jerarquía de operaciones (paréntesis, raíces y potencias, multiplicaciones y divisiones, sumas y restas, todo esto de izquierda a derecha). El resultado será proporcionado en decimales por lo que si sale un número periódico mostrará solo los primeros seis caracteres y el último lo redondeará.

El programa podría mostrar un error cuando el usuario se haya equivocado en la operación y haya escrito alguna letra o algún carácter no válido a la hora de calcular.

[1]Por lo que las respuestas de la aplicación pueden ser:

- Error de Sintaxis. Este error aparecerá en el caso de escribir mal una expresión matemática o si hay algún problema con el formato del modo de cálculo.
- Error de Cálculo. El error matemático significa que tu operación no puede calcularse porque supera el rango de cálculo permitido.
- Stack Error. Este error indica que la operación escrita tiene más operadores (ejecuciones de cálculo o comandos) de los permitidos.
- Argument Error. En el caso de haber alguna incoherencia en el argumento de cálculo aparecerá este aviso.
- El resultado. Esto aparecerá cuando ninguno de los errores anteriores suceda, mostrará la resolución del cálculo.

El programa se iniciará cuando en la caja de operaciones haya algún tipo de cálculo matemático y el usuario pulse la tecla de iniciar, esto emergerá una ventana cuando se haya completado la operación y junto a eso se mostrará el cálculo completo terminado. Si el usuario expone un tipo de cálculo que no se puede comprobar o terminar, en la ventana emergente saldrá que hay algún dato erróneo o que no se puede ejecutar.

Este programa también dispondrá de unos botones que permitirán al usuario copiar el resultado, guardar o exportar ese resultado o borrar los datos que se encuentran en pantalla.

Plan de evaluación de la usabilidad

Para empezar, necesitamos saber que son los usuarios y sobre cual vamos a trabajar. El usuario es quien interactúa directamente con el software, puede ser una persona real o una simulación de la máquina. Se define por tres dimensiones, el conocimiento del dominio (área de especialización relacionada con el entorno de la aplicación), la experiencia con la tecnología (experiencia del usuario utilizando la tecnología general) y la experiencia con el sistema (la experiencia del usuario usando el software que está siendo desarrollado). De todas estas puede ser novato/ignorante o experto.

Las tareas que evaluaremos en estas pruebas son las consideradas como historias de usuario, que definen todos, o la mayoría de, los casos que se le pueden presentar al usuario a la hora del funcionamiento.

La observación, al ser una aplicación sencilla tendría un periodo de observación de una semana. Trabajaríamos sobre un sistema simulado, las pruebas las harían varios usuarios reales, como mínimo habría que comprender seis usuarios, para tener la máxima capacidad de pruebas posibles, los observadores no intervendrían, pero estarían estudiando las pruebas de los usuarios. Al final de cada día, se sometería a los usuarios a entrevistas para recopilar más información, no solo de las pruebas y el funcionamiento, sino también de la apariencia de la aplicación, la capacidad de respuesta y la comodidad de la misma.

Hay que respetar y valorar a los usuarios-test, no hay que someterlos a presión sin razón aparente, tienen que saber sobre que trata el estudio al que se les esta sometiendo, dejarle las tareas claras para no confundirles y terminar las pruebas cuando el usuario lo pida, ya que puede causar problemas psicológicos al mismo.

Llevar un recuento del tiempo que se ha tardado en completar este plan de evaluación, de las tareas que se han ido probando, aunque no hayan podido ser finalizadas o comenzadas, el porcentaje de error que el software proporciona y el tiempo que el equipo toma para reparar ciertos errores.

Referencias

Errores Calculadora. Usado para tener en cuenta que errores puede dar la calculadora. [1]