Texto

Descripción generada automáticamente con confianza media

Jueves 16 de marzo del 2023

Calculadora empleando métodos de pilas

 Captura de pantalla de un celular

Descripción generada automáticamente

Autores:

María Meraz Inostrosa

Martin Alejandro Pozos Arreola

Mauricio Yáñez Barrera

Instituto Tecnológico Autónomo de México (ITAM)Logotipo, nombre de la empresa

Descripción generada automáticamente

Índice

**Portada1**

**Descripción del problema3**

Objetivo3

Requisitos3

**Solución diseñada4-5**

**Pruebas5-6**

**Limitaciones de la solución7**

**Posibles mejoras y conclusiones7-8**

**Bibliografías y referencias8**

**Apéndice del código** **8**

* **Descripción del problema: (objetivo, requisitos)**

**Objetivo**:

El objetivo de este proyecto es crear una calculadora empleando el uso de pilas y de interfaz gráfica para crear el diseño de la calculadora. Esta calculadora será capaz de realizar operaciones básicas como: suma, resta, multiplicación, división, obtener el inverso de un número, etc.; además la calculadora usará el punto decimal, podrá borrar elemento por elemento o incluso borrará toda la operación ingresada al oprimir un botón.

La calculadora evaluará operaciones aritméticas usando paréntesis “()”, con el fin de checar si la operación esta equilibrada (es decir, debe tener la misma cantidad de paréntesis que abren y paréntesis que cierran), en caso de que la operación ingresada no este equilibrada, mandará un mensaje de error ya que no se ingresó de manera correcta la operación.

**Requisitos:**

La calculadora debe ser capaz de manejar números negativos y positivos, debe realizar las operaciones aritméticas básicas (suma, resta, multiplicación, división), debe leer de forma correcta el punto decimal y poder trabajar sin ningún inconveniente, realizando todas las operaciones ingresadas por él usuario.

**Restricciones:**

Las restricciones que se presentan para este proyecto son:

* Se debe de usar la aplicación Java NetBeans.
* Se deben usar los métodos de pilas.
* Se debe crear una interfaz gráfica “agradable” para el usuario.
* La calculadora debe ser capaz de obtener los resultados a partir de las operaciones ingresadas.
* La calculadora deberá mostrar los resultados correctos en la pantalla de la interfaz gráfica del programa realizado.
* **Solución diseñada (UML de clases, algoritmos principales)**

|  |
| --- |
| **Calcu** |
| memoria1: String  memoria2: String  signo: String |
| +Calcu()  +resta(pantallatxt)  +suma(pantallatxt)  +multiplicacion(pantallatxt)  +division(pantallatxt)  +borrar()  +borrarTodo()  +inversa()  +calcula(String memoria1,String memoria2, String signo): String  +existepunto(String cadena): boolean |

|  |
| --- |
| **PilaA** |
| -colect: T  -tope:int  -MAX: int |
| +PilaA()  +PilaA(int max)  +push(T dato):void  +pop():T  +peek():T  +isEmpty():Boolean  +aumentaCapacidad():void |

|  |
| --- |
| **PilaADT** |
| +pop():T  +push(T dato):T  +peek():T  +isEmpty():boolean |

|  |
| --- |
| **Posfijo** |
| -cadena:String |
| +Posfijo()  +Posfijo(String cadena)  +obtieneTokens():String[]  +conviertePostfija():String[]  -noEsOperador(String dato):boolean  -prioridad(String dato):int |
| **Calculadora** |
| -resultado:double |
| +Calculadora()  +getResultado():double  +calcula(String entrada):double  -revisa(String entrada):boolean  -obtieneTokens[](String entrada):String  -conviertePostfija[](String entrada):String  -noEsOperador(String dato):boolean  -prioridad(String dato):int  -evalúa(String postfija[]):double |

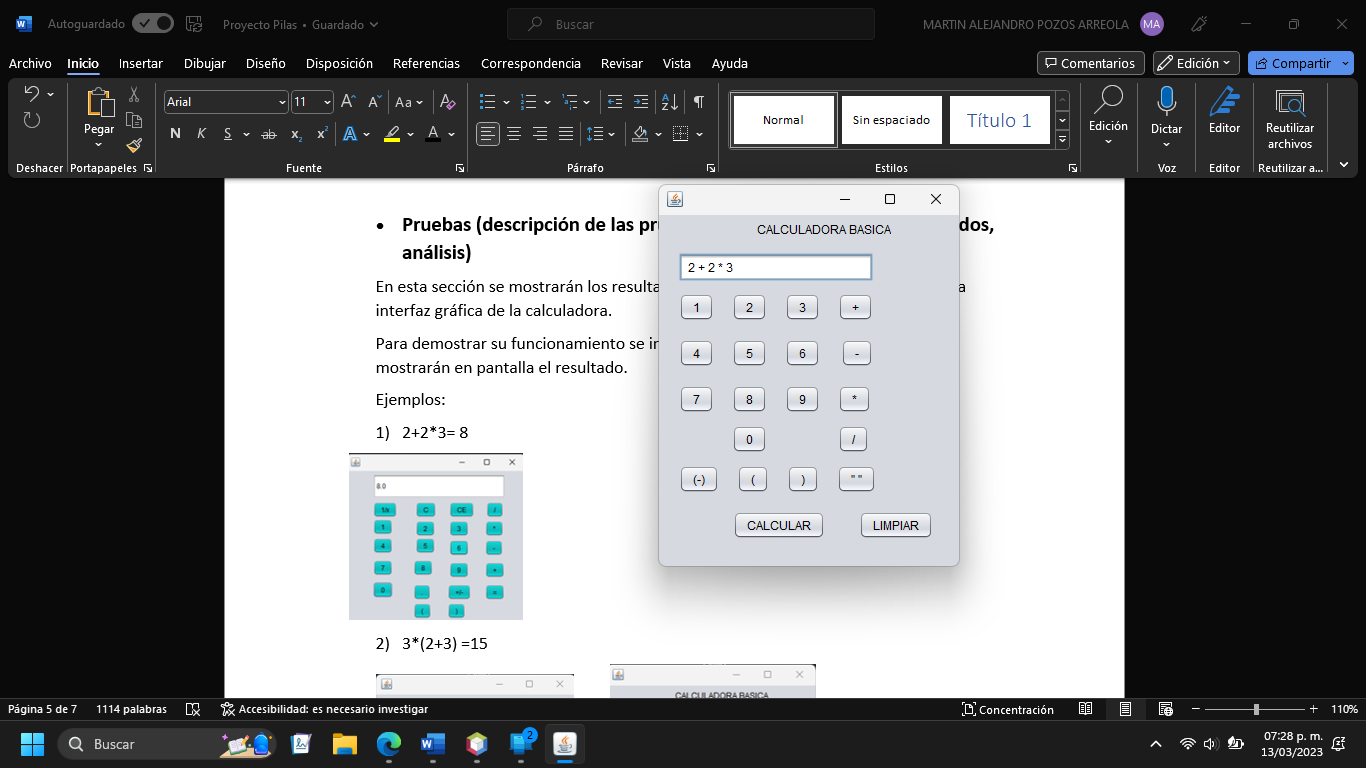
* **Pruebas** **(descripción de las pruebas realizadas, resultados obtenidos, análisis)**

En esta sección se mostrarán los resultados obtenidos al realizar el programa de la interfaz gráfica de la calculadora.

Para demostrar su funcionamiento se ingresarán las siguientes operaciones y se mostrarán en pantalla el resultado.

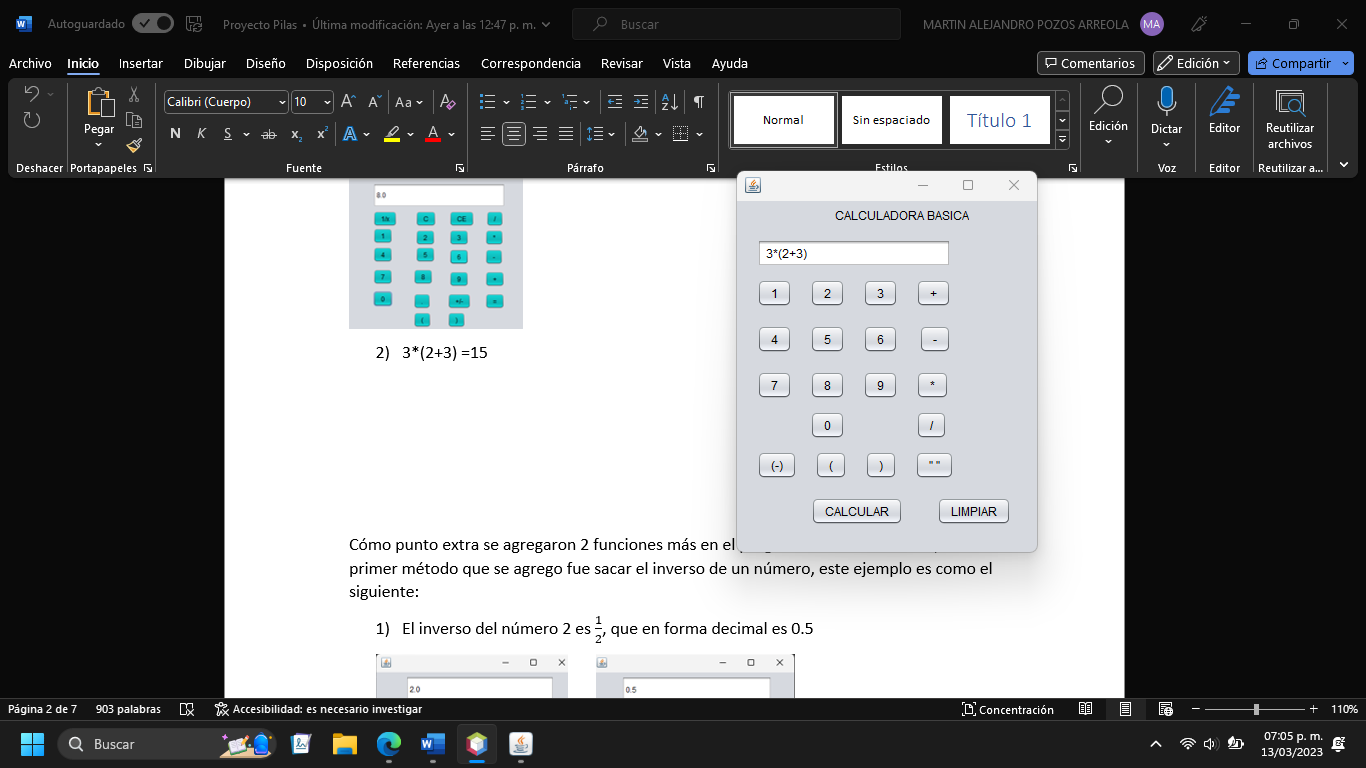
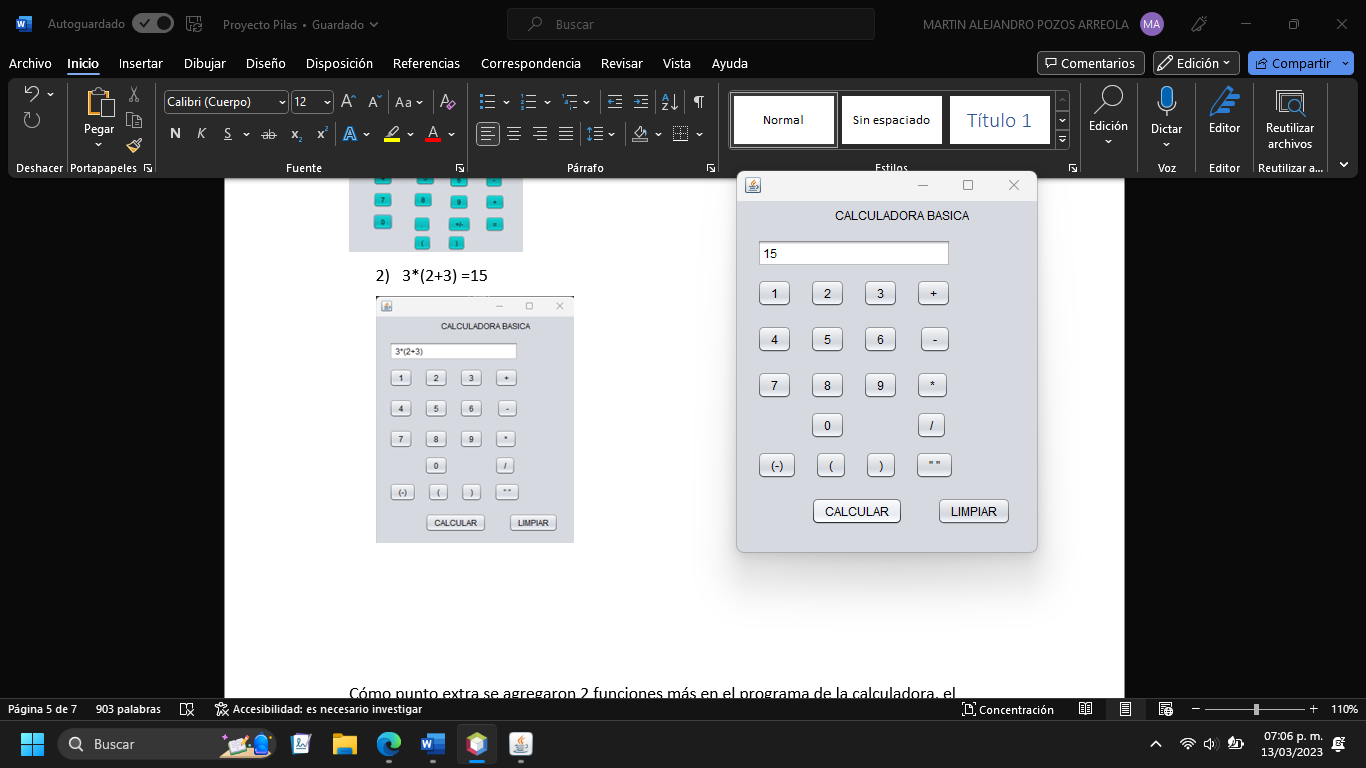
Ejemplos:

1. 2+2\*3= 8

 Captura de pantalla de computadora

Descripción generada automáticamente

1. 3\*(2+3) =15

1. 50/25

Captura de pantalla de computadora

Descripción generada automáticamenteCaptura de pantalla de computadora

Descripción generada automáticamente

Estos resultados se verificaron con el uso calculadora solamente para ver si el resultado estaba correcto.

Viendo los resultados que se realizaron anteriormente, los resultados están correctos, por lo que podemos llegar a que la calculadora puede realizar operaciones aritméticas básicas de manera correcta.

* **Limitaciones de la solución**

Al momento de realizar este proyecto se presentaron limitaciones en el programa de la interfaz gráfica de la calculadora.

Solamente se presento una limitación en el programa; la limitación es que al momento de ingresar los valores numéricos es que es necesario ingresar espacios después de cada número, paréntesis u operador para que de esta forma el programa funcione correctamente.

En caso de que no se ingresen los espacios como se mencionó con anterioridad, al ingresar los valores, mandará error de código y no funcionará el programa.

Es la única limitación que se presenta en el programa, que a futuro se puede mejorar para evitar estos percances que se presentan en el programa.

* **Posibles mejoras y conclusiones**

Algunas posibles mejorar que se pueden realizar al programa podría ser que se pueda mejorar es la parte de la presentación de la interfaz gráfica, se podría añadir más funciones matemáticas a la calculadora para que de una cierta forma sea lo más parecido a las funciones de una calculadora científica, otra mejora que se puede realizar es mejorar el aspecto que al ingresar los valores en la calculadora para que no necesite añadir espacios entre cada valor ingresado, que sea posible ingresar los valores juntos y de esta forma, también se pueda ver en pantalla el resultado correcto

Cómo conclusión al elaborar este proyecto nos dio una buena y clara idea de como es el diseño y el programa que utilizan las calculadoras que hoy en día usamos.

Posiblemente la mayoría de las personas han pensado que programar una calculadora es fácil, en teoría si es algo fácil, pero tiene su chiste realizar cada paso de la calculadora; como por ejemplo la parte de la interfaz gráfica de la calculadora tiene que ser “agradable” a la vista del usuario (es decir, que a simple vista se entienda los comandos que puede realizar la calculadora), otro ejemplo es la programación de la calculadora, porque primero se realiza la parte de la interfaz gráfica y a partir de aquí, se empieza a programar y darle la función a cada parte que conformará la calculadora.

En este proyecto el reto que enfrentamos fue juntar el programa con los métodos de pilas que se tenían que emplear, porque en esta sección del programa tiene que evaluar expresiones aritméticas y mostrar el resultado correcto. Un ejemplo para explicar esto sería como esta siguiente expresión:

1. 2+3\*2= 8 🡪 Primero se realiza la multiplicación por jerarquía de operaciones, y al final la suma.
2. 2\*(2+2) =8🡪En este caso se realiza primero lo que esta dentro del paréntesis, después se realizará la multiplicación
3. (3\*(8/2) =? 🡪 En este caso cómo no se ingresaron la cantidad de paréntesis en la expresión, mandará un error porque no está completa la expresión matemática para realizar la operación, porque el programa recorre la expresión matemática ingresada y checa los paréntesis; observa que hay un paréntesis que abre “(“, después checa que hay otro paréntesis que abre “(“, luego hay un paréntesis que cierra “)” por lo que al anterior paréntesis que abre ya está completa una pequeña parte de la operación, pero al final no encuentra otro paréntesis que cierre por lo que el paréntesis que abre del principio no cierra en ningún momento, así que la expresión matemática que se ingresó está incompleta y no se realizará la operación hasta que se complete la expresión.

Este fue el reto más interesante del proyecto, pero se logró conseguir el objetivo, por lo que nos deja una enseñanza a cada miembro del equipo que a futuro se podrá utilizar este aprendizaje para proyectos.

* **Bibliografías y referencias**

Guardati Buemo Silvia, (2015), ***“Estructura de datos básicas. Programación orientada a objetos con Java”***, Primera edición, Alfaomega Grupo Editor, México, págs.333-347, Recuperado de: [ESTRUCTURAS DE DATOS BÁSICAS - Programación Orientada a Objetos con Java (elogim.com)](https://alfaomegacloud2.itam.elogim.com/read/9786076225004/index).

* **Apéndice del código**