Universidad Estatal a Distancia de Costa Rica.

Escuela de ciencias exactas y naturales

FUNDAMENTOS DE PROGRAMACION WEB

PROYECTO #1

Olman Rojas Espinoza

Carnet: 0110900754

MARZO 2015

CONTENIDO.

[INTRODUCCION 3](#_Toc412988803)

[DESARROLLO. 4](#_Toc412988804)

[MANUAL 5](#_Toc412988805)

[CONCLUSION 7](#_Toc412988806)

[CONCLUSIONES. 8](#_Toc412988807)

[BIBLIOGRAFIA 9](#_Toc412988808)

# INTRODUCCION

Estimado lector:

En el siguiente trabajo Ud. encontrará el diseño de una calculadora, esta calculadora resuelve operaciones binarias y algunas básicas del cálculo matemático.

La Calculadora utiliza una arquitectura cliente-servidor, donde el cliente solo provee una expresión matemática y el servidor resuelve la operación matemática devolviendo un resultado fiable.

Se utiliza el patrón de diseño MVC, en donde:

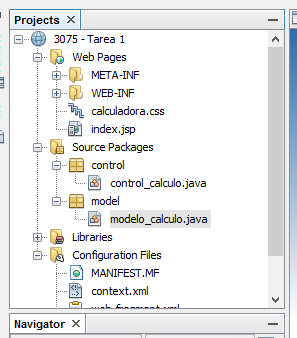
* El Modelo es donde se resuelven las operaciones matemáticas.
* El Controlador se encarga dirigir el tráfico entre la y el Modelo.
* La Vista donde el cliente puede visualizar las entradas y salidas.

Para construir las expresiones matemáticas es necesario hacerlo conforme al manual y se puede utilizar únicamente el mouse de la computadora.

La Interfaz utiliza elementos de la programación en WEB, utiliza CSS y JavaScript, El procesamiento del servidor es utilizando Tomcat y Java.

# DESARROLLO.

A continuación abra el proyecto en NetBeans y observe la siguiente estructura:



Donde Web Pages = Vista.

Dicho esto, a continuación se presentan los 4 archivos del proyecto que se utilizaron, sírvase en leerlos, dentro del código existe documentación interna donde se explican los diferentes algoritmos utilizados.

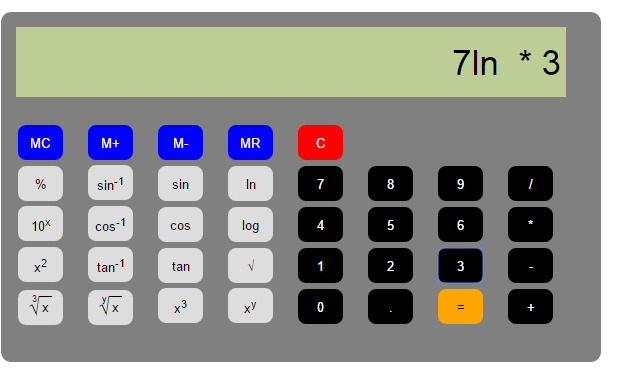


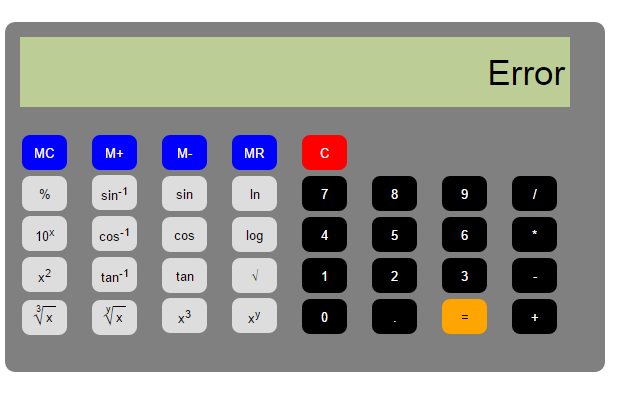
# MANUAL

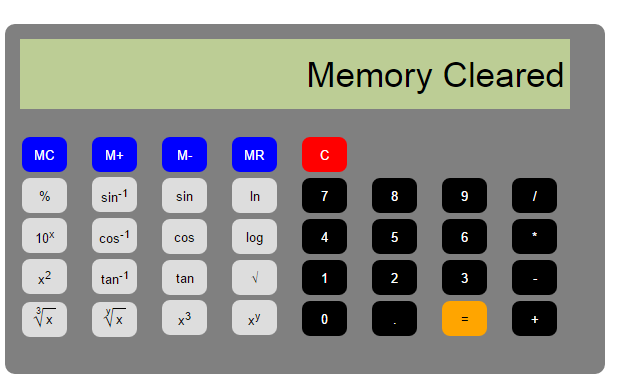
Esta calculadora solo soporta un máximo de dos operaciones.

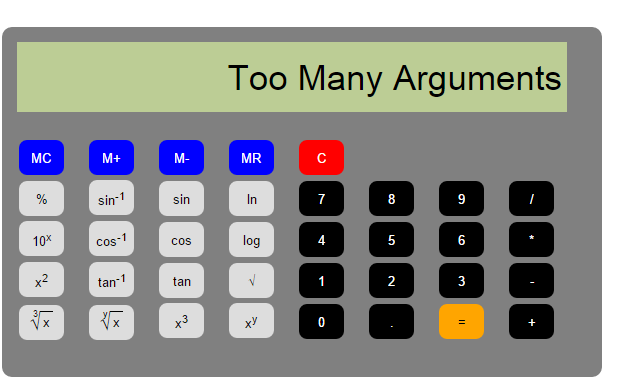
Por lo tanto:

* Argumentos matemáticos de más de 1 operador dará como resultado un Error:
* Expresiones Matemáticas extensas no se procesan y en su lugar una leyenda que reza “Too Many Arguments”.
* Para visualizar los resultados únicamente con el “=” (botón naranja).
* Cada operación nueva debe utilizar un “reset” (botón color rojo).









# CONCLUSION

Como se mencionó en un principio este proyecto se utilizó el paradigma de programación MVC.

La interfaz es únicamente donde se toman los datos y se le dan al controlador el cual se encarga de procesar los datos recibidos.

Uno de los problemas que se encontraron durante el desarrollo de este problema fue para el control de memoria, ya que las variables desaparecen una vez que concluye la comunicación entre la clase y la interfaz.

Por lo tanto se tuvo que utilizar una sesión y asignarle los valores, las sesiones permiten guardar valores durante el tiempo que el usuario tenga su navegador abierto.

Fue necesario utilizar elementos de Java Script para formatear muchos mensajes de entrada, con esto se evitó grandemente errores de validación, los cuales existen pero son muy lógicos.

# CONCLUSIONES.

El patrón de Diseño MVC es muy práctico, permite más eficiencia a la hora de cambio de métodos o agregar nuevas validaciones.

Si se quiere ampliar la calculadora se puede cambiar de Parser, existen algunas opciones en Internet Gratuitas o se pueden desarrollar, por la forma en que fue diseñada esta calculadora permite cambiar el parser y de esta manear se podría utilizar expresiones matemáticas avanzadas.

Si se hiciera con implementaciones o 3rd party, es únicamente de añadir la clase nueva e instanciar en la clase original y la calculadora seguiría siendo la misma.

Las validaciones de entrada son muy importantes en este tipo de patrón.

A mi parecer siento más sencillo la programación con este tipo de paradigma. (MVC).

# BIBLIOGRAFIA

Generalic, E. (2015, March 1). *WRITING MATHEMATICAL AND CHEMICAL EQUATIONS ON THE WEB WITH CSS.* Retrieved from http://www.periodni.com/.

Saez, H. S. (2014, 09 26). *Ejemplo Basico MVC - JAVA - Netbeans.* Retrieved from youtube.com: https://www.youtube.com/watch?v=S-FGoEBjTSA