

**FACULTAD DE CIENCIAS EXACTAS Y NATURALES**

**ESCUELA DE INFORMÁTICA**

**CURSO: Paradigmas de programación**

**Proyecto I**

**Profesor: Christian Garita**

**Estudiantes**

**Sergio Villegas**

**Ou Ming**

**Ciclo II, 2014**

**Evaluación**

|  |  |  |
| --- | --- | --- |
| **Rubro por evaluar** | **Total**  **porcentaje** | **Porcentaje**  **obtenido** |
| **DOCUMENTO DEL PROYECTO** | **30** |  |
| **SOLUCIÓN DEL PROBLEMA EN SCHEME** | **40** |  |
| **INTERFAZ GRÁFICA** | **20** |  |
| **DEFENSA** | **10** |  |
| **PORCENTAJE TOTAL OBTENIDO** | **100** |  |

Tabla de contenido

[Introducción: 4](#_Toc400916252)

[Objetivo General: 4](#_Toc400916253)

[Objetivos Específicos: 4](#_Toc400916254)

[Limitaciones: 4](#_Toc400916255)

[Uso del programa. 5](#_Toc400916256)

[Desarrollo 7](#_Toc400916257)

[Explicación de la Programación 7](#_Toc400916258)

[Conclusión 11](#_Toc400916259)

[Bibliografía 12](#_Toc400916260)

# Introducción:

El siguiente documento contiene los pasos que se realizaron para lograr desarrollar una calculadora en scheme. Recopilando e investigando informaciones a través de sitios de web y documentos, el cual nos proporcionó la información para su realización.

# Objetivo General:

Desarrollar una calculadora gráfica en Scheme que nos permita fortalecer nuestros conocimientos en Scheme.

# Objetivos Específicos:

1. Investigar sobre el entorno gráfico en Scheme

2. Implementar las diferentes funcionalidades para el cumplimiento del proyecto.

3. Realizar un documento escrito en donde se especifique y explique la solución planteada para el proyecto.

# Limitaciones:

Hasta el momento no se han encontrado ninguna limitación.

# Uso del programa.

A continuación se explica cómo se utiliza la calculadora.

1. Como el programa no es una versión construida para ser instalada, es necesario tener instalado el programa DrRacket para poder ejecutar el archivo “calculadora.rkt”.
2. Debemos abrir el programa DrRacket y acontinuación el archivo “calculadora.rkt” y se procede a presionar el botón “run”. De esta manera se ejecutará el programa.
3. El programa en ejecución luce como la siguiente imagen: 

Ilustración Programa de la calculadora en ejecución. Tomada de la ejecución del programa

1. Si se desea realizar las operaciones básicas (suma, resta, multiplicación y división) se debe proceder a presionar el primer número, siguiente el operador que deseamos aplicar y a continuación el segundo número. Para saber su resultado debemos presionar la tecla: .
2. Si deseamos elevar un numero al cuadrado o al cubo simplemente se debe presionar el número que se desea elevar y a continuación  si es para elevar al cuadrado, ó  para elevar al cubo.
3. Si se desea elevar un número, y que el exponente no sea 2 o 3, debemos seguir estos pasos: primero presionar el número a elevar, segundo presionar, tercero presionar la cantidad a la que se desea elevar presionado otro número. Finalmente se presiona la tecla .
4. Si se desea sacar el módulo se debe presionar el primer número, siguiente se presiona la tecla  y a continuación se presiona el otro número. Luego presionamos la tecla.
5. Otras funciones que se pueden realizar: por ejemplo se debe presionar el numero deseado y a continuación cualquiera de las siguientes teclas según se desee.
   1. Logaritmo Natural .
   2. Seno .
   3. Coseno .
   4. Tangente .
   5. Notación científica 
   6. Logaritomo 
   7. Raiz cuadrada 
   8. Factorial 

# Desarrollo

## Explicación de la Programación

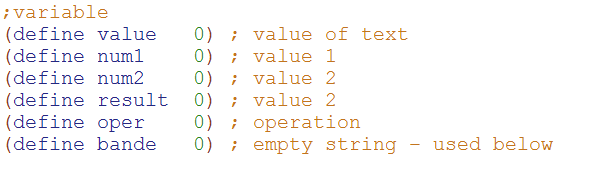
****

Ilustración Variables globales. Tomada de código de la calculadora

Se crean 6 variables globales para guardar diferentes valores

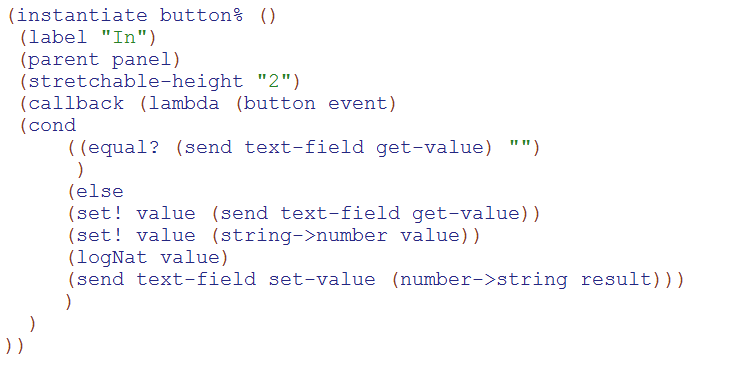


Ilustración Botones. Variables globales. Tomada de código de la calculadora

Así se implementan todos los botones y su respectiva validación. Cuando se da click en cualquier botón se guardan los valores a evaluar y se llama a los métodos necesarios para llegar al resultado.

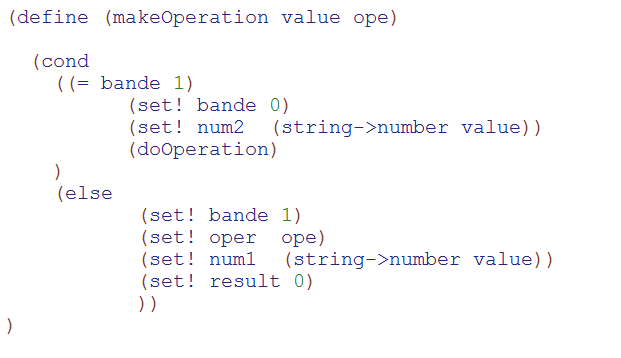


Ilustración Método makeOperation. Tomada de código de la calculadora

Se implementó el método makeOperation que recibe como parámetro un valor y un operador, donde al entrar por primera al método cambia el estado de la bandera a 1 y almacena el operador y el primer valor que se quiere evaluar, al entrar por segunda vez, se cambia el estado de la bandera a 0, recibe el segundo número a evaluar y llama al método doOperation.

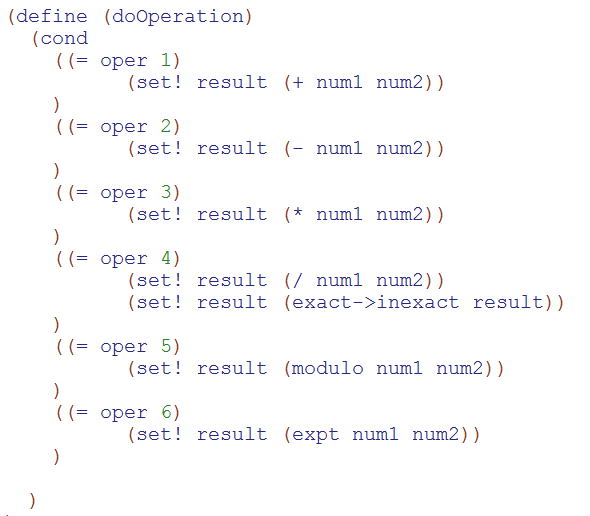


Ilustración método doOperation. Tomada de código de la calculadora.

Donde en este método se evalúa el operador, ya que cada número del 1 al 6 representa un operador diferente, como por ejemplo si se guardó un 1 en la variable oper se realizara la suma y así dependiendo del valor almacenado en la variable oper.

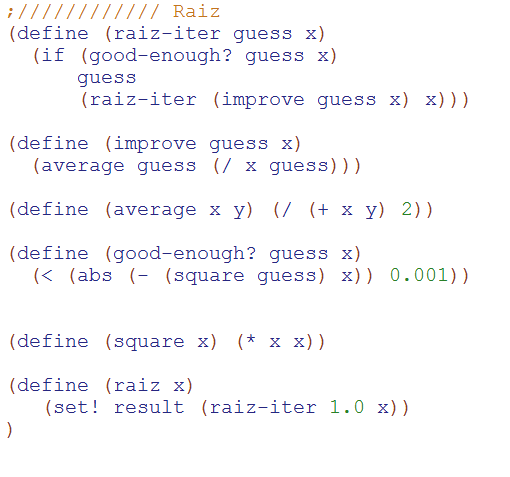


Ilustración Método para calcular la raíz cuadrada de un valor dado. Tomada de código de la calculadora.

Método para encontrar la Raíz cuadrada de un número.

# Conclusión

Con respecto al proyecto se logró adquirir los conocimientos necesarios en la programación grafica en scheme para poder finalizar dicho trabajo. Fue muy importante la investigación a través de diferentes recursos. Ya que no estamos acostumbrados a programar en este lenguaje, por lo tanto tuvimos a acoplarnos a este lenguaje y a su lógica de programación. Al final se logró todo los objetivos esperados en este proyecto y el conocimiento sobre la programación grafica en este lenguaje.

# Bibliografía

Matthew Flatt, Robert Bruce Findler and John Clements. August 1, 2014. The Racket Graphical Interface Toolkit. Cambridge University.

Matthias Felleisen. (El 7 de Junio de 2010). Racket Documentation. 01/10/2014, de Copyright (c) 2010-2014 PLT Design Inc. Sitio web: http://docs.racket-lang.org/index.html

[danielmiladinov](https://github.com/danielmiladinov). ( Feb 17, 2014). newtons-method-sqrt.scm. 01/10/2014. Sitio web: https://github.com/danielmiladinov/sicp-scheme/commits/master/newtons-method-sqrt.scm