

# TRABAJO PRÁCTICO FINAL

Docentes:

Catalano, Ignacio

Nomdedeu, Iván

Materia: Introducción a Ingeniería en Computación

Código: B6001

Comisión: 3

Estudiantes:

Ceballos, Bruno Andrés

Yaben, Luciano Hugo

Mail:

[brunoandresceballos@gmail.com](mailto:brunoandresceballos@gmail.com)

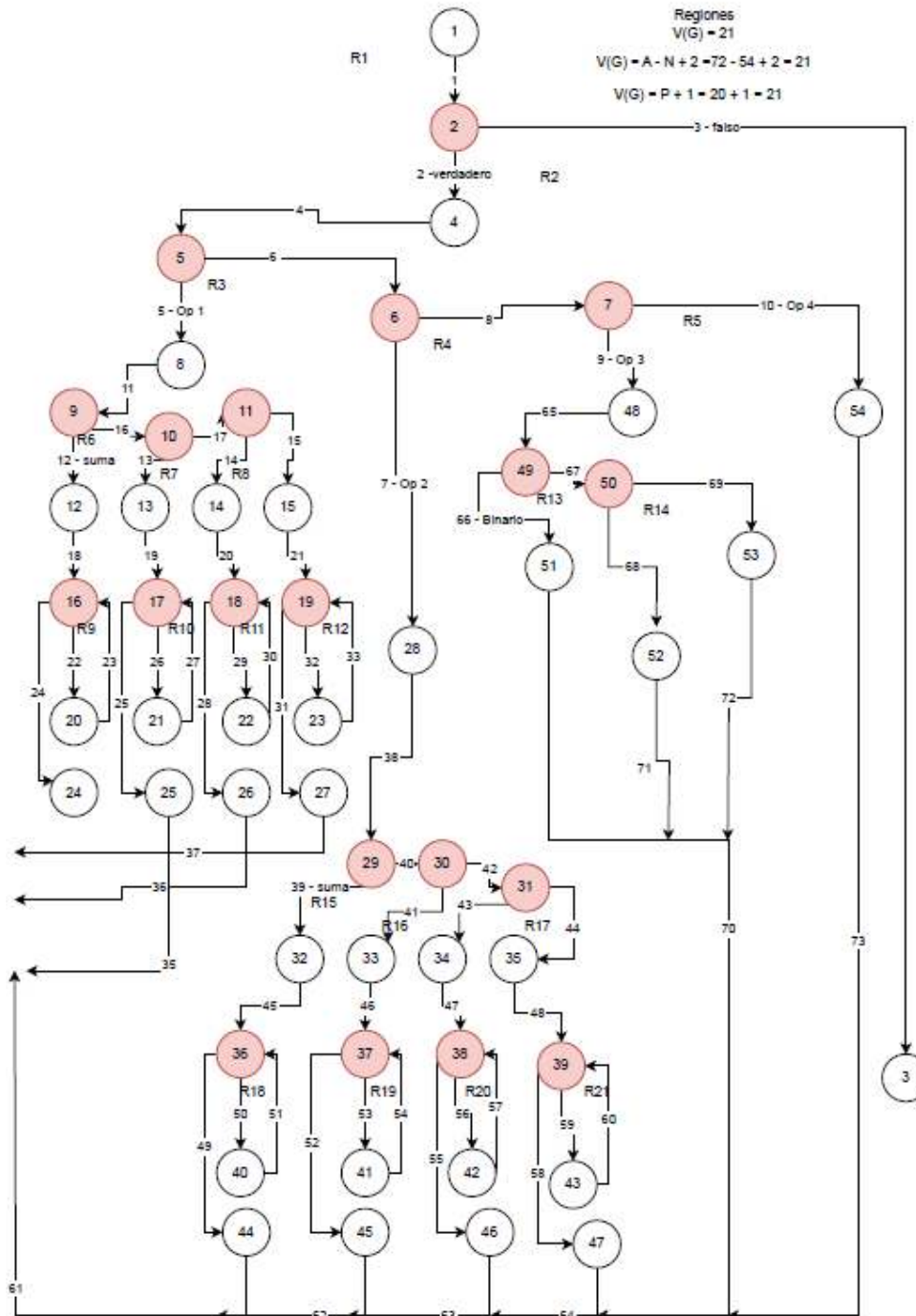
[lyaben@gmail.com](mailto:lyaben@gmail.com)

Fecha de entrega: 22/06/2023

Diagrama de Flujo



## Complejidad Ciclomática



## Códigos Python

```

Calculadora Clásica
#Calculadora Clásica Trabajo Grupal Final - Ceballos - Yaben

#Función que indica si el valor ingresado es un flotante o un signo '='
def float_val(entrada):
    if entrada == '=':
        valido = True
    else:
        try:
            entrada = float(entrada)
        except:
            valido = False
        else:
            valido = True
    return valido

#Función que indica si el operador ingresado es uno válido
def op_val(entrada2):
    if entrada2 == '+':
        valido2 = True
    elif entrada2 == '-':
        valido2 = True
    elif entrada2 == '*':
        valido2 = True
    elif entrada2 == '/':
        valido2 = True
    elif entrada2 == '=':
        valido2 = True
    else:
        valido2 = False
    return valido2

#Cuerpo principal de la calculadora, se ejecuta desde el selector
def clasica():
    op = ""
    base = input("Ingrese número: ")          #Ingreso el número que va a servir
    como base
    while not float_val(base):
        base = input("Ingrese un número válido: ")    #Mientras el número no sea un float
    o un signo igual, lo sigo pidiendo
    if base != '=':
        #Si la base ingresada no es un igual, lo convierto
    a float y pido operando
        base = float(base)
        while op != '=':
            #Comienzo a operar. Se puede salir ingresando
    un igual en cualquier momento (cuando se pide un operando o un número)
            op = input("Ingrese un operando (+, -, *, /, =): ")
            while not op_val(op):
                op = input("Ingrese un operando válido (+, -, *, /, =): ") #Si el operando
    ingresado no válido, lo sigo pidiendo
            if op != '=':
                #Si el operando no es un igual, pido otra
    número para operar

```

```

        if op == '/':
            #Separo el caso de la división para
            asegurarme de no poder dividir por 0
            num = input("Ingrese número: ")
            while (not float_val(num)):
                num = input("Ingrese un número válido: ")
                #Si el número no es un
                float o un signo igual, lo sigo pidiendo
            if num != '=':
                if num == '0':
                    #Si el número ingresado es un 0,
                    salgo de la calculadora e imprimo un mensaje de error
                    op = '='
                    base = "Error al dividir por 0"
                else:
                    num = float(num)
                    #Si el número es válido, hago la
                    división
                    base = base/num
            else:
                op = '='
        else:
            num = input("Ingrese número: ")
            #Si el operador no es el de
            división, pido un número y hago la operación que corresponda
            while not float_val(num):
                num = input("Ingrese un número válido: ")
                #Si el número no es un
                flotante o un signo igual, lo sigo pidiendo
            if num != '=':
                num = float(num)
                if op == '+':
                    base = base + num
                elif op == '-':
                    base = base - num
                else:
                    base = base * num
            else:
                op = '='
        print(f"Resultado: {base}")
        #Imprimo el resultado de las
        operaciones cuando ingreso '=' o el mensaje de error en caso de dividir por 0
    else:
        print("Resultado: 0")
        #Imprimo 0 como resultado en
        caso de ingresar un '=' inmediatamente después de iniciar la calculadora

```

## Calculadora de Fracciones

#Calculadora de Fracciones Trabajo Grupal Final - Ceballos - Yaben

#Esta función ve si el número ingresado es una fracción.

def fraccionValida(n):

try:

if n != "=":

#Descomponer fracción

resultado = 0

longitud = len(n)

inicio = 0

ubicacionBarra = 0

hayBarra = False

while inicio &lt; longitud:

#print(f"Caracter: '{n[inicio:inicio+1]}'")

inicio = inicio + 1

if n[inicio:inicio+1] == "/":

ubicacionBarra = inicio #arrancando en 0 la cadena

hayBarra = True

#print(f"ubicacionBarra: '{ubicacionBarra}'")

if hayBarra:

numerador = n[:ubicacionBarra]

numeradorInt = int(numerador) #####

denominador = n[ubicacionBarra+1:]

denominadorInt = int(denominador) #####

resultado = int(numerador) / int(denominador)

#print(f"resultado: '{resultado}'")

else:

#puede ser que sea un numero sin denominador

resultado = int(n) / 1

#print(f"resultado: '{resultado}'")

return True

else:

return True

except ValueError:

print(f"El valor no es valido")

return False

#Esta función ve si el número ingresado es una fracción pero no se da ningun mensaje.

def fraccionValidaST(n):

try:

if n != "=":

#Descomponer fracción

resultado = 0

longitud = len(n)

inicio = 0

ubicacionBarra = 0

hayBarra = False

while inicio &lt; longitud:

#print(f"Caracter: '{n[inicio:inicio+1]}'")

inicio = inicio + 1

if n[inicio:inicio+1] == "/":

ubicacionBarra = inicio #arrancando en 0 la cadena

hayBarra = True

```

        #print(f"ubicacionBarra: '{ubicacionBarra}'")

    if hayBarra:
        numerador = n[:ubicacionBarra]
        numeradorInt = int(numerador) #####
        denominador = n[ubicacionBarra+1:]
        denominadorInt = int(denominador) #####
        resultado = int(numerador) / int(denominador)
        #print(f"resultado: '{resultado}'")
    else:
        #puede ser que sea un numero sin denominador
        resultado = int(n) / 1
        #print(f"resultado: '{resultado}'")
    return True
except ValueError:
    return False

#Esta función devuelve el numerador de una fracción.
def fraccionNumerador(n):
    try:
        longitud = len(n)
        inicio = 0
        ubicacionBarra = 0
        while inicio < longitud:
            #print(f"Caracter: '{n[inicio:inicio+1]}'")
            inicio = inicio + 1
            if n[inicio:inicio+1] == "/":
                ubicacionBarra = inicio #arrancando en 0 la cadena
                #print(f"ubicacionBarra: '{ubicacionBarra}'")
        if ubicacionBarra == 0:
            return n
        else:
            return n[:ubicacionBarra]
    except ValueError:
        print(f"El valor '{n}' no pudo ser procesado...")
        return 0

#Esta función devuelve el denominador de una fracción.
def fraccionDenominador(n):
    try:
        longitud = len(n)
        inicio = 0
        ubicacionBarra = 0
        while inicio < longitud:
            #print(f"Caracter: '{n[inicio:inicio+1]}'")
            inicio = inicio + 1
            if n[inicio:inicio+1] == "/":
                ubicacionBarra = inicio #arrancando en 0 la cadena
                #print(f"ubicacionBarra: '{ubicacionBarra}'")
        if ubicacionBarra == 0:
            return 1
        else:
            return n[ubicacionBarra+1:]

```

```

except ValueError:
    print(f"El valor '{n}' no pudo ser procesado...")
    return 0

#Esta función devuelve la suma de dos fracciones
def sumarFraccion(f1, f2):
    try:
        resultado = ""
        numeradorTotal = 0
        num1 = 0
        num2 = 0
        den1 = 0
        den2 = 0

        if f1 == "":
            return f2
        else:
            #aca hago la suma de f1 + f2
            numerador1 = fraccionNumerador(f1)
            denominador1 = fraccionDenominador(f1)
            numerador2 = fraccionNumerador(f2)
            denominador2 = fraccionDenominador(f2)
            if denominador1 != denominador2:
                #Me fijo si un denominador es divisor del otro.
                if (int(denominador1) % int(denominador2)) == 0 or (int(denominador2) %
int(denominador1)) == 0:
                    if int(denominador1) > int(denominador2):
                        #1ra fracción
                        num1 = int(numerador1)
                        den1 = int(denominador1)
                        #2da fracción
                        num2 = int(numerador2) * (int(denominador1)/int(denominador2))
                        den2 = int(denominador2) * (int(denominador1)/int(denominador2))

                        numeradorTotal = num1 + num2
                        resultado = str(int(numeradorTotal)) + "/" + str(int(den1))
                    else:
                        #1ra fracción
                        num1 = int(numerador1) * (int(denominador2)/int(denominador1))
                        den1 = int(denominador1) * (int(denominador2)/int(denominador1))
                        #2da fracción
                        num2 = int(numerador2)
                        den2 = int(denominador2)

                        numeradorTotal = num1 + num2
                        resultado = str(int(numeradorTotal)) + "/" + str(int(den1))
                    else:
                        #1ra fracción
                        num1 = int(numerador1) * int(denominador2)
                        den1 = int(denominador1) * int(denominador2)
                        #2da fracción
                        num2 = int(numerador2) * int(denominador1)
                        den2 = int(denominador2) * int(denominador1)

                        numeradorTotal = num1 + num2

```



```

        resultado = str(int( NumeradorTotal )) + "/" + str(int(den1))
    else:
        NumeradorTotal = int(Numerador1) + int(Numerador2)
        resultado = str(int(NumeradorTotal)) + "/" + str(int(Denominador1))

    return resultado
except ValueError:
    print(f"La suma de fracciones no pudo ser procesada...")
    return ""

#Esta función devuelve la resta de dos fracciones
def restarFraccion(f1, f2):
    try:
        resultado = ""
        NumeradorTotal = 0
        num1 = 0
        num2 = 0
        den1 = 0
        den2 = 0

        if f1 == "":
            return f2
        else:
            #aca hago la resta de f1 - f2
            Numerador1 = fraccionNumerador(f1)
            Denominador1 = fraccionDenominador(f1)
            Numerador2 = fraccionNumerador(f2)
            Denominador2 = fraccionDenominador(f2)
            if Denominador1 != Denominador2:
                #Me fijo si un denominador es divisor del otro.
                if (int(Denominador1) % int(Denominador2)) == 0 or (int(Denominador2) %
int(Denominador1)) == 0:
                    if int(Denominador1) > int(Denominador2):
                        #1ra fracción
                        num1 = int(Numerador1)
                        den1 = int(Denominador1)
                        #2da fracción
                        num2 = int(Numerador2) * (int(Denominador1)/int(Denominador2))
                        den2 = int(Denominador2) * (int(Denominador1)/int(Denominador2))

                        NumeradorTotal = num1 - num2
                        resultado = str(int(NumeradorTotal)) + "/" + str(int(den1))
                    else:
                        #1ra fracción
                        num1 = int(Numerador1) * (int(Denominador2)/int(Denominador1))
                        den1 = int(Denominador1) * (int(Denominador2)/int(Denominador1))
                        #2da fracción
                        num2 = int(Numerador2)
                        den2 = int(Denominador2)

                        NumeradorTotal = num1 - num2
                        resultado = str(int(NumeradorTotal)) + "/" + str(int(den1))
                    else:
                        #1ra fracción
                        num1 = int(Numerador1) * int(Denominador2)

```

```

        den1 = int(denominador1) * int(denominador2)
        #2da fracción
        num2 = int( Numerador2 ) * int(denominador1)
        den2 = int(denominador2) * int(denominador1)

        NumeradorTotal = num1 - num2
        resultado = str(int(NumeradorTotal)) + "/" + str(int(den1))
    else:
        NumeradorTotal = int(Numerador1) - int(Numerador2)
        resultado = str(int(NumeradorTotal)) + "/" + str(int(denominador1))

    return resultado
except ValueError:
    print(f"La resta de fracciones no pudo ser procesada...")
    return ""

#Esta función devuelve la multiplicación de dos fracciones
def multiplicarFraccion(f1, f2):
    try:
        resultado = ""
        NumeradorTotal = 0
        num1 = 0
        num2 = 0
        den1 = 0
        den2 = 0

        if f1 == "":
            return f2
        else:
            Numerador1 = fraccionNumerador(f1)
            denominador1 = fraccionDenominador(f1)
            Numerador2 = fraccionNumerador(f2)
            denominador2 = fraccionDenominador(f2)
            num = int(Numerador1) * int(Numerador2)
            den = int(denominador1) * int(denominador2)
            resultado = str(int(num)) + "/" + str(int(den))
        return resultado
    except ValueError:
        print(f"La multiplicación de fracciones no pudo ser procesada...")
        return ""

#Esta función devuelve la división de dos fracciones
def dividirFraccion(f1, f2):
    try:
        resultado = ""
        NumeradorTotal = 0
        num1 = 0
        num2 = 0
        den1 = 0
        den2 = 0

        if f1 == "":
            return f2
        else:
            Numerador1 = fraccionNumerador(f1)

```

```

        denominador1 = fraccionDenominador(f1)
        numerador2 = fraccionNumerador(f2)
        denominador2 = fraccionDenominador(f2)
        num = int(numerador1) * int(denominador2)
        den = int(denominador1) * int(numerador2)
        resultado = str(int(num)) + "/" + str(int(den))
    return resultado
except ValueError:
    print(f"La división de fracciones no pudo ser procesada...")
    return ""

```

#### #Calculadora de Fracciones

```

def fracciones():
    print('Calculadora de Fracciones')
    print("Operaciones disponibles:")
    print("1: Suma")
    print("2: Resta")
    print("3: Multiplicación")
    print("4: División")
    operacion = input()
    #print(f"Operación '{operacion}'")
    resultado = ""

    if operacion != "1" and operacion != "2" and operacion != "3" and operacion != "4":
        print("Ingrese una opción valida.")
        operacion = "5"

```

#### #Suma

```

if int(operacion) == 1:
    fraccion = ""
    fraccion = input("Ingrese numero:")
    if fraccionDenominador(fraccion) != "0":
        if fraccionValida(fraccion):
            while fraccion != "=":
                resultado = sumarFraccion(resultado, fraccion)
                if fraccionValidaST(resultado):
                    fraccion = input("Ingrese numero:")
                    if fraccionValida(fraccion):
                        if fraccionDenominador(fraccion) == "0":
                            print("El denominador no puede ser cero.")
                            fraccion = "="
                            resultado = ""
                        else:
                            fraccion = "="
                if fraccionValidaST(resultado):
                    if resultado != "":
                        print(f"Resultado {resultado}")
            else:
                print("El denominador no puede ser cero.")

```

#### #Resta

```

if int(operacion) == 2:
    fraccion = ""
    fraccion = input("Ingrese numero:")
    if fraccionDenominador(fraccion) != "0":
        if fraccionValida(fraccion):

```

```

while fraccion != "=":
    resultado = restarFraccion(resultado, fraccion)
    if fraccionValidaST(resultado):
        fraccion = input("Ingrese numero:")
        if fraccionValida(fraccion):
            if fraccionDenominador(fraccion) == "0":
                print("El denominador no puede ser cero.")
                fraccion = "="
                resultado = ""
            else:
                fraccion = "="
        if fraccionValidaST(resultado):
            if resultado != "":
                print(f"Resultado {resultado}")
    else:
        print("El denominador no puede ser cero.")
#Multiplicación
if int(operacion) == 3:
    fraccion = ""
    fraccion = input("Ingrese numero:")
    if fraccionNumerador(fraccion) != "0":
        if fraccionDenominador(fraccion) != "0":
            if fraccionValida(fraccion):
                while fraccion != "=":
                    resultado = multiplicarFraccion(resultado, fraccion)
                    if fraccionValidaST(resultado):
                        fraccion = input("Ingrese numero:")
                        #if fraccionValida(fraccion):
                        if fraccionDenominador(fraccion) == "0":
                            print("El denominador no puede ser cero.")
                            fraccion = "="
                            resultado = ""
                        else:
                            fraccion = "="
                    if fraccionValidaST(resultado):
                        if resultado != "":
                            print(f"Resultado {resultado}")
            else:
                print("El denominador no puede ser cero.")
        else:
            print(f"Resultado 0")
#División
if int(operacion) == 4:
    fraccion = ""
    fraccion = input("Ingrese numero:")
    if fraccionNumerador(fraccion) != "0":
        if fraccionDenominador(fraccion) != "0":
            if fraccionValida(fraccion):
                while fraccion != "=":
                    resultado = dividirFraccion(resultado, fraccion)
                    if fraccionValidaST(resultado):
                        fraccion = input("Ingrese numero:")
                        if fraccionValida(fraccion):
                            #if fraccionDenominador(fraccion) == "0":
                            if fraccionNumerador(fraccion) == "0":

```

```
        print("El denominador no puede ser cero.")
        fraccion = "="
        resultado = ""
    else:
        fraccion = "="
    if fraccionValidaST(resultado):
        if resultado != "":
            print(f"Resultado {resultado}")
    else:
        print("El denominador no puede ser cero.")
else:
    print(f"Resultado 0")
```

## Calculadora Conversora

#Calculadora Conversora Trabajo Grupal Final - Ceballos - Yaben

#Función que verifica que la entrada sea un entero positivo

```
def es_int(valor):  
    try:  
        valor = int(valor)  
    except:  
        valido = False  
    else:  
        if valor >= 0:  
            valido = True  
        else:  
            valido = False  
    return valido
```

#Función que verifica que la opción elegida sea válida

```
def op_val(opcion):  
    if opcion == '1':  
        valido2 = True  
    elif opcion == '2':  
        valido2 = True  
    elif opcion == '3':  
        valido2 = True  
    else:  
        valido2 = False  
    return valido2
```

#Función que convierte un número en base decimal en un string en base 2 u 8 y lo imprime

```
def conv_bin_oct(numero, base):  
    orig = numero  
    resul = ""  
    while numero >= base:  
        resul = str(numero%base) + resul  
        numero = numero//base  
    resul = str(numero) + resul  
    print (f"{orig} en base {base} es: {resul}")
```

#Función que convierte dígitos en base decimal a caracteres en base hexadecimal

```
def dig_hex(digito):  
    if digito < 10:  
        hexa = str(digito)  
    elif digito == 10:  
        hexa = 'A'  
    elif digito == 11:  
        hexa = 'B'  
    elif digito == 12:  
        hexa = 'C'  
    elif digito == 13:  
        hexa = 'D'  
    elif digito == 14:  
        hexa = 'E'  
    elif digito == 15:  
        hexa = 'F'
```

```

return hexa

#Función que convierte un número en base decimal en un sting en base 16 y lo imprime
def conv_hex(numero):
    orig = numero
    resul = ""
    while numero >= 16:
        resul = dig_hex(numero%16) + resul
        numero = numero//16
    resul = dig_hex(numero) + resul
    print (f"{orig} en base 16 es: {resul}")

#Cuerpo principal de la calculadora, se ejecuta desde el selector
def conversora():
    numero = input("Ingrese el número a convertir: ") #Pido el número a convertir,
    verificando que sea un entero positivo
    while not es_int(numero):
        numero = input("Ingrese un valor válido (entero positivo): ")
    numero = int(numero)
    print ("_____ \n" #Imprimo la interfaz
    de selección
    "|Seleccione base a la que quiere convertir: |\n"
    "|1. Bin                               |\n"
    "|2. Hex                               |\n"
    "|3. Oct                               |\n"
    "|_____|\n")
    op = input() #Ingreso la opción elegida
    while not op_val(op): #Verifico que la opción ingresada sea válida
        op = input("Seleccione una opción válida (1-3): ")
    if op == '1': #Derivo la opción a la función correspondiente
        conv_bin_oct(numero, 2)
    elif op == '2':
        conv_hex(numero)
    else:
        conv_bin_oct(numero, 8)

```

## Selector de Calculadora

#Selector de calculadoras Trabajo Grupal Final - Ceballos - Yaben

#Importo los módulos de cada calculadora

import calc\_clasica as clas

import calc\_fracciones as frac

import calc\_conversora as conv

#Función que imprime la interfaz del selector de calculadoras

def mostrar\_interfaz():

print("\_\_\_\_\_ \n")

"|Ingrese función deseada: | \n"

"|1. Calculadora Clásica | \n"

"|2. Calculadora de Fracciones | \n"

"|3. Calculadora de Conversiones | \n"

"|4. Salir | \n"

"|\_\_\_\_\_ |")

#Función que indica si la opción elegida es válida o no

def check\_val(valor):

if valor == '1':

valido = True

elif valor == '2':

valido = True

elif valor == '3':

valido = True

elif valor == '4':

valido = True

else:

valido = False

return valido

#Cuerpo principal, donde el usuario ingresa la opción deseada

op = '0'

while op != '4':

mostrar\_interfaz()

op = input("Función: ")

while not check\_val(op): #Verifico que la opción elegida sea una válida

op = input("Ingrese una función correcta (1-4): ")

if op == '1':

clas.clasica()

elif op == '2':

frac.fracciones()

elif op == '3':

conv.conversora()



**Casos de prueba para caminos:**

Camino: Calculadora clásica, suma, resta, multiplicación y división.

|Ingrese función deseada: |  
 |1. Calculadora Clásica |  
 |2. Calculadora de Fracciones |  
 |3. Calculadora de Conversiones |  
 |4. Salir |  
 |  
 Función: 1  
 Ingrese número: 12  
 Ingrese un operando (+, -, \*, /, =): +  
 Ingrese número: 8  
 Ingrese un operando (+, -, \*, /, =): -2  
 Ingrese un operando válido (+, -, \*, /, =): -  
 Ingrese número: 2  
 Ingrese un operando (+, -, \*, /, =): \*  
 Ingrese número: 3  
 Ingrese un operando (+, -, \*, /, =): /  
 Ingrese número: 10  
 Ingrese un operando (+, -, \*, /, =): =  
 Resultado: 5.4

Camino: Calculadora de fracciones. Suma.

|Ingrese función deseada: |  
 |1. Calculadora Clásica |  
 |2. Calculadora de Fracciones |  
 |3. Calculadora de Conversiones |  
 |4. Salir |  
 |  
 Función: 2  
 Calculadora de Fracciones  
 Operaciones disponibles:  
 1: Suma  
 2: Resta  
 3: Multiplicación  
 4: División  
 1  
 Ingrese numero:1/2  
 Ingrese numero:2/5  
 Ingrese numero:6/4  
 Ingrese numero:2/5  
 Ingrese numero:=  
 Resultado 112/40

Camino: Calculadora de fracciones. Resta.

|Ingrese función deseada: |  
 |1. Calculadora Clásica |  
 |2. Calculadora de Fracciones |  
 |3. Calculadora de Conversiones |  
 |4. Salir |

<p>Función: 2</p> <p>Calculadora de Fracciones</p> <p>Operaciones disponibles:</p> <p>1: Suma</p> <p>2: Resta</p> <p>3: Multiplicación</p> <p>4: División</p> <p>2</p> <p>Ingrese numero:9/5</p> <p>Ingrese numero:2/4</p> <p>Ingrese numero:3/78</p> <p>Ingrese numero:21/58</p> <p>Ingrese numero:=</p> <p>Resultado 81384/90480</p>
Camino: Calculadora de fracciones. Multiplicación.
<p>Ingrese función deseada:</p> <p>1. Calculadora Clásica</p> <p>2. Calculadora de Fracciones</p> <p>3. Calculadora de Conversiones</p> <p>4. Salir</p> <p>Función: 2</p> <p>Calculadora de Fracciones</p> <p>Operaciones disponibles:</p> <p>1: Suma</p> <p>2: Resta</p> <p>3: Multiplicación</p> <p>4: División</p> <p>3</p> <p>Ingrese numero:1/5</p> <p>Ingrese numero:2/56</p> <p>Ingrese numero:21/4</p> <p>Ingrese numero:5</p> <p>Ingrese numero:32/6</p> <p>Ingrese numero:7/53</p> <p>Ingrese numero:=</p> <p>Resultado 47040/356160</p>
Camino: Calculadora de fracciones. División.
<p>Ingrese función deseada:</p> <p>1. Calculadora Clásica</p> <p>2. Calculadora de Fracciones</p> <p>3. Calculadora de Conversiones</p> <p>4. Salir</p> <p>Función: 2</p> <p>Calculadora de Fracciones</p> <p>Operaciones disponibles:</p> <p>1: Suma</p> <p>2: Resta</p> <p>3: Multiplicación</p>

4: División  
4  
Ingrese numero:1/53  
Ingrese numero:2/4  
Ingrese numero:23/7  
Ingrese numero:9/5  
Ingrese numero:4/5  
Ingrese numero:23  
Ingrese numero:=  
Resultado 700/2018664

Camino: Calculadora de conversiones. Binario.

Ingrese función deseada: |  
1. Calculadora Clásica |  
2. Calculadora de Fracciones |  
3. Calculadora de Conversiones |  
4. Salir |  
|

Función: 3  
Ingrese el número a convertir: 1546

Seleccione base a la que quiere convertir: |  
1. Bin |  
2. Hex |  
3. Oct |  
|

1  
1546 en base 2 es: 11000001010

Camino: Calculadora de conversiones. Hexadecimal.

Ingrese función deseada: |  
1. Calculadora Clásica |  
2. Calculadora de Fracciones |  
3. Calculadora de Conversiones |  
4. Salir |  
|

Función: 3  
Ingrese el número a convertir: 4786542

Seleccione base a la que quiere convertir: |  
1. Bin |  
2. Hex |  
3. Oct |  
|

2  
4786542 en base 16 es: 49096E

Camino: Calculadora de conversiones. Octal.

Ingrese función deseada: |  
1. Calculadora Clásica |  
2. Calculadora de Fracciones |

3. Calculadora de Conversiones
4. Salir
Función: 3
Ingrese el número a convertir: 5462145
Seleccione base a la que quiere convertir:
1. Bin
2. Hex
3. Oct
3
5462145 en base 8 es: 24654201

Camino: Salida.
Ingrese función deseada:
1. Calculadora Clásica
2. Calculadora de Fracciones
3. Calculadora de Conversiones
4. Salir
Función: 4
>>>

**Particiones de equivalencia:**

**Clases Válidas:**

- Calculadora clásica: numero flotantes.
- Calculadora de fracciones: fracciones ingresadas x/y o enteros.
- Calculadora de conversiones: números enteros.

**Clases no Válidas:**

- Para las tres calculadoras: Caracteres o cadenas de caracteres.
- Calculadora clásica: cualquier ingreso distinto de números flotantes.
- Calculadora de fracciones: cualquier ingreso distinto a fracciones x/y o enteros.
- Calculadora de conversiones: cualquier ingreso distinto a números enteros.

**Valores Límite:**

- El limite será el de la capacidad de procesamiento de la pc donde se ejecute.

**Casos de prueba para clases no válidas:**

**Pruebas de caja negra**

**Calculadora clásica**

|Ingrese función deseada: |  
|1. Calculadora Clásica |  
|2. Calculadora de Fracciones |  
|3. Calculadora de Conversiones |  
|4. Salir |  
|  
Función: 1  
Ingrese número: s  
Ingrese un número válido: 25  
Ingrese un operando (+, -, \*, /, =): /  
Ingrese número: 25as  
Ingrese un número válido: 14.12  
Ingrese un operando (+, -, \*, /, =): 24e  
Ingrese un operando válido (+, -, \*, /, =): =  
Resultado: 1.7705382436260624

|Ingrese función deseada: |  
|1. Calculadora Clásica |  
|2. Calculadora de Fracciones |  
|3. Calculadora de Conversiones |  
|4. Salir |  
|  
Función:

**Calculadora de fracciones**

|Ingrese función deseada: |  
|1. Calculadora Clásica |  
|2. Calculadora de Fracciones |  
|3. Calculadora de Conversiones |  
|4. Salir |  
|  
Función: 2  
Calculadora de Fracciones  
Operaciones disponibles:  
1: Suma  
2: Resta  
3: Multiplicación  
4: División  
1  
Ingrese numero:2/5  
Ingrese numero:54d  
El valor no es valido  
La suma de fracciones no pudo ser procesada...

Ingrese función deseada: |  
1. Calculadora Clásica |  
2. Calculadora de Fracciones |  
3. Calculadora de Conversiones |  
4. Salir |

Función: 2  
Calculadora de Fracciones  
Operaciones disponibles:  
1: Suma  
2: Resta  
3: Multiplicación  
4: División

a  
Ingrese una opción válida.

Ingrese función deseada: |  
1. Calculadora Clásica |  
2. Calculadora de Fracciones |  
3. Calculadora de Conversiones |  
4. Salir |

Función: 2  
Calculadora de Fracciones  
Operaciones disponibles:  
1: Suma  
2: Resta  
3: Multiplicación  
4: División

1  
Ingrese numero:as  
El valor no es valido

Ingrese función deseada: |  
1. Calculadora Clásica |  
2. Calculadora de Fracciones |  
3. Calculadora de Conversiones |  
4. Salir |

Función: 2  
Calculadora de Fracciones  
Operaciones disponibles:  
1: Suma  
2: Resta  
3: Multiplicación  
4: División

1  
Ingrese numero:25  
Ingrese numero:23\*  
El valor no es valido  
La suma de fracciones no pudo ser procesada...

Ingrese función deseada: |  
1. Calculadora Clásica |  
2. Calculadora de Fracciones |

|3. Calculadora de Conversiones |  
|4. Salir |  
|  
Función: 2  
Calculadora de Fracciones  
Operaciones disponibles:  
1: Suma  
2: Resta  
3: Multiplicación  
4: División  
1  
Ingrese numero:2/5  
Ingrese numero:3/5  
Ingrese numero:1/2  
Ingrese numero:56a  
El valor no es valido  
La suma de fracciones no pudo ser procesada...

|Ingrese función deseada: |  
|1. Calculadora Clásica |  
|2. Calculadora de Fracciones |  
|3. Calculadora de Conversiones |  
|4. Salir |  
|

Función: 2  
Calculadora de Fracciones  
Operaciones disponibles:  
1: Suma  
2: Resta  
3: Multiplicación  
4: División  
1  
Ingrese numero:0.25  
El valor no es valido

|Ingrese función deseada: |  
|1. Calculadora Clásica |  
|2. Calculadora de Fracciones |  
|3. Calculadora de Conversiones |  
|4. Salir |  
|

Función:

### Calculadora de conversiones

|Ingrese función deseada: |  
|1. Calculadora Clásica |  
|2. Calculadora de Fracciones |  
|3. Calculadora de Conversiones |  
|4. Salir |  
|

Función: 3

Ingrese el número a convertir: 12s4w

Ingrese un valor válido (entero positivo): sasa22

Ingrese un valor válido (entero positivo): 25

Seleccione base a la que quiere convertir: |

1. Bin |

2. Hex |

3. Oct |

|

5

Seleccione una opción válida (1-3): 3

25 en base 8 es: 31

Ingrese función deseada: |

1. Calculadora Clásica |

2. Calculadora de Fracciones |

3. Calculadora de Conversiones |

4. Salir |

|

Función: 3

Ingrese el número a convertir: 1548948875415484845544668486468486484684864

Seleccione base a la que quiere convertir: |

1. Bin |

2. Hex |

3. Oct |

|

1

1548948875415484845544668486468486484684864 en base 2 es:

1000111000111111100111001111000100110000101010011110111100111110000101  
0011000111101010111011000110000111001101111101010001110111000001000000

Ingrese función deseada: |

1. Calculadora Clásica |

2. Calculadora de Fracciones |

3. Calculadora de Conversiones |

4. Salir |

|

Función: 3

Ingrese el número a convertir:

54484894894894848948541654684984s84984894984894894

Ingrese un valor válido (entero positivo):

6548498489498498489489498489441654874654168

Seleccione base a la que quiere convertir: |

1. Bin |

2. Hex |

3. Oct |

|

2

6548498489498498489489498489441654874654168 en base 16 es:

4B2C4EB52B836C7F29CD13BBC057C23D45D8



Ingrese función deseada:	
1. Calculadora Clásica	
2. Calculadora de Fracciones	
3. Calculadora de Conversiones	
4. Salir	
Función:	