

# MEMORIA PRÁCTICA CALCULADORA



Juan Ariza Onecha, Iván Royo Gutierrez, Raquel Pérez Ariza,  
Mario Gómez Peña y Daniel Buxton Sierras

## ÍNDICE DE CONTENIDOS

1. Introducción .....	3
2. Fases del proyecto .....	3
2.1. Preparación del proyecto .....	3
2.2. Distribución de las tareas .....	3
2.3. Ejecución de 'tests' .....	4
3. Programa calculadora.....	4
3.1. Pruebas.....	5
3.1.1. Pruebas función suma .....	5
3.1.2. Pruebas función resta .....	7
3.1.3. Pruebas función producto .....	9
3.1.4. Pruebas función división.....	11
3.1.5. Pruebas secuencia de Fibonacci.....	13
3.1.6. Pruebas errores del programa .....	15
4. Bibliografía .....	16

## 1. Introducción

El objetivo del siguiente trabajo es la realización de una calculadora mediante el lenguaje ensamblador que utilizan los computadores con una arquitectura de "Microprocessor Without Interlocked Pipeline (MIPS)". Las funciones que puede realizar es la suma, resta, multiplicación y división entre un número entero y número real además de obtener los valores de la secuencia de Fibonacci, recibiendo como parámetro de entrada un número entero.

## 2. Fases del proyecto

Para la preparación de este proyecto cada uno de los integrantes del grupo decidió realizar una prueba del código escrita en los lenguajes de programación 'C' y 'C++' con el objetivo de que el producto final tuviera un comportamiento similar a los prototipos planteados. A continuación, se comparte de manera gráfica las aportaciones de cada uno de los participantes de este proyecto:

*Propuesta\_JuanAriza.jpg* → Juan Ariza Onecha

*Propuesta\_IvanRoyo.pdf* → Iván Royo Gutiérrez

*Propuesta\_RaquelPerez.pdf* → Raquel Pérez Ariza

*Propuesta\_MarioGomez.pdf* → Mario Gómez Peña

*Propuesta\_DanielBuxton.pdf* → Daniel Buxton Sierras

### Fases del proyecto

Una vez que presentamos nuestras propuestas, procedimos a distribuir el trabajo siguiendo las pautas que mostramos seguidamente. Hemos de remarcar que las instrucciones en el lenguaje ensamblador de la arquitectura MIPS han sido escritas basándonos en el estándar o32 y las pruebas del proyecto se han realizado sobre el software 'QtSpim'.

### 2.1. Preparación del proyecto

Tal y como se nos presentó en la asignatura, hemos tenido que hacer una preparación previa en lenguajes de alto nivel para posteriormente traducirla al lenguaje ensamblador, lenguaje de bajo nivel, en la arquitectura de ordenadores MIPS.

Una vez entendido nuestro código en alto nivel, nuestro grupo se reunió para hacer una puesta en común de cómo realizar dicho trabajo.

### 2.2. Distribución de las tareas

Tras la reunión inicial que realizamos, llegamos a la siguiente conclusión respecto a la distribución de tareas:

- Iván: Menú, suma, resta, producto, pila, revisión del cumplimiento del o32
- Juan: Menú, suma, Fibonacci, pila y revisión del cumplimiento del o32
- Raquel: División, menú, Fibonacci y revisión del cumplimiento del o32

- Daniel: Suma, división, Fibonacci, gestión de pila, revisión del cumplimiento del o32
- Mario: Suma, división, Fibonacci, gestión de pila, memoria y revisión del cumplimiento del o32

**\*\* Anotación:** a pesar de que cada uno previamente teníamos asignadas varias tareas, todos los miembros del grupo han tenido que participar en cada una de las tareas propuestas en el proyecto.

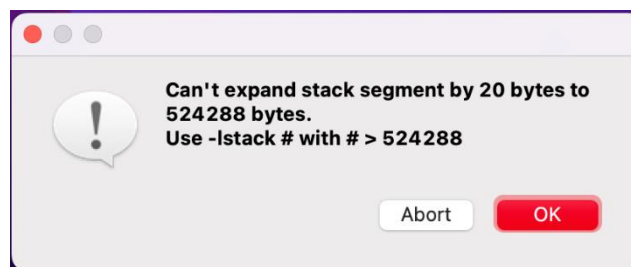
### 2.3. Ejecución de 'tests'

Para comprobar el correcto funcionamiento de cada una de las secciones del programa, lo que hemos llevado a cabo ha sido la creación de distintos archivos que verifican el cumplimiento de los requisitos mediante una serie de pruebas o 'tests'. A posteriori, se adjuntan los 3 archivos que nos han servido como base del producto final.

*Menu.txt:* inicialmente tuvimos una serie de problemas a la hora de leer el carácter '.' introducido por el usuario. Por esta razón, vimos necesario la creación de este fichero para realizar una serie de pruebas que no modificaran el proyecto principal con el objeto de arreglarlo.

*Suma.txt:* nos enfrentamos a una serie de errores a la hora de convertir el número entero introducido por el usuario en un valor de tipo float. Este incidente pudimos solventarlo con la creación de este módulo.

*Fibonacci\_prueba.txt:* el mayor incidente que tuvimos fue a la hora de crear la secuencia de Fibonacci. Nos apareció el siguiente error referido a la gestión de la pila.



*Captura de pantalla del error "Can't expand stack segment by 20 bytes to 524388 bytes"*

El siguiente problema nos estaba indicando que estábamos gestionando de forma indebida la pila. De manera similar a los 2 incidentes anteriores, decidimos extraer ese código del archivo principal y lo gestionamos individualmente.

## 3. Programa calculadora

A continuación, adjuntamos un breve esquema que hemos seguido para la creación de los valores introducidos en el segmento de datos, los argumentos introducidos y devueltos en las rutinas y las fotos de las pruebas que hemos realizado a la hora de comprobar las funciones de nuestra calculadora.

### **3.1. Pruebas**

#### **3.1.1. Pruebas función suma**

Programa CALCULADORA

Pulse la inicial para seleccionar operación:

<S>uma

<R>esta

<P>roducto

<D>ivisión

<F>ibonacci

> S

Introduzca un valor entero: 15

Introduzca un valor real: 10.0

El resultado es: 25.00000000

Programa CALCULADORA

Pulse la inicial para seleccionar operación:

<S>uma

<R>esta

<P>roducto

<D>ivisión

<F>ibonacci

> S

Introduzca un valor entero: 10

Introduzca un valor real: -5.0

El resultado es: 5.00000000

Programa CALCULADORA

Pulse la inicial para seleccionar operación:

<S>uma

<R>esta

<P>roducto

<D>ivisión

<F>ibonacci

> S

Introduzca un valor entero: -10

Introduzca un valor real: -15.0

El resultado es: -25.00000000

### **3.1.2. Pruebas función resta**

Programa CALCULADORA

Pulse la inicial para seleccionar operación:

<S>uma

<R>esta

<P>roducto

<D>ivisión

<F>ibonacci

> R

Introduzca un valor entero: 20

Introduzca un valor real: 13.0

El resultado es: 7.00000000

Programa CALCULADORA

Pulse la inicial para seleccionar operación:

<S>uma

<R>esta

<P>roducto

<D>ivisión

<F>ibonacci

> R

Introduzca un valor entero: -22

Introduzca un valor real: -12.0

El resultado es: 12.00000000

Programa CALCULADORA

Pulse la inicial para seleccionar operación:

<S>uma

<R>esta

<P>roducto

<D>ivisión

<F>ibonacci

> R

Introduzca un valor entero: 23

Introduzca un valor real: -11.0

El resultado es: 34.00000000



### **3.1.3. Pruebas función producto**

Programa CALCULADORA

Pulse la inicial para seleccionar operación:

<S>uma

<R>esta

<P>roducto

<D>ivisión

<F>ibonacci

> P

Introduzca un valor entero: 7

Introduzca un valor real: 9.1

El resultado es: 63.70000458

Programa CALCULADORA

Pulse la inicial para seleccionar operación:

<S>uma

<R>esta

<P>roducto

<D>ivisión

<F>ibonacci

> P

Introduzca un valor entero: 19

Introduzca un valor real: -13.4

El resultado es: -254.59999084

Programa CALCULADORA

Pulse la inicial para seleccionar operación:

<S>uma

<R>esta

<P>roducto

<D>ivisión

<F>ibonacci

> P

Introduzca un valor entero: -17

Introduzca un valor real: -13.4

El resultado es: 227.79998779

### **3.1.4. Pruebas función división**

Programa CALCULADORA

Pulse la inicial para seleccionar operación:

<S>uma

<R>esta

<P>roducto

<D>ivisión

<F>ibonacci

> D

Introduzca un valor entero: 2

Introduzca un valor real: 0.5

El resultado es: 4.00000000

Programa CALCULADORA

Pulse la inicial para seleccionar operación:

<S>uma

<R>esta

<P>roducto

<D>ivisión

<F>ibonacci

> D

Introduzca un valor entero: 2

Introduzca un valor real: -1.5

El resultado es: -1.33333337

Programa CALCULADORA

Pulse la inicial para seleccionar operación:

<S>uma

<R>esta

<P>roducto

<D>ivisión

<F>ibonacci

> D

Introduzca un valor entero: -5

Introduzca un valor real: -1.2

El resultado es: 4.16666651

### 3.1.5. Pruebas secuencia de Fibonacci

Programa CALCULADORA

Pulse la inicial para seleccionar operación:

<S>uma

<R>esta

<P>roducto

<D>ivisión

<F>ibonacci

> F

Introduzca un valor entero: 7

El resultado es: 13

Pulse la inicial para seleccionar operación:

<S>uma

<R>esta

<P>roducto

<D>ivisión

<F>ibonacci

> F

Introduzca un valor entero: 23

El resultado es: 28657

Pulse la inicial para seleccionar operación:

<S>uma

<R>esta

<P>roducto

<D>ivisión

<F>ibonacci

> F

Introduzca un valor entero: 10

El resultado es: 55

### **3.1.6. Pruebas errores del programa**

Programa CALCULADORA

Pulse la inicial para seleccionar operación:

<S>uma

<R>esta

<P>producto

<D>ivisión

<F>ibonaci

> y

ERROR. DATO INTRODUCIDO NO VÁLIDO

#### **4. Bibliografía**

[1] Apuntes del profesor 'Gabriel Mariano Marro Gros', Plataforma Docente Universitaria, Universidad San Jorge "Arquitectura de Ordenadores I"



