

# Reporte de práctica 3

Alejandro Guirado García

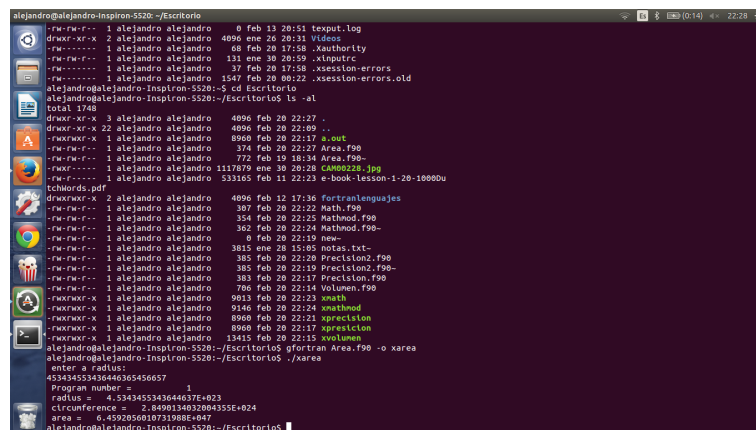
24 de Febrero del 2015

## ■ Introducción

El presente trabajo es la actividad numero tres del curso de Programación y lenguaje Fortran impartido en la Universidad de Sonora. En la cual, calcularemos el area de un circulo, el volumen de una sección esférica, la precisión numérica de una máquina de 4 y 8 bits, funciones trigonométricas y una subrutina.

## ■ Área del círculo

Este programa realiza los cálculos para obtener el área de un círculo con la propiedad de que podemos elegir el radio deseado.



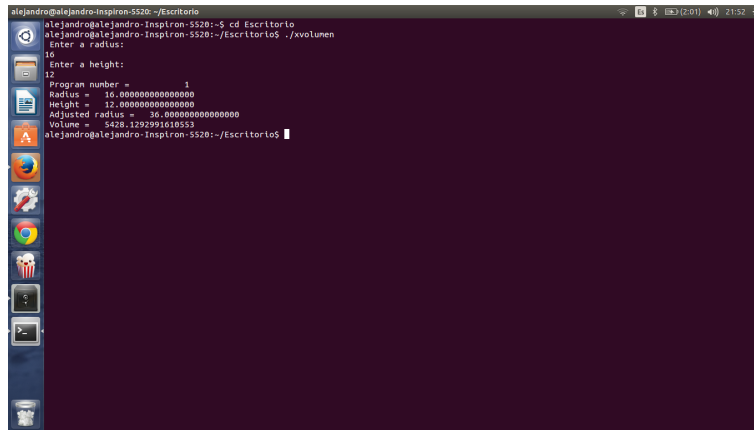
The screenshot shows a terminal window with a dark background. The top part displays the output of a 'ls -al' command, listing various files and their permissions, owners, sizes, and dates. The bottom part shows the execution of a Fortran program named 'circle\_area'. The program prompts the user to enter a radius, and the output displays the calculated circumference and area for a radius of 1.0.

```
alejandro@alejandro-Inspiron-5520:~/Escritorio$ ls -al
total 1748
drwxr-xr-x 3 alejandro alejandro 4096 feb 20 22:27 .
drwxr-xr-x 22 alejandro alejandro 4096 feb 20 22:09 ..
-rw-rw-r-- 1 alejandro alejandro 8960 feb 20 22:17 a.out
-rw-rw-r-- 1 alejandro alejandro 374 feb 20 22:27 Area.f90
-rw-rw-r-- 1 alejandro alejandro 772 feb 19 18:36 Area.f90~
-rw-rw-r-- 1 alejandro alejandro 1117879 ene 30 20:28 CAM00228.jpg
-rw-rw-r-- 1 alejandro alejandro 533165 feb 11 22:23 e-book-lesson-1-20-1000bu
tochords.pdf
drwxr-xr-x 2 alejandro alejandro 4096 feb 12 17:36 fortranlenguajes
-rw-rw-r-- 1 alejandro alejandro 387 feb 20 22:22 math.f90
-rw-rw-r-- 1 alejandro alejandro 354 feb 20 22:25 Mathmod.f90
-rw-rw-r-- 1 alejandro alejandro 362 feb 20 22:24 Mathmod.f90~
-rw-rw-r-- 1 alejandro alejandro 0 feb 20 22:18 new
-rw-rw-r-- 1 alejandro alejandro 3815 ene 28 15:05 notas.txt
-rw-rw-r-- 1 alejandro alejandro 385 feb 20 22:20 Precision2.f90
-rw-rw-r-- 1 alejandro alejandro 385 feb 20 22:19 Precision2.f90~
-rw-rw-r-- 1 alejandro alejandro 383 feb 20 22:17 Precision.f90
-rw-rw-r-- 1 alejandro alejandro 786 feb 20 22:24 Volumen.f90
-rw-rw-r-- 1 alejandro alejandro 9013 feb 20 22:23 wmath
-rw-rw-r-- 1 alejandro alejandro 9146 feb 20 22:24 wmathmod
-rw-rw-r-- 1 alejandro alejandro 8960 feb 20 22:21 wprecision
-rw-rw-r-- 1 alejandro alejandro 8960 feb 20 22:17 wprecision
-rw-rw-r-- 1 alejandro alejandro 13415 feb 20 22:15 wvolumen
alejandro@alejandro-Inspiron-5520:~/Escritorio$ ./xarea
enter a radius:
1
Program number = 1
radius = 4.53455343644637E+023
circumference = 2.8490134032004355E+024
area = 8.450265081073198E+047
alejandro@alejandro-Inspiron-5520:~/Escritorio$
```

```
Program circle_area
Implicit None
Real *8 :: radius, circum, area
Real *8 :: PI = 4.0 * atan(1.0)
Integer :: model_n = 1
print *, 'enter a radius:'
read (*,*) radius
circum = 2.0 * PI * radius
area = radius * radius * PI
print *, 'Program number =', model_n
print *, 'radius =', radius
print *, 'circumference =', circum
print *, 'area =', area
end Program circle_area
```

## ■ Volumen de una sección esférica.

Este Programa de manera similar al anterior, pide un radio al usuario. Pero además debemos elegir una altura. Con estos dos datos el programa nos muestra el volumen.



! Volumen . f90 : Calcula el volumen de una region esferica

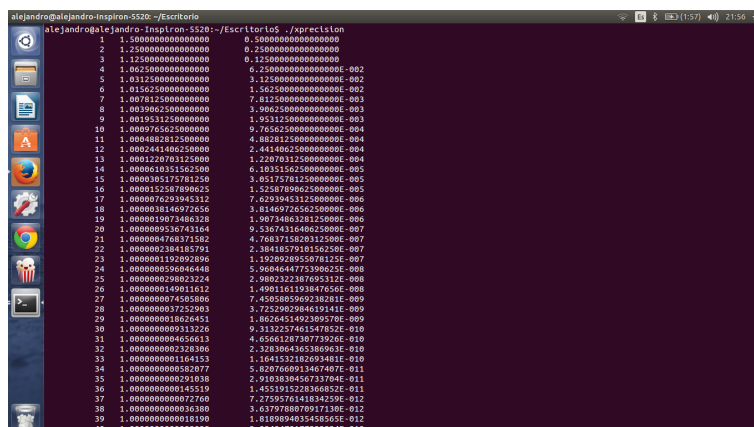
```

Program volumenregion
Implicit None
Real *8 :: radius , radiusx , volume , height
Real *8 :: PI = 4.0 * atan(1.0)
Integer :: model_n = 1
print * , 'Enter a radius:'
read * , radius
print * , 'Enter a height:'
read * , height
radiusx = 3.00 * radius - height
volume = 0.3333 * PI * height * height * radiusx
print * , 'Program number =' , model_n
print * , 'Radius =' , radius
print * , 'Height =' , height
print * , 'Adjusted radius =' , radiusx
print * , 'Volume =' , volume
End Program volumenregion

```

- Determinando la precisión de la máquina.

El programa determina la precisión utilizando una herramienta de 8 bits.



! Precision . f90 : Determina la precision de la computadora

```

Program Precision
Implicit None
Integer :: i , n
Real *8 :: epsilon_m , one

```

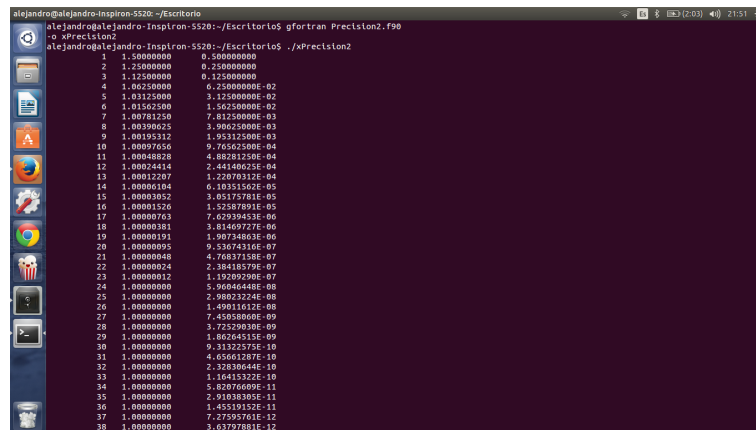
```

n=60
epsilon_m = 1.0
one = 1.0
do i = 1, n , 1
epsilon_m = epsilon_m / 2.0
one = 1.0 + epsilon_m
print * , i , one , epsilon_m
end do
End Program Precision

```

- Determinando la precisión de la máquina de 4.

El programa determina la precisión utilizando una herramienta de 4 bits.



```

alejandro@alejandro-inspiron-5520:~/Escritorio
$ ./Precision2
alejandro@alejandro-inspiron-5520:~/Escritorio$ ./Precision2
1 1.50000000 0.50000000
2 1.25000000 0.25000000
3 1.12500000 0.12500000
4 1.06250000 0.06250000e-02
5 1.03125000 3.12500000e-02
6 1.01562500 1.56250000e-02
7 1.00781250 7.81250000e-03
8 1.00390625 3.90625000e-03
9 1.00195312 1.95312500e-03
10 1.00097656 9.76562500e-04
11 1.00048828 4.88281250e-04
12 1.00024414 2.44140625e-04
13 1.00012207 1.22070312e-04
14 1.00006104 6.10351562e-05
15 1.00003052 3.05175781e-05
16 1.00001526 1.52587891e-05
17 1.00000763 7.62939453e-06
18 1.00000381 3.81469727e-06
19 1.00000191 1.90734863e-06
20 1.00000095 9.53674316e-07
21 1.00000048 4.76837158e-07
22 1.00000024 2.38418579e-07
23 1.00000012 1.19209289e-07
24 1.00000000 5.96046448e-08
25 1.00000000 2.98023224e-08
26 1.00000000 1.49011612e-08
27 1.00000000 7.45058060e-09
28 1.00000000 3.72529030e-09
29 1.00000000 1.86264515e-09
30 1.00000000 9.31322575e-10
31 1.00000000 4.65661287e-10
32 1.00000000 2.32830644e-10
33 1.00000000 1.16415322e-10
34 1.00000000 5.82076609e-11
35 1.00000000 2.91038305e-11
36 1.00000000 1.45519152e-11
37 1.00000000 7.27595761e-12
38 1.00000000 3.63797881e-12

```

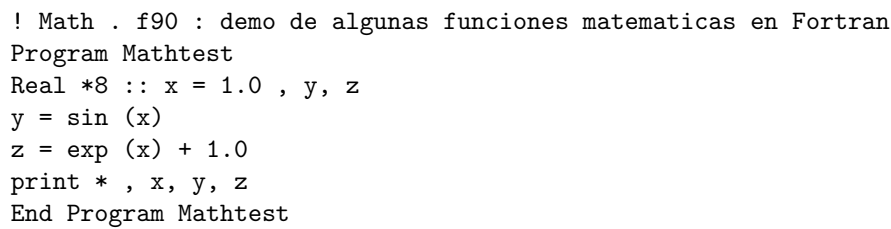
```

! Precision . f90 : Determina la precision de la computadora
Program Precision2
Implicit None
Integer :: i , n
Real *4 :: epsilon_m , one
n=60
epsilon_m = 1.0
one = 1.0
do i = 1, n , 1
epsilon_m = epsilon_m / 2.0
one = 1.0 + epsilon_m
print * , i , one , epsilon_m
end do
End Program Precision2

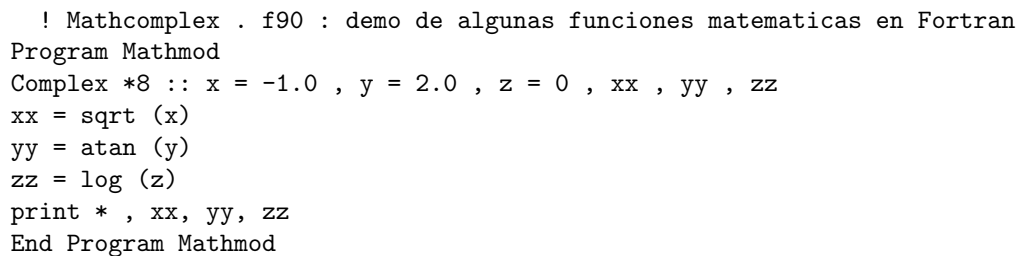
```

- Funciones trigonométricas y exponenciales.

El programa realiza los cálculos pertinentes para mostrarnos el resultado de estas operaciones.



- El programa realiza de manera similiar al anterior, pero en valores indefinidos. Por lo cual, marcan error.



- 4

```

alejandro@alejandro-inspiron-5520:~/Escritorio
alejandro@alejandro-inspiron-5520:~$ cd Escritorio
alejandro@alejandro-inspiron-5520:~/Escritorio$ ./xfunction
f(Xin, Yin) = 1.4794255386842030
alejandro@alejandro-inspiron-5520:~/Escritorio$

```

El programa solamente es un ejemplo de una función de dos variables.

```

! Funcion . f90 : Creando funciones
Real *8 Function f (x,y)
Implicit None
Real *8 :: x, y
f = 1.0 + sin (x*y )
End Function f
Program Main
Implicit None
Real *8 :: Xin =0.25 , Yin =2. , c , f
c = f ( Xin , Yin )
write ( * , * ) 'f(Xin, Yin) = ' , c
End Program Main

```

#### ■ Subrutina

Es un ejemplo de una subrutina.

```

alejandro@alejandro-inspiron-5520:~/Escritorio
alejandro@alejandro-inspiron-5520:~$ cd Escritorio
alejandro@alejandro-inspiron-5520:~/Escritorio$ ./xsubrutina
The answers are: 1.4794255386842030 2.18869999242743364
alejandro@alejandro-inspiron-5520:~/Escritorio$

```

```

! Subrutina . f90 : Demuestra la llamada de una subrutina
Subroutine g(x, y, ans1 , ans2 )
Implicit None
Real (8) :: x , y , ans1 , ans2
ans1 = sin (x*y) + 1.
ans2 = ans1**2
End Subroutine g
Program Mainprogram
Implicit None

```

```
Real *8 :: Xin =0.25 , Yin =2.0 , Gout1 , Gout2
call g( Xin , Yin , Gout1 , Gout2 )
write ( * , *) 'The answers are: ' , Gout1 , Gout2
End Program Mainprogram
```