

Reporte de práctica 3

Alejandro Guirado García

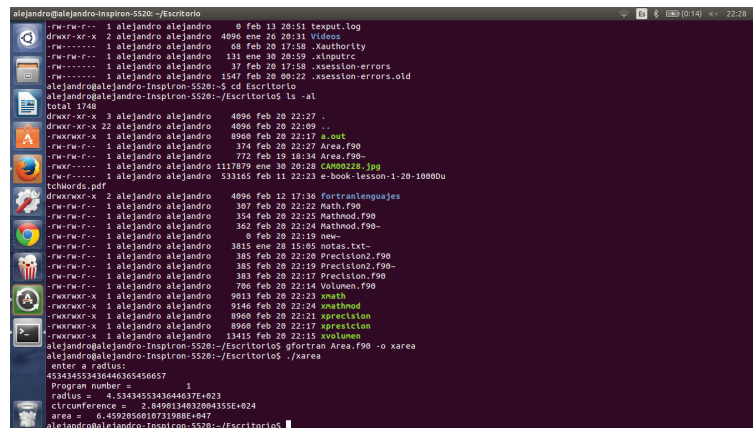
24 de Febrero de 2015

- 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 seccion esferica, la precision numerica de una maquina de 4 y 8 bits, funciones trigonometricas y una subrutina.

- Area del circulo

Este programa realiza los calculos para obtener el area de un circulo con la propiedad de que podemos elegir el radio deseado.



```
alejandro@alejandro-inspiron-5520:~/Escritorio
-rw-rw-r-- 1 alejandro alejandro 0 feb 13 20:51 texput.log
drwxr-xr-x 2 alejandro alejandro 4096 ene 20 20:21 videos
-rw-r--r-- 1 alejandro alejandro 68 feb 20 17:58 .xauthority
-rw-rw-r-- 1 alejandro alejandro 131 ene 30 20:59 .xinputrc
-rw-r--r-- 1 alejandro alejandro 37 feb 20 17:58 .xsession-errors
-rw-r--r-- 1 alejandro alejandro 1547 feb 20 00:22 .xsession-errors.old
alejandro@alejandro-inspiron-5520:~$ cd Escritorio
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 ..
-rwxrwxr-x 1 alejandro alejandro 8960 feb 20 22:17 a.out
-rw-rw-r-- 1 alejandro alejandro 374 feb 20 22:22 Area.f90
-rw-rw-r-- 1 alejandro alejandro 772 feb 19 18:34 Area.f90~
-rwxr-xr-x 1 alejandro alejandro 1117879 ene 30 20:28 GIMM0223.jpg
-rw-r--r-- 1 alejandro alejandro 533565 feb 11 22:23 e-book-lesson-1-20-1000du
tchWords.pdf
drwxrwxr-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 22:19 new~
-rw-rw-r-- 1 alejandro alejandro 3815 ene 20 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 385 feb 20 22:19 Precision.f90
-rw-rw-r-- 1 alejandro alejandro 706 feb 20 22:14 Volumen.f90
-rwxrwxr-x 1 alejandro alejandro 9013 feb 20 22:23 smath
-rwxrwxr-x 1 alejandro alejandro 9146 feb 20 22:24 smathmod
-rwxrwxr-x 1 alejandro alejandro 8960 feb 20 22:21 sprecision
-rwxrwxr-x 1 alejandro alejandro 8960 feb 20 22:17 sprecision
-rwxrwxr-x 1 alejandro alejandro 13451 feb 20 22:15 svolumes
alejandro@alejandro-inspiron-5520:~/Escritorio$ gfortran Area.f90 -o xarea
alejandro@alejandro-inspiron-5520:~/Escritorio$ ./xarea
enter a radius:
453434553436446365456657
Program number = 1
radius = 4.5343455343644637E+023
circumference = 2.8496134032064355E+024
area = 6.450205691073198E+047
alejandro@alejandro-inspiron-5520:~/Escritorio$
```

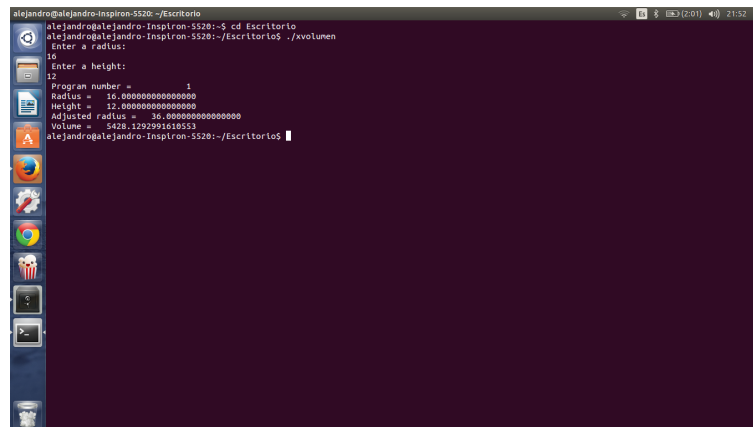
```

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 seccion esferica

Este Programa de manera similar al anterior, pide un radio al usuario. Pero ademas 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)

```

- Determinando la precision de la maquina

```
alejandro@alejandro-Inspiron-5520: ~$ ./Escriorio
alejandro@alejandro-Inspiron-5520: ~$ ./Escriorio105 ./aprecion
1 1.3600000000000000 0.3600000000000000
2 1.2500000000000000 0.2500000000000000
3 0.1200000000000000 0.1200000000000000
4 1.0025000000000000 0.2500000000000000-002
5 1.0312500000000000 3.1250000000000000-002
6 0.1016250000000000 1.5625000000000000-002
7 1.0078125000000000 7.8125000000000000-003
8 1.0039062500000000 3.9062500000000000-003
9 1.0019531250000000 1.9531250000000000-003
10 1.0009765625000000 9.7656250000000000-004
11 1.0004882812500000 4.8828125000000000-004
12 1.0002441406250000 2.4414062500000000-004
13 1.0001220781250000 1.2207812500000000-004
14 1.0000610351562500 6.1035156250000000-005
15 1.0000305178125000 3.0517812500000000-005
16 1.0000152507890625 1.5250789062500000-005
17 1.0000076293945312 7.6293945312500000-006
18 1.0000038146972656 3.8146972656250000-006
19 1.0000019073406250 1.9073406250000000-006
20 1.0000009536743164 9.5367431640625000-007
21 1.0000004768371582 4.7683715820125000-007
22 1.0000002384185781 2.3841857810156250-007
23 1.0000001192089296 1.19208928955078125-007
24 1.0000000596046418 5.9604641773390625-008
25 1.0000000298023224 2.98029802367905312-008
26 1.0000000149011612 1.4901161193847656-008
27 1.0000000074513068 7.451306859238281-009
28 1.0000000037252983 3.75298096491941-009
29 1.0000000018626451 1.8626451492395978-009
30 1.0000000009313226 9.31322675461547852-010
31 1.0000000004656613 4.65661287397326-010
32 1.000000000232836 2.32836465389678-010
33 1.0000000001164513 1.164513282690481E-010
34 1.0000000000582077 5.820766913746707E-011
35 1.0000000000291038 2.910383085673374E-011
36 1.0000000000145519 1.4551515228366852E-011
37 1.0000000000072760 7.275976181834259E-012
38 1.0000000000036390 3.6379788870917130E-012
39 1.0000000000018199 1.818998045538505E-012
40 1.0000000000009095 9.09497017722824E-013
```

3

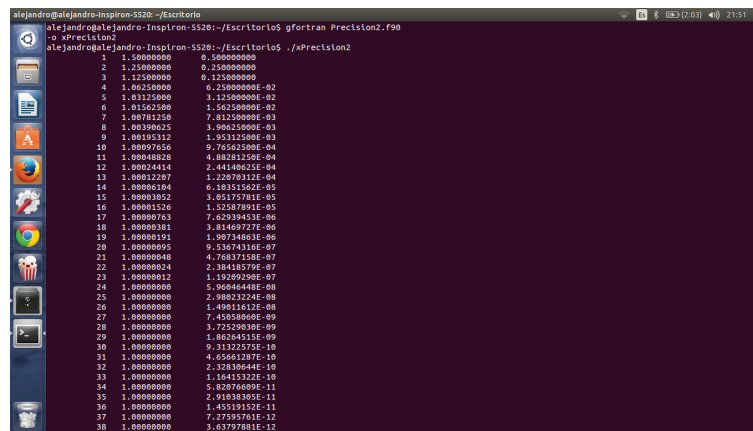
```

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 precision de la maquina de 4

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



```

alejandro@alejandro-Inspiron-5520:~/Escritorio$ gfortran Precision2.f90
-o xPrecision2
alejandro@alejandro-Inspiron-5520:~/Escritorio$ ./xPrecision2
1 1.0000000 0.50000000
2 1.2500000 0.25000000
3 1.1250000 0.12500000
4 1.0625000 0.06250000E-02
5 1.0312500 3.1250000E-02
6 1.0156250 1.5625000E-02
7 1.0078125 7.8125000E-03
8 1.0039062 3.9062500E-03
9 1.0019531 1.9531250E-03
10 1.0009766 9.7656250E-04
11 1.0004883 4.8828125E-04
12 1.0002441 2.4414062E-04
13 1.0001220 1.2207031E-04
14 1.0000610 6.1035156E-05
15 1.0000305 3.0517578E-05
16 1.0000152 1.5258789E-05
17 1.0000076 7.6293945E-06
18 1.0000038 3.8146972E-06
19 1.0000019 1.9073486E-06
20 1.0000009 9.5367431E-07
21 1.0000004 4.7683715E-07
22 1.0000002 2.3841857E-07
23 1.0000001 1.1920929E-07
24 1.0000000 5.9604648E-08
25 1.0000000 2.9802324E-08
26 1.0000000 1.4901162E-08
27 1.0000000 7.4505809E-09
28 1.0000000 3.7252903E-09
29 1.0000000 1.8626451E-09
30 1.0000000 9.3132257E-10
31 1.0000000 4.6566128E-10
32 1.0000000 2.3283064E-10
33 1.0000000 1.1641532E-10
34 1.0000000 5.8207660E-11
35 1.0000000 2.9103830E-11
36 1.0000000 1.4551915E-11
37 1.0000000 7.2759576E-12
38 1.0000000 3.6379788E-12

```

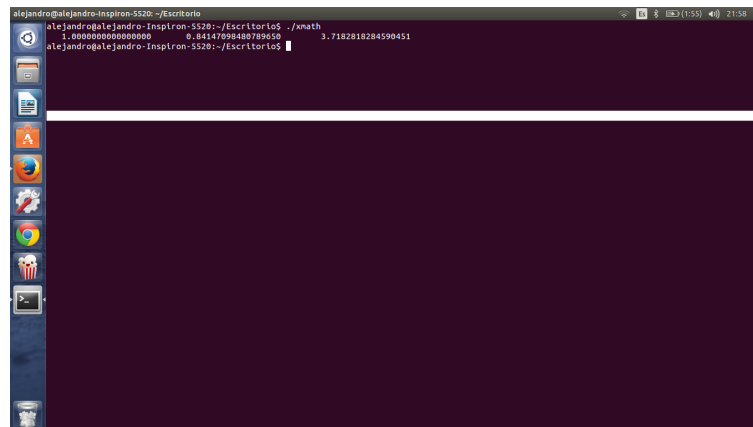
```

! 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 trigonometricas y exponenciales

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

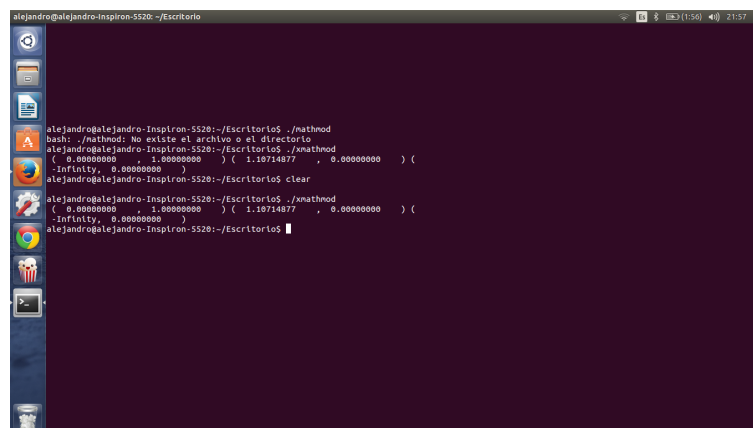


```
alejandro@alejandro-inspiron-5520:~/Escritorio$ ./xmath
1.0000000000000000      0.84147098480789650      3.7182818284590451
alejandro@alejandro-inspiron-5520:~/Escritorio$
```

```
! Math . f90 : demo de algunas funciones matematicas en Fortran
Program Mathtest
Real *8 :: x = 1.0 , y, z
y = sin (x)
z = exp (x) + 1.0
print * , x, y, z
End Program Mathtest
```

- Funciones erroneas

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



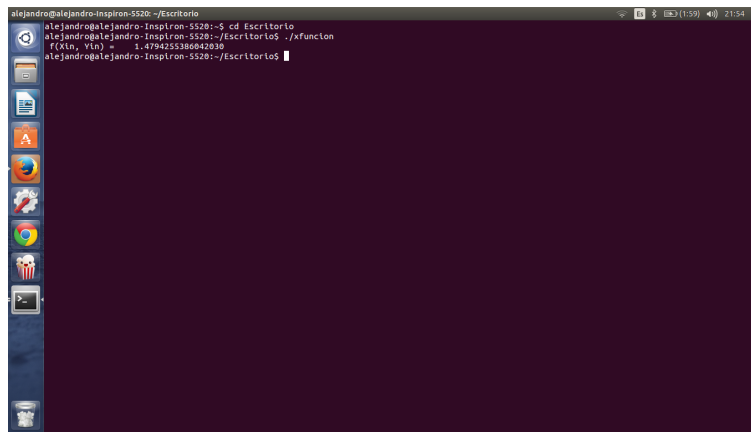
```
alejandro@alejandro-inspiron-5520:~/Escritorio$ ./mathmod
bash: ./mathmod: No existe el archivo o el directorio
alejandro@alejandro-inspiron-5520:~/Escritorio$ ./mathmod
( 0.00000000 , 1.00000000 ) ( 1.10714877 , 0.00000000 ) (
-infinity, 0.00000000
)
alejandro@alejandro-inspiron-5520:~/Escritorio$ clear
alejandro@alejandro-inspiron-5520:~/Escritorio$ ./mathmod
( 0.00000000 , 1.00000000 ) ( 1.10714877 , 0.00000000 ) (
-infinity, 0.00000000
)
alejandro@alejandro-inspiron-5520:~/Escritorio$
```

```

! Mathcomplex . f90 : demo de algunas funciones matematicas en Fortran
Program Mathmod
Complex *8 :: x = -1.0 , y = 2.0 , z = 0 , xx , yy , zz
xx = sqrt (x)
yy = atan (y)
zz = log (z)
print * , xx, yy, zz
End Program Mathmod

```

- Funcion $f(x,y)$



El programa solamente es un ejemplo de una funcion de dos variables

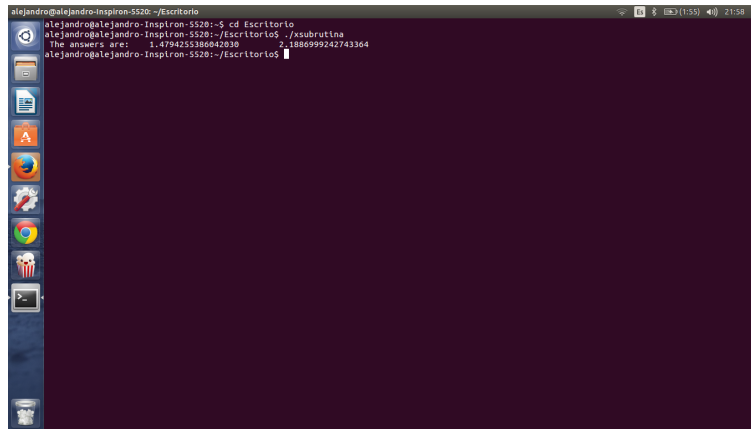
```

! 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.



```
alejandrogalejandro-Inspiron-5520 ~/Escritorio
alejandrogalejandro-Inspiron-5520:~$ cd Escritorio
alejandrogalejandro-Inspiron-5520:~/Escritorio$ ./asubrutina
The answers are: 1.4794255380642830 2.5886999242743364
alejandrogalejandro-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
```