

## Técnicas Digitales III

### Trabajo práctico: Programación usando números en punto flotante

#### Ejercicio 1

Compile el siguiente código en C en su PC:

```
#include <stdio.h>
#include <float.h>
#include <math.h>
#include <fenv.h>

int main(void)
{
    float a, b, c, r1, r2;

    a = 10000000000.0;
    b = 200000000.0;
    c = 200000000.0;

    r1 = (a * b) * c;
    r2 = a * (b * c);

    printf("r1 = %f \n", r1 );
    printf("r2 = %f \n", r2 );
    printf("error = %f \n", r1 - r2 );

    return 0;
}
```

1. Inspeccione el código y determine el objetivo del programa
2. Analice los valores de las variables `r1` y `r2`.
3. ¿Qué conclusión puede obtener a partir de estos valores?

#### Ejercicio 2

Analice y compile el archivo `ex_02.c`.

1. ¿Para qué sirven las funciones `fegetround()` y `fesetround()`?
2. ¿Cuál es el modo de redondeo por defecto con el que arranca el programa?
3. Ejecute el programa para los modos de redondeo `FE_DOWNWARD`, `FE_UPWARD` y `FE_TOWARDZERO`, y compárelos con el modo `FE_TONEAREST`.
4. ¿Observa diferencias? ¿Estas diferencias son consistentes con los modos de redondeos?

#### Ejercicio 3

Verifique que los valores de `a1` y `a2` no son iguales para un conjunto de números particulares en formato en punto flotante precisión simple:

1. `a1 = b + b * c; a2 = b * (1.0 + c);`
2. `a1 = b / 1001.0; a2 = b * 0.1001;`
3. `a1 = b / b; a2 = 1.0;`
4. `a1 = b - b; a2 = 0.0;`
5. `a1 = b + 0.0; a2 = b;`

#### Ejercicio 4

Analice y compile el archivo `ex_04.c`.

1. Analice los resultados impresos por consola ¿Son los resultados consistentes con las operaciones ejecutadas?
2. ¿Qué observa por consola? ¿Cuál es la función de `feenableexcept()`?

### **Ejercicio 5**

Analice y compile el archivo `ex_05.c`.

1. Analice cómo se implementa el manejo de excepciones por operaciones con punto flotante.
2. Descomente de a una las líneas con operaciones matemáticas. Vuelva a compilar y ejecutar.
3. ¿Que observa?