

# Trabajo práctico

## Programación usando números en formato punto flotante

### Ejercicio 1

Analice y compile el archivo `ex_01.c`.

1. Inspeccione el código y determine el objetivo del programa.
2. Analice los valores de las variables `f1`, `f2`, `d1`, `acum_1` y `acum_2`.
3. ¿Qué conclusión puede obtener a partir de estos valores?

### Ejercicio 2. Modos de redondeo

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. Valores especiales

Investigue como inicializar en C variables en punto flotante con los valores especiales NaN, Inf e -Inf. Además, escriba un programa en C que produzca como resultado estos valores especiales.

### Ejercicio 4. Excepciones

Analice y compile el archivo `ex_04.c`.

1. Analice los resultados impresos por consola ¿Son los resultados consistentes con las operaciones ejecutadas?
2. Explique qué hacen las funciones `feclearexcept()`, `feraiseexcept()` y `fetestexcept()`.

### Ejercicio 5. Excepciones

Utilizando el código visto en el ejercicio 4, `ex_04.c`, genere los 4 tipo de excepciones estudiadas pero ejecutando diferentes operaciones matemáticas en lugar de utilizar la función `feraiseexcept()`.

### Ejercicio 6. Manejo de excepciones

Analice y compile el archivo `ex_06.c`.

3. Analice los resultados impresos por consola ¿Son los resultados consistentes con las operaciones ejecutadas?
4. Descomente las líneas 38 a 43 y vuelva a compilar.
5. ¿Qué observa por consola? ¿Cuál es la función de `feenableexcept()`?
6. Descomente la línea 36 y vuelva a compilar.
7. ¿Qué observa por consola? ¿Cuál es la función de `signal(SIGFPE, fpe_handler)`?