



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA EN CIENCIAS DE LA COMPUTACIÓN

PERÍODO ACADÉMICO: 2025-A

ASIGNATURA: ICCD412 Métodos Numéricos

GRUPO: GR2

TIPO DE INSTRUMENTO: Tarea1

FECHA DE ENTREGA LÍMITE: 04/05/2025

ALUMNO: Sebastián Chicaiza

TEMA

Tipos de Errores

OBJETIVOS

- Poder identificar los diferentes tipos de errores que aparecen al aplicar métodos numéricos

DESARROLLO

Los números para que un computador pueda procesarlos son representados en binario usando memoria para almacenar cada uno de los bits que componen un número, aunque esta forma de representarlos es útil tiene limitaciones.

Los errores numéricos suelen aparecer como resultado del uso de aproximaciones para representar operaciones y cantidades matemáticas exactas [1].

Se realizó el experimento de en Python con el objetivo de observar el comportamiento cuando se almacena un número que sobrepase su capacidad de almacenamiento.

El código para lograr el desbordamiento utilizado fue:

```
import sys
print("Máximo flotante:", sys.float_info.max)
print("Mínimo flotante:", sys.float_info.min)
print("Máximo entero:", sys.maxsize)

# la variable maximo toma el valor máximo permitido por python
maximo = sys.float_info.max

print(maximo)

# error toma el valor de maximo*2 para ocasionar un inf
error = maximo *2

print(error)

# error ahora opera inf para ocasionar un Nan
error = error - error

# fatality trata de tomar el valor entero de la variable error
# para ocasionar un error en la ejecución
fatality = int(error)
print(error)
```

El resultado de la ejecución del código, fue un error al intentar de operar con el valor máximo permitido por Python. El resultado del experimento evidencia como es que un computador tiene limitaciones en el almacenamiento y como consecuencia una limitación en la precisión de las operaciones ejecutadas.



```
02:55:39 metodosNumericos
21:55:39 metodosNumericos master 77 ~1
> C:/Users/aleja/AppData/Local/Programs/Python/Python313/python.exe c:/Users/aleja/Desktop/EPN/metodosNumericos/deber.py
Máximo flotante: 1.7976931348623157e+308
Mínimo flotante: 2.2250738585072014e-308
Máximo entero: 9223372036854775807
1.7976931348623157e+308
inf
Traceback (most recent call last):
  File "c:/Users/aleja/Desktop/EPN/metodosNumericos/deber.py", line 16, in <module>
    fatality = int(error)
ValueError: cannot convert float NaN to integer
```

REFERENCIAS

- [1] S. C. Chapra, R. P. Canale, R. S. G. Ruiz, V. H. I. Mercado, E. M. Díaz, and G. E. Benites, *Métodos numéricos para ingenieros*. McGraw-Hill New York, NY, USA,

2011, vol. 5.