

# 1. Fórmula de Stirling para cálculo de factoriales

## 1.1 Contexto:

1. La fórmula de Stirling, aproxima  $n!$  de la siguiente manera:

$$n! \approx n^n \cdot e^{-n} \cdot \sqrt{2\pi \cdot n}$$

- a) Escriba un algoritmo para aproximar los valores de  $n!$  con la fórmula de Stirling.  
b) Utilice el algoritmo de la parte **a.** para aproximar  $n!$  con  $n = 0, 1, \dots, 25$ . Organice los resultados de la forma:

$n$	$n!$	Aproximación de Stirling	Error Absoluto	Error Relativo

- c) Analice los resultados

## 1.2 Librerías necesarias para trabajar.

```
In [1]: import math
import pandas as pd
```

## 1.3 Implementación de las funciones necesarias para calcular la fórmula de Stirling así como las funciones para calcular los errores absolutos y relativos.

La función `aproximación_stirling` calcula la aproximación definida en el ejercicio 1, según la premisa indicada.

La función `error_absoluto()` calcula el error absoluto de la aproximación.

La función `error_relativo()` calcula el error relativo de la aproximación.

**nf** es la variable para almacenar el número factorial.

Se define el rango que incluya de 0 a 25 según lo indicado en el inciso 2.

La condición de **x == 0** se incluye para evitar una excepción que se genera por división entre 0.

Para la condición **x == 1** se incluye para evitar que la definición del valor de **nf**, no se mantenga de manera permanente en 0 y calcule de manera correcta los subsecuentes valores factoriales.

```
In [2]: def aproximacion_stirling(n):
        return math.pow(n, n) * math.pow(math.e, -n) * math.sqrt(2 * math.pi * n)

def error_absoluto(real, aprox):
    return abs(real - aprox)

def error_relativo(error_absoluto, real):
    try:
        return abs(error_absoluto) / abs(real)
    except ZeroDivisionError:
        return None

def calculate_stirling(n):
    nf = 0
    data = []
    for x in range(n+1):
        if x == 0 or x == 1:
            nf = x
        nf *= x
        aprox = aproximacion_stirling(x)
        error_abs = error_absoluto(nf, aprox)
        error_rel = error_relativo(error_abs, nf)
        row = {'n': x, 'n!': nf, 'Aproximación de Stirling': aprox,
               'Error Absoluto': error_abs,
               'Error Relativo': error_rel}
        data.append(row)
    df = pd.DataFrame(data)
    return df
```

**1.4 Input de valores para el funcionamiento de la función principal `calculate_stirling()` definiendo que el rango  $0 < x < 25$ .**

```
In [3]: df = calculate_stirling(25)

df
```

Out[3]:

	<b>n</b>	<b>n!</b>	<b>Aproximación de Stirling</b>	<b>Error Absoluto</b>	<b>Error Relativo</b>
<b>0</b>	0	0	0.000000e+00	0.000000e+00	NaN
<b>1</b>	1	1	9.221370e-01	7.786299e-02	0.077863
<b>2</b>	2	2	1.919004e+00	8.099565e-02	0.040498
<b>3</b>	3	6	5.836210e+00	1.637904e-01	0.027298
<b>4</b>	4	24	2.350618e+01	4.938249e-01	0.020576
<b>5</b>	5	120	1.180192e+02	1.980832e+00	0.016507
<b>6</b>	6	720	7.100782e+02	9.921815e+00	0.013780
<b>7</b>	7	5040	4.980396e+03	5.960417e+01	0.011826
<b>8</b>	8	40320	3.990240e+04	4.176045e+02	0.010357
<b>9</b>	9	362880	3.595369e+05	3.343127e+03	0.009213
<b>10</b>	10	3628800	3.598696e+06	3.010438e+04	0.008296
<b>11</b>	11	39916800	3.961563e+07	3.011749e+05	0.007545
<b>12</b>	12	479001600	4.756875e+08	3.314114e+06	0.006919
<b>13</b>	13	6227020800	6.187239e+09	3.978132e+07	0.006389
<b>14</b>	14	87178291200	8.666100e+10	5.172895e+08	0.005934
<b>15</b>	15	1307674368000	1.300431e+12	7.243646e+09	0.005539
<b>16</b>	16	20922789888000	2.081411e+13	1.086755e+11	0.005194
<b>17</b>	17	355687428096000	3.539483e+14	1.739099e+12	0.004889
<b>18</b>	18	6402373705728000	6.372805e+15	2.956908e+13	0.004618
<b>19</b>	19	121645100408832000	1.211128e+17	5.323138e+14	0.004376
<b>20</b>	20	2432902008176640000	2.422787e+18	1.011516e+16	0.004158
<b>21</b>	21	51090942171709440000	5.088862e+19	2.023248e+17	0.003960
<b>22</b>	22	1124000727777607680000	1.119751e+21	4.249233e+18	0.003780
<b>23</b>	23	25852016738884976640000	2.575853e+22	9.349137e+19	0.003616
<b>24</b>	24	620448401733239439360000	6.182979e+23	2.150475e+21	0.003466
<b>25</b>	25	15511210043330985984000000	1.545959e+25	5.161521e+22	0.003328

## 1.5 Análisis de los resultados:

Según se identifica en los datos reflejados el algoritmo de stirling aproxima con cierta amplitud que va creciendo conforme el valor de referencia sea más alto.

Esa diferencia del crecimiento de amplitud se demuestra de manera clara en el valor absoluto que se vuelve más grande conforme se utiliza un número más alto que el anterior donde la aproximación no es tan cercana.

Caso contrario sucede con el error relativo, que conforme crece el valor de referencia la aproximación va mejorando, pero a un ritmo muy lento al contrario del escenario del error absoluto que crece de manera bastante marcada.