

# Analisis de algoritmos de ordenamiento

Bejar Merma Ángel Andrés  
Universidad Nacional de San Agustín  
andresbjar97@gmail.com  
Ciudad de Arequipa

## Resumen

Esta es una plantilla simple para crear un artículo L<sup>A</sup>T<sub>E</sub>X en español, con algunos comandos que se usarán frecuentemente para hacer tareas de la licenciatura en Física.

## 1. Introducción

¡Tu introducción va aquí! A continuación, se enumeran algunos ejemplos de comandos y funciones de uso común para ayudarte a comenzar.

cuaderno de trabajo[1]

- a)
- b)
- c)

1.1. Implemente los siguientes algoritmos en tres lenguajes de programación

## 2. Marco Teórico

### 3. Metodología

- a) Bubble sort
- b) Heap sort
- c) Insertion sort
- d) Selection sort
- e) Shell sort
- f) Merge sort
- g) Quick sort

```

+-----+-----+-----+-----+-----+-----+-----+-----+
| NVIDIA-SMI 470.74          | Driver Version: 460.32.03 | CUDA Version: 11.2 |
+-----+-----+-----+-----+-----+-----+-----+-----+
| GPU  Name      Persistence-M| Bus-Id        Disp.A     | Volatile Uncorr. ECC |
| Fan  Temp  Perf    Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|====+=====+====+=====+=====+=====+=====+=====+
|  0  Tesla  K80      Off      | 00000000:00:04:0  Off  |           0          |
| N/A   40C    P0      58W / 149W| 121MiB / 11441MiB |      0%      Default |
+-----+-----+-----+-----+-----+-----+-----+-----+
|
+-----+-----+-----+-----+-----+-----+-----+-----+
| Processes: |
| GPU   GI    CI          PID    Type    Process name                      | GPU Memory |
|   ID   ID                                 |            | Usage      |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

Figura 1:

```

Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:            Little Endian
CPU(s):                2
On-line CPU(s) list:   0,1
Thread(s) per core:    2
Core(s) per socket:    1
Socket(s):             1
NUMA node(s):          1
Vendor ID:             GenuineIntel
CPU family:            6
Model:                 63
Model name:            Intel(R) Xeon(R) CPU @ 2.30GHz
Stepping:              0
CPU MHz:               2299.998
BogoMIPS:              4599.99
Hypervisor vendor:     KVM
Virtualization type:   full
L1d cache:             32K
L1i cache:             32K
L2 cache:              256K
L3 cache:              46080K
NUMA node0 CPU(s):    0,1
Flags:                 fpu vme de pse tsc msr pae mce cx8

```

Figura 2:

The screenshot shows a Jupyter Notebook titled 'Funcional20.ipynb'. The main code cell contains the following Python code:

```

[3] 29 A=[27,17,2,4,52,3,4,3]
    30 B=mergesort(A)
    31 print(B)
    32

```

The output of the code is the sorted list: `[2, 3, 3, 4, 4, 17, 27, 52]`.

Below the output, there is a plot showing the performance of two sorting algorithms: mergesort and insertion. The x-axis is labeled 'n' and the y-axis is labeled 'tiempo'. The legend indicates that the green line represents 'mergesort' and the blue line represents 'insertion'.

On the right side of the notebook, there is a file named 'mod.py' which contains the following code:

```

17 tValues = []
18 for n in range
19     # run myFn
20     runtime =
21     for t in r
22         global
23         lst =
24         start
25         myFn(
26         end =
27         runtim
28         runtime =
29         nValues.ap
30         tValues.a
31
32 with open("inp
33
34     for item i
35         f.writ
36     return nValues
37
38 def printe():
39     print(lst)
40
41 lst=[]
42

```

The bottom status bar indicates that the notebook is running for 2 hours, 15 minutes, and 59 seconds. The current cell is 'Cé' and the execution path is 'mimodulo() > mergesort() > mergesort() > mergesort() > mergesort() > merge()'. There is a pink highlight on the status bar.

Figura 3:

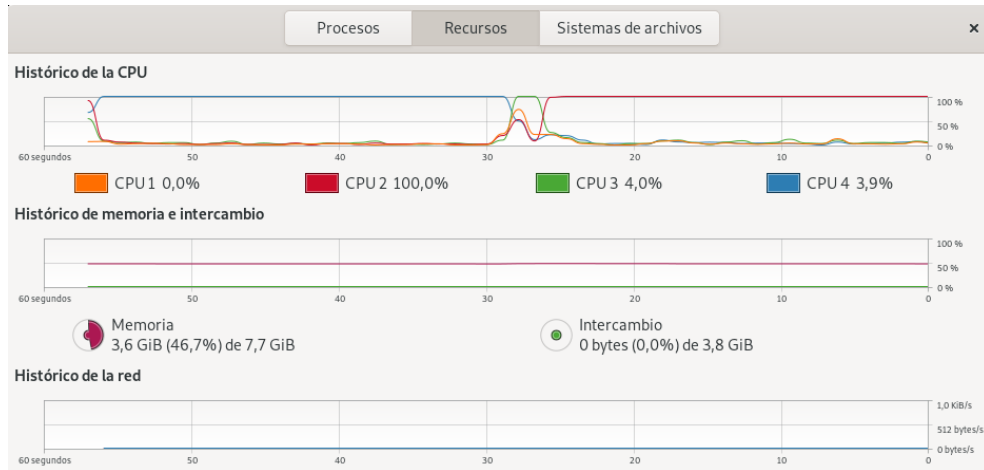


Figura 4:

## 4. Resultados

## 5. Conclusiones

## 6. Discusión

Documentar sus hallazgos en un formato de artículo de investigación (formato de libre elección), considerando aspectos del marco teórico (estado del arte), metodología, resultados, conclusiones, discusión y bibliografía [2].

## Referencias

- [1] Bejar Merma Angel Andres. Cuaderno de trabajo. <https://colab.research.google.com/drive/1FZjcnchMuRvmnAxCTwf54DTL1DiJmsNm?usp=sharing>, 2021. [Online; accessed read].
- [2] Jose Fager. *Estructura de datos*. Proyecto Latin. latin, 2014.