

Practica de Python problema de Generales Bizantinos

GÓMEZ EDGAR, Universidad Central de Venezuela, Venezuela

Este trabajo presenta una implementación práctica en Python del *Problema de los Generales Bizantinos*, se investiga experimentalmente la condición teórica $N \geq 3m + 1$, necesaria para alcanzar un consenso a pesar de la presencia de m traidores. Mediante la ejecución de un script de simulación con diferentes configuraciones de generales leales (l) y traidores (t), se demuestra cómo el incumplimiento de esta condición conduce a resultados inconsistentes, mientras que su cumplimiento permite alcanzar un acuerdo mayoritario.

CCS Concepts: • Organización de sistemas informáticos → Sistemas Distribuidos; Tolerancia a Fallas.

Additional Key Words and Phrases: Sistemas Distribuidos, Tolerancia a Fallos, Generales Bizantinos, Consenso Distribuido

ACM Reference Format:

GÓMEZ EDGAR. 2026. Practica de Python problema de Generales Bizantinos. *Proc. ACM Hum.-Comput. Interact.* 1, 1, Article 1 (January 2026), 2 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

1. Introducción

El Problema de los Generales Bizantinos modela el desafío de lograr consenso en un sistema distribuido donde algunos componentes pueden fallar o actuar de manera maliciosa, siendo fundamental para la tolerancia a fallos en redes y sistemas críticos. Esta práctica implementa en Python una simulación del algoritmo clásico para validar experimentalmente la condición teórica $N \geq 3m + 1$, donde N es el número total de generales y m el número de traidores tolerables. A través de diferentes configuraciones de nodos leales y traidores, se analiza la capacidad del sistema para alcanzar un acuerdo unánime a pesar de la información contradictoria, demostrando la relación directa entre la configuración del sistema y la posibilidad de consenso.

2. Resultados

2.1. Primer Comando

Ejecutamos en consola el comando y se obtiene el resultado lo que se muestra en Fig 1.

```
PS C:\Users\Edgar Gomez\Desktop\byzantine_generals-master> python3 byzantine_generals.py -m 1 -G 1,t,1 -O ATTACK
General 0: [('RETREAT', 1), ('ATTACK', 1)]
General 1: [('ATTACK', 2)]
General 2: [('RETREAT', 1), ('ATTACK', 1)]
```

Fig. 1

Analizando la salida, se observa que no se alcanza un consenso entre los generales. Los generales 0 y 2 muestran un empate en su decisión (un voto para 'ATTACK' y uno para 'RETREAT'), lo que indica una falta de acuerdo unánime. Este resultado corrobora que no se cumple la relación $N \geq 3m + 1$. En este caso, con tres generales ($N = 3$) y un traidor ($m = 1$), se tiene $3 \geq 4$, lo cual es falso.

Author's Contact Information: GÓMEZ EDGAR, Universidad Central de Venezuela, Caracas, Venezuela.

2026. ACM 2573-0142/2026/1-ART1
<https://doi.org/10.1145/nnnnnnn.nnnnnnn>

2.2. Segundo Comando

Ejecutamos en consola el comando y se obtiene el resultado lo que se muestra en Fig 2.

```
PS C:\Users\Edgar Gomez\Desktop\byzantine_generals-master> python3 byzantine_generals.py -m 1 -G t,1,1 -O ATTACK
General 0: [('ATTACK', 1), ('RETREAT', 1)]
General 1: [('ATTACK', 1), ('RETREAT', 1)]
General 2: [('ATTACK', 1), ('RETREAT', 1)]
```

Fig. 2

En este escenario, nuevamente no se alcanza un consenso. Todos los generales (0, 1 y 2) presentan vectores de decisión con un voto para cada opción, evidenciando la indecisión provocada por el traidor. Esto vuelve a demostrar el incumplimiento de la condición $N \geq 3m + 1$ ($3 \geq 4$ es falso), imposibilitando un acuerdo claro.

2.3. Tercer Comando

Ejecutamos en consola el comando y se obtiene el resultado lo que se muestra en Fig 3.

```
PS C:\Users\Edgar Gomez\Desktop\byzantine_generals-master> python3 byzantine_generals.py -m 1 -G 1,t,1,1,1 -O ATTACK
General 0: [('ATTACK', 3), ('RETREAT', 1)]
General 1: [('ATTACK', 4)]
General 2: [('ATTACK', 3), ('RETREAT', 1)]
General 3: [('ATTACK', 4)]
General 4: [('ATTACK', 3), ('RETREAT', 1)]
```

Fig. 3

En esta configuración, se observa un claro consenso hacia la acción 'ATTACK'. Los generales 0, 2 y 4 muestran una mayoría de tres votos para atacar frente a uno para retirarse, mientras que los generales 1 y 3 registran cuatro votos para atacar. Este acuerdo mayoritario confirma que se cumple la relación $N \geq 3m + 1$. Con cinco generales ($N = 5$) y un traidor ($m = 1$), se tiene $5 \geq 4$, lo cual es cierto, permitiendo al sistema superar la influencia del traidor y alcanzar una decisión coherente.

3. Conclusión

La implementación práctica en Python del algoritmo de los Generales Bizantinos ha permitido validar experimentalmente la condición teórica $N \geq 3m + 1$ para alcanzar consenso en presencia de fallos arbitrarios. Los resultados demuestran que cuando esta condición no se cumple (tres generales con un traidor), el sistema genera decisiones inconsistentes e incompatibles, mientras que al satisfacerla (cinco generales con un traidor) se logra un acuerdo mayoritario claro y coherente. Esta práctica confirma la importancia crítica de la relación entre el número total de nodos y la cantidad de fallos tolerables en el diseño de sistemas distribuidos resilientes, ilustrando cómo la teoría se materializa en resultados computacionales concretos.