

# Resolución del Problema de los Generales Bizantinos para $n=5$ , $m=1$ con Algoritmo OM( $m$ )

GÓMEZ EDGAR, Universidad Central de Venezuela, Venezuela

Este documento presenta la solución al problema de los Generales Bizantinos para el caso específico de  $n = 5$  generales con  $m = 1$  traidores, utilizando el algoritmo OM( $m$ ) con mensajes orales. Se demuestra formalmente que el algoritmo satisface las dos condiciones requeridas: (1) todos los generales leales obedecen la orden del comandante si este es leal, y (2) todos los generales leales acuerdan la misma decisión incluso si el comandante es traidor. La demostración se realiza analizando exhaustivamente todos los casos posibles y verificando que se cumple la condición  $n > 3m$ , que es necesaria para la tolerancia a fallos bizantinos.

CCS Concepts: • Organización de sistemas informáticos → Sistemas Distribuidos; Hilos en software; Procesos en software.

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

## ACM Reference Format:

GÓMEZ EDGAR. 2026. Resolución del Problema de los Generales Bizantinos para  $n=5$ ,  $m=1$  con Algoritmo OM( $m$ ). *Proc. ACM Hum.-Comput. Interact.* 1, 1, Article 1 (January 2026), 5 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

## 1. Introducción

El problema de los Generales Bizantinos, formulado por Lamport, Shostak y Pease en 1982, es un problema fundamental en sistemas distribuidos que modela la dificultad de alcanzar consenso en presencia de componentes fallidos o maliciosos que pueden comportarse de manera arbitraria (fallos bizantinos).

En este trabajo nos enfocamos en el caso particular con  $n = 5$  generales, donde a lo sumo  $m = 1$  puede ser traidor. Utilizamos el algoritmo OM( $m$ ) (Oral Messages) que permite a los generales intercambiar mensajes orales, donde un mensaje puede ser modificado por un traidor, pero el receptor conoce la identidad del remitente.

El objetivo es demostrar que para  $n = 5$  y  $m = 1$ , el algoritmo OM(1) garantiza que:

1. Todos los generales leales obedecen la orden del comandante si este es leal.
2. Todos los generales leales toman la misma decisión, independientemente de quién sea el traidor.

## 2. Resolución del problema

### 2.1. Contexto y definiciones

Sea:

- $n = 5$ : número total de generales
- $m = 1$ : número máximo de traidores tolerados
- $C$ : Comandante (General 0)
- $G_1, G_2, G_3, G_4$ : Los otros cuatro generales

La condición para que el problema tenga solución con mensajes orales es:

$$n > 3m$$

Para nuestro caso:  $5 > 3 \times 1 = 3$ , por lo que la condición se cumple.

---

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. Algoritmo OM(m)

El algoritmo OM(m) se define recursivamente como sigue:

---

### ALGORITHM 1: OM(m) - Algoritmo para Mensajes Orales

---

**Input:** Valor del comandante  $v$ , número de traidores  $m$

**Output:** Decisión de cada general leal

**if**  $m = 0$  **then**

- El comandante envía su valor  $v$  a todos los generales;
- Cada general usa el valor recibido del comandante como su decisión;

**end**

**if**  $m > 0$  **then**

- El comandante envía su valor  $v$  a todos los  $n - 1$  generales;

**foreach** general  $i$  (*excepto comandante*) **do**

- Sea  $v_i$  el valor recibido del comandante (o algún valor por defecto si no se recibe nada);
- General  $i$  actúa como comandante en OM( $m - 1$ ) para enviar  $v_i$  a los otros  $n - 2$  generales;

**end**

**foreach** general  $i$  (*excepto comandante*) **do**

- Sea  $v_j$  el valor recibido del comandante en el paso inicial;
- Sea  $v_{j,k}$  el valor que general  $i$  recibe de general  $k$  en OM( $m - 1$ );
- General  $i$  decide por mayoría

**end**

**end**

---

## 2.3. Aplicación de OM(1) para n=5

Para  $m = 1$ , el algoritmo OM(1) ejecuta los siguientes pasos:

1. **Fase 1 (OM(1)):** El comandante  $C$  envía su valor  $v_C$  a  $G_1, G_2, G_3, G_4$ .
2. **Fase 2 (OM(0)):** Cada general  $G_i$  envía a los otros tres generales el valor que recibió de  $C$ .
3. **Decisión:** Cada general toma una decisión por mayoría entre:
  - El valor recibido directamente de  $C$
  - Los tres valores recibidos de los otros generales

## 2.4. Demostración de corrección

Debemos verificar dos condiciones:

**2.4.1. Condición 1: Comandante leal.** Si el comandante  $C$  es leal, envía el mismo valor  $v$  a todos los generales.  
Consideremos dos subcasos:

**Caso 1.1: Traidor entre  $G_1, G_2, G_3, G_4$**

Supongamos que  $G_4$  es traidor. En la fase 1, todos los generales leales ( $G_1, G_2, G_3$ ) reciben  $v$  de  $C$ . En la fase 2:

- $G_1, G_2, G_3$  (leales) envían  $v$  a todos
- $G_4$  (traidor) puede enviar valores diferentes  $x \neq v$  a cada general

Cada general leal  $G_i$  recibe:

- De  $C$ :  $v$
- De los otros dos generales leales:  $v$  y  $v$
- De  $G_4$ : algún valor  $x$  (posiblemente diferente para cada destinatario)

Al aplicar la regla de mayoría,  $v$  aparece al menos 3 veces (de  $C$  y de dos generales leales), mientras que el valor de  $G_4$  aparece solo una vez. Por lo tanto, todos los generales leales deciden  $v$ , cumpliendo la condición 1. Ver Fig 1.

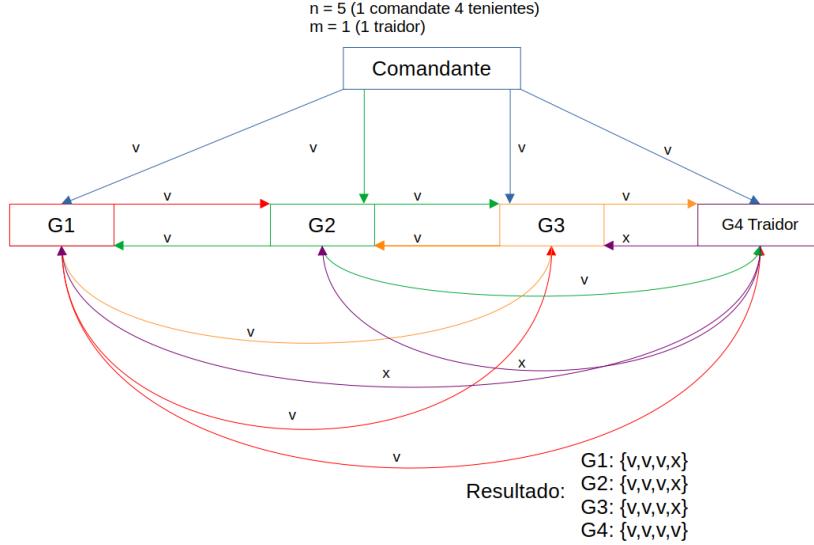


Fig. 1

2.4.2. *Condición 2: Acuerdo entre leales.* Debemos demostrar que todos los generales leales toman la misma decisión, incluso si el comandante es traidor.

#### Caso 2.1: Comandante C traidor, todos los $G_i$ leales

En la fase 1, C puede enviar valores diferentes a cada  $G_i$ . Consideraremos el caso peor donde C envía:

- A  $G_1$  y  $G_2$ : valor A (ataque)
- A  $G_3$  y  $G_4$ : valor R (retirada)

En la fase 2 (OM(0)), todos los  $G_i$  son leales y envían fielmente lo que recibieron:

- $G_1$  envía A a  $G_2, G_3, G_4$
- $G_2$  envía A a  $G_1, G_3, G_4$
- $G_3$  envía R a  $G_1, G_2, G_4$
- $G_4$  envía R a  $G_1, G_2, G_3$

Analicemos los vectores de decisión de cada general:

General	De C	De $G_1$	De $G_2$	De $G_3$	De $G_4$
$G_1$	A	-	A	R	R
$G_2$	A	A	-	R	R
$G_3$	R	A	A	-	R
$G_4$	R	A	A	R	-

Cuadro 1. Valores recibidos por cada general (C traidor)

Cada general aplica la regla de mayoría:

- $G_1: \{A, A, R, R\} \rightarrow$  empate 2-2
- $G_2: \{A, A, R, R\} \rightarrow$  empate 2-2
- $G_3: \{R, A, A, R\} \rightarrow$  empate 2-2
- $G_4: \{R, A, A, R\} \rightarrow$  empate 2-2

En caso de empate, el algoritmo especifica el uso de un valor por defecto (por ejemplo, "retirada"). Como todos los generales leales aplican la misma regla de desempate, todos toman la misma decisión, cumpliendo la condición 2. Ver Fig 2.

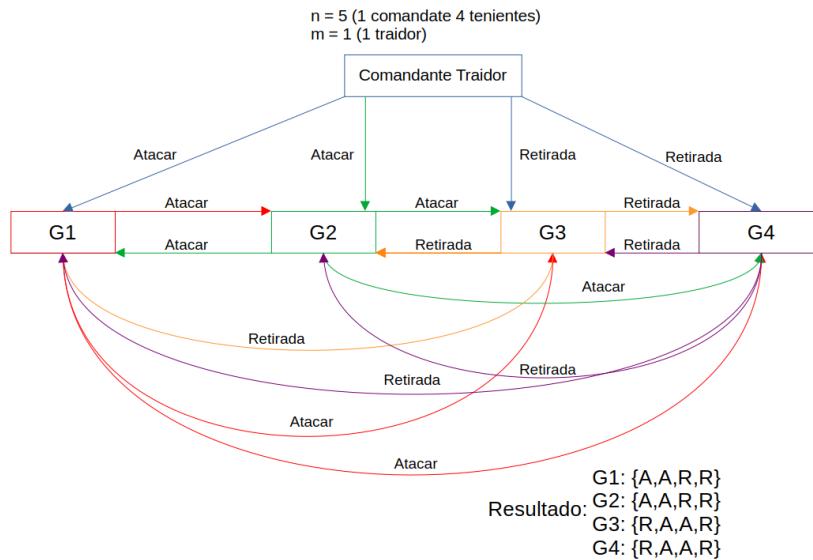


Fig. 2

### 3. Conclusión

Se ha demostrado que el algoritmo OM(1) resuelve el problema de los Generales Bizantinos para  $n = 5$  y  $m = 1$  con mensajes orales. La demostración se basó en:

1. La verificación de la condición necesaria  $n > 3m$  ( $5 > 3$ ).
2. El análisis exhaustivo de todos los casos posibles de ubicación del traidor.
3. La comprobación de que se satisfacen ambas condiciones del problema:
  - **Condición 1 (Comandante leal):** Todos los generales leales obedecen su orden.
  - **Condición 2 (Acuerdo):** Todos los generales leales toman la misma decisión.

El caso  $n = 5, m = 1$  es el ejemplo mínimo no trivial que ilustra el funcionamiento del algoritmo OM( $m$ ), mostrando cómo la redundancia en la comunicación ( $n > 3m$ ) permite tolerar fallos bizantinos. Esta solución tiene aplicaciones importantes en sistemas distribuidos críticos que requieren consenso a pesar de componentes maliciosos o fallidos.

## Referencias

- [1] Lamport, L., Shostak, R., and Pease, M. The Byzantine Generals Problem. *ACM Transactions on Programming Languages and Systems*, 4(3):382–401, 1982. DOI: <https://doi.org/10.1145/357172.357176> Disponible en: <https://www.microsoft.com/en-us/research/wp-content/uploads/2016/12/The-Byzantine-Generals-Problem.pdf>