

Transparencia en Sistemas Distribuidos

GOMEZ EDGAR, Universidad Central de Venezuela, Venezuela

Los sistemas distribuidos son un conjunto de componentes independientes que el usuario percibe como un solo sistema coherente. Estos componentes colaboran entre sí. El modo en el que colaboran está, en la mayoría de los casos, oculto al usuario, al igual que su organización interna. Los sistemas distribuidos son fáciles de escalar sin que el usuario note la diferencia al agregar, reemplazar o eliminar componentes. Igualmente, al ocurrir una falla, el usuario final no percibe qué o cuál componente está fallando.

CCS Concepts: • **Organización de sistemas informáticos** → **Sistemas Distribuidos**; *Transparencia en Sistemas Distribuidos*.

Additional Key Words and Phrases: Sistemas Distribuidos, Transparencia en Sistemas Distribuidos, middleware capa de software

ACM Reference Format:

GOMEZ EDGAR. 2025. Transparencia en Sistemas Distribuidos. *Proc. ACM Hum.-Comput. Interact.* 1, 1, Article 1 (December 2025), 4 pages. <https://doi.org/10.1145/nnnnnnnn.nnnnnnnn>

1. Introduction

Los sistemas distribuidos consisten en una red de computadoras autónomas que trabajan conjuntamente para lograr un objetivo en común. Este conjunto de computadoras posee ciertas características, como: cada componente de la red es autónomo; el usuario tiene la percepción de que está lidiando con un solo sistema; los componentes autónomos colaboran entre sí y esta comunicación es transparente al usuario; las diferencias y la organización entre componentes del sistema están ocultas al usuario. Los usuarios pueden interactuar consistente y uniformemente con un sistema distribuido, independientemente de dónde y cuándo esta interacción tenga lugar. Estos sistemas distribuidos son fácilmente escalables debido a que cada unidad del sistema es independiente. Por lo regular, el sistema siempre está disponible aunque ciertas partes de él no lo estén. Los usuarios no notan que ciertas partes del sistema son reemplazadas, reparadas o añadidas para servir a más usuarios o aplicaciones.

Para dar la percepción de un sistema homogéneo (o heterogéneo, si esa es la intención), el sistema se organiza en capas de software. Una capa llamada “middleware” es colocada entre el nivel superior de usuarios y aplicaciones y el nivel inferior consistente en sistemas operativos y comunicaciones, como se muestra en la Fig 1.

Author’s Contact Information: GOMEZ EDGAR, Universidad Central de Venezuela, Caracas, Venezuela.

2025. ACM 2573-0142/2025/12-ART1
<https://doi.org/10.1145/nnnnnnnn.nnnnnnnn>

Proc. ACM Hum.-Comput. Interact., Vol. 1, No. 1, Article 1. Publication date: December 2025.

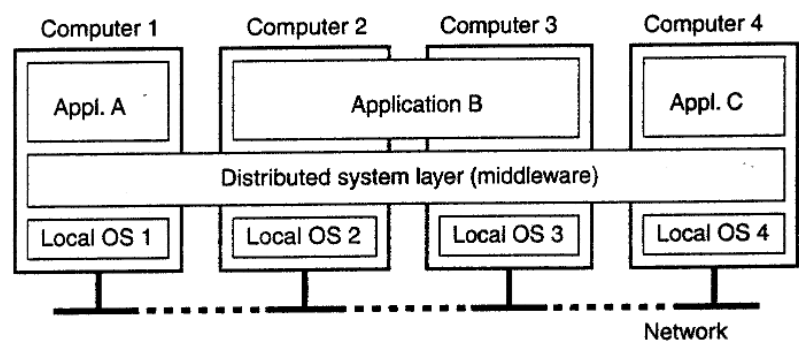


Fig. 1. Estructura del middleware en sistemas distribuidos (Adaptado de [1])

Fig 1 muestra cuatro computadoras en red, tres aplicaciones A, B y C, donde B está distribuida entre la computadora 2 y 3. Lo que está haciendo la capa de middleware es ocultar cómo estas aplicaciones se comunican entre sí y qué aspectos del hardware comparten.

2. Transparencia en Sistemas Distribuidos

Un sistema distribuido debe proporcionar una manera fácil de que los recursos sean accesibles. Debe esconder el hecho de que los recursos están distribuidos en la red. Esta es una de las metas a las que debe apuntar un sistema distribuido: esconder que los procesos y recursos están distribuidos en múltiples computadoras. El sistema distribuido debe poder presentarse ante usuarios y aplicaciones como un solo sistema; cuando esto ocurre, se dice que el sistema es transparente.

2.1. Tipos de transparencia

El concepto de transparencia aplica a varios aspectos de los sistemas distribuidos. Los más importantes se muestran en el Cuadro 1.

Cuadro 1. Aspectos de transparencia

Transparencia	Descripción
Acceso	Ocultar diferencias entre cómo la información es representada y cómo se accede al recurso.
Localización	Ocultar dónde está localizado el recurso.
Migración	Ocultar que el recurso pudo ser movido a otro lugar.
Relocalización	Ocultar que el recurso pudo ser movido a otro lugar mientras está en uso.
Replicación	Ocultar que el recurso es replicado.
Concurrencia	Ocultar que el recurso es compartido por diferentes usuarios al mismo tiempo.
Falla	Ocultar la falla y la recuperación del recurso.

La transparencia de acceso se encarga de ocultar la representación del recurso y la manera en cómo es accedido por el usuario. Esto se enfoca en establecer un acuerdo sobre cómo la información es mostrada por diferentes computadoras y sistemas operativos. Por ejemplo, varios sistemas operativos difieren en la convención de cómo los archivos son nombrados, y también en las maneras diferentes de cómo los archivos son manipulados. Todos estos detalles deberían ser ocultos para usuarios y aplicaciones.

La transparencia de localización se refiere a que un usuario o aplicación no debe saber dónde está la información de un recurso físicamente alojada. Por ejemplo, se pueden usar nombres donde la localización del recurso no esté secretamente codificada. Una URL cumple esta función: no nos da pistas sobre la localización física de un recurso.

En los sistemas distribuidos en los cuales los recursos pueden ser movidos sin afectar cómo se accede a esos recursos, se dice que el sistema soporta transparencia de migración. Incluso en los casos en que el recurso es movido en el momento que está en uso y el usuario o aplicación no se entera, se dice que el sistema es transparente en la relocalización. Un ejemplo es cuando los usuarios móviles usan sus laptops conectadas inalámbricamente mientras se mueven de lugar sin ser desconectados.

En el caso de la transparencia de replicación, los recursos de un sistema distribuido son replicados para incrementar la disponibilidad y el rendimiento, colocando una copia cerca del lugar donde se accede. La transparencia de replicación oculta el hecho de que diferentes copias de un recurso existen. Para ocultar las réplicas, es necesario que el recurso tenga el mismo nombre. Por consiguiente, un recurso que soporte transparencia de replicación generalmente debe soportar transparencia de localización, porque sería imposible referirse a réplicas en localizaciones diferentes. Un ejemplo puede ser los servicios de alojamiento de archivos en la nube como Google Drive, que manejan réplicas de los archivos de los usuarios y colocan los recursos en servidores cercanos donde el usuario pueda tener acceso.

En muchos casos, el compartir recursos con usuarios se hace de manera concurrente. Esto quiere decir que múltiples usuarios acceden a un recurso al mismo tiempo. De esta manera, entra en juego la transparencia de concurrencia. Por ejemplo, dos usuarios independientes quieren guardar sus archivos en el mismo servidor de archivos o acceder a las mismas tablas en una base de datos compartida. En tales casos, es importante que cada usuario no note que otro usuario está haciendo uso del mismo recurso. A este fenómeno, como hemos dicho, se le denomina transparencia de concurrencia. Un aspecto importante es que el acceso concurrente a un archivo lo mantiene en un estado consistente. Esto puede lograrse a través de mecanismos de bloqueo que puedan dar un acceso exclusivo al recurso por turnos o mediante una transacción.

Por último, tenemos la transparencia de falla. Esta es una característica donde el usuario final no nota que un recurso ha fallado y no funciona apropiadamente, pero el sistema en general sigue funcionando con normalidad mientras se trata de reponer el recurso fallido. Esta es una de las características de transparencia más difíciles de implementar, ya que es difícil distinguir entre un recurso muerto y otro muy lento. Por ejemplo, si contactamos a un servicio web muy ocupado y nos indica que el servicio no está disponible, a ese punto el usuario no puede concluir que el servidor está realmente caído.

3. Conclusión

En este documento, hemos explorado los sistemas distribuidos y su meta principal: la transparencia. La transparencia es fundamental para ocultar la complejidad y la naturaleza distribuida de los recursos y procesos, presentando al usuario y a las aplicaciones una imagen de un sistema único y coherente. Discutimos los tipos clave de transparencia acceso, localización, migración, replicación, concurrencia y falla y cómo la capa de “middleware” juega un papel crucial en la implementación de estos mecanismos de ocultación. Un diseño que prioriza la transparencia permite la escalabilidad y la alta disponibilidad, elementos esenciales para la infraestructura moderna de software.

Referencias

- [1] Tanenbaum, A. S. and Van Steen, M. *Distributed Systems: Principles and Paradigms (2nd Edition)*. Pearson Prentice Hall, Upper Saddle River, NJ 07458, 2007.