



---

## INGENIERIA EN SISTEMAS COMPUTACIONALES

---

- ✚ ESCUELA: UNIVERSIDAD DEL SURESTE.
- ✚ CARRERA: INGENIERIA EN SISTEMAS COMPUTACIONALES.
- ✚ DOCENTE: EDUARDO GENNER ESCALANTE CRUZ.
- ✚ MATERIA: SISTEMAS OPERATIVOS DISTRIBUIDOS.
- ✚ MODULO: 1° - MODULO.
- ✚ TEMA: MAPA CONCEPTUAL DE TRANSACCION.
- ✚ ALUMNA: LAURA DENIS TON HERNANDEZ
- ✚ SEMESTRE: 6 CUATRIMESTRE
- ✚ FECHA: 15/ JUNIO/ 2020

# SISTEMAS DISTRUIDOS

Los clientes (o programas que representan entidades que necesitan servicios) y los servidores (o programas que proporcionan servicios) son objetos separados desde un punto de vista lógico y que se comunican atravésdeunaredde comunicaciones pararealizaruna o variastareasdeformaconjunta. Un cliente hace una petición de un servicio y recibe la respuesta a dicha petición; un servidor recibe y procesa la petición, y devuelve la respuestasolicitada.

## Modelos

## INOVACION DE METODOS

Cliente/servidor: • Cliente: incluye sistema operativo (SO) sobre con interfaz gráfico de usuario (GUI) o interfaz orientado a objetos de usuario (OOUI) • Servidor: ejecuta software especializado • Middleware: software distribuido para interacciones entre cliente y servidor.

ATM: Este algoritmo está basado en el uso del tiempo coordinado universal (siglas en inglés, UTC), el cual es recibido por un equipo dentro del sistema distribuido.

CLUSTER: Clúster, En un sistema informático, un clúster es un grupo de servidores y otros recursos que actúan como un único sistema y permiten una alta disponibilidad y, en algunos casos, balanceo de carga y procesamiento paralelo.

TRANSACCION: La meta de las transacciones es asegurar que todos los objetos gestionados por un servidor permanecen en un estado consistente cuando dichos objetos son accedidos por múltiples clientes y/o en presencia de caídas del servidor.

RPC: RPC admite paradigmas de programación de procedimientos, por lo tanto, está basado en C, mientras que RMI admite paradigmas de programación orientada a objetos y está basado en Java

RMI: RMI es un paquete de JAVA que permite manejar objetos (y sus respectivos métodos) de manera remota, para utilizar los recursos de un servidor de manera transparente para el usuario local.

## SINCRONIZACION

PROCESOS: Dos procesos no deben encontrarse al mismo tiempo dentro de una SC con el mismo