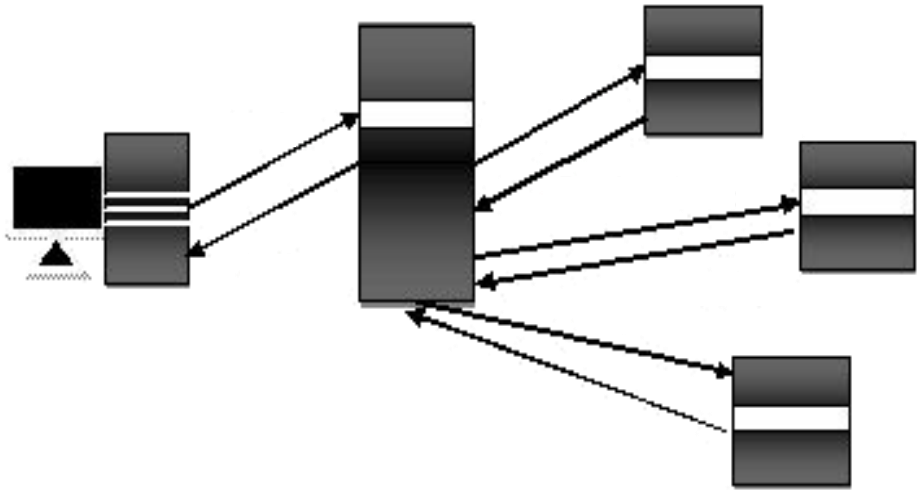
Programación Distribuida

Dr. Freddy Alberto Paz Espinoza

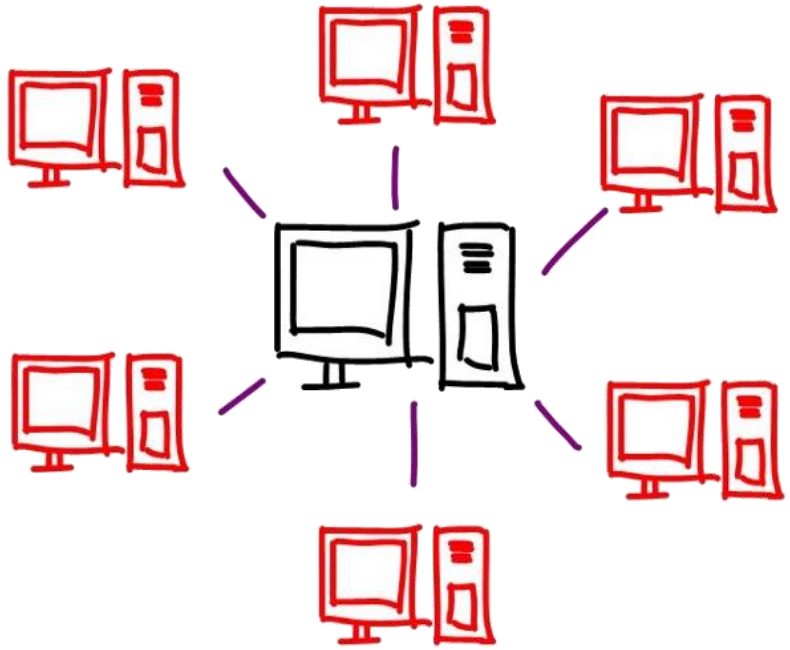
Programación Distribuida

- La evolución de los sistemas de software fue exigiendo, conforme estos crecían en complejidad en respuesta a nuevas necesidades, que las actividades que involucraban a todo el sistema fuese distribuyéndose entre elementos de software y hardware separados cada cual especializado en una actividad.



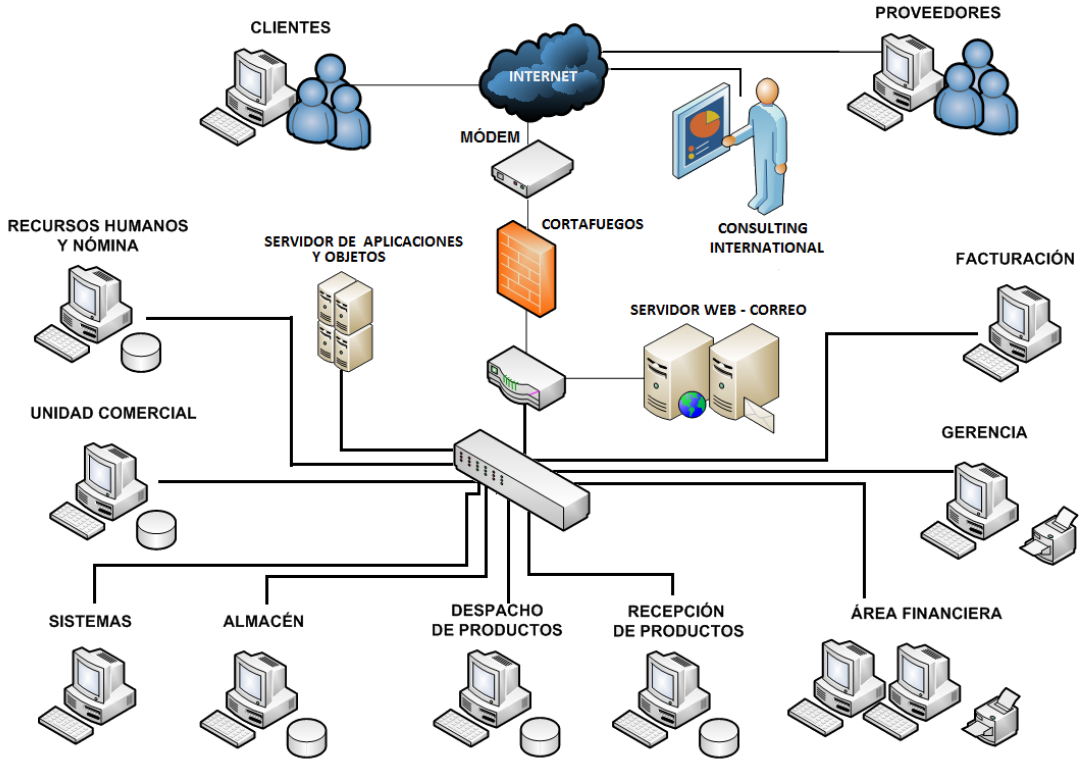
Programación Distribuida

- Existen razones por las cuales un sistema de software requiere distribuir sus actividades:
  - Los datos utilizados por el sistema requieren estar distribuidos.
  - La computación con dichos datos requiere estar distribuida.
  - Los usuarios de la aplicación requieren estar distribuidos.



Programación Distribuida

- Por *distribuido* se entiende el hecho de que de las actividades de un sistema se realicen en dos o más nodos, cada nodo se comunica con el otro mediante una red (del tipo que sea necesario) y cada nodo esta representado comúnmente por un computador.



Programación Distribuida

- Uno de las primeras tecnologías de gran aceptación (implementada en muchos lenguajes de programación) para la PSD fueron los sockets (aparece por primera vez en Berkeley Unix en 1981). Los sockets son soportados por la gran mayoría de sistemas operativos:

  - Los S.O. BSD Unix brindan sockets como parte del kernel.
  - Los S.O. Windows brindan un API para sockets llamado WinSock.
  - MS DOS , Mac OS, OS/2, etc., brindan librerías de sockets.
  - Lenguajes POO brindan clases que facilitan el trabajo con sockets.

Programación Distribuida

- Los sockets homogenizan el trabajo con diferentes protocolos de comunicación, a la par de ser muy eficientes.
- Los sockets sólo permiten trasmitir y recibir datos no estructurados (paquetes de bytes), los que deben de crearse y enviarse (mediante un formato propietario) de un nodo a otro de una red.
- Adicionalmente, el programador debe lidiar con los problemas de pérdida de conexión, pérdida de paquetes de datos, problemas de inversión de los bits y sincronización del trabajo entre el que envía y el que recibe.

Programación Distribuida

- Un nivel algo mayor de abstracción se brindo con RPC (Remote Procedure Call, presentada por primera vez por Birrelly Nelson en 1985). RPC permite a un programa llamar/invocar/ejecutar un procedimiento/función/rutina en un sistema remoto.
- La ventaja de realizar un RPC con respecto a utilizar una tecnología de envío y recepción de datos, es la de simular la llamada a un procedimiento local, simplificando enormemente la lógica de programación.

Programación Distribuida

- Los Sistemas de Objetos Distribuidos distribuyen datos y funcionalidad. Los objetos se prestan como una forma muy clara y natural de modelar dicha distribución, y de compartirla entre los diferentes nodos de una red.
  - COM (Component Object Model)
  - .NET Remoting
  - WCF (Windows Communication Foundation)
  - RMI (Java Remote Method Invocation)
  - CORBA (Common Object Request Broker Architecture)

Programación Distribuida

- En la PSOD, un objeto distribuido puede o bien:
  - Ser accedido de manera distribuida.
  - Ser transmitido.
- El objeto distribuido es creado en un nodo de la red y sus métodos son llamados remotamente desde otros nodos. Los datos del objeto existen y sus métodos se ejecutan en el nodo donde el objeto es creado.
- En el segundo caso, una copia del objeto es trasmitida, por lo que los datos y métodos de esta copia existen en otro nodo de la red.

Programación Distribuida

- Para los objetos distribuidos mediante una referencia remota, el control de su ciclo de vida difiere de los objetos co-locales. Un objeto distribuido puede:
  - Preexistir (en un programa en ejecución) en un nodo y ser accedido desde otro objeto en otro nodo.
  - Ser creado en un nodo a solicitud de un objeto en otro nodo.
  - Permanecer vivo hasta que el último objeto que lo refería descarte dicha referencia. Esto puede ser realizado por el programador de manera explícita o por algún mecanismo automático.
- Permanecer vivo hasta finalizar el programa donde fue creado.

Referencias:

- Coulouris, G., Dollimore, J. and Kindberg, T. (2001). Distributed Systems: Concepts and Design. 3era Edición. Estados Unidos: Addison Wesley.