

# Sistemas distribuidos

## Grado en Ingeniería Informática

### Tema 01-03: Desafíos los Sistemas Distribuidos

Departamento de Ingeniería Informática  
Universidad de Cádiz



Escuela Superior de Ingeniería  
Dpto. de Ingeniería Informática



Curso 2019 – 2020

# Indice

- 1 Desafíos
- 2 Heterogeneidad
- 3 Extensibilidad
- 4 Seguridad
- 5 Escalabilidad
- 6 T.Fallos
- 7 Concurrencia
- 8 Transparencia
- 9 Bibliografía

# Sección 1 | Desafíos

# Objetivos

## de los sistemas distribuidos

### Objetivos de los sistemas distribuidos

- **Compartir** recursos (de cualquier tipo)
- Conseguir **transparencia** (de distribución frente al usuario)
- **Extensibilidad** (facilidad de adoptar cambios de SW/HW)
- **Escalabilidad** (de tamaño, geográfica, administrativa)

# Desafíos

## de los sistemas distribuidos

### Propiedades de los SD

- Heterogeneidad
- Extensibilidad
- Seguridad
- Escalabilidad
- Tratamiento de Fallos
- Concurrencia
- Transparencia

## Sección 2 | Heterogeneidad

# Heterogeneidad

## Heterogeneity

Internet permite que los usuarios accedan a servicios y ejecuten aplicaciones sobre un conjunto “heterogéneo” de redes y computadores. Esta heterogeneidad se aplica a:

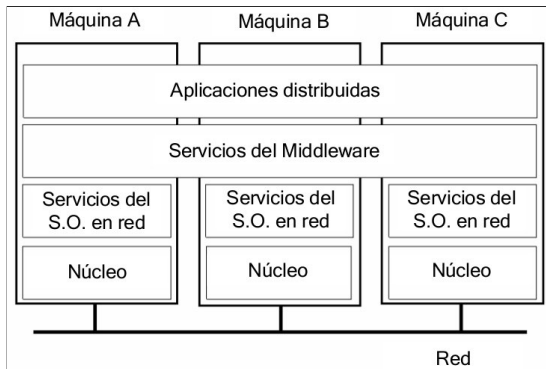
- Redes.(protocolos, estándares,...)
- Hardware de computadores. (procesamientos, recursos,...)
- Sistemas operativos. (llamadas,...)
- Lenguajes de programación. (tipos de datos,...)
- Implementaciones de diferentes desarrolladores. (versiones,...)

Para que esto no ocurra es necesario concertar y adoptar estándares

# Heterogeneidad

## Middleware

El término **middleware** se aplica al estrato de software que provee una abstracción de programación, así como un enmascaramiento de la heterogeneidad subyacente de las redes, hardware, sistemas operativos y lenguajes de programación.

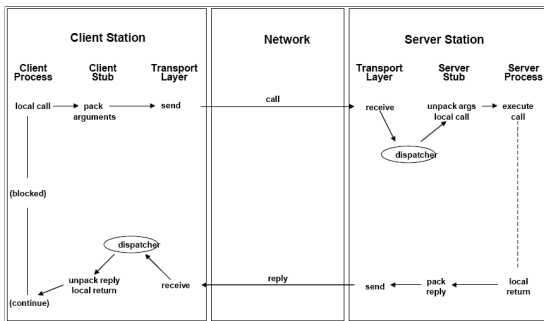




# Heterogeneidad

## Servicios del Middleware

- Servicios de comunicación
  - RPC (Llamadas a procedimiento remoto)
  - RMI (Invocaciones a procedimiento remoto)
  - Sistemas de notificaciones de eventos
  - Sistemas de “streams” o cola avanzados



# Heterogeneidad

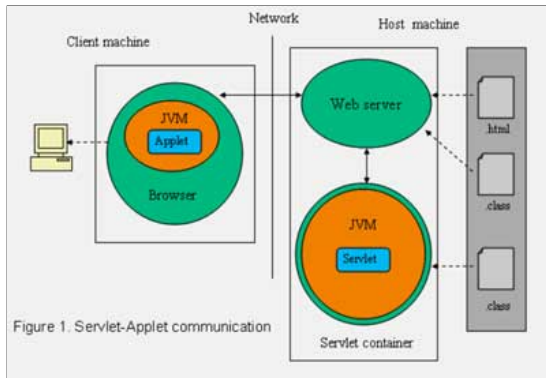
## Servicios del Middleware

- Servicios para sistemas de información: servicios que permitan gestionar datos de manera distribuida
  - Servicios de nombres distribuidos
  - Servicios de directorio y búsquedas distribuidos
  - Servicios de seguimiento de recursos móviles
  - Servicios de persistencia, caching y replicación de datos
- Servicios de control: servicios que permiten controlar a las aplicaciones cómo, cuándo y dónde acceden los datos
  - Servicios de transacciones distribuidas
  - Servicios de migración de código
- Servicios de seguridad
  - Servicios de autenticación y autorización
  - Servicios de cifrado
  - Servicios de auditoría

# Heterogeneidad

## y código movil

El **código movil** se emplea para referirse al código que puede ser enviado desde un computador a otro y ejecutarse en este.



La aproximación de máquina virtual provee un modo de crear código ejecutable sobre cualquier hardware.

## Sección 3 | Extensibilidad

# Extensibilidad

## Openness

Característica que determina si el sistema puede ser extendido y reimplementado en diversos aspectos.

- No es posible obtener extensibilidad a menos que la especificación y documentación de las interfaces software clave de los componentes de un sistema estén disponibles para los desarrolladores.
- Según estimaciones el 80 % de los usuarios de los programas que se hacen hoy en día no son personas humanas, sino otros programas
- Los sistemas diseñados para dar soporte a la compartición de recursos se etiquetan como sistemas distribuidos abierto (Open Distributed Systems).
- Los sistemas puede ser extensibles a nivel de hardware mediante por ejemplo la inclusión de nuevos servidores
- A nivel de software por la introducción de nuevos servicios y reimplementación de los antiguos.

# Extensibilidad

## Sistema abierto

- Un sistema de **software es abierto** si todas sus especificaciones son públicas. Esto incluye las normas y estándares internacionales sobre los que se apoya, la arquitectura del sistema y la especificación de las interfaces que son necesarias para poder interactuar con él, compartir información y extenderlo. Lo contrario de un sistema abierto es un sistema **cerrado o propietario**
- La información disponible de un sistema debe incluir: la **signatura** de todas las operaciones que componen sus interfaces, los **protocolos** de comunicación que utiliza, las **funciones y servicios que requiere** de otros programas y aplicaciones externas, los **formatos de representación** de los datos que maneja, así como el **significado y semántica** de los datos y operaciones que implementa.

# Extensibilidad

## Organizaciones independientes que publican normas

- **ISO** (International Organization for Standardization, <http://www.iso.org>) y el subcomité JTC (Joint Technical Committee 1 <http://www.jtc1-sc7.org>) encargado de normas sobre la **Ingeniería del Software y de los Sistemas**.
- **W3C** (World Wide Web Consortium, <http://www.w3c.org>), cuyas normas cubren los lenguajes, protocolos e interfaces de los sistemas web. (XML - Extensible Markup Language, WSDL-Web Service Description Language, SOAP-Simple Object Access Protocol)
- La **IEEE Computer Society** (<http://www.computer.org>) es la sociedad dependiente de la IEEE (Institute of Electrical and Electronics Engineers) dedicada a temas de software. Entre las normas más relevantes de la IEEE Computer Society están las relativas a la medición del software y la de descripción arquitectónica de sistemas de software (IEEE Std. 1471)

# Extensibilidad

## Organizaciones independientes que publican normas

- **ITU-T** (International Telecommunication Union, <http://www.itu.int>) está especializada en la especificación de normativa sobre telecomunicaciones. En el ámbito del software, las especificaciones más relevantes establecidas por ITU-T incluyen, entre otras, las de los lenguajes de especificación de protocolos, como, por ejemplo, SDL (Specification and Description Language) y MSC (Message Sequence Charts).
- **OMG** (Object Management Group, <http://www.omg.org>) es un consorcio internacional sin ánimo de lucro compuesto por más de ochocientas empresas y organizaciones, cuya misión es impulsar, mejorar y ofrecer mecanismos para garantizar la interoperabilidad entre los sistemas de software. UML (Unified Modelling Language), MOF (Meta-Object Facility), CORBA (Common Object Request Broker Architecture) y los relativos a la nueva iniciativa MDA (Model Driven Architecture, <http://www.omg.org/mda>).



# Extensibilidad

## Estándares de Fabricantes

Hay fabricantes que han definido notaciones, protocolos o arquitecturas propias, pero que han terminado haciéndose del dominio público y convirtiéndose en estándares de facto.

- Los protocolos de la familia TCP/IP fueron desarrollados en 1972 por el Departamento de Defensa de Estados Unidos, y terminaron convirtiéndose en los protocolos de comunicación más extendidos en Internet.
- La arquitectura de componentes Java 2 Enterprise Edition (J2EE) fue propuesta inicialmente por Sun Microsystems. Compite con la arquitectura definida para CORBA por la OMG y con la propuesta .NET de Microsoft.
- IBM ha desarrollado numerosos lenguajes y protocolos de coordinación, que han terminado superando a los desarrollados por W3C. Un ejemplo es BPEL4WS (Business Process Execution Language for Web Services), un lenguaje de coordinación para servicios web ampliamente extendido hoy en día (junto con la propuesta WS-CDL de W3C).

# Extensibilidad

## Importancia de las normas

- **Vida más larga.** Mientras se respeten las especificaciones del sistema será una implementación válida y, por tanto, podremos ir actualizando sus componentes por nuevas versiones cuando éstas aparezcan, o reemplazando algunas de sus partes por implementaciones de otros fabricantes si estas últimas ofrecen mejores características
- **Abaratar costes, aumentar la calidad.** Al permitir contar con distintas implementaciones del sistema, o partes del mismo donde además podremos integrar nuestra aplicación sin problemas.
- **Vocabulario preciso.** Que pueda ser entendido y compartidos por todas las áreas de la organización interna, por los clientes y proveedores externos.
- **Integrables.** Permitir que los sistemas se integren con sistemas de la propios de la empresa o externos.

# Extensibilidad

## Sistema de código abierto

Un sistema es de **código abierto** si ofrece el código fuente del programa, de forma que el usuario pueda modificarlo y recompilarlo si fuera necesario.

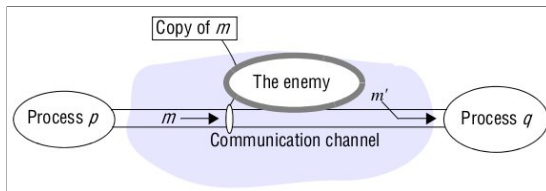
- Los sistemas operativos tipo Unix son sistemas abiertos, aunque no todas las implementaciones de Unix que hay en el mercado son de código abierto. Linux es un ejemplo de un sistema operativo abierto y de código abierto, mientras que Solaris es una implementación comercial de Unix.
- CORBA es un sistema middleware abierto, puesto que sus especificaciones son públicas. Al igual que ocurre con Unix y con otros sistemas abiertos, actualmente se dispone tanto de implementaciones comerciales de CORBA como de código abierto.

## Sección 4 | Seguridad

# Seguridad

## Security

- La seguridad de los recursos debe cubrir tres aspectos
  - Confidencialidad
  - Integridad
  - Disponibilidad
- Existen componentes vulnerables
  - Canales
  - Procesos
- Problemas actuales
  - Ataques denegación de servicio
  - Seguridad del código móvil (wireless)



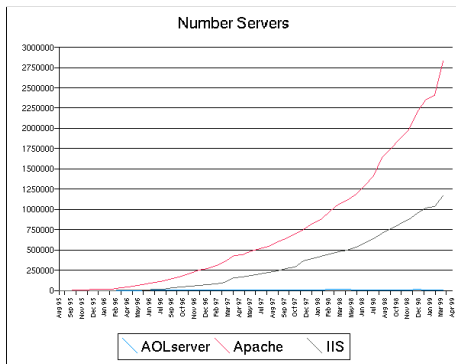
## Sección 5 | Escalabilidad

# Escalabilidad

## Scalability

Un sistema es **escalable** si conserva su efectividad cuando ocurre un **significante incremento en**

- Número de recursos.
- Número de usuarios.
- La arquitectura y la implementación deben permitirlo.
- Los algoritmos deben ser eficientes antes circunstancias determinadas.



# Escalabilidad

## Restos que se afrontan

- Control del coste de los recursos físicos. Según crece la demanda de un recurso debería ser posible extender el sistema a un coste razonable.
- Control de las pérdidas de prestaciones. Un incremento de tamaño traerá pérdida en prestaciones, se tardará más en acceder a la información.
- Prevención de desbordamiento. Un buen ejemplo es la escalabilidad del número de direcciones IP.
- Evitación de cuellos de botella. Antes el Sistema de Nombres de Dominio se alojaba en un solo archivo que se descargaba en un solo archivo. Con el crecimiento de Internet se convirtió en algo de administrar
- **Soluciones**
  - Comunicaciones asíncronas (formularios)
  - Cache
  - Distribución (particionamiento de la información)



## Sección 6 | T.Fallos

# T.Fallos

## Failure handling

Los fallos en un sistema distribuido son parciales; es decir, algunos componentes fallan mientras otros siguen funcionando.

- **Detección de fallos.** Cheksum para detección de datos corruptos
- **Enmascaramiento de fallos.** Los fallos han sido detectados, pueden ocultarse o atenuarse
  - Los mensajes pueden retransmitirse cuando falla la recepción
  - Discos redundantes
- **Tolerancia de fallos** Cuando se produce un fallo puede que no sea práctico detectar y ocultar los fallos (página no encontrada)
- **Recuperación frente a fallos** Cuando se recupera los datos pueden encontrarse en un estado inconsistente
- **Redundancia** Conseguir la tolerancia de fallos mediante:
  - Dos rutas diferentes de acceso a Internet.
  - El Sistema de Nombre de Dominios deberá estar replicado.
  - Base de datos replicadas en varios servidores.

## Sección 7 | Concurrency

# Concurrency

## Concurrency

Varios usuarios pueden acceder a la vez a un recurso compartido.

- Problemas de la concurrencia
  - Integridad de los datos.
  - Datos replicados. (Consistencia de la información)
  - Datos particionados. (Como determinar el estado de un sistema)
  - Mensajes. (Orden de los mensajes)
- No hay un reloj global

## Sección 8 | Transparencia

# Transparencia

## Transparency

- En el sistema se pueden producir fallos, pero el usuario no los va a notar; el sistema podrá crecer, pero para ello no habrá que parar el sistema; se podrán compartir recursos, pero no será necesario saber dónde están o a dónde se han movido.
- Las organizaciones ISO e ITU-T se plantearon la elaboración conjunta de una serie de estándares para el desarrollo de aplicaciones abiertas y distribuidas. Su objetivo era definir un modelo de referencia que permitiera integrar toda una serie de estándares sobre estos temas, asegurando la consistencia entre ellos. Dicho modelo se denomina modelo de referencia para el procesamiento abierto y distribuido, o RM-ODP (Reference Model – Open Distributed Processing).
- El manual de referencia ANSA y RM-ODP identifican ocho formas de transferencia.

# Transparencia

## características

Transparencia	Descripción
Acceso	Oculto diferencias en la representación de los datos y en cómo se accede a los recursos
Ubicación	Oculto dónde se ubican los recursos
Migración	Oculto el hecho de que un recurso puede migrar de un lugar a otro
Reubicación	Oculto el hecho de que un recurso puede moverse de un lugar a otro mientras se utiliza
Replicación	Oculto el hecho de que un recurso puede tener más de una réplica, haciendo indistinguible a los usuarios la réplica que realmente utilizan
Concurrencia	Oculto el hecho de que un recurso pueda ser utilizado simultáneamente por más de un usuario.
Fallos	Oculto el fallo y la recuperación de los recursos
Persistencia	Oculto el hecho de que un recurso esté ubicado en memoria volátil o en memoria persistente.

## Sección 9 | Bibliografía



- Jussi Kangasharju. Distributed Systems. University of Helsinki
- Coulouris, G., Dollimore, J., & Kindberg, T. (2005). Distributed systems: concepts and design.