#### Clase del día - 10/03/2021

La clase de hoy vamos a ver los requisitos de diseño y los tipos de los sistemas distribuidos.

**Requisitos de diseño**

El diseño de un sistema consiste en la definición de la **arquitectura** del sistema, la especificación detallada de sus componentes y la especificación del entorno tecnológico que soportará al sistema.

La arquitectura de un sistema puede verse como el "plano" dónde aparecen los componentes de software y hardware del sistema y sus interacciones. A partir de la arquitectura se establecen las especificaciones de construcción del sistema.

En la arquitectura se incluye la forma en que se particiona físicamente el sistema, la organización del sistema en sub-sistemas de diseño, la especificación del entorno tecnológico, los requisitos de operación, administración, seguridad, control de acceso, así como los requisitos de calidad, esto es, las características que el sistema debe cumplir.

A continuación veremos algunos de los requisitos de diseño de los sistemas distribuidos, también conocidos como requisitos arquitectónicos o requerimientos no funcionales.

Calidad de Servicio (QoS)

Los requisitos de **calidad de servicio** (QoS) son aquellos que describen las características de calidad que los servidores debe cumplir, como son los tiempos de respuesta, la tasa de errores permitida, la disponibilidad del servicio, el volumen de peticiones, seguridad, entre otras.

Balance de carga

Los sistemas distribuidos distribuyen procesamiento y datos. Para que un sistema distribuido sea eficiente, es necesario balancear la carga del procesamiento y del acceso a los datos, con la finalidad de evitar que uno o más computadoras se conviertan en un cuello de botella que ralentice el sistema completo.

Por tanto, es importante definir los **requisitos de balance de carga** del sistema, esto es, qué criterios se utilizarán para balancear la carga de procesamiento y de acceso a los datos.

Más adelante en el curso veremos cómo implementar el balance de carga en la nube.

Tolerancia a fallas

Como vimos anteriormente, un sistema distribuido es más tolerante a las fallas que un sistema centralizado, debido a que la falla en un componente de un sistema distribuido no necesariamente implica la falla del sistema completo, como es el caso de un sistema centralizado.

Los requisitos de tolerancia a fallas de un sistema distribuido definen las estrategias que el sistema implementará para soportar la falla en determinados componentes, algunas estrategias empleadas para la tolerancia a fallas son la replicación de datos y la replicación de código.

Seguridad

Posiblemente el requisito de diseño más importante es la seguridad, debido a las amenazas a las que se expone un sistema que se conecta a Internet.

Además de las vulnerabilidades del sistema operativo y del hardware, los sistemas introducen vulnerabilidades propias.

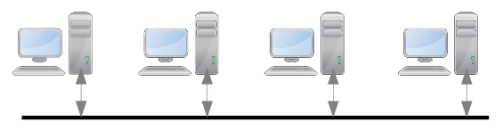
Por tanto, es muy importante definir los **requisitos de seguridad** para el sistema, entre otros: seguridad física del sistema, comunicación encriptada (SSL, TLS, HTTPS), utilización de usuarios no administrativos, configuración detallada de los permisos, programar para la prevención de ataques (p.e. SQL injection), seguridad en el proceso de desarrollo, etc.

**Tipos de sistemas distribuidos**

En clases anteriores vimos que podemos dividir los sistemas distribuidos en sistemas que distribuyen el procesamiento (cómputo) y sistemas que distribuyen los datos.

Los sistemas distribuidos de cómputo pueden a su vez dividirse en sistemas que ejecutan sobre un **cluster**y sistemas que ejecutan sobre una **malla**(*grid*).

Un cluster es un conjunto de computadoras homogéneas con el mismo sistema operativo conectadas mediante una red local (LAN) de alta velocidad.

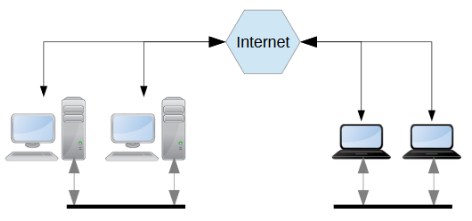


Los clusters se utilizan para el cómputo de alto rendimiento, dónde los programas se distribuyen entre los

diferentes nodos del cluster, con la finalidad de lograr rendimientos superiores.

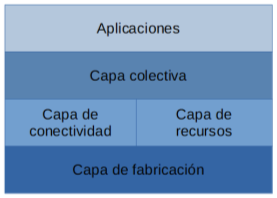
En el [TOP500](https://www.top500.org/statistics/list/) 474 sistemas son clusters, mientras que sólo 40 sistemas son MPP (*Massively Parallel Processing*). Un ejemplo de sistema MPP es la malla (grid).

Una malla es un conjunto de computadoras generalmente heterogéneas (hardware, sistema operativo, redes, etc.) agrupadas en organizaciones virtuales.



Una organización virtual es un conjunto de recursos (servidores, clusters, bases de datos, etc.) y los usuarios que los utilizan.

La arquitectura de una malla se puede dividir en cuatro capas:



La **capa de fabricación** está constituida por interfaces para los recursos locales de una ubicación. En esta capa se implementan funciones que permiten el intercambio de recursos dentro de la organización virtual, tales como consulta del estado del recurso, la capacidad del recurso, así como funciones administrativas para iniciar el recurso, apagar el recurso o bloquear el recurso.

La **capa de conectividad** incluye los protocolos de comunicación que utilizan los recursos para comunicarse, así como autenticación de usuarios y procesos.

La **capa de recursos** permite administrar recursos individuales incluyendo el control de acceso a los recursos (autorización).

La **capa colectiva** permite el acceso a múltiples recursos, incluyendo el descubrimiento de recursos, ubicación de recursos, planificación de tareas en los recursos, protocolos especializados.

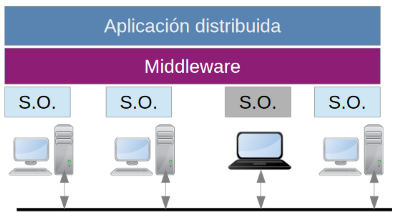
La **capa de aplicaciones** está compuesta por las aplicaciones que ejecutan dentro de la organización virtual.

**Middleware**

Un middleware (software enmedio) es una capa de software distribuido que actúa como “puente” entre las aplicaciones y el sistema operativo. Ofrece la vista de un sistema único en un ambiente de computadoras y/o redes heterogéneas.

La transparencia (datos, ubicación, migración, re-ubicación, replicación, concurrencia, fallas) de un sistema distribuido se implementa mediante middleware.

El middleware se distribuye entre las diversas máquinas ofreciendo a las aplicaciones una misma interfaz, no obstante las computadoras podrían ejecutar deferentes sistemas operativos.



Actividades individuales a realizar

En esta actividad veremos como crear una máquina virtual con Windows, cómo conectarse a la máquina virtual y cómo transferir archivos.

Es muy importante que el alumno elimine la máquina virtual una vez haya terminado de utilizarla, ya que mantener encendida una máquina virtual genera costo, lo que representa una disminución en el crédito que tiene el alumno como parte del programa Azure for Students.

**Creación de una máquina virtual con Windows**

1. En el portal de Azure seleccionar "Máquinas virtuales".

2. Seleccionar la opción "+Agregar".

3. Seleccionar el grupo de recursos o crear uno nuevo.

4. Ingresar el nombre de la máquina virtual.

5. Seleccionar la región dónde se creará la máquina virtual. Notar que el costo de la máquina virtual depende de la región.

6. Seleccionar la imagen, en este caso vamos a seleccionar Windows 10 Pro.

7. Seleccionar el tamaño de la máquina virtual, en este caso vamos a seleccionar una máquina virtual con al menos 2 GB de memoria.

8. Ingresar el nombre del usuario administrador y la contraseña.

9. En las "Reglas de puerto de entrada" se deberá dejar abierto el puerto 3389 para utilizar Remote Desktop Protocol (RDP).

10. Dar click en el botón "Siguiente: Discos>"

11. Seleccionar el tipo de disco de sistema operativo, en este caso vamos a seleccionar HDD estándar.

12. Dar click en el botón "Siguiente: Redes>"

13. Dar click en el botón "Siguiente: Administración>"

14. En el campo "Diagnóstico de arranque" seleccionar "Desactivado".

15. Dar click en el botón "Revisar y crear".

16. Dar click en el botón "Crear".

17. Dar click a la campana de notificaciones para verificar que la maquina virtual se haya creado.

18. Dar click en el botón "Ir al recurso".

19. Seleccionar la opción "Conectar". Seleccionar "RDP".

20. Dar click en el botón "Descargar archivo RDP".

21. Ejecutar "cmd" en la computadora local.

22. Vamos a crear un directorio en la computadora local. La máquina virtual recién creada va a ver este directorio como un disco lógico. Por ejemplo, el directorio se llamará "prueba". Ejecutar el siguiente comando en la ventana de Símbolo del sistema:

mkdir prueba

23. Ahora vamos a crear un disco lógico como alias del directorio creado. Ejecutar el siguiente comando:

subst f: prueba

Podemos ver que el disco lógico aparece en el explorador de archivos de Windows.

24. Buscar el archivo de conexión en la carpeta de descargas (un archivo con el nombre de la máquina virtual y la extensión ".rdp").

22. Dar click derecho al archivo de conexión y seleccionar "Editar".

23. Seleccionar la pestaña "Recursos locales".

24. Dar click en el botón "Mas..."

25. Abrir la sección "Unidades".

26. Marcar la casilla "Windows (F:)"

27. Dar click en el botón "Aceptar".

28. Dar click en el botón "Conectar" en la pantalla de advertencia.

29. Ingresar el nombre de usuario administrador y la contraseña.

30. Dar click en el botón "Sí" en la ventana de advertencia. Entonces se abrirá una ventana de escritorio remoto, la cual nos dará acceso al escritorio de la máquina virtual.

31. Configurar los parámetros de privacidad y dar click en el botón "Accept".

32. En la ventana "Networks" dar click en el botón "No".

33. Para ver el disco lógico creado en el paso 23, abrir el explorador de Windows de la máquina virtual. Entonces para enviar archivos desde la computadora local a la máquina virtual se deberá colocar los archivos en el directorio creado en el paso 22, y para enviar archivos desde la máquina virtual a la computadora local se deberá colocar los archivos en el disco F de la máquina virtual.

**Nota**. El teclado local podría no coincidir con la configuración del teclado de la maquina remota.

34. Para desconectarse de la máquina virtual, dar click en el botón "X" del escritorio remoto.