

**INFORME SOBRE SISTEMAS DISTRIBUIDOS**

**CURSO**

SISTEMAS DISTRIBUIDOS

**DOCENTE**

SAENZ TARAZONA, MANUEL ENRIQUE

**INTEGRANTES:**

* ARIAS SALCEDO, JEANPIERRE JESUS
* COCHACHIN QUITO, FRANK RONALD
* GONZALES ASTOCONDOR, MICHAEL
* MAZA CERNA, DENNIS ALEXIS
* MULLISACA BARRIENTOS, LUCIO ALBERTO

**2018**

Índice

[1. Introducción 5](#_Toc512212770)

[2. Sistemas Distribuidos 6](#_Toc512212771)

[2.2. Definición 6](#_Toc512212772)

[2.3. Características 6](#_Toc512212773)

[2.4. Objetivos: 6](#_Toc512212774)

[2.5. Tipo de transparencia: 7](#_Toc512212775)

[2.6. Tipos de Sistemas Distribuidos: 8](#_Toc512212776)

[2.6.1. Sistemas Distribuidos de Cómputo 8](#_Toc512212777)

[2.6.2. Sistemas de Computo en Cluster 8](#_Toc512212778)

[2.6.3. Sistemas de Computo en Grid 9](#_Toc512212779)

[3. Sistemas de Computo en Cluster 10](#_Toc512212780)

[3.2. Beneficios de la tecnología clúster 11](#_Toc512212781)

[3.3. Clasificación de los clústeres. 11](#_Toc512212782)

[3.4. Componentes de un clúster. 12](#_Toc512212783)

[A. Nodos: 12](#_Toc512212784)

[B. Almacenamiento: 13](#_Toc512212785)

[C. Sistemas Operativos: 13](#_Toc512212786)

[D. Conexiones de Red: 14](#_Toc512212787)

[E. Middleware: 16](#_Toc512212788)

[F. Ambientes de Programación Paralela: 16](#_Toc512212789)

[3.5. Sistemas de clústeres implementados 17](#_Toc512212790)

[A. Beowulf 17](#_Toc512212791)

[B. Berkeley NOW 17](#_Toc512212792)

[C. Google 17](#_Toc512212793)

[D. Cluster PS2 17](#_Toc512212794)

[E. Cluster X 17](#_Toc512212795)

[F. Red Española de Supercomputación 17](#_Toc512212796)

[G. Thunder 18](#_Toc512212797)

[H. ASCI Q 18](#_Toc512212798)

[3.6. Gestión de recursos distribuidos: Sistemas Gestores De Colas 18](#_Toc512212799)

[3.7. Balanceadores de carga: Linux Virtual Server 19](#_Toc512212800)

[3.8. Clústeres en aplicaciones científicas. 19](#_Toc512212801)

[3.9. Clústeres en aplicaciones empresariales. 20](#_Toc512212802)

[4. Computación Grid 21](#_Toc512212803)

[4.2. Computación grid, proyectos, aplicaciones y herramientas 22](#_Toc512212804)

[4.2.1. Proyectos de Grid Computing 22](#_Toc512212805)

[4.2.2. Aplicación de la computación Grid 24](#_Toc512212806)

[4.2.3. Herramientas informáticas de Grid 25](#_Toc512212807)

[5. Computación en la nube (Cloud computing) 34](#_Toc512212808)

[5.2. Cloud computing proyecto, aplicación y herramientas 35](#_Toc512212809)

[5.2.1. Proyectos de Cloud Computing 35](#_Toc512212810)

[5.2.2. Aplicación de Cloud Computing 36](#_Toc512212811)

[5.2.3. Herramientas de computación en la nube 37](#_Toc512212812)

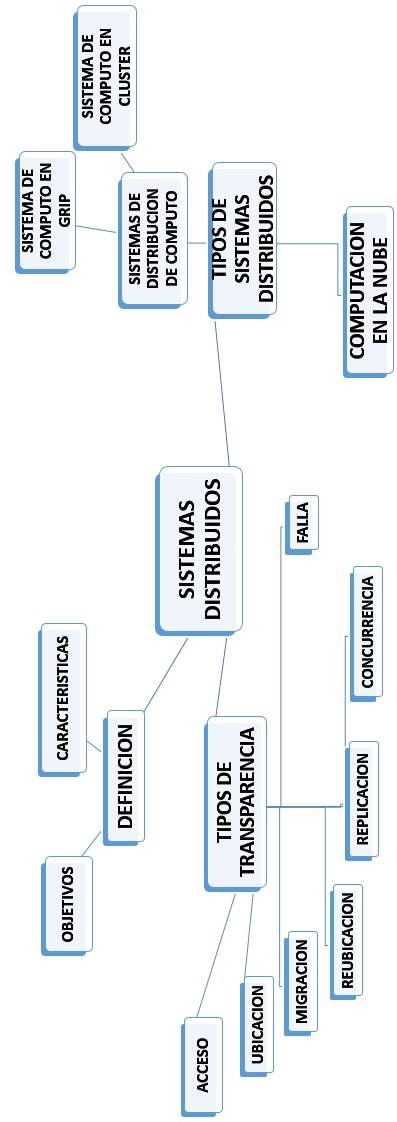
[5.3. Cluster, grid y cloud computing: comparación 37](#_Toc512212813)

[6. Discusión en cluster, grid y cloud 39](#_Toc512212814)

[7. Conclusión 39](#_Toc512212815)

[8. Bibliografía 40](#_Toc512212816)

**MAPA MENTAL**



# Introducción

El uso de una computación de alto rendimiento (HPC: high performance computing) fue previamente restringido a una organización que tiene la capacidad para permitirse exorbitante y altamente dedicado superordenador en el período. Por lo tanto, había una necesidad urgente para computadoras de alto rendimiento en relativamente pequeña escala a un costo comparativamente menor, solicita un enfoque más computacional como grupo informática. El advenimiento de la informática en clúster fue orquestado por un número variable de proyectos académicos tales como Berkerly NOW, HPVM y Beowulf. Sin embargo, la emergente demanda de internet y la disponibilidad de sofisticadas computadoras que incluyen componentes de productos básicos de bajo costo como las tecnologías de redes de alta velocidad han cambiado drásticamente cómo operamos la informática. La computación grid se originó a partir de la concepción académica durante la década de 1990 con el objetivo de permitir al usuario que cuando las veces que sus sistemas locales están ocupados, utilizar remotamente capacidades de computación inactivas en otros centros de computación. Al principio, era básicamente referido como computación en malla y adherentemente mente poseía comandos de poca audiencia. En consecuencia, después de muchos años de concertación los esfuerzos en el desarrollo se convirtieron en medios eficientes y efectivos de recursos informáticos compartidos y la solución de problemas en entorno virtual dinámico.

# Sistemas Distribuidos

## Definición

Un sistema distribuido se define como una colección de computadores  conectados por una red, y con el software distribuido adecuado para que el sistema sea visto por los usuarios como una única entidad capaz de proporcionar facilidades de computación.

Los sistemas distribuidos se implementan en diversas plataformas hardware, desde unas pocas estaciones de trabajo conectadas por una red de área local, hasta Internet, con una colección de redes  extensas interconectados, que en enlazan millones de ordenadores.

## Características

* **Concurrencia.-** Esta característica de los sistemas distribuidos permite que los recursos disponibles en la red puedan ser utilizados simultáneamente por los usuarios y/o agentes que interactúan en la red.
* **Carencia de reloj global.-** Las coordinaciones para la transferencia de mensajes entre los diferentes componentes para la realización de una tarea, no tienen una temporización general, está más bien distribuida a los componentes.
* **Fallos independientes de los componentes.-** Cada componente del sistema puede fallar independientemente, con lo cual los demás pueden continuar ejecutando sus acciones. Esto permite el logro de las tareas con mayor efectividad, pues el sistema en su conjunto continua trabajando.

## Objetivos:

* **Transparencia:** Se deberá de poder acceder a los distintos recursos sin conocer la localización de los mismos, es decir, para acceder a un recurso remoto no habrá que conocer de qué nodo depende.
* **Fiabilidad:**
* **Fiabilidad como disponibilidad:** es decir, se busca un sistema de alta disponibilidad mediante la redundancia de nodos y recursos.
* **Fiabilidad como coherencia:** se tiene que buscar que la información que procesa el sistema siempre sea coherente, aspecto que en sistemas en los que se utiliza la redundancia se dificulta bastante.
* **Rendimiento:**
* Debe ser mejor que un sistema centralizado y debe ser proporcional al número de procesadores empleados
* En este aspecto el principal problema es que a más número de procesadores más elementos críticos corren el riesgo de convertirse en cuellos de botella, por ejemplo la red de comunicaciones.
* **Escalabilidad:** Si se diseña de forma cuidadosa y planificada, el que el sistema crezca mediante la adición de nuevos nodos al sistema nos proporcionará un aumento del rendimiento proporcional con el número de procesadores que añadamos.
* **Flexibilidad:** La flexibilidad se entiende como la capacidad de ampliar o extender el sistema con nuevas funcionalidades de forma sencilla.

**Ejemplo:**

* **SETI@home:** es un experimento científico que utiliza ordenadores conectados a Internet para la búsqueda de inteligencia extraterrestre (en inglés SETI son las siglas de "Search for Extraterrestrial Intelligence"). Puedes participar ejecutando un programa libre que descarga y analiza datos obtenidos por radio telescopios.
* **Cluster Beowulf:** es un sistema de cómputo paralelo basado en clusters de ordenadores personales conectados a través de redes informáticas estándar, sin el uso de equipos desarrollados específicamente para la computación paralela. Fue desarrollado por primera vez en 1994 por Donald Becker y Thomas Sterling en la NASA, agruparon 16 procesadores Intel DX4 de 100 MHz, los interconectaron con tecnología Ethernet a 10 Mbps, en equipo viejos con Linux instalado como sistema operativo.

## Tipo de transparencia:

Es posible aplicar el concepto de transparencia a distintos aspectos de un sistema distribuido

|  |  |
| --- | --- |
| **TRANSPARENCIA** | **DESCRIPCIÓN** |
| **Acceso** | Oculta diferencias en la representación de los datos y la forma en que un recurso accede a los datos. |
| **Ubicación** | Oculta la localización de un recurso |
| **Migración** | Oculta el que un recurso pudiera moverse a otra ubicación |
| **Reubicación** | Oculta el que un recurso pudiera moverse a otra ubicación mientras está en uso. |
| **Replicación** | Oculta el número de copias de un recurso. |
| **Concurrencia** | Oculta que un recurso puede ser compartido por varios usuarios que compiten por él |
| **Falla** | Oculta la falla y recuperación de un recurso |

## Tipos de Sistemas Distribuidos:

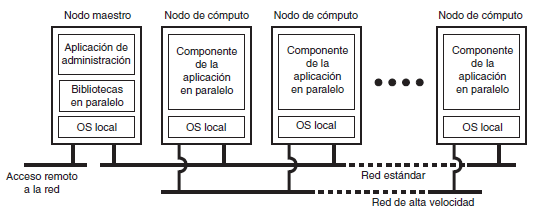
### Sistemas Distribuidos de Cómputo

Una clase importante de sistemas distribuidos es la utilizada para realizar tareas de cómputo de alto rendimiento. Hablando claro, podemos hacer una distinción entre dos subgrupos. En el cómputo en cluster, el hardware subyacente consta de una colección de estaciones de trabajo similares, o computadoras personales, conectadas cercanamente por medio de una red de área local de alta velocidad. Además, cada nodo ejecuta el mismo sistema operativo.

La situación se torna bastante diferente en el caso del cómputo en malla (grid). Este subgrupo consta de sistemas distribuidos construidos generalmente como un conjunto de sistemas de cómputo, en donde cada sistema podría caer dentro de un dominio administrativo diferente, y podría ser muy diferente cuando nos referimos al hardware, software, y la tecnología de red instalada.

### Sistemas de Computo en Cluster

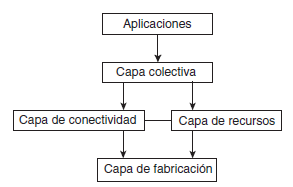
Los sistemas de cómputo en cluster adquirieron popularidad cuando mejoró la relación precio-rendimiento de las computadoras personales y las estaciones de trabajo. Hasta cierto punto, tanto financiera como técnicamente se volvió atractivo construir una supercomputadora usando tecnología directamente del estante mediante la simple conexión de una colección de computadoras sencillas ubicadas dentro de una red de alta velocidad. En virtualmente todos los casos, la computación en cluster se utiliza para la programación en paralelo donde un solo programa (de cálculo intensivo) corre paralelamente en múltiples máquinas.



### Sistemas de Computo en Grid

Una característica del cómputo basado en cluster es su homogeneidad. En la mayoría de los casos, las computadoras en cluster son en esencia las mismas, todas tienen el mismo sistema operativo, y están conectadas a través de la misma red. Por el contrario, los sistemas de cómputo basados en grid tienen un alto grado de heterogeneidad: no se hacen suposiciones de ninguna índole con respecto al hardware, sistemas operativos, redes, dominios administrativos, políticas de seguridad, etcétera.

Una cuestión clave en un sistema de cómputo en grid es reunir los recursos de diferentes organizaciones para permitir la colaboración de un grupo de personas o instituciones. Tal colaboración se realiza en la forma de una organización virtual. La gente que pertenece a la misma organización virtual tiene derechos de acceso a los recursos que proporciona la organización. Por lo general, los recursos constan de servidores (incluso supercomputadoras, posiblemente implementadas como clústeres de computadoras), facilidades de almacenamiento, y bases de datos. Además, también es posible agregar dispositivos especiales de red como telescopios, sensores, etcétera.



# Sistemas de Computo en Cluster

El término **clúster**​ (del inglés *cluster*, que significa *grupo* o *racimo*) se aplica a los conjuntos o conglomerados de [ordenadores](https://es.wikipedia.org/wiki/Ordenador) unidos entre sí normalmente por una red de alta velocidad y que se comportan como si fuesen una única computadora.

La tecnología de clústeres ha evolucionado en apoyo de actividades que van desde aplicaciones de supercómputo y *software* para aplicaciones críticas, servidores web y comercio electrónico, hasta bases de datos de alto rendimiento, entre otros usos.

El cómputo con clústeres surge como resultado de la convergencia de varias tendencias actuales que incluyen la disponibilidad de microprocesadores económicos de alto rendimiento y redes de alta velocidad, el desarrollo de herramientas de *software* para cómputo distribuido de alto rendimiento, así como la creciente necesidad de potencia computacional para aplicaciones que la requieran.

Simplemente, un clúster es un grupo de múltiples [ordenadores](https://es.wikipedia.org/wiki/Ordenador) unidos mediante una [red](https://es.wikipedia.org/wiki/Red_de_ordenadores) de alta velocidad, de tal forma que el conjunto es visto como un único ordenador, más potente que los comunes de escritorio.

Los clústeres son usualmente empleados para mejorar el rendimiento o la disponibilidad por encima de la que es provista por un solo computador típicamente siendo más económico que computadores individuales de rapidez y disponibilidad comparables.

De un clúster se espera que presente combinaciones de los siguientes servicios:

1. [Alto rendimiento](https://es.wikipedia.org/wiki/Cluster_de_alto_rendimiento)
2. [Alta disponibilidad](https://es.wikipedia.org/wiki/Cluster_de_alta_disponibilidad)
3. [Balanceo de carga](https://es.wikipedia.org/wiki/Cluster_de_balanceo_de_carga)
4. [Escalabilidad](https://es.wikipedia.org/wiki/Escalabilidad)

La construcción de los ordenadores del clúster es más fácil y económica debido a su flexibilidad: pueden tener todos la misma configuración de [*hardware*](https://es.wikipedia.org/wiki/Hardware) y [sistema operativo](https://es.wikipedia.org/wiki/Sistema_operativo)(clúster homogéneo), diferente rendimiento pero con arquitecturas y sistemas operativos similares (clúster semihomogéneo), o tener diferente hardware y sistema operativo (clúster heterogéneo), lo que hace más fácil y económica su construcción.

Para que un clúster funcione como tal, no basta solo con conectar entre sí los ordenadores, sino que es necesario proveer un sistema de manejo del clúster, el cual se encargue de interactuar con el usuario y los procesos que corren en él para optimizar el funcionamiento.

## Beneficios de la tecnología clúster

Las aplicaciones paralelas escalables requieren: buen rendimiento, baja latencia, comunicaciones que dispongan de gran ancho de banda, redes escalables y acceso rápido a archivos. Un clúster puede satisfacer estos requisitos usando los recursos que tiene asociados a él.

Los clústeres ofrecen las siguientes características a un costo relativamente bajo:

* Alto rendimiento
* Alta disponibilidad
* Alta eficiencia
* Escalabilidad

La tecnología clúster permite a las organizaciones incrementar su capacidad de procesamiento usando tecnología estándar, tanto en componentes de hardware como de software que pueden adquirirse a un costo relativamente bajo.

## Clasificación de los clústeres.

El término clúster tiene diferentes connotaciones para diferentes grupos de personas. Los tipos de clústeres, establecidos de acuerdo con el uso que se dé y los servicios que ofrecen, determinan el significado del término para el grupo que lo utiliza. Los clústeres pueden clasificarse según sus características:

* HPCC (*High Performance Computing Clusters:* clústeres de alto rendimiento).
* HA o HACC (*High Availability Computing Clusters:* clústeres de alta disponibilidad).
* HT o HTCC (*High Throughput Computing Clusters:* clústeres de alta eficiencia).

**Alto rendimiento**: Son clústeres en los cuales se ejecutan tareas que requieren de gran capacidad computacional, grandes cantidades de memoria, o ambos a la vez. El llevar a cabo estas tareas puede comprometer los recursos del clúster por largos periodos de tiempo.

**Alta disponibilidad**: Son clústeres cuyo objetivo de diseño es el de proveer disponibilidad y confiabilidad. Estos clústeres tratan de brindar la máxima disponibilidad de los servicios que ofrecen. La confiabilidad se provee mediante software que detecta fallos y permite recuperarse frente a los mismos, mientras que en hardware se evita tener un único punto de fallos.

**Alta eficiencia**: Son clústeres cuyo objetivo de diseño es el ejecutar la mayor cantidad de tareas en el menor tiempo posible. Existe independencia de datos entre las tareas individuales. El retardo entre los nodos del clúster no es considerado un gran problema.

Los clústeres pueden también clasificar en:

* clústeres de [IT](https://es.wikipedia.org/wiki/Tecnolog%C3%ADa_de_la_informaci%C3%B3n) comerciales (de alta disponibilidad y alta eficiencia) y
* clústeres científicos (de alto rendimiento).

A pesar de las discrepancias a nivel de requisitos de las aplicaciones, muchas de las características de las arquitecturas de hardware y software, que están por debajo de las aplicaciones en todos estos clústeres, son las mismas. Más aún, un clúster de determinado tipo, puede también presentar características de los otros.

## Componentes de un clúster.

En general, un clúster necesita de varios componentes de software y hardware para poder funcionar:

#### Nodos:

Pueden ser simples ordenadores, sistemas multiprocesador o [estaciones de trabajo](https://es.wikipedia.org/wiki/Estaci%C3%B3n_de_trabajo). En informática, de forma muy general, un nodo es un punto de intersección o unión de varios elementos que confluyen en el mismo lugar. Ahora bien, dentro de la informática la palabra nodo puede referirse a conceptos diferentes según el ámbito en el que nos movamos:

* En redes de computadoras cada una de las máquinas es un nodo, y si la red es Internet, cada servidor constituye también un nodo.
* En estructuras de datos dinámicas un nodo es un registro que contiene un dato de interés y al menos un puntero para referenciar (apuntar) a otro nodo. Si la estructura tiene solo un puntero, la única estructura que se puede construir con él es una lista, si el nodo tiene más de un puntero ya se pueden construir estructuras más complejas como árboles o grafos.

El clúster puede estar conformado por nodos dedicados o por nodos no dedicados.

En un clúster con nodos dedicados, los nodos no disponen de teclado, ratón ni monitor y su uso está exclusivamente dedicado a realizar tareas relacionadas con el clúster. Mientras que, en un clúster con nodos no dedicados, los nodos disponen de teclado, ratón y monitor y su uso no está exclusivamente dedicado a realizar tareas relacionadas con el clúster, el clúster hace uso de los ciclos de reloj que el usuario del computador no está utilizando para realizar sus tareas.

Cabe aclarar que a la hora de diseñar un clúster, los nodos deben tener características similares, es decir, deben guardar cierta similitud de arquitectura y sistemas operativos, ya que si se conforma un clúster con nodos totalmente heterogéneos (existe una diferencia grande entre capacidad de procesadores, memoria, disco duro) será ineficiente debido a que el middleware delegará o asignará todos los procesos al nodo de mayor capacidad de cómputo y solo distribuirá cuando este se encuentre saturado de procesos; por eso es recomendable construir un grupo de ordenadores lo más similares posible.

#### Almacenamiento:

El almacenamiento puede consistir en una [NAS](https://es.wikipedia.org/wiki/Network-attached_storage), una [SAN](https://es.wikipedia.org/wiki/Red_de_%C3%A1rea_de_almacenamiento), o almacenamiento interno en el servidor. El protocolo más comúnmente utilizado es NFS (Network File System), sistema de ficheros compartido entre servidor y los nodos. Sin embargo existen sistemas de ficheros específicos para clústeres como Lustre (CFS) y PVFS2.

Tecnologías en el soporte del almacenamiento en discos duros:

* [IDE o ATA](https://es.wikipedia.org/wiki/Integrated_Drive_Electronics): velocidades de 33, 66, 100, 133 y 166 [MB](https://es.wikipedia.org/wiki/Megabyte)/s
* [SATA](https://es.wikipedia.org/wiki/Serial_ATA): velocidades de 150, 300 y 600 MB/s
* [SCSI](https://es.wikipedia.org/wiki/Small_Computer_System_Interface): velocidades de 160, 320, 640 MB/s. Proporciona altos rendimientos.
* [SAS](https://es.wikipedia.org/wiki/Serial_Attached_SCSI): aúna SATA-II y SCSI. Velocidades de 300 y 600 MB/s
* Las unidades de cinta ([DLT](https://es.wikipedia.org/wiki/Digital_Linear_Tape)) son utilizadas para copias de seguridad por su bajo costo.

[NAS](https://es.wikipedia.org/wiki/Network-attached_storage) (Network Attached Storage) es un dispositivo específico dedicado al almacenamiento a través de red (normalmente [TCP/IP](https://es.wikipedia.org/wiki/TCP/IP)) que hace uso de un sistema operativo optimizado para dar acceso a través de protocolos [CIFS](https://es.wikipedia.org/wiki/Server_Message_Block), [NFS](https://es.wikipedia.org/wiki/Network_File_System), [FTP](https://es.wikipedia.org/wiki/File_Transfer_Protocol) o [TFTP](https://es.wikipedia.org/wiki/TFTP).

Por su parte, [DAS](https://es.wikipedia.org/wiki/Direct_Attached_Storage) (Direct Attached Storage) consiste en conectar unidades externas de almacenamiento SCSI o a una [SAN](https://es.wikipedia.org/wiki/Red_de_%C3%A1rea_de_almacenamiento) (*storage area network:* ‘red de área de almacenamiento’) a través de un [canal de fibra](https://es.wikipedia.org/wiki/Canal_de_fibra). Estas conexiones son dedicadas.

Mientras NAS permite compartir el almacenamiento, utilizar la red, y tiene una gestión más sencilla, DAS proporciona mayor rendimiento y mayor fiabilidad al no compartir el recurso.

#### Sistemas Operativos:

Un [sistema operativo](https://es.wikipedia.org/wiki/Sistema_operativo) debe ser [multiproceso](https://es.wikipedia.org/wiki/Multiproceso) y [multiusuario](https://es.wikipedia.org/wiki/Multiusuario). Otras características deseables son la facilidad de uso y acceso. Un sistema operativo es un programa o conjunto de programas de computadora destinado a permitir una gestión eficaz y segura de sus recursos. Comienza a trabajar cuando el [gestor de arranque](https://es.wikipedia.org/wiki/Gestor_de_arranque) carga en memoria su [núcleo](https://es.wikipedia.org/wiki/N%C3%BAcleo_(inform%C3%A1tica)) y gestiona el hardware de la máquina desde los niveles más básicos, permitiendo también la interacción con el usuario. Se puede encontrar normalmente en la mayoría de los aparatos electrónicos que utilicen microprocesadores para funcionar, ya que gracias a estos podemos entender la máquina y que ésta cumpla con sus funciones (teléfonos móviles, reproductores de DVD, radios, computadoras, etc.).

Ejemplos:

* [GNU/Linux](https://es.wikipedia.org/wiki/GNU_Linux)
  + [ABC GNU/Linux](https://es.wikipedia.org/wiki/ABC_GNU/Linux)[2](https://es.wikipedia.org/wiki/Cl%C3%BAster_(inform%C3%A1tica)#cite_note-2)​
  + [OpenMosix](https://es.wikipedia.org/wiki/OpenMosix)
  + [Rocks](https://es.wikipedia.org/wiki/Rocks_Clusters)[3](https://es.wikipedia.org/wiki/Cl%C3%BAster_(inform%C3%A1tica)#cite_note-3)​
  + [Kerrighed](https://es.wikipedia.org/w/index.php?title=Kerrighed&action=edit&redlink=1)
  + [Cóndor](http://www.cs.wisc.edu/condor/)
  + [Sun Grid Engine](https://es.wikipedia.org/wiki/Sun_Grid_Engine)
* [Unix](https://es.wikipedia.org/wiki/Unix)
  + [Solaris](https://es.wikipedia.org/wiki/Sistema_Operativo_Solaris)
  + [HP-UX](https://es.wikipedia.org/wiki/HP-UX)
  + [AIX](https://es.wikipedia.org/wiki/AIX)
* [Windows](https://es.wikipedia.org/wiki/Windows)
  + [NT](https://es.wikipedia.org/wiki/Windows_NT)
  + [2000 Server](https://es.wikipedia.org/wiki/Windows_2000_Server)
  + [2003 Server](https://es.wikipedia.org/wiki/Windows_2003_Server)
  + [2008 Server](https://es.wikipedia.org/wiki/Windows_Server_2008)
  + [2012 server](https://es.wikipedia.org/w/index.php?title=Windows_server_2012&action=edit&redlink=1)
* [Mac OS X](https://es.wikipedia.org/wiki/Mac_OS_X)
  + [Xgrid](http://www.apple.com/server/macosx/technology/xgrid.html)
* [Solaris](https://es.wikipedia.org/wiki/Solaris_(sistema_operativo))
* [FreeBSD](https://es.wikipedia.org/wiki/FreeBSD)

#### Conexiones de Red:

Los nodos de un clúster pueden conectarse mediante una simple red [Ethernet](https://es.wikipedia.org/wiki/Ethernet) con placas comunes (adaptadores de red o NICs), o utilizarse tecnologías especiales de alta velocidad como [Fast Ethernet](https://es.wikipedia.org/wiki/Fast_Ethernet), [Gigabit Ethernet](https://es.wikipedia.org/wiki/Gigabit_Ethernet), [Myrinet](https://es.wikipedia.org/wiki/Myrinet), [InfiniBand](https://es.wikipedia.org/wiki/InfiniBand), [SCI](https://en.wikipedia.org/wiki/Scalable_Coherent_Interface), etc.

* [Ethernet](https://es.wikipedia.org/wiki/Ethernet)
  + Son las redes más utilizadas en la actualidad, debido a su relativo bajo coste. No obstante, su tecnología limita el tamaño de paquete, realizan excesivas comprobaciones de error y sus protocolos no son eficientes, y sus velocidades de transmisión pueden limitar el rendimiento de los clústeres. Para aplicaciones con paralelismo de grano grueso puede suponer una solución acertada.
  + La opción más utilizada en la actualidad es Gigabit Ethernet (1 Gbit/s), siendo emergente la solución [10 Gigabit Ethernet](https://es.wikipedia.org/wiki/10_Gigabit_Ethernet) (10 [Gbit](https://es.wikipedia.org/wiki/Gbit)/s). La latencia de estas tecnologías está en torno a los 30 a 100 μs, dependiendo del protocolo de comunicación empleado.
  + En todo caso, es la red de administración por excelencia, así que aunque no sea la solución de red de altas prestaciones para las comunicaciones, es la red dedicada a las tareas administrativas.
* [Myrinet](https://es.wikipedia.org/wiki/Myrinet) (Myrinet 2000 y Myri-10G).
  + Su latencia es de 99 a 10 μs, y su ancho de banda es de 2 a 10 Gbit/s (para Myrinet 2000 y Myri-10G, respectivamente).
  + Es la red de baja latencia más utilizada en la actualidad, tanto en clústeres como en MPP; está presente en más de la mitad de los sistemas del top500. Tiene dos bibliotecas de comunicación a bajo nivel (GM y MX). Sobre estas bibliotecas están implementadas MPICH-GM, MPICH-MX, Sockets-GM y Sockets MX, para aprovechar las excelentes características de Myrinet. Existen también emulaciones IP sobre TCP/IP, IPoGM e IPoMX.
* [InfiniBand](https://es.wikipedia.org/wiki/InfiniBand)
  + Es una red surgida de un estándar desarrollado específicamente para realizar la comunicación en clústeres. Una de sus mayores ventajas es que mediante la agregación de canales (x1, x4 y x12) permite obtener anchos de banda muy elevados. La conexión básica es de 2 Gbit/s efectivos y con ‘quad connection’ x12 alcanza los 96 Gbit/s. No obstante, los startups no son muy altos, se sitúan en torno a los 10 μs.
  + Define una conexión entre un nodo de computación y un nodo de I/O. La conexión va desde un Host Channel Adapter (HCA) hasta un Target Channel Adapter (TCA). Se está usando principalmente para acceder a arrays de discos SAS.
* [SCI](https://es.wikipedia.org/wiki/SCI) *(scalable coherent interface)* IEEE standard 1596-1992
  + Su latencia teórica es de 1,43 μs y su ancho de banda de 5333 [Mbit](https://es.wikipedia.org/wiki/Mbit)/s bidireccional. Al poder configurarse con topologías de anillo (1D), toro (2D) e hipercubo (3D) sin necesidad de switch, se tiene una red adecuada para clústeres de pequeño y mediano tamaño.
  + Al ser una red de extremadamente baja latencia, presenta ventajas frente a Myrinet en clústeres de pequeño tamaño al tener una topología punto a punto y no ser necesaria la adquisición de un conmutador. El software sobre SCI está menos desarrollado que sobre Myrinet, pero los rendimientos obtenidos son superiores, destacando SCI Sockets (que obtiene startups de 3 microsegundos) y ScaMPI, una biblioteca MPI de elevadas prestaciones.
  + Además, a través del mecanismo de preloading (LD\_PRELOAD) se puede conseguir que todas las comunicaciones del sistema vayan a través de SCI-SOCKETS (transparencia para el usuario).

#### Middleware:

El [***middleware***](https://es.wikipedia.org/wiki/Middleware) es un software que generalmente actúa entre el sistema operativo y las aplicaciones con la finalidad de proveer a un clúster lo siguiente:

* Una interfaz única de acceso al sistema, denominada [SSI](https://es.wikipedia.org/wiki/SSI) (*Single System Image*), la cual genera la sensación al usuario de que utiliza un único ordenador muy potente;
* Herramientas para la optimización y mantenimiento del sistema: migración de procesos, *checkpoint-restart* (congelar uno o varios procesos, mudarlos de servidor y continuar su funcionamiento en el nuevo host), balanceo de carga, tolerancia a fallos, etc.;
* Escalabilidad: debe poder detectar automáticamente nuevos servidores conectados al clúster para proceder a su utilización.

Existen diversos tipos de middleware, como por ejemplo: [MOSIX](http://www.mosix.org/), [OpenMOSIX](http://openmosix.sourceforge.net/), [Cóndor](http://www.cs.wisc.edu/condor/), [OpenSSI](https://en.wikipedia.org/wiki/OpenSSI), etc.

El middleware recibe los trabajos entrantes al clúster y los redistribuye de manera que el proceso se ejecute más rápido y el sistema no sufra sobrecargas en un servidor. Esto se realiza mediante políticas definidas en el sistema (automáticamente o por un administrador) que le indican dónde y cómo debe distribuir los procesos, por un sistema de monitorización, el cual controla la carga de cada CPU y la cantidad de procesos en él.

El middleware también debe poder migrar procesos entre servidores con distintas finalidades:

* Balancear la carga: si un servidor está muy cargado de procesos y otro está ocioso, pueden transferirse procesos a este último para liberar de carga al primero y optimizar el funcionamiento;
* Mantenimiento de servidores: si hay procesos corriendo en un servidor que necesita mantenimiento o una actualización, es posible migrar los procesos a otro servidor y proceder a desconectar del clúster al primero;
* Priorización de trabajos: en caso de tener varios procesos corriendo en el clúster, pero uno de ellos de mayor importancia que los demás, puede migrarse este proceso a los servidores que posean más o mejores recursos para acelerar su procesamiento.

#### Ambientes de Programación Paralela:

Los ambientes de programación paralela permiten implementar algoritmos que hagan uso de recursos compartidos: [CPU](https://es.wikipedia.org/wiki/CPU) (*central processing unit:* ‘unidad central de proceso’), memoria, datos y servicios.

## Sistemas de clústeres implementados

##### Beowulf

En 1994, Donald Becker y Thomas Sterling construyeron la primera [Beowulf](https://es.wikipedia.org/wiki/Cluster_Beowulf). Fue construida con 16 computadores personales con procesadores Intel DX4 de 100 MHz, que estaban conectados a través de un switch Ethernet. El rendimiento teórico era de 3,2 GFlops.

##### Berkeley NOW

El sistema NOW de Berkeley estuvo conformado por 105 estaciones de trabajo Sun Ultra 170, conectadas a través de una red Myrinet. Cada estación de trabajo contenía un microprocesador Ultra1 de 167 MHz, caché de nivel 2 de 512 [KiB](https://es.wikipedia.org/wiki/KiB), 128 [MiB](https://es.wikipedia.org/wiki/MiB) de memoria, dos discos de 2,3 [GB](https://es.wikipedia.org/wiki/Gigabyte), tarjetas de red Ethernet y Myrinet. En abril de 1997, NOW logró un rendimiento de 10 GFlops.

##### Google

Durante el año 2003, el clúster Google llegó a estar conformado por más de 1,5 millones de computadores personales. Una consulta en Google lee en promedio cientos de megabytes y consume algunos billones de ciclos de [CPU](https://es.wikipedia.org/wiki/CPU).

##### Cluster PS2

En el año 2004, en la [Universidad de Illinois](https://es.wikipedia.org/wiki/Universidad_de_Illinois) (en [Urbana-Champaign](https://es.wikipedia.org/wiki/Urbana-Champaign), Estados Unidos), se exploró el uso de consolas Play Station 2 (PS2) en cómputo científico y visualización de alta resolución. Se construyó un clúster conformado por 70 PS2; utilizando Sony Linux Kit (basado en Linux Kondora y Linux Red Hat) y MPI.

##### Cluster X

En la lista «TOP 500» de noviembre de 2004 fue considerado el séptimo sistema más rápido del mundo; sin embargo, en julio de 2005 ocupaba la posición catorce. Cluster X fue construido en el Tecnológico de Virginia en el 2003; su instalación fue realizada por estudiantes de ese instituto. Está constituido por 2200 procesadores Apple G5 de 2,3 GHz. Utiliza dos redes: Infiniband 4x para las comunicaciones entre procesos y Gigabit Ethernet para la administración. Cluster X posee 4 [TiB](https://es.wikipedia.org/wiki/TiB) de [memoria RAM](https://es.wikipedia.org/wiki/Memoria_RAM) y 176 TB de disco duro, su rendimiento es de 12,25 [TFlops](https://es.wikipedia.org/w/index.php?title=TFlop&action=edit&redlink=1). Se lo conoce también como Terascale.

##### Red Española de Supercomputación

En el año [2007](https://es.wikipedia.org/wiki/2007) se creó la [Red Española de Supercomputación](https://es.wikipedia.org/wiki/Red_Espa%C3%B1ola_de_Supercomputaci%C3%B3n), compuesta por 7 clústeres distribuidos en distintas instituciones españolas.

Todos los clústeres (a excepción de las segundas versiones de [Magerit](https://es.wikipedia.org/wiki/Magerit) y [MareNostrum](https://es.wikipedia.org/wiki/MareNostrum), y el más reciente [Caléndula](https://es.wikipedia.org/wiki/Cal%C3%A9ndula_(supercomputador))) están formados por un número variable de nodos con procesadores [PowerPC 970FX](https://es.wikipedia.org/wiki/PowerPC_970FX) a 2,2 [GHz](https://es.wikipedia.org/wiki/GHz)interconectados con una red [Myrinet](https://es.wikipedia.org/wiki/Myrinet). El rendimiento de las máquinas oscilan entre los casi 65 [TeraFLOPS](https://es.wikipedia.org/wiki/FLOPS) proporcionados por los más de 10000 [núcleos](https://es.wikipedia.org/wiki/CPU) de [Marenostrum](https://es.wikipedia.org/wiki/Marenostrum), los casi 16 [TeraFLOPS](https://es.wikipedia.org/wiki/FLOPS) de [Magerit](https://es.wikipedia.org/wiki/Magerit) (primera versión) con 2400 procesadores o los casi 3 [TeraFLOPS](https://es.wikipedia.org/wiki/FLOPS) de los 5 nodos restantes.

La actualización de [Magerit](https://es.wikipedia.org/wiki/Magerit) en [2011](https://es.wikipedia.org/wiki/2011) mantiene la arquitectura clúster por su versatilidad y reemplazando los elementos de cómputo por nodos [IBM](https://es.wikipedia.org/wiki/IBM) PS702 con procesadores [POWER7](https://es.wikipedia.org/wiki/IBM_POWER) a 3,0 [GHz](https://es.wikipedia.org/wiki/GHz) y logrando un rendimiento más de 72 [TeraFLOPS](https://es.wikipedia.org/wiki/FLOPS) lo que le convierte en el más poderoso de España. Esto demuestra la sencillez y flexibilidad de la arquitectura: actualizando algunos elementos se obtienen sistemas más potentes sin grandes complicaciones.

##### Thunder

Thunder fue construido por el Laboratorio Nacional Lawrence Livermore de la Universidad de California. Está conformado por 4096 procesadores Intel Itanium2 Tiger4 de 1,4 GHz. Utiliza una red basada en tecnología Quadrics. Su rendimiento es de 19,94 TFlops. Se ubicó en la segunda posición del «TOP 500» durante junio de 2004, luego en la quinta posición en noviembre de 2004 y en la lista de julio de 2005 se ubicó en la séptima posición.

##### ASCI Q

ASCI Q fue construido en el año 2002 por el Laboratorio Nacional Los Álamos, Estados Unidos. Está constituido por 8192 procesadores AlphaServer SC45 de 1,25 GHz. Su rendimiento es de 13,88 TFlops. Se ubicó en la segunda posición del «TOP 500» durante junio y noviembre de 2003, luego en la tercera posición en junio de 2004, en la sexta posición en noviembre de 2004 y en la duodécima posición en julio de 2005.

## Gestión de recursos distribuidos: Sistemas Gestores De Colas

Los sistemas de gestión de colas, gestionan una cola de ejecución, planifican la ejecución de las tareas y gestionan los recursos, para minimizar costes y maximizar rendimiento de las aplicaciones.

* Funcionamiento:
* Los usuarios envían trabajos con qsub indicando requisitos de memoria, tiempo de procesador y espacio en disco.
* El gestor de recursos registra el trabajo.
* Tan pronto los recursos pedidos se hallen disponibles, el gestor de colas pone a ejecución el trabajo solicitado que según su planificación es el que tiene mayor prioridad. Se utiliza el planificador del gestor de colas en ausencia de planificadores más avanzados (como Maui / Moab cluster suite, los cuales pueden ser integrables en el sistema de colas).
* Se puede consultar el estado de los trabajos, en ejecución, en espera o terminados, a través de qstat
* Se puede eliminar un trabajo mediante qdel.
* El gestor de colas se configura con qconf.
* Salida estándar de trabajos: job.o#job
* Salida de error de trabajos: job.e#job
* Sistemas gestores de colas populares: Sun Grid Engine (SGE), PBS, Open PBS y Torque.

## Balanceadores de carga: Linux Virtual Server

* Linux irtual Server (LVS, IPVS en kernels 2.6.x) es un servicio de red altamente escalable y de alta disponibilidad que realiza:
* Equilibrado de carga mediante NAT (Network Address Translation), tunneling IP o enrutamiento directo (DR) por medio de un nodo maestro que da servicio a peticiones FTP y HTTP a los nodos de un clúster. Este servicio es provisto a nivel de kernel (ha de estar compilado el soporte para LVS/IPVS).
* NAT hace que el clúster funcione con una única IP pública, siendo los paquetes reescritos por el nodo maestro para ocultar los nodos internos. Solo es aceptable para un número pequeño de nodos, por la sobrecarga que acarrea.
* Tunneling IP es similar a NAT, pero el nodo maestro ya no reescribe los paquetes, siendo su tarea mucho más liviana.
* El enrutamiento directo (DR) es un sistema aún más ligero, pero necesita que todos los servidores compartan el mismo segmento de red.

## Clústeres en aplicaciones científicas.

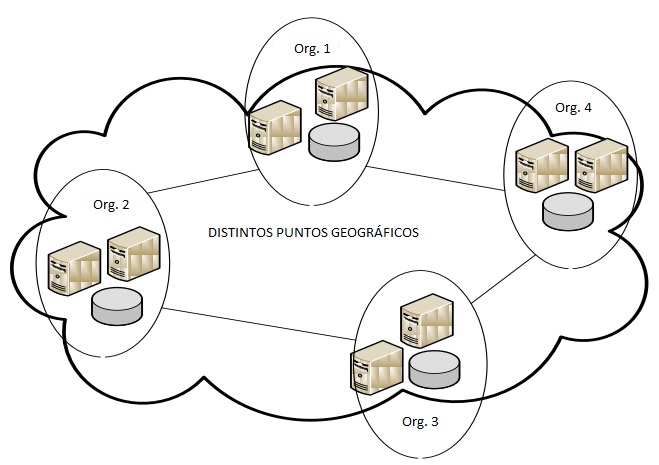
* Se suelen caracterizar por ser aplicaciones computacionalmente intensivas
* Sus necesidades de recursos son muy importantes en almacenamiento y especialmente memoria.
* Requieren nodos y sistemas dedicados, en entornos HPC y HTC.
* Suelen estar controlados los recursos por planificadores tipo Maui y gestores de recursos tipo PBS.
* Son en muchas ocasiones códigos legacy, difíciles de mantener, ya que los dominios de aplicación suelen ser difícilmente paralelizables.
* Ejemplos: Simulaciones (earth simulator), genómica computacional, predicción meteorológica (MM5), simulación de corrientes y vertidos en el mar, aplicaciones en química computacional.

## Clústeres en aplicaciones empresariales.

* Suelen ser aplicaciones no especialmente intensivas computacionalmente, pero que demandan alta disponibilidad y respuesta inmediata, con lo que los servicios se están ejecutando continuamente y no controlados por un sistema de colas
* Es usual que un sistema provea varios servicios. Una primera aproximación para realizar una distribución del trabajo es separar los servicios:
* Un servidor web con la BD en un nodo, el contenedor EJB en otro y el servidor de páginas web en otro constituye un claro ejemplo de distribución en el ámbito empresarial.
* Otra aproximación es instalar una aplicación web en un clúster squid como proxy-caché, apache/tomcat como servidor web/de aplicaciones web, memcached como caché de consultas a la base de datos y mysql como base de datos. Estos servicios pueden estar replicados en varios nodos del clúster.
* Ejemplos: [Flickr](https://es.wikipedia.org/wiki/Flickr), [Wikipedia](https://es.wikipedia.org/wiki/Wikipedia) y [Google](https://es.wikipedia.org/wiki/Google).

# Computación Grid

La computación Grid surge a mediados de 1990 con el propósito de resolver problemas de computación a gran escala combinando múltiples recursos de computación de varios grupos de sistemas autónomos, heterogéneos, geográficamente distribuidos como se muestra en las siguientes figuras.



<http://www.ijfcc.org/vol4/361-CS306/grid.jsp>



<https://www.ramonmillan.com/tutoriales/gridcomputing.php>

El término fue originalmente acuñado como una descripción de la tecnología informática que permitirá a los usuarios acceder a las potencias informáticas bajo demanda. La computación grid se define como un tipo de sistemas distribuidos que permiten la agregación, selección, intercambio de grupos de recursos autónomos geográficamente dispersos dinámicamente en tiempos de ejecución en función de su rendimiento, disponibilidad, costo y las necesidades de calidad de servicio del usuario. La idea innovadora detrás de la computación grid es permitir la agregación y procesamiento de recursos en múltiples ubicaciones para resolver un problema que no puede se resuelve utilizando la capacidad de procesamiento de una solo computadora. Conceptualmente, las iniciativas de computación grid se menciona que tiene la forma de la red de energía eléctrica, el enchufe de pared nos permite conectar infraestructura de recursos que genera y distribuye electricidad. Sin embargo, cuando accedemos a la red eléctrica, no consideramos que sea importante saber cómo está siendo la corriente eléctrica conectado a nuestra casa o donde se encuentra la planta y esto es similar al acceso a los recursos de los usuarios en grid.

La estrategia principal de computación grid es utilizar middleware para administrar diversos recursos informáticos en múltiples redes compartiendo y asignando tareas del programa entre varias computadoras al mismo tiempo, lo que les permite actuar o funcionar como una sola entidad virtual. Los dos objetivos del grid son: proporcionar a los usuarios acceso remoto a recursos TI y al mismo tiempo agregar el poder del procesamiento del sistema. El recurso principal de grid es el procesador, aunque también incluye aplicación, sistema de almacenamiento de datos, sensores, etc. El tamaño de la red difiere desde una pequeña estación de trabajo de una computadora de red dentro de una empresa a una gran colaboración en muchas redes organizacionales. Hay dos pilares importantes para el grid que son Open Grid Service Architecture y Globus Toolkit.

## Computación grid, proyectos, aplicaciones y herramientas

Esta sección se usará para delinear algunos aplicaciones, proyectos y herramientas en el entorno grid computing.

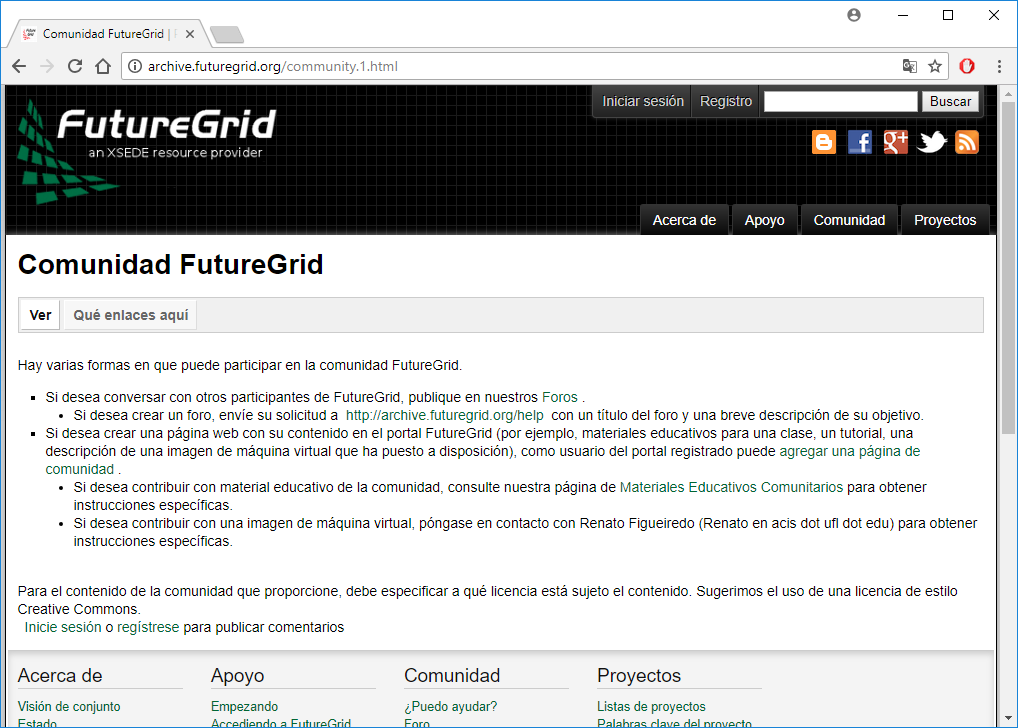
### Proyectos de Grid Computing

**EGI-InSPIRE (Integrated Sustainable Pan-European Infrastructure for Researchers in Europe)** proyecto: Iniciado en 2010 y financiado directamente por europeos Comisión durante cuatro años como un esfuerzo conjunto que abarca 50 instituciones en más de 40 países. El objetivo es establecer una infraestructura de red sostenible en Europa mediante unir nuevas infraestructuras informáticas distribuidas que incluye: sistemas de supercomputación, escritorio de nubes redes para los beneficios de los usuarios en el Espacio Europeo de Investigación comunidades. El objetivo crucial es establecer un entorno para científicos europeos, incluidos sus socios internacionales con una infraestructura eficiente y confiable que ayudará ellos tienen éxito en su mega análisis de datos de alto nivel.



**Major Grid Projects:**

* NASA, información de la red de energía.
* National Technological Grid.
* Particles Physics Data Grid, se formó en 1999 porque sus miembros eran muy conscientes de la necesidad de los servicios Data Grid para permitir el modelo de computación distribuida en todo el mundo de los experimentos actuales y futuros de alta energía y física nuclear. Inicialmente financiado por la iniciativa NGI y más tarde por los programas DOE MICS y HENP, ha brindado una oportunidad para el desarrollo temprano de la arquitectura Grid de datos, así como para la evaluación de prototipos de middleware Grid. PPDG desarrollará, adquirirá y entregará herramientas vitalmente necesarias habilitadas con Grid para los requerimientos intensivos de datos de la física nuclear y de partículas.
* European Data Grid, es una serie de esfuerzos para proporcionar acceso a recursos informáticos de alto rendimiento en toda Europa utilizando técnicas de computación grid. Vincula centros en diferentes países europeos para apoyar la investigación internacional en muchas disciplinas científicas. Después de una serie de proyectos de investigación como DataGrid y Enabling Grids para E-sciencE, la fundación se formó en 2010.
* Future Grid, hace posible que los investigadores aborden desafíos complejos de investigación en informática relacionados con el uso y la seguridad de redes y nubes. Estos incluyen temas que van desde autenticación, autorización, programación, virtualización, diseño de middleware, diseño de interfaz y ciberseguridad, hasta la optimización de esquemas computacionales habilitados para la nube y habilitados para la red para investigadores en astronomía, química, biología, ingeniería, ciencia atmosférica y epidemiología. El equipo del proyecto proporciona a la comunidad investigadora una nueva grilla de computación experimentales y un banco de pruebas en la nube, llamado FutureGrid, junto con el apoyo del usuario para investigadores externos que realizan experimentos en FutureGrid.



* Globus Alliance, es un software de red de código abierto específicamente diseñado para eliminar problemas relacionados con el uso compartido de recursos en sistemas distribuidos. Globus DemoGrid fue el primero en ser lanzado como una herramienta para el desarrollo de entorno de grilla instruccional capaz de ser desplegado usando máquinas virtuales en un recurso físico o en la nube ambientes. El objetivo del proyecto era proporcionar un entorno fácil y accesible con diferentes herramientas de grilla, sin la necesidad de instalar las herramientas o requerir una cuenta en la cuadrícula activa.

### Aplicación de la computación Grid

Drug Discovery Grid (DDGrid): El objetivo del proyecto es desarrollar plataformas para esfuerzos combinados hacia el descubrimiento de fármacos a través de P2P y tecnología de computación grid. Se espera que resuelva problemas de mega computación y aplicación intensiva de datos en el área de biología molecular, química y medicina usando grid middleware.

**Mammo Grid**: es una aplicación de red médica en la que se basa principalmente como arquitectura orientada a servicios. El objetivo es principalmente entregar un conjunto de prototipos evolutivos para mostrar análisis de mamo gramas, particularmente especialistas radiólogos trabajan en el cribado del cáncer de mama que puede utilizar la información de infraestructura grid para resolver problemas de análisis de imagen.

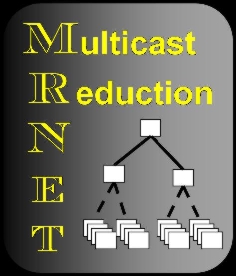
**Geodise (Grid Enabled Optimisation and Design Search for Engineering)** se utiliza principalmente para proporcionar una red basada en marco de integración para datos multidisciplinarios intensivos en tareas de optimización de diseño y computación.

### Herramientas informáticas de Grid

**Condor-G** es un proyecto de trabajo de Globus y Cóndor representación que permite la gran colección de recursos, la utilización se extiende a través de múltiples redes en tal forma como si pertenecen solo al dominio de usuario único o personal red. El uso de protocolos para acceso estandarizado y las comunicaciones entre dominios a sistemas remotos por lotes vienen de Globus mientras preocupaciones como recuperación de errores, trabajo presentación y creación amigable del entorno de ejecución viene de Cóndor.

**GRID computing and Business (Gridbus)**, es un clúster y proyecto de tecnología middleware del kit de herramientas de grid asociado con su diseño y desarrollo orientado al servicio informática. El kit de herramientas permite servicios de extremo a extremo agregar o arrendar servicios de recursos distribuidos basados ​​en su capacidad, disponibilidad, costo, requerimientos de QoS (Calidad de servicio) y performance.

**Paradyn**: desarrolla una tecnología que ayuda a los desarrolladores de herramientas y aplicaciones en su búsqueda de software de alto rendimiento, escalable, paralelo y distribuido. El proyecto principal, Paradyn, aprovecha una técnica llamada instrumentación dinámica para obtener de manera eficiente perfiles de rendimiento de ejecutables no modificados. Esta tecnología dinámica de instrumentación binaria está disponible independientemente para los investigadores a través de la API Dyninst. Otra investigación del proyecto Paradyn incluye la instrumentación dinámica de ejecutar núcleos de sistemas operativos, el proyecto Kerninst y el desarrollo de middleware para aplicaciones escalables, eficientes y robustas en la red de multidifusión / reducción (MRNet). MRNet es una red de superposición de software que proporciona comunicaciones eficientes de multidifusión y reducción para herramientas y sistemas distribuidos en paralelo. MRNet utiliza un árbol de procesos entre el front-end y los back-ends de la herramienta para mejorar el rendimiento de comunicación del grupo. Estos procesos internos también se utilizan para distribuir muchas actividades importantes de herramientas, reduciendo el tiempo de análisis de datos y manteniendo las cargas de front-end de la herramienta manejables.

<http://www.paradyn.org/mrnet/>

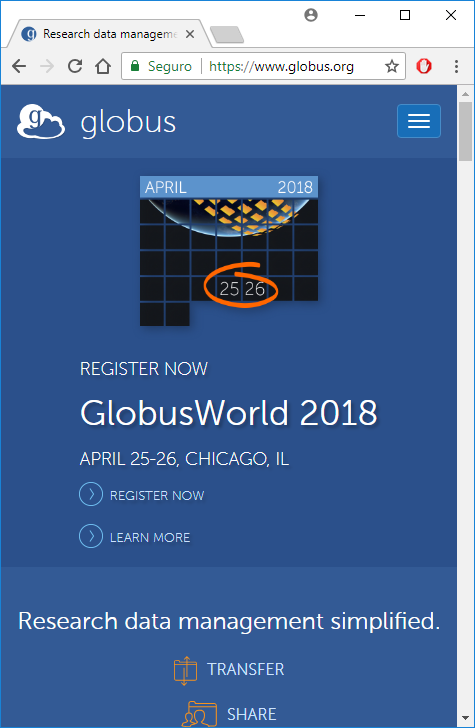
**Globus** es un juego de herramientas de software de código abierto que permite grid computacional y construcción de aplicaciones basadas en grid a través de fronteras geográficas.

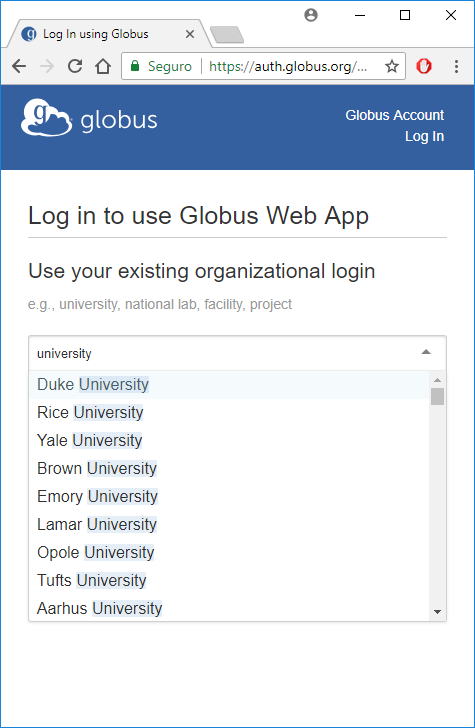
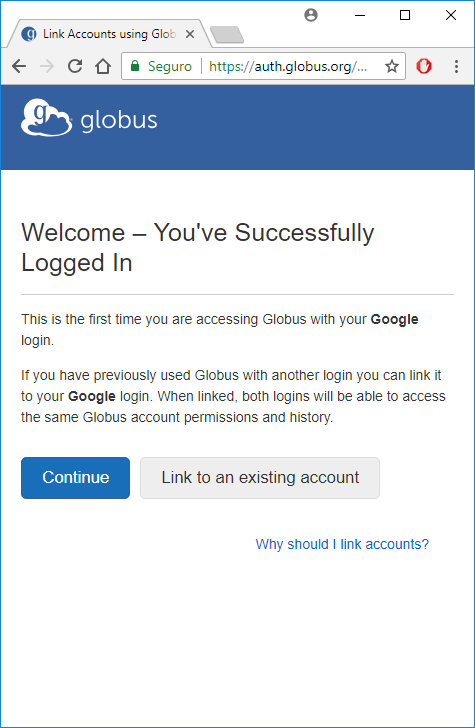
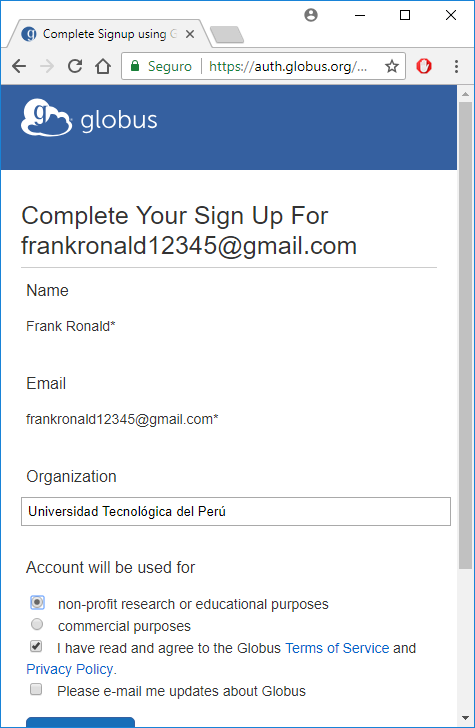
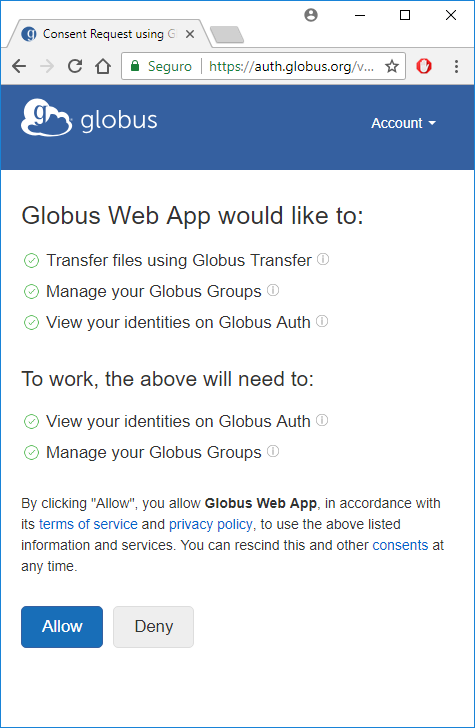
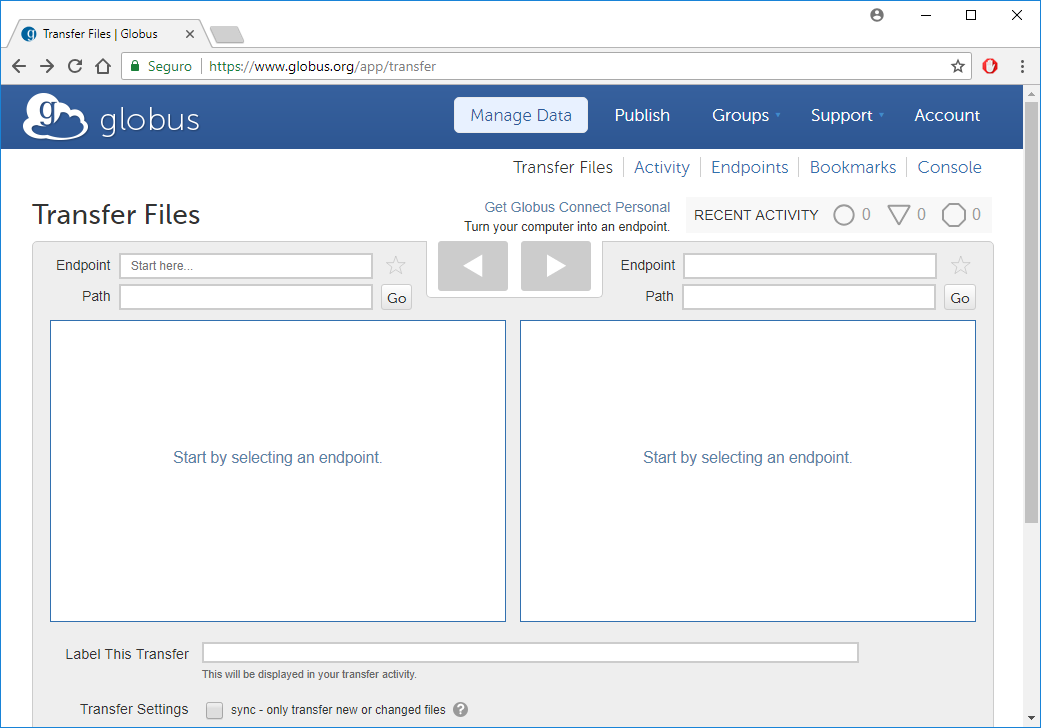
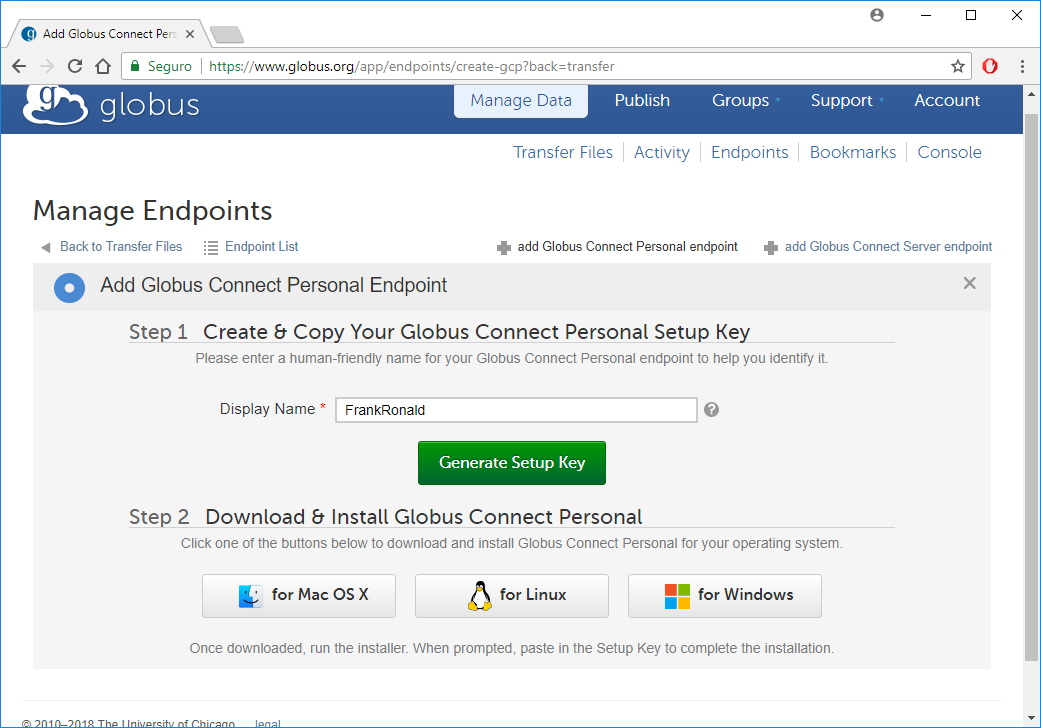
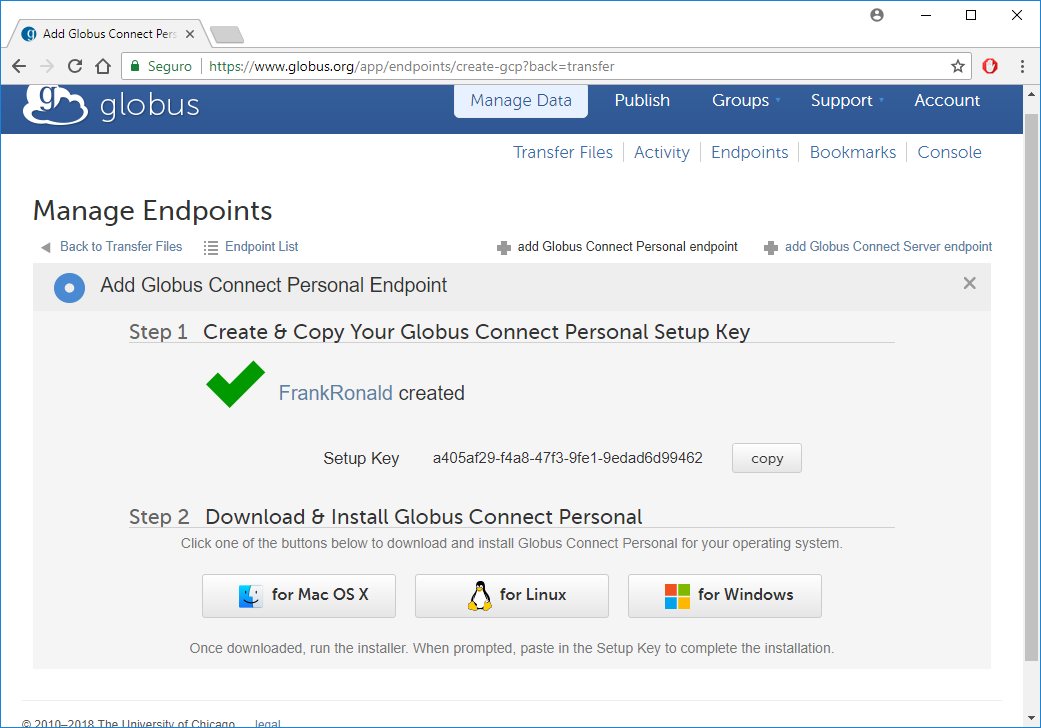
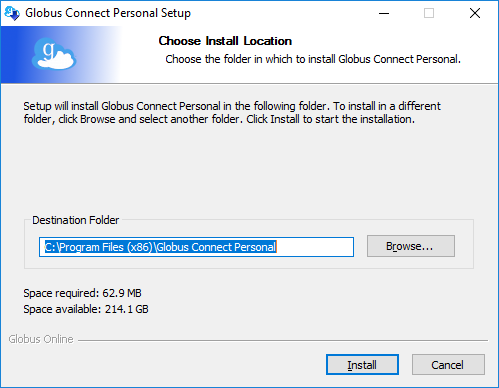
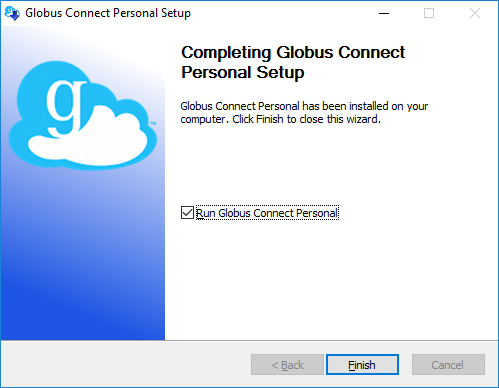
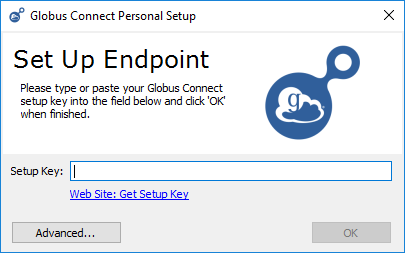
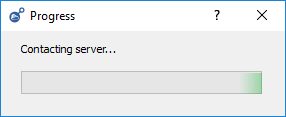
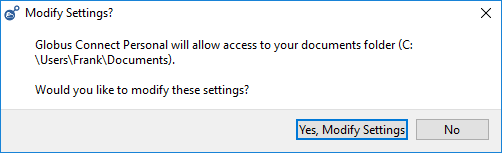
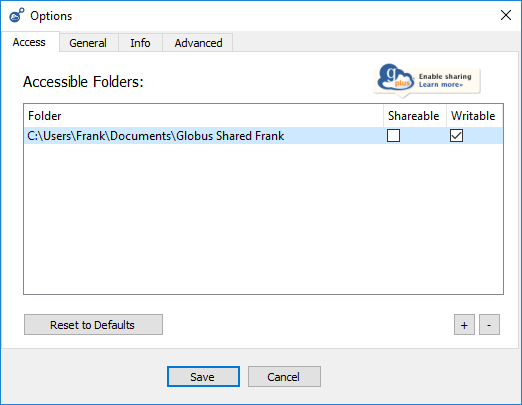
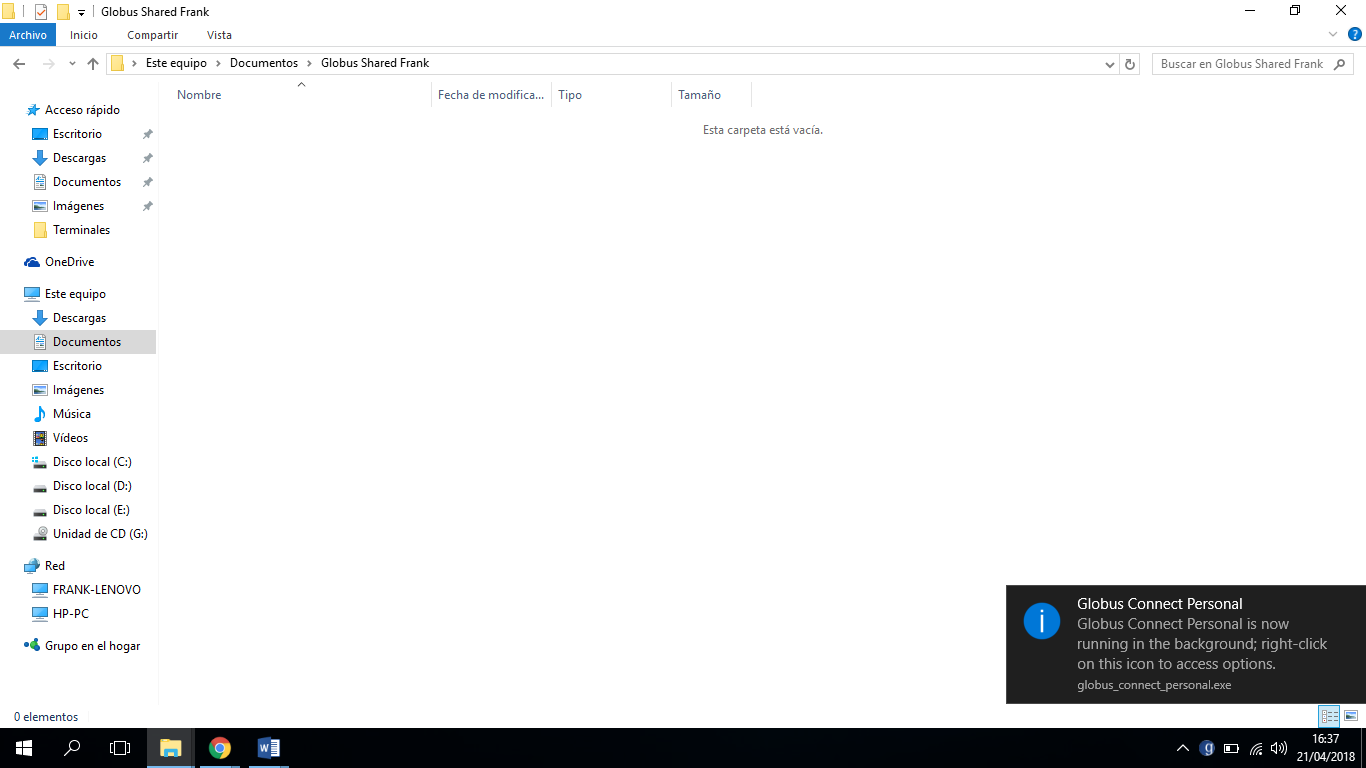
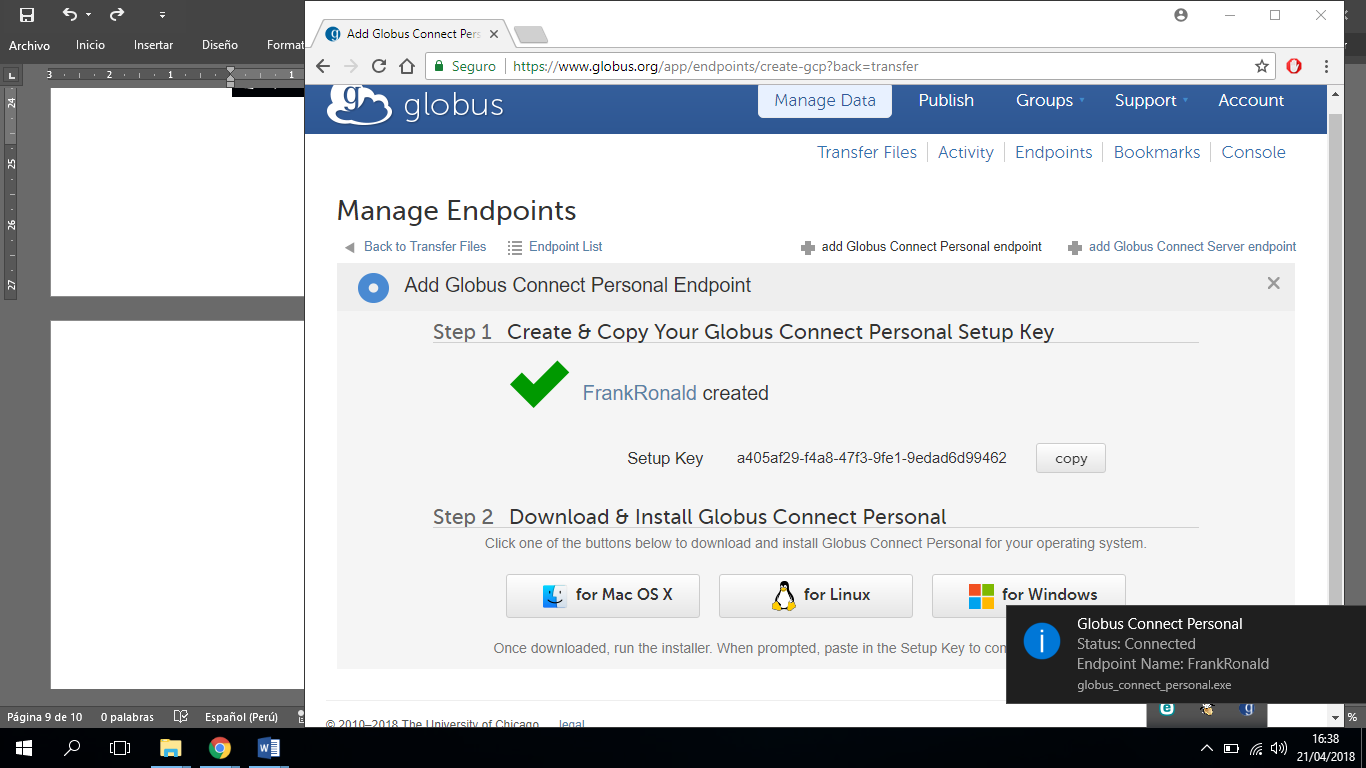
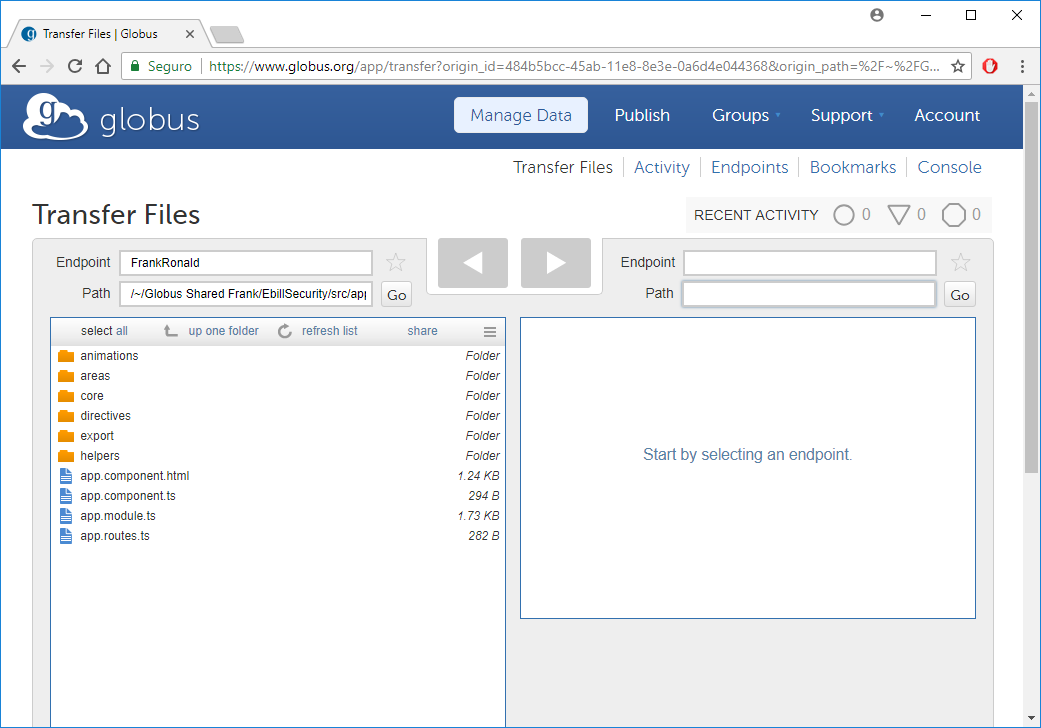
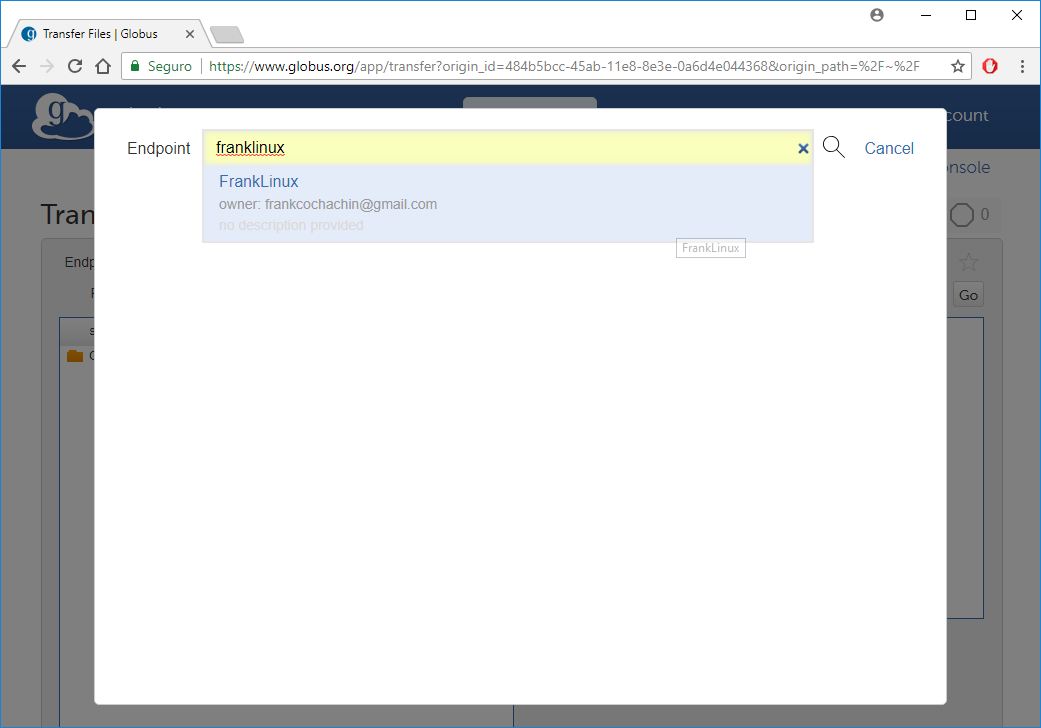
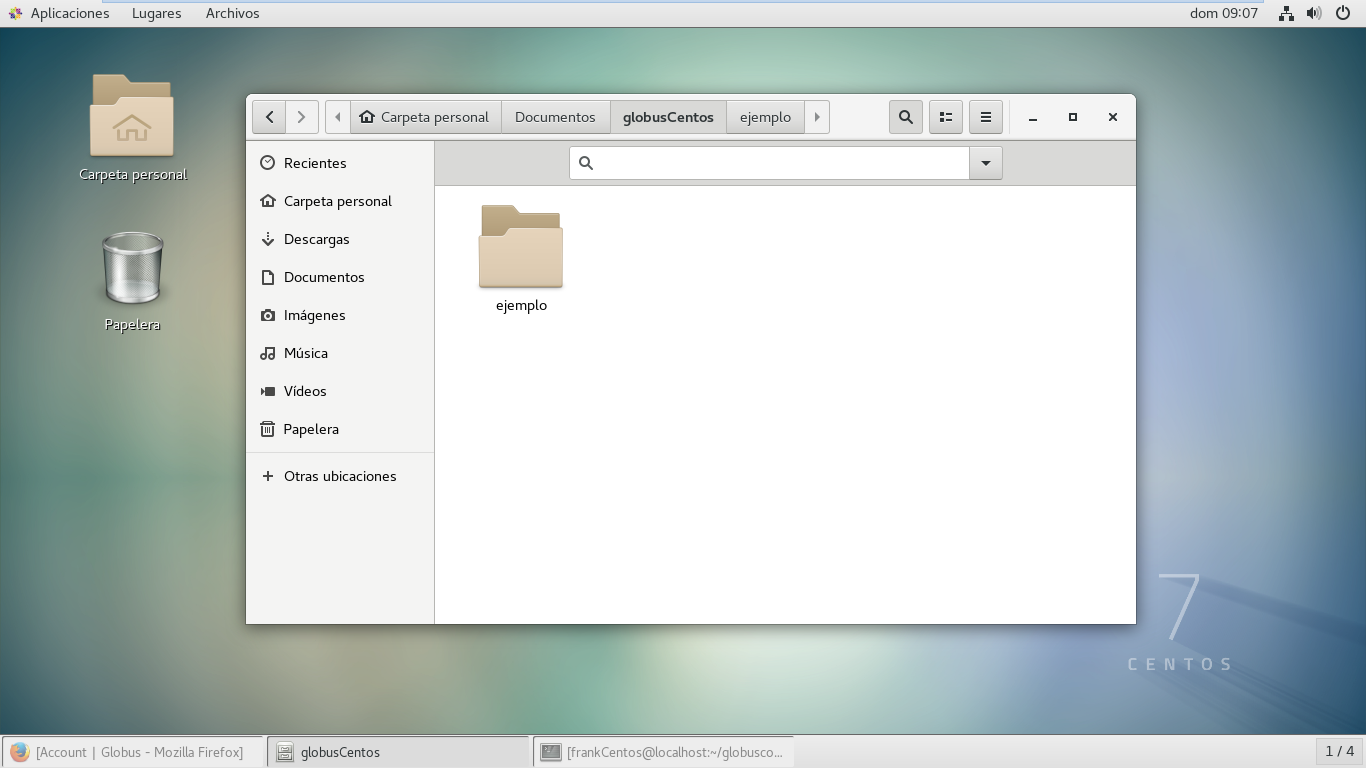
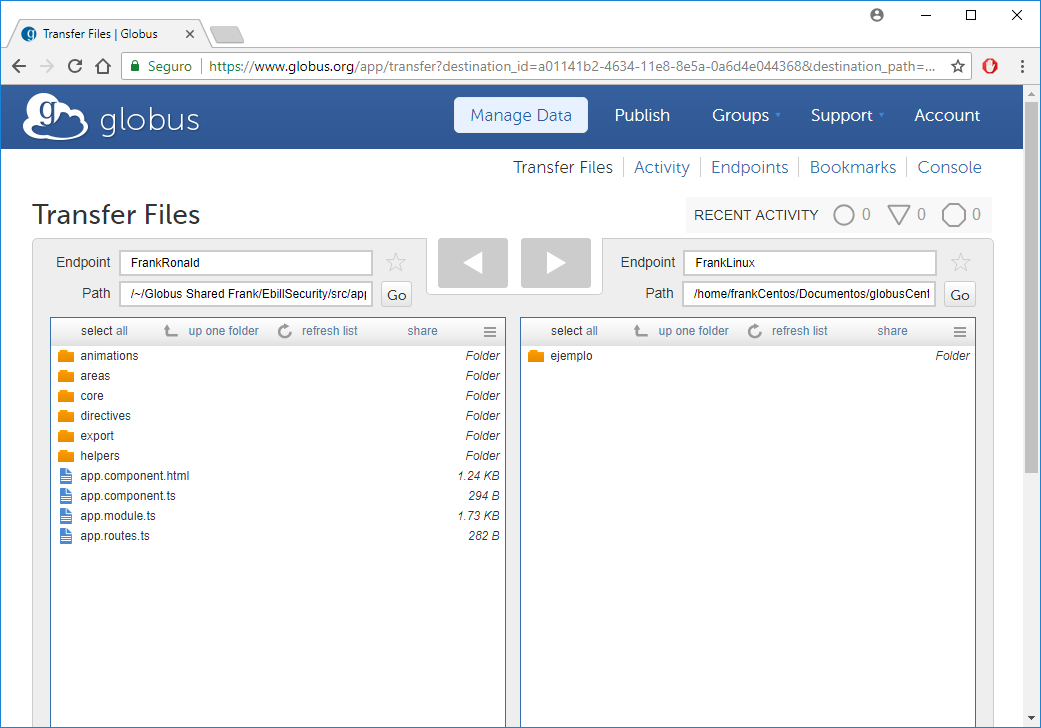
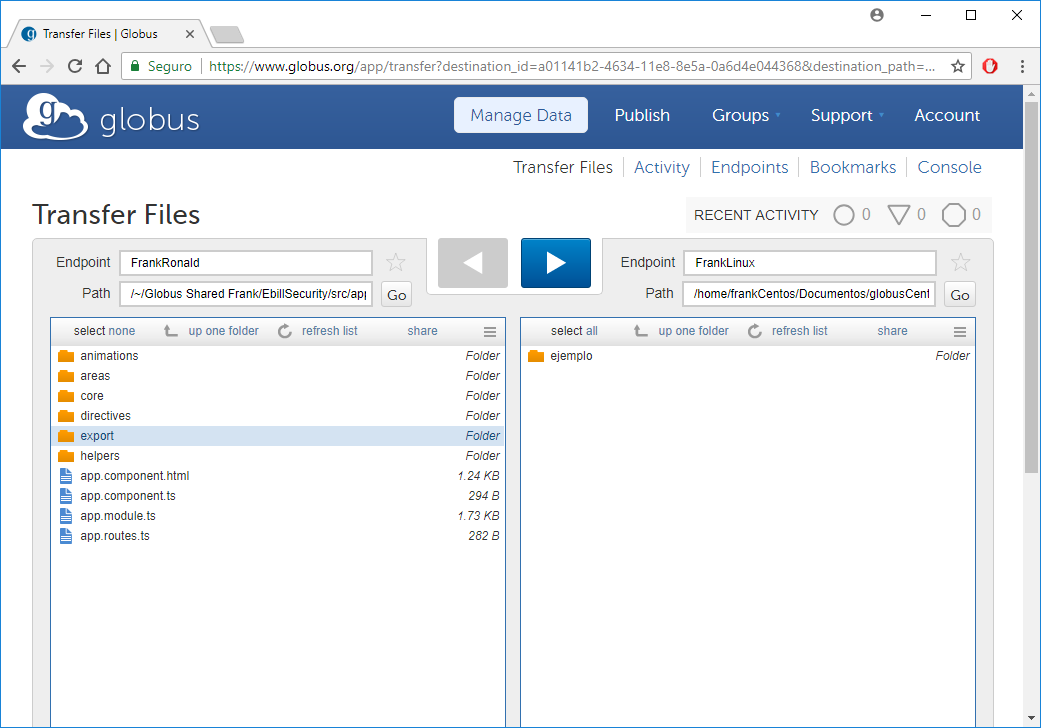
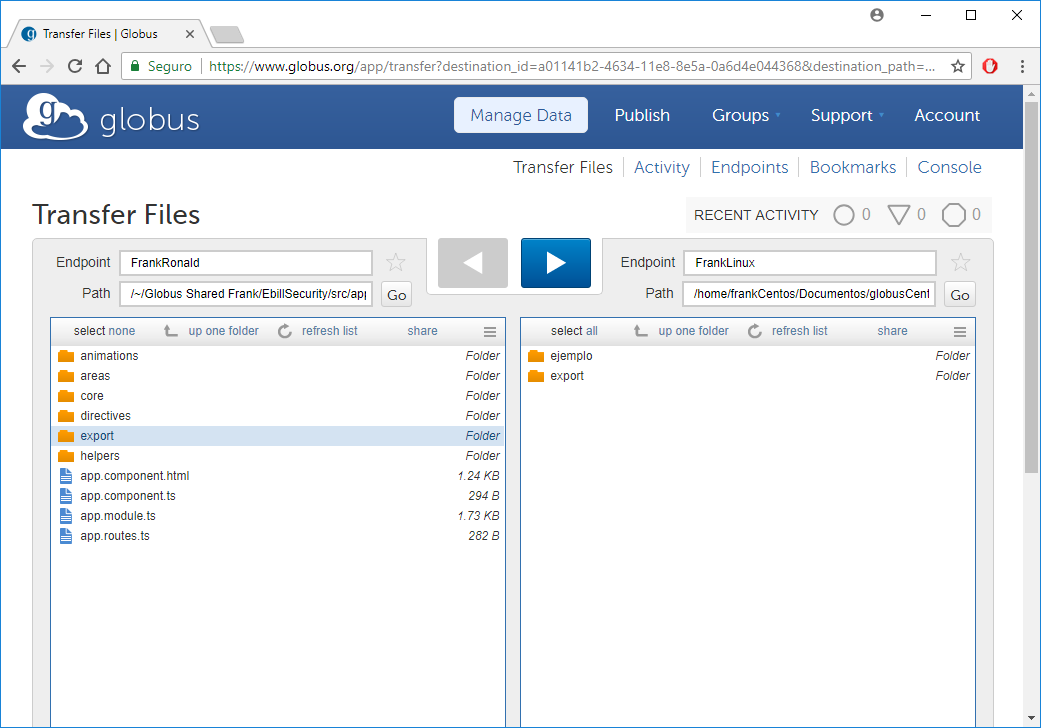
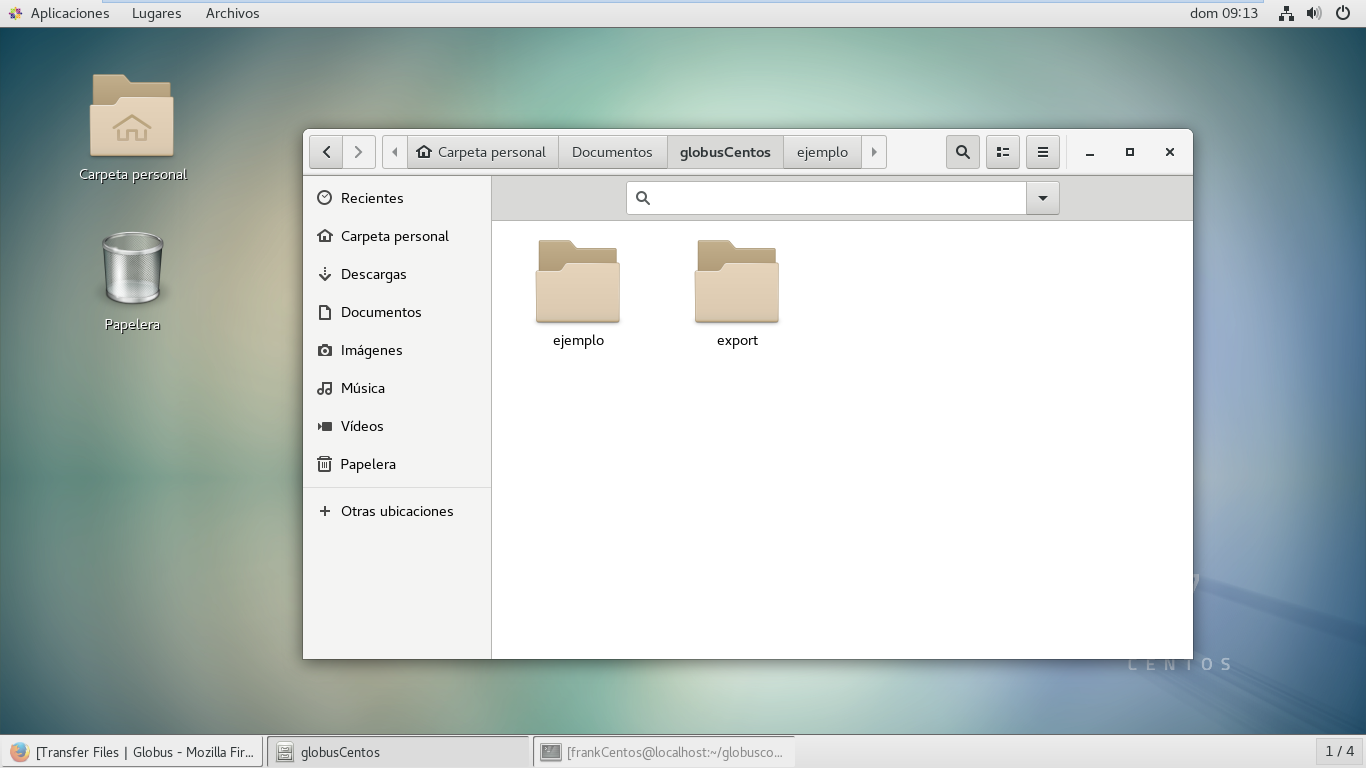
La Globus Alliance implementa alguno de los estándares desarrollados por el Open Grid Forum (OGF) en el Globus Toolkit que ellos han desarrollado. Un middleware publicado como software libre sobre el que se pueden construir servicios grid, los principales estándares que cumple son:

* Open Grid Services Architecture (OGSA)
* Open Grid Services Infrastructure (OGSI) -- especificación reemplazada por WSRF y WS-Management.
* Web Services Resource Framework (WSRF)
* Job Submission Description Language (JSDL)
* Distributed Resource Management Application API (DRMAA)
* WS-Management
* WS-BaseNotification
* SOAP
* WSDL
* Grid Security Infrastructure (GSI)

**Ejemplo**: Emplearemos un ejemplo de un servicio que nos ofrece la plataforma de Globus; para este ejemplo será el compartir archivos.

* Ingresamos a la plataforma globus.org



* Si participásemos de un grupo, podríamos buscarlo y unirnos
* 
* Nos logeamos
* 
* 
* 
* 
* 
* 
* 
* 
* 
* 
* 
* 
* 
* 
* 
* 
* 
* 
* 
* 
* 

**Legion**: es un metasistema basado en objetos que permite gestión de datos, autonomía del sitio, programación del núcleo de modo transparente, tolerancia a fallas y además un middleware con amplia gama de varias opciones de seguridad.

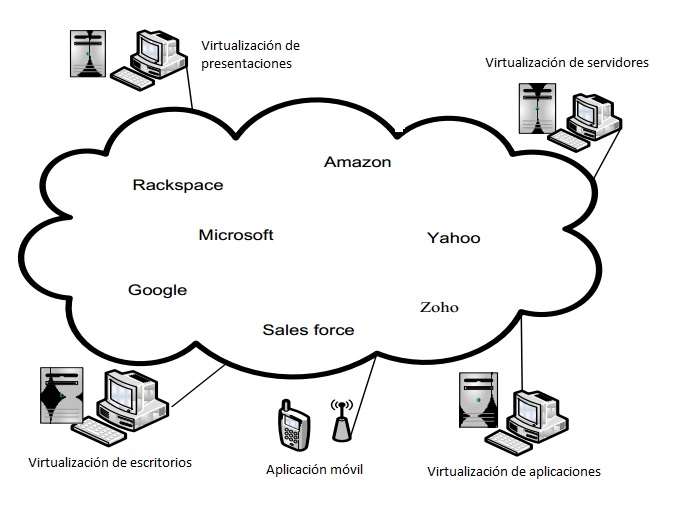
Metasistema, Al igual que los dispositivos de almacenamiento, los metasistemas deben incorporar recursos de capacidades variadas, enlazados por redes no confiables, situados en campos administrativos diferentes. Sin embargo, la necesidad de un rendimiento superior, puede necesitar modelos de programación e interfaces absolutamente diferentes.

**Nimrod-G**, utiliza el servicio middleware Globus para despachar trabajo a través de la grid computacional y al mismo tiempo efectúa el descubrimiento dinámico de recursos. Los ingenieros y científicos están habilitados para organizar de forma transparente el programa y datos en sitios remotos, así como ejecutar el programa en un particular elemento de un conjunto de datos específicamente en la diferencia de la máquina y finalmente, ensamble todos los resultados a un sitio de usuario desde el sitio remoto. Nimrod es una herramienta para la parametrización de programas en serie para crear y ejecutar programas embarazosamente paralelos a través de una grilla computacional. Nimrod fue una de las primeras herramientas para hacer uso de recursos heterogéneos en una grilla para un solo cálculo. La herramienta fue creada como un proyecto de investigación financiado por el Centro de Tecnología de Sistemas Distribuidos. El investigador principal es el profesor David Abramson de la Universidad de Monash

**Simuladores**: hay otros simuladores utilizados en la red informática como Optorsim, ZENTURIO, Chicsim y Gridsim.

# Computación en la nube (Cloud computing)

La computación en la nube es un paradigma informático novedoso basado en avance de la tecnología de virtualización que proporciona enormes grupos dinámicos y escalables de recursos virtuales de TI bajo demanda como un servicio a través de la tecnología de internet. Esta es un nuevo concepto computacional que magníficamente da paso a un moderno y eficiente significado transformacional para computación clúster, grilla y distribuida. La computación en la nube se refiere a representación de recursos de TI, como aplicaciones, almacenamiento y el software del sistema entregado por el proveedor de servicios en la nube a través de la web en función de los requisitos del usuario como se muestra en la siguiente imagen.



De acuerdo con Buyya una nube es una tipo de sistema paralelo y distribuido que consiste en colección de computadoras interconectadas y virtualizadas que se aprovisionan dinámicamente y se presentan como uno o más recursos informáticos unificados basado en el nivel de servicio de acuerdos, negociados y respetados por el tiempo entre los consumidores y proveedores de servicios en la nube. La infraestructura aprovisionada en la nube incluye recursos escalables como red, almacenamiento y aplicación que se puede desplegar elásticamente dentro de un corto tiempo, reduciendo así los gastos generales de gestión a la nube proveedor y tiempo administrativo innecesario para los usuarios finales. La computación en la nube para los consumidores implica pago por uso en modo de demanda de recursos de TI compartidos, a los que se accede a través de Internet desde cualquier computadora de usuario final en cualquier momento en cualquier lugar a través del Globo sin limitación a los recursos requeridos o demandas de hardware. Los recursos de la nube son visibles para los usuarios, lo que significa que los usuarios pueden arrendar dinámicamente y usar los recursos de la nube sin la necesidad de saber dónde y cómo los recursos de la nube existen. Los tipos de implementación de Cloud Computing incluyen: públicos, nube privada, híbrida y de la comunidad, mientras que, los servicios consisten en:

* + 1. Infraestructura como un servicio (IaaS).
    2. Software como servicio (SaaS).
    3. Plataforma como servicio (PaaS).

## Cloud computing proyecto, aplicación y herramientas

Esta sección se usará para delinear algunas aplicaciones, proyectos y herramientas en el entorno de la computación en la nube.

### Proyectos de Cloud Computing

**CESWP** (plataforma meteorológica espacial habilitada en la nube): El objetivo del proyecto es utilizar la flexibilidad de la computación en la nube y la oportunidad de hacer espacio físico del tiempo. El objetivo es para superar el obstáculo y hacer que sea más fácil para los físicos conducir su ciencia respectiva, como desarrollar el espacio modelos climáticos, permiten la colaboración con otros científicos, emprender tareas de simulación, producir virtualización y permitir procedencia

**CERN: (The European for Nuclear Research)** está desarrollando una computación en nube a gran escala generalmente para ser utilizada en distribuir datos a científicos de todo el mundo específicamente como parte del proyecto LHC (El Gran Colisionador de Hadrones, GCH (en inglés Large Hadron Collider, LHC)).

**TCloud** este proyecto es para el desarrollo de prototipos de infraestructura avanzada de la nube computacional, capaz de entregar una nueva forma de almacenamiento informático seguro, privado y resistente que será escalable, simple y rentable

**OpenNebula** es un proyecto de código abierto que tiene lanzado con éxito una empresa avanzada, flexible y notable herramienta de gestión de computación en la nube preparada. La meta es desarrollar una computación avanzada en la nube, también escalable y adaptable paquete de herramientas de software. El objetivo es simplificar la administración de la nube, incluida la tolerancia a fallas funcionalidad para garantizar el tiempo de actividad de la nube, nuevas funcionalidades de seguridad, imágenes y plantillas, optimización de centros de datos y arquitectura multicapa mejorada.

**UCI (Unified Cloud Interface)**: Cloud Computing Interoperability Forum (CCIF) es un organismo que trabaja para el desarrollo de API estándar. Propusieron un proyecto de UCI para ser utilizado por todos los proveedores de servicios de computación en la nube para resolver problemas de interoperabilidad en la nube.

### Aplicación de Cloud Computing

**Cloudo**, es una computadora gratuita que reside en el Internet a través del navegador web. Esto puede ser utilizado para acceder música, documentos, películas, fotos y otros archivos importantes desde cualquier teléfono móvil o PC.

**Panda Cloud Antivirus**, es el primer antivirus gratuito software de computación en la nube. La aplicación es conocida por ocupar solo un poco de recursos del sistema y tiene un historial de actualizaciones regulares, además hace uso de combinación de servidores inteligentes para detección rápida potencial, se ejecuta fuera de línea protección y abarca una interfaz de usuario simple.

**Library Resource**: en vista de la rápida expansión de la computación en la nube dentro de muchas bibliotecas universitarias hay indicios de que mover lo físico a digital ayudará a mantener los recursos de la biblioteca y generará la satisfacción de los usuarios.

**El proyecto RoboEarth**, está dirigido por la institución holandesa “Eindhoven University of Technology”, un proyecto europeo para desarrollo de WWW para robots. Se compone de una mega base de datos donde los objetos, el entorno y la información de tareas pueden ser compartidos por robots. Además, en “AORO A-Star Social Robotics Laboratory Singapore”, una infraestructura de computación en la nube permite la generación de mapas tridimensionales por robots en su entorno mucho más rápido de lo que podrían ser utilizando su PC a modo normal.

### Herramientas de computación en la nube

**CloudSim y CloudAnalyst**: la herramienta es valiosa para desarrolladores, ya que se utiliza para evaluar la nube a gran escala, requisito de la aplicación con respecto a las cargas de trabajo del usuario y distribución geográfica del servidor de computación. El último fue desarrollado para ayudar a los desarrolladores a tener una idea de las formas de adoptar la distribución de aplicaciones dentro de las infraestructuras de la nube, incluidos los servicios de valor agregado, mientras que el primero fue modelado con el objetivo de estudiar los comportamientos de aplicación dentro de varias configuraciones de implementación.

**Zenoss**, es una herramienta integrada utilizada para el monitoreo de infraestructura completa de IT, independientemente de la ubicación desplegada en la nube, tanto virtual como física. Proporciona administración a implementaciones en la nube, dispositivos virtuales, almacenamiento, servidores, etc.

**Cloudera**, es un software de código abierto Hadoop, marco muy utilizado para la implementación de la computación en la nube basado en su flexibilidad con querys de datos intensivos y otras tareas principales.

Spring Roo: esta es una herramienta moderna para la aplicación desarrollo, con la fuerza combinada de Google Toolkit (GWT) que permite a los desarrolladores en entornos de producción empresarial modelar una aplicación de navegador enriquecida.

## Cluster, grid y cloud computing: comparación

Mirando un punto de vista más amplio, clúster, grid y nube, parecen compartir características en muchas formas. Esta sección arroja una imagen completa de diferenciar en varias perspectivas claramente.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Clúster** | **Grid** | **Cloud** |
| Propiedad | Propiedad única | Propiedad múltiple | Propiedad única |
| Precios del servicio | Limitado | Privado o público asignado | Utilidad /Descuento para grandes usuarios |
| Virtualización | Media | Media | Si |
| Administración de recursos | Recurso centralizado | Recurso distribuido | Ambos |
| Tamaño escalable | 100s | 1000s | 100 a 1000s |
| Estandarizado | Si | Si | No |
| Interoperabilidad | Si | Si | No del todo |
| Interconexión red/velocidad | Dedicado de alta gama con baja latencia y alta ancho de banda | Mayormente con internet de alta latencia y bajo ancho de banda | Alta gama dedicada con baja latencia y alto ancho de banda |
| Imagen de sistema único | Si | No | Si / Opcional incluido |
| Negociación de servicios | Limitado | Si, basado en SLA | Si, basado en SLA |
| Descubrimiento de membresías | Servicios de membresía | Servicios de información descentralizada / indexación centralizada | Aperturas de servicio de membresía |
| Sistema operativo | Windows / Linux | Cualquier estándar (dominado por Unix) | Un hipervisor (VM) que se ejecuta en múltiples SO |
| Controladores de aplicaciones | Negocios, centros de datos, informática empresarial | Colaborativo científico y aplicaciones de alto rendimiento | Aplicaciones Web de entrega de contenido, provisionado dinámicamente |
| Estándares / interoperabilidad | Interfaz virtual, arquitectura VIA | Algunos foros de Open-Grid | Web Services (SOAP y REST) |
| Escalable | No | A medias | Si |
| Gestión de fallas | Limitado (a menudo falla y reinicia) | Limitado (a menudo al fallar la tarea o aplicación se reinicia) | Usualmente al fallar existe replicación de contenido; migración de VM de un nodo a otro es soportado |
| Capacidad | Estable y capacidad de garantía | Varios, pero de alta capacidad | Provisionado según demanda |
| Seguridad | Tradicional basado en usuario/contraseña | Publico/privado, basado en autenticación y mapeo de un usuario a una cuenta | Cada usuario/aplicación es proveído con una máquina virtual |
| Privacidad | Nivel medio de privacidad, depende de los privilegios de usuario | Apoyo limitado para privacidad | Alta seguridad/privacidad es garantizada, apoyo para establecer la lista de control de acceso por archivo (ACL) |
| Población | Computadoras de productos básicos | Alta gama de sistemas (clúster, servidores) | Comodidad de pc, red de servidores de alta gama con almacenamiento adjunto |
| Presentación del usuario final | Presentado como un sistema dinámico y diversificado | Presentado como una imagen de sistema único | Presentado como un modelo de uso basado en autoservicios |

# Discusión en cluster, grid y cloud

Conceptualmente, sus características parecen relacionadas, aunque existen diferencias menores. Por ejemplo, todo el modelo tiene cierto nivel de tamaños escalables y virtualización, etc. Sin embargo, difieren de muchas maneras en término de gestión de recursos, estandarización, autoservicio, multi-tenancy, etc. Los recursos de computación de clúster son administrativamente ubicados y administrados en un solo dominio mientras que los recursos de grid se distribuyen geográficamente a través múltiples dominios administrativos con sus respectivos objetivos y políticas de gestión. Por el contrario, la computación en la nube la tecnología posee características principales de ambos grupos y grid computing, además de sus propias capacidades y atributos tales como soporte extensivo para virtualización, aprovisionamiento dinámico de recursos informáticos como utilidad forma de consumo, es decir, almacenamiento, red y aplicación bajo demanda a través de la interfaz web a los usuarios sin referencia a infraestructura alojada

# Conclusión

* La computación en la nube es un nuevo paradigma informático que tiene evolucionado desde el avance de la tecnología de virtualización en tiempos recientes. La popularidad actual atraída por la computación en la nube de una audiencia divergente se puede atribuir a su diseño arquitectónico, modelos de desarrollo y servicios específicamente modelados como un objetivo hacia la utilidad basada consumo y tecnología. Este documento ha sido utilizado para presentar una relación evolutiva y diferencias marginales de la tecnología de computación en la nube a otros modelos de computación como Grid computing y Clúster computing, más especialmente sus diversas aplicaciones, herramientas y proyectos relevantes para su crecimiento tecnológico.
* Los sistemas distribuidos están en todas partes, y el Internet facilita que los usuarios de todo el mundo accedan a sus servicios desde cualquier parte del mundo. Es posible construir pequeños sistemas distribuidos con computadoras y otros dispositivos pequeños conectados a una red alámbrica o inalámbrica.
* El principal motivo de los sistemas distribuidos es la compartición de recursos, como impresoras, archivos, páginas web, bases de datos, que se administran mediante servidores apropiados.

# Bibliografía

* https://research.cs.wisc.edu/htcondor/
* https://cds.cern.ch/record/46085?ln=es
* http://www.moldiscovery.com/software/grid/
* https://wiki.egi.eu/wiki/EGI-InSPIRE:Main\_Page
* http://www.openmp.org/
* https://es.wikipedia.org/wiki/Interfaz\_de\_Paso\_de\_Mensajes
* https://www.mmm.ucar.edu/weather-research-and-forecasting-model
* https://www.ecured.cu/Computaci%C3%B3n\_distribuida
* http://www.ijfcc.org/vol4/361-CS306.jsp
* https://www.ramonmillan.com/tutoriales/gridcomputing.php
* http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.463.3695&rep=rep1&type=pdf
* https://es.slideshare.net/gargishankar1981/open-grid-service-architecture-by-gargishankar-verma-rcet-bhilai
* http://blogs.solidq.com/es/big-data/que-es-mapreduce/
* https://es.wikipedia.org/wiki/Cluster\_Beowulf
* http://conceptodefinicion.de/metasistema/
* https://en.wikipedia.org/wiki/TeraGrid
* https://www.britannica.com/topic/TeraGrid
* https://ieeexplore.ieee.org/abstract/document/4738445/
* http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.465.8919&rep=rep1&type=pdf
* https://bioinfopublication.org/files/articles/1\_1\_1\_JGDC.pdf
* http://crc.aut.ac.ir/cloudcourse93/Papers/Cloud%20Computing%20Issues%20and%20Challenges%20for%20Ultimate%20Interoperability.pdf
* https://www.techopedia.com/definition/87/grid-computing
* https://computer.howstuffworks.com/grid-computing.htm
* https://www.globus.org/
* https://www.cloudera.com/
* https://www.zenoss.com/
* http://www.cloudbus.org/cloudsim/
* http://roboearth.ethz.ch/
* https://www.genbeta.com/web/cloudo-otro-sistema-operativo-basado-en-web
* https://opennebula.org/
* http://www.tcloudtech.com/
* http://ceswp.uaic.ro/
* https://es.wikipedia.org/wiki/Nimrod
* http://www.paradyn.org/
* http://www.cloudbus.org/middleware/
* https://es.wikipedia.org/wiki/Apache\_Hadoop
* http://www.revista.unam.mx/vol.4/num2/art3/cluster.htm
* https://laredinfinita.wordpress.com/2014/05/10/cluster-y-grid-colecciones-de-sistemas/