Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ciencias y Sistemas

PROPUESTA PARA LA IMPLEMENTACIÓN DE UN GRID EN EL CAMPUS CENTRAL DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

Mario Stuardo Porras Corado
Asesorado por el Ing. Moisés Fabriccio Díaz Arévalo

Guatemala, octubre de 2008

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

PROPUESTA PARA LA IMPLEMENTACIÓN DE UN GRID EN EL CAMPUS CENTRAL DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA
POR

MARIO STUARDO PORRAS CORADO

ASESORADO POR EL ING. MOISÉS FABRICCIO DÍAZ ARÉVALO

A CONFERÍRSELE EL TÍTULO DE

INGENIERO EN CIENCIAS Y SISTEMAS

GUATEMALA, OCTUBRE DE 2008

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

NÓMINA DE JUNTA DIRECTIVA

DECANO	Ing. Murphy Olympo Paiz Recinos
VOCAL I	Inga. Glenda Patricia García Soria
VOCAL II	Inga. Alba Maritza Guerrero de López
VOCAL III	Ing. Miguel Ángel Dávila Calderón
VOCAL IV	Br. Milton De León Bran
VOCAL V	Br. Isaac Sultan Mejía
SECRETARIA	Inga. Marcia Ivónne Véliz Vargas

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANO	Ing. Murphy Olympo Paiz Recinos
EXAMINADORA	Inga. Virginia Victoria Tala Ayerdi
EXAMINADOR	Ing. Edgar Estuardo Santos
EXAMINADOR	Ing. Juan Álvaro Díaz Ardavin
SECRETARIA	Inga. Marcia Ivónne Véliz Vargas

HONORABLE TRIBUNAL EXAMINADOR

Cumpliendo con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

PROPUESTA PARA LA IMPLEMENTACIÓN DE UN GRID
EN EL CAMPUS CENTRAL DE LA
UNIVERSIDAD DE SAN CARLOS DE GUATEMALA,

tema que me fuera asignado por la Dirección de la Escuela de Ciencias y Sistemas, el 30 de julio de 2007.

Mario Eduardo Porras Corado

Guatemala, Agosto 29 del 2008

Ingeniero
Carlos Alfredo Azurdia Morales
Coordinador
Comisión de Aprobación y Revisión de Trabajo de Graduación
Carrera de Ingeniería en Ciencias y Sistemas
Facultad de Ingeniería
Universidad de San Carlos de Guatemala

Estimado Ingeniero:

Por este medio hago de su conocimiento que he considerado el trabajo de tesis titulado **Propuesta para la Implementación de un Grid en el Campus Central de la Universidad de San Carlos de Guatemala**, que el estudiante Mario Stuardo Porras Corado ha desarrollado como proyecto de graduación para optar al titulo de Ingeniero en Ciencias y Sistemas. Este trabajo ha sido de mi interés, por la experiencia con que cuento en el área de investigación que el estudiante ha elegido, y a mi consideración el contenido del trabajo es satisfactorio y esta finalizado en un 100%.

Como Asesor de este trabajo me hago responsable de la calidad y del contenido completo que la tesis ha alcanzado.

Sin más por el momento y agradeciendo la oportunidad de colaboración a la educación e investigación universitaria que me da, me despido.

Atentamente:

Moisés Fabriccio Díaz Arévalo
Ingeniero en Ciencias y Sistemas
Colegiado: 3563

Universidad San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, 22 de Septiembre de 2008

Ingeniero
Marlon Antonio Pérez Turk
Director de la Escuela de Ingeniería
En Ciencias y Sistemas

Respetable Ingeniero Pérez:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación del estudiante **MARIO STUARDO PORRAS CORADO** titulado: **"PROPUESTA PARA LA IMPLEMENTACION DE UN GRID EN EL CAMPUS CENTRAL DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA"**, y a mi criterio el mismo cumple con los objetivos propuestos para su desarrollo, según el protocolo.

Al agradecer su atención a la presente, aprovecho la oportunidad para suscribirme,

Atentamente,

Ing. Carlos Alfredo Azurdia
Coordinador de Privados
y Revisión de Trabajos de Graduación

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
INGENIERIA EN CIENCIAS Y SISTEMAS
COORDINACION TESIS

ESCUELA DE CIENCIAS Y SISTEMAS

UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA

FACULTAD DE INGENIERÌA
ESCUELA DE CIENCIAS Y SISTEMAS
TEL: 24767644

El Director de la Escuela de Ingeniería en Ciencias y Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer el dictamen del asesor con el visto bueno del revisor y del Licenciado en Letras, de trabajo de graduación titulado ***"PROPUESTA PARA LA IMPLEMENTACIÓN DE UN GRID EN EL CAMPUS CENTRAL DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA"*** *presentado por el estudiante* ***MARIO STUARDO PORRAS CORADO,*** *aprueba el presente trabajo y solicita la autorización del mismo.*

"ID Y ENSEÑAD A TODOS"

Ing. Marlon Antonio Pèrez Turk
Director, Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, 08 de octubre 2008

Universidad de San Carlos
de Guatemala

Facultad de Ingeniería
Decanato

Ref. DTG.319.2008

El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería en Ciencias y Sistemas, al trabajo de graduación titulado: **PROPUESTA PARA LA IMPLEMENTACIÓN DE UN GRID EN EL CAMPUS CENTRAL DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA,** presentado por el estudiante universitario **Mario Stuardo Porras Corado,** procede a la autorización para la impresión del mismo.

IMPRÍMASE.

Ing. Murphy Olympo Paiz Recinos
DECANO

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
DECANO
FACULTAD DE INGENIERIA

Guatemala, Octubre de 2008

/gdech

AGRADECIMIENTOS A:

Quiero agradecer a todos los que, de forma directa o indirecta, han contribuido con el desarrollo de este trabajo de graduación, ya sea mediante su cariño, su apoyo, sus conocimientos, su tiempo, su paciencia, su interés o su compañía; porque cada uno ha desempeñado un papel realmente importante. Quisiera, no solo agradecer a las personas que han estado en esta etapa de mi vida, sino también a todos aquellos que a lo largo de mi vida han influido en mi carácter, en mis decisiones y en lo que ahora soy, porque de una u otra forma han colaborado con mi desarrollo personal. Desearía mencionar a todas, y a cada una de las personas, que considero mi familia, mis amigos; sin embargo la lista sería demasiado larga, ya que, gracias a Dios, durante cada una de las etapas de mi vida he convivido con personas extraordinarias. Es por esto que creo que los nombres sobran, porque creo que cada uno sabe que ha aportado algo importante para el logro de esta meta.

No puedo dejar de agradecer a Dios, por haberme dado la vida, la sabiduría y la paciencia para lograr alcanzar mis metas, en especial esta. A mis padres, Mario René y Carmen Ileana, por brindarme su amor incondicional, su confianza, su apoyo y por guiarme siempre. A mi esposa, Mariel, por todo su amor, por su paciencia y por brindarme su apoyo a lo largo de estos años. A mi hermano, Carlitos, por su amor, por su apoyo y por su amistad. A mis abuelitos, Bernabé y María, porque durante su vida me brindaron siempre mucho amor, afecto, comprensión y muchas enseñanzas. A mi familia, por brindarme siempre su cariño y su apoyo, en especial a mis padrinos, Deirdre y Mynor; a mis primos Lis, Alma, Helen, Belia; a mis sobrinos Alfredito, Andrés, José, Dianita, Faby, Ray, Scarlet, Almita, Marcia, Yessenia; a mis tíos Mima, Miguel, Berta, Aura, Luis, Julio, Lilian, Edy, Panchito; y también a Alfredo.

Agradezco a mis amigos, por los buenos momentos y por todo lo que aprendimos juntos, a mis amigos del Colegio Loyola, del Centro de Estudios Diversificado, de la Universidad de San Carlos de Guatemala (principalmente al QPTeam y a los 12 Monkeys), del grupo de jóvenes Semilla, del grupo de jóvenes La Red y de la Pequeña ECO Santa Ana.

Quiero agradecer también a las instituciones que durante estos años me brindaron los conocimientos, que me hicieron llegar a este punto de mi vida, colegio Carrousel, Colegio Loyola, Centro de Estudios Diversificados y Universidad de San Carlos de Guatemala, así como a todos sus maestros y catedráticos. Quiero agradecer de forma especial al Padre Jesús, ex director del colegio Loyola, porque sus enseñanzas y su sed de conocimiento son parte integral de este logro.

Agradezco a mi asesor, Fabriccio Díaz, por el tiempo, esfuerzo y dedicación que me brindó para la revisión y correcciones de este trabajo, así como todos sus aportes. Adicionalmente, también quiero agradecer el apoyo de Luis Quezada, quien colaboró conmigo para la selección de este tema y me proporcionó las bases para su desarrollo.

Quiero agradecer a BYTE, por la oportunidad de desarrollarme profesionalmente en el ámbito de mi carrera, y por proporcionarme el tiempo para continuar con mis estudios.

Agradezco a mi patria, por brindarme este bello suelo en el que nací, por las oportunidades que en ella he encontrado y por la oportunidad de trabajar para formar un mejor país.

Gracias a todos.

DEDICATORIA

Dedico este trabajo a Dios, a mis padres, mi esposa y mi hermano. Porque ellos son mi inspiración para seguir siempre adelante.

"La gota de agua no perfora la roca por su fuerza, sino por su persistencia"

Publio Ovidio Nasón

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES

FIGURAS

TABLAS

GLOSARIO

AFS — *Andrew File System*: es un sistema de archivos que se puede compartir a través de la red.

Agentes — Pieza de *software* o *hardware* que recibe y envía mensajes.

Ancho de banda — Se denomina ancho de banda a la cantidad de datos que se pueden transmitir en una unidad de tiempo. Esto es, la tasa de transferencia máxima permitida por el sistema, que depende del ancho de banda analógico, de la potencia de la señal, de la potencia de ruido y de la codificación de canal.

Aplicación — Generalmente se utiliza el término aplicación, para un trabajo de alto nivel dentro del *Grid*. Sin embargo, algunas veces el término trabajo es utilizado de forma equivalente. Una aplicación puede ser dividida en varios trabajos.

Aplicación de alto rendimiento — Aplicación en que el *Grid* es utilizado para administrar un gran número de tareas que están entrelazadas o que son independientes, con el objetivo de utilizar los ciclos de procesador que no están siendo utilizados (frecuentemente de una terminal ociosa).

Aplicación de colaboración	Aplicación que está concebida, principalmente, para establecer y mejorar la interacción humano - humano. Y son frecuentemente estructuradas en términos de espacio virtual compartido, de manera que están creadas para permitir que se compartan recursos computacionales tales como archivos de datos y simulaciones.
Aplicación de computación distribuida	Aplicación que utiliza el *Grid* para hacer frente a problemas que no podrían ser solucionados en un único sistema, agregando sustancialmente recursos computacionales, cuando sea necesario.
Aplicación de procesamiento de datos	Aplicación que se enfocan en sintetizar nueva información que se encuentra geográficamente distribuida en repositorios, bibliotecas digitales o bases de datos. Este proceso de síntesis requiere tanto de comunicación intensiva, como de capacidad computacional.
Aplicación por demanda	Aplicación que utiliza las capacidades del *Grid* para satisfacer los requerimientos de recursos, que serán utilizados por tiempos cortos, y que por razones de rentabilidad o conveniencia, no necesitan estar en la misma localidad.
Archivo	Agrupación de información que puede ser manipulada de forma unitaria por el sistema operativo de un ordenador. Un archivo puede ser de tipo binario o texto, estructurado o no estructurado, puede contener imágenes, documentos, programas ejecutables, etc. El termino archivo, no debe confundirse con el de base de datos, aunque frecuentemente se utilizan de forma indistinta.

Caché	Conjunto de datos duplicados de otros originales, con la propiedad de que los datos originales son costosos de acceder, normalmente en termino de tiempo de acceso, respecto a la copia en el caché. Cuando se accede por primera vez a un dato, se hace una copia en el caché; los accesos siguientes se realizan a dicha copia, haciendo que el tiempo de acceso al dato sea menor.
Capacidad de almacenamiento	Lo constituye la cantidad de almacenamiento, incluso si es temporal, para la utilización del *Grid*, el cual es provisto por una máquina, incluye memoria de procesador o almacenamiento secundario, utilizando discos duros u otro dispositivo de almacenamiento permanente.
Capacidad de procesamiento	Lo constituyen los ciclos de procesamiento que provee el procesador, de las máquinas del *Grid*. El procesador puede variar en velocidad, arquitectura, plataforma de software y otros factores asociados, como memoria, almacenamiento o conectividad.
CIM	Arquitectura para la administración de sistemas de información, que intenta unificar la administración de entornos empresariales. CIM es un estándar de la DMTF, para dar a conocer datos acerca de sistemas, aplicaciones, redes y dispositivos; además permite el intercambio de información, de los elementos administrados, entre las partes interesadas, permitiendo que las aplicaciones de administración tengan acceso a los datos, control de dispositivos o a los sistemas sin importar la plataforma, facilitando así la interoperabilidad.

DFS	*Distributed File System*: es un sistema de archivos que se puede compartir a través de la red.
DMTF	*Distributed Management Task Force*: es una organización dedicada al desarrollo y mantenimiento de estándares, relacionados con la administración de sistemas y entornos de tecnológicos de empresas e Internet.
Entidad proveedora	Persona u organización, que provee un agente apropiado para implementar un servicio en particular.
Entidad solicitante	Persona u organización, que desea hacer uso de una entidad que provea algún servicio.
Federar	Conjunto de reglas que sientan las bases, para el enlace de recursos a gran escala, sobre sistemas que administran recursos distribuidos.
Globus Toolkit	Conjunto de herramientas de software para la construcción de sistemas utilizando la tecnología *Grid*, el cual implementa librerías para el monitoreo, descubrimiento y administración de recursos, así como seguridad y administración de archivos. Además, es una herramienta que está basada completamente en la filosofía de *Grid Computing*, provee los servicios básicos para la construcción de la infraestructura y la puesta en marcha de las aplicaciones; descansa sobre la base del código abierto, protocolos estándares y proporciona interfaces de programación.
GPFS	*General Parallel File System*: es un sistema de archivos que se puede compartir a través de la red.

GRAM	*Grid Resource Allocation Manager*: es un servicio que permite realizar la asignación de recursos, monitoreo y administración de trabajos.
Grid	Es un tipo de sistema distribuido y paralelo que permite el intercambio, la selección y la combinación de recursos autónomos, geográficamente distribuidos de forma dinámica dependiendo de su disponibilidad, capacidad, desempeño, costo y requerimientos de los usuarios.
Grid Computing	Es la utilización de estándares y protocolos abiertos, para permitir la virtualización de recursos computacionales distribuidos, con el objetivo de crear la imagen de un único sistema, a través de entornos informáticos heterogéneos y dispersos geográficamente.
Grid Service	*Web Service* que provee un conjunto de interfaces bien definidas, así como convenciones específicas que definen su comportamiento.
Inspeccionar	Proceso por el cual una máquina que se encuentra ociosa informa su estado al nodo administrador.
Interoperabilidad	En un entorno de red, la interoperabilidad significa protocolos comunes.

MDA — *Model Driven Architecture*: es una especificación basada en UML. Proporciona las herramientas para la creación de especificaciones y desarrollo de aplicaciones, separándolas de la tecnología subyacente de la plataforma utilizada. Permite diseñar interfaces y relaciones entre las aplicaciones, independientemente de la plataforma o el lenguaje de programación.

MDS-Archive — Servicio que permite almacenar la información de los recursos en una base de datos persistente, la cual puede ser consultada posteriormente por los usuarios.

MDS-Index — Servicio que permite consultas acerca de las características de los recursos.

MDS-Trigger — Servicio que permite realizar ciertas tareas definidas por los usuarios, siempre que la información recolectada cumpla con los criterios definidos por el usuario.

NFS — *Network File System*: es un sistema de archivos que se puede compartir a través de la red.

OGSA — Arquitectura computacional para la interacción de recursos distribuidos, que está orientada a los servicios, asegurando la interoperabilidad sobre sistemas heterogéneos, para que, recursos de diferentes tipos puedan comunicarse y compartir información.

OGSA-DAI	Servicio que proporciona herramientas para tareas relacionadas con la localización, transferencia y administración de datos distribuidos, tanto para datos estructurados como para datos parcialmente estructurados.
Orientación a los servicios	El enfoque orientado a los servicios, consiste en adoptar una representación generalizada de los recursos informáticos, donde todos pueden ser tratados como servicios.
Planificar	Proceso por el cual se busca automáticamente la máquina apropiada, donde pueda correr un trabajo que esté esperando para ser ejecutado.
Protocolo	Definición de los mecanismos básicos para que los usuarios de las organizaciones virtuales y los recursos negocien, establezcan, administren y exploten las relaciones para compartir.
QoS	*Quality of Service:* en Internet y otro tipo de redes, es la idea que las tasas de transmisión, las tasas de errores y otras características, puedan ser medidas, mejoradas y, hasta cierto punto, garantizarlas por adelantado.
Recursos	Colección de máquinas, algunas veces nombradas como nodos, recursos, miembros, donadores, clientes, hosts, motores y muchos otros términos. Todos ellos contribuyen, con alguna combinación de recursos, para que el *Grid* sea uno.
Red de Servicios Integrados	Infraestructura de red, que proporciona conectividad entre los edificios, de las diferentes unidades académicas y administrativas del Campus Universitario de la Universidad de San Carlos de Guatemala, a través de una conexión de fibra óptica.

Reservar	Proceso por el cual, se calendarizan de forma anticipada los recursos para un conjunto de trabajos designados.
RTF	*Reliable File Transfer*: es un servicio que permite preparar los archivos de entrada y salida, así como su transferencia confiable.
Servicio	Es el conjunto abstracto de funcionalidades que el agente provee.
SOAP	*Simple Object Access Protocol:* es un protocolo basado en XML para el intercambio de información entre computadoras. Puede ser utilizado por una variedad de sistemas de mensajes y entregado por medio de varios protocolos de transporte.
Soft-state Protocol	Protocolo asociado a la política de desechar el estado de un proceso, establecido en una ubicación remota, a menos que este se actualice de manera periódica.
UDDI	*Universal Description, Discovering and Integration*: es una infraestructura que define un modelo de datos, para el registro y descubrimiento de información de negocios, incluyendo los *web services* que los negocios publican. UDDI es producto de un consorcio independiente, para el desarrollo de un estándar de Internet que ayude a la descripción, registro y descubrimiento de *web services*.

UML — *Unified Modeling Language*: es una notación estándar utilizada para el diseño, modelado de aplicaciones y modelado de sistemas de manera visual. El UML es un lenguaje y no una metodología, y como tal, es independiente de los lenguajes de programación y de las plataformas. Puede ser utilizado en conjunto con cualquier lenguaje de programación para crear el diseño de un sistema e ilustrar su estructura.

Virtualización — La virtualización permite acceso coherente a los recursos, entre plataformas heterogéneas; el mapeo de múltiples instancias, de un recurso lógico, sobre el mismo recurso físico y facilita la administración de los recursos de una organización virtual, basándose en la integración de recursos de bajo nivel.

W3C — *World Wide Web Consortium*: es la principal organización de estándares para protocolos y especificaciones web.

WBEM — *Web Based Enterprise Management*: es una iniciativa que busca diseñar estándares para la administración de entornos computacionales. La meta es proveer estándares, para la industria, que permitan la creación de herramientas, basadas en estándares, para la integración en el uso de las tecnologías web.

Web Service — Sistema de software diseñado para soportar la interacción de operaciones máquina a máquina sobre una red. Esta tiene una interfaz descrita en un formato procesable por la máquina (específicamente WSDL). Otros sistemas interactúan con los *web services* de la forma descrita por esta especificación utilizando mensajes SOAP, típicamente transmitidos a través de HTTP con XML en conjunto con otros estándares web relacionados.

WSD	Especificación de la interfaz *web service*, la cual debe ser procesada por una máquina y debe estar escrita en WSDL. En este se definen los formatos de los mensajes, tipos de datos, protocolos de transporte y la serialización de de los formatos de transporte, que serán utilizados por los agentes (solicitantes y proveedores).
WSDL	*Web Service Definition Language*: representa la capa de descripción de servicios. El WSDL es una gramática XML que permite especificar una interface pública para la utilización del *web service.*
WS-I Basic Profile	Describe la manera en que las cuatro tecnologías clave, de la especificación *web services*, pueden ser implementadas para alcanzar un nivel de coherente de interoperabilidad. Las cuatro especificaciones claves son: XML *Schema*, SOAP, WSDL y UDDI.
WSS - Web Service Security	Protocolo de comunicación que proporciona los medios para aplicar la seguridad de los *web services*.
XML	*Extended Markup Language*: es una familia de tecnologías, que utiliza un lenguaje de etiquetas, estandarizado y estructurado, para definir la representación de datos y documentos. Específicamente el XML, es un lenguaje para la creación y utilización de información estructurada, definido con una semántica comprensible tanto para humanos como para computadoras.

XML *Schema* Documento XML que define los elementos, tipos de datos, contenido y estructura para un documento XML asociado. Fue desarrollado por el W3C, como una alternativa para definir la estructura de un documento XML, utilizando sintaxis XML.

XSLT *Extensible Stylesheet Language Transformation*: es un lenguaje basado en XML utilizado para la transformación de documentos XML. El documento original no cambia, en vez de ello, se crea uno nuevo basado en el contenido del que ya existe. El nuevo documento puede ser generado por el proceso en XML estándar o en algún otro formato, como puede ser HTML o texto plano.

RESUMEN

El contenido de este documento utiliza un conjunto de conceptos, ideas y resultados recolectados mediante la investigación de documentos, para presentar al lector, de una manera comprensiva, los conceptos en que se basa la integración de sistemas distribuidos y heterogéneos, utilizando el *Grid*. Para, posteriormente, presentar una propuesta para la implementación de un *Grid*, como herramienta, para la integración de los recursos computacionales de la Universidad de San Carlos de Guatemala, de manera que estos puedan ser utilizados en el desarrollo de tareas científicas, que requieran alto desempeño para el procesamiento de datos; así como en tareas de otras áreas de la investigación, que requieran almacenamiento masivo de información.

El primer capítulo incluye una descripción general del *Grid*, características que lo identifican, componentes que lo integran, tipos de aplicaciones en los que actualmente es utilizado y la descripción de su arquitectura. Como complemento se presenta un segundo capítulo sobre la tecnología *Web Service*, donde presenta una descripción general, descripción de su funcionamiento, componentes, actores y la interacción entre ambos; y una breve descripción de los principales estándares involucrados. El tercer capítulo, describe la fisiología del *Grid*, donde se describe el modelo OGSA, como arquitectura para la interacción de recursos distribuidos y heterogéneos.

La propuesta de implementación, describe la actual infraestructura de red de la Universidad, el diseño que se sugiere para los diferentes niveles de abstracción del *Grid*, una descripción de los diferentes elementos involucrados, una descripción de los beneficios esperados y finalmente un plan de implementación con el detalle de las actividades a realizar y el recurso humano involucrado.

OBJETIVOS

Generales:

1. Proporcionar base teórica referente al *Grid*, como marco de trabajo para la integración de sistemas distribuidos y heterogéneos.
2. Proporcionar una propuesta coherente, que permita la implementación de un *Grid* en el Campus Central de la Universidad de San Carlos de Guatemala.

Específicos:

1. Proporcionar una descripción del propósito, estructura interna y arquitectura de los sistemas *Grid*.
2. Proporcionar una descripción general acerca de los *Web Services* y de los principales estándares involucrados en la integración de sistemas distribuidos, relacionados con la construcción de un *Grid*.
3. Identificar los conceptos y tecnologías, requeridos para soportar la operación entre recursos distribuidos y heterogéneos, integrando las tecnologías *Grid* y *Web Services*.
4. Desarrollar una propuesta para la implementación de un *Grid*, como una herramienta para integrar los recursos computacionales de la Universidad, tomando como base la actual infraestructura física y de redes.
5. Definir los diferentes niveles de abstracción que deben ser implementados, así como los componentes que integrarán el *Grid*.

INTRODUCCIÓN

Durante décadas, la utopía de crear una infraestructura capaz entrelazar recursos heterogéneos a través de redes de alta velocidad, y romper las barreras técnicas que separan a las diferentes plataformas, constituyo uno de los mayores retos para investigadores y desarrolladores. Pero, no es sino hasta ahora, que los conceptos y las tecnologías han evolucionado, que puede verse los resultados de estos esfuerzos.

La creciente popularidad del Internet junto con la disponibilidad de poderosos computadores y redes de alta velocidad, así como componentes de bajo costo, está ayudando a cambiar la forma de hacer computación. Estas nuevas tecnologías están permitiendo la unión de una variedad de recursos geográficamente distribuidos, como supercomputadoras paralelas, sistemas de almacenamiento, bases de datos y dispositivos especiales; esta integración de recursos es popularmente conocida como *Grid* [1]

El Grid Computing, en su estado más simple, es un sistema distribuido llevado hacia su siguiente nivel en la evolución. La meta es crear la ilusión de una computadora virtual simple, pero a su vez grande y poderosa, que tenga la capacidad de auto gestionarse, con un conjunto de sistemas heterogéneos conectados y compartiendo cualquier tipo de recursos. [2]

La realización de esta meta requiere de la desintegración de numerosas barreras que normalmente separan a los diferentes sistemas informáticos dentro y a través de las organizaciones, de modo que computadoras, aplicaciones, datos y otros recursos, puedan ser utilizados cuando sea requerido, sin preocuparse por su localización.

La clave para esta realización es la estandarización, de modo que los diversos componentes puedan ser descubiertos, asignados, monitoreados, etc., y en general que puedan ser administrados como un sistema virtual. La estandarización es crítica si se desea crear sistemas y recursos que operen entre sí, que puedan ser portados a diversas plataformas y que sean reutilizables; esto puede también contribuir al desarrollo de sistemas seguros, robustos y escalables. [3]

Este documento, proporciona una base teórica que permita al lector, introducirse en el complejo, pero interesante, mundo de la tecnología *Grid*. Además, se presenta la visión del modelo OGSA (*Open Grid Service Architecture*), que define los elementos necesarios para la integración de recursos distribuidos y heterogéneos, siendo el resultado de esto, un sistema de servicios distribuido, basado en estándares, que permite la creación de redes sofisticadas, requeridas en los entornos modernos de negocios e ínter organizacionales.

Este documento presenta el marco de trabajo teórico, basado en los documentos presentados por las diferentes organizaciones que se dedican al desarrollo de tecnologías *Grid*. Por consiguiente cualquier resultado presentado, será el reflejo de los diferentes estudios realizados por estas organizaciones.

La descripción del propósito, estructura interna y arquitectura de los sistemas *Grid*, presenta los conceptos previos al desarrollo del tema principal, y podría ser objeto de una investigación completa, en este capítulo se abarcarán temas puntuales, permitiendo al lector comprender, de forma general, la tecnología *Grid* y los componentes involucrados en su construcción.

La descripción de Web Services y los principales estándares involucrados en la integración de sistemas distribuidos, serán cubiertos de forma general, con el objetivo de proporcionar una base teórica, pero sin la finalidad de cubrir a detalle cada uno de los aspectos involucrados en el tema.

La descripción de principios y fundamentos para la integración de recursos heterogéneos, está orientada a la utilización del modelo OGSA (*Open Grid Services Architecture*). Se procederá a la definición del modelo abstracto de los requerimientos y posteriormente a la definición de las capacidades especificas para satisfacerlos, presentando los aspectos teóricos necesarios, sin entrar en los detalles técnicos de implementación.

Finalmente, se realizará la para la implementación del *Grid* en el Campus Central de la Universidad de San Carlos de Guatemala, describiendo la infraestructura de red de la Universidad, el diseño que se sugiere para los diferentes niveles de abstracción del *Grid*, una descripción de los diferentes elementos involucrados, una descripción de los beneficios esperados y finalmente un plan de implementación con el detalle de las actividades a realizar y el recurso humano involucrado. Dentro de los diferentes niveles de abstracción se describe cada uno de los entes involucrados en la implementación, así como los recursos y elementos que lo conformaran.

Es necesario remarcar que las organizaciones no deben considerar al *Grid* como una solución por sí misma, sino como una forma de mejorar la efectividad de la infraestructura que poseen o como una forma de mejorar el proceso del negocio a través de la transformación. La Universidad actualmente cuenta con la infraestructura de red necesaria para lograr la implementación de un proyecto de esta naturaleza, y convertirse en una entidad académica con una herramienta para la investigación científica, mediante la optimización de los recursos computacionales.

1. *GRID COMPUTING*

Los dos primeros capítulos pretenden exponer de manera breve, pero concisa, los conceptos que servirán como marco de referencia al resto del documento. Se provee los fundamentos teóricos acerca de la tecnología *Grid* y la tecnología *Web Service*s, con lo que se pretende aclarar el propósito, estructura interna y arquitectura de los sistemas *Grid*, así como describir los aspectos generales de los *Web Services* y de los estándares relacionados con esta tecnología.

1.1. *Grid*

Durante décadas, la utopía de crear una infraestructura capaz entrelazar recursos heterogéneos a través de redes de alta velocidad, y romper las barreras técnicas que separan a las diferentes plataformas, constituyo uno de los mayores retos para investigadores y desarrolladores. Pero, no es sino hasta ahora, que los conceptos y las tecnologías han evolucionado, que puede verse los resultados de estos esfuerzos.

La creciente popularidad del Internet junto con la disponibilidad de poderosos computadores y redes de alta velocidad, así como componentes de bajo costo, está ayudando a cambiar la forma de hacer computación. Estas nuevas tecnologías están permitiendo la unión de una variedad de recursos geográficamente distribuidos, como supercomputadoras paralelas, sistemas de almacenamiento, bases de datos y dispositivos especiales; esta integración de recursos es popularmente conocida como *Grid*.[4]

1.2. Definición de *Grid Computing*

El *Grid Computing*, puede significar diferentes cosas para diferentes individuos. La visión global, es frecuentemente presentada como una analogía de las redes eléctricas, donde los usuarios obtienen acceso a la electricidad a través del enchufe, sin prestar importancia o considerar desde donde la electricidad está siendo generada. En esta visión del *Grid*, el potencial computacional está integrado, y los usuarios individuales obtienen acceso a diferentes recursos (procesadores, dispositivos de almacenamiento, datos, aplicaciones y otros) conociendo poco o nada acerca de donde los recursos están localizados, o bajo que tecnología (hardware, sistema operativo y otros detalles).[5]

Aunque esta visión puede hacer volar nuestra imaginación, es necesario establecer una definición clara, si se quiere que el *Grid* obtenga la credibilidad y enfoque que este necesita para crecer y evolucionar.

Hacia 1998, Ian Foster y Karl Kasselman, en su libro "*The Grid: Blueprint for a New Computing Infrastructure*", sientan las bases de teóricas de esta nueva tecnología. Y describen al *Grid* como:

"Un *Grid* computacional, es una infraestructura de hardware y software que provee acceso confiable, consistente, integrado y económico a sofisticadas capacidades informáticas."[6]

Posteriormente en el 2000, en un artículo titulado "*The Anatomy of the Grid*", escrito en colaboración con Steve Tucke, se perfecciona la definición, agregando aspectos políticos y sociales, indicando que el *Grid* se preocupa por "regular el uso de los recursos compartidos y la solución de problemas en organizaciones virtuales dinámicas y multi - institucionales." El concepto clave es la habilidad de negociar los arreglos de recursos compartidos, a través de un conjunto de participantes (proveedores y consumidores), y entonces utilizar el conjunto resultante para algún propósito. Haciendo notar esto: [7]

"El intercambio de archivos compartidos no es la parte que nos preocupa, sino el acceso directo a computadoras, software, datos y otros recursos, que es requerida para la resolución de un rango de problemas que requieren colaboración y estrategias de asignación de recursos, que emergen en la industria, la ciencia y la ingeniería. Para compartir, es necesario, un alto control de los recursos de proveedores y consumidores, donde se defina claramente que esta compartido, quien tiene permitido compartir y las condiciones bajo las cuales esto sucede. Un conjunto de individuos y/o instituciones con reglas para compartir, será llamada organización virtual."[8]

Por lo tanto, el *Grid* puede verse como un camino en donde es necesario integrar diversas tecnologías y soluciones para llegar a la meta final. La clave está en la subyacente infraestructura tecnológica de computación distribuida, que provee el soporte, para que se puedan compartir aplicaciones y recursos entre organizaciones (virtualización entre tecnologías, plataformas y organizaciones). Este tipo de virtualización es solamente adquirible a través del uso de estándares abiertos. Los estándares abiertos, permiten estar seguros, que las aplicaciones podrán de manera transparente aprovechar los recursos que se encuentran disponibles para ellos. Un ambiente que posee la habilidad de compartir y proveer acceso transparente a los recursos a través de un entorno distribuido y heterogéneo, también requiere tecnologías y estándares en áreas de planificación, seguridad, cuentas (usuarios de acceso), administración de sistemas y otros.[9]

Para concretar el tema, es necesario integrar los conceptos e ideas que se han presentado hasta ahora, y seleccionar una definición estándar. De tal cuenta, se seleccionó la siguiente definición:

"El *Grid Computing* utiliza estándares y protocolos abiertos, para permitir la virtualización de recursos computacionales distribuidos, con el objetivo de crear la imagen de un único sistema, a través de entornos informáticos heterogéneos y dispersos geográficamente."[10]

1.3. Características para identificar un *Grid*

Ian Foster sugiere que la esencia de las definiciones antes descritas puede ser capturada en una simple lista de tres puntos, de acuerdo a lo que un sistema *Grid* es:

1. "Recursos coordinados que no son sujetos de un control centralizado. (Un *Grid* integra, coordina recursos y usuarios que se encuentran en diferentes dominios de control, por ejemplo, el usuario de un computador de escritorio vs. la computadora central; diferentes unidades administrativas en una misma compañía; o en diferentes compañías; y dirigidos por aspectos de seguridad, políticas, pagos, membresías y otros que surjan en este escenario. De otra manera se estaría tratando con un sistema administrado de forma local)."[11]

2. "Usando protocolos e interfaces de propósito general que sean estándares y abiertos. (Un *Grid* está construido sobre protocolos multi - propósitos e interfaces, donde el direccionamiento es un aspecto fundamental para la autenticación, autorización, búsqueda de recursos y acceso a recursos. Como se discutió anteriormente, es importante que estos protocolos e interfaces sean estándares y abiertos. De otra manera, se estaría tratando con una aplicación para un sistema específico)."[12]

3. "Para entregar un QoS[*] que no sea trivial. (Un *Grid* permite que los recursos que los constituyen puedan ser utilizados de una forma coordinada, para entregar varios QoS, por ejemplo el tiempo de respuesta, rendimiento, disponibilidad y seguridad, y/o alojamiento de múltiples tipos recursos para satisfacer las complejas demandas de los usuarios, así que la utilidad combinada del sistema sea significativamente más grande que la suma de sus partes)."[13]

[*] QoS (*Quality of Service*). En Internet y otro tipo de redes, es la idea que las tasas de transmisión, las tasas de errores y otras características, puedan ser medidas, mejoradas y, hasta cierto punto, garantizarlas por adelantado.

1.4. Componentes

1.4.1. Recursos

Un *Grid* es una colección de máquinas, algunas veces nombradas como nodos, recursos, miembros, donadores, clientes, hosts, motores y muchos otros términos. Todos ellos contribuyen, con alguna combinación de recursos, para que el *Grid* sea uno. Algunos recursos pueden ser utilizados por todos los usuarios del *Grid*, mientras que otros poseen restricciones para su utilización. A continuación se describirán brevemente algunos de estos recursos, y su posible utilización dentro del *Grid*. [14]

1.4.1.1. Capacidad de procesamiento

El recurso más común, lo constituyen los ciclos que provee el procesador, de las máquinas del *Grid*. El procesador puede variar en velocidad, arquitectura, plataforma de software y otros factores asociados, como memoria, almacenamiento o conectividad. Existen tres formas primarias para explotarlo como recurso dentro del *Grid*.

- La primera, y la más simple, es utilizar y correr una aplicación existente en una máquina disponible dentro del *Grid* en vez de hacerlo localmente.

- La segunda, es usar una aplicación designada para dividir el trabajo, de tal manera que, al separar las partes, estas puedan ser ejecutadas en paralelo en diferentes procesadores.

- La tercera es correr una aplicación, que necesita ser ejecutada muchas veces, en diferentes máquinas en el *Grid*.

Acá entra en juego la escalabilidad, que representa la medida de eficiencia, en que múltiples procesadores en un *Grid* pueden ser utilizados. Si el doble de procesadores puede hacer que una aplicación se complete en la mitad del tiempo, entonces se dice que es perfectamente escalable. Sin embargo, existen límites para la escalabilidad, cuando las aplicaciones pueden solamente dividirse en un número limitado de partes que puedan correr, o si las partes experimentan algún tipo de interdependencia.[15]

1.4.1.2. Almacenamiento

El segundo recurso, más comúnmente utilizado, es el almacenamiento de datos. Un *Grid* que provee una visión integrada de los datos almacenados, es llamado normalmente un *Data Grid*. Cada máquina usualmente provee una cantidad de almacenamiento para la utilización del *Grid*, incluso si es temporal. El almacenamiento puede ser: memoria de procesador o almacenamiento secundario, utilizando discos duros u otro dispositivo de almacenamiento permanente. La memoria del procesador es frecuentemente más rápida, pero es volátil. Esta puede ser mejor utilizada para *caché*[†] de datos o para almacenar temporalmente las aplicaciones que estén ejecutándose.

El almacenamiento secundario en un *Grid*, puede ser utilizado de maneras interesantes para compartir datos, incrementar la capacidad, desempeño y fiabilidad. Muchos sistemas, utilizan aplicaciones de montaje de red para sus sistemas de archivos[‡], como *Andrew File System* (AFS), *Network File System* (NFS), *Distributed File System* (DFS) o *General Parallel File System* (GPFS). Estos ofrecen una variedad de características y grados de fiabilidad, desempeño y seguridad.

[†] Caché. Es un conjunto de datos duplicados de otros originales, con la propiedad de que los datos originales son costosos de acceder, normalmente en termino de tiempo de acceso, respecto a la copia en el caché. Cuando se accede por primera vez a un dato, se hace una copia en el caché; los accesos siguientes se realizan a dicha copia, haciendo que el tiempo de acceso al dato sea menor.

[‡] Sistema de archivos. Consta de tipos de datos abstractos, que son necesarios para el almacenamiento, organización jerárquica, manipulación, navegación, acceso y consulta de datos, , especialmente en lo referente a su integración en el sistema operativo del disco

La capacidad puede ser incrementada con el uso de almacenamiento en múltiples máquinas como un sistema de archivos unificado. Cada base de datos o cada archivo[§] pueden estar distribuidos en varios dispositivos y máquinas, eliminando restricciones de tamaño, frecuentemente impuestas por el sistema de archivos del sistema operativo. Un sistema de archivos unificado puede, también, proveer un nombre común para el espacio de almacenamiento. Facilitando, el poder hacer referencia a datos almacenados en el *Grid*, sin conocer su ubicación exacta. En forma similar, software especial de base de datos puede federar[**] un conjunto de bases de datos y archivos para formar una base de datos más grande y completa, que sea accesible utilizando consultas.

Sistemas de archivos más avanzados en un *Grid* pueden automáticamente duplicar un conjunto de datos, para proveer redundancia e incrementar la fiabilidad y el desempeño. Un planificador inteligente de *Grid*, puede ayudar a seleccionar los dispositivos de almacenamiento apropiados, para grabar los datos, basándose en los patrones de utilización. Entonces los trabajos pueden ser planificados para utilizar los datos más cercanos, preferiblemente en máquinas directamente conectadas a los dispositivos de almacenamiento que los requieren.[16]

1.4.1.3. Comunicaciones

Otro importante recurso del *Grid*, lo constituye la capacidad de comunicación de datos. Esto incluye comunicación dentro y fuera del *Grid*. Las comunicaciones dentro del *Grid*, son importantes para el envío de trabajos y sus respectivos datos. Algunos trabajos requieren una gran cantidad de datos para ser procesados, y estos no siempre pueden residir en la máquina donde se está ejecutando.

[§] Se entiende por archivo, a una agrupación de información que puede ser manipulada de forma unitaria por el sistema operativo de un ordenador. Un archivo puede ser de tipo binario o texto, estructurado o no estructurado, puede contener imágenes, documentos, programas ejecutables, etc. El termino archivo, no debe confundirse con el de base de datos, aunque frecuentemente se utilizan de forma indistinta.

[**] Conjunto de reglas que sientan las bases, para el enlace de recursos a gran escala. Sobre sistemas que administran recursos distribuidos.

El ancho de banda disponible para las comunicaciones puede, frecuentemente, ser un recurso crítico que limite la utilización del *Grid*.

Dentro de los sistemas que administran el *Grid*, así como en otros sistemas, las formas de comunicación redundante son muchas veces necesarias para manejar potenciales fallas en la red o excesivo tráfico de datos. Ya que en algunos casos, redes de alta velocidad deben encontrarse para poder cumplir con la demanda de transferir grandes cantidades de datos.[17]

1.4.1.4. Software y licencias

Dentro del *Grid*, se puede tener software instalado, que sería muy caro instalar en cada máquina. De esta forma, los trabajos que requieran de ese software, serán enviados a máquinas particulares donde ese software este instalado. Cuando los costos de licencia son significantes, este aprovechamiento puede significar ahorro para una organización.

Algunas licencias permiten que el software este instalado en todas las máquinas del *Grid*, pero limita el número de instalaciones que pueden utilizarla simultáneamente. La administración de licencias permite monitorear cuantas copias de software están siendo utilizadas y prevenir que más del número permitido este en uso. El planificador de trabajos del *Grid* puede ser configurado para colocar las licencias dentro de una cuenta, desde donde opcionalmente se puede manejar las prioridades y políticas.[18]

1.4.1.5. Equipo especial, capacidad, arquitectura y políticas

Las plataformas en el *Grid* frecuentemente tienen diferentes arquitecturas, sistemas operativos, dispositivos, capacidades y equipo. Cada uno de estos ítems, representa un tipo diferente de recurso, que el *Grid* puede utilizar a su criterio para asignar trabajos a las máquinas.

Mientras algún software puede estar disponible para varias arquitecturas, algunos otros están designados para correr únicamente en un hardware en particular y con un sistema operativo. Estos atributos deben ser considerados cuando se asignan trabajos a un recurso en el *Grid*.

En algunos casos, el administrador del *Grid* puede crear un nuevo tipo de recurso que sea utilizado por el planificador para asignar trabajo de acuerdo a políticas y otras restricciones. Por supuesto el administrador necesita imponer una clasificación de cada tipo de trabajo para certificar que pueda utilizar cierto tipo de recurso.[19]

1.4.2. Trabajos y aplicaciones

Así como varios tipos de recursos pueden ser compartidos y utilizados, ellos son usualmente accedidos a través de la ejecución de un trabajo o una aplicación. Generalmente se utiliza el término aplicación, para un trabajo de alto nivel dentro del *Grid*. Sin embargo, algunas veces el término trabajo es utilizado de forma equivalente. Una aplicación puede ser dividida en varios trabajos. Estos a su vez, pueden ser divididos en trabajos más pequeños. En la industria se utilizan otros términos como transacción, unidad de trabajo, sumisión, que significan lo mismo que trabajo.

Un trabajo es un programa que es ejecutado en un punto apropiado del *Grid*. Este puede calcular algo, ejecutar uno o más comandos del sistema, mover o recolectar datos, u operar maquinaria. Una aplicación *Grid,* que está organizada como una colección de trabajos, usualmente es designada para que estos trabajos se ejecuten en paralelo en diferentes máquinas en el *Grid*.

Los trabajos pueden tener dependencias específicas que impidan que se ejecuten en paralelo en algunos casos. Algunos pueden requerir que alguna salida sea producida por otros trabajos y no pueda ejecutarse, hasta que estos prerrequisitos se hayan completado.

Los trabajos pueden desprender nuevos trabajos, dependiendo de los datos a procesar. Este esquema de trabajo puede crear una jerarquía de trabajos y sub trabajos. Finalmente, el resultado de todos los trabajos debe ser recolectado y ensamblado apropiadamente para producir una última salida resultante de la aplicación.[20]

1.4.3. Planificar, reservar e inspeccionar

El *Grid* es responsable de enviar un trabajo a una máquina para que este sea ejecutado. En un *Grid* simple, el usuario puede seleccionar la máquina donde correrá su trabajo, y entonces ejecutar un comando en el *Grid* para que se envíe el trabajo a la máquina seleccionada. En *grids* más avanzados, se debe incluir un planificador de trabajos, que de alguna manera, automáticamente busque la máquina apropiada, donde pueda correr un trabajo que esté esperando para ser ejecutado. El termino planificación, no debe ser confundido con reservación de recursos.

Al inspeccionar un sistema *Grid*, cualquier máquina que llegue a estar ociosa informa su estado al nodo administrador. El nodo administrador, puede asignar esta máquina ociosa, al siguiente trabajo, para el cual sus requerimientos sean satisfechos por los recursos de la máquina. La inspección, usualmente es implementada, de tal forma, que no obstruya el uso normal de la máquina. Si la máquina llegara a ocuparse con un trabajo que no es del *Grid*, el trabajo del *Grid* será suspendido o retrasado. Esta situación crea tiempos algo impredecibles para lograr completar algunos trabajos en el *Grid*, aunque esto no causa una ruptura para las máquinas que donan recursos.

Las aplicaciones *Grid* que corren en modo de inspeccionar frecuentemente se marcan a sí mismas con un nivel de prioridad bajo en el sistema operativo. De esta forma, solamente corren cuando no existe otro proceso que esté pendiente. Debido al desempeño de los modernos procesadores y a los algoritmos de planificación de los sistemas operativos, puede correr tan brevemente como unos pocos milisegundos, incluso entre cada tecleo del usuario.

Para crear un comportamiento más predecible, las máquinas del *Grid* deberían estar dedicadas a esta tarea, evitando así que los trabajos sean retrasados. Esto permite a los planificadores calcular el tiempo aproximado para que los trabajos sean completados, cuando están corriendo con características conocidas.

Como un siguiente paso, los recursos pueden ser reservados adelantadamente para un conjunto de trabajos designados. Tales reservaciones, funcionan como un sistema de calendario, utilizado para reservar las salas de conferencia para reuniones. Esto se hace para encontrar fechas tope y garantizar la QoS. Cuando las políticas lo permitan, la reservación de recursos adelantada podrá ser también examinada, para correr trabajos con baja prioridad, cuando ellos no estén ocupados durante el periodo de reservación. Así, varias combinaciones de planificación, reservación e inspección podrán ser utilizadas, para aprovechar de forma más completa el *Grid*.

La planificación y la reservación es bastante acertada cuando solamente un tipo de recurso, usualmente el CPU, está implicado. Sin embargo, adicionalmente el *Grid* debe poder considerar más recursos para la planificación y el proceso de reservación. Por ejemplo, sería deseable asignar trabajos a ejecutar a máquinas que se encuentren cercanas a los datos que los trabajos requieren. Esto reduciría el tráfico de la red y posiblemente reduciría los límites de escalabilidad. La planificación óptima, considera múltiples recursos, y es un problema matemático bastante difícil. Por eso, los planificadores utilizan la heurística. La heurística, agrupa un conjunto de reglas que están diseñadas para mejorar la probabilidad de encontrar la mejor combinación de planificación y reservación, que permita optimizar el rendimiento o cualquier otra métrica.[21]

1.4.4. Componentes de software

Existen muchos aspectos en el *Grid* que típicamente pueden ser controlados a través del software. Pero inicialmente muchos de estos aspectos son manejados de forma manual, hasta que la disponibilidad de software más avanzado permita administrarlos. Sin embargo, se presentan algunos aspectos del software que debe ser considerado cuando se diseña y desarrolla en un ambiente *Grid*.[22]

1.4.4.1. Componentes de administración

Cualquier sistema *Grid* posee algún componente de administración. Primeramente, existe un componente que permite seguir el rastro de los recursos disponibles en el *Grid* y de los usuarios que son miembros. Esta información es usada para decidir que trabajos pueden asignársele.

Segundo, existen componentes de medición que determinará, la capacidad de los nodos y su actual nivel de utilización en cualquier momento. Esta información es utilizada para planificar los trabajos dentro del *Grid*, así como para determinar el estado del *Grid* y alertar al personal acerca de problemas de transporte, congestionamiento o sobrecarga. También es utilizada para determinar la utilización de caminos y estadísticas, que permitan mantener un registro y contabilizar la utilización del *Grid*.

La tercera, la administración avanzada, puede automáticamente manejar varios aspectos del *Grid*. Esto es conocido como computación automática, o computación orientada a la recuperación. Este software automáticamente se recobra de varios tipos de fallas, buscando maneras de finalizar el proceso cargado.[23]

1.4.4.2. Administración distribuida

Los grandes *grids,* pueden tener jerarquías u otro tipo de topología organizacional, usualmente basada en la topología de conectividad. El *Grid* puede ser organizado en una jerarquía consistente en un *cluster* de *clusters*. El trabajo consistente en la administración de un *Grid* se distribuye para incrementar la escalabilidad. Mientras que el conjunto de operaciones y recursos de datos, así como la planificación de trabajos está distribuida para encajar con la topología del *Grid*. Por ejemplo, un planificador central de trabajos no podría planificar un trabajo directamente a la máquina donde se ejecutara. En vez de eso, el trabajo es enviado a un planificador de menor nivel que tiene a su cargo un conjunto de máquinas, y este se encarga de la asignación a la máquina específica. De manera similar, la recolección de información estadística está distribuida. [24]

1.4.4.3. Donadores

Cada máquina que contribuye con el *Grid,* generalmente, necesita enrolarse como miembro e instalar algún software que administre el uso de sus recursos por parte del *Grid*. Usualmente, alguna clase de procesos de identificación y autenticación deben ser realizados antes de que una máquina pueda ser agregada al *Grid*. Generalmente, pueden usarse certificados para establecer y asegurar la identidad de la máquina donante, así como también identificar a los usuarios, e incluso al *Grid* mismo.

Algunos sistemas *Grid* proveen su propio acceso, mientras otros dependen de una autenticación nativa del sistema operativo. En el último de los casos, el usuario puede necesitar que se le conceda permisos sobre las diferentes máquinas. Esto usualmente recibe mantenimiento manual por parte del administrador del *Grid*, él determina, a que usuario de cada máquina, se le puede asignar un identificador, y luego agrega esto a una base de datos protegida. De esta forma, cuando los trabajos son remitidos a diferentes máquinas por un usuario, la máquina local lo identifica y verifica sus permisos.

Aunque en algunos sistemas *Grid*, es posible unirse sin ninguna autenticación, y en otros es posible que cualquier usuario envié trabajos al *Grid*, esto puede provocar serios problemas de seguridad.

El sistema *Grid* constantemente recopila información acerca de los nuevos recursos disponibles. Para esto, la máquina donante puede tener alguna clase de monitor que determine o realice alguna medida de que tan ocupada esta la máquina, y el porcentaje de utilización de los recursos. Esta información es suministrada al software de administración y es utilizada para planificar la utilización de los recursos.

El software instalado en una máquina especifica, puede aceptar un trabajo del sistema de administración y ejecutarlo. Inicialmente un usuario cualquiera puede enviar un trabajo para su ejecución. El administrador debe comunicarse con el software donante para enviarle el trabajo. El software donante debe ser capaz de recibir el archivo ejecutable o seleccionar las copias apropiadas del mismo pre instaladas en la máquina. El trabajo es ejecutado y la respuesta es retornada a quien lo envió. Implementaciones más avanzadas pueden ajustar dinámicamente la prioridad de un trabajo que está corriendo, suspenderlo y reiniciar su ejecución más tarde, o tener la posibilidad de continuarlo, en una máquina diferente. Este tipo de acciones pueden ser necesarias para responder a problemas de carga balanceada o políticas de prioridades.[25]

1.4.4.4. Software de sometimiento

Usualmente cualquier miembro de un *Grid* puede ser utilizado para someter un trabajo e iniciar consultas en el *Grid*. Sin embargo, en algunos sistemas, esta función es implementada como un componente separado en nodos de sometimiento o clientes de sometimiento.[26]

1.4.4.5. Planificadores

Muchos sistemas *Grid* incluyen algún tipo de software planificador de tareas. Este Software localiza una máquina en el *Grid* en donde el trabajo sometido por el usuario pueda ser ejecutado. En este caso simple, esto podría resolverse asignando el trabajo utilizando la estrategia *round-robin* hacia la siguiente máquina que cumpla con los requerimientos. Sin embargo, existen ventajas si se utilizan planificadores más avanzados.

Algunos planificadores implementan un sistema de prioridades de trabajo. Esto se hace utilizando varias colas, cada una con diferente prioridad, tan pronto una máquina esté disponible para ejecutar un trabajo, se inicia por tomar de la cola de más alta prioridad. Se pueden implementar políticas de varios tipos en el planificador, estas pueden incluir varios tipos de restricciones sobre trabajos, usuarios y recursos. Por ejemplo, restringir que un trabajo se ejecute a cierta hora del día.

Planificadores más avanzados pueden monitorear el progreso de los trabajos administrando todo el flujo del trabajo. Si el trabajo se perdiera por fallas en la red o el sistema, un buen planificador podría automáticamente reenviarlo. Asimismo, si un trabajo pareciera estar en un ciclo infinito y alcanza su tiempo máximo de espera, el trabajo no debería ser re planificado. Típicamente, los trabajos tienen diferentes tipos de códigos cuando son completados, algunos son sujetos a re envió y otros no.

La reserva de recursos sobre el *Grid,* se complementa, con la reservación del sistema. Esto es más que un planificador, es un sistema basado en el calendario para la reservación de recursos por tiempos específicos, y prevenir que otros reserven el mismo recurso para la misma fecha. Este debe ser capaz también, de remover o suspender trabajos que estén corriendo en cualquier máquina o recurso cuyo periodo de reservación ha concluido.[27]

1.4.4.6. Comunicaciones

El *Grid* puede incluir software para ayudar a los trabajos a comunicarse con otros. Por ejemplo, una planificación puede dividirse en un gran número de trabajos. Cada uno es un trabajo separado en el *Grid*. Sin embargo, la aplicación puede implementar un algoritmo que requiere que los trabajos transfieran alguna información entre ellos. Los sub trabajos necesitan ser capaces de localiza a un trabajo específico, establecer comunicación con él y enviarle la información adecuada. [28]

1.4.4.7. Observación y mediciones

Usualmente el software donador, incluye alguna herramienta para medir la carga actual de actividad en una máquina dada, utilizando las características del sistema operativo o por mediciones directas. Algunos sistemas *Grid*, proveen los conceptos para implementar sensores personalizados, para otros componentes diferentes al CPU y dispositivos de almacenamiento.

Algunas mediciones de información son útiles para el planificador y otras no, pero de cualquier manera estas permiten describir el conjunto de patrones de utilización en el *Grid*. Las estadísticas pueden mostrar tendencias que sean señal de la necesidad de hardware adicional. También, la información de mediciones a cerca de un trabajo especifico, pueden ser recolectadas y utilizadas para una mejor predicción de los recursos requeridos, por un trabajo en la siguiente corrida. Una mejor predicción, puede hacer más eficiente la administración de la carga de trabajo en el *Grid*.

La información de mediciones puede también ser almacenada para propósitos de control, o para definir las bases de planificación o administración de prioridades. Esta información puede ser desplegada en varias formas para visualizar mejor la actividad y utilización del *Grid*.[29]

1.5. Aplicaciones del *Grid*

Los enfoques computacionales, orientados a la resolución de problemas, tienen probado su valor en casi todos los campos del conocimiento humano. Las computadoras son utilizadas para modelar y simular problemas complejos en el área científica y de la ingeniería, diagnosticar condiciones médicas, controlar equipo industrial, medición del clima, administración de inventarios y muchos otros propósitos. Por tal motivo, es válido preguntarse, ¿a qué tipo de aplicaciones podría orientarse el *Grid*?

Ian Foster y Carl Kesselman, basados en su experiencia proponen cinco categorías de aplicaciones, las cuales agrupan a las aplicaciones de acuerdo a su propósito general.

Tabla I - Categorías de las aplicaciones *Grid*

Categoría	Características
Computación Distribuida	Problemas realmente grandes, que necesitan gran cantidad de recursos (CPU, memoria, disco, etc.)
Alto Rendimiento	Aprovechar los recursos que se encuentren ociosos para incrementar el rendimiento total.
Por Demanda	Integrar los recursos remotos con la capacidad de cómputo local, para disminuir el tiempo de respuesta
Procesamiento de Datos	Sintetizar nueva información de muchas fuentes de datos o de fuentes de datos muy grandes.
Colaboración	Soporte de comunicación o trabajo en colaboración entre múltiples participantes.

Fuente: Ian Foster y Carl Kesselman, Five major classes of *Grid* applications, p. 22

1.5.1. Computación distribuida

Las aplicaciones de computación distribuida, utilizan el *Grid* para hacer frente a problemas que no podrían ser solucionados en un único sistema, agregando sustancialmente recursos computacionales, cuando sea necesario. Dependiendo del *Grid*, estos recursos agregados, pueden estar compuestos, en su mayoría, por súper computadoras de una localidad o simplemente por estaciones de trabajo dentro de una compañía.

Los aspectos desafiantes desde la perspectiva de la arquitectura del *Grid*, incluyen la necesidad de planificar los recursos que a menudo son escasos o costosos, la escalabilidad de protocolos y algoritmos para cientos o miles de nodos, algoritmos de tolerancias y lograr mantener altos niveles de desempeño a través de sistemas heterogéneos. He aquí algunos ejemplos de este tipo de aplicaciones.

- La Simulación Interactiva Distribuida (DIS), es una técnica utilizada para el entrenamiento y la planificación militar. Donde escenarios reales pueden envolver cientos o miles de entidades, cada una con un patrón complejo de comportamiento. Incluso las súper computadoras más grandes pueden manejar un máximo de 20,000 entidades. En un trabajo reciente, el equipo de investigación del *California Institute of Technology*, ha demostrado como múltiples computadoras pueden ser enlazadas para alcanzar niveles de desempeño sin precedentes, en este tipo de experimentos.

- La simulación exacta de procesos físicos complejos puede requerir resolución temporal y espacial con el objetivo de resolver el más mínimo detalle. Las computadoras unidas pueden ser utilizadas para superar barreras de la resolución y por tanto obtener nuevos resultados científicos.[30]

1.5.2. Alto rendimiento

En las aplicaciones de alto rendimiento, el *Grid* es utilizado para administrar un gran número de tareas que están entrelazadas o que son independientes, con el objetivo de utilizar los ciclos de procesador que no están siendo utilizados (frecuentemente de una terminal ociosa). El resultado puede ser, como en una supercomputadora distribuida, enfocar los recursos disponibles sobre un problema, pero la naturaleza casi independiente de las tareas puede envolver un conjunto de problemas diferentes y métodos para solucionarlos. He aquí algunos ejemplos de este tipo de aplicaciones.

- *Platform Computing Corporation*, reportó que la empresa de manufacturación AMD, utilizó este tipo de tecnología durante de la fase de diseño de sus microprocesadores K6 y K7, para explotar más de mil computadores. Estas computadoras están localizadas en los escritorios de los ingenieros de AMD y son utilizadas para el diseño, únicamente cuando se verifica que no están siendo utilizadas por los ingenieros.

- El sistema Cóndor, de la universidad de Wisconsin, es utilizado para manejar grupos de cientos de estaciones de trabajo, de la universidad y laboratorios alrededor del mundo. Estos recursos han sido utilizados para estudios y diversas simulaciones moleculares de cristal líquido, estudios geológicos con radares y diseño de motores de diesel.

- Otras pocas organizaciones tienen amarradas decenas de miles de computadoras distribuidas en el mundo para la resolución de problemas de criptografía.[31]

1.5.3. Por demanda

Las aplicaciones por demanda utilizan las capacidades del *Grid* para satisfacer los requerimientos de recursos que serán utilizados por tiempos cortos, y que por razones de rentabilidad o conveniencia, no necesitan estar en la misma localidad. Estos recursos pueden ser: procesador, software, repositorios de datos, sensores especializados y otros. En contraste con las aplicaciones de computación distribuida, estas aplicaciones son frecuentemente manejadas por costo – beneficio, en vez del desempeño absoluto.

Los aspectos desafiantes en aplicaciones por demanda, derivan principalmente de la naturaleza dinámica de los requerimientos de recursos y de la gran cantidad de usuarios y de recursos que, potencialmente, pueden estar involucrados. Estos aspectos incluyen la ubicación de recursos, planificación, administración de código, configuración, tolerancia a errores, seguridad y mecanismos de pago. He aquí algunos ejemplos de este tipo de aplicaciones.

- NEOS y NetSolve, una red mejorada para sistemas de resolución de problemas numéricos, permite que los usuarios, a través de una aplicación en el escritorio, puedan emparejar software y recursos remotos, para luego despachar a estos los cálculos que demandan o que requieren de un software especial.

- Un sistema desarrollado por la *Aerospace Corporation*, para procesamiento de datos meteorológicos procedentes de satélites, utiliza recursos dinámicos adquiridos de supercomputadoras, para ejecutar algoritmos de detección de nubosidad y entregar los resultados a los meteorólogos casi en tiempo real.[32]

1.5.4. Procesamiento de datos

Las aplicaciones de procesamiento de datos, se enfocan en sintetizar nueva información que se encuentra geográficamente distribuida en repositorios, bibliotecas digitales o bases de datos. Este proceso de síntesis requiere tanto de comunicación intensiva como de capacidad computacional.

Los aspectos desafiantes en aplicaciones de procesamiento de datos, está en la complejidad para la planificación y configuración del flujo de altos volúmenes de datos, que deben atravesar por múltiples niveles de jerarquías. He aquí algunos ejemplos de este tipo de aplicaciones.

- Experimentos futuros de física con energía, pueden generar terabytes de datos en un día, y cerca de un petabyte por año. Y las consultas complejas utilizadas para detectar eventos interesantes, pueden requerir una gran cantidad de estos datos. Los científicos colaboradores que deseen acceder estos datos pueden estar distribuidos, así como también los sistemas de datos en que se almacena esta información puede estar distribuida.

- El *Digital Sky Survey* es un sistema que deberá manejar muchos terabytes de información fotográfica disponible en numerosas bases de datos. Esta facilidad proporcionará un nuevo enfoque en la investigación astronómica, basándose en análisis distribuidos.

- Los sistemas meteorológicos modernos, que pronostican el clima, hacen uso extensivo la asimilación de datos para incorporarlo a las observaciones remotas de los satélites. El proceso completo implica los movimientos y procesamiento de muchos gigabytes de datos.[33]

1.5.5. Colaboración

Las aplicaciones de colaboración están concebidas principalmente para establecer y mejorar la interacción humano - humano. Y son frecuentemente estructuradas en términos de espacio virtual compartido, de manera que están creadas para permitir que se compartan recursos computacionales tales como archivos de datos y simulaciones.

Los aspectos desafiantes en las aplicaciones de colaboración están enfocados en la arquitectura del *Grid,* debido a los requerimientos en tiempo real impuestos por las capacidades de percepción humana y la gran variedad de interacciones que pueden tomar forma. En este caso estas aplicaciones, pueden compartir ciertas características con las categorías descritas anteriormente, he aquí algunos ejemplos de este tipo de aplicaciones.

- El sistema *Boiler Maker* desarrollado en el *Argonne National Laboratory,* permite a múltiples usuarios, colaborar en el diseño del sistema de control de emisiones de los incineradores industriales. Los diferentes usuarios interactúan con otros y con la simulación del incinerador.

- El sistema CAVE5D soporta, remotamente, colaboración para la exploración de grandes grupos de datos geofísicos y los modelos que los generan.

- El sistema NICE desarrollado en la Universidad de Illinois en Chicago permite a los niños participar en la creación y mantenimiento de mundos virtuales, para entretenimiento y educación.[34]

1.6. Arquitectura del *Grid*

El establecimiento, administración y explotación de organizaciones virtuales de forma dinámica, requiere de nueva tecnología. Y esta tecnología, tiene como base una arquitectura capaz de identificar los componentes fundamentales de los sistemas, especificar el propósito y función de los componentes, así como la interacción entre ellos.

La definición de la arquitectura del *Grid*, inicia desde la perspectiva de las operaciones requeridas por las organizaciones virtuales, para establecer relaciones de colaboración con cualquier participante potencial.

La interoperabilidad es el principal aspecto a tomar en cuenta. En un entorno de red, la interoperabilidad significa protocolos comunes. Por tanto, la arquitectura del *Grid* es una arquitectura orientada a los protocolos, porque con los protocolos, se definen los mecanismos básicos para que los usuarios de las organizaciones virtuales y los recursos negocien, establezcan, administren y exploten las relaciones para compartir.

La importancia de la interoperabilidad, surge de la necesidad de asegurar que las relaciones de cooperación puedan ser iniciadas por cualquier miembro, acomodándose dinámicamente a nuevos participantes y utilizando diferentes plataformas, lenguajes y entornos de programación. Sin la interoperabilidad, las aplicaciones de las organizaciones virtuales y los participantes estarían forzados a entrar en arreglos bilaterales de recursos compartidos, esto sin la certeza de que los mecanismos utilizados entre dos participantes puedan ser extendidos a otros participantes. En este contexto, los mecanismos sirven a pequeños propósitos, están definidos e implementados para operar entre los límites de las organizaciones, políticas de operación y tipos de recursos.

La definición de protocolos, por su parte, especifica como los elementos distribuidos de un sistema interactuaran con otros, de tal forma que se adquiera un comportamiento determinado; y especifica la estructura de información que se intercambia durante esta interacción. El enfoque de los protocolos se centra en lo externo (iteración) y no en lo interno (software, características de los recursos) teniendo importantes beneficios pragmáticos. Las organizaciones virtuales tienden a ser cambiantes, por tanto, los mecanismos utilizados para descubrir recursos, establecer identidades, determinar autorizaciones e iniciar a compartir, deben ser flexibles y livianos, para que los arreglos de recursos compartidos puedan ser establecidos y cambiados rápidamente. La interacción de componentes por medio de protocolos garantiza que el control local sea preservado.

En la arquitectura, una base estándar y abierta, facilita la ampliación, interoperabilidad, portabilidad y la posibilidad de compartir código, así mismo un protocolo estándar permite definir servicios estándar que proporcionen capacidades mejoradas. Un servicio es definido solamente por el protocolo con el que se comunica y el comportamiento que este implementa. La definición de servicios estándar (para acceso del procesador, datos, búsqueda de recursos, planificación, replicación de datos y otros), permite ampliar los servicios que se ofrecen a los participantes de la organización virtual, así como abstraer detalles específicos de los recursos, que de otra manera dificultarían el desarrollo de aplicaciones.

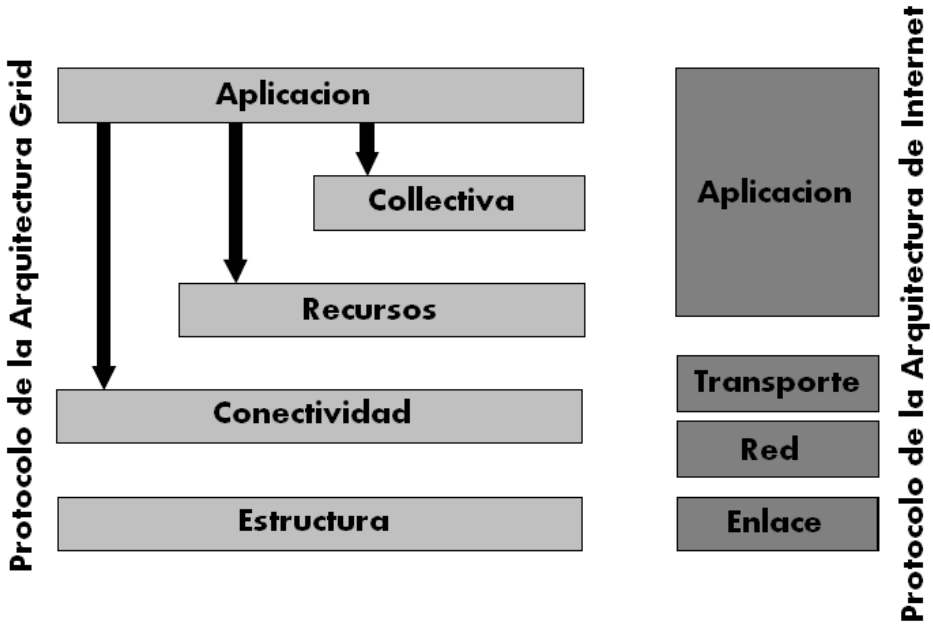
Además de interoperabilidad, protocolos y servicios, los desarrolladores deben tener la capacidad de desarrollar aplicaciones sofisticadas en entornos complejos y dinámicos, mientras que los usuarios deben ser capaces de poder operarlas. Esto conlleva a la construcción de *APIs (Application Programming Interfaces)* y *SDKs (Software Development Kits)*, que provean al desarrollador de la abstracción requerida para crear aplicaciones *Grid* que puedan ser aprovechables. Los conceptos de costos de desarrollo, costos de mantenimiento, correcciones y que las aplicaciones sean robustas es importante. Por tal motivo los APIs y SDKs son una adición, y no una alternativa, a los protocolos.

Juntos, la tecnología y la arquitectura, constituyen lo que frecuentemente se conoce como middleware (los servicios necesarios para soportar un conjunto común de aplicaciones en un entorno de red distribuido).[35]

1.7. Descripción de la arquitectura

La meta de describir la arquitectura del *Grid*, no es proveer una enumeración completa de todos los protocolos requeridos (servicios, APIs y SDKs), sino identificar los requerimientos para una clase de componentes generales. El resultado es una extensa, arquitectura abierta, en cuya estructura pueden situarse soluciones para requerimientos claves de una organización virtual. La arquitectura organiza los componentes en capas, como se muestra en la figura 1, los componentes dentro de cada capa comparten características comunes, pero pueden estar construidas sobre capacidades y comportamientos de una capa inferior.

Figura 1 - Capas de la arquitectura *Grid* y su relación con la arquitectura IP



Fuente: Ian Foster y otros, ***The Anatomy of the Grid***, p. 7

La especificación de capas de la arquitectura *Grid*, sigue los principios del "modelo de reloj de arena". En el angosto cuello del reloj, se define un pequeño conjunto de núcleos de abstracción y protocolos (ej. HTTP, TCP), sobre los cuales pueden delinearse una amplia variedad de comportamientos de alto nivel (la parte alta del reloj), y que de igual manera pueden ser diseccionados a diferentes tecnologías subyacentes (la base del reloj). Por definición el número de protocolos definidos en el cuello debe ser pequeño, en este caso está conformado por las capas de **recursos** y **conectividad**, orientados a facilitar la tarea de compartir recursos individuales. Los protocolos de estas capas están diseñados para que puedan ser implementados sobre los diferentes tipos de recursos, representado por la capa de **estructura**, y que puedan a su vez ser utilizados para construir una amplia variedad de servicios globales y comportamientos específicos para cada aplicación, en la capa **colectiva**[††].

La descripción de la arquitectura, se hace a alto nivel por lo que provee pocas condiciones para el desarrollo y la implementación, y constituyen una abstracción de la funcionalidad de cada capa.[36]

1.7.1. Estructura. Interfaces para el control local

La capa de estructura, provee los recursos, con los cuales los protocolos del *Grid* podrán mediar, para acceder a sus características compartidas (capacidad de cálculo, almacenamiento, catálogos, recursos de red, etc.). Un recurso puede ser una entidad lógica, como un sistema de archivos distribuido, un *cluster* de computadores o un *pool* de computadoras distribuido; en donde, la implementación de un recurso puede envolver protocolos internos (como por ejemplo, almacenamiento NFS o administración de recursos del cluster), lo cual ya no concierne a la arquitectura del *Grid*.

[††] Llamada así porque envuelve un uso coordinado (colectivo) de los recursos.

Los componentes de la capa de estructura implementan lo local, operaciones específicas que ocurren sobre recursos específicos (tanto físicas como lógicas), que son el resultado de operaciones efectuadas en niveles más altos, sobre lo que se encuentra compartido. Creando así, una interdependencia, entre las funciones implementadas a nivel de la capa de estructura y las operaciones soportadas para compartir. Una funcionalidad rica en la capa de estructura, permite operaciones mucho más sofisticadas sobre los recursos compartidos, como por ejemplo la reservación anticipada de recursos.

Ian Foster y Carl Kasselman, basados en su experiencia, sugieren que como mínimo los recursos deben implementar mecanismos de consulta, que les permitan descubrir su propia estructura, estado y capacidades (ej. reservación anticipada), así como mecanismos de administración, que provean algún tipo de control de la calidad de servicio (QoS) entregada. La siguiente es una lista breve y parcial sobre las características específicas de los recursos.

- Recursos de cómputo. Requiere de mecanismos para iniciar programas, y posteriormente monitorear y controlar la ejecución de los procesos resultantes. Es importante contar con mecanismos de administración, que permitan tener el control sobre los recursos asignados, así como mecanismos de reservación anticipada. Las funciones de búsqueda son necesarias para determinar las características del hardware o software, así como información relevante sobre su estado, tal como carga actual o el estado de las colas, en caso de que los recursos sean administrados de forma planificada.

- Recursos de red. Requiere de mecanismos de administración que provean el control sobre los recursos asignados a la transferencia de red (ej. prioridades y reservación). Las funciones de búsqueda pueden ayudar a determinar las características de la red, así como información sobre su estado y carga.

- Recursos de almacenamiento. Requiere de mecanismos para la colocación y obtención de archivos. Es importante contar con mecanismos para la lectura y escritura de porciones de archivos y/o la ejecución de un grupo de datos seleccionados. Además de mecanismos de administración y reservación anticipada, que permitan el control sobre los recursos asignados para la transferencia de datos (espacio, ancho de banda del disco, ancho de banda de la red, CPU, etc.). Las funciones de búsqueda son requeridas para determinar las características de hardware y software, así como información relevante de la carga, tal como espacio disponible y la utilización del ancho de banda.

- Repositorios de código. Constituye una forma especializada de almacenamiento, que requiere de mecanismos para la administración de versiones de código y de objetos, por ejemplo un sistema de control como el CVS.

- Catálogos. Constituye una forma especializada de almacenamiento, que requiere de mecanismos para la implementación de consultas a catálogos y operaciones de actualización, por ejemplo una base de datos relacional.[37]

1.7.2. Conectividad. Comunicación fácil y segura.

La capa de conectividad, define un núcleo de comunicaciones y protocolos de autenticación, requeridos por las transacciones de red relativas al *Grid*. El protocolo de comunicación permite el intercambio de datos entre los recursos de la capa de estructura. Mientras que los protocolos de autenticación, construidos sobre los servicios de comunicación, proveen mecanismos de seguridad criptográfica, para verificar la identidad de los usuarios y los recursos.

La comunicación incluye aspectos como transporte, selección de rutas e identificación de recursos por nombre. Debido a la existencia de alternativas viables, las capas de red, transporte y aplicación del protocolo TCP/IP, representan la base para el protocolo de comunicación. Sin embargo, en el futuro, las comunicaciones del *Grid* pueden demandar de nuevos protocolos.

En los aspectos de seguridad de la capa de conectividad, la complejidad de los problemas demanda que se adopten estándares existentes cuando esto sea posible. Así como en las comunicaciones, muchos de los estándares de seguridad desarrollados dentro del contexto del protocolo de Internet son aplicables.

Las soluciones de autenticación, en el entorno de las organizaciones virtuales, deben tener las siguientes características:

- Inicialización. Los usuarios deben tener la capacidad de hacer *"log on"* (autenticarse), una única vez, y entonces tener accesos a los recursos del *Grid* definidos en la capa de estructura, sin la intervención del usuario.

- Delegación. El usuario debe de proporcionar un programa, con la habilidad de ejecutarse sobre lo que se encuentra compartido, además debe ser capaz de acceder a los recursos en los que el usuario está autorizado. El programa debe (opcionalmente) de ser capaz de delegar derechos a otros programas.

- Integración con las soluciones de seguridad local. Cada sitio o recurso compartido, pueden emplear una gran variedad de soluciones locales de seguridad, incluyendo Kerberos o la seguridad de Unix. La seguridad del *Grid* debe ser capaz de interactuar con esas soluciones. En vez de requerir que se reemplace las soluciones de seguridad local, se deben permitir el mapeo del entorno.

- Relaciones basada en la confianza. La utilización simultanea de recursos, de múltiples proveedores, no debe requerir la autenticación o reconfiguración del sistema por cada nuevo recurso utilizado. Por ejemplo, si un usuario tiene acceso a utilizar los sitios A y B, el usuario tiene capacidad de utilizar ambos sin necesidad de que se deba interactuar con la seguridad de cada sitio.

Las soluciones de seguridad, deben proveer un soporte flexible para la protección de las comunicaciones (ej. control del grado de protección, protección independiente de datos para los protocolos que no son fiables, soporte para protocolos de transporte que son fiables además del TCP) y permitir el control sobre los participantes que toman decisiones de autorización, incluyendo la habilidad de restringir la delegación de derechos en varias formas.[38]

1.7.3. Recursos. Compartir recursos individuales.

La capa de recursos, construida sobre la capa de conectividad, define los protocolos (así como APIs y SDKs) para la negociación segura, iniciación, monitoreo, control, contabilización y pago de operaciones sobre recursos individuales. La implementación de los protocolos de la capa de recursos, llama a las funciones de la capa de estructura, para acceder y controlar los recursos locales. Los protocolos de la capa de recursos, están orientados al manejo de los recursos individuales y por tanto ignoran temas como el estado global y acciones atómicas entre colecciones de recursos distribuidos.

Existen dos clases de protocolos para la capa de recursos:

- Protocolos de información. Que son utilizados para obtener información acerca de la estructura y estado de los recursos, por ejemplo, su configuración, carga actual y políticas para su utilización (ej. costo).

- Protocolos de administración. Que son utilizados para negociar el acceso a los recursos compartidos, especificando, por ejemplo, los requerimientos de recursos (incluyendo reservación anticipada y calidad del servicio) y las operaciones a ser realizadas (tales como creación de procesos y acceso de datos). El protocolo de administración será responsable por la inicialización de las relaciones para compartir, y debe asegurar que las operaciones requeridas, cumplan con las políticas bajo las cuales el recurso ha sido compartido. Entre los aspectos que deben ser considerados se incluyen la contabilización y el pago, así como puede considerarse también soporte para el monitoreo del estado y control de una operación.

Los protocolos de la capa de recursos (y conectividad) forman el cuello del modelo de reloj de arena, y deben ser limitados a un pequeño grupo especializado. Estos protocolos deben ser seleccionados, de tal manera que capturen los mecanismos fundamentales para compartir a través diferentes tipos de recursos.

La lista de funcionalidades para la capa de estructura, resume las características requeridas en la capa de recursos. Agregando a esta lista, la necesidad de contar con una única semántica para muchas operaciones, con disponibilidad para reportar errores.[39]

1.7.4. Colectiva. Coordinación de múltiples recursos.

Mientras la capa de recursos está enfocada en la interacción con un único recurso, la siguiente capa en la arquitectura contiene protocolos y servicios (así como APIs y SDKs) que no se encuentran asociados con ningún recurso especifico, en vez de esto, tienen una naturaleza global y capturan la interacción entre los conjuntos de recursos. Por esta razón, la siguiente capa de la arquitectura es conocida como capa colectiva.

Los componentes de la capa colectiva se construyen sobre la delgada capa de recursos y conectividad (el cuello en el modelo de reloj de arena), y pueden implementar una amplia variedad de comportamientos compartidos, sin imponer nuevos requerimientos a los recursos compartidos.

A continuación se ilustra, con algunos ejemplos, la amplia variedad de protocolos y servicios que pueden estar disponibles en la capa colectiva. Notando que, mientras los protocolos de la capa de recursos pueden ser de propósito general y extensamente implementados, los protocolos de la capa colectiva van del espectro general a aplicaciones de propósitos específicos.

- Servicios de directorio. Permiten a los participantes de las organizaciones virtuales, descubrir la existencia y/o propiedades de los recursos. Un servicio de directorio, puede permitir a sus usuarios la consulta de recursos por su nombre y/o atributos como: tipo, disponibilidad o carga. Los protocolos, del nivel de recursos, son utilizados para construir los directorios.

- Reservación, planificación y administración de servicios. Permite a los participantes de las organizaciones virtuales, consultar sobre la reservación de uno o más recursos para un propósito específico y la planificación de tareas, con los recursos apropiados.

- Servicios de monitoreo y diagnóstico. Soportan el monitoreo de los recursos de las organización virtual por fallas, ataques de adversarios (detección de intrusos) y sobrecarga.

- Servicios de replicación de datos. Permiten la administración de los recursos de almacenamiento de las organizaciones virtuales (así como también de los recursos de red y procesamiento), para maximizar el desempeño en el acceso a los datos, basándose en métricas tales como tiempo de respuesta, disponibilidad y costo.

- Sistemas de programación *Grid*. Proporciona modelos familiares de programación para ser utilizados en entornos *Grid*, utilizando servicios existentes para el descubrimiento de recursos, seguridad, reservación de recursos y otros.

- Sistema de administración de la carga de trabajo y marco de trabajo para la colaboración. También conocido como entornos para la resolución de problemas (PSEs *Problem Solving Environments*), proporciona las facilidades necesarias para solucionar un problema objetivo, en un dominio específico, mediante la descripción, administración y utilización de flujos de trabajo.

- Servicios de descubrimiento de software. Descubren y seleccionan, la mejor implementación de software y plataforma de ejecución, basándose en los parámetros de resolución del problema.

- Servicios comunitarios de autorización. Refuerza las políticas comunitarias que gobiernan el acceso a los recursos, generando capacidades que los miembros de la comunidad pueden utilizar para acceder a los recursos.

- Contabilidad y pago de servicios. Recopila información acerca de la utilización de los recursos, con el propósito de contabilizar, remunerar, y/o limitar la utilización de los recursos por parte de la comunidad.

- Servicios de colaboración. Soportan el intercambio coordinado de información entre usuarios potenciales de la comunidad, sea síncrono o asíncrono.

Las funciones colectivas pueden ser implementadas como servicios persistentes (con protocolos asociados) o como un SDK (con un API asociado) diseñado para enlazarse con aplicaciones. En ambos casos, su implementación puede construirse sobre la capa de recursos (u otra capa colectiva) y APIs.

Los componentes de la capa colectiva pueden ser confeccionados de acuerdo a los requerimientos específicos de un usuario de la comunidad, de la organización virtual o basada en una aplicación, como por ejemplo un servicio de reservación para un recurso de red especifico. Otros componentes en cambio, pueden ser de propósito más general, como por ejemplo un sistema para el descubrimiento de recursos en la organización virtual. Pero lo principal, en la capa colectiva, es que los protocolos que lo componen y sus APIs están basados en estándares.[40]

1.7.5. Aplicación.

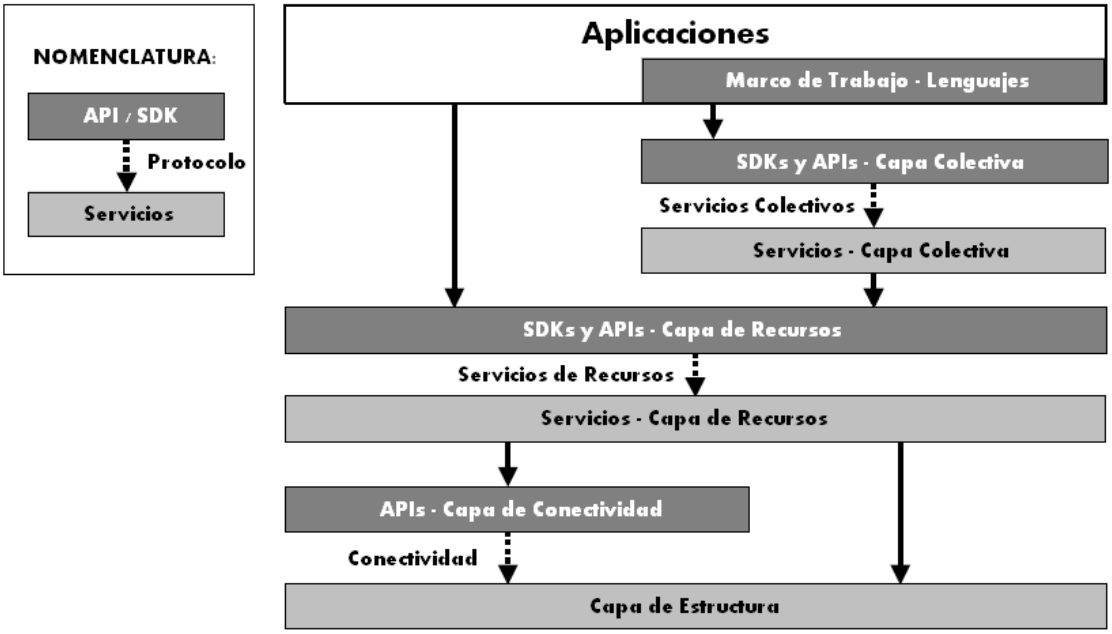
La capa final de la arquitectura *Grid* comprende las aplicaciones para los usuarios que operaran dentro del entorno de la organización virtual. Las aplicaciones son construidas en base a los servicios definidos en cada capa, para luego ser ejecutadas. En cada capa, existen protocolos bien definidos que proporcionan acceso a muchos servicios útiles: administración de recursos, acceso a datos, búsqueda de recursos y otros. Y en cada capa, los APIs pueden estar definidos para implementar el intercambio de mensajes con un servicio apropiado que ejecute las acciones deseadas.

Se hace énfasis en que, lo que se etiqueta como "aplicaciones", y que se muestra en una simple capa en la figura 2, puede ser en la práctica sofisticados marcos de trabajo y librerías, que se caractericen por una estructura interna mucho más compleja. Estos marcos de trabajos pueden por si mismos definir protocolos, servicios y/o APIs.

La figura 2 ilustra la arquitectura *Grid* desde el punto de visita de un desarrollador de aplicaciones. Los APIs están implementados por los SDKs, quienes a su vez utilizan los protocolos del *Grid* para interactuar con los servicios de red que proveen las capacidades a los usuarios finales.

En un nivel alto, un SDK puede proveer funcionalidad que no está relacionada directamente a un protocolo especifico, pero puede combinar operaciones con protocolos que llamen a APIs adicionales para implementar operaciones locales. Las líneas sólidas representan una llamada directa, las líneas discontinuas representan interacciones con los protocolos.[41]

Figura 2 - Arquitectura *Grid*, visión del desarrollador de aplicaciones.



Fuente: Ian Foster y otros, ***The Anatomy of the Grid***, p. 13

2. *WEB SERVICES*

"Hoy, el principal uso del *World Wide Web*, es el acceso interactivo a documentos y aplicaciones. En casi todos los casos, tal acceso es realizado por humanos, típicamente trabajando a través de navegadores web, reproductores de música u otro sistema interactivo de aplicaciones. La web puede crecer significativamente en poder y alcance, si se amplia para soportar comunicaciones entre aplicaciones de un programa a otro."[42]

La naturaleza dinámica de las comunicaciones, y su rápido crecimiento, incrementa la necesidad de crear aplicaciones que interactúen con otras aplicaciones, incluso trascendiendo los límites de la organización y de las tecnologías asociadas a esta.

La web es una red informática, de gran éxito, que permite interacciones simples humano/máquina a escalas de Internet. El protocolo HTTP y el lenguaje HTML, son utilizados, hoy en día, por los navegadores para proyectar interfaces de usuario sobre una amplia variedad de dispositivos. La clave de este éxito, esta en la amplia aceptación alcanzada por el HTTP y el HTML, que tiene como base la relativa simplicidad de ambos, ya que están basados en texto y pueden ser implementados en una gran variedad de sistemas operativos.

Los *web services* toman muchos de los principios aplicados en la web, y aplican estos términos a la interacción computadora/computadora. Se comunican utilizando una serie de protocolos fundamentales, que comparten una arquitectura común y están pensados para que puedan ser implementados en una variedad de sistemas. Así como la web, los protocolos de los *web services* le deben mucho de su entorno basado en texto, al Internet y sus capas están diseñadas para que sean tan autónomas como sea posible.

El amplio espectro de los requerimientos para la interacción entre programas, ha forzado a los *web services* a tener un propósito más general que los primeros protocolos web. Los *web services* difieren de la web en su alcance. Mientras los protocolos web están diseñados para realizar búsquedas interactivas y presentar contenido, que en su gran mayoría es estático, la arquitectura *web services*, esta diseñada para soportar gran cantidad de interacciones dinámicas entre programas.[43]

El propósito de los *web services*, es proporcionar un modelo conceptual, para la creación de un ambiente distribuido, en el cual un número de aplicaciones o componentes de aplicaciones, puedan operar de forma transparente entre organizaciones, basándose en un diseño neutral, tanto en la plataforma como en el lenguaje.

2.1. Definición de *Web Services*

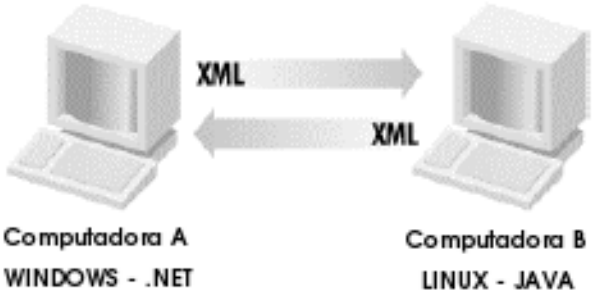
Luego de establecer el marco de referencia que rodea a los *web services*, es necesario proporcionar una definición concreta, para lo cual se ha seleccionado la definición proporcionada por el W3C[‡‡], publicada en el documento *Web Service Architecture*.

"Un *web service* es un sistema de software diseñado para soportar la interacción de operaciones máquina a máquina sobre una red. Esta tiene una interfaz descrita en un formato procesable por la máquina (específicamente WSDL[§§]). Otros sistemas interactúan con los *web services* de la forma descrita por esta especificación utilizando mensajes SOAP[***], típicamente transmitidos a través de HTTP con XML en conjunto con otros estándares web relacionados."[44]

[‡‡] *World Wide Web Consortium*. Es la principal organización de estándares para protocolos y especificaciones web.
[§§] *Web Service Definition Language*. Representa la capa de descripción de servicios. El WSDL es una gramática XML que permite especificar una interfase pública para la utilización del *web service*.
[***] *Simple Object Access Protocol*. Es un protocolo basado en XML para el intercambio de información entre computadoras. Puede ser utilizado por una variedad de sistemas de mensajes y entregado por medio de varios protocolos de transporte.

Los *web services* proveen una capa de abstracción, sobre sistemas de software existentes, utilizando XML para interactuar con programas, objetos, bases de datos o funciones de negocios integrales. Utilizando un documento XML creado en forma de mensaje, el programa envía una petición a un *web service* a través de la red, y, opcionalmente, recibe una respuesta, también en forma de documento XML. Un *web service* básico define el formato del mensaje, especifica la interfaz con que el mensaje será enviado, describe los contratos para determinar el contenido del mensaje (dentro y fuera del programa que implementa el servicio) y define los mecanismos para publicarlo, de manera que pueda ser descubierto por las interfaces de servicios.

Figura 3 - *Web Service* básico



Fuente: Cerami, Ethan, ***Web Service Essentials***, p. 6

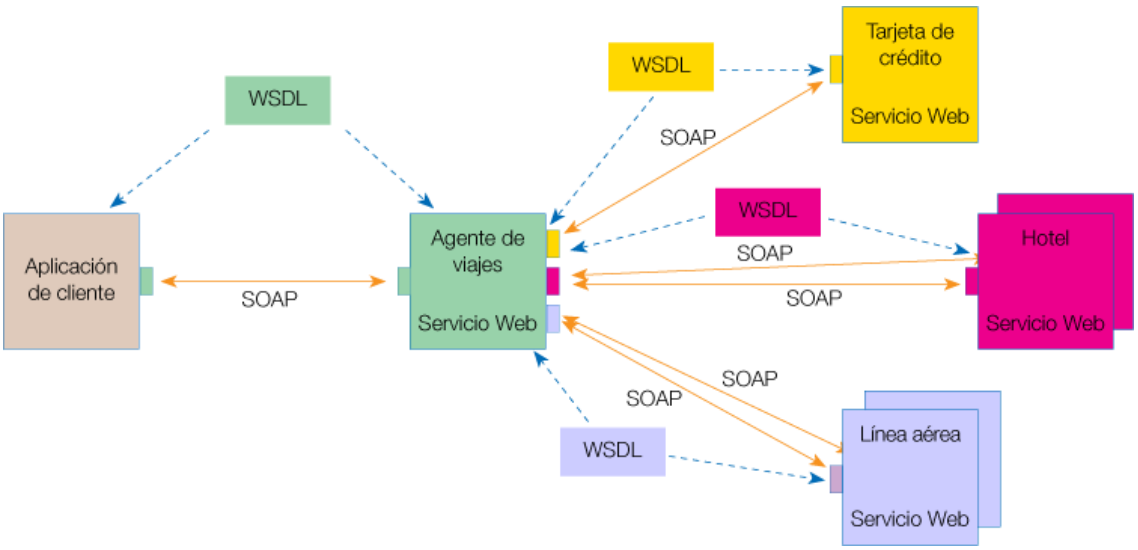
Un *web service* completo es aquel servicio que cumple con:

- Está disponible sobre Internet o sobre una red privada.
- Utiliza un sistema de mensajes estándar basado en XML.
- No está atado a ningún lenguaje de programación, ni a ningún sistema operativo.
- Se auto describe utilizando una gramática XML.
- Se puede descubrir utilizando un mecanismo de búsqueda simple.

2.2. Funcionamiento

En la siguiente gráfica muestra cómo interactúa un conjunto de *web services*:

Figura 4 – Funcionamiento de los *Web Services*

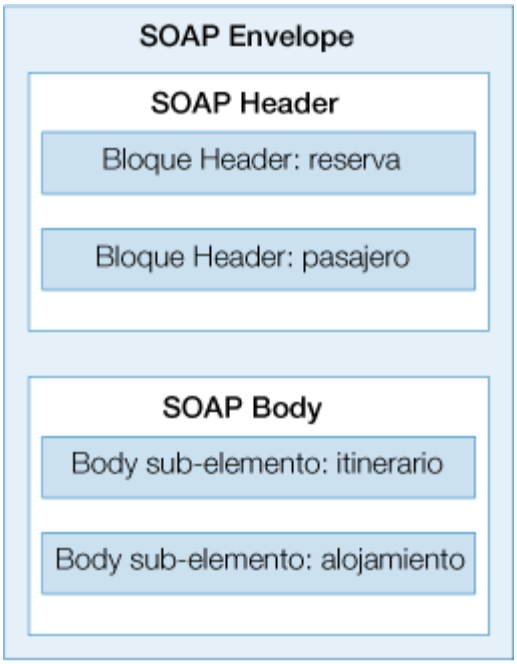


Fuente: W3C, http://www.w3c.es/divulgacion/guiasbreves/ServiciosWeb

Según el ejemplo del gráfico, un usuario (que juega el papel de cliente dentro de los *web services*), a través de una aplicación, solicita información sobre un viaje que desea realizar haciendo una petición a una agencia de viajes, que ofrece sus servicios a través de Internet. La agencia de viajes ofrecerá a su cliente (usuario) la información requerida. Para proporcionar al cliente la información que necesita, esta agencia de viajes solicita, a su vez, información a otros recursos (otros web services) en relación con el hotel y la línea aérea. La agencia de viajes obtendrá información de estos recursos, lo que la convierte, a su vez, en cliente de esos otros web services que le van a proporcionar la información solicitada sobre el hotel y la línea aérea. Por último, el usuario realizará el pago del viaje a través de la agencia de viajes, que servirá de intermediario entre el usuario y el web service que gestionará el pago.

En todo este proceso intervienen una serie de tecnologías que hacen posible la circulación de información. Por un lado se encuentra SOAP, protocolo basado en XML, que permite la interacción entre varios dispositivos, mediante el intercambio de mensajes, y que tiene la capacidad de transmitir información compleja. Los datos pueden ser transmitidos a través de diferentes protocolos, pero usualmente se utiliza HTTP/HTTPS. SOAP especifica el formato de los mensajes, en donde el mensaje está compuesto por un envelope (sobre), cuya estructura está formada por los siguientes elementos: header (cabecera) y body (cuerpo).

Figura 5 - Estructura de los mensajes



Fuente: W3C, http://www.w3c.es/divulgacion/guiasbreves/ServiciosWeb

Por otro lado, WSDL permite que un servicio y un cliente establezcan un acuerdo en lo que se refiere a los detalles de transporte de mensajes y su contenido, a través de un documento procesable por dispositivos. WSDL representa una especie de contrato entre el proveedor y el que solicita. El WSDL especifica la sintaxis y los mecanismos de intercambio de mensajes. [45]

2.3. Agentes y servicios

Un *web service* es una noción abstracta que debe ser implementada por un agente concreto. Un agente es una pieza de software o hardware que recibe y envía mensajes, mientras un servicio es el conjunto abstracto de funcionalidades que el agente provee.

2.4. Solicitantes y proveedores

La entidad proveedora es una persona u organización, que provee un agente apropiado para implementar un servicio en particular.

Una entidad solicitante, es un apersona u organización, que desea hacer uso de una entidad que provea algún servicio.

2.5. Descripción de servicios

Los mecanismos para el intercambio de mensajes, se encuentran documentados en el *web services description* (WSD). El WSD es una especificación de la interfaz *web service*, la cual debe ser procesada por una máquina y debe estar escrita en WSDL. En este se definen los formatos de los mensajes, tipos de datos, protocolos de transporte y la serialización de de los formatos de transporte, que serán utilizados por los agentes (solicitantes y proveedores). También pueden especificar una o más ubicaciones de la red, de donde se puede invocar al agente, y puede proporcionar cierta información acerca del patrón que se puede esperar durante el intercambio de mensajes. En esencia, representa un acuerdo que gobierna los mecanismos para interactuar con un servicio.

2.6. Semántica

La semántica del *web service*, representa la conducta esperada de un servicio en particular, como respuesta a los mensajes que le son enviados. En general representa un contrato, de los propósitos y consecuencias resultantes de la interacción de la entidad proveedora y la entidad solicitante.

Aunque este contrato representa el acuerdo general entre la entidad del solicitante y la entidad del proveedor, para la interacción de sus respectivos agentes, existe la probabilidad de que el contrato no se haya suscrito de forma explícita en un medio escrito. Los contratos pueden ser explícitos o implícitos, orales o escritos, procesados por una máquina o con intervención humana, y pueden ser un acuerdo legal o un acuerdo informal.

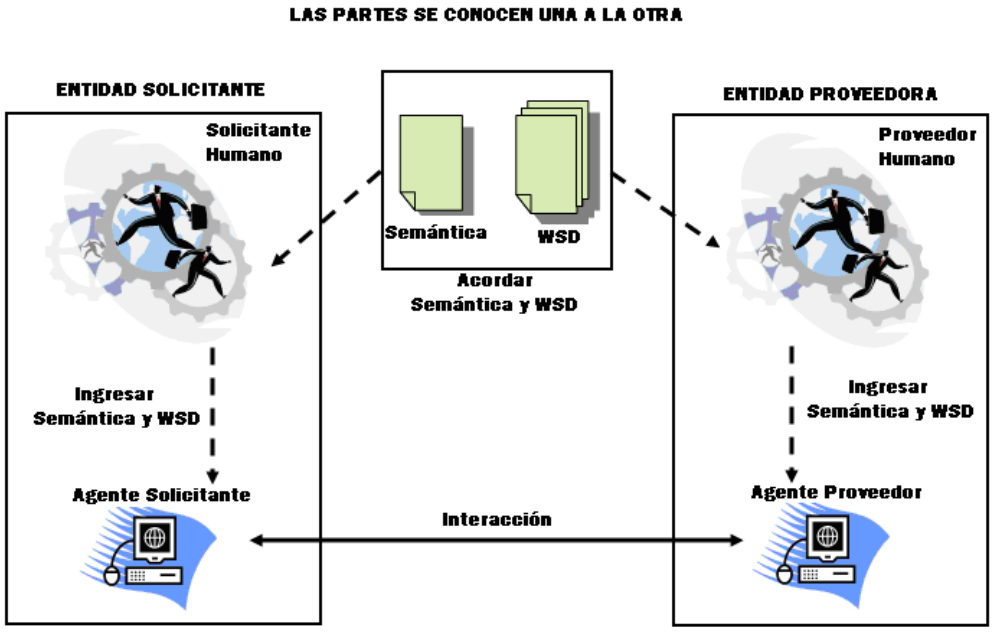
Mientras la descripción del servicio representa un contrato que gobierna a los mecánicos para interactuar con un servicio particular, el semántico representa un contrato que gobierna el significado y el propósito de esa interacción.

2.7. Vista generalizada de la interacción de *Web Services*

Existen muchas maneras en que una entidad solicitante puede interactuar y hacer uso de un *web service*. En general los siguientes pasos son requeridos para esta tarea:

- Las entidades (solicitante y proveedor) se conocen, o al menos una conoce a la otra.
- Las entidades (solicitante y proveedor) acuerdan, de alguna manera, la semántica y la descripción de servicio que gobierna la interacción entre los agentes.
- La semántica y la descripción son trasladadas a los agentes de ambas partes.
- Los agentes intercambian mensajes, desempeñando tareas para las entidades.

Figura 6 - Proceso general de intercambio entre *Web Services*



Fuente: W3C Working Group, ***Web Service Architecture***, p. 9

2.8. Estándares

"El *Grid Computing* utiliza estándares y protocolos abiertos, para permitir la virtualización de recursos computacionales distribuidos, con el objetivo de crear la imagen de un único sistema, a través de entornos informáticos heterogéneos y dispersos geográficamente."[46]

La clave para esta realización es la estandarización, de modo que los diversos componentes puedan ser descubiertos, asignados, monitoreados, etc., y en general que puedan ser administrados como un sistema virtual. La estandarización es crítica si se desea crear sistemas y recursos que operen entre sí, que puedan ser portados a diversas plataformas y que sean reutilizables; esto puede también contribuir al desarrollo de sistemas seguros, robustos y escalables.

Aunque existe gran cantidad de estándares relacionados con el *Grid*, aplicables a diferentes categorías como mensajería, seguridad y administración, por nombrar algunas, existen tecnologías complementarias que tienen especial relevancia con las especificaciones del *Grid*.

Figura 7 - Especificaciones *Grid* y estándares relacionados

Fuente: Jacob, Bart y otros, *On Demand Operating Environment*, p. 223

2.8.1. UML - *Unified Modeling Language*

UML es una notación estándar utilizada para el diseño, modelado de aplicaciones y modelado de sistemas de manera visual. El UML es un lenguaje y no una metodología, y como tal, es independiente de los lenguajes de programación y de las plataformas. Puede ser utilizado en conjunto con cualquier lenguaje de programación para crear el diseño de un sistema e ilustrar su estructura. Los documentos y diagramas sirven esencialmente como diseño del proyecto. Así mismo puede ser utilizado para el desarrollo de aplicaciones, modelado de negocios y otros tipos de sistemas que no estén relacionados con software. [47]

2.8.2. MDA – *Model Driven Architecture*

MDA es una especificación basada en UML. Haciendo uso de los modelos UML, MDA proporciona las herramientas para la creación de especificaciones y desarrollo de aplicaciones, separándolas de la tecnología subyacente de la plataforma utilizada. Usando MDA, los desarrolladores pueden diseñar interfaces y relaciones entre las aplicaciones, independientemente de la plataforma o el lenguaje de programación. Las aplicaciones diseñadas de esta forma pueden ser creadas en una variedad de plataformas, abiertas o propietarias. Cuando se utiliza una nueva tecnología, el modelo no debe ser repetido.

Utilizando MDA, se debe iniciar con un Modelo Independiente de la Plataforma (PIM). Este debe ser un modelo de funcionalidad y comportamiento, sin detalles de la implementación técnica. Una herramienta de MDA, es entonces utilizada para transformar el PIM a un Modelo de Plataforma Especifico (PSM). Este proceso que es parcialmente automático, genera un modelo UML.

La herramienta MDA utiliza el modelo PSM para generar el código, incluyendo interfaces, archivos de configuración y otros. Dependiendo de la complejidad del modelo o la aplicación, la herramienta genera todo o casi todo el código necesario. En este punto el código puede ser manipulado para afinar la aplicación antes de liberarla.[48]

2.8.3. XML – *Extended Markup Language*

XML es una familia de tecnologías, que utiliza un lenguaje de etiquetas, estandarizado y estructurado, para definir la representación de datos y documentos. Específicamente el XML, es un lenguaje para la creación y utilización de información estructurada, definido con una semántica comprensible tanto para humanos como para computadoras.

La tecnología fue desarrollada, hacia 1996, por el *XML Working Group*, formado bajo el auspicio del W3C, basándose en SGML[†††] [ISO 8879]. Desde la aparición de su primera definición formal, en 1998, su uso se ha extendido para adaptarse a gran cantidad de estándares, debido a su flexibilidad.

Un documento XML está compuesto por unidades de almacenamiento llamadas entidades, que contiene datos analizados o datos sin analizar. Los datos analizados están compuestos por caracteres que, en algunos casos forman los datos y en otros las etiquetas. Las etiquetas proporcionan una descripción, de la disposición de almacenamiento y de la estructura lógica.

Algunos de los beneficios inherentes al XML incluyen:

- Es fácil de usar, está basado en estándares abiertos para la descripción de datos y proporciona un punto de convergencia entre aplicaciones y componentes heterogéneos.

- Es una estructura orientada a los elementos, lo que significa que puede ser extendido con facilidad. A diferencia de los formatos propietarios, que frecuentemente están limitados a un número finito de características, la estructura de etiquetas del XML hace que la adición de nuevas etiquetas y atributos sea sencilla.

- Los documentos XML, generalmente hablando, son fáciles de leer y entender por los humanos, más que los formatos separados por comas o delimitados por otros caracteres.

[†††] *Standard Generalized Markup Language*. Es un estándar internacional para la definición de lenguajes de marcas, lo que significa que es un metalenguaje. Las marcas consisten en notaciones llamadas etiquetas, que especifican la función de una porción de texto o la forma en como esta será desplegada. SGML enfatiza las marcas descriptivas.

- El XML permite definir un lenguaje para describir la estructura particular de un documento XML, con el objetivo de evaluar su contenido en una aplicación. El estándar XML es utilizado para describir la construcción sintáctica de etiquetas que debe contener el documento y describir las reglas de validación de datos.[49]

2.8.4. XML *Schema*

El *XML Schema* es un documento XML que define: los elementos, tipos de datos, contenido y estructura para un documento XML asociado. Fue desarrollado por el W3C, como una alternativa para definir la estructura de un documento XML, utilizando sintaxis XML.

El modelo de contenido de los *XML Schema* es abierto, esto quiere decir que es posible añadir elementos y atributos secundarios, que no hubieran sido definidos en el esquema del documento. El modelo de contenido abierto proporciona flexibilidad, porque permite ampliar el vocabulario de los esquemas y seguir siendo capaz de crear documentos válidos. Esto ayuda a la reutilización del software, porque es posible usar un *XML Schema* existente, al que le podría faltarle algún elemento o atributo, para validar un nuevo documento XML.

Los *XML Schema*, permiten la herencia de elementos, consistente en asociar elementos derivados entre sí, reteniendo las relaciones existentes entre ellos. También permiten los tipos de datos restringidos, con los que se puede controlar el intervalo o el formato de los datos de un determinado elemento o atributo.

El uso de los espacios de nombres en los *XML Schema*, permite que dos elementos distintos tengan el mismo nombre (cada uno correspondiendo a su espacio de nombres). Para ello utilizan los URI (*Uniform Resource Identifiers*), que garantiza que sean únicos los recursos a los que se hace referencia.[50]

2.8.5. XSLT - *Extensible Stylesheet Language Transformation*

El XSLT es un lenguaje basado en XML utilizado para la transformación de documentos XML. El documento original no cambia, en vez de ello, se crea uno nuevo basado en el contenido del que ya existe. El nuevo documento puede ser generado por el proceso en XML estándar o en algún otro formato, como puede ser HTML o texto plano. XSLT es frecuentemente utilizado, para transformar datos entre diferentes XML *Schemas* o convertir el XML en páginas web o documentos PDF.

El XSLT, es un lenguaje declarativo, en lugar de listar una serie de acciones a desempeñar en un ambiente condicional, el XSLT consiste en una colección de reglas, cada una de las cuales especifican "que" se debe agregar al árbol resultante mientras el proceso va escaneando el árbol de origen, de acuerdo a un algoritmo fijo busca el nodo que cumpla con las condiciones dadas.

El XSLT define la transformación en término de árboles de origen y destino, para evitar el bloqueo de su implementación en sistemas específicos.[51]

2.8.6. SOAP – Simple Object Access Protocol

SOAP es un protocolo basado en XML para el intercambio de mensajes sobre una red de computadoras, normalmente utilizando HTTP. SOAP forma parte fundamental de la capa de *web services*, proporcionando una estructura básica para la mensajería, que las capas más abstractas pueden utilizar. El SOAP puede ser utilizado para crear una arquitectura orientada a los servicios.

Existen varios tipos diferentes de pautas para la mensajería en SOAP, pero la más común es el *Remote Procedure Call* (RPC), donde un nodo de la red (el cliente) envía un mensaje de petición a otro nodo (el servidor), y el servidor envía inmediatamente un mensaje de respuesta al cliente. [52]

SOAP define un "sobre", que permite el intercambio de datos formateados en XML, entre los clientes y los proveedores de servicios, sin importar la plataforma o el lenguaje de programación. SOAP es fundamentalmente un modelo de comunicación de una vía, que asegura que mensajes coherentes sean transmitidos del solicitante al receptor, potencialmente puede incluir intermediarios que pueden procesar parte del mensaje o agregar unidades al mensaje.[53]

2.8.7. WSDL – *Web Service Description Language*

WSDL es un *XML Schema* que define un marco de trabajo extensible, para la descripción de interfaces de *web services*. Inicialmente fue desarrollado por Microsoft e IBM, y posteriormente fue sometido para la aprobación del W3C por 25 compañías. WSDL es el corazón de la infraestructura de *web services*, proporcionando una forma común para la presentación de los tipos de datos enviados en los mensajes.

WSDL como el resto de tecnologías XML, es extensible y contiene varias opciones para asegurar la compatibilidad y la interoperabilidad entre diferentes implementaciones. Si el solicitante y el receptor pueden compartir y entender el mismo WSDL de la misma forma, la interoperabilidad está asegurada.[54]

2.8.8. UDDI – *Universal Description, Discovering and Integration*

UDDI es una infraestructura que define un modelo de datos, para el registro y descubrimiento de información de negocios, incluyendo los *web services* que los negocios publican. UDDI es producto de un consorcio independiente, para el desarrollo de un estándar de Internet que ayude a la descripción, registro y descubrimiento de *web services*.

UDDI, es similar al concepto del directorio de las páginas amarillas. Los negocios registran su información de contacto, incluyendo detalles tales como teléfono, fax, dirección postal y sitio web. El registro incluye información acerca de las categorías para la búsqueda, como ubicación geográfica, tipo de industria, tipo de negocio y otros. Un negocio puede también descubrir información acerca de *web services* específicos en el registro, que típicamente se encuentra en una URL que apunta a un archivo WSDL. El registro de un negocio por medio de UDDI consiste de tres componentes:[55]

- Páginas Blancas: dirección, contactos, identificadores.

- Páginas Amarillas: tipo de industria, tipo de negocio (basados en categorías estándar)

- Páginas Verdes: información técnica acerca de los servicios proporcionados.

2.8.9. WS-I *Basic Profile*

El WS-I describe la manera en que las cuatro tecnologías clave, de la especificación web services, pueden ser implementadas para alcanzar un nivel de coherente de interoperabilidad. Las cuatro especificaciones claves son: XML Schema, SOAP, WSDL y UDDI. Este no se añade a las especificaciones, sino que procura mostrar cómo pueden trabajar juntos.

2.8.10. WSS - *Web Service Security*

WSS es un protocolo de comunicación que proporciona los medios para aplicar la seguridad de los web services. Desarrollado inicialmente por IBM. Microsoft y VeriSign, fue liberado el 19 de abril del 2004 como un estándar.

El protocolo contiene especificaciones sobre como la integridad y confidencialidad, pueden imponerse en la mensajería de los *web services*. WSS, incorpora en el encabezado de los mensajes SOAP, características de seguridad para que estas puedan ser utilizadas en la capa de aplicación.[56]

2.8.11. CIM - *Common Information Model*

CIM, es una arquitectura para la administración de sistemas de información, que intenta unificar la administración de entornos empresariales. CIM es un estándar de la DMTF[‡‡‡] para dar a conocer datos acerca de sistemas, aplicaciones, redes y dispositivos; además permite el intercambio de información, de los elementos administrados, entre las partes interesadas permitiendo que las aplicaciones de administración tengan acceso a los datos, control de dispositivos o a los sistemas sin importar la plataforma, facilitando así la interoperabilidad.

CIM proporciona clases de objetos, propiedades, métodos y asociaciones comunes para las aplicaciones de administración, en forma de un esquema de administración organizado en tres capas:

- Un modelo central, que gestiona los elementos que abarcan todas áreas de la administración.

- Modelos comunes, gestiona los elementos que se encuentran en áreas específicas de administración, tales como sistemas, aplicaciones, redes, y dispositivos, independiente de las tecnologías o implementaciones.

[‡‡‡] *Distributed Management Task Force.* Es una organización dedicada al desarrollo y mantenimiento de estándares, relacionados con la administración de sistemas y entornos de tecnológicos de empresas e Internet.

- Esquemas de extensión, gestiona todo lo relacionado con tecnologías específicas (sistemas operativos o plataformas).

CIM es independiente del método utilizado para la implementación.[57]

2.8.12. WBEM - *Web Based Enterprise Management*

WBEM, es una iniciativa que busca diseñar estándares para la administración de entornos computacionales. La meta es proveer estándares, para la industria, que permitan la creación de herramientas, basadas en estándares, para la integración en el uso de las tecnologías web. Esta iniciativa es un esfuerzo del DMTF y tiene como principal elemento el CIM.[58]

3. FISIOLOGÍA DEL *GRID*

"El *Grid computing* es visto, cada vez más, como la siguiente fase de la computación distribuida. Construido sobre estándares sólidos de Internet, el *Grid computing* permite a las organizaciones compartir recursos computacionales y de información, traspasando los límites de los departamentos y los de la organización, de una forma segura y altamente eficiente. Las organizaciones alrededor del mundo, están utilizando hoy *Grid computing* en diversas áreas como la investigación científica, descubrimiento de medicamentos, análisis de riesgo financiero y diseño de productos. El *Grid computing* permite que las organizaciones orientadas a la investigación, puedan solucionar problemas, que anteriormente, no eran viables de resolver debido a las limitaciones informáticas y de integración de recursos. El *Grid* también reduce costos, a través de la automatización y de una mejora en la utilización de los recursos de IT. Finalmente el *Grid computing*, puede incrementar la agilidad de la organización, permitiendo procesos de negocios más eficientes y mayor interés por el cambio. Con el tiempo el *Grid computing* permitirá una mayor flexibilidad, eficiencia y una utilidad parecida a la de una infraestructura global de computación. La clave para lograr los beneficios del *Grid computing* es la estandarización, para que los diversos recursos, que integran el moderno entorno computacional, puedan ser descubiertos, accedidos, alojados, monitoreados y en general administrados como un simple sistema virtual - incluso cuando sean proporcionados por diferentes proveedores o/y operados por diferentes organizaciones. La estandarización del *Grid computing* es liderada por el *Global Grid Forum* (GGF). El GGF ha desarrollado el *Open Grid Service Architecture* (OGSA) y está trabajando junto con la industria para defender esta arquitectura y las especificaciones asociadas que permitan la adopción del *Grid* en el mundo de los negocios y la investigación."[59]

En el presente capítulo se presenta la descripción general del modelo OGSA, como una arquitectura que, construida sobre las bases del *Grid* y los *web services*, busca definir facilidades y mecanismos estándar que permitan la integración de las organizaciones virtuales.

3.1. OGSA - *Open Grid Service Architecture*

La clave para la realización del *Grid* es la estandarización, la cual tiene como objetivo que los diversos recursos que integran el moderno entorno computacional, puedan ser descubiertos, accedidos, alojados, monitoreados y en general administrados como un solo sistema virtual. La estandarización es crítica, si se desea crear componentes y sistemas que sean reutilizables, portables y que puedan interactuar entre si. Esta estandarización puede, al mismo tiempo, contribuir al desarrollo de sistemas *Grid* que sean: seguros, robustos y escalables, facilitando el uso de buenas prácticas.

OGSA es una arquitectura orientada a los servicios, que dirige los esfuerzos encaminados a la estandarización del *Grid*, definiendo las principales capacidades y conductas que conciernen a este tipo de sistemas. Entre los aspectos importantes que conciernen a los sistemas *Grid* están: establecer la identidad de un participante, negociar la autenticación, definición de políticas, descubrimiento de servicios, determinar los niveles de acceso, administración de las afiliaciones, comunicación con la organización virtual, organización jerárquica de servicios, entrega fiable y escalable de servicios, integración de recursos, monitoreo y administración de los servicios.[60]

Resumiendo, puede decirse que OGSA es una arquitectura computacional para la interacción de recursos distribuidos, que está orientada a los servicios, asegurando la interoperabilidad sobre sistemas heterogéneos, para que, recursos de diferentes tipos puedan comunicarse y compartir información.

3.2. Orientación a los servicios

El enfoque orientado a los servicios, consiste en adoptar una representación generalizada de los recursos informáticos, donde todos pueden ser tratados como servicios. Para OGSA un servicio se define como: cualquier entidad del entorno de red que tenga alguna capacidad para el intercambio de mensajes (procesamiento, almacenamiento, redes, programas, bases de datos, etc.)

Los mecanismos de interoperabilidad son críticos en un entorno G*rid*, por lo que, desde el punto de vista de la orientación a los servicios, este se divide en dos sub problemas: la definición de las interfaces de servicio y la identificación de protocolos que puedan invocar una interfaz en particular.

La orientación a los servicios establece la necesidad de contar con mecanismos para la definición de interfaces estándar, transparencia local y remota, adaptación a servicios del sistema operativo y semánticas de servicio uniforme. Así mismo, busca simplificar la virtualización a través del encapsulamiento de diversas implementaciones, detrás de una interfaz general.[61]

3.3. Virtualización

La virtualización permite acceso coherente a los recursos, entre plataformas heterogéneas; el mapeo de múltiples instancias, de un recurso lógico, sobre el mismo recurso físico y facilita la administración de los recursos de una organización virtual, basándose en la integración de recursos de bajo nivel.

Así mismo, la virtualización permite la integración de servicios, para formar servicios más sofisticados, sin dar importancia a como esta implementada esta integración. Y permite también el mapeo transparente de semánticas comunes de comportamiento, sobre las facilidades de la plataforma.

En la virtualización, las funciones de los servicios deben ser expresadas de una forma estándar, de manera que cualquier implementación del servicio sea invocada de la misma manera; para lo cual se hace uso del WSDL. WSDL distingue entre la definición de la interfaz de servicio y el protocolo asociado que se utiliza para la invocación; una sola interfaz puede tener múltiples asociaciones, que incluyen protocolos de comunicación distribuida, como el HTTP, así como asociaciones locales, tales como el IPC local, para interacciones sobre el mismo servidor.

La definición de la interfaz de servicio y el acceso asociado, son distintas de la implementación de los mismos, ya que un servicio puede soportar múltiples implementaciones en diferentes plataformas. Esto permite aprovechar, no solamente la posibilidad de utilizar las facilidades nativas de la plataforma, sino también, agrupar las implementaciones de varios servicios, para la creación de recursos virtuales.

Dependiendo de la plataforma y el contexto, las implementaciones pueden seguir las siguientes tendencias:

- Construir una implementación de referencia con portabilidad completa a múltiples plataformas.
- Utilizar las facilidades nativas que la plataforma posee, para hacer entrega de la funcionalidad que debe proporcionar el servicio.
- Aplicar ambos mecanismos de forma recursiva, para la construcción de servicios de alto nivel, integrando varios servicios de bajo nivel.

En general la arquitectura de servicios soporta transparencia local y remota, con respecto a la localización de los servicios y su implementación. Provee la asociación con múltiples protocolos, permitiendo optimizar la invocación de los servicios, cuando estos se encuentran en la misma máquina que el proceso solicitante; así como la negociación entre protocolos, a través de los flujos de la red, que se encuentran optimizados para diferentes propósitos. Adicionalmente, una implementación, de un servicio en particular, puede utilizar funciones y capacidades nativas de la plataforma, que no sean distribuidas.[62]

3.4. Semántica de Servicio – *Grid Service*

La virtualización y creación de servicios dependen no solamente de la definición de interfaces estándar, también requiere de semánticas estándar que definan la interacción de los servicios. Con este fin, OGSA define lo que se llama *Grid Service*, que consiste en un *web service* que provee un conjunto de interfaces bien definidas, así como convenciones específicas que definen su comportamiento. Las interfaces permiten realizar tareas tales como: descubrimiento, creación dinámica de servicios, administración del ciclo de vida, notificación, administración de dispositivos, actualizaciones y reconocimiento de dispositivos por nombre; así como también el control de la autorización, la concurrencia y tareas definidas por los usuarios. La definición de un conjunto de interfaces consistentes, para la implementación de los *Grid services*, permite la construcción de una jerarquía de servicios de alto nivel, que puede ser utilizada de manera uniforme entre las diferentes capas de abstracción.

Las interfaces y convenciones que define al *Grid service*, cobran mayor importancia cuando están relacionados con la administración del comportamiento de servicios transitorios; ya que las organizaciones virtuales, generalmente, mantienen un conjunto de servicios, que no son necesariamente estáticos, para manejar las solicitudes de actividades complejas de los clientes.

La creación de servicios transitorios de manera dinámica, se encuentra asociada a un estado específico de una actividad que ha sido solicitada, permitiendo manejar tareas de administración e interacción para ese estado en particular. Cuando el estado de la actividad no es utilizado por algún tiempo, el servicio puede ser destruido.

Para comprender mejor la estructura y funcionamiento del *Grid service,* existen algunos conceptos importantes que es necesario conocer. Y estos se detallan a continuación:[63]

- Las interfaces (conocidas en términos de WSDL como: *portTypes*), permiten definir el *Grid service*, así como las interacciones que cada uno tiene consigo mismo y con otras interfaces. Las interfaces pueden hacer referencia a aplicaciones específicas creadas por los usuarios o pueden ser de carácter general como las proporcionadas por OGSA.

- OGSA define un conjunto de interfaces estándar que permiten realizar tareas tales como la administración, el descubrimiento de servicios, la notificación, la autenticación, el manejo del ciclo de vida, la creación de servicios, el registro de servicios, el monitoreo, la concurrencia y algunos otros aspectos relacionados con los servicios en general.

- La colección de interfaces con las que puede interactuar un *Grid service*, se definen en un WSDL adjunto al que se le conoce como *serviceType*. Este, además, proporciona información adicional relacionada con las versiones.

- Asociado a cada interfaz, existe un conjunto dinámico de elementos, que proporcionan una representación estándar de la información relacionada a cada instancia de un servicio, a esta representación se le conoce como *service data element*. Esta característica del modelo OGSA, es de gran importancia, debido a que provee las bases para el descubrimiento y la administración de los servicios dinámicos.

- Un *Grid service* puede mantener de forma interna un registro de su ciclo de vida. La existencia de estados, permite distinguir una instancia de un servicio de alguna otra proporcionada por la misma interfaz. El termino *Grid service instance*, se refiere a una instancia en particular, de un servicio.

- Debido a la naturaleza dinámica del *Grid service*, es necesario distinguir una instancia de un servicio de otra, para lo cual cada *Grid servicie* recibe un único nombre, al que se le conoce como *Grid service handle*. Este nombre permite distinguir entre cada instancia de un servicio que haya existido anteriormente, que exista actualmente o que pueda existir en el futuro.

3.5. Interfaces Estándar

OGSA define una serie de comportamientos estándar para la administración de los servicios, y define la interfaz asociada a cada uno de estos comportamientos.[64]

- Descubrimiento: Las aplicaciones requieren de mecanismos que le permitan descubrir los servicios disponibles, determinar sus características y configurarse a sí mismas para utilizar estos servicios; para lo cual se define una operación estándar denominada *FindServiceData*. Además, se requiere de operaciones para registrar las instancias de un servicio (*Registry*), y para el mapeo de las referencias hacia los manejadores de servicios (*HandleMap*).

- Creación dinámica de servicios: Una de las características básicas del modelo OGSA, es que debe contar con mecanismos que permitan crear y administrar instancias de nuevos servicios. Para lo cual se define una interfaz estándar denominada *Factory*.

- Administración del ciclo de vida: Los servicios pueden ser creados y destruidos de forma dinámica, por lo que deben ser destruidos de manera explícita. Aunque también existe la posibilidad de que sean destruidos por una falla en el sistema, tal como una falla en el sistema operativo o la red. Para garantizar este control se definen interfaces que controlan el ciclo de vida de los servicios y los estados asociados a estos. La operación estándar denominada *SetTerminationTime*, es la encargada de lograr este propósito utilizando una administración del ciclo de vida basada en *soft-state protocol*.[§§§]

- Notificación: Un conjunto de servicios, distribuidos y dinámicos, debe ser capaz de notificarse entre ellos, cambios significativos en sus estados. OGSA define las abstracciones comunes y las interfaces para la suscripción y entrega de tales notificaciones. Para lo que se define la interfaz denominada *Notifications*.

- Administración: Operacionalmente, se debe contar con mecanismos que permitan monitorear y administrar conjuntos potencialmente grandes de instancias de servicios.

3.6. Ambiente Anfitrión – *Hosting Environment*

OGSA se encarga de definir la semántica de las instancias de un servicio, definiendo como debe ser creada y nombrada, como determinar su ciclo de vida, el protocolo de comunicación seleccionado, etc. Sin embargo, solamente se encarga de definir el comportamiento básico de los servicios, pero no coloca restricciones sobre el que debe hacer un servicio y como debe hacerlo. Por lo mismo, no se encarga de aspectos tales como el modelo de programación para la implementación, el lenguaje de programación, entorno de ejecución, etc.

[§§§] El termino *soft-state protocol*, esta asociado a la política de desechar el estado de un proceso, establecido en una ubicación remota, a menos que este se actualice de manera periódica.

En la práctica la creación de instancias de un *Grid Service*, se realiza sobre un ambiente de ejecución o ambiente anfitrión. El ambiente anfitrión define no solamente el modelo de programación para la implementación, el lenguaje de programación, herramientas de desarrollo, etc., sino también proporciona herramientas para que los servicios puedan cumplir con la semántica del *Grid*. Existen dos tipos generales para el ambiente anfitrión:[65]

- Los servicios pueden depender de procesos nativos del sistema operativo, quedando limitados a las funcionalidades que este le provea, por lo que algunas funciones deberán ser implementadas directamente en la aplicación.

- Los servicios pueden depender de contenedores o componentes más sofisticados como J2EE, .Net, Websphere o Sun ONE, que proporcionan un ambiente de trabajo, en el cual se pueden construir y crear instancias, para aplicaciones más complejas.

Como se ha mencionado anteriormente, OGSA define las interacciones entre servicios, independientemente del ambiente anfitrión. Para lograr esta interacción, proporciona una base de características, que todos los ambientes deben poseer, para facilitar la implementación exitosa de un *Grid Service*. Básicamente un ambiente anfitrión debe poseer lo siguiente:[66]

- Mapeo de una amplia variedad de nombres o manejadores de servicios del *Grid*, dentro entidades especificas de la implementación, tales como punteros de lenguaje C o referencias a objetos en Java.

- Despacho de eventos de notificación y llamadas del *Grid*, dentro de acciones específicas de la implementación, tales como eventos y llamadas a procedimientos.

- Protocolos de procesamiento y formateo de datos para la transmisión por la red.

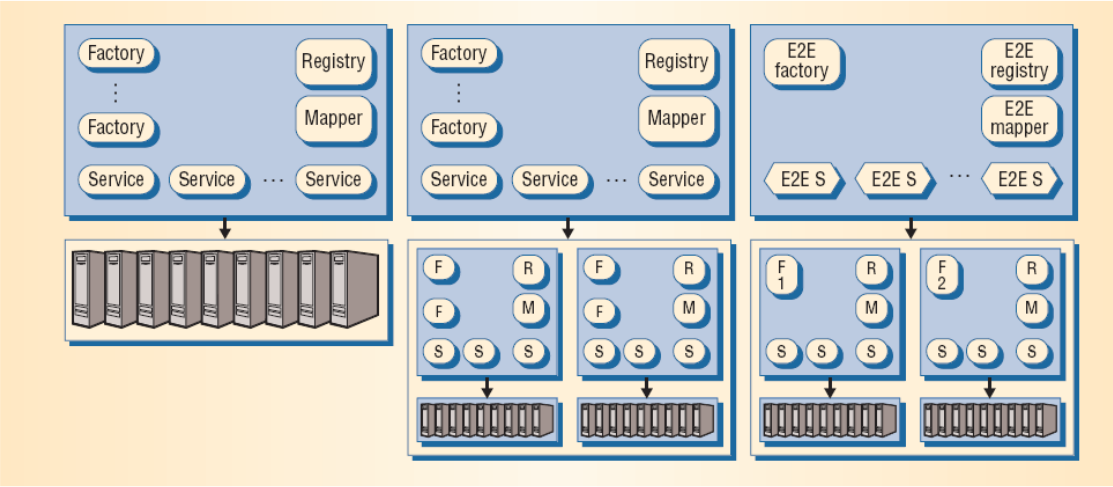
- Administración del ciclo de vida de las instancias de un servicio.

- Autenticación entre servicios.

3.7. Construcción de organizaciones virtuales

Las aplicaciones y usuarios deben tener la capacidad de crear servicios transitorios, descubrir servicios existentes y determinar las propiedades de estos servicios. Así mismo, las interfaces, descritas con anterioridad, proporcionan la base para la creación de instancias de servicio, asociadas a las Organizaciones Virtuales, con las cuales se pueden construir una variedad de estructuras de servicios, como se describe a continuación:[67]

Figura 8 – Estructura de servicios para las Organizaciones Virtuales



Fuente: Foster, Ian y otros, *Grid Service for Distributed System Integration*, p. 44

- **Ambiente anfitrión simple**. Es un conjunto de recursos localizados dentro de un dominio de administración simple y que se apoya en facilidades nativas para la gestión de servicios, por ejemplo: un servidor de aplicaciones J2EE, el sistema .NET de Microsoft o un cluster de Linux. El ambiente se encuentra estructurado, típicamente, por un elemento de tipo *Registry*, uno o más elementos de tipo *Factory* y un elemento de tipo *HandleMap*. Cada elemento Factory es almacenado en un registro, para permitir que los clientes puedan descubrir los servicios disponibles. Cuando este elemento recibe una solicitud, por parte de un cliente, utiliza las capacidades específicas del ambiente para crear una nueva instancia del servicio, asigna un manejador, registra la instancia dentro del elemento *Registry* y coloca el manejador en estado disponible para el elemento *HandleMap*. La implementación de estos servicios hace referencia a operaciones locales.

- **Ambiente anfitrión virtual**. En ambientes más complejos, los recursos asociados a la OV, pueden abarcar elementos heterogéneos, geográficamente distribuidos. Sin embargo, este ambiente virtual es accesible al usuario utilizando las mismas interfaces descritas anteriormente. Se crea uno o más elementos de tipo Factory, en los niveles más altos de la jerarquía, que se encargan de delegar la atención de solicitudes a los niveles más bajos. Similarmente, se crean elementos de tipo Registry, en los niveles más altos de la jerarquía, los cuales tienen conocimiento acerca de los elementos de tipo Factory, que se encuentran en su mismo nivel, las instancias de servicio que estos han creado, e información acerca de las políticas que rigen el uso de los servicios en la organización. Los clientes pueden utilizar los elementos Registry para localizar elementos de tipo Factory o instancias de servicio asociadas con la organización, estos procedimientos retornan un manejador, el cual puede ser utilizado para interactuar directamente con el elemento solicitado. Es de hacer notar que en el caso descrito anteriormente, se utilizan nombres globales que son únicos, para facilitar su administración.

- **Servicios de cooperación**. Esta estructura permite construir un ambiente anfitrión virtual, que provee a los participantes servicios más sofisticados para trabajos de cooperación o punto a punto. En este caso, el elemento de tipo Registry se encarga de monitorear y descubrir elementos, de tipo Factory, que hayan creado instancias de servicio en los niveles más altos de la jerarquía. Tales instancias, están implementadas para la creación de múltiples instancias de servicio, en los niveles más bajos, y para integrar el comportamiento de estas instancias como un único servicio.

Los tres casos descritos anteriormente, ilustran como la utilización de mecanismos basados en servicios *Grid*, permiten la integración de recursos distribuidos, ya sea traspasando los límites de la organización o aplicándolo a una infraestructura local.

4. PROPUESTA

"La Universidad de San Carlos de Guatemala, es una institución estatal, cuya misión es la de dirigir, organizar y desarrollar la educación superior, así como la difusión de la cultura en todas sus manifestaciones. Además de promover por todos los medios a su alcance, la investigación en todas las esferas del saber del ser humano, y cooperar con el estudio y solución de los problemas nacionales."

"La Universidad de San Carlos de Guatemala es la institución de educación superior estatal, autónoma, con una cultura democrática, con enfoque multi e intercultural, vinculada y comprometida con el desarrollo científico, social y humanista, con una gestión actualizada, dinámica y efectiva y con recursos óptimamente utilizados para alcanzar sus fines y objetivos, formadora de profesionales con principios éticos y excelencia académica."[68]

En los párrafos anteriores se hace referencia a la misión y visión de la Universidad de San Carlos de Guatemala, dentro de las cuales se encuentra plasmado el compromiso de la Universidad con el desarrollo de la investigación y el compromiso de utilizar de manera óptima los recursos a su alcance, para el cumplimiento de sus objetivos. Tomando como base estos preceptos, en este capítulo, se presenta una propuesta para la implementación de un *Grid*, como una herramienta[****], que integre los recursos computacionales de la Universidad, de manera que estos puedan ser utilizados en el desarrollo de tareas científicas, que requieran alto desempeño para el procesamiento de datos; así como en tareas de otras áreas de la investigación, que requieran almacenamiento masivo de información.

[****]La propuesta está orientada a que el *Grid* sea utilizado como un medio, para ayudar al desarrollo de la investigación en la Universidad, y no como un fin, por lo que a lo largo de la propuesta no se hace referencia a ninguna tarea que haga utilización inmediata del mismo.

4.1. Descripción general

Las actividades de diseño e implementación del *Grid*, toman como base el Campus Central de la Universidad de San Carlos de Guatemala, la infraestructura de red existente, denominada Red de Servicios Integrados; así como el supuesto de que gran parte del equipo de cómputo del Campus estará conectado a dicha infraestructura de red.

La creación del *Grid*, se hará utilizando *Globus Toolkit*, que es un conjunto de herramientas de software para la construcción de sistemas utilizando la tecnología *Grid*, el cual implementa librerías para el monitoreo, descubrimiento y administración de recursos, así como seguridad y administración de archivos. Además, es una herramienta que está basada completamente en la filosofía de *Grid Computing*, provee los servicios básicos para la construcción de la infraestructura y la puesta en marcha de las aplicaciones; descansa sobre la base del código abierto, protocolos estándares y proporciona interfaces de programación. *Globus Toolkit*, proporciona los siguientes servicios básicos:[69]

- Permite registrarse, proporciona autenticación, delegación a través de certificados y otras interfaces de seguridad, mediante la utilización de WS-Security, MyProxy y *Simplified Certification Authority*.

- Permite realizar la asignación de recursos, monitoreo y administración de los trabajos, mediante la utilización de *Grid Resource Allocation Manager* (GRAM).

- Permite preparar los archivos de entrada y salida, así como su transferencia confiable, mediante la utilización de *Reliable File Transfer* (RFT).

- Permite consultas acerca de las características de los recursos, mediante la utilización de *MDS-Index*.

- Permite realizar ciertas tareas definidas por los usuarios, siempre que la información recolectada cumpla con los criterios definidos por el usuario, mediante la utilización de *MDS-Trigger*.

- Permite almacenar la información de los recursos en una base de datos persistente, la cual puede ser consultada posteriormente por los usuarios, mediante la utilización de *MDS-Archive*.

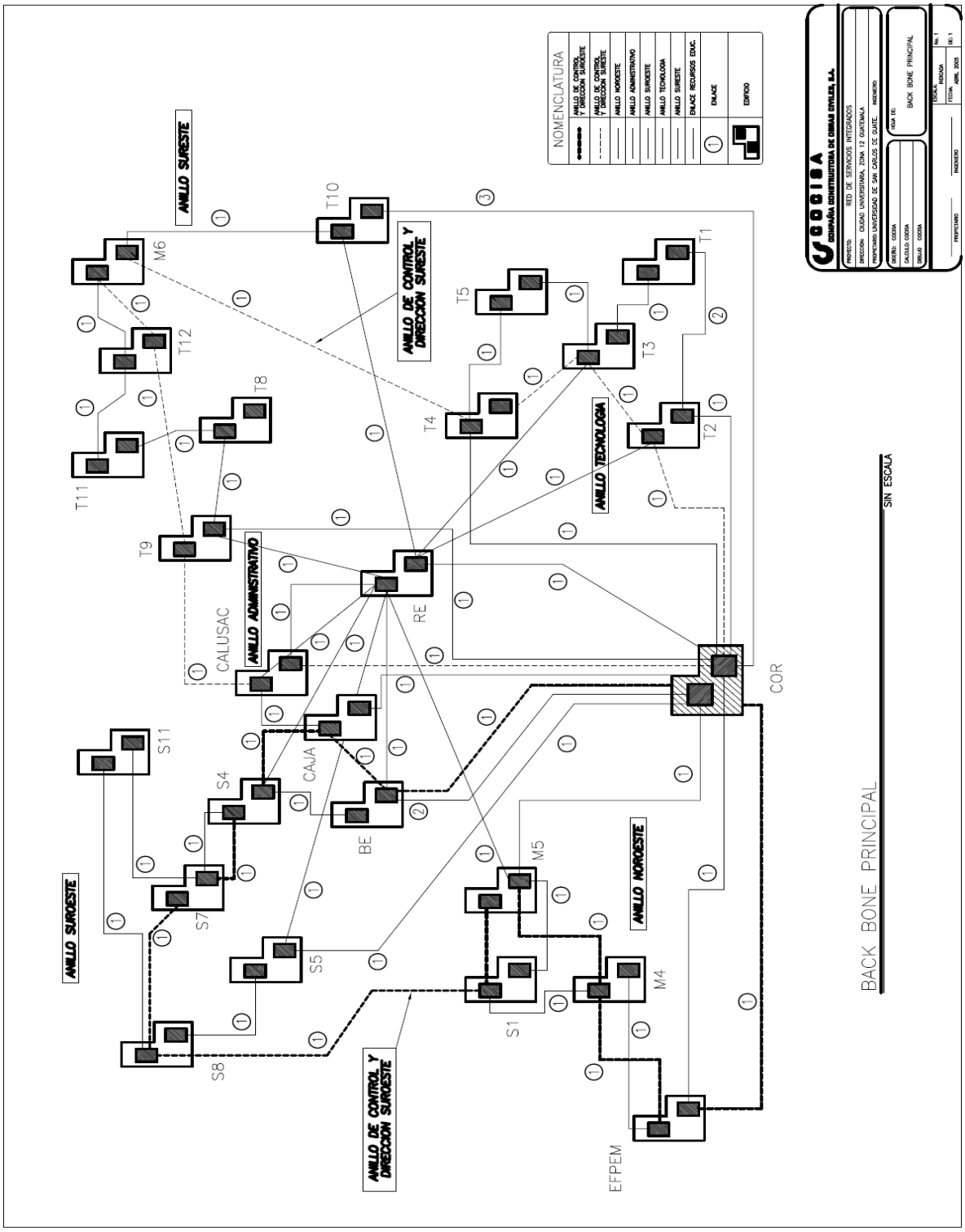
- Proporciona herramientas para tareas relacionadas con la localización, transferencia y administración de datos distribuidos, tanto para datos estructurados como para datos parcialmente estructurados, mediante la utilización de OGSA-DAI.

4.2. Infraestructura

La Universidad de San Carlos de Guatemala, cuenta con una infraestructura de red, denominada "Red de Servicios Integrados", que proporciona conectividad entre los edificios, de las diferentes unidades académicas y administrativas del Campus Universitario, a través de una conexión de fibra óptica a 100Mbps.

La Red de Servicios Integrados, está conformada por siete áreas, denominadas anillos, existen cuatro de estas áreas que agrupan a un conjunto de edificios, de las diferentes unidades académicas, de acuerdo a su posicionamiento geográfico dentro del Campus (anillo suroeste, anillo noroeste, anillo sureste, anillo tecnológico), un área agrupa a los edificios administrativos (anillo administrativo) y dos áreas denominadas anillos de control. Todas las áreas convergen en el edificio de Recursos Educativos, desde donde se administra la interconexión entre las áreas.

Figura 9 – Red de Servicios Integrados



Fuente: División de Servicios Generales, Universidad de San Carlos de Guatemala

Además de la instalación de la fibra óptica, cada uno de los edificios cuenta con equipo de red para la distribución de la señal en cada una de las aulas, oficinas administrativas y en puntos estratégicos dentro del edificio. De manera que cada edificio además de pertenecer a la red de la Universidad, posee una estructura de red independiente.

Actualmente dentro del campus están instalados 11,684 puntos de red, distribuidos de la siguiente forma:

Tabla II - Distribución de puntos de red

EDIFICIO	PUNTOS	EDIFICIO	PUNTOS
Bienestar Estudiantil	868	T2	592
Rectoría	892	T3	363
EFPEM	82	T4	408
M4	491	T5	299
M5	105	T11	273
M6	414	T12	600
S1	376	T8	478
S7	510	T9	306
S8	497	CUM C	651
S5	346	CUM A	375
S11	453	CUM B	181
Caja	148	CALUSAC	265
Recursos Educativos	601	S4	355
T1	454	T10	301

Fuente: División de Servicios Generales, Universidad de San Carlos de Guatemala

4.3. Diseño

La estructura del *Grid*, ha sido diseñada de manera que puedan observarse los diferentes niveles de abstracción que deberán ser considerados al momento de la implementación. El diseño reposa sobra la anteriormente mencionada infraestructura de red, denominada Red de Servicios Integrados, y proporciona una idea general de la interrelación que existe entre los diferentes entes que estarán involucrados en el *Grid*.

En un principio se han creado cuatro niveles de abstracción, que van desde la relación del *Grid* con el mundo exterior, hasta la presentación de los elementos individuales que lo conforman. Estos cuatro niveles representan la visión del *Grid* actualmente, sin embargo, este esquema podría ampliarse de acuerdo a las necesidades que se presenten en el futuro. A continuación se presentan los cuatro niveles del *Grid*, acompañados de una breve explicación:

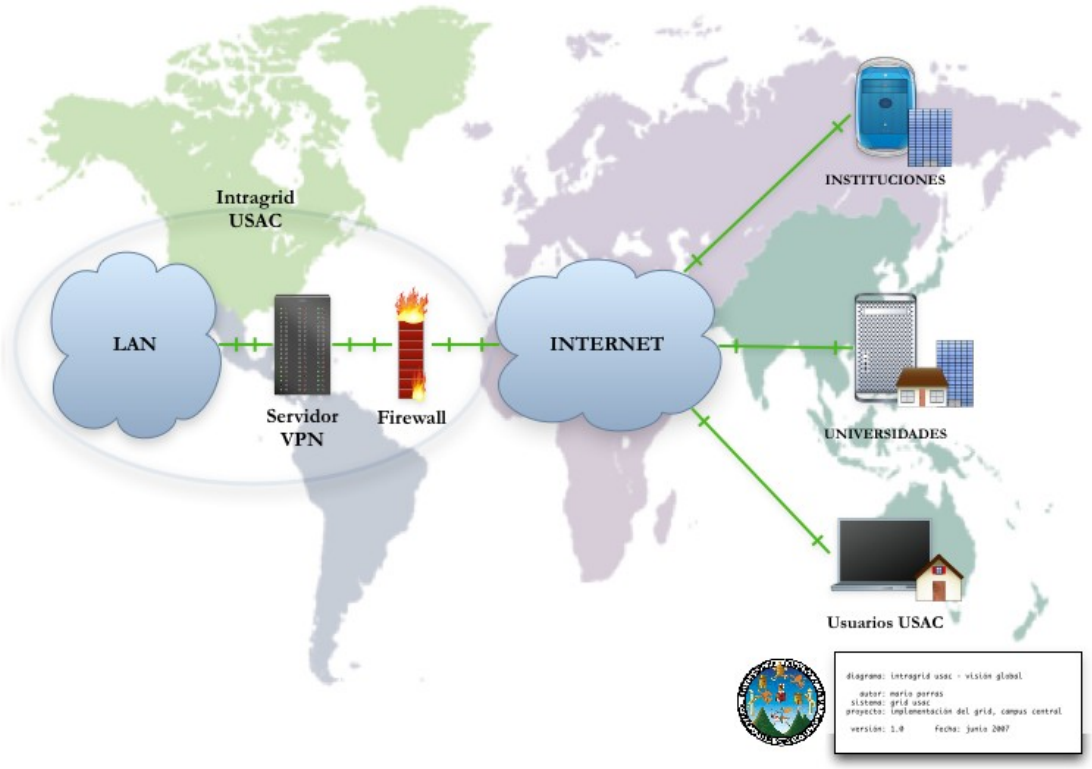
- **Nivel 1.** El primer nivel representa la relación del *Grid* con el mundo exterior.

  El *Grid* de la Universidad, denominado "**Intragrid USAC**", permite relacionarse con entidades externas a través de la utilización del Internet. Estas entidades externas pueden estar representadas por instituciones estatales o privadas, universidades locales o extranjeras, y por usuarios individuales de la Universidad que se encuentren fuera del campus.

  El propósito principal para esta interconexión, es el de proveer los servicios del *Grid* a estas entidades externas, de manera que puedan aprovechar las capacidades de procesamiento y almacenamiento, para el desarrollo de proyectos conjuntos de investigación.

  El Intragrid USAC debe estar protegido de ataques externos por un *firewall*, sea este de hardware o software, además detrás del mismo debe existir un servidor VPN, el cual será el encargado de proporcionar los accesos a la red local. Una vez dentro de la red, la interacción de los usuarios con el *Grid* estará regida por las políticas del mismo.

Figura 10 – Relación del *Grid* con entidades externas

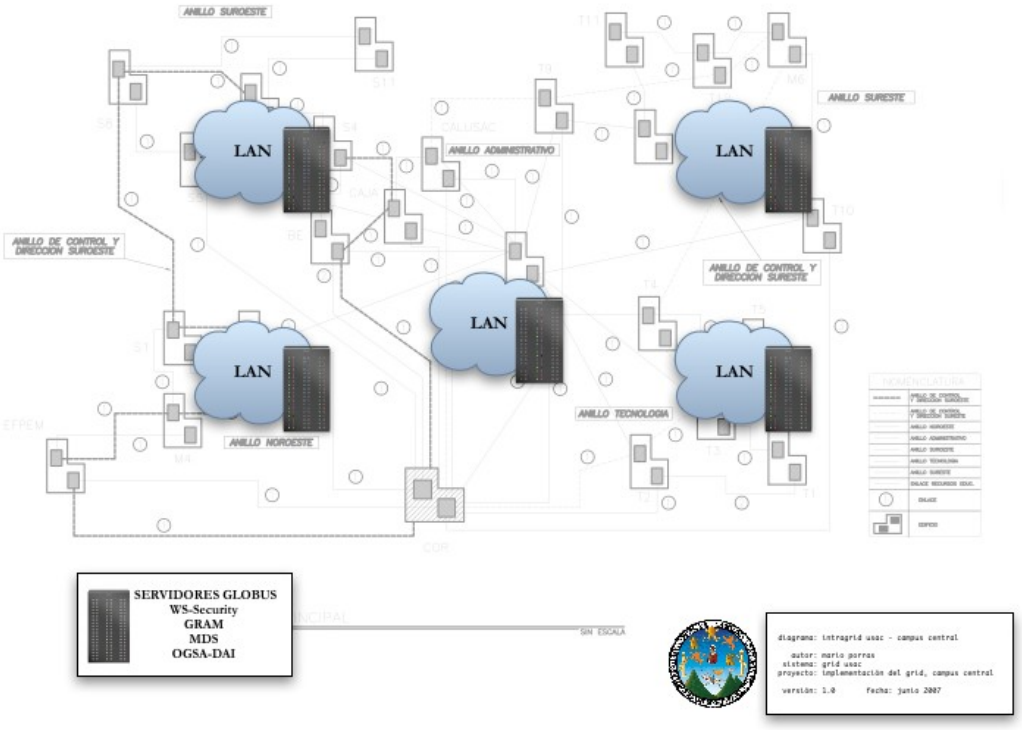


- **Nivel 2.** El segundo nivel representa la relación del *Grid* dentro de la Universidad.

  El *Grid* de la Universidad, denominado "**Intragrid USAC**", aprovecha las facilidades proporcionadas por la infraestructura de red, para crear su propia estructura jerárquica. El Intragrid, estará conformado por cinco *Grid* internos, uno por cada uno de los anillos principales (anillo suroeste, anillo noroeste, anillo sureste, anillo tecnológico, anillo administrativo), cada uno con sus propias políticas y estructura interna. Adicionalmente, los cinco, estarán integrados para la creación de un *Grid* global, desde el cual se tendrá el control de todos los *grids*, y que poseerá su propia estructura y sus propias políticas.

De esta cuenta la distribución de los anillos proporcionara autonomía a cada conjunto de unidades académicas, para la ejecución de sus proyectos. De manera que los proyectos que provengan del mismo anillo, tendrán más prioridad que los que provengan de anillos exteriores, a menos que la prioridad de los proyectos globales demande la utilización de estos recursos.

Figura 11 – Intragrid USAC



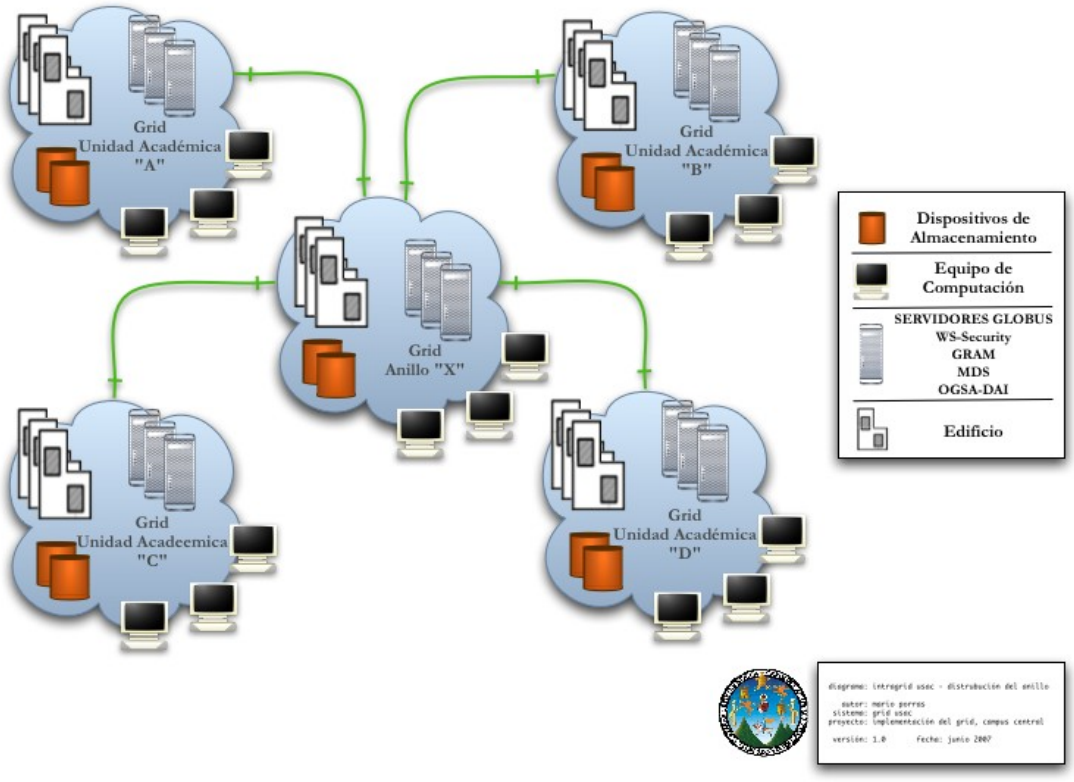
- **Nivel 3.** El tercer nivel representa la relación del *Grid* dentro de un anillo.

  Cada uno de los anillos posee un *Grid* principal, que es el encargado de la administración de los recursos de todas las unidades académicas asociadas al anillo. Sin embargo, si una unidad académica lo requiere, podrá tener su propio *Grid* de tal forma que pueda administrar sus recursos.

De esta cuenta la distribución, por unidades académicas, proporcionará autonomía para la ejecución de proyectos propios de la unidad. De manera que, los proyectos que provengan de una misma unidad académica, tendrán más prioridad que los que provengan del anillo o de anillos exteriores, a menos que la prioridad de los proyectos globales demande la utilización de estos recursos.

A este nivel es posible identificar los principales componentes del *Grid*, así como la relación que existe entre cada uno de ellos. Ya que, como se puede observar, es el *Grid* del anillo quien se encarga de realizar la interacción entre los recursos de las diferentes unidades académicas y los solicitantes.

Figura 12– *Grid* de un anillo

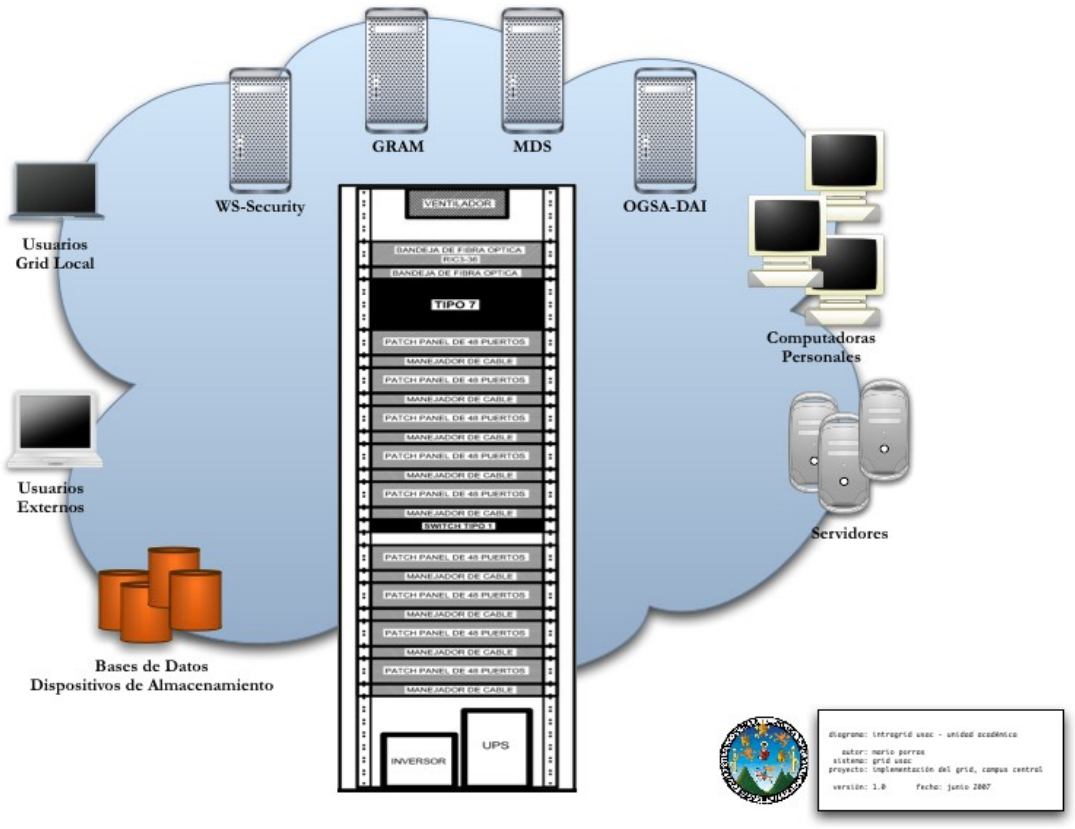


- **Nivel 4.** El cuarto nivel representa el *Grid* y sus componentes.

  El nivel más bajo de la jerarquía está representado por una entidad específica: el *Grid*. En este nivel de abstracción es posible identificar los diferentes componentes que integran el *Grid*, inicialmente se pueden encontrar los servidores de Globus, que conforman la parte administrativa del *Grid* (WS-Security, GRAM. MDS, OGSA-DAI).

  Posteriormente puede observarse el equipo de red, de un edificio en particular, el equipo de cómputo, los servidores y los dispositivos de almacenamiento, que representan el conjunto de recursos a administrar. Y finalmente están los usuarios tanto internos como externos, que interactuaran con el *Grid*.

Figura 13 – *Grid*



La implementación de estos cuatro niveles de abstracción, dependen no solamente de la parte lógica, descrita anteriormente, sino también de una parte física, ya que como se puede observar en los diferentes diagramas, las agrupaciones creadas para cada nivel, requieren de equipo de cómputo especializado para la administración del *Grid*, una infraestructura de red, que sirva como transporte, y un conjunto de recursos a ser administrados.

En los cuatro niveles descritos anteriormente, se refleja el alcance de la implementación del *Grid* en el Campus Central de la Universidad.

4.4. Arquitectura

En esta sección se describen los conceptos básicos a utilizar en la creación del *Grid*, información acerca de los usuarios y recursos, así como la arquitectura, componentes y servicios proporcionados.

4.4.1. Usuarios y recursos

La creación de los usuarios estará a cargo de un usuario especial, al que se denominará líder de proyectos, quien posee la autoridad de crear y remover usuarios, así como asignarle el acceso a los recursos.

Inicialmente deben existir dos usuarios predefinidos: el administrador, quien tendrá la responsabilidad de todo lo relacionado con la administración del *Grid*; y el invitado, que servirá para realizar las pruebas de los nuevos recursos instalados. El usuario invitado, también puede ser utilizado para experimentar con los servicios del *Grid*, sin necesidad de crear un nuevo usuario.

Un recurso, es cualquier computadora o dispositivo de almacenamiento que este registrado en el *Grid*. Cada uno de los recursos debe poseer un propietario, quien debe tener privilegios de administrador sobre ese recurso, además de ser el responsable por la instalación y configuración del dispositivo dentro del *Grid*. Se denominará *pool*, a un conjunto de recursos dispuestos geográficamente de manera arbitraria. Existen dos tipos de *pool*, que son los privados y los públicos. Donde el administrador de un *pool* privado, es propietario de todos los recursos del *pool* y tiene autoridad para remover y agregar recursos al *pool*; mientras el administrador del *pool* público, puede asignar a otros la autoridad para agregar y remover recursos del *pool*.

Inicialmente, se tendrá una política de autorización que permita a todos los usuarios, del *Grid*, sin importar a que proyecto pertenezcan, la posibilidad de tener acceso en cualquier momento, a cualquier recurso. Sin embargo, esta política puede ser modificada por el propietario del recurso. Por ejemplo, pueden hacerse restricciones por ventanas de tiempo o restringiendo su uso, solamente para ciertos miembros o grupos.[70]

4.4.2. Componentes

Un *pool* de recursos será denominado homogéneo, sí todos los recursos que lo conforman son del mismo tipo, y será denominado heterogéneo, si por lo menos uno de los recursos que lo conforman es de un tipo diferente. El *Grid* estará conformado por un conjunto de *pools*, cada uno integrado por un servidor MDS4 y un grupo de recursos de computadoras y dispositivos de almacenamiento. Adicionalmente, existen estaciones de trabajo externas que pueden hacer uso del *Grid*, sin pertenecer a ninguno de estos *pools*.

Inicialmente, se debe desarrollar una aplicación web, que permita la creación de proyectos dentro del *Grid*, el registro de recursos, la creación de *pools*, la administración de trabajos y el monitoreo de los mismos.

Cada una de las tareas mencionadas anteriormente, está asociada a un *web service*, dentro del *Globus Toolkit*, por lo que, en el desarrollo de la aplicación, se debe hacer uso de esta herramienta para realizar las funciones solicitadas.

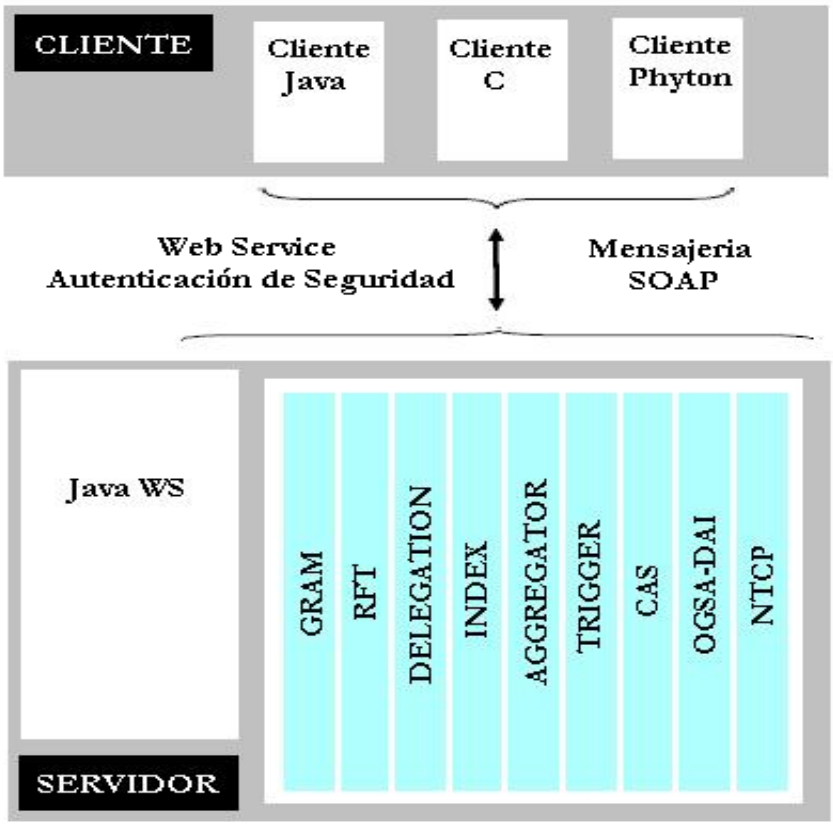
Cada usuario y cada recurso son únicos, y están identificados por una llave privada y asociados con un certificado público, los cuales son proporcionados por el *Grid*. Para la autenticación se hace uso de WS-Security, que es un mecanismo para la autenticación de las credenciales asociadas a una petición y al usuario que la solicito, además se utiliza MyProxy, que es un repositorio de credenciales, el cual elimina la necesidad de copiar las llaves y certificados hacia la máquina del usuario, estas pueden ser recuperadas en el momento que sean requeridas, utilizando una frase que solamente el usuario conozca.

La planificación, monitoreo y administración de trabajos, permiten que los usuarios sometan trabajos en el *Grid*, la asignación de los recursos necesarios para realizar dichos trabajos y la capacidad de establecer el estado en el que se encuentran. Para administrar estas tareas existe GRAM, que tiene la capacidad de ejecutar programas arbitrarios, monitorear el estado de los procesos, controlar el ciclo de vida de los procesos, administración de credenciales, preparación de archivos y la facilidad de interactuar con los planificadores.

El monitoreo y descubrimiento de recursos es una tarea que contempla la obtención, distribución, creación de índices y almacenamiento de la información acerca de la configuración y estado, de los servicios y los recursos. Ambos deben ser capaces de recolectar información de múltiples fuentes, que quizá se encuentre distribuidas. Para este propósito existe MDS4, que posee tres servicios agregados con diferentes comportamientos: *MDS-Index*, que soporta consultas acerca de las características obtenidas de las fuentes de información; *MDS-Trigger*, que permite realizar ciertas tareas definidas por los usuarios, siempre que la información recolectada cumpla con ciertos criterios definidos por el usuario; *MDS-Archive*, que almacena la información de los recursos en una base de datos persistente, la cual puede ser consultada posteriormente.

Las tareas relacionadas con la localización, transferencia y administración de datos distribuidos, tanto para datos estructurados como para datos parcialmente estructurados. Es una tarea que requiere de herramientas capaces de interactuar con las diferentes fuentes de información, de manera que pueda tener acceso a los datos en ellas almacenadas e integrar los mismos para proveer la información requerida. De esta cuenta *Globus Toolkit*, proporciona OGSA-DAI que es un interfaz para el acceso e integración de datos, que utiliza como base OGSA.[71]

Figura 14 – Arquitectura de Servicios. Vista de componentes.



Fuente: Grid Alliance, *Globus Toolkit Primer*, p. 20

4.5. Distribución de Software

El Software del *Grid* se distribuye a través de dos paquetes: el paquete de *Globus Toolkit* y el paquete del *Grid*. El paquete del *Grid* es un complemento del primero, y necesita que *Globus Toolkit* haya sido instalado con anterioridad.

Para garantizar la compatibilidad de las librerías y versiones del compilador en cada una de las plataformas, en las máquinas que tengan instalado alguna variedad de Linux o Unix, se seleccionó la versión que contiene el código fuente del paquete, en vez de la versión binaria. Mientras que para las máquinas que tengan instalado Windows, se procederá a proporcionar la versión binaria de la herramienta.

El paquete del *Grid*, debe proporcionar un conjunto de scripts y de herramientas, que permitan al usuario agregar y configurar de manera fácil y rápida un recurso al *Grid*. El paquete del *Grid* debe proporcionar las siguientes funciones:[72]

- Configuración de la seguridad.
  - Generación de las configuraciones de seguridad, así como los archivos y directorios asociados con esta.
  - Obtención dinámica de los certificados de autenticación.
  - Autenticación automática en el *Grid*, a través de *web services*.
- Configuración del MDS.
  - Descubrir o acceder a la información.
  - Registrar los recursos.
  - Recolectar la información de los recursos.
  - Agregar la información recolectada a las tablas específicas, para poder consultar esta información.

4.6. Interfaz para tareas administrativas

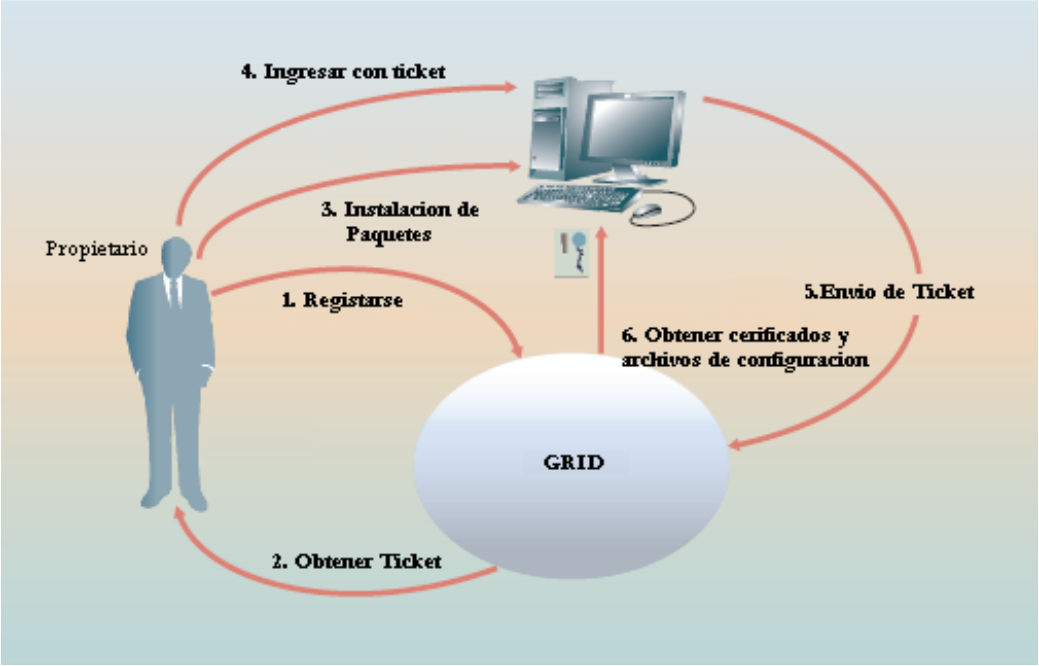
En esta sección se describe las interacciones de los usuarios dentro del *Grid*, desde la perspectiva del administrador al crear un *pool* de recursos, desde el punto de vista del usuario al configurar un recurso, desde el punto de vista del líder de un proyecto al crear un proyecto y desde el punto de vista del usuario al configurar una máquina cliente.

- **Creación de un *pool* de recursos y configuración de un recurso**.[73] Para poder crear un *pool* de recursos, primeramente el administrador debe registrarse en la aplicación web, utilizando su usuario y contraseña. El administrador debe proporcionar el nombre del *pool*, el tipo del *pool* (público o privado), la ruta completa de la máquina que actuará como MDS-Index. Cuando la creación del *pool* haya concluido, se debe proporcionar un ticket que será utilizado para configurar el servidor MDS-Index.

  Una vez que se encuentra creado el *pool*, pueden agregarse recursos al mismo. Si el *pool* es privado solamente el administrador podrá registrar nuevos recursos; se necesita la autoridad del administrador para realizar tareas de configuración e instalación. Si el *pool* es público, cualquier usuario con privilegios de administrador puede registrar recursos.

  La figura presentada a continuación ilustra los pasos que se requieren para la configuración de un recurso.

Figura 15 – Configuración de recursos.



Fuente: Meliksetian y otros, *Design and Implementation of an Enterprise Grid*, p. 651

- **Creación de un proyecto y agregar usuarios al proyecto.**[74] La creación de un proyecto, estará a cargo de un usuario de tipo líder de proyecto. El usuario se conecta a la aplicación web, especifica el nombre del proyecto y una breve descripción del propósito del proyecto. Posteriormente se pueden asignar miembros al proyecto, utilizando el ID que corresponde al usuario.

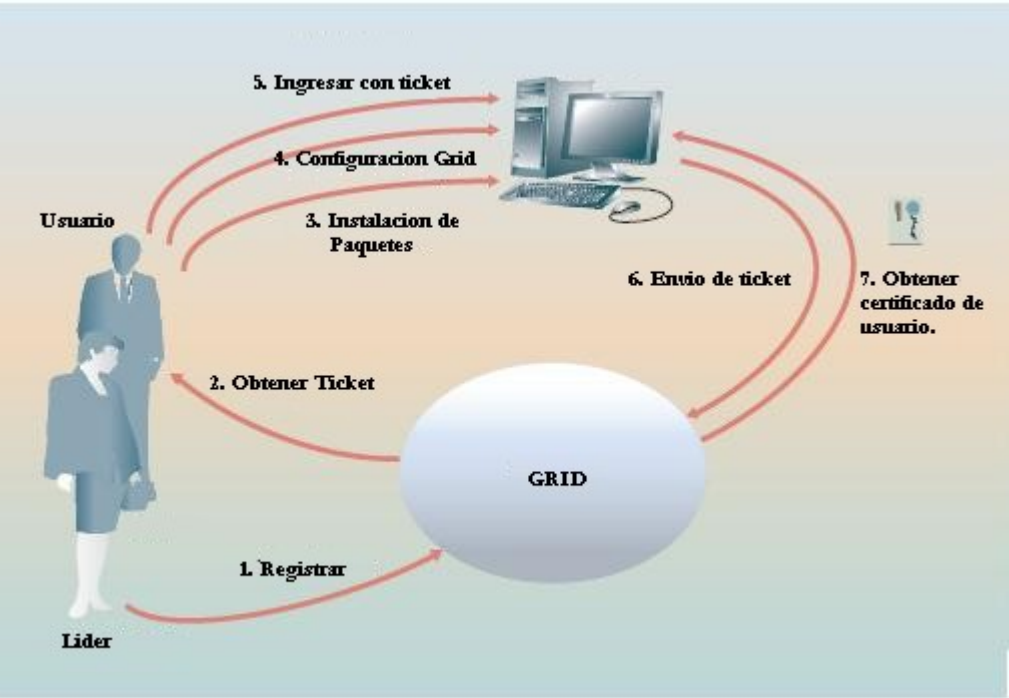
  El líder del proyecto no necesariamente debe pertenecer al grupo que ha sido creado. De tal manera que si quiere ser miembro del proyecto, deberá agregarse como tal. Un usuario puede ser miembro de múltiples proyectos.

- **Configuración de la máquina del cliente**.[75] La máquina que será configurada en el *Grid*, puede estar registrada en el *Grid* como un recurso, o puede ser una máquina independiente. De cualquiera de las dos formas, el usuario debe tener una cuenta en esa máquina.

  Como se menciono previamente, el usuario puede ser miembro de varios proyectos utilizando una identidad diferente, por lo que es necesario obtener un certificado por cada proyecto asociado.

  La siguiente figura muestra los pasos que están involucrados en la configuración de la máquina del cliente:

Figura 16 – Configuración de la máquina de un cliente.



Fuente: Meliksetian y otros, *Design and Implementation of an Enterprise Grid*, p. 652

4.7. Servicios de información

Los servicios de información permiten el monitoreo y el descubrimiento de servicios y recursos en sistemas distribuidos. Una fuente de información puede ser en esencia una entidad de la cual, mediante un servicio agregado, se puede obtener información. En general, se debe poder obtener información de cualquier recurso que esté conectado al *Grid*. Los componentes que conforman el servicio de información son:

- *MDS-Index Service*. Recolecta la información de los recursos disponibles y la convierte a un formato XML. Más específicamente, los datos son almacenados como archivo con formato *WS Resource Framework* (WSRF), que es un marco de trabajo para el modelado y acceso a los estados de un recurso utilizado *web services*. Almacena información estática tal como: sistema operativo, procesador, dispositivos de almacenamiento, dispositivos de red, etc. Así como información dinámica: carga del CPU, colas de datos, sistemas de archivos.

    - Los usuarios pueden construir sus propias aplicaciones que recolecten información, como interfaces estándar.
    - La herramienta de línea de comando, puede ser utilizada para obtener las propiedades de un recurso, proporcionando la ruta del recurso.
    - Existe una herramienta para presentar la información de los recursos en formato HTML, dentro de cualquier browser.

- *MDS-Trigger Service.* Define una interfaz *web service*, que permite ejecutar tareas definidas por los usuarios, siempre que la información recolectada cumpla con ciertos criterios definidos por el usuario

- *MDS-Archive*. Permite almacenar la información de los recursos en una base de datos persistente, la cual puede ser consultada posteriormente por los usuarios.

La información que es recolectada por cada uno de los *MDS-Index,* debe ser almacenada por el *MDS-Archive*, para su posterior utilización. Adicionalmente, cada cierto tiempo se debe verificar que los recursos sigan estando disponibles. La información que es recolectada por el *MDS-Index*, puede estar almacenada por un tiempo en memoria para que pueda ser utilizada, de una forma más eficiente, sin embargo después de un periodo de tiempo, la información es considerada obsoleta. De igual forma, el *MDS-Index* global, verifica cada cierto tiempo que los *MDS-Index* de cada uno de los *pools*, se encuentre disponible.

El principal proceso que consume información acerca de los servicios y recursos, es el encargado de ejecutar los trabajos. Ya que su función es la de establecer un conjunto de recursos que satisfagan, los requerimientos de los proceso definidos por el usuario. Inicialmente se verifican las características globales de cada uno de los *pools*, y posteriormente se entra en detalle en cada uno de sus componentes, para establecer la carga de trabajo y las características de los mismos. [76]

4.8. Servicios administrativos

Los servicios administrativos, están diseñados para facilitar la interacción de los usuarios dentro del *Grid*. Los servicios administrativos son: servicio de registro, que soporta el registro de usuarios y recursos; y el servicio de actualización, que soporta la actualización automática de componentes.

En este punto se han detectado tres categorías de usuarios: los proveedores, los consumidores y los administradores.

- **Servicio de registro**. Desde el punto de vista administrativo, es necesario contar con un servicio que permita administrar y controlar los recursos disponibles en el *Grid*, además de poder rastrear su utilización. Este servicio almacena la información de proyectos, usuarios, *pools* y recursos. Posee una interfaz, con soporte de envío de mail para enviar notificaciones en caso de que una acción requiera de la aprobación de un usuario.

  El servicio de registro, será implementado como una aplicación web, basada en el modelo de patrones MVC (*Model View Controller*). El controlador, que administra todas las interacciones entre usuarios y aplicaciones, consistirá en un conjunto de *servlets* que implementen métodos específicos para cada una de las acciones. El controlador enviará los parámetros de entrada al modelo, que consistirá en una sesión de Enterprise JavaBeans, para traer y guardar información en las bases de datos. La respuesta generada será presentada al usuario mediante una página HTML, que será generada de forma dinámica por un *servlet*.[77]

- **Servicio de actualización**. La actualización de código y el mantenimiento de una configuración consistente representa un desafío, debido a que la cantidad de recursos disponibles puede ser bastante grande. Para lograr esto se necesita de un sistema que pueda automáticamente realizar estas tareas, con la menor intervención del ser humano.

  Existen varios tipos de actualizaciones que pueden ser requeridas, entra las que se pueden mencionar: la propagación de la estructura de recursos, la creación y eliminación de usuarios y recursos, modificaciones en el código de los paquetes. Para lograr monitorear estas actualizaciones, se debe contar con varios agentes que se ejecuten como *deamons*, los cuales estén encargados de estas tareas específicas. [78]

4.9. Servicio de administración de trabajos

El acceso a los recursos, usualmente, requiere de la instalación de algún software y de credenciales en la máquina que actuara como cliente. Por lo que el *Grid* implementará un mecanismo de automatización de este proceso, para facilitar el acceso de los usuarios. Esta facilidad estará disponible cuando el usuario ingrese utilizando la aplicación web, que permite la administración de los trabajos.

- **Aplicación *Web*.** El proceso para ejecutar un trabajo estará definido por los siguientes pasos:
    - Creación de credenciales.
    - Definición de las características del trabajo.
    - Selección de recursos.
    - Recolección de resultados.

Todas las interacciones con la aplicación web, se darán luego de que el usuario se haya autenticado en la aplicación de forma exitosa, por lo que la aplicación debe mantenerse integrada con el control de usuarios del *Grid*. Si el usuario estuviera asignado a múltiples proyectos, se deberá seleccionar el proyecto deseado para trabajar, por el contrario si no tiene ningún proyecto asignado se le proporcionará el perfil de visitante.

Los certificados serán manejados desde una ventana, donde el usuario podrá solicitar un certificado ingresando una frase a su elección, que le servirá de contraseña. Es de hacer notar que el certificado para un proyecto determinado, únicamente debe ser generado una vez, a menos que se solicite nuevamente su creación por parte de MyProxy.

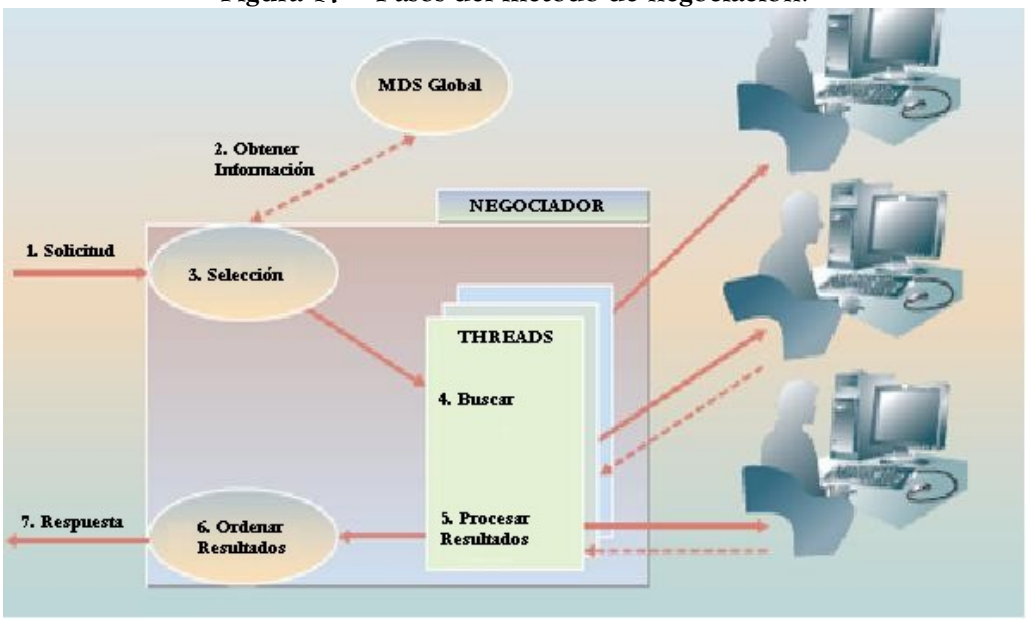
Una vez que se ha seleccionado el proyecto y los certificados han sido generados, puede iniciarse con la ejecución de trabajos dentro del *Grid*. La aplicación hará uso de GRAM, para someter los trabajos, proporcionándole las características del trabajo a ejecutar.

Existen dos modos de ejecución de trabajos: el modo interactivo, que envía una petición y espera la respuesta a esta petición, este método es generalmente utilizado para trabajos cortos; el modo *batch*, implementa manejadores de eventos de tal cuenta que, cuando el trabajo termina, la aplicación recibe la notificación de que el trabajo ha sido finalizado.

La aplicación cuenta con dos formas de seleccionar los recursos que serán utilizados para la ejecución de un trabajo, que son: el método de selección de recursos registrados y el método de selección por negociación. En el primer método la aplicación proporciona al usuario una lista de todos los recursos que se encuentran registrados. En el método de negociación, el usuario selecciona los criterios, pero la aplicación es la encargada de determinar qué recursos cumplen con dichos criterios y selecciona el recurso que mejor se adapte a estas necesidades.

La aplicación debe permitir el despliegue de los trabajos recientes, ejecutados por el usuario, de esta lista se puede seleccionar cualquiera de los trabajos, para poder observar sus detalles específicos.

Figura 17 – Pasos del método de negociación.



Fuente: Meliksetian y otros, *Design and Implementation of an Enterprise Grid*, p. 659

4.10. Creación de servicios

La creación de servicios dependerá de las necesidades de cada una de las unidades académicas o de las necesidades de un usuario del *Grid*, de tal cuenta los servicios serán creados por el ente que las requiera. Para dar paso a la creación y puesta en marcha, de los servicios, es necesario que la infraestructura tanto física como lógica del *Grid* se encuentre implementada, por lo menos en el nivel elemental.

Para la creación de los servicios se propone que el desarrollo se realice utilizando Java *Platform Enterprise Edition* (Java EE), que es un estándar que permite el desarrollo de aplicaciones Java, del lado del servidor, que sean portables, robustas, escalables y seguras.[††††] Java es un lenguaje de programación bastante completo, que permite portabilidad a una gran cantidad de plataformas, sin que se requiera de ningún esfuerzo adicional, lo que podría facilitar la convivencia de los servicios, con la diversidad de plataformas que existen dentro de la Universidad.

El tema específico de la creación de servicios, no será abordada en este documento, ya que es un tema es bastante amplio, y que esta fuera del alcance de la investigación.

†††† http://java.sun.com/javaee/

4.11. Beneficios

La implementación del *Grid,* como se ha mencionado anteriormente, más que un proyecto de desarrollo en sí mismo, está orientada a que el *Grid* pueda ser utilizado como un medio para el desarrollo de propuestas de investigación, que requieran de gran capacidad de procesamiento o gran capacidad de almacenamiento de datos. Teniendo como principal beneficio la sinergia de los recursos de la Universidad para el desarrollo de proyectos.

Los beneficios para la Universidad de San Carlos de Guatemala, podrán verse reflejados a mediano y largo plazo, debido a que, en el corto plazo es probable que los proyectos que se desarrollen, no utilicen las capacidades del *Grid* en un cien por ciento; ya que no se posee ninguna experiencia en este tipo de desarrollos, ni en la utilización de un *Grid*. Pero una vez que estos conocimientos maduren, como en otros países que están más adelantados en el uso de esta tecnología, podrán apreciarse los resultados.

Se enumeraran a continuación los principales beneficios, que se espera, que puedan alcanzarse en la Universidad con la implementación del *Grid*.

- **Mejorar la productividad de personas y sistemas**. Permitir el acceso a los recursos computacionales y de almacenamiento cuando sea requerido, permitir la integración de recursos heterogéneos y proporcionar métodos sencillos para compartir, acceder o administrar información; así como para colaborar a través de la organización.

- **Minimizar costos.** Maximizando la productividad, de manera que los recursos con que cuenta la Universidad sean utilizados de una forma eficiente.

- **Estar a la vanguardia.** Al mejorar la productividad, la eficiencia y la colaboración entre los diferentes entes, es posible reducir los tiempos en los procesos y obtener resultados de forma más rápida. Con esta tecnología es posible establecer vínculos de colaboración con otras universidades y comunidades científicas, de manera que se puedan desarrollar proyectos de investigación conjuntos tales como: nutrición, la búsqueda de la cura contra el cáncer y otras enfermedades, investigaciones sobre nanotecnología, predicción de terremotos y muchos otros proyectos (actualmente, en Estados Unidos y Europa, este tipo de proyectos ya se encuentran en funcionamiento).

- **Aplicación a la investigación y desarrollo.** Las actividades de investigación y desarrollo, dirigidas por computadora, requieren de gran capacidad de procesamiento, métodos de análisis, extracción y procesamiento de datos. Por lo que con el *Grid*, se puede poner a disposición de la comunidad de investigación y desarrollo esta tecnología, de manera que pueden disponer de las capacidades de procesamiento de datos y almacenamiento para los proyectos que así lo requieran.

4.12. Plan de implementación

El plan detalla, a nivel macro, la lista de tareas que serán necesarias para realizar la implementación del *Grid* en la Universidad de San Carlos de Guatemala, en los cuatro niveles de abstracción mencionados anteriormente en este capítulo. Al finalizar estas tareas el *Grid* deberá encontrarse en un estado operativo, para que pueda ser utilizado por las unidades académicas que así lo requieran.

4.12.1. Objetivos

- **Generales**:
    - Implementar los cuatro niveles de abstracción que se diseñaron en la propuesta.

- **Específicos**:
    - Realizar la implementación del *Grid* en la unidad académica de Ingeniería, así como la implementación del anillo administrativo y del anillo tecnológico.
    - Capacitar al personal de informática de las diferentes unidades académicas y personal de informática de la Universidad, para que puedan realizar la instalación de los anillos restantes, así como la de las unidades académicas que lo requieran.
    - Supervisar al personal de informática en la tarea de instalación y configuración de los anillos o unidades académicas.
    - Integración de los anillos, y configuración general para establecer comunicación con entidades externas.

4.12.2. Alcance

Aunque ya se ha mencionado con anterioridad, es necesario recalcar, que la implementación del *Grid*, está orientada a que pueda ser utilizado como un medio o herramienta para el desarrollo de propuestas de investigación, que requieran de gran capacidad de procesamiento o gran capacidad de almacenamiento de datos. Por lo que, el alcance de la propuesta está orientada a la implementación de esta infraestructura, y no al desarrollo de aplicaciones especificas que utilicen el *Grid*.

Dentro del alcance de la implementación, no se incluye el desarrollo de ningún servicio, por lo menos con el objetivo de colocarlo en producción, para la resolución de algún tema en especial. Pero, se utilizarán algunos ejemplos para ponerla a prueba.

Se implementaran todos los niveles descritos en el diseño, y las tareas de instalación serán compartidas con las áreas de informática que se encuentren involucradas. Se tiene previsto involucrar al área de informática de la Universidad de San Carlos de Guatemala, al área de informática de la Facultad de Ingeniería, así como a las áreas de informática de las unidades académicas que deseen instalar un *Grid*.

- **Instalación de herramientas administrativas:** anteriormente se realizó una descripción de las herramientas contenidas en *Globus Toolkit*, para la gestión de tareas administrativas, servicios administrativos, servicios de información, administración de trabajos y la creación de servicios. Tomando esta información como base, se realizará la instalación y configuración de *Globus Toolkit*, y de los siguientes componentes, como herramienta para la administración del *Grid*.

    - WS-Security
    - GRAM
    - MDS
    - OGSA-DAI

- **Implementación de componentes *web*:** anteriormente se hizo mención de un conjunto de componentes web, que permiten realizar la gestión de tareas administrativas, gestión de servicios administrativos, gestión de servicios de información, administración de trabajos y creación de servicios de una forma sencilla, sin que el usuario tenga que aprender la sintaxis de cada una de las herramientas, ni su funcionamiento especifico. De esta forma se busca, que la administración de los componentes sea transparente para el usuario final. A continuación se listan los componentes web que serán implementados.

    - Distribución de software
    - Interfaz para tareas administrativas
    - Servicios de información
    - Servicios administrativos
    - Servicio de administración de trabajos

4.12.3. Plan de trabajo

La descripción del plan de trabajo, que se presenta a continuación, incluye información acerca de las distintas actividades a realizar, para lograr alcanzar los objetivos planteados. En general las tareas inician con la implementación del *Grid* en la Faculta de Ingeniería, que representa el nivel más bajo de abstracción, y con la implementación del Anillo Tecnológico y del Anillo Administrativo, por parte del coordinador del proyecto. Una vez que este objetivo se cumpla, se procederá a la capacitación de los grupos involucrados en el proyecto, uno en representación de la Universidad y de preferencia uno por cada unidad académica, de tal forma que sean ellos los que continúen con la implementación de los *grids* involucrados con su unidad académica; siempre bajo la supervisión del coordinador del proyecto. Finalmente se debe realizar la integración de los *grids*, para la creación del "Intragrid USAC", de tal forma que exista comunicación e interacción de todos los *grids* implementados, logrando así la sinergia esperada.

- **Reuniones informativas:**

Para iniciar el proyecto, se realizaran una serie de reuniones informativas, dirigidas a todas las aéreas de IT de la Universidad de San Carlos de Guatemala, donde se plantea la solución propuesta, el alcance y los beneficios que se busca alcanzar con este proyecto. De esta forma se busca crear interés en la propuesta e involucrar a todas las aéreas que puedan colaborar con la implementación.

- **Inventario de recursos:**

Realizar un inventario de los recursos, de cada una de las unidades académicas del Campus Central de la Universidad de San Carlos de Guatemala, para establecer los recursos existentes, sus características, localización y la disponibilidad que se tenga para que estos pertenezcan al *Grid*.

- **Selección de equipo:**

Posteriormente a la realización del inventario, se procederá a evaluar que parte del equipo existente y disponible, cumple con las especificaciones para poder ser utilizado como servidor para la instalación de *Globus Toolkit*, de lo contrario se realizara la solicitud del equipo necesario.

- **Implementación:**

La implementación del *Grid*, está dividida en dos fases: la instalación y configuración de *Globus Toolkit* y la instalación de los componentes web para la administración. Esta tarea será replicada en cada una de las unidades académicas y anillos que sean instalados, por lo que dentro del cronograma esta actividad se efectuará en múltiples ocasiones, con actores diferentes. Adicionalmente durante la primera instalación se realizará el desarrollo de los componentes web para la administración.

Cada una de las fases mencionadas, requiere de tiempo y recurso humano para el control de calidad, pruebas y puesta en marcha. Por lo que dentro del cronograma se pretende mostrar detalladamente cada una de las actividades, tiempo requerido y personal que debe estar involucrado.

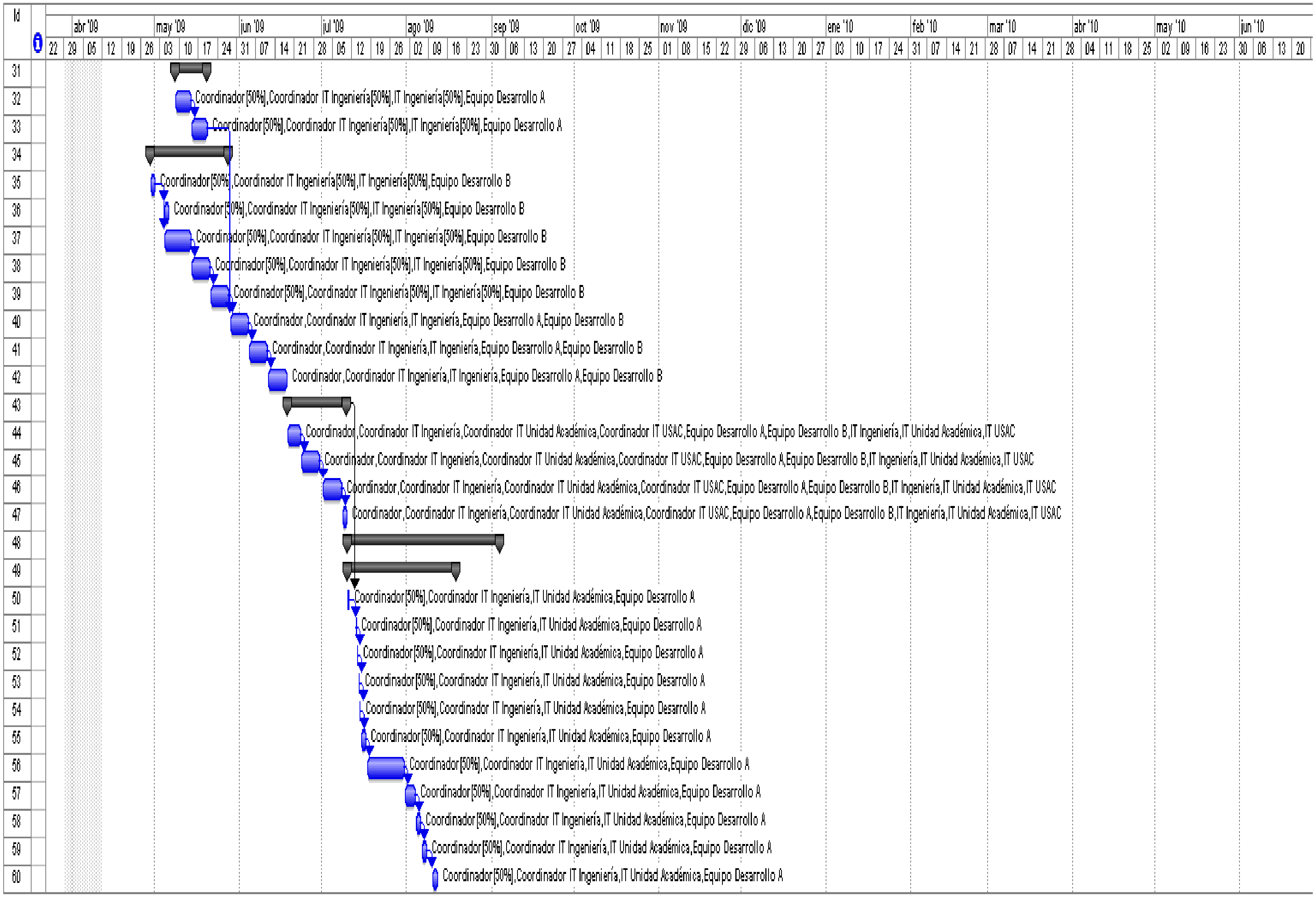
- **Capacitación:**

Finalizada la primera instalación, se procederá a la realizar una capacitación dirigida a todas las aéreas de IT de la Universidad de San Carlos de Guatemala. La capacitación está dirigida a introducir el concepto de *Grid*, definir los pasos para su implementación en el campus, definir los pasos para implementar los componentes web de la administración y ofrecer un taller, para que puedan ponerse en práctica los conocimientos adquiridos.

Id	Nombre de tarea	Duración	Comienzo	Fin	Predecesor	Nombres de los recursos
1	Reuniones Informativas	5 días	lun 05/01/09	vie 09/01/09		Coordinador,IT USAC,IT Ingeniería,IT Unidad Académica
2	Inventario de Recursos	15 días?	lun 12/01/09	vie 30/01/09	1	Coordinador,IT USAC,IT Ingeniería,IT Unidad Académica
3	Selección de Equipo	2 días	lun 02/02/09	mar 03/02/09	2	Coordinador
4	Implementación Facultad de Ingeniería	85 días	mié 04/02/09	mié 17/06/09		
5	Instalación Globus Toolkit	28 días	mié 04/02/09	vie 13/03/09		
6	Instalación de Linux	1 día	mié 04/02/09	mié 04/02/09	3	Coordinador,Coordinador IT Ingeniería,IT Ingeniería
7	Nombres y direcciones	0.5 días	jue 05/02/09	jue 05/02/09	6	Coordinador,Coordinador IT Ingeniería,IT Ingeniería
8	Certificados	0.5 días	jue 05/02/09	jue 05/02/09	7	Coordinador,Coordinador IT Ingeniería,IT Ingeniería
9	Usuarios y grupos	0.5 días	vie 06/02/09	vie 06/02/09	8	Coordinador,Coordinador IT Ingeniería,IT Ingeniería
10	Directorios	0.5 días	vie 06/02/09	vie 06/02/09	9	Coordinador,Coordinador IT Ingeniería,IT Ingeniería
11	Instalación de herramientas complementarias	2 días	lun 09/02/09	mar 10/02/09	10	Coordinador,Coordinador IT Ingeniería,IT Ingeniería
12	Instalación de GT3	10 días	mié 11/02/09	mar 24/02/09	11	Coordinador,Coordinador IT Ingeniería,IT Ingeniería
13	Configuración WS- Security	2 días	mié 25/02/09	jue 26/02/09	12	Coordinador,Coordinador IT Ingeniería,IT Ingeniería
14	Configuración GRAM	2 días	vie 27/02/09	lun 02/03/09	13	Coordinador,Coordinador IT Ingeniería,IT Ingeniería
15	Configuración MDS	2 días	mar 03/03/09	mié 04/03/09	14	Coordinador,Coordinador IT Ingeniería,IT Ingeniería
16	Configuración OGSA-DAI	2 días	jue 05/03/09	vie 06/03/09	15	Coordinador,Coordinador IT Ingeniería,IT Ingeniería
17	Pruebas controladas	5 días	lun 09/03/09	vie 13/03/09	16	Coordinador,Coordinador IT Ingeniería,IT Ingeniería
18	Desarrollo Componentes WEB	57 días	lun 16/03/09	mié 17/06/09		
19	Ambiente para distribución de software	10 días	lun 16/03/09	vie 27/03/09		
20	Configuración de la seguridad	5 días	lun 16/03/09	vie 20/03/09	17	Coordinador[50%],Coordinador IT Ingeniería[50%],IT Ingeniería[50%],Equipo Desarrollo A
21	Configuración del MDS	5 días	lun 23/03/09	vie 27/03/09	20	Coordinador[50%],Coordinador IT Ingeniería[50%],IT Ingeniería[50%],Equipo Desarrollo A
22	Crear Interfaz para Tareas Administrativas	18 días	lun 13/04/09	jue 07/05/09		
23	Creación de pool de recursos, configuración	8 días	lun 13/04/09	mié 22/04/09	21	Coordinador[50%],Coordinador IT Ingeniería[50%],IT Ingeniería[50%],Equipo Desarrollo A
24	Creación de proyectos	4 días	jue 23/04/09	mar 28/04/09	23	Coordinador[50%],Coordinador IT Ingeniería[50%],IT Ingeniería[50%],Equipo Desarrollo A
25	Creación de usuarios	4 días	mié 29/04/09	mar 05/05/09	24	Coordinador[50%],Coordinador IT Ingeniería[50%],IT Ingeniería[50%],Equipo Desarrollo A
26	Configuración de máquina cliente	2 días	mié 06/05/09	jue 07/05/09	25	Coordinador[50%],Coordinador IT Ingeniería[50%],IT Ingeniería[50%],Equipo Desarrollo A
27	Crear Servicios de Información	12 días	lun 13/04/09	mar 28/04/09		
28	Configurar MDS-Index Service	4 días	lun 13/04/09	jue 16/04/09	21	Coordinador[50%],Coordinador IT Ingeniería[50%],IT Ingeniería[50%],Equipo Desarrollo B
29	Configurar MDS-Trigger Service	4 días	vie 17/04/09	mié 22/04/09	28	Coordinador[50%],Coordinador IT Ingeniería[50%],IT Ingeniería[50%],Equipo Desarrollo B
30	Configurar MDS-Archive	4 días	jue 23/04/09	mar 28/04/09	29	Coordinador[50%],Coordinador IT Ingeniería[50%],IT Ingeniería[50%],Equipo Desarrollo B

Coordinador,IT USAC,IT Ingeniería,IT Unidad Académica
Coordinador,IT USAC,IT Ingeniería,IT Unidad Académica
Coordinador
Coordinador,Coordinador IT Ingeniería,IT Ingeniería
Coordinador,Coordinador IT Ingeniería,IT Ingeniería
Coordinador,Coordinador IT Ingeniería,IT Ingeniería
Coordinador,Coordinador IT Ingeniería,IT Ingeniería
Coordinador,Coordinador IT Ingeniería,IT Ingeniería
Coordinador,Coordinador IT Ingeniería,IT Ingeniería
Coordinador,Coordinador IT Ingeniería,IT Ingeniería
Coordinador,Coordinador IT Ingeniería,IT Ingeniería
Coordinador,Coordinador IT Ingeniería,IT Ingeniería
Coordinador,Coordinador IT Ingeniería,IT Ingeniería
Coordinador,Coordinador IT Ingeniería,IT Ingeniería
Coordinador[50%],Coordinador IT Ingeniería[50%],IT Ingeniería[50%],Equipo Desarrollo A
Coordinador[50%],Coordinador IT Ingeniería[50%],IT Ingeniería[50%],Equipo Desarrollo A
Coordinador[50%],Coordinador IT Ingeniería[50%],IT Ingeniería[50%],Equipo Desarrollo A
Coordinador[50%],Coordinador IT Ingeniería[50%],IT Ingeniería[50%],Equipo Desarrollo A
Coordinador[50%],Coordinador IT Ingeniería[50%],IT Ingeniería[50%],Equipo Desarrollo A
Coordinador[50%],Coordinador IT Ingeniería[50%],IT Ingeniería[50%],Equipo Desarrollo B
Coordinador[50%],Coordinador IT Ingeniería[50%],IT Ingeniería[50%],Equipo Desarrollo B
Coordinador[50%],Coordinador IT Ingeniería[50%],IT Ingeniería[50%],Equipo Desarrollo B

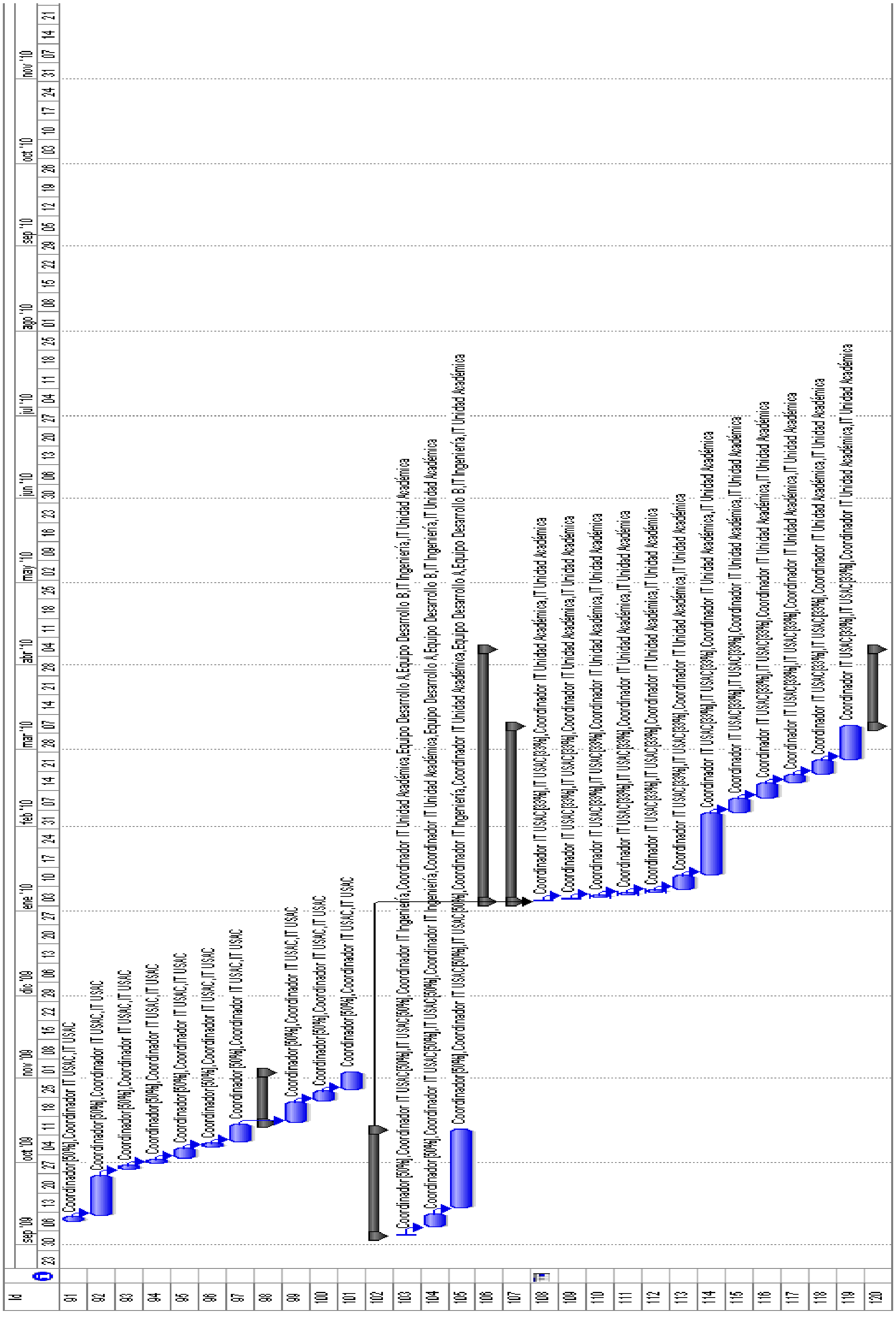
Id	Nombre de tarea	Duración	Comienzo	Fin	Predecesor	Nombres de los recursos
31	Crear Servicios Administrativos	8 días	vie 08/05/09	mar 19/05/09		
32	Configurar Servicio de Registro	4 días	vie 08/05/09	mié 13/05/09	26	Coordinador[50 %],Coordinador IT Ingeniería[50 %],IT Ingeniería[50 %],Equipo Desarrollo A
33	Configurar Servicio de Actualización	4 días	jue 14/05/09	mar 19/05/09	32	Coordinador[50 %],Coordinador IT Ingeniería[50 %],IT Ingeniería[50 %],Equipo Desarrollo A
34	Crear Servicio de Administración de Trabajos	20 días	mié 29/04/09	mié 27/05/09		
35	Creación de Credenciales	2 días	mié 29/04/09	jue 30/04/09	30	Coordinador[50 %],Coordinador IT Ingeniería[50 %],IT Ingeniería[50 %],Equipo Desarrollo B
36	Definición de características del trabajo	2 días	lun 04/05/09	mar 05/05/09	35	Coordinador[50 %],Coordinador IT Ingeniería[50 %],IT Ingeniería[50 %],Equipo Desarrollo B
37	Selección de recursos	8 días	lun 04/05/09	mié 13/05/09	35	Coordinador[50 %],Coordinador IT Ingeniería[50 %],IT Ingeniería[50 %],Equipo Desarrollo B
38	Recolección de resultados	5 días	jue 14/05/09	mié 20/05/09	37	Coordinador[50 %],Coordinador IT Ingeniería[50 %],IT Ingeniería[50 %],Equipo Desarrollo B
39	Administración de trabajos	5 días	jue 21/05/09	mié 27/05/09	38	Coordinador[50 %],Coordinador IT Ingeniería[50 %],IT Ingeniería[50 %],Equipo Desarrollo B
40	Control de Calidad	5 días	jue 28/05/09	mié 03/06/09	39,33	Coordinador,Coordinador IT Ingeniería,IT Ingeniería,Equipo Desarrollo A,Equipo Desarrollo B
41	Cambios y adecuaciones	5 días	jue 04/06/09	mié 10/06/09	40	Coordinador,Coordinador IT Ingeniería,IT Ingeniería,Equipo Desarrollo A,Equipo Desarrollo B
42	Pruebas Controladas	5 días	jue 11/06/09	mié 17/06/09	41	Coordinador,Coordinador IT Ingeniería,IT Ingeniería,Equipo Desarrollo A,Equipo Desarrollo B
43	Capacitación	15 días	jue 18/06/09	jue 09/07/09		
44	Introducción al Grid	3 días	jue 18/06/09	lun 22/06/09	4	Coordinador,Coordinador IT Ingeniería,Coordinador IT Unidad Académica,Coordinador IT USAC,Equipo Desarrollo A,Equipo Desarrollo B,IT Ingeniería,IT Unidad Académica,IT USAC
45	Implementación del Grid	5 días	mar 23/06/09	lun 29/06/09	44	Coordinador,Coordinador IT Ingeniería,Coordinador IT Unidad Académica,Coordinador IT USAC,Equipo Desarrollo A,Equipo Desarrollo B,IT Ingeniería,IT Unidad Académica,IT USAC
46	Implementación de Componentes Web	5 días	mié 01/07/09	mar 07/07/09	45	Coordinador,Coordinador IT Ingeniería,Coordinador IT Unidad Académica,Coordinador IT USAC,Equipo Desarrollo A,Equipo Desarrollo B,IT Ingeniería,IT Unidad Académica,IT USAC
47	Taller	2 días	mié 08/07/09	jue 09/07/09	46	Coordinador,Coordinador IT Ingeniería,Coordinador IT Unidad Académica,Coordinador IT USAC,Equipo Desarrollo A,Equipo Desarrollo B,IT Ingeniería,IT Unidad Académica,IT USAC
48	Implementación Anillo Tecnológico	40 días	vie 10/07/09	jue 03/09/09		
49	Instalación Globus Toolkit	28 días	vie 10/07/09	mar 18/08/09		
50	Instalación de Linux	1 día	vie 10/07/09	vie 10/07/09	43	Coordinador[50 %],Coordinador IT Ingeniería,IT Unidad Académica,Equipo Desarrollo A
51	Nombres y direcciones	0.5 días	lun 13/07/09	lun 13/07/09	50	Coordinador[50 %],Coordinador IT Ingeniería,IT Unidad Académica,Equipo Desarrollo A
52	Certificados	0.5 días	lun 13/07/09	lun 13/07/09	51	Coordinador[50 %],Coordinador IT Ingeniería,IT Unidad Académica,Equipo Desarrollo A
53	Usuarios y grupos	0.5 días	mar 14/07/09	mar 14/07/09	52	Coordinador[50 %],Coordinador IT Ingeniería,IT Unidad Académica,Equipo Desarrollo A
54	Directorios	0.5 días	mar 14/07/09	mar 14/07/09	53	Coordinador[50 %],Coordinador IT Ingeniería,IT Unidad Académica,Equipo Desarrollo A
55	Instalación de herramientas complementarias	2 días	mié 15/07/09	jue 16/07/09	54	Coordinador[50 %],Coordinador IT Ingeniería,IT Unidad Académica,Equipo Desarrollo A
56	Instalación de GT3	10 días	vie 17/07/09	jue 30/07/09	55	Coordinador[50 %],Coordinador IT Ingeniería,IT Unidad Académica,Equipo Desarrollo A
57	Configuración WS- Security	2 días	vie 31/07/09	lun 03/08/09	56	Coordinador[50 %],Coordinador IT Ingeniería,IT Unidad Académica,Equipo Desarrollo A
58	Configuración GRAM	2 días	mar 04/08/09	mié 05/08/09	57	Coordinador[50 %],Coordinador IT Ingeniería,IT Unidad Académica,Equipo Desarrollo A
59	Configuración MDS	2 días	jue 06/08/09	vie 07/08/09	58	Coordinador[50 %],Coordinador IT Ingeniería,IT Unidad Académica,Equipo Desarrollo A
60	Configuración OGSA-DAI	2 días	lun 10/08/09	mar 11/08/09	59	Coordinador[50 %],Coordinador IT Ingeniería,IT Unidad Académica,Equipo Desarrollo A


Id
abr '09
may '09
jun '09
jul '09
ago '09
sep '09
oct '09
nov '09
dic '09
ene '10
feb '10
mar '10
abr '10
may '10
jun '10
22 29 05 12 19 26 03 10 17 24 31 07 14 21 28 05 12 19 26 02 09 16 23 30 06 13 20 27 04 11 18 25 01 08 15 22 29 06 13 20 27 03 10 17 24 31 07 14 21 28 07 14 21 28 04 11 18 25 02 09 16 23 30 06 13 20
31
32 Coordinador[50%],Coordinador IT Ingeniería[50%],IT Ingeniería[50%],Equipo Desarrollo A
33 Coordinador[50%],Coordinador IT Ingeniería[50%],IT Ingeniería[50%],Equipo Desarrollo A
34
35 Coordinador[50%],Coordinador IT Ingeniería[50%],IT Ingeniería[50%],Equipo Desarrollo B
36 Coordinador[50%],Coordinador IT Ingeniería[50%],IT Ingeniería[50%],Equipo Desarrollo B
37 Coordinador[50%],Coordinador IT Ingeniería[50%],IT Ingeniería[50%],Equipo Desarrollo B
38 Coordinador[50%],Coordinador IT Ingeniería[50%],IT Ingeniería[50%],Equipo Desarrollo B
39 Coordinador[50%],Coordinador IT Ingeniería[50%],IT Ingeniería[50%],Equipo Desarrollo B
40 Coordinador,Coordinador IT Ingeniería,IT Ingeniería,Equipo Desarrollo A,Equipo Desarrollo B
41 Coordinador,Coordinador IT Ingeniería,IT Ingeniería,Equipo Desarrollo A,Equipo Desarrollo B
42 Coordinador,Coordinador IT Ingeniería,IT Ingeniería,Equipo Desarrollo A,Equipo Desarrollo B
43
44 Coordinador,Coordinador IT Ingeniería,Coordinador IT Unidad Académica,Coordinador IT USAC,Equipo Desarrollo A,Equipo Desarrollo B,IT Ingeniería,IT Unidad Académica,IT USAC
45 Coordinador,Coordinador IT Ingeniería,Coordinador IT Unidad Académica,Coordinador IT USAC,Equipo Desarrollo A,Equipo Desarrollo B,IT Ingeniería,IT Unidad Académica,IT USAC
46 Coordinador,Coordinador IT Ingeniería,Coordinador IT Unidad Académica,Coordinador IT USAC,Equipo Desarrollo A,Equipo Desarrollo B,IT Ingeniería,IT Unidad Académica,IT USAC
47 Coordinador,Coordinador IT Ingeniería,Coordinador IT Unidad Académica,Coordinador IT USAC,Equipo Desarrollo A,Equipo Desarrollo B,IT Ingeniería,IT Unidad Académica,IT USAC
48
49
50 Coordinador[50%],Coordinador IT Ingeniería,IT Unidad Académica,Equipo Desarrollo A
51 Coordinador[50%],Coordinador IT Ingeniería,IT Unidad Académica,Equipo Desarrollo A
52 Coordinador[50%],Coordinador IT Ingeniería,IT Unidad Académica,Equipo Desarrollo A
53 Coordinador[50%],Coordinador IT Ingeniería,IT Unidad Académica,Equipo Desarrollo A
54 Coordinador[50%],Coordinador IT Ingeniería,IT Unidad Académica,Equipo Desarrollo A
55 Coordinador[50%],Coordinador IT Ingeniería,IT Unidad Académica,Equipo Desarrollo A
56 Coordinador[50%],Coordinador IT Ingeniería,IT Unidad Académica,Equipo Desarrollo A
57 Coordinador[50%],Coordinador IT Ingeniería,IT Unidad Académica,Equipo Desarrollo A
58 Coordinador[50%],Coordinador IT Ingeniería,IT Unidad Académica,Equipo Desarrollo A
59 Coordinador[50%],Coordinador IT Ingeniería,IT Unidad Académica,Equipo Desarrollo A
60 Coordinador[50%],Coordinador IT Ingeniería,IT Unidad Académica,Equipo Desarrollo A

Id	Nombre de tarea	Duración	Comienzo	Fin	Predecesor	Nombres de los recursos
61	Pruebas controladas	5 días	mié 12/08/09	mar 18/08/09	60	Coordinador[50 %],Coordinador IT Ingeniería,IT Unidad Académica,Equipo Desarrollo A
62	Implementación de Componentes WEB	12 días	mié 19/08/09	jue 03/09/09		
63	Instalación de componentes	5 días	mié 19/08/09	mar 25/08/09	61	Coordinador[50 %],Coordinador IT Ingeniería,IT Unidad Académica,Equipo Desarrollo A
64	Configuración de componentes	2 días	mié 26/08/09	jue 27/08/09	63	Coordinador[50 %],Coordinador IT Ingeniería,IT Unidad Académica,Equipo Desarrollo A
65	Pruebas Controladas	5 días	vie 28/08/09	jue 03/09/09	64	Coordinador[50 %],Coordinador IT Ingeniería,IT Unidad Académica,Equipo Desarrollo A
66	Implementación Anillo Administrativo	40 días	vie 10/07/09	jue 03/09/09		
67	Instalación Globus Toolkit	28 días	vie 10/07/09	mar 18/08/09		
68	Instalación de Linux	1 día	vie 10/07/09	vie 10/07/09	43	Coordinador[50 %],Coordinador IT USAC,IT USAC,Equipo Desarrollo B
69	Nombres y direcciones	0.5 días	lun 13/07/09	lun 13/07/09	68	Coordinador[50 %],Coordinador IT USAC,IT USAC,Equipo Desarrollo B
70	Certificados	0.5 días	lun 13/07/09	lun 13/07/09	69	Coordinador[50 %],Coordinador IT USAC,IT USAC,Equipo Desarrollo B
71	Usuarios y grupos	0.5 días	mar 14/07/09	mar 14/07/09	70	Coordinador[50 %],Coordinador IT USAC,IT USAC,Equipo Desarrollo B
72	Directorios	0.5 días	mar 14/07/09	mar 14/07/09	71	Coordinador[50 %],Coordinador IT USAC,IT USAC,Equipo Desarrollo B
73	Instalación de herramientas complementarias	2 días	mié 15/07/09	jue 16/07/09	72	Coordinador[50 %],Coordinador IT USAC,IT USAC,Equipo Desarrollo B
74	Instalación de GT3	10 días	vie 17/07/09	jue 30/07/09	73	Coordinador[50 %],Coordinador IT USAC,IT USAC,Equipo Desarrollo B
75	Configuración WS- Security	2 días	vie 31/07/09	lun 03/08/09	74	Coordinador[50 %],Coordinador IT USAC,IT USAC,Equipo Desarrollo B
76	Configuración GRAM	2 días	mar 04/08/09	mié 05/08/09	75	Coordinador[50 %],Coordinador IT USAC,IT USAC,Equipo Desarrollo B
77	Configuración MDS	2 días	jue 06/08/09	vie 07/08/09	76	Coordinador[50 %],Coordinador IT USAC,IT USAC,Equipo Desarrollo B
78	Configuración OGSA-DAI	2 días	lun 10/08/09	mar 11/08/09	77	Coordinador[50 %],Coordinador IT USAC,IT USAC,Equipo Desarrollo B
79	Pruebas controladas	5 días	mié 12/08/09	mar 18/08/09	78	Coordinador[50 %],Coordinador IT USAC,IT USAC,Equipo Desarrollo B
80	Implementación de Componentes WEB	12 días	mié 19/08/09	jue 03/09/09		
81	Instalación de componentes	5 días	mié 19/08/09	mar 25/08/09	79	Coordinador[50 %],Coordinador IT USAC,IT USAC,Equipo Desarrollo B
82	Configuración de componentes	2 días	mié 26/08/09	jue 27/08/09	81	Coordinador[50 %],Coordinador IT USAC,IT USAC,Equipo Desarrollo B
83	Pruebas Controladas	5 días	vie 28/08/09	jue 03/09/09	82	Coordinador[50 %],Coordinador IT USAC,IT USAC,Equipo Desarrollo B
84	Implementacion IntraGrid	40 días	vie 04/09/09	lun 02/11/09		
85	Instalación Globus Toolkit	28 días	vie 04/09/09	mié 14/10/09		
86	Instalación de Linux	1 día	vie 04/09/09	vie 04/09/09	48,66	Coordinador[50 %],Coordinador IT USAC,IT USAC
87	Nombres y direcciones	0.5 días	lun 07/09/09	lun 07/09/09	86	Coordinador[50 %],Coordinador IT USAC,IT USAC
88	Certificados	0.5 días	lun 07/09/09	lun 07/09/09	87	Coordinador[50 %],Coordinador IT USAC,IT USAC
89	Usuarios y grupos	0.5 días	mar 08/09/09	mar 08/09/09	88	Coordinador[50 %],Coordinador IT USAC,IT USAC
90	Directorios	0.5 días	mar 08/09/09	mar 08/09/09	89	Coordinador[50 %],Coordinador IT USAC,IT USAC

Id
jul '09
ago '09
sep '09
oct '09
nov '09
dic '09
ene '10
feb '10
mar '10
abr '10
may '10
jun '10
jul '10
ago '10
sep '10
Coordinador[50%],Coordinador IT Ingeniería,IT Unidad Académica,Equipo Desarrollo A
Coordinador[50%],Coordinador IT Ingeniería,IT Unidad Académica,Equipo Desarrollo A
Coordinador[50%],Coordinador IT Ingeniería,IT Unidad Académica,Equipo Desarrollo A
Coordinador[50%],Coordinador IT Ingeniería,IT Unidad Académica,Equipo Desarrollo A
Coordinador[50%],Coordinador IT Ingeniería,IT Unidad Académica,Equipo Desarrollo A
Coordinador[50%],Coordinador IT USNC,IT USNC,Equipo Desarrollo B
Coordinador[50%],Coordinador IT USNC,IT USNC,Equipo Desarrollo B
Coordinador[50%],Coordinador IT USNC,IT USNC,Equipo Desarrollo B
Coordinador[50%],Coordinador IT USNC,IT USNC,Equipo Desarrollo B
Coordinador[50%],Coordinador IT USNC,IT USNC,Equipo Desarrollo B
Coordinador[50%],Coordinador IT USNC,IT USNC,Equipo Desarrollo B
Coordinador[50%],Coordinador IT USNC,IT USNC,Equipo Desarrollo B
Coordinador[50%],Coordinador IT USNC,IT USNC,Equipo Desarrollo B
Coordinador[50%],Coordinador IT USNC,IT USNC,Equipo Desarrollo B
Coordinador[50%],Coordinador IT USNC,IT USNC,Equipo Desarrollo B
Coordinador[50%],Coordinador IT USNC,IT USNC,Equipo Desarrollo B
Coordinador[50%],Coordinador IT USNC,IT USNC,Equipo Desarrollo B
Coordinador[50%],Coordinador IT USNC,IT USNC,Equipo Desarrollo B
Coordinador[50%],Coordinador IT USNC,IT USNC,Equipo Desarrollo B
Coordinador[50%],Coordinador IT USNC,IT USNC
Coordinador[50%],Coordinador IT USNC,IT USNC
Coordinador[50%],Coordinador IT USNC,IT USNC
Coordinador[50%],Coordinador IT USNC,IT USNC
Coordinador[50%],Coordinador IT USNC,IT USNC

Id	Nombre de tarea	Duración	Comienzo	Fin	Predecesor	Nombres de los recursos
91	Instalación de herramientas complementarias	2 días	mié 09/09/09	jue 10/09/09	90	Coordinador[50%],Coordinador IT USAC,IT USAC
92	Instalación de GT3	10 días	vie 11/09/09	vie 25/09/09	91	Coordinador[50%],Coordinador IT USAC,IT USAC
93	Configuración WS- Security	2 días	lun 28/09/09	mar 29/09/09	92	Coordinador[50%],Coordinador IT USAC,IT USAC
94	Configuración GRAM	2 días	mié 30/09/09	jue 01/10/09	93	Coordinador[50%],Coordinador IT USAC,IT USAC
95	Configuración MDS	2 días	vie 02/10/09	lun 05/10/09	94	Coordinador[50%],Coordinador IT USAC,IT USAC
96	Configuración OGSA-DAI	2 días	mar 06/10/09	mié 07/10/09	95	Coordinador[50%],Coordinador IT USAC,IT USAC
97	Pruebas controladas	5 días	jue 08/10/09	mié 14/10/09	96	Coordinador[50%],Coordinador IT USAC,IT USAC
98	Implementación de Componentes WEB	12 días	jue 15/10/09	lun 02/11/09		
99	Instalación de componentes	5 días	jue 15/10/09	jue 22/10/09	97	Coordinador[50%],Coordinador IT USAC,IT USAC
100	Configuración de componentes	2 días	vie 23/10/09	lun 26/10/09	99	Coordinador[50%],Coordinador IT USAC,IT USAC
101	Pruebas Controladas	5 días	mar 27/10/09	lun 02/11/09	100	Coordinador[50%],Coordinador IT USAC,IT USAC
102	Implementacion del Grid Gobal	26 días	vie 04/09/09	lun 12/10/09		
103	Configuracion DMZ	1 día	vie 04/09/09	vie 04/09/09	48,66	Coordinador[50%],Coordinador IT USAC[50%],IT USAC[50%],Coordinador IT Ingeniería,Coordinador IT Unidad Académica,Equipo Desarrollo A,Equipo Desarrollo B,IT Ingeniería,IT Unid
104	Configuracion del Firewall	5 días	lun 07/09/09	vie 11/09/09	103	Coordinador[50%],Coordinador IT USAC[50%],IT USAC[50%],Coordinador IT Ingeniería,Coordinador IT Unidad Académica,Equipo Desarrollo A,Equipo Desarrollo B,IT Ingeniería,IT Unid
105	Definicion de politicas de seguridad	20 días	lun 14/09/09	lun 12/10/09	104	Coordinador[50%],Coordinador IT USAC[50%],IT USAC[50%],Coordinador IT Ingeniería,Coordinador IT Unidad Académica,Equipo Desarrollo A,Equipo Desarrollo B,IT Ingeniería,IT Unid
106	Implementacion Anillo Suroeste	66.49 días	lun 04/01/10	mar 06/04/10		
107	Instalación Globus Toolkit	46.29 días	lun 04/01/10	mar 09/03/10		
108	Instalación de Linux	0.84 días	lun 04/01/10	lun 04/01/10	84,102	Coordinador IT USAC[33%],IT USAC[33%],Coordinador IT Unidad Académica,IT Unidad Académica
109	Nombres y direcciones	0.84 días	lun 04/01/10	mar 05/01/10	108	Coordinador IT USAC[33%],IT USAC[33%],Coordinador IT Unidad Académica,IT Unidad Académica
110	Certificados	0.84 días	mar 05/01/10	mié 06/01/10	109	Coordinador IT USAC[33%],IT USAC[33%],Coordinador IT Unidad Académica,IT Unidad Académica
111	Usuarios y grupos	0.84 días	mié 06/01/10	jue 07/01/10	110	Coordinador IT USAC[33%],IT USAC[33%],Coordinador IT Unidad Académica,IT Unidad Académica
112	Directorios	0.84 días	jue 07/01/10	vie 08/01/10	111	Coordinador IT USAC[33%],IT USAC[33%],Coordinador IT Unidad Académica,IT Unidad Académica
113	Instalación de herramientas complementarias	3.37 días	vie 08/01/10	mié 13/01/10	112	Coordinador IT USAC[33%],IT USAC[33%],Coordinador IT Unidad Académica,IT Unidad Académica
114	Instalación de GT3	16.84 días	mié 13/01/10	vie 05/02/10	113	Coordinador IT USAC[33%],IT USAC[33%],Coordinador IT Unidad Académica,IT Unidad Académica
115	Configuración WS- Security	3.37 días	vie 05/02/10	mié 10/02/10	114	Coordinador IT USAC[33%],IT USAC[33%],Coordinador IT Unidad Académica,IT Unidad Académica
116	Configuración GRAM	3.37 días	mié 10/02/10	mar 16/02/10	115	Coordinador IT USAC[33%],IT USAC[33%],Coordinador IT Unidad Académica,IT Unidad Académica
117	Configuración MDS	3.37 días	mar 16/02/10	vie 19/02/10	116	Coordinador IT USAC[33%],IT USAC[33%],Coordinador IT Unidad Académica,IT Unidad Académica
118	Configuración OGSA-DAI	3.37 días	vie 19/02/10	mié 24/02/10	117	Coordinador IT USAC[33%],IT USAC[33%],Coordinador IT Unidad Académica,IT Unidad Académica
119	Pruebas controladas	8.42 días	mié 24/02/10	mar 09/03/10	118	Coordinador IT USAC[33%],IT USAC[33%],Coordinador IT Unidad Académica,IT Unidad Académica
120	Implementación de Componentes WEB	20.2 días	mar 09/03/10	mar 06/04/10		


Id
sep '09
oct '09
nov '09
dic '09
ene '10
feb '10
mar '10
abr '10
may '10
jun '10
jul '10
ago '10
sep '10
oct '10
nov '10
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120

Id	Nombre de tarea	Duración	Comienzo	Fin	Predecesor	Nombres de los recursos
121	Instalación de componentes	8.42 días	mar 09/03/10	vie 19/03/10	119	Coordinador IT USAC[33%],IT USAC[33%],Coordinador IT Unidad Académica,IT Unidad Académica
122	Configuración de componentes	3.37 días	vie 19/03/10	jue 25/03/10	121	Coordinador IT USAC[33%],IT USAC[33%],Coordinador IT Unidad Académica,IT Unidad Académica
123	Pruebas Controladas	8.42 días	jue 25/03/10	mar 06/04/10	122	Coordinador IT USAC[33%],IT USAC[33%],Coordinador IT Unidad Académica,IT Unidad Académica
124	Implementacion Anillo Noroeste	67.34 días	lun 04/01/10	mié 07/04/10		
125	Instalación Globus Toolkit	47.14 días	lun 04/01/10	mié 10/03/10		
126	Instalación de Linux	1.68 días	lun 04/01/10	mar 05/01/10	84,102	Coordinador IT USAC[33%],IT USAC[33%],Coordinador IT Unidad Académica,IT Unidad Académica
127	Nombres y direcciones	0.84 días	mar 05/01/10	mié 06/01/10	126	Coordinador IT USAC[33%],IT USAC[33%],Coordinador IT Unidad Académica,IT Unidad Académica
128	Certificados	0.84 días	mié 06/01/10	jue 07/01/10	127	Coordinador IT USAC[33%],IT USAC[33%],Coordinador IT Unidad Académica,IT Unidad Académica
129	Usuarios y grupos	0.84 días	jue 07/01/10	vie 08/01/10	128	Coordinador IT USAC[33%],IT USAC[33%],Coordinador IT Unidad Académica,IT Unidad Académica
130	Directorios	0.84 días	vie 08/01/10	lun 11/01/10	129	Coordinador IT USAC[33%],IT USAC[33%],Coordinador IT Unidad Académica,IT Unidad Académica
131	Instalación de herramientas complementarias	3.37 días	lun 11/01/10	jue 14/01/10	130	Coordinador IT USAC[33%],IT USAC[33%],Coordinador IT Unidad Académica,IT Unidad Académica
132	Instalación de GT3	16.84 días	jue 14/01/10	lun 08/02/10	131	Coordinador IT USAC[33%],IT USAC[33%],Coordinador IT Unidad Académica,IT Unidad Académica
133	Configuración WS- Security	3.37 días	lun 08/02/10	jue 11/02/10	132	Coordinador IT USAC[33%],IT USAC[33%],Coordinador IT Unidad Académica,IT Unidad Académica
134	Configuración GRAM	3.37 días	jue 11/02/10	mar 16/02/10	133	Coordinador IT USAC[33%],IT USAC[33%],Coordinador IT Unidad Académica,IT Unidad Académica
135	Configuración MDS	3.37 días	mar 16/02/10	lun 22/02/10	134	Coordinador IT USAC[33%],IT USAC[33%],Coordinador IT Unidad Académica,IT Unidad Académica
136	Configuración OGSA-DAI	3.37 días	lun 22/02/10	jue 25/02/10	135	Coordinador IT USAC[33%],IT USAC[33%],Coordinador IT Unidad Académica,IT Unidad Académica
137	Pruebas controladas	8.42 días	jue 25/02/10	mié 10/03/10	136	Coordinador IT USAC[33%],IT USAC[33%],Coordinador IT Unidad Académica,IT Unidad Académica
138	Implementación de Componentes WEB	20.2 días	mié 10/03/10	mié 07/04/10		
139	Instalación de componentes	8.42 días	mié 10/03/10	lun 22/03/10	137	Coordinador IT USAC[33%],IT USAC[33%],Coordinador IT Unidad Académica,IT Unidad Académica
140	Configuración de componentes	3.37 días	lun 22/03/10	jue 25/03/10	139	Coordinador IT USAC[33%],IT USAC[33%],Coordinador IT Unidad Académica,IT Unidad Académica
141	Pruebas Controladas	8.42 días	jue 25/03/10	mié 07/04/10	140	Coordinador IT USAC[33%],IT USAC[33%],Coordinador IT Unidad Académica,IT Unidad Académica
142	Implementacion Anillo Sureste	67.34 días	lun 04/01/10	mié 07/04/10		
143	Instalación Globus Toolkit	47.14 días	lun 04/01/10	mié 10/03/10		
144	Instalación de Linux	1.68 días	lun 04/01/10	mar 05/01/10	84,102	Coordinador IT USAC[33%],IT USAC[33%],Coordinador IT Unidad Académica,IT Unidad Académica
145	Nombres y direcciones	0.84 días	mar 05/01/10	mié 06/01/10	144	Coordinador IT USAC[33%],IT USAC[33%],Coordinador IT Unidad Académica,IT Unidad Académica
146	Certificados	0.84 días	mié 06/01/10	jue 07/01/10	145	Coordinador IT USAC[33%],IT USAC[33%],Coordinador IT Unidad Académica,IT Unidad Académica
147	Usuarios y grupos	0.84 días	jue 07/01/10	vie 08/01/10	146	Coordinador IT USAC[33%],IT USAC[33%],Coordinador IT Unidad Académica,IT Unidad Académica
148	Directorios	0.84 días	vie 08/01/10	lun 11/01/10	147	Coordinador IT USAC[33%],IT USAC[33%],Coordinador IT Unidad Académica,IT Unidad Académica
149	Instalación de herramientas complementarias	3.37 días	lun 11/01/10	jue 14/01/10	148	Coordinador IT USAC[33%],IT USAC[33%],Coordinador IT Unidad Académica,IT Unidad Académica
150	Instalación de GT3	16.84 días	jue 14/01/10	lun 08/02/10	149	Coordinador IT USAC[33%],IT USAC[33%],Coordinador IT Unidad Académica,IT Unidad Académica

Id
ene '10
feb '10
mar '10
abr '10
may '10
jun '10
jul '10
ago '10
sep '10
oct '10
nov '10
dic '10
ene '11
feb '11
mar '11
Coordinador IT U9AC[839#],IT U9AC[839#],Coordinador IT Unidad Académica,IT Unidad Académica
Coordinador IT U9AC[839#],IT U9AC[839#],Coordinador IT Unidad Académica,IT Unidad Académica
Coordinador IT U9AC[839#],IT U9AC[839#],Coordinador IT Unidad Académica,IT Unidad Académica
Coordinador IT U9AC[839#],IT U9AC[839#],Coordinador IT Unidad Académica,IT Unidad Académica
Coordinador IT U9AC[839#],IT U9AC[839#],Coordinador IT Unidad Académica,IT Unidad Académica
Coordinador IT U9AC[839#],IT U9AC[839#],Coordinador IT Unidad Académica,IT Unidad Académica
Coordinador IT U9AC[839#],IT U9AC[839#],Coordinador IT Unidad Académica,IT Unidad Académica
Coordinador IT U9AC[839#],IT U9AC[839#],Coordinador IT Unidad Académica,IT Unidad Académica
Coordinador IT U9AC[839#],IT U9AC[839#],Coordinador IT Unidad Académica,IT Unidad Académica
Coordinador IT U9AC[839#],IT U9AC[839#],Coordinador IT Unidad Académica,IT Unidad Académica
Coordinador IT U9AC[839#],IT U9AC[839#],Coordinador IT Unidad Académica,IT Unidad Académica
Coordinador IT U9AC[839#],IT U9AC[839#],Coordinador IT Unidad Académica,IT Unidad Académica
Coordinador IT U9AC[839#],IT U9AC[839#],Coordinador IT Unidad Académica,IT Unidad Académica
Coordinador IT U9AC[839#],IT U9AC[839#],Coordinador IT Unidad Académica,IT Unidad Académica
Coordinador IT U9AC[839#],IT U9AC[839#],Coordinador IT Unidad Académica,IT Unidad Académica
Coordinador IT U9AC[839#],IT U9AC[839#],Coordinador IT Unidad Académica,IT Unidad Académica
Coordinador IT U9AC[839#],IT U9AC[839#],Coordinador IT Unidad Académica,IT Unidad Académica
Coordinador IT U9AC[839#],IT U9AC[839#],Coordinador IT Unidad Académica,IT Unidad Académica
Coordinador IT U9AC[839#],IT U9AC[839#],Coordinador IT Unidad Académica,IT Unidad Académica
Coordinador IT U9AC[839#],IT U9AC[839#],Coordinador IT Unidad Académica,IT Unidad Académica
Coordinador IT U9AC[839#],IT U9AC[839#],Coordinador IT Unidad Académica,IT Unidad Académica
Coordinador IT U9AC[839#],IT U9AC[839#],Coordinador IT Unidad Académica,IT Unidad Académica
Coordinador IT U9AC[839#],IT U9AC[839#],Coordinador IT Unidad Académica,IT Unidad Académica
Coordinador IT U9AC[839#],IT U9AC[839#],Coordinador IT Unidad Académica,IT Unidad Académica
Coordinador IT U9AC[839#],IT U9AC[839#],Coordinador IT Unidad Académica,IT Unidad Académica
Coordinador IT U9AC[839#],IT U9AC[839#],Coordinador IT Unidad Académica,IT Unidad Académica
Coordinador IT U9AC[839#],IT U9AC[839#],Coordinador IT Unidad Académica,IT Unidad Académica
Coordinador IT U9AC[839#],IT U9AC[839#],Coordinador IT Unidad Académica,IT Unidad Académica
Coordinador IT U9AC[839#],IT U9AC[839#],Coordinador IT Unidad Académica,IT Unidad Académica
Coordinador IT U9AC[839#],IT U9AC[839#],Coordinador IT Unidad Académica,IT Unidad Académica

Id	Nombre de tarea	Duración	Comienzo	Fin	Predecesor	Nombres de los recursos
151	Configuración WS- Security	3.37 días	lun 08/02/10	jue 11/02/10	150	Coordinador IT USAC[33%],IT USAC[33%],Coordinador IT Unidad Académica,IT Unidad Académica
152	Configuración GRAM	3.37 días	jue 11/02/10	mar 16/02/10	151	Coordinador IT USAC[33%],IT USAC[33%],Coordinador IT Unidad Académica,IT Unidad Académica
153	Configuración MDS	3.37 días	mar 16/02/10	lun 22/02/10	152	Coordinador IT USAC[33%],IT USAC[33%],Coordinador IT Unidad Académica,IT Unidad Académica
154	Configuración OGSA-DAI	3.37 días	lun 22/02/10	jue 25/02/10	153	Coordinador IT USAC[33%],IT USAC[33%],Coordinador IT Unidad Académica,IT Unidad Académica
155	Pruebas controladas	8.42 días	jue 25/02/10	mié 10/03/10	154	Coordinador IT USAC[33%],IT USAC[33%],Coordinador IT Unidad Académica,IT Unidad Académica
156	Implementación de Componentes WEB	20.2 días	mié 10/03/10	mié 07/04/10		
157	Instalación de componentes	8.42 días	mié 10/03/10	lun 22/03/10	155	Coordinador IT USAC[33%],IT USAC[33%],Coordinador IT Unidad Académica,IT Unidad Académica
158	Configuración de componentes	3.37 días	lun 22/03/10	jue 25/03/10	157	Coordinador IT USAC[33%],IT USAC[33%],Coordinador IT Unidad Académica,IT Unidad Académica
159	Pruebas Controladas	8.42 días	jue 25/03/10	mié 07/04/10	158	Coordinador IT USAC[33%],IT USAC[33%],Coordinador IT Unidad Académica,IT Unidad Académica

Id
ene '10
feb '10
mar '10
abr '10
may '10
jun '10
jul '10
ago '10
sep '10
oct '10
nov '10
dic '10
ene '11
feb '11
mar '11
151 Coordinador IT USAC|839kj,IT USAC|839kj,Coordinador IT Unidad Académica,IT Unidad Académica
162 Coordinador IT USAC|839kj,IT USAC|839kj,Coordinador IT Unidad Académica,IT Unidad Académica
163 Coordinador IT USAC|839kj,IT USAC|839kj,Coordinador IT Unidad Académica,IT Unidad Académica
164 Coordinador IT USAC|839kj,IT USAC|839kj,Coordinador IT Unidad Académica,IT Unidad Académica
165 Coordinador IT USAC|839kj,IT USAC|839kj,Coordinador IT Unidad Académica,IT Unidad Académica
166
167 Coordinador IT USAC|839kj,IT USAC|839kj,Coordinador IT Unidad Académica,IT Unidad Académica
168 Coordinador IT USAC|839kj,IT USAC|839kj,Coordinador IT Unidad Académica,IT Unidad Académica
169 Coordinador IT USAC|839kj,IT USAC|839kj,Coordinador IT Unidad Académica,IT Unidad Académica

- **Recurso humano:**

    - Coordinador de Proyecto: será el encargado de la administración global del proyecto, de realizar la comunicación con las diversas áreas involucradas, la coordinación de los grupos de trabajo y de la gestión del presupuesto.

    - Coordinador de Área: será el responsable de la gestión y coordinación dentro de su grupo de trabajo, trasladar sus opiniones al coordinador general y supervisar el cumplimiento de los compromisos.

    - Grupo de Informática: se creará un grupo de informática por cada unidad académica o administrativa que participe en el proyecto, siendo responsable de las tareas de implementación de los módulos, trasladar sus opiniones al coordinador del área y de la realización del control de calidad y pruebas de funcionalidad.

    - Equipo de Desarrollo: se crearán dos grupos de desarrollo, para realizar las tareas de programación, control de calidad, pruebas e implementación de los componentes web para la administración del *Grid*. Adicionalmente, finalizado el desarrollo, compartirán las tareas de implementación de los módulos con los grupos de informática.

CONCLUSIONES

1. Las organizaciones no deben considerar al *Grid* como una solución por sí misma. Deben verla como una forma de mejorar la efectividad de la infraestructura que poseen o como una forma de mejorar el proceso del negocio a través de la transformación, o quizá como una oportunidad de innovar, que puede beneficiar el negocio.

2. Existen cuatro preguntas básicas que debe hacerse una empresa antes de aventurarse a desarrollar una aplicación *Grid*, ¿Cuáles son los beneficios que desea alcanzar la organización? ¿Cuál es el alcance que se desea alcanzar con el *Grid*, en el corto y mediano plazo? ¿Qué aspectos del *Grid* deberían ser explotados primero en la organización? ¿Posee la organización las habilidades necesarias, tanto a nivel técnico como de recursos humanos?

3. Actualmente en Guatemala, la utilización de *Grid Computing* es nula, debido a que es una tecnología relativamente nueva y a la complejidad que supone el diseño e instalación de un ambiente de este tipo.

4. La implementación de un *Grid* en la Universidad de San Carlos de Guatemala, puede proporcionar una herramienta para la investigación científica, como la simulación de fenómenos físicos, simulación de procesos químicos, procesamiento masivo de cálculos matemáticos, entre otros, mediante la optimización de los recursos computacionales del Campus.

5. La Universidad cuenta con la infraestructura física, de red y de equipo necesaria, para llevar a cabo un proyecto de esta magnitud.

RECOMENDACIONES

1. Hacer uso a la Red de Servicios Integrados, de manera que la tecnología pueda ser aprovechada por todas las unidades académicas, y no solamente por las que tienen relación directa con el uso de la misma. En la actualidad esta red, a pesar de estar ya en funcionamiento, no es utilizada por la mayoría de unidades académicas.

2. Tomar en cuenta la propuesta de la implementación de un *Grid*, ya sea utilizando esta propuesta o alguna propuesta propia, ya que los beneficios para proyectos de investigación y cooperación con otras Universidades podría traer frutos importantes a la USAC. Inicialmente, la Universidad puede adicionarse a proyectos existentes como el de la búsqueda de la cura del cáncer u otras enfermedades, mientras desarrolla sus propios proyectos.

3. Sugerir que las diferentes unidades académicas promuevan el uso de la tecnología, debido a que cada vez está tomando mayor importancia dentro de la sociedad. Y especialmente en el ámbito laboral, donde los conocimientos tecnológicos, dan un valor agregado a los conocimientos teóricos y prácticos del estudiante.

4. Crear un comité de dirección, integrado por un miembro de cada unidad académica y administrativa, que será el responsable de dar seguimiento a la implementación del proyecto. Una vez finalizada la implementación, el comité asumirá el mismo papel que el coordinador de proyecto.

5. Crear un comité científico, integrado por un miembro de cada unidad académica y administrativa, preferiblemente que esté involucrado en el desarrollo de un proyecto, que será el responsable de la gestión y coordinación de los proyectos científicos y de investigación que puedan ser integrados al *Grid*.

BIBLIOGRAFÍA

1. **Buyya Rajkumar y Mark Baker** The Grid: A New Network Computing Infrastructure [Libro]. - India : IEEE/ACM International Workshop on Grid Computing, 2000.

2. **Cabrera Luis Felipe y Chris Kurt** Web Services Architecture and Its Specifications: Essentials for Understanding WS [Libro]. - Estados Unidos : Microsoft Press, 2005.

3. **Easton John** Geographically Dispersed Grid, Part 1: Aligning Computation and Data [En línea]. - Noviembre de 2004. - http://www-128.ibm.com/developerworks/*Grid*/library/gr-gdg1/?ca=dgr-gridw09GeoGridP1.

4. **Foster Ian y Carl Kesselman** The Grid: Blueprint for a New Computing Infraestructure [Libro]. - Estados Unidos : Morgan Kaufmann, 2003. - Segunda.

5. **Foster Ian y otros** Grid Services for Distributed System Integration [En línea]. - Junio de 2002. - http://www.globus.org/alliance/publications/papers/ieee-cs-2.pdf.

6. **Foster Ian y otros** The Anatomy of the Grid: Enabling Scalable Virtual Organizations [Libro]. - Estados Unidos : International Journal of Supercomputer Applications, 2001.

7. **Foster Ian y otros** The Open Grid Services Architecture, Version 1.0 [En línea]. - Enero de 2005. - http://www.gridforum.org/documents/GFD.30.pdf.

8. **Foster Ian y otros** The Physiology of the Grid [En línea]. - Junio de 2002. - http://www.globus.org/alliance/publications/papers/ogsa.pdf.

9. **Foster Ian y otros.** What is the Grid? A Three Point Checklist [Libro]. - Chicago : Argonne National Laboratory & University of Chicago, 2002.

10. **Globus Alliance** A Globus Primer, Describing Globus Toolkit Version 4 [En línea]. - 2006. - http://globus.org/toolkt/docs/gt4_primer_0.6.pdf.

11. **Jacob Bart y otros** On Demand Operating Environment: Creating Business Flexibility [Libro]. - Estados Unidos : Redbooks, 2004.

12. **Jacob Bart y otros.** Introduction to Grid Computing [Libro]. - EEUU : Redbooks, 2005. - 2a : págs. 3-4.

13. **Mahan Michael y Otros** XML Protocol Working Group Charter [En línea]. - Febrero de 2005. - http://www.w3.org/2005/07/XML-Protocol-Charter.

14. **Mendez Ferreiro Brais** Estructura en XML [En línea]. - Febrero de 2004. - http://www.elrincondelprogramador.com/default.asp?pag=articulos/leer.asp&id=21.

15. **Newcomer Eric** Understanding Web Services [Libro]. - Estados Unidos : Addison-Wesley, 2002.

16. **Maliksetian y otros** Design and Implementation of an Enterprise Grid [Libro]. - Estados Unidos : IBM Systems Journal, 2004.

17. **Universidad de San Carlos de Guatemala** Misión y Visión de la USAC [En línea]. - 2004. - http://www.usac.edu.gt.

18. **W3C** Guía Breve de Servicios Web [En línea]. - Febrero de 2006. - http://www.w3c.es/divulgacion/guiasbreves/ServiciosWeb.

19. **W3C** Web Service Architecture [En línea]. - Febrero de 2004. - http://www.w3.org/TR/ws-arch/.

20. **Wikipedia** SOAP [En línea]. - Mayo de 2006. - http://en.wikipedia.org/wiki/SOAP.

21. **Wikipedia** WS-Security [En línea]. - Mayo de 2006. - http://en.wikipedia.org/wiki/WS-Security.

22. **Wikipedia** XSLT [En línea]. - Mayo de 2006. - http://en.wikipedia.org/wiki/Xslt.

[1] Buyya, Rajkumar y Mark Baker. *The Grid: A New Network Computing Infrastructure*. (India: IEEE/ACM International Workshop on Grid Computing, 2000) pp. 2-4

[2] Ferreira, Luis y otros. *Introduction to Grid Computing whit Globus.* Segunda Edición. Estados Unidos: Redbooks, 2003

[3] Foster, Ian y otros. *The Open Grid Service Architecture, Version 1.0.* Global Grid Forum. 2003

[4] Buyya, Rajkumar y Mark Baker. *The Grid: A New Network Computing Infrastructure*. (India: IEEE/ACM International Workshop on Grid Computing, 2000) pp. 2-4

[5] Jacob, Bart y otros. *Introduction to Grid Computing*. (2ª. Edición; EEUU: Redbooks, 2005) pp. 3-4

[6] Foster, Ian. *"What is the Grid? A Three Point Checklist"*. **Argonne National Laboratory & University of Chicago, Julio 2002.** pp. 1-2

[7] Ibid. pp. 2-4

[8] Foster, Ian y otros. "*The Anatomy of the Grid: Enabling Scalable Virtual Organizations"*. **International Journal of Supercomputer Applications**. 2001.

[9] Jacob, Bart y otros, *op. cit.*, pp. 3-4

[10] Easton, John. *"Geographically Dispersed Grid, Part 1: Aligning Computation and Data"*. **http://www-128.ibm.com/developerworks/grid/library/gr-gdg1/?ca=dgr-gridw09GeoGridP1**. Noviembre 2004.

[11] Foster, Ian. *op. cit.* p. 2

[12] Foster, Ian. *op. cit*. p. 2

[13] Foster, Ian. *op. cit.* p. 3

[14] Jacob, Bart y otros. *op. cit.* p. 20

[15] Jacob, Bart y otros. *op. cit.* p. 20

[16] Jacob, Bart y otros. *op. cit.* pp. 20 - 22

[17] Jacob, Bart y otros. *op. cit.* p. 22

[18] Jacob, Bart y otros. *op. cit.* pp. 22-23

[19] Jacob, Bart y otros. *op. cit.* p. 23

[20] Jacob, Bart y otros. *op. cit.* pp. 23 -24

[21] Jacob, Bart y otros. *op. cit.* pp. 24 - 26

[22] Jacob, Bart y otros. *op. cit.* p. 26

[23] Jacob, Bart y otros. *op. cit.* p. 26

[24] Jacob, Bart y otros. *op. cit.* pp. 26 - 27

[25] Jacob, Bart y otros. *op. cit.* pp. 27 - 28

[26] Jacob, Bart y otros. *op. cit.* p. 28

[27] Jacob, Bart y otros. *op. cit.* pp. 28 - 29

[28] Jacob, Bart y otros. *op. cit.* p. 29

[29] Jacob, Bart y otros. *op. cit.* p. 29

[30] Foster, Ian y Carl Kesselman. "*The Grid: Blueprint for a New Computing Infraestructure*". (Segunda Edición. Estados Unidos. Morgan Kaufmann. 2003) pp. 21 - 23

[31] Foster, Ian y Carl Kesselman. *op. cit.* p. 23

[32] Foster, Ian y Carl Kesselman. *op. cit.* pp. 23 -24

[33] Foster, Ian y Carl Kesselman. *op. cit.* pp. 24 - 25

[34] Foster, Ian y Carl Kesselman. *op. cit.* pp. 25 - 26

[35] Foster, Ian y otros. *"The Anatomy of the Grid: Enabling Scalable Virtual Organizations."* (International Journal of Supercomputer Applications. 2001) pp. 5 - 6

[36] Ibid. pp. 6 – 7

[37] Ibid. pp. 7 – 8

[38] Ibid. pp. 9 – 10

[39] Ibid. pp. 10 -11

[40] Ibid. pp. 11 – 13

[41] Ibíd. p. 13

[42] Mahan, Michael y Otros. *"XML Protocol Working Group Charter"* http://www.w3.org/2005/07/XML-Protocol-Charter, Febrero, 2005

[43] Cabrera ,Luis Felipe y Chris Kurt. *"Web Services Architecture and Its Specifications: Essentials for Understanding WS"*. Microsoft Press, 2005. pp. 1

[44] W3C. *"Web Service Architecture"*. **http://www.w3.org/TR/ws-arch/**. Febrero 2004

[45] W3C. "Guía Breve de Servicios Web". **http://www.w3c.es/divulgacion/guiasbreves/ServiciosWeb**. Febrero 2006.

[46] Easton, John. *"Geographically Dispersed Grid, Part 1: Aligning Computation and Data"*. **http://www-128.ibm.com/developerworks/grid/library/gr-gdg1/?ca=dgr-gridw09GeoGridP1**. Nov 2004.

[47] Jacob, Bart y otros. "*On Demand Operating Environment: Creating Business Flexibility*". Primera Edición. Estados Unidos: Redbooks, 2004. p. 223

[48] Ibid. pp. 223 – 224

[49] Ibid. pp. 217 – 218

[50] Mendez Ferreiro, Brais. "Estructura en XML". **http://www.elrincondelprogramador.com/default.asp?pag=articulos/leer.asp&id=21**. Febrero 2004

[51] Wikipedia. *"XSLT"*. **http://en.wikipedia.org/wiki/Xslt**. Mayo 2006.

[52] Wikipedia. *"SOAP"*. **http://en.wikipedia.org/wiki/SOAP**. Mayo 2006.

[53] Newcomer, Eric. *"Understanding Web Services"*. Addison-Wesley. 2002. pp. 27 - 28

[54] Newcomer, Eric. *"Understanding Web Services"*. Addison-Wesley. 2002. pp. 24 - 25

[55] Newcomer, Eric. *"Understanding Web Services"*. Addison-Wesley. 2002. pp. 29 – 31

[56] Wikipedia. *"WS-Security"*. **http://en.wikipedia.org/wiki/WS-Security**. Mayo 2006.

[57] Jacob, Bart y otros. "*On Demand Operating Environment: Creating Business Flexibility"*. Primera Edición. Estados Unidos: Redbooks, 2004. p. 224

[58] Jacob, Bart y otros. "*On Demand Operating Environment: Creating Business Flexibility"*. Primera Edición. Estados Unidos: Redbooks, 2004. pp. 224 -225

[59] Foster, Ian y otros. "*Grid Services for Distributed System Integration*". **http://www.globus.org/alliance/publications/papers/ieee-cs-2.pdf**. Junio 2002. pp. 37

[60] Foster, Ian y otros. "*The Open Grid Services Architecture, Version 1.0"*. **http://www.gridforum.org/documents/GFD.30.pdf**. Enero 2005. pp. 4

[61] Foster, Ian y otros. "*The Physiology of the Grid*". **http://www.globus.org/alliance/publications/papers/ogsa.pdf.** Junio 2002. pp. 10 - 11

[62] Ibid. pp. 10 - 11

[63] Ibid. pp. 12 – 13

[64] Ibid. pp. 13 – 14

[65] Ibid. pp. 14 - 15

[66] Ibid. pp. 14 -15

[67] Ibid. pp. 15 -17

[68] Universidad de San Carlos de Guatemala. "Misión y Visión de la USAC". **http://www.usac.edu.gt**. 2004.

[69] Globus Alliance. "A Globus Primer, Describing Globus Toolkit Version 4". **http://globus.org/toolkt/docs/gt4_primer_0.6.pdf**. 2006. Pp. 28, 43, 45-48, 50.

[70] Maliksetian y otros. "Design and Implementation of an Enterprise Grid". **IBM Systems Journal.** Vol 43, No, 4. 2004. pp. 648

[71] Ibid. pp. 648 - 649

[72]Ibid. pp. 650

[73] Ibid. pp. 650

[74] Ibid. pp. 651

[75] Ibid. pp. 652

[76] Ibid. pp. 653- 654

[77] Ibid. pp. 654

[78] Ibid. pp. 655