

ARQUITECTURA DE VISUALIZACIÓN PARA APLICACIONES CIENTÍFICAS COMO SERVICIO

CARLOS DAVID PARADA ANDRADE

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICOMECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA
2017

ARQUITECTURA DE VISUALIZACIÓN PARA APLICACIONES CIENTÍFICAS COMO SERVICIO

CARLOS DAVID PARADA ANDRADE

Trabajo de investigación para optar el título de Ingeniero de Sistemas

Director:

PhD CARLOS JAIME BARRIOS HERNÁNDEZ

Doctor en Informática

Codirector:

PhD GABRIEL RODRIGO PEDRAZA FERREIRA

Doctor en Informática

Asesor:

DEYBERTH HERNANDO RIAÑO NUÑEZ

Magister en Informática

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICOMECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA
2017

Dedicatoria

A mi familia, por apoyarme en mis proyectos y su compañía fortalecedora.
A mi padre, por su apoyo incondicional y sus grandes enseñanzas para mantenerme
enfocado.
A mi madre, por su inmenso amor, su cariño sincero y su dulzura que tranquiliza.
A mis hermanos y mis sobrinos por compartirme su alegría y felicidad.

Agradecimientos

Este trabajo no hubiera sido posible sin mi director Carlos Jaime Barrios, siempre comprometido y dispuesto

A mi codirector por sus consejos y enseñanzas.

Al grupo de investigación CAGE que me acogió para enseñarme lo bueno que es trabajar en grupo y aprender en conjunto.

CONTENIDO

INTRODUCCIÓN	13
1. PLANTEAMIENTO DEL PROBLEMA	15
2. OBJETIVOS	16
2.1 OBJETIVO GENERAL	16
2.2 OBJETIVOS ESPECÍFICOS.....	16
3. MARCO TEÓRICO	17
3.1 SISTEMAS HETEROGÉNEOS.....	17
3.2 COMPUTACIÓN EN LA NUBE	17
3.2.1 Arquitectura en la Nube.....	17
3.2.2 Virtualización en la nube.....	18
3.3 GRID.....	19
3.3.1 Características.....	19
3.3.2 Nvidia GRID	20
3.4 VIRTUALIZACIÓN.....	20
3.4.1 Escritorios	21
3.4.2 Aplicaciones	21
3.5 VISUALIZACIÓN.....	22
3.6 VISUALIZACIÓN CIENTÍFICA	23
3.6.1 Visualización de Volúmenes.....	23
3.6.2 Visualización de Flujo	23
3.6.3 Visualización como Servicio	24
3.7 APLICACIONES	25
3.7.1 Científicas	25
3.7.2 Videojuegos.....	27
4. MARCO TÉCNICO	28
4.1 HARDWARE.....	28
4.1.1 Servidor	28
4.1.2 Computador	29
4.2 SOFTWARE.....	29
4.2.1 Solución VMware.....	29
4.2.2 Docker	32
4.2.3 Sistema de Ventanas X11.....	34
5. METODOLOGÍA	35
6. DESARROLLO DEL PROYECTO	37
6.1 COMPONENTES	37
6.1.1 Hardware.....	37

6.1.2	Software	37
6.2	IMPLEMENTACIÓN	38
6.2.1	VMware	38
6.2.2	Docker	40
6.2.3	Híbrida	42
6.3	APLICACIONES	43
6.3.1	Características	43
6.3.2	Restricciones	44
6.4	MÓDULO ACCESO	45
6.5	CONFIGURACIÓN	47
6.5.1	Servidor Docker	47
6.5.2	Frontend	51
6.5.3	Clientes	51
6.6	PROTOTIPO	55
7.	CONCLUSIONES	60
8.	RECOMENDACIONES Y TRABAJO FUTURO	61
9.	LIMITACIONES	62
	REFERENCIAS	63
	BIBLIOGRAFÍA	65

ÍNDICE DE FIGURAS

Figura 1. Nvidia GRID Gráficos Remotos SDK.....	20
Figura 2. Paraview Arquitectura [9]	26
Figura 3. Integración VMWare vSphere / Horizon con la Nvidia Grid.	32
Figura 4. Arquitectura VMWare.....	39
Figura 5. Arquitectura Docker	41
Figura 6. Flujo de comunicación en una aplicación vía Sistema de Ventanas X11	46
Figura 7. Flujo de comunicación en una aplicación web en un contenedor.	46
Figura 8. Esquema del Componente de Visualización Docker	47
Figura 9. Configuración PuTTY	52
Figura 10. Acceder a las Preferencias en Xquartz	53
Figura 11. Cuadro de Preferencias de XQuartz	53
Figura 12. Prototipo de Inicio de Sesión	55
Figura 13. Prototipo Tablero Administrador de Docker	56
Figura 14. Prototipo Inicio Usuario de Docker	56
Figura 15. Prototipo Contenedores Disponibles en Solución Docker	57
Figura 16. Prototipo para el seguimiento de un contenedor	58
Figura 17. Prototipo apartado de Estadísticas del Contenedor.....	58
Figura 18. Prototipo Imágenes Disponibles en Solución Docker	59

ÍNDICE DE TABLA

Tabla 1. Características Servidor	28
Tabla 2. Características Nvidia Grid K2	28
Tabla 3. Características Computador Administrador	29
Tabla 4. Especificaciones GTX 660	29
Tabla 5. Lista de Componentes Software	38
Tabla 6. Infraestructura Nativa VMWare.....	40
Tabla 7. Infraestructura Nativa Docker	42
Tabla 8. Infraestructura VM Docker en VMWare	42
Tabla 9. Características de Paraview en Docker	43
Tabla 10. Características de Netbeans en Docker.....	44
Tabla 11. Estructura archivo Dockerfile	48

RESUMEN

TÍTULO: ARQUITECTURA DE VISUALIZACIÓN PARA APLICACIONES CIENTÍFICAS COMO SERVICIO*

AUTOR: CARLOS DAVID PARADA ANDRADE**

PALABRAS CLAVE: Visualización, Aplicaciones, Docker

DESCRIPCIÓN:

La utilización de aplicaciones hace que las tareas de adquisición, manipulación y análisis de datos sean mucho más fáciles, además la interpretación de la información generada es la fuente para producir más conocimiento científico. El acceso por parte los estudiantes e investigadores a estas aplicaciones generalmente enfrenta algunas barreras, desde el acceso a una maquina con las características de cómputo de alto rendimiento hasta la imposibilidad de ser usada por múltiples usuarios.

Debido esto se planteó dar solución a estos problemas implementando, en un Sistema Heterogéneo, un servicio de visualización de aplicaciones científicas e integrándole una interfaz de usuario, que facilitara su acceso, donde se lograra desplegar gráficamente aplicaciones científicas para dar solución a las problemáticas.

Para lograr esta implementación se empezó inicialmente con el estudio de los componentes hardware y software necesarios para el servicio de visualización de aplicaciones científicas, luego, con el análisis de las aplicaciones a implementar, se determinó las características y restricciones, enseguida se plantea un componente de acceso e interacción con el servicio, y finalmente se definieron los criterios de configuración de cada componente que se propuso en las etapas anteriores. Todo esto con la finalidad de desarrollar un prototipo funcional, donde se pueda validar de manera global todos los componentes y su comportamiento al integrar en un Sistema Heterogéneo.

* Trabajo de Grado en la Modalidad de Investigación

** Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería de Sistemas e Informática. Director: PhD. Carlos Jaime Barrios. Codirector: PhD. Gabriel Rodrigo Pedraza

ABSTRACT

TITLE: Visualization Architecture for Scientific Applications as a Service[†]

AUTHOR: CARLOS DAVID PARADA ANDRADE^{**}

KEYWORDS: Visualization, Applications, Docker

DESCRIPTION:

The use of applications makes data acquisition, manipulation, and analysis tasks much easier, in addition, the interpretation of the information generated is the source to produce more scientific knowledge. The access of students and researchers to these applications generally faces some barriers, from the access to a machine with the characteristics of high performance computing, to the impossibility of being used by multiple users.

Due to this, it was proposed to solve these problems by implementing, in a Heterogeneous System, a service of visualization of scientific applications and integrating to this a user interface, that would facilitate its access, where it was possible to graphically display scientific applications to solve the problems.

To achieve this implementation, this was initially started with the study of the hardware and software components necessary for the service of visualization of scientific applications, then, with the analysis of the applications to be implemented, the characteristics and restrictions were determined, next, an access and interaction component was proposed with the service, and finally the configuration criteria of each component that were proposed in the previous stages were defined. All with the purpose of developing a functional prototype, where it is possible to validate in a global way all the components and their behavior when integrating in a Heterogeneous System.

[†] Undergraduate final Project, research modality

^{**} School of Physical-Mechanical Engineering. Department of Systems Engineering and Informatics. Advisor: PhD. Carlos Jaime Barrios. Co-advisor: PhD. Gabriel Rodrigo Pedraza.

INTRODUCCIÓN

La utilización de herramientas por parte del hombre para la solución de problemas siempre ha sido una ventaja evolutiva para él, a medida que la humanidad avanza los problemas son diferentes y se relacionan con el entorno en que se encuentran, actualmente las computadoras y aplicaciones científicas son las utilizadas como herramientas para ayudar a tomar decisiones e interpretar datos en diferentes ambientes.

La utilización de computadoras cada vez más equipadas y construidas para tratar la información mediante aplicaciones científicas se ven restringidas por el acceso del personal científico a máquinas con alta capacidad de cómputo, la falta de la misma aplicación y la configuración pertinente para la correcta utilización de esta.

La necesidad de estandarizar el acceso de la información que actualmente está ocurriendo viene acompañada del acceso a las herramientas para el tratamiento de la misma, pues las utilidades de estos datos sirven para dar solución a problemas que actualmente se están analizando e intentando darle solución.

El entorno para la visualización de aplicaciones surgió luego de la posibilidad de la visualización de escritorios, que contaban con un sistema operativo y una cantidad de aplicaciones pre-instaladas del mismo, donde luego se instalaba la aplicaciones que se necesitaba, después de haber configurado la máquina físicamente se procedía a la utilización de software que posibilitaba la forma de hacer visualizar y operar de forma remota la máquina configurada de tal forma que se puedan aprovechar las aplicaciones, pero el rendimiento de la aplicaciones se veía restringida por la capacidad de la máquina, además, no existía la posibilidad de que fuera utilizada por múltiples usuarios, y la calidad de la visualización y su fluidez eran proporcionales a la calidad de la conexión a internet y el nivel de animación que necesitaba.

Partiendo de la necesidad de la utilización de una máquina especializada para la ejecución y visualización de aplicaciones científicas por varios usuarios sin afectar su interacción y manteniendo unos niveles razonables de fluidez es que se sustenta este proyecto.

La finalidad de este proyecto será la implementación de un servicio de visualización de aplicaciones científicas, pues con un servicio de este tipo se espera darle solución a la problemática de acceso a las aplicaciones por parte de múltiples usuarios, al integrar los recursos hardware con una solución software de visualización, que permita desplegar aplicaciones científicas o entornos de desarrollo para estas mismas.

El camino para finalizar este proyecto inicia es analizando los componentes de hardware y software, que son necesarios para el sistema propuesto, luego, como parte complementaria

se hizo la prueba física al implementar las soluciones software disponibles para visualizar aplicaciones, finalizando esta primera etapa ya se tuvo la solución software seleccionada, para así implementar la arquitectura necesaria de la misma. Al tener esta implementación se logró establecer las características y restricciones de las aplicaciones que se despliegan en este entorno.

El acceso a los servicios por parte del usuario siempre ha sido un problema, este se abordó en las etapas iniciales del proyecto, por eso en el punto intermedio del desarrollo del proyecto se definió un componente de acceso e interfaz gráfica para que los usuarios además de acceder a el servicio, logran desplegar e interactuar gráficamente con una o varias aplicaciones.

En las secciones más avanzadas del proyecto se definieron los criterios para la implementación del servicio, donde se muestra los procesos de instalación y configuración para cada componente, también los requerimientos que debe cumplir el usuario para lograr desplegar las aplicaciones. En los pasos finales del desarrollo del proyecto que está plasmado en este documento se puede observar un piloto del Servicio de Visualización de Aplicaciones al implementar cada tarea hecha en las diferentes secciones.

Dejando como resultado un prototipo funcional, gracias a que se finalizó en buen término el proyecto, además puede funcionar como guía de cómo seleccionar e implementar los componentes necesarios de un Servicio de Visualización de Aplicaciones.

1. PLANTEAMIENTO DEL PROBLEMA

Las aplicaciones en las nuevas tecnologías tanto de hardware como de software contribuyen a un creciente volumen y diversidad de datos que necesitan ser manejados, explorados y visualizados, sin embargo, esto se está volviendo cada vez más difícil y complicado por la variedad de usuarios que exigen mayor funcionalidad, para esto las arquitecturas de visualización deben ser extensibles, modificables y que contribuyan a que se mejore la experiencia de usuario.

En el marco del Proyecto se propone desarrollar e implementar una arquitectura de visualización para las aplicaciones científicas de tal manera que además de facilitar el procesamiento de los datos gestione la correcta visualización de los resultados y a su vez permita la manipulación en entornos gráficos para poder interactuar con los mismos, también que brinde el acceso a múltiples usuarios a múltiples aplicaciones sin afectar el rendimiento, la interacción de la visualización y su fluidez.

Esta arquitectura será una herramienta que posibilite las visualizaciones para las aplicaciones científicas que el usuario final necesite, disfrutando de los recursos de computación remota, de tal forma que el uso del software científico sea el óptimo manteniendo el rendimiento sin importar la plataforma por la que se quiere acceder.

2. OBJETIVOS

2.1 OBJETIVO GENERAL

Implementar una Arquitectura de Visualización como Servicio para Aplicaciones Científicas en un Sistema Heterogéneo GPGPU.

2.2 OBJETIVOS ESPECÍFICOS

- Identificar los componentes de hardware / software y sus características que son necesarios en un Sistema Heterogéneo para la visualización de aplicaciones como servicio.
- Establecer las características y restricciones de las aplicaciones seleccionadas a implementar para la interacción.
- Proponer un componente para el acceso e interacción del Servicio de Visualización de Aplicaciones.
- Definir criterios para la implementación de componentes que permiten la configuración, acceso e interacción del servicio de visualización de aplicaciones sobre un Sistema Heterogéneo.
- Desarrollar un prototipo funcional para validar el comportamiento de la arquitectura de visualización como Servicio.

3. MARCO TEÓRICO

3.1 SISTEMAS HETEROGÉNEOS

Un Sistema heterogéneo es aquello que se encuentra compuesto por hardware con características físicas distintas entre sí, y software con características operativas distintas entre sí, pero que se pueden comunicar utilizando medios comunes [1].

La heterogeneidad es muy importante para la construcción de un ambiente, pues este tipo de sistemas se caracterizan por estar conectados a diferentes tipos de plataformas de cómputo las cuales deben tener resueltas las incompatibilidades que presentan entre sí como diferencias de formatos de archivos, longitud de palabra de los equipos y representación interna de caracteres. Para implementar una solución para esta problemática generalmente lo que se hace es construir software de traducción que permite convertir la información de un ambiente hacia otro. En el lugar de implementar todas las alternativas de conversión para los equipos que se tienen instalados, lo recomendable es generar un formato de intercambio estándar para todos, de forma tal que cada plataforma si lo necesite el desarrollo del sistema de traducción de sus datos hacia dicho formato estándar [2].

3.2 COMPUTACIÓN EN LA NUBE

Cuando se hace referencia a la nube, se está aludiendo a un término con algunos años de historia y que es una forma metafórica de nombrar a Internet. Básicamente la computación en la nube consiste en los servicios ofrecidos a través de la red tales como correo electrónico, almacenamiento, uso de aplicaciones, los cuales son normalmente accesibles mediante un navegador web. Al utilizar estos servicios, la información utilizada y almacenada, así como la mayoría de las aplicaciones requeridas, son procesadas y ejecutadas por un servidor en Internet [3].

Dicho en otras palabras, se trata de una implementación que pretende transformar el arquetipo habitual de la computación y la informativa y trasladarla a Internet.

El modelo de esta arquitectura tiene como base a “Las granjas de servidores”¹, estas eran similares en su arquitectura al procesamiento en red (grid), sin embargo, mientras que las redes se utilizan para aplicaciones de procesamiento técnico con un acoplamiento más bien débil (consistentes en un sistema compuesto de subsistemas con cierta autonomía de acción que mantienen una interrelación continua entre ellos formando una “supercomputadora virtual” para realizar grandes tareas), la nube orientó sus aplicaciones a los servicios de Internet.

3.2.1 Arquitectura en la Nube

La computación en la nube basa su arquitectura haciendo una separación entre hardware, plataforma y aplicaciones quedando las siguientes capas:

3.2.1.1 Software como Servicio (SaaS)

Se encuentra en la capa más alta y consiste en la entrega de aplicaciones completas como un servicio es donde el proveedor de tecnologías de información y comunicación (TIC) ofrece el SaaS (Software as a Service). Para ello dispone de una aplicación que se encarga de operar y mantener y que frecuentemente es desarrollada por el mismo. Con ella se encarga de dar servicio a multitud de clientes a través de la red, sin que estos tengan que instalar ningún software adicional. La distribución de la aplicación tiene el modelo de uno a muchos, es decir, se elabora un producto y el mismo lo usan varios clientes.

Las actividades son gestionadas desde alguna ubicación central, en lugar de hacerlo desde la sede de cada cliente, permitiendo a los clientes el acceso remoto a las aplicaciones mediante la web. Igualmente, las actualizaciones son centralizadas, eliminando la necesidad de descargar parches por parte de los usuarios finales.

3.2.1.2 . Plataforma como Servicio (PaaS)

En orden descendente, PaaS (Platform as a Service) es la siguiente capa. Básicamente su objetivo se centra en un modelo en el que se proporciona un servicio de plataforma con todo lo necesario para dar soporte al ciclo de planteamiento, desarrollo y puesta en marcha de aplicaciones y servicios web a través de la misma.

El proveedor es el encargado de escalar los recursos en caso de que la aplicación lo requiera, de que la plataforma tenga un rendimiento óptimo, de la seguridad de acceso, etc. Para desarrollar software se necesitan bases de datos, herramientas de desarrollo y en ocasiones servidores y redes. Con PaaS el cliente únicamente se enfoca en desarrollar, depurar y probar ya que la herramienta necesaria para el desarrollo de software es ofrecida a través de Internet, lo que teóricamente permite aumentar la productividad de los equipos de desarrollo.

3.2.1.3 Infraestructura como Servicio (IaaS)

IaaS (Infrastructure as a Service) corresponde a la capa más baja. La idea básica es la de hacer uso externo de servidores para espacio en disco, base de datos, ruteadores, switches así como tiempo de cómputo evitando de esta manera tener un servidor local y toda la infraestructura necesaria para la conectividad y mantenimiento dentro de una organización. Con una IaaS lo que se tiene es una solución en la que se paga por consumo de recursos solamente usados: espacio en disco utilizado, tiempo de CPU, espacio para base de datos, transferencia de datos, etc.

3.2.2 Virtualización en la nube

La virtualización es un elemento fundamental en el desarrollo óptimo de la computación en la nube, y se enfoca principalmente a la plataforma. Puede considerarse como una abstracción de los recursos tecnológicos que permite a los servidores crear dispositivos

virtuales los cuales pueden ser usados para aumentar dichos recursos más que como sistemas separados. Mediante la virtualización se permite tratar a un servidor como muchos servidores [4].

Otro método usado es el clustering, que consiste en tratar a muchos servidores como uno solo. Esto permite muchas mejoras como:

- Reducción de los costos de espacio y consumo.
- Rápida incorporación de nuevos recursos para los servidores virtualizados.
- Administración global centralizada y simplificada.
- Facilidad para la creación de entornos de test que permiten poner en marcha nuevas aplicaciones sin detener el desarrollo, agilizando el proceso de las pruebas.
- Aislamiento: un fallo en una máquina virtual no afecta al resto de máquinas virtuales.

3.3 GRID

Un grid de cómputo es una infraestructura de hardware y software que provee acceso confiable, consistente, extensivo y barato a sistemas de cómputo con gran capacidad de procesamiento. Un grid es un entorno compartido implementado mediante una infraestructura de servicios permanente y basada en estándares capaz de soportar, crear y compartir recursos [5]. Los recursos pueden ser computadoras, espacio de almacenamiento, instrumentos, aplicaciones (software), y datos, todos ellos conectados mediante una red (típicamente Internet) y una capa de software (middleware), la cual provee servicios de seguridad, monitoreo, gestión de recursos, etc. Los recursos pertenecen potencialmente a diferentes organizaciones y por lo tanto son compartidas bajo políticas que definen qué se comparte, quién puede acceder y bajo qué condiciones puede hacerlo. El problema real y específico del concepto grid es la coordinación de los recursos compartidos y cómo resolver problemas bajo una organización virtual multi-institucional.

3.3.1 Características

Algunas de las características principales son:

- Los recursos y servicios pueden incorporarse y retirarse dinámicamente del grid.
- Los recursos son heterogéneos, distribuidos geográficamente y normalmente conectados vía WAN.
- Los recursos son accesibles “on-demand” por un conjunto de usuarios autorizados que conforman una comunidad virtual.
- Se configura utilizando equipamiento de propósito general y protocolos estándar.
- Un objetivo es alcanzar un nivel definido de calidad de servicio.
- Un grid permite configurar múltiples máquinas paralelas virtuales para varios usuarios/aplicaciones simultáneas
- En un grid la heterogeneidad se extiende a la red de comunicaciones y al tipo de componentes en cada nodo que pueden ser procesadores, instrumentos, sensores, entre otros.

3.3.2 Nvidia GRID

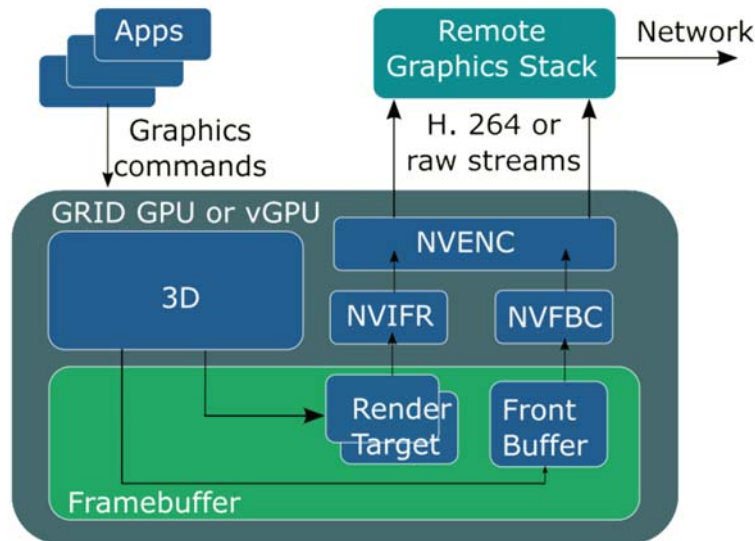
La tecnología NVIDIA GRID fue creada por Nvidia que aprovechan la capacidad de los procesadores gráficos para ofrecer a los usuarios aplicaciones, escritorios y juegos acelerados en la GPU a través de la red.

La gran capacidad de virtualización de la arquitectura de la NVIDIA GRID permite a múltiples usuarios compartir las GPUs de forma simultánea y con una altísima velocidad de streaming que elimina cualquier retardo en la visualización y produce la sensación de estar ejecutando la aplicación en un ordenador local. El software para implementar esta tecnología contiene librerías de virtualización, acceso remoto y gestión de sesiones aceleradas en la GPU que permiten a múltiples usuarios acceder a juegos y aplicaciones de alta carga gráfica utilizando GPUs compartidas. Proporcionando un rendimiento extraordinario y con baja latencia en las operaciones de captación, compresión, streaming y visualización de los juegos y aplicaciones a través de la red, esto brinda una excelente experiencia de visualización desde cualquier lugar y dispositivo [5].

El NVIDIA GRID SDK proporciona dos componentes para las API [1]:

- NvFBC: captura el escritorio, incluyendo las aplicaciones de Windows.
- NvIFR: captura desde un objetivo de renderizado específico

Figura 1. Nvidia GRID Gráficos Remotos SDK



3.4 VIRTUALIZACIÓN

Según la industria la virtualización es el proceso en el que crear una representación basada en software (o virtual), en lugar de una representación física [6]. Esta se puede aplicar a servidores, aplicaciones, almacenamiento y redes, y es un camino muy eficaz de reducir los costos de TI y aumentar la eficiencia y la agilidad de los procesos en la industria sin importar el tamaño [4].

3.4.1 Escritorios

La implementación de escritorios como un servicio administrado le permite responder con mayor rapidez a las necesidades y las oportunidades cambiantes. Puede reducir costos y aumentar el servicio mediante el suministro rápido y sencillo de escritorios y aplicaciones virtualizados a las sucursales, a los empleados en el extranjero y tercerizados, y a los empleados móviles con tabletas iPad y Android.

La virtualización del escritorio alojada en cliente crea un entorno de sistema operativo independiente en el escritorio lo que hace posible que casi cualquier aplicación que no es compatible puedan funcionar dentro de su propio entorno sobre un sistema operativo más corriente o permitan que dos entornos TI se ejecuten de manera concurrente en el mismo dispositivo físico. La infraestructura de escritorio virtual (VDI) es un modelo que hace posible que las cargas de trabajo de escritorio del cliente (sistema operativo, aplicaciones, datos de usuario) se alojen y ejecuten en servidores del centro de datos. Los usuarios pueden comunicarse con sus escritorios virtuales a través de un dispositivo cliente que ofrece soporte para protocolos de escritorio remoto tales como el RDP.

3.4.2 Aplicaciones

La virtualización de las aplicaciones consiste en aislar la componente "lógica de la aplicación" del componente sistema operativo.

El objetivo de esta virtualización es conseguir que las aplicaciones puedan funcionar con independencia de las características concretas del entorno en que se ejecutan. De este modo se eliminan los problemas de compatibilidad entre aplicaciones, o entre componentes de aplicaciones, y otros errores típicos de la ejecución concurrente de aplicaciones [7].

La virtualización de aplicaciones no virtualiza la presentación, es decir, el usuario sigue interactuando directamente con el equipo en que se ejecuta la aplicación virtualizada. Para conseguir su objetivo la virtualización de aplicaciones crea un entorno específico de ejecución para cada instancia de aplicación. Cualquier recurso que la aplicación tenga que necesitar (dll, claves de registro, controles ActiveX) está disponible de manera exclusiva para ella .

Estas aplicaciones se descargan bajo demanda desde un servidor en la red que suministrará el paquete que contiene la aplicación y todo el entorno y configuraciones necesarias para su ejecución. Esta aplicación se ejecutará en el sistema local en un entorno virtual protegido sin que se modifique absolutamente nada en el sistema local ni que interfiera con el resto de aplicaciones.

Las ventajas más significativas de este modelo son:

- Reducción de las necesidades de mantenimiento: no hay necesidad de mantenimiento, instalación y desinstalación de aplicaciones en el PC local.
- Disponibilidad de la aplicación en cualquier momento y lugar. Sólo es necesario acceso a la red y un dispositivo compatible (PC, terminal, portátil)

- Se pueden ejecutar varias versiones del mismo programa o diferentes programas cuya instalación simultánea y ejecución podría ser incompatible en el mismo sistema (cada aplicación se ejecuta en su propio entorno virtual por lo que no interfieren).
- Rapidez y facilidad para el despliegue de aplicaciones
- Se optimizan el número de licencias en caso de software propietario. Por ejemplo si una institución educativa encarga a sus alumnos trabajos con una aplicación propietaria puede estudiar el número máximo de usuarios concurrentes que podría haber y ofrecer ese número de licencias por red en vez de asignar una licencia fija a cada posible alumno.
- Todo ello redundaría en una reducción del TCO (Total Cost Ownership o Costo total de propiedad)
- Algunas soluciones extendidas actualmente en el mercado son:
 - Microsoft SoftGrid
 - Vmware ThinApp
 - Citrix XenApp
 - Docker

3.5 VISUALIZACIÓN

La visualización por computadora es un proceso de mapeo de las representaciones hechas por la computadora a representaciones perceptuales, eligiendo técnicas de codificación para maximizar el entendimiento y comunicación con los seres humanos. Básicamente, la visualización nos permite interpretar datos que se obtienen de investigaciones matemáticas o científicas. Se utilizan los sistemas computacionales no para simular, sino para representar estos datos.

La tecnología de visualización es una integración de las áreas de graficación, procesamiento de imágenes, visión computacional, modelado geométrico, diseño asistido por computadora, psicología perceptual, estudios de interfaces de usuarios, etc. Por lo tanto, las personas que se encargan del estudio de sistemas de visualización deben contar con conocimientos en áreas como diseño gráfico, ciencias, matemáticas, graficación por computadora y animación [8].

Hay tres partes importantes en un sistema de visualización.

- Construcción de un modelo empírico de los datos: Este modelo puede tener consideraciones sobre teoría del muestreo, como el teorema Nyquist y esquemas de interpolación matemática. También debemos tomar en cuenta la probabilidad de que haya errores en los datos.
- Selección de esquemas: Significa tomar como modelo un objeto de visualización abstracta (un mapa por ejemplo).
- La representación de la imagen en un ambiente gráfico.

3.6 VISUALIZACIÓN CIENTÍFICA

La visualización científica es la transformación de datos científicos y abstractos en imágenes. Es una forma especial de la visualización que procura encontrar una representación visual apropiada para un conjunto de datos que permita mayor efectividad en el análisis y evaluación de los mismos. Simplifica el análisis, comprensión y la comunicación de modelos, conceptos y datos en la ciencia y la ingeniería. Entonces la visualización científica es la responsable de la transformación de datos en imágenes a partir de la abstracción de los datos.

El software de visualización científica usa la forma, la posición en el espacio, el color, la brillantez y el movimiento para ayudarnos a comprender relaciones invisibles para nosotros. Este software ya no se limita a supercomputadoras, algunos de los programas más innovadores se han creado para usarse en arquitecturas menos robustas, y son acelerados mediante el uso de paralelismo [9].

Según R.A. Earnshaw, la meta de la visualización científica es promover un nivel más profundo de los datos que se están investigando y proporcionar mayor profundidad a los procesos, confiando en el sistema visual humano y en su habilidad para interpretar datos visuales. Las herramientas y técnicas de visualización han sido utilizadas para analizar grandes volúmenes de datos multidimensionales de forma que permitan al usuario extraer resultados significativos de forma rápida y fácil.

3.6.1 Visualización de Volúmenes

La Visualización de Volúmenes se refiere generalmente a campos escalares. Se extiende desde el examen de datos científicos a la reconstrucción de datos dispersos y a la representación de objetos geométricos sin una descripción matemática de su superficie. Permite el examen del interior de un volumen utilizando técnicas como las del slicing y la transparencia. Una cuestión no tan madura en la Visualización de volúmenes la constituye el problema de la dimensionalidad: cuando la dimensión de los datos excede cierta dimensión, la visualización no puede manejarse usando los métodos tradicionales y se requieren nuevas técnicas [10].

3.6.2 Visualización de Flujo

Este campo se utiliza para la visualización en general de Sistemas Dinámicos, es decir, aquellos sistemas en los que hay involucrados datos que evolucionan en el tiempo. El comportamiento cualitativo de dichos sistemas puede comprenderse adecuadamente a partir de la estructura de la evolución temporal de sus trayectorias. Estos contienen implícitamente una gran cantidad de datos que no es directa ni fácilmente observable.

3.6.3 Visualización como Servicio

La Visualización como un servicio es una estrategia que se utiliza para acceder a datos críticos en tiempo real, de manera segura y asequible "nube". Durante décadas, las empresas han buscado este servicio para tomar más decisiones basadas en datos. Como los datos se reunieron en una escala sin precedentes y mediciones detalladas ahora son factibles en casi en tiempo real, un reto importante es ser capaz de integrar los datos y visualizar de forma que sea significativa. Las empresas luchan con el costo y la complejidad de estos esfuerzos. No bien constituye un esfuerzo completo, entonces el negocio tiene que cambiar y es necesario un nuevo esfuerzo importante. La visualización de datos como servicio ofrece una solución para afrontar estos retos.

Este servicio abre la puerta a un modelo eficaz y flexible costo. Dada la naturaleza global de la industria, el alcance y la amplitud de todo, desde la investigación hasta la fabricación presenta un reto significativo cuando se aspira a crear una historia unificada y significativa con los datos. Los datos útiles pueden ser suministrados a los usuarios en la demanda, con independencia de cualquier separación orgánica o geográfica entre los consumidores y los proveedores. La Visualización como Servicio elimina la redundancia y reduce los gastos asociados al acomodar los datos vitales en un solo lugar, permitiendo el uso de datos y/o modificación por varios usuarios a través de un único punto de actualización.

La Visualización de datos como servicio integra una amplia variedad de fuentes de datos dispares en una interfaz habilitada en la nube. Es importante destacar que la nube ofrece escalabilidad a la rampa hacia arriba (o hacia abajo) se alimenta de datos. Para la industria está escalabilidad permite a las empresas centrarse en lo que mejor saben hacer, descubrir, fabricación y venta de soluciones.

La Visualización de datos es utilizada directamente por las compañías como las farmacéuticas, así como por empresas de soluciones de gestión de muestras para proporcionar inteligencia unificada que produce un mejor tiempo de salida al mercado y una mayor flexibilidad en cómo se entrega la investigación y la información clínica. Estas empresas han hecho activos de datos de muestra más accesibles, reutilizables y proporcionado a los investigadores con un motor operativo que proporciona entradas de datos utilizando ya sea un enfoque salpicadero, un método de información, o bien un enfoque portal para navegar por los datos. La visión y un análisis que es obtenida mediante la visualización de datos esto reduce los costos de investigación, tiempo de salida al mercado, y el número de muestras recogidas.

Aparte de la flexibilidad, la accesibilidad y la seguridad, la capacidad de tomar datos verdaderamente comprensibles es la clave. Los datos sin un medio de visualizar de manera eficaz se desperdician. Hasta la fecha, esto ha sido un reto importante. se utiliza sólo una pequeña fracción de los datos recogidos por las empresas. La Visualización como Servicio puede ayudar a interpretación de la gran cantidad de datos ya través de las visualizaciones estándar o personalizados, que se necesitan para la toma las decisiones que ayudan a conducir eficazmente sus organizaciones hacia adelante. Con la apertura de datos a más personas y tomadores de decisiones a través de la unificación de los servicios de Visualización, las decisiones pueden ser más oportuna e informada. Como un servicio, las empresas acceder a las herramientas de vanguardia y formas innovadoras de relación con los datos.

El uso de la Visualización de datos abarca muchas industrias, incluyendo el seguro, la prestación de atención de salud, la investigación y el descubrimiento de fármacos, ensayos clínicos y aprobaciones, fabricación y ventas, defensa del gobierno, análisis y funciones corporativas.

3.7 APLICACIONES

Una aplicación es un programa informático diseñado como una herramienta que permite al usuario realizar una tarea en un dispositivo informático, estas nacen al existir una necesidad concreta del usuario.

3.7.1 Científicas

El desarrollo de aplicaciones tuvo un gran empuje cuando la industria la necesito para mejorar sus procesos, pero para llegar a esto debían fomentar la ciencia que hace posible los avances de la industria.

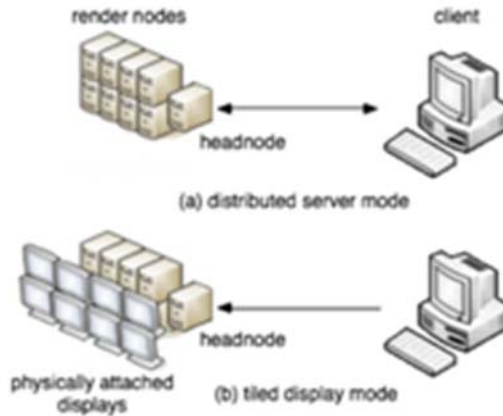
3.7.1.1 Paraview

ParaView es una herramienta de código abierto y multiplataforma de visualización científica diseñada para analizar grandes conjuntos de datos mediante el aprovechamiento de los entornos informáticos distribuidos de memoria. Mientras que un usuario puede realizar los cálculos y la representación en una sola máquina mediante la ejecución de una única instancia de ParaView, varias instancias que funcionan en paralelo se prefiere para la visualización de grandes conjuntos de datos [11].

Soporta varios modos de operación en un entorno informático distribuido [Figura 2].

- **Modo cliente-servidor:** un servidor (un solo PC), que se encuentra de forma remota, realiza el cálculo y la representación, y los píxeles resultantes se transmite a un cliente de escritorio. Este modo permite la visualización remota, pero no es compatible con la computación paralela y renderizado.
- **Modo de servidor distribuida:** un conjunto de ordenadores realiza el cálculo y la representación en paralelo. En este modo, el nodo maestro del clúster Compuestos la imagen final y lo envía al cliente. Por lo tanto, la resolución de la imagen final se limita a la resolución del escritorio del cliente.
- **Modo de Pantalla de gres:** un conjunto de ordenadores realiza el cálculo y la representación en paralelo. Cada nodo del clúster y materiales compuestos hace que su fragmento de imagen y lo muestra en pantalla conectada físicamente. Este modo no es compatible con la visualización remota.

Figura 2. Paraview Arquitectura [9]



En [2] Los dos modos de ParaView para soportar la visualización escalable: (a) En el modo de servidor distribuido, el nodo principal envía la imagen final al cliente. (B) En el modo de visualización de baldosa, la imagen final se visualiza en los monitores conectados a la prestación de clúster.

Paraview soporta renderizado paralelo a distancia a través de su modo de servidor distribuido, y la representación de ultra-alta resolución a través de su modo de visualización de baldosas.

3.7.1.2 Visualization Toolkit (VTK)

Es un sistema de software libre y de código abierto para computación gráfica en 3D, procesamiento de imágenes y visualización. VTK consiste en una librería de clases de C++ y varias capas de interfaz interpretadas incluyendo Tcl, Java y Python. VTK soporta una amplia variedad de algoritmos de visualización y técnicas de modelamiento avanzado. VTK es multiplataforma y funciona en Linux, Windows, Mac y plataformas Unix [12].

3.7.1.3 Netbeans

Netbeans es un entorno de desarrollo gratuito y de código abierto que en el momento de escribir este artículo está en su versión 7.4. Permite el uso de un amplio rango de tecnologías de desarrollo tanto para escritorio, como aplicaciones Web, o para dispositivos móviles. Da soporte a las siguientes tecnologías, entre otras: Java, PHP, Groovy, C/C++, HTML5, Además puede instalarse en varios sistemas operativos: Windows, Linux, Mac OS.

Características principales:

Suele dar soporte a casi todas las novedades en el lenguaje Java. Cualquier preview del lenguaje es rápidamente soportada por Netbeans.

Asistentes para la creación y configuración de distintos proyectos, incluida la elección de algunos frameworks.

Buen editor de código, multilenguaje, con el habitual coloreado y sugerencias de código, acceso a clases pinchando en el código, control de versiones, localización de ubicación de la clase actual, comprobaciones sintácticas y semánticas, plantillas de código, coding tips, herramientas de refactorización. También hay tecnologías donde podemos usar el pulsar y arrastrar para incluir componentes en nuestro código.

Simplifica la gestión de grandes proyectos con el uso de diferentes vistas, asistentes de ayuda, y estructurando la visualización de manera ordenada, lo que ayuda en el trabajo diario. Una vez que nos metemos en una clase java, por poner un ejemplo, se nos mostrarán distintas ventanas con el código, su localización en el proyecto, una lista de los métodos y propiedades (ordenadas alfabéticamente), también hay una vista que nos presenta las jerarquías que tiene nuestra clase y otras muchas opciones. Por supuesto personalizable según el gusto de cada usuario.

Acceso a base de datos: desde el propio Netbeans podemos conectarnos a distintos sistemas gestores de bases de datos, como pueden ser Oracle, MySql y demás, y ver las tablas, realizar consultas y modificaciones, y todo ello integrado en el propio IDE.

Se integra con diversos servidores de aplicaciones, de tal manera que podemos gestionarlos desde el propio IDE: inicio, parada, arranque en modo debug, despliegues. Entre otros podemos usar Apache Tomcat, GlassFish, JBoss, WebLogic, Sailfin, Sun Java System Application Server.

3.7.2 Videojuegos

Los videojuegos son una gran herramienta para la utilización del Servicio de Visualización, pues su desarrollo con motores gráficos 3D están implementando librerías como OpenGL que permiten la utilización de múltiples hilos en la ejecución del juego, basándose en el modelo que consiste en dejar de lado la actualización serial de cada subsistema en el motor Gráfico y reemplazarlo por un modelo donde cada subsistema del mismo se actualiza de manera independiente pero siempre manteniendo cierta coherencia en los tiempos transcurridos en la simulación. Este modelo permite utilizar mejor los recursos disponibles en un microprocesador de múltiples cores y con un Sistema Heterogéneo adecuado es posible especializar para un componente en específico.

OpenGL multihilo saca ventaja de esto distribuyendo el procesamiento de las escenas de gráficos 3D complejos entre múltiples procesadores o núcleos de procesamiento múltiples. Empresas como Blizzard afirma que el soporte de OpenGL multihilo puede, de hecho, doblar el rendimiento de World of Warcraft dependiendo de las capacidades de hardware y los ajustes especificados en el juego.

Actualmente Quake 4 y la saga de Call of Duty tienen la opción de los multihilos para la ejecución del juego.

4. MARCO TÉCNICO

Para el desarrollo del proyecto se contó con varios elementos de Hardware como de Software, que fueron útiles para la experimentación de la arquitectura y el funcionamiento de la visualización de las aplicaciones.

4.1 HARDWARE

El hardware utilizado para la realización de este proyecto se presenta en las siguientes tablas, donde se detallan sus principales características técnicas.

4.1.1 Servidor

Para la realización del Servicio de Visualización de Aplicaciones se contó con un Servidor con las características del Tabla 1.

Tabla 1. Características Servidor

Servidor	
Marca	Hewlett Packard Enterprise
Modelo	ML350R
Procesador	Intel Xeon 6 Core Processor E5-2609
Memoria ROM	500 Gb HDD
Memoria RAM	16 Gb

4.1.1.1 Nvidia GRID K2

Para la realización del Servicio de Visualización de Aplicaciones se contó con un Tarjeta Nvidia Grid K2 con las características mostradas en la Tabla 2 .

Tabla 2. Características Nvidia Grid K2

NVIDIA GRID	
Modelo	GRID K2
Número de GPUs	2 GPUs Kepler de gama alta
Memoria RAM	8 Gb GDDR5
Potencia Máxima	225 W

4.1.2 Computador

Para la realización del Servicio de Visualización de Aplicaciones se contó con un Computador con las características mostradas en la Tabla 3.

Tabla 3. Características Computador Administrador

Computador	
Marca	Dell
Modelo	Optiplex 9020
Memoria ROM	120 Gb
Memoria RAM	16 GB

4.1.2.1 Nvidia GTX660

Para la realización del Servicio de Visualización de Aplicaciones se contó con un Computador con las características mostradas en la Tabla 3 y contaba con una tarjeta gráfica con las características mostrada en la Tabla 4

Tabla 4. Especificaciones GTX 660

Tarjeta Gráfica	
Marca	MSI
Modelo	N610GT-MD1G3/LP
GPU	GeForce GT 610
Memoria RAM	1 Gb DDR3

4.2 SOFTWARE

Se utilizaron diversas herramientas software para probar la funcionalidad de la arquitectura y lograr utilizarla con base a los parámetros del proyecto. A continuación se describe cada uno de ellos.

4.2.1 Solución VMware

VMware como empresa presenta una solución [13] completa para administrar, generar y monitorear un Sistema de Virtualización y Visualización de escritorios y aplicaciones, para esto dispone que algunos software para labores especifica. Asi, para la implementacion de esta solucion se conto con los siguientes softwares.

4.2.1.1 VMWare vSphere

Con vSphere, aprovechará al máximo el rendimiento, la disponibilidad y la eficacia de la infraestructura y las aplicaciones. Es la base ideal para cualquier entorno de nube.

VMware vSphere es la plataforma de virtualización que permite a los usuarios ejecutar aplicaciones críticas para el negocio con confianza y responder con mayor rapidez a las necesidades empresariales

Dicha solución le permitirá comprender todo su ambiente virtual y así llegar al análisis de la causa raíz de problemas de forma más rápida y sencilla y desde un solo lugar.

Estudios analíticos de rendimiento, Permite conocer y comprender la salud y el rendimiento de su entorno, es decir aplicar la inteligencia y visibilidad necesaria para sacar más provecho de su entorno virtualizado con VMware. Los niveles de servicio para los servicios de TI están garantizados.

Capacidad de Gestión y Optimización, Vsphere con Operations Manager Suite posee instrumentos que permiten conocer la capacidad que tiene actualmente en el centro de datos y así determinar con exactitud los componentes que realmente necesita expandir, optimizando así la eficiencia y costos del mismo.

4.2.1.2 VMWare vSphere Hypervisor

Es una versión gratuita del producto VMWare ESXi, con el poder crear máquinas virtuales de hasta 8 vCPU y la imposibilidad de ser gestionado a través de un vCenter.

4.2.1.3 VMWare vCenter Server

Punto de control para servicios de centro de datos como control de acceso, supervisión de rendimiento y administración de alarmas, además conecta los hosts vSphere a vCenter Server para la gestión centralizada de todos los hosts y máquinas virtuales.

Este servicio actúa como administrador central de VMWare ESXi servers que están conectados en una red. VCenter Server, proporciona el punto central para configurar, aprovisionar y administrar máquinas virtuales en el centro de datos.

Además de utilizar estas máquinas virtuales como fuentes de agrupaciones de equipos de máquinas virtuales, puede utilizar máquinas virtuales para alojar los componentes de servidor de Vista, incluidas las instancias de View Connection Server, los servidores Active Directory, los hosts RDS de Microsoft y las instancias de vCenter Server.

4.2.1.4 View Connection Server

Este servicio de software actúa como intermediario para las conexiones de clientes. View Connection Server autentica a los usuarios a través de Windows Active Directory y dirige la solicitud a la máquina virtual, el PC físico o el host RDS de Microsoft.

View Connection Server proporciona las siguientes capacidades de administración:

- Autenticación de usuarios
- Habilitación de usuarios a escritorios y piscinas específicos
- Asignación de aplicaciones empaquetadas con VMware ThinApp a equipos de escritorio y grupos de servidores específicos en Gestión de sesiones de escritorio y aplicaciones remotas
- Establecimiento de conexiones seguras entre usuarios y escritorios y aplicaciones remotas n Habilitación de inicio de sesión único
- Establecimiento y aplicación de políticas

Dentro del cortafuego corporativo, instala y configura un grupo de dos o más instancias de View Connection Server. Sus datos de configuración se almacenan en un directorio LDAP incrustado y se replican entre los miembros del grupo.

Fuera del cortafuego corporativo, en la DMZ, puede instalar y configurar View Connection Server como un servidor de seguridad. Los servidores de seguridad en la DMZ se comunican con View Connection Servers dentro del cortafuegos corporativos. Los servidores de seguridad garantizan que el único tráfico remoto de escritorio y aplicación que pueda ingresar al centro de datos corporativo es el tráfico en nombre de un usuario con autenticación fuerte. Los usuarios sólo pueden acceder a los recursos a los que están autorizados a acceder.

Los servidores de seguridad ofrecen un subconjunto de funcionalidades y no se requiere que estén en un dominio de Active Directory. Instala View Connection Server en un servidor Windows Server 2008, Windows Server 2012 o Windows Server 2012 R2, preferentemente en una máquina virtual de VMware.

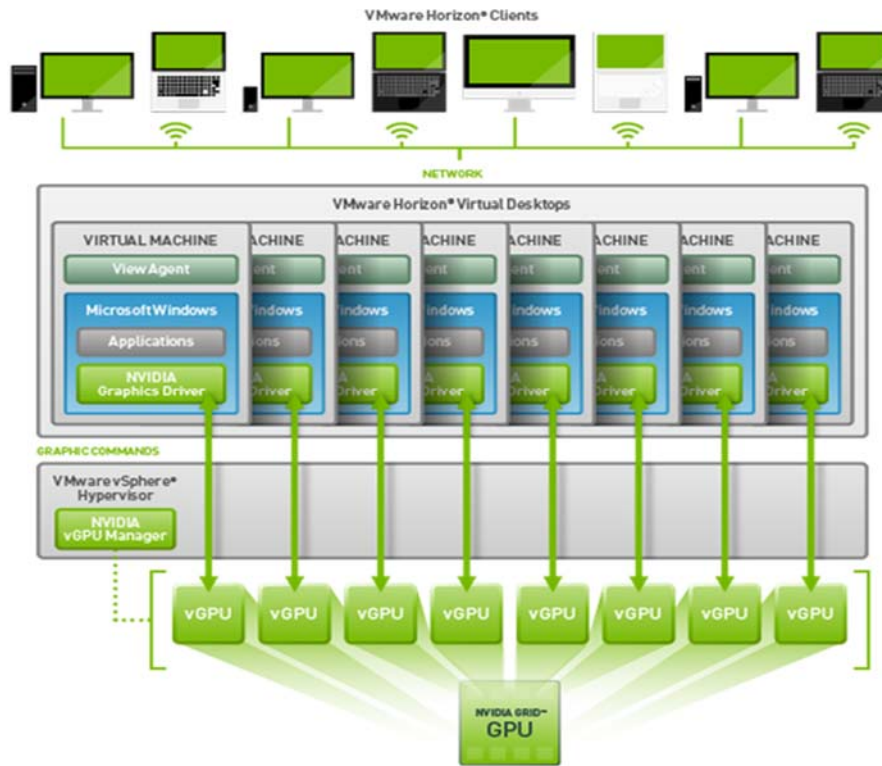
4.2.1.5 VMWare Horizon Suite

Es un conjunto de aplicaciones que permite realizar la virtualización de escritorios (VDI) y de forma simple y práctica. La implementación de Horizon View esta permite reducir el esfuerzo y trabajo de administrar perfiles por medio de Persona Management y el tamaño en disco por medio de Composer. Con Horizon Mirage se administran las imágenes de Windows en diferentes capas, comunes para VDI, Notebooks, Desktops [14].

Permite una gestión centralizada también hacer fácilmente los respaldos de máquinas virtualizadas (VM). Por último, se presenta como el espacio de trabajo para que los usuarios acceden vía Web de manera segura a los archivos, a las aplicaciones fueron creadas de manera práctica y que permite independizarse de las aplicaciones físicas o instalables además cuenta con la integración a servicios de directorios como Microsoft Active Directory u otros servicios como Google.

La integración [Figura 3] muestra la integración de la Nvidia Grid [4.1.1.1] con la Solución Vmware.

Figura 3. Integración VMWare vSphere / Horizon con la Nvidia Grid.



4.2.1.6 VMware ThinApp

Es una aplicación que crea aplicaciones portátiles, pues esta virtualiza las aplicaciones al encapsular el registro y los archivos de estas en un solo paquete, además se puede implementar, gestionar y actualizar la aplicación independientemente del sistema operativo de la máquina del cliente. Las aplicaciones virtualizadas no modifican el sistema operativo en las que son desplegadas y siguen funcionando de la misma forma en las distintas configuraciones, lo que garantiza compatibilidad, facilidad de gestión y una experiencia de usuario uniforme.

4.2.2 Docker

Docker implementa una API de alto nivel para proporcionar contenedores livianos que ejecutan procesos de manera aislada.

Construido sobre las facilidades proporcionadas por el kernel de Linux (principalmente cgroups y namespaces), un contenedor Docker, a diferencia de una máquina virtual, no requiere incluir un sistema operativo independiente. En su lugar, se basa en las funcionalidades del kernel y utiliza el aislamiento de recursos (CPU, la memoria, el bloque

E / S, red, etc.) y namespaces separados para aislar de vista la aplicación del sistema operativo. Docker accede a la virtualización del kernel de Linux ya sea directamente a través de la biblioteca libcontainer (disponible desde Docker 0.9), o indirectamente a través de libvirt, LXC o systemd-nspawn [15].

4.2.2.1 Contenedores

Son como un directorio, contiene todo lo necesario para que una aplicación pueda funcionar sin necesidad de acceder a un repositorio externo al contenedor. Cada uno de éstos es una plataforma de aplicaciones segura y aislada del resto que podamos encontrar o desplegar en la misma máquina host.

4.2.2.2 Imágenes

La imagen Docker podríamos entenderla como un SO con aplicaciones instaladas (Por ejemplo, un Debian con un paquete ofimático). Sobre esta base podremos empezar a añadir aplicaciones que vayamos a necesitar en otro equipo donde tengamos intención de usar la imagen. Además, Docker nos ofrece una forma muy sencilla de actualizar las imágenes que tengamos creadas, así como un sencillo método para crear nuevas imágenes.

4.2.2.3 Repositorios:

También conocidos como Registros Docker, contienen imágenes creadas por los usuarios y puestas a disposición del público. Podemos encontrar repositorios públicos y totalmente gratuitos o repositorios privados donde tendremos que comprar las imágenes que necesitemos. Estos registros permiten desarrollar o desplegar aplicaciones de forma simple y rápida en base a plantillas, reduciendo el tiempo de creación o implementación de aplicaciones o sistemas.

Mediante el uso de contenedores, los recursos pueden ser aislados, los servicios restringidos, y se otorga a los procesos la capacidad de tener una visión casi completamente privada del sistema operativo con su propio identificador de espacio de proceso, la estructura del sistema de archivos, y las interfaces de red. Contenedores múltiples comparten el mismo núcleo, pero cada contenedor puede ser restringido a utilizar sólo una cantidad definida de recursos como CPU, memoria y E / S.

Usando Docker para crear y gestionar contenedores puede simplificar la creación de sistemas altamente distribuidos, permitiendo múltiples aplicaciones, las tareas de los trabajadores y otros procesos para funcionar de forma autónoma en una única máquina física o en varias máquinas virtuales. Esto permite que el despliegue de nodos se realice a medida que se dispone de recursos o cuando se necesiten más nodos, lo que permite una plataforma como servicio (PaaS - Platform as a Service) de estilo de despliegue y ampliación de los sistemas como Apache Cassandra, MongoDB o Riak. Docker también simplifica la creación y el funcionamiento de las tareas de carga de trabajo o las colas y otros sistemas distribuidos.

4.2.3 Sistema de Ventanas X11

El Sistema de Ventanas X es un software creado para proporcionar de una interfaz gráfica a los sistemas Unix. Este protocolo permite la interacción gráfica en red entre un usuario y una o más computadoras haciendo transparente la red para éste, actualmente está en uso la versión 11 (X11). X es el encargado de mostrar la información gráfica de forma totalmente independiente del sistema operativo.

X fue diseñado primariamente para implementar clientes ligeros, donde mucha gente usaba simultáneamente la capacidad de procesamiento de un mismo computador trabajando en tiempo compartido. Cada persona usaba un terminal en red que tenía capacidades limitadas para dibujar la pantalla y aceptar la entrada del usuario. Debido a la ubicuidad del soporte para el software X en Unix, es usado en los computadores personales incluso cuando no hay necesidad del tiempo compartido.

Este sistema permite distribuir el procesamiento de las aplicaciones creando un enlace entre el cliente y el servidor, mientras el servidor X11 proporciona servicios para acceder a la pantalla, teclado y ratón, el cliente X11 proporciona las aplicaciones que utilizan los recursos para interacción con el usuario.

De este modo mientras el servidor se ejecuta de manera local, las aplicaciones pueden ejecutarse remotamente desde otras máquinas, proporcionando así el concepto de transparencia de red.

Debido a este esquema cliente-servidor, se puede decir que X se comporta como un terminal gráfico virtual. [16]

5. METODOLOGÍA

El proyecto se llevará a cabo siguiendo los principios de las metodologías ágiles de desarrollo iterativo e incremental, El desarrollo ágil permite tener una visión del avance del proyecto, se realiza una lista de ítems a realizar y se dividen en un conjunto de paquetes o etapas, cada paquete cuenta con determinadas actividades a realizar, esto conlleva a que cada entrega de una parte funcional del producto pueda ser evaluada y de ser necesario, corregidas en la siguiente iteración.

PT: Paquete de trabajo. A: Actividad. E: Entregable

PT1: Análisis de Componentes Hardware y Software.

A1.2: Identificar los componentes de hardware y software necesarios en un sistema heterogéneo para implementar la Visualización como Servicio.

E1.1: Lista de Componentes Hardware y Software de un Sistema Heterogéneo para implementar la visualización como Servicio.

E1.2: Descripción y justificación de la selección de Hardware y Software.

PT2: Análisis de Aplicaciones

A2.1: Identificar las aplicaciones que sean adecuadas para la visualización

A2.2: Caracterizar las aplicaciones identificadas en A2.1 para determinar grado de optimización en su utilización en un Sistema Heterogéneo.

E2.1: Lista de Aplicaciones con sus características y restricciones que serán implementadas para la interacción

PT3: Análisis del componente de Acceso e Interacción del Servicio.

A3.1: Identificar los componentes existentes para realizar la visualización en un Sistema Heterogéneos. A3.2: Analizar los componentes y seleccionar el componente que más se adapte a el hardware y software disponible para la visualización.

E3.1: Detalle descriptivo del componente seleccionado para la visualización.

PT4: Análisis de Criterios para la Implementación del Componente E3.1

A4.1: Identificar los procedimientos de configuración del Componente para el Servicio de Visualización de Aplicaciones.

A4.2: Analizar la configuración del Componente para priorizar el acceso e interacción de las aplicaciones en la Visualización.

E4.1: Descripción de criterios para la implementación de las aplicaciones E2.1 en el componente de visualización E3.1 sobre el Sistema Heterogéneo E1.1.

PT5: Desarrollo del Prototipo Funcional

A5.1: Implementación de la arquitectura de Visualización en el Sistema Heterogéneo E1.1.

A5.2: Implementación de las Aplicaciones E2.1 en el Componente de Visualización E3.1. bajo los criterios descritos en E4.1

A5.3: Análisis de requerimientos de un prototipo donde se pueda validar el comportamiento de la Arquitectura de Visualización como Servicio

A5.4: Desarrollar un Prototipo Funcional implementando los requerimientos A5.3

E5.1: Prototipo Funcional a partir de los requerimientos A5.3

6. DESARROLLO DEL PROYECTO

6.1 COMPONENTES

El desarrollo del proyecto inició siguiendo la metodología planteada, de manera que en esta sección se identificarán los componentes de hardware y software, con sus características, que son necesarios en un Sistema Heterogéneo para la visualización de aplicaciones como servicio.

6.1.1 Hardware

Los componentes de hardware para formar el sistema heterogéneo que se tenían para el desarrollo de este proyecto fueron:

- Un servidor que cuenta con las especificaciones presentadas en la Tabla 1, está en la instalación del Parque Tecnológico de Guatiguará que está equipado con una Nvidia Grid K2 y de especificaciones técnicas descritas en la **¡Error! No se encuentra el origen de la referencia.** que funciona para la administración de sesiones, la virtualización y acceso de la GPU para múltiples usuarios, este fue adquirido por el grupo de Supercomputación y Cálculo Científico a Hewlett Packard Enterprise con estas especificaciones [16].
- Un computador que cuenta con las especificaciones presentadas en la Tabla 3, está instalado en las instalaciones del Centic de la Universidad Industrial de Santander en las oficinas del Grupo de Supercomputación y Cálculo Científico (SC3) este funcionó como nodo administrador de la implementación de VMWare y frontend además cuenta con la tarjeta gráfica con las especificaciones técnicas mostradas en la Tabla 4, este también funcionó como servidor para la implementación nativa de Docker [4.2.2].

6.1.2 Software

Actualmente existen varias plataformas para la virtualización y visualización aplicación algunas de las empresas que desarrollan soluciones con un coste en la licencias (de pago) son VMware (VMware Horizon 6.1) y Citrix Server (Citrix 7.6 XenDesktop) estos desarrollos son muy completos para la creación, administración y monitorea de escritorios con sistemas operativos Windows o Linux y aplicaciones estas están muy enfocadas al sector industrial, para las pruebas realizadas se contó con licencias académicas; Por otro lado se encuentran XenServer que es una versión de código abierto de Citrix Server y esta Docker que es otro enfoque a para la visualización de aplicaciones pues pasa de crear un aplicación virtual a la creación de contenedores en la máquina instalada.

De acuerdo a la disponibilidad de una licencia de prueba se implementó la solución de VMware pero debido al alto costo de sus licencias se exploró e implementó la solución de Docker que al ser de código abierto permite que la implementación sea más viable en el

ámbito educativo.

Tabla 5. Lista de Componentes Software

SOFTWARE	VERSIÓN	LICENCIA
Microsoft Server	2012 R2	Microsoft Imagine
Debian	8.0	Open Source
vSphere and vSphere with Operations Management	6.0	Educativa de Prueba
VMware Horizon	7.0	Educativa de Prueba
VMware ThinApp	5.2	Educativa de Prueba
Docker	1.24	Apache Licencia 2.0

6.2 IMPLEMENTACIÓN

Se hicieron varias implementaciones preliminares, para poder seleccionar una definitiva y que fuera la base para el desarrollo del prototipo funcional.

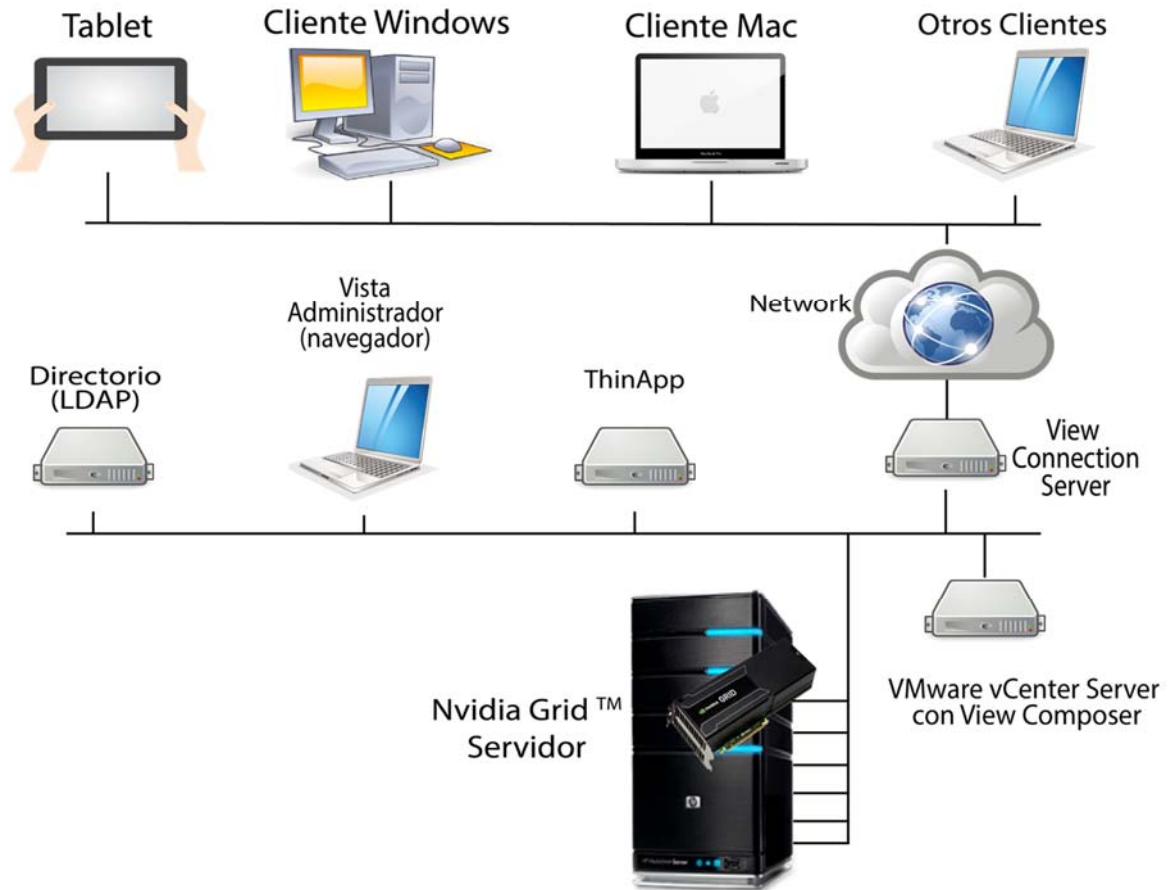
6.2.1 VMware

La implementación de la solución de virtualización y visualización de aplicaciones ofrecida por VMWare con VMWare vSphere [4.2.1.1] y la VMware Horizon Suite [4.2.1.5], tiene como principales beneficios la variedad de opciones de configuración y el módulo de administrador que cuenta diferentes secciones para el manejo de usuarios, utilización de la red y los recursos de hardware, la seguridad que se puede configurar con diferentes niveles de acceso para los usuarios.

La arquitectura mostrada en la Figura 4 para una utilización de la solución VMWare empieza por el servidor que tiene instalado la Nvidia Grid K2 [4.1.1.1] a este se le instala primero el VMWare vSphere Hypervisor [4.2.1.2], debido a la restricciones de máquinas físicas se optó por virtualizar los servidores con el sistema operativo Microsoft Server 2012 R2 esto se hizo utilizando Virtual Box en el computador administrador [Tabla 3], se instaló VMware vCenter Server [4.2.1.3] en uno de los servidores virtualizados, el VMware View Connection Server [4.2.1.4] en un

segundo servidor virtualizado y por último lugar el servidor de ThinApp [4.2.1.6], virtualizar estas máquinas está contemplado en el manual de instalación [13] como una posible solución a la falta de disponibilidad de máquinas físicas por ende el rendimiento no se afecta para la implementación de la solución por último está el servidor LDAP que es el directorio de usuarios que funciona para validar el ingreso de los usuarios este se configura desde el View Connection Server.

Figura 4. Arquitectura VMWare

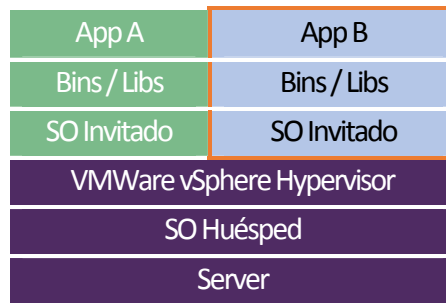


Esta solución implementada presento el problema con las licencias de funcionamiento, pues se trabajó con una licencia educativa de prueba que VMWare ofrece que es por 90 días, ese tiempo fue suficiente para probar el funcionamiento básico de la herramienta ThinApp [4.2.1.6] logrando así desplegar la aplicación Paraview [3.7.1.1] entre otras para el usuario de prueba, se utilizó el VMWare Horizon Client que está disponible en la sitio web [17] para una gran variedad de Sistemas Operativos para la visualización de la aplicación.

La utilización de la arquitectura mostrada en la Figura 4 y su implementación para la solución de visualización se base en la capacidad del

VMWare vSphere Hypervisor [4.2.1.2] y lo complementa con la utilización de la Nvidia Grid K2 [4.1.1.1] logrando que las aplicaciones virtualizadas con los controladores gráficos de Nvidia instalados de acuerdo a la infraestructura de la Tabla 6 puedan hacer uso de los recursos de GPU del servidor.

Tabla 6. Infraestructura Nativa VMWare

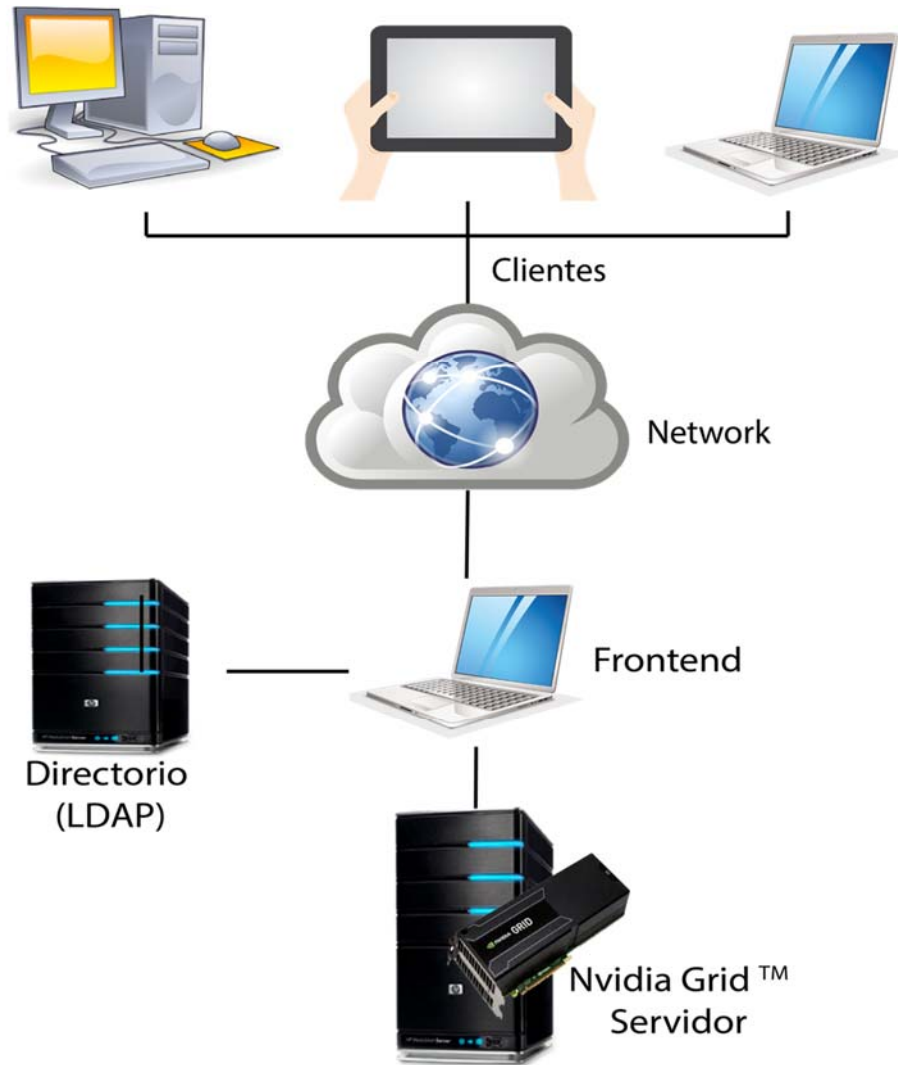


6.2.2 Docker

La implementación de la solución de visualización de aplicaciones ofrecida por Docker en tiene como principales beneficios la fácil instalación en el hardware con el que se dispone dando una gran portabilidad además ligereza de la solución, Docker no contiene un sistema completo, sino únicamente aquellas librerías, archivos y configuraciones necesarias para desplegar las aplicaciones también se encarga de la gestión del contenedor y de las aplicaciones mediante varias interfaces de usuario como es DockerUI o Portainer.

La arquitectura mostrada en la Figura 5 para una utilización de la solución Docker empieza por el servidor que tiene instalado la Nvidia Grid K2 [4.1.1.1] a este se le instala primero la distribución Linux que en este caso fue Debian agregándole las librerías de Nvidia diseñadas para la utilización de Docker de tal forma que las aplicaciones que están en los contenedores puedan utilizar las GPU del servidor y aprovechar los beneficios de la Nvidia GRID K2, luego está el nodo administrador o frontend también con Debian como distribución Linux y por último está el servidor LDAP que es el directorio de usuarios que funciona para validar el ingreso de los usuarios en el componente de acceso a la implementación.

Figura 5. Arquitectura Docker

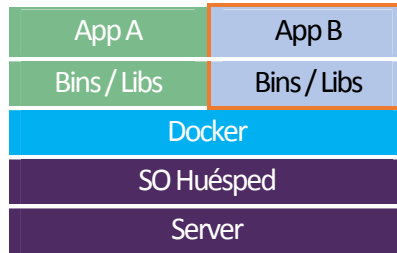


Gran parte del trabajo de implementación de esta solución es vía comandos, se instaló en el computador administrador con las especificaciones mostradas en la Tabla 3, la implementación ejecutada deja el Servidor y Frontend en la misma máquina pues para utilizar Docker no es necesario un hardware exclusivo para su funcionamiento.

El funcionamiento de la infraestructura nativa de Docker Tabla 7 tiene como característica principal la ejecución de aplicaciones de tal forma que cada contenedor tiene integrado las librerías y archivos necesarios para ejecutarse, pues en el momento de la compilación de la imagen con las especificaciones dadas para la aplicación por medio del Dockerfile esta compilación recoge la información del sistema operativo huésped y se adapta de tal forma

que si se cambia el sistema operativo no es necesario cambiar la estructura del Dockerfile si no por el contrario bastaría con volver a compilar la imagen para así posteriormente crear el contenedor con la misma.

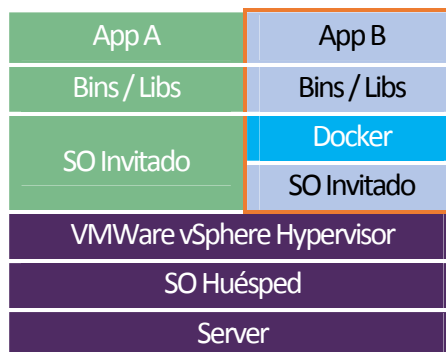
Tabla 7. Infraestructura Nativa Docker



6.2.3 Híbrida

Esta implementación híbrida para la infraestructura de la Tabla 8 para ejecutar la solución Docker dentro de la VMWare es una opción para contar con las dos soluciones, pues la infraestructura de la solución VMWare permite la creación de máquinas virtuales con los beneficios de la contar con la Nvidia GRID K2 entonces al colocar el servidor Docker en una de estas máquinas virtuales se implementaría de una manera básica la arquitectura necesaria para el funcionamiento.

Tabla 8. Infraestructura VM Docker en VMWare



En funcionamiento de esta implementación tendría las ventajas de la solución VMware y de Docker pero tendría los problemas o restricciones de ambas de tal forma que no es un servicio de visualización recomendado.

Para una implementación híbrida pero de manera contraria, es decir, la solución VMware [4.2.1.1] dentro de la Solución Docker [4.2.2] se tendría que hacerse más contenedores con

las imágenes de las partes restantes de los módulos de VMWare.

6.3 APLICACIONES

Existen una gran cantidad de aplicaciones científicas que son candidatas para implementarlas en el servicio de visualización pues son para un uso específicos y también son aplicaciones que funcionan como un entorno de desarrollo integrado, de esta manera se seleccionó las aplicaciones principalmente Paraview [3.7.1.1] y Netbeans [3.7.1.3] pero la solución Docker ofrece desplegar un gran número de aplicaciones que en la fase de pruebas probaron.

6.3.1 Características

Pensando en las necesidades al momento de utilizar una aplicación científica por parte de los investigadores, desarrolladores y demás personal se analizó que el gran problema era el acceso a máquinas con características de alto rendimiento y la utilización de aplicaciones de código abierto o licencias gratuitas debido al alto costo que tienen las licencias algunas aplicaciones en la industria, basados en esos principios las aplicaciones desplegadas son de código abierto o de licencia gratuita además que debido a la falta de máquinas con las configuraciones y especificaciones apropiada para la interacción correcta con ellas se veía disminuido el tiempo de desarrollo y el rendimiento de la misma.

Tabla 9. Características de Paraview en Docker

Paraview	
Versión	5.2
Licencia	Permiso BSD compatible con GPL
Análisis	Paraview es una aplicación que exige una máquina con configuración específicas y una variedad de librerías o software para sacarle el máximo provecho a la aplicación, algunas de ellas son Qt (versión 4.8) para el desarrollo de interfaces dentro de la aplicación y Python (versión 3.4) para la ejecución de script estas se necesita que estén instaladas en la computadora antes de intentar instalar Paraview pues son fundamentales para que la aplicación y sus desarrollos trabajen de manera fluida en la visualización de datos.
Ventajas	Al desplegar esta aplicación utilizando la solución Docker se hace más fácil el acceso a esta por parte de los usuarios además su utilización es más óptima pues el procesamiento, almacenamiento de proyectos y herramientas de desarrollo internas estarán en el servidor Docker y están disponibles de manera inmediata sin importar la computadora por la cual se accedan a ellas.

Tabla 10. Características de Netbeans en Docker

Netbeans IDE	
Versión	8.0.1
Licencia	CDDL - GPL2
Análisis	Netbeans es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java por esa razón el mayor reto al momento de instalarlo es contar con el kit de desarrollo de Java SE (JDK), que consta del entorno de ejecución de Java y de herramientas para la compilación, depuración y ejecución de aplicaciones escritas en lenguaje Java también los controladores de MySQL entre otros, es necesario contar con las herramientas correctas para cada sistema operativo para poder ejecutar esta aplicación de manera correcta y que no afecte su funcionamiento al momento de desarrollar los proyectos en esta aplicación.
Ventajas	Al poder desplegar esta aplicación por medio de la solución Docker hace más fácil para el usuario contar con un entorno listo y con las configuraciones apropiadas para el desarrollo de aplicaciones en Java, con un fácil acceso e ideal para los desarrollos colaborativos pues todos los repositorios serán accesados por los usuarios, pero también se puede hacer un despliegue y almacenamiento individual para cada usuario.

Una de las ventajas más significativas respecto a el manejo del almacenamiento es el poder comunicar varias aplicaciones de tal forma que puedan acceder a diferentes archivos y carpetas para así lograr utilizarlos en sus desarrollos de manera conjunta.

6.3.2 Restricciones

De acuerdo a la implementación de la solución Docker algunas de las restricciones al momento de desplegar las aplicaciones y en su interacción son:

- El acceso al almacenamiento por parte la aplicación está sujeta a la disponibilidad y configuración inicial de la Imagen Docker.
- El proceso de desarrollo del script de configuración para compilar una imagen Docker varía según la aplicación, a pesar de tener una estructura en el script ya predefinido por Docker el contenido varía de acuerdo a el análisis hecho para cada aplicación.
- Las aplicaciones como Paraview necesita de varios archivos para su configuración ideal además del Dockerfile, dando lugar que en el posible caso de necesitar desplegar aplicaciones con especificaciones diferentes se deban modificar estos archivos para

- crear otra imagen con esos requerimientos.
- La imposibilidad de usar el hardware de la máquina cliente para añadir funciones a la visualización, un ejemplo de ello son los puertos USB.
 - La implementación de aplicaciones con licencia comercial tendría que incluir otro archivo con las misma para luego ser indexado en el archivo de configuración, por ese motivo las licencias se verían expuestas.

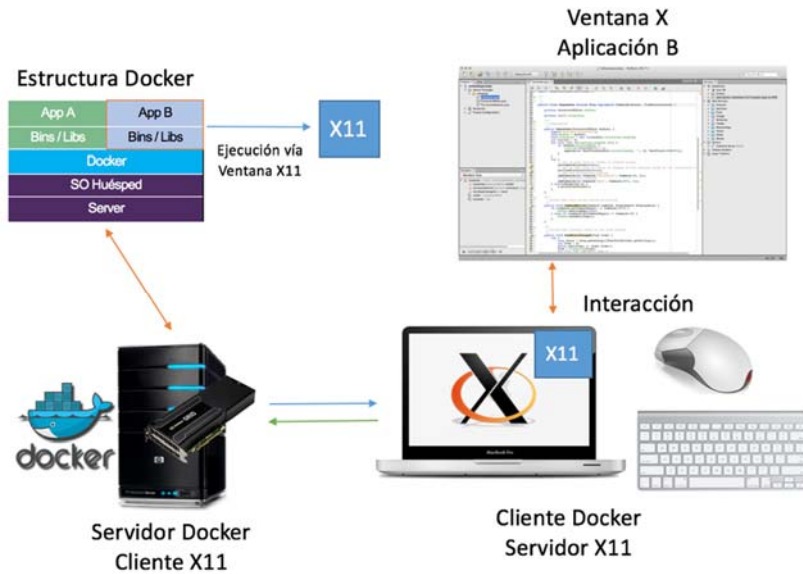
6.4 MÓDULO ACCESO

El componente de acceso que hará posible la administración e interacción con el servicio será implementada un versión de Portainer [18] pues permite la administración de los contenedores e imágenes entre otros, la interfaz web funciona para acceder a el servicio y con un inicio de sesión con los datos de acceso a Guane de esta forma no será necesario crear un registro nuevo para los usuarios sino que en cambio se aprovechara el directorio de usuarios que utilizan los recursos de supercómputo del grupo de Supercomputación y Cálculo Científico - SC3 , dentro del componente la primera página será el tablero general del servicio donde se muestra el estado actual del servidor mostrando el estado de las imágenes, los contenedores, el almacenamiento y redes disponibles.

Dentro de la sección de Imágenes se podrán ver las imágenes disponibles en el sistema y la opción de crear más y la sección de contenedores muestra de además de listar los contenedores disponibles en el servicio el estado de cada uno y la posibilidad de iniciarlos, pausarlos, borrarlos o modificarlos, estas secciones son las más importantes para el componente de acceso para el servicio. La sección de contenedores tiene un apartado donde se podrá ver el rendimiento de CPU, consumo de red y los log para cada contenedor que es muy importante para el administrador y para el usuario si desea ver lo que pasa en la aplicación que está ejecutando, de esta forma se pueden revisar posibles errores o advertencias del contenedor.

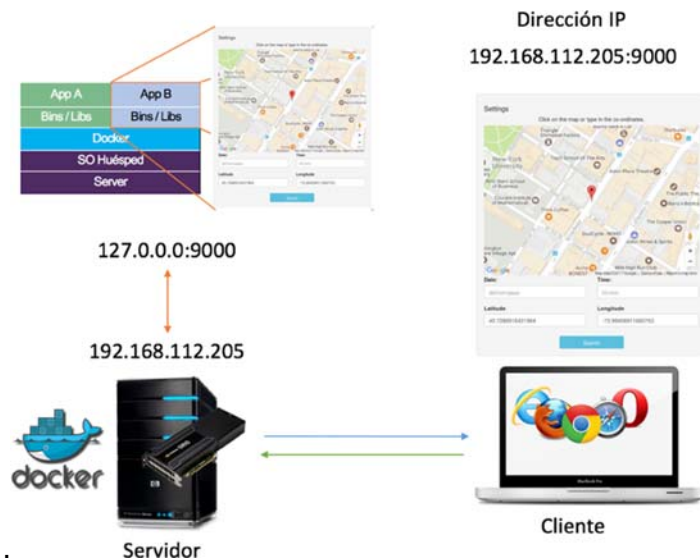
El despliegue de las aplicaciones puede variar de dos formas principalmente en primer lugar están las aplicaciones que se despliega visualmente gracias a el software de Sistema de Ventanas X con el protocolo X11 aprovechando sus características [4.2.3], es posible visualizar aplicaciones remotamente con la posibilidad de interactuar en ellas pero que de base están siendo ejecutadas en un contenedor utilizando la implementación Docker en el servidor, el flujo de la comunicación para visualización vía sistemas de ventanas X11 se puede ver en la es necesario señalar que para desplegar aplicación en computadores con sistema operativo de Apple es necesario instalar XQuartz como manejador de ventanas pues X11 necesita uno para poder funcionar en este sistema.

Figura 6. Flujo de comunicación en una aplicación vía Sistema de Ventanas X11



En segundo lugar están las aplicaciones de entorno web que mediante un servidor web como muestra la Figura 7 desplegado en el contenedor con la configuración para ejecutar la aplicación en el local host en un puerto definido especificado al momento del despliegue siendo posible acceder a enlazando el local host del contenedor con el local host del servidor un ejemplo es poder acceder a una aplicación con la dirección 192.168.112.205:9000 donde el 9000 es el puerto donde se despliega la aplicación web.

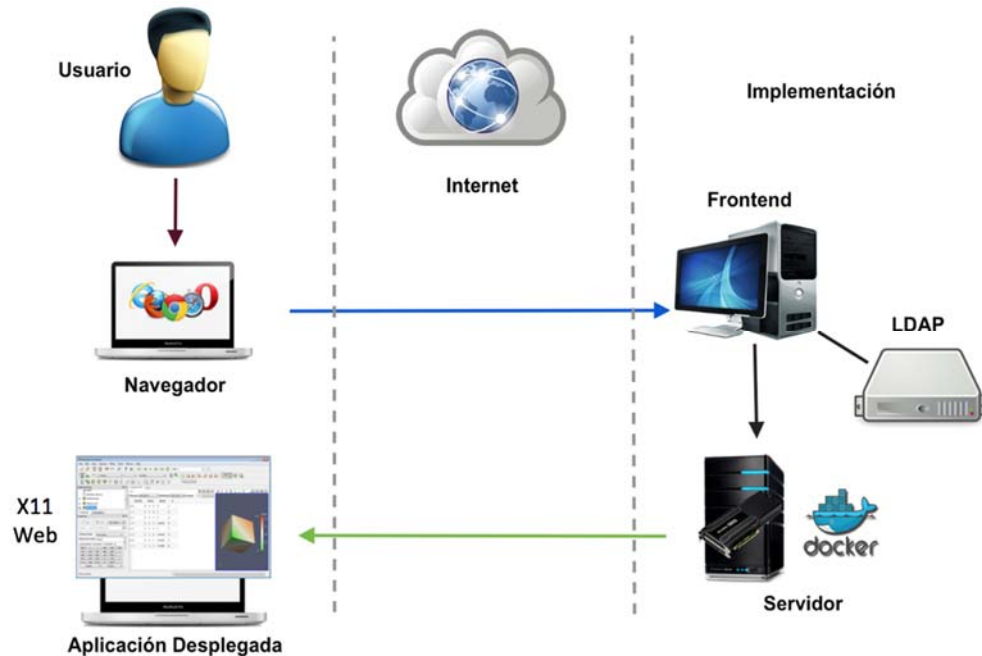
Figura 7. Flujo de comunicación en una aplicación web en un contenedor.



El esquema del componente de visualización del servicio de visualización de aplicaciones se puede englobar como se ve en la Figura 8, donde el usuario accede a la plataforma por

medio de un navegador e inicia la aplicación que está en alguno de los contenedores y dependiendo del tipo de aplicación la visualización será por medio del Sistema de Ventanas X11 o web.

Figura 8. Esquema del Componente de Visualización Docker



6.5 CONFIGURACIÓN

La configuración e implementación desarrollada para el servicio de visualización se compone de la sección para el servidor donde estará los criterios de instalación de la Solución Docker además de la configuración de las imágenes y los contenedores, el frontend de la solución que será la vía de acceso por parte de los usuarios al sistema de visualización y por último la configuración necesaria del cliente para poder visualizar las aplicaciones desplegadas bajo el Sistema de Ventanas X11.

6.5.1 Servidor Docker

Para comenzar la configuración del servidor docker se inició con la instalación del sistema operativo Debian GNU/Linux en la versión 8 conocida como Jessie siguiendo la guía [19] con un entorno gráfico, la tecnología Docker es solo posible en sistemas operativos basados en Unix, como se tiene instalado en el servidor un componente de hardware de Nvidia es necesario instalar los controladores desarrollados por ellos para la tecnología Docker esto se hizo siguiendo la guía [20] proporcionadas por Nvidia gracias a estos es posible utilizar

las GPU de servidor por parte de las aplicaciones en los contenedores al incluir los controladores al momento de crear la imagen a partir del Dockerfile pues nativamente Docker no incluye esta característica, luego se hizo la instalación de la tecnología Docker en el servidor de acuerdo a la guía [15] proporcionadas por Docker.

El siguiente paso es la configuración del archivo de configuración para cada imagen que es llamado "Dockerfile" en la solución esta configuración tiene la estructura mostrada en la Tabla 11.

Tabla 11. Estructura archivo Dockerfile

ESTRUCTURA DOCKERFILE			
#	Instrucción	Descripción	Ejemplo
1	FROM	Indica la imagen que tomamos como base.	FROM ubuntu:14.04
2	MANTAINER	Especifica el autor de la imagen	MAINTAINER Carlos Parada "carlos.parada@correo.uis.edu.co "
3	ENV	Definimos una variables de entorno en la imagen base.	ENV HOME /home/developer
4	RUN	Ejecuta una sentencia sobre la imagen base	RUN apt-get update && apt-get install -y software-properties-common &&
5	ADD	Copia los nuevos archivos o directorios y los agrega al sistema de archivos de la imagen en la ruta de acceso <dest>.	ADD run /usr/local/bin/netbeans
6	USER	Define el usuario del contenedor	USER developer
7	WORKDIR	La instrucción WORKDIR establece el directorio de trabajo para cualquier instrucción	WORKDIR /home/developer
8	VOLUME	Funciona para crear un punto de montaje con el nombre especificado y lo marca como un de volúmenes montado externamente desde el host nativo u otros contenedores.	VOLUME /.netbeans-docker /home/developer/.netbeans
9	EXPOSE	Exponemos el puerto 80 del contenedor para que pueda ser mapeado por la máquina anfitrión.	EXPOSE 8000
10	ENTRYPOINT	Indicamos que se ejecute un código cada vez que arranquemos el contenedor.	ENTRYPOINT <código> -D FOREGROUND

Al tener el Dockerfile configurado correctamente es necesario construir la imagen de acuerdo a esos parámetros, la sentencia para ello es:

```
# sudo docker build -t nombre_imagenes.  
# sudo docker build -t nombre_repositorio/nombre_imagenes.
```

Es necesario tener en cuenta que para ejecutar este comando deberá el usuario tener acceso de administrador o root, si se ejecuta el comando desde el directorio del Dockerfile de la aplicación la construcción se basará en ese archivo así contenga el nombre del repositorio la sentencia en caso que el comando sea ejecutado en un directorio donde no exista un archivo de configuración Dockerfile se debe agregar el nombre de repositorio

creado en la página <http://hub.docker.com> donde a su vez debe estar el archivo de configuración Dockerfile.

Otra consideración es la necesidad del punto al final pues es fundamental para que se ejecute el comando, en la consola se mostrará el estado de la creación de la imagen es decir mostrará el paso que está ejecutando de acuerdo a la estructura del Dockerfile, al final este proceso será posible ver la imagen creada en Docker mediante el comando:

```
# sudo docker images -a
```

El resultado de ejecutar este comando listará las imágenes disponibles en la solución Docker además de mostrar su estado actual dará algunas características propias de la misma que servirán para monitorearlas, un ejemplo de la respuesta es:

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
carlosdparada/netbeans	latest	e0cfc4d179bb	2 days ago	1.171 GB
portainer/portainer	latest	91b49618d792	3 days ago	9.162 MB
carlosdparada/paraview	latest	0b0996b457f7	3 days ago	3.761 GB

En la sección de REPOSITORY es el nombre que se le dio a la imagen al crearla, el TAG es la versión de la imagen esta se especifica en la sección del FROM del Dockerfile, la IMAGE ID es un índice creado automáticamente por Docker, en la parte de CREATED es la fecha cuando se creó esta imagen y SIZE muestra el tamaño de esta imagen en la solución Docker.

Para la administración de las imágenes los siguientes comandos son necesarios para realizar dicha administración:

- Borrar una imagen
 - # sudo docker rmi <IMAGE ID>
- Hacer una copia de una imagen para conservar su configuración
 - # sudo docker commit <IMAGEID:TAG>
 - Útil si se va volver a cargar la imagen de un Dockerfile modificado.
- Guardar una imagen un archivo .tar
 - # sudo docker save <IMAGE ID> > <Repository>.tar
- Cargar una imagen un archivo .tar
 - # sudo docker load < <Repository>.tar
- Eliminar las imágenes
 - # sudo docker rmi \$(docker images -f "dangling=true" -q)

Al tener la imagen generada el siguiente paso es empaquetarla en un contenedor esto se hace por medio del comando ejecutado en modo administrador o root:

```
# sudo docker run -d -it -e DISPLAY=$DISPLAY -v /tmp/.X11-unix:/tmp/.X11-unix -v `pwd`:/workspace -name allison_netbeans nombre_imagen
```

El comando anterior es para una aplicación visualizada por medio del Sistema de Ventanas X11, la estructura del comando RUN tiene elementos con fines específicos como:

- **-d** se utiliza para que contenedor que siga corriendo por debajo después de terminar la ejecución del comando en el lanzamiento en caso de no tenerlo no seguirá corriendo en segundo plano.
- **-it** se utiliza para abrir un canal con el contenedor y así poder ejecutar una terminal.
- **-e** sirve para asignar una variable de entorno en este caso \$DISPLAY es la dirección IP 0.0.0.0:10.
- **-v** es un volumen del contenedor que puede ser un directorio en el servidor, u otro directorio de otro contenedor o puede ser del localhost, en este caso se están vinculando las carpetas para el Sistema de Ventanas X11.
- **-p** este comando sirve para asignar puertos entre el local y el contenedor
- **-link** este permite enlazar un contenedor con otro.
 - --link aplicacionA:aplicacionB
- **-name** funciona para asignarle un nombre.
- **nombre_imagen** al final se incluye el nombre de la imagen que se desea empaquetar.

Una vez teniendo el contenedor desplegado con la aplicación y logrando su visualización e interacción, se puede listar los contenedores que están actualmente en Docker mediante el comando:

```
# sudo docker ps -a
```

El resultado de ejecutar este comando además de listar los contenedores en Docker dará información sobre su estado, el puerto en el que está siendo visualizado (aplicaciones web), el comando de ejecución además de su nombre, un ejemplo de la respuesta es:

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
520c34091d38	portainer/portainer	"/portainer --logo 'h'"	15 hours ago	Up 15 hours
59f0ff4dc7c	carlosdparada/netbeans	"/bin/sh -c /usr/local"	21 hours ago	Exited (0) 17 hours ago
110aaedd3e34	app_gnss	"python main.py"	2 days ago	Up 15 hours

A su vez se puede hacer una administración de los contenedores disponibles en Docker mediante los comandos siguientes:

- Listar los contenedores
 - # sudo docker ps -a
- Parar uno de los contenedores
 - # sudo docker stop <CONTAINER ID>
- Iniciar nuevamente un contenedor
 - # sudo docker run <IMAGE>
- Eliminar un contenedor
 - # sudo docker rm <CONTAINER ID>
- Parar todos los contenedores
 - # sudo docker stop \$(docker ps -a -q)
- Eliminar todos los contenedores
 - # sudo docker \$(docker ps -a -q)
- Desplegar un consola dentro del contenedor

- o `# sudo docker rm -i -t <CONTAINER ID> /bin/bash`

De esta forma el servidor está configurado para poder utilizar la solución Docker y además siguiendo los parámetros de construcción de Dockerfile para la visualización de aplicaciones se tendrían disponibles para el uso de los usuarios.

6.5.2 Frontend

Para comenzar la configuración del frontend se aprovechó la solución Docker implementada en el Servidor, pues la propuesta del componente de acceso está basada en la adaptación y utilización de la Portainer, al ser una interfaz de usuario de código abierto permite esta adaptación para el servicio de visualización de aplicación y su administración.

La instalación de esta interfaz de usuario se hizo en base a la documentación [21] ofrecida por los desarrolladores, la modificación del código base está disponible en el repositorio [22].

El comando para compilar la imagen de la interfaz de usuario Portainer fue:

```
# sudo docker build --rm -t portainer -f build/linux/Dockerfile .
```

Y el comando para la ejecución de la imagen generada con su visualización en el puerto 9000 del contenedor y enlazarlo con el puerto 9000 del servidor Docker fue el siguiente:

```
# sudo docker run -d -p 9000:9000 -v /var/run/docker.sock:/var/run/docker.sock --v /path/on/host/data:/data portainer --logo "sc3_logo.png "
```

Por ser una aplicación web dentro de un contenedor la visualización del componente de acceso al servicio de visualización de aplicaciones por parte de los usuarios se hace en la dirección IP 192.168.112.205:8080 y en los administradores se hace en la dirección IP 192.168.112.205:9000/

6.5.3 Clientes

La configuración que necesitan los usuarios del servicio de visualización de aplicaciones para poder utilizarla empieza primero con el acceso a internet por medio de un navegador web, para este requerimiento no se necesita una versión en especial para cada navegador puede ser el de preferencia del usuario (Google Chrome, Mozilla Firefox, Safari u otro), el segundo es contar con una cuenta para utilizar los recursos de supercómputo del grupo de Supercomputación y Cálculo Científico este se puede solicitar accediendo a <https://www.sc3.uis.edu.co//usuarios//index2.php>, luego de contar con el usuario y contraseña podrá iniciar sesión en el componente de acceso del servicio este se accede a actualmente a la dirección <https://192.268.112.205:9000>, donde podrá visualizar el panel de inicio de la misma luego de estar en la sección de contenedores como se especificó anteriormente existen 2 formas de desplegar las aplicaciones vía web y con el Sistema de

Ventanas X11 para la primera opción se utiliza el mismo navegador por donde accedió a el componente de acceso, sin embargo para la segunda opción es necesario instalar un componente para desplegar el Sistema de Ventanas X11 que varía según el sistema operativo.

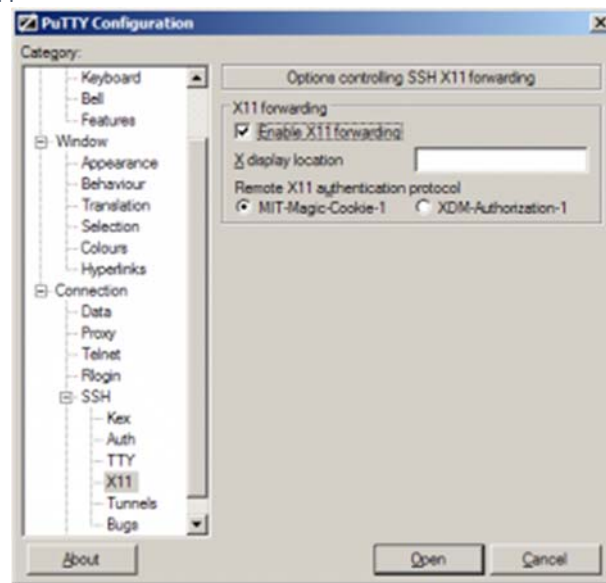
6.5.3.1 Windows

En windows se recomienda instalar PuTTY desde la página <http://www.putty.org>.

Para su configuración se necesita navegar hasta la sección de conexiones posteriormente al apartado SSH, luego seleccionar la opción X11 en el panel lateral se marcar la opción “Habilitar reenvío X11” y finalmente en el botón inferior Abrir/Open.

La Figura 9 muestra la interfaz de la aplicación PuTTY con la sección para la configuración para poder desplegar las aplicaciones del Sistema de Visualización implementado.

Figura 9. Configuración PuTTY

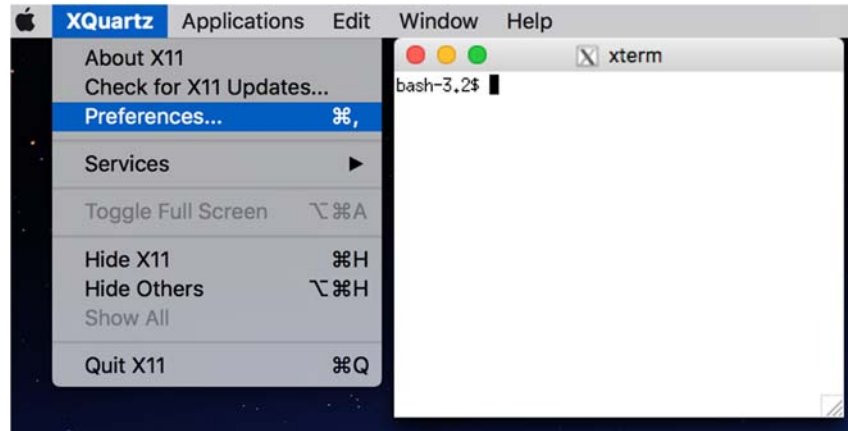


6.5.3.2 Mac OS

En MAC OS existe un componente avalado por Apple que es Xquartz , este se puede descargar desde la página <https://www.xquartz.org> y descargando la versión específica para la versión de MAC OS que tenga la máquina del usuario.

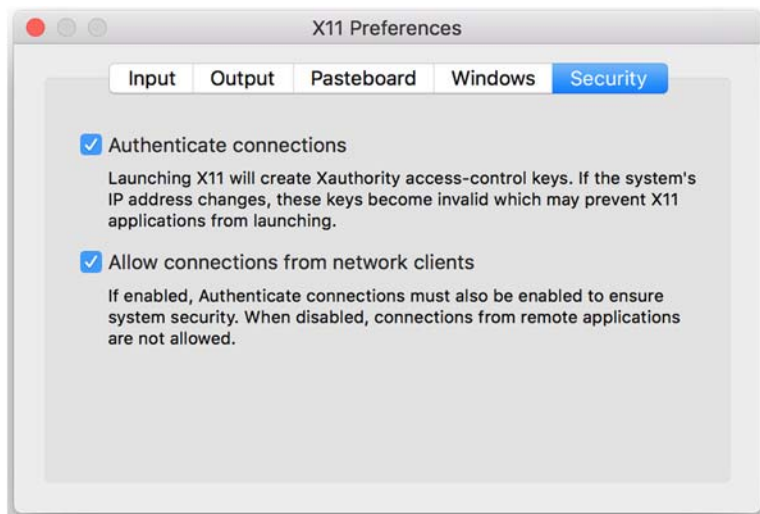
Para su configuración es necesario entrar a Preferencias en el menú Xquartz como la muestra la Figura 10.

Figura 10. Acceder a las Preferencias en Xquartz



Luego se marcarán las opciones Autenticar conexiones y permitir conexiones desde clientes de red.

Figura 11. Cuadro de Preferencias de XQuartz



Luego obtenga su dirección IP desde el Sistema de Preferencias de MAC OS o pues de ejecutar en consola de comandos el siguiente comando para obtener su dirección IP y guardarla en la variable \$ip

```
# ip = $(/sbin/ifconfig en0 | gre inet | awk`$1=="inet" {print $2}`) xhost + $ip
```

Finalmente se ejecuta el siguiente comando para permitir conexiones a su la maquina del usuario es que el Xquartz X Server

```
# xhost + $ip
```

Luego de ese comando se mostrará un mensaje que le informa que su dirección IP está lista para recibir conexiones, estos serian los pasos para configurar el Sistema de Ventanas X11.

6.5.3.3 Linux

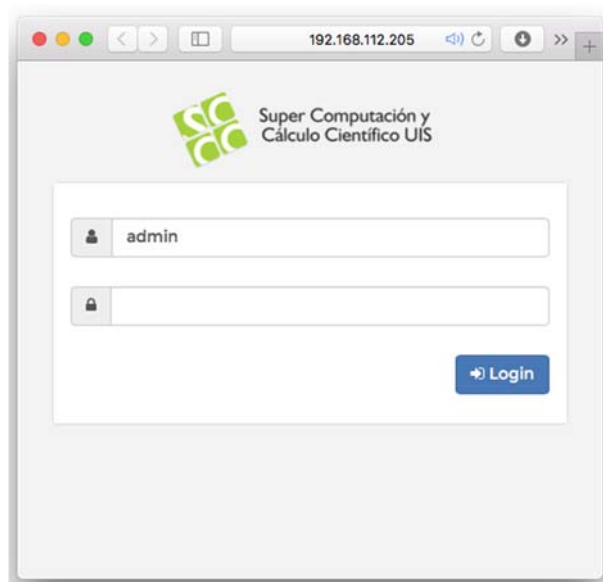
Las distribuciones de Linux que cuentan con un entorno gráfico tienen instalado y configurado el Sistema de Ventanas X11, en caso contrario deberá seguir la guía de instalación [23].

6.6 PROTOTIPO

El prototipo de la implementación del servicio de visualización de aplicaciones empieza con el componente de acceso como se puede observar en el la Figura 12, en esta pagina accesada actualmente en la direccion IP <http://192.168.112.205:9000>, los datos iniciales seran: el nombre de usuario y la contraseña.

Los datos necesario para iniciar sesion son los mismo para acceder a los recursos de supercómputo del grupo de Supercomputación y Cálculo Científico - SC3.

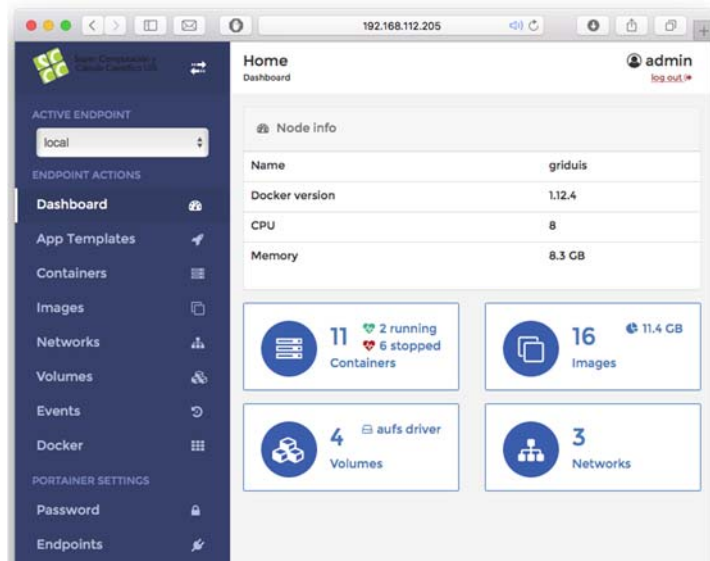
Figura 12. Prototipo de Inicio de Sesión



Una vez hecho el inicio de sesión y validados los datos la sección inicial que se le presenta a el usuario es como la mostrada en la Figura 13, donde se presentan de forma general el número de contenedores en el sistema, también la cantidad de ellos que se están ejecutando y los que se encuentran pausados, las imágenes de aplicaciones disponibles para su uso y el espacio que actualmente ocupan en el servidor Docker, el número de discos disponibles, como nota el reconoce la partición del disco existen al general las aplicaciones, por ende en un futuro cada sección del disco estaría disponible para cada tipo de proyectos o se vincularía con el espacio asignado por otro dedicado especialmente para algún desarrollo.

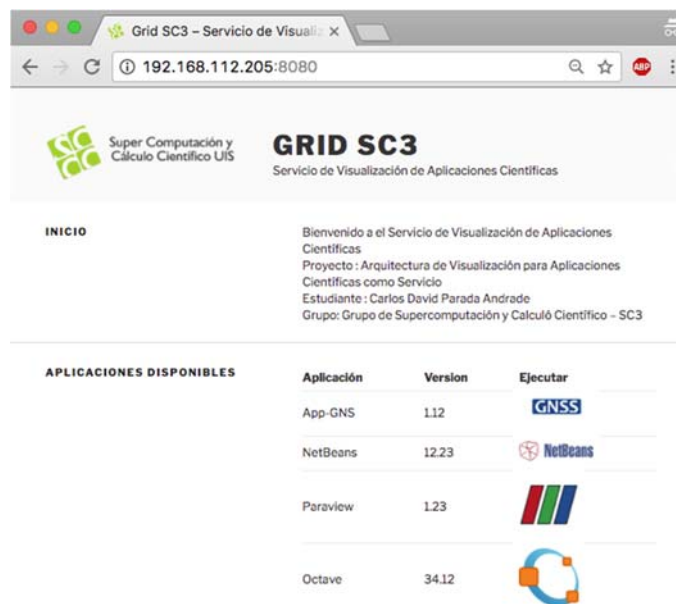
Esta sección al igual que las siguientes contiene un menú lateral para poder desplazarse entre las diferentes secciones del componente.

Figura 13. Prototipo Tablero Administrador de Docker



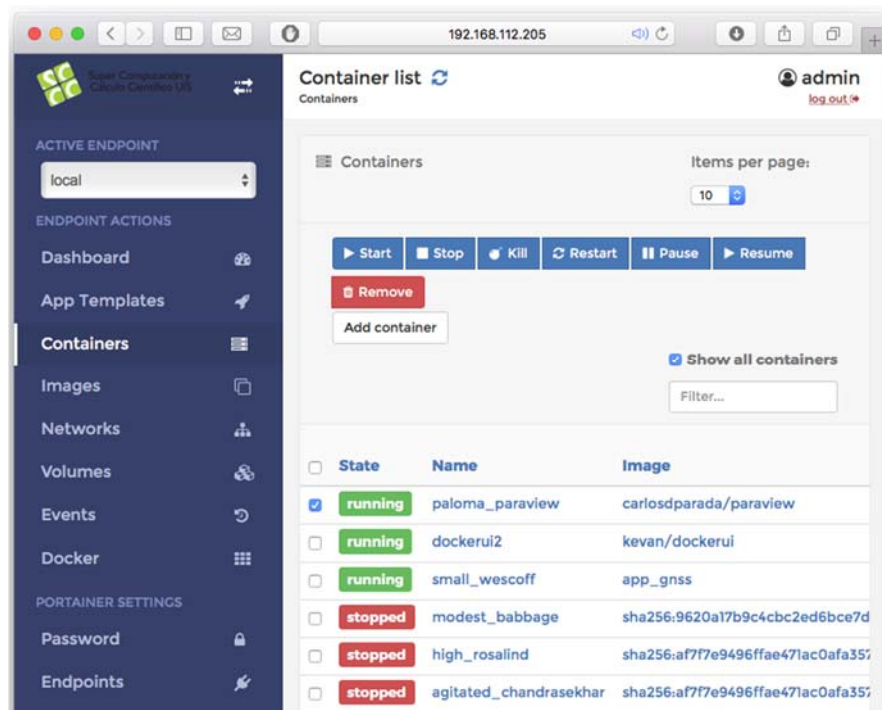
En la Figura 14 se muestra la página de inicio a la que accede un usuario después de haber ingresado su usuario y contraseña, en esta sección al seleccionar la aplicación, la visualización iniciará por medio de página web o ventanas de Sistema X11.

Figura 14. Prototipo Inicio Usuario de Docker



Al seleccionar la opción de Contenedores en el menú lateral al usuario se le presentara una sección como la mostrada en la Figura 15 esta contiene en forma de lista los contenedores disponibles, tanto activos como inactivos además se le presenta las opciones disponibles para los mismos, como iniciarlo, pausarlos o eliminarlo entre otras.

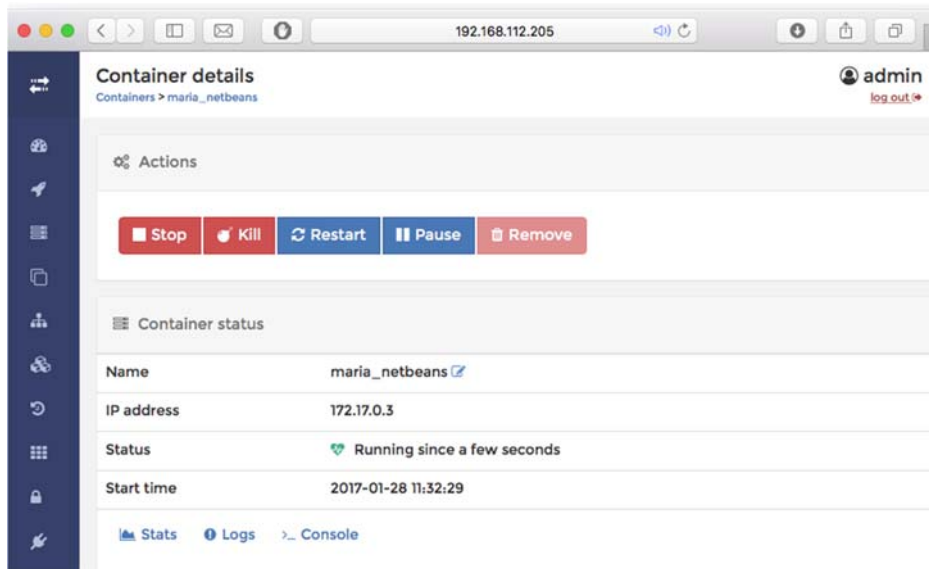
Figura 15. Prototipo Contenedores Disponibles en Solución Docker



Al seleccionar de la lista un contenedor, lo llevara a la sección de Detalles del Contenedor como se muestra en la Figura 16, en esta se le muestran más detalles del contenedor seleccionado.

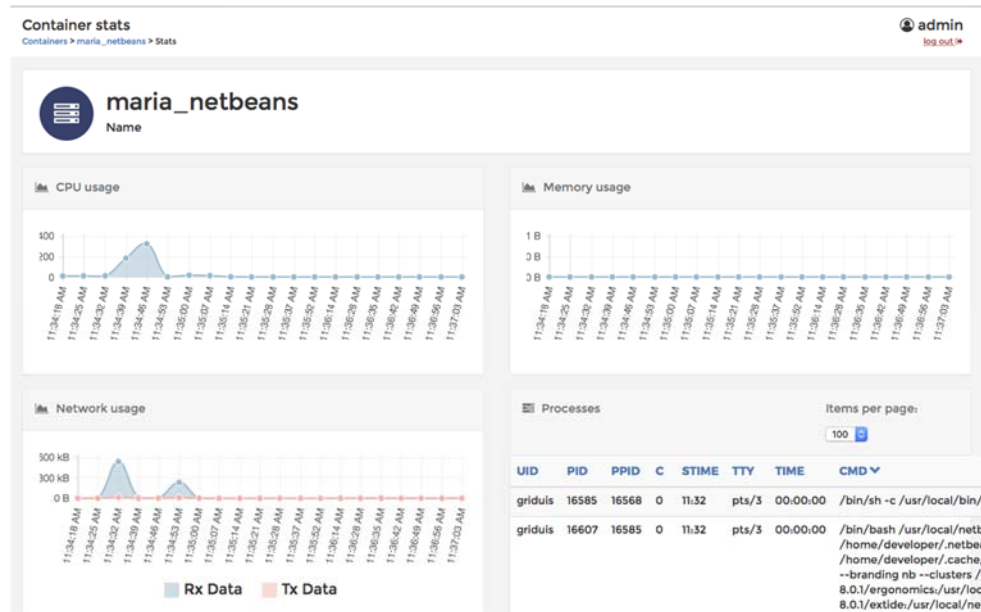
Desde esta sección podrá hacer el seguimiento correspondiente para cada uno, pues podrá acceder a las estadísticas del contenedor, los log del mismo y si una consola, esta última depende de la estructura del contenedor, en la mayoría de las aplicaciones web esta consola no es disponible.

Figura 16. Prototipo para el seguimiento de un contenedor



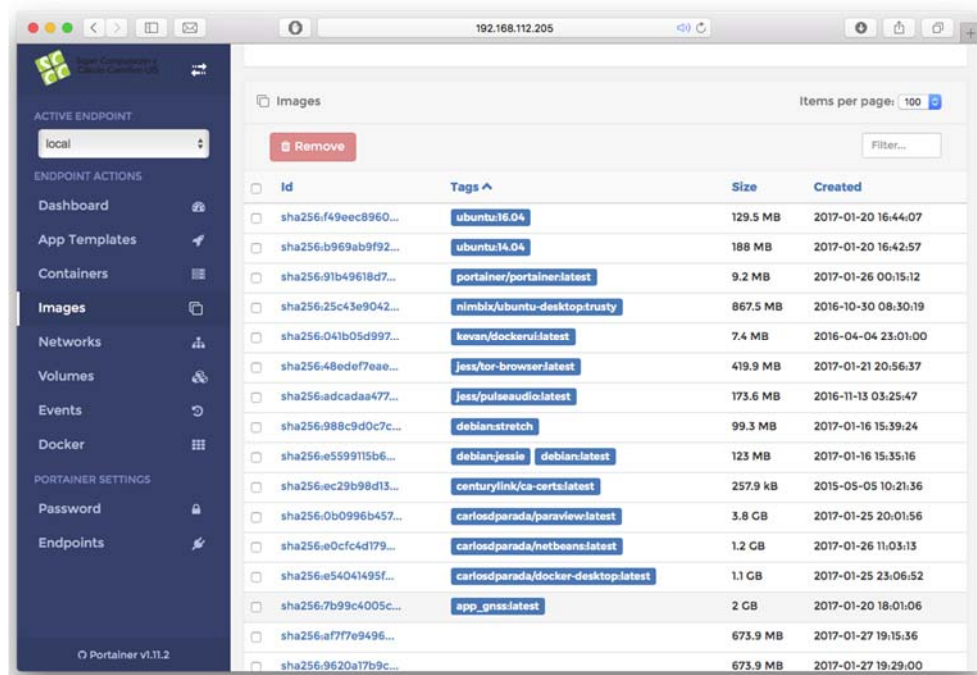
En el apartado de estadísticas del contenedor se muestra como la Figura 17 usuario un comportamiento del contenedor en el Servidor Docker, mostrara el uso de CPU, Memoria y Red usada en una línea de tiempo además de la sección interna para los procesos que se ejecutan o ejecutaron en el contenedor.

Figura 17. Prototipo apartado de Estadísticas del Contenedor



Al seleccionar la opción de Imágenes en el menú lateral al usuario se le presentara una sección como la mostrada en la Figura 18, en esta sección se listarán todas las imágenes disponibles en el Servicio.

Figura 18. Prototipo Imágenes Disponibles en Solución Docker



Id	Tags	Size	Created
sha256:f49ecc8960...	ubuntu:16.04	129.5 MB	2017-01-20 16:44:07
sha256:b969ab9f92...	ubuntu:14.04	188 MB	2017-01-20 16:42:57
sha256:91b49618d7...	portainer/portainer:latest	9.2 MB	2017-01-26 00:15:12
sha256:25c43e9042...	nimbix/ubuntu-desktop:trusty	867.5 MB	2016-10-30 08:30:19
sha256:041b05d997...	kevan/dockerui:latest	7.4 MB	2016-04-04 23:01:00
sha256:48ede70eae...	jesz/tor-browser:latest	419.9 MB	2017-01-21 20:56:37
sha256:adcadaa477...	jesz/pulseaudio:latest	173.6 MB	2016-11-13 03:25:47
sha256:988c9d0c7c...	debian:stretch	99.3 MB	2017-01-16 15:39:24
sha256:e559115b6...	debian:jessie debian:latest	123 MB	2017-01-16 15:35:16
sha256:ec29b98d13...	centurylink/ca-certs:latest	257.9 kB	2015-05-05 10:21:36
sha256:0b0996b457...	carlosdparada/paraview:latest	3.8 GB	2017-01-25 20:01:56
sha256:e0cfc4d179...	carlosdparada/netbeans:latest	1.2 GB	2017-01-26 11:03:13
sha256:e54041495f...	carlosdparada/docker-desktop:latest	1.1 GB	2017-01-25 23:06:52
sha256:7b99c4005c...	app_gnsz:latest	2 GB	2017-01-20 18:01:06
sha256:af77e9496...		673.9 MB	2017-01-27 19:15:36
sha256:9620a17b9c...		673.9 MB	2017-01-27 19:29:00

7. CONCLUSIONES

A partir del trabajo realizado para el desarrollar los objetivos específicos de este proyecto se dan las siguientes conclusiones.

1. Los componentes de hardware necesarios para la implementación de un servicio de visualización de aplicaciones en un Sistema Heterogéneo dependen del componente de software que se desea implementar.
2. La característica principal de las aplicaciones seleccionadas para la implementación en el SAV es que sean posible instalarlas en sistemas operativos basados en Unix y de preferencia que sean de código abierto o de licencias gratuitas, para evitar la interrupción del acceso a ella por parte de los usuarios por el vencimiento de licencias o renovaciones de las mismas.
3. La visualización de aplicaciones por medio de los contenedores Docker, es más portable, ligera y de fácil administración que la visualización por medio de maquinas virtuales pues implica menos costos de configuración y mantenimiento para el usuario por su facil instalacion al ser desplegadas sin importar el Sistema Operativo, el tamaño de las imagens es mas reducido, cuenta con licencia OpenGL ademas de la flexibilidad para entornos de desarrollo y produccion.
4. El componente de acceso al servicio de visualización mediante una interfaz de usuario vía Web es la opción más efectiva, pues basándose que en se necesita conexión a internet para acceder a el servicio es considerablemente más práctico que desarrollar un componente de visualización que necesite una instalación por parte del usuario en la maquina cliente.
5. Al implementar Portainer en el componente de administración se aprovecha que está sistema de contenedores pues se evita la configuraciones de conexiones externas al servidor del SVA
6. El componente de acceso al SVA vía Web es la opción más recomendada, pues vía X11 el costo de implementación y recursos es mayor.
7. La configuración definida anteriormente de los componentes para el SVA, es la esencial para su funcionamiento.

8. RECOMENDACIONES Y TRABAJO FUTURO

El desarrollo e implementación del componente de visualización para el servicio de visualización de aplicaciones aún no concluye, por el momento el módulo no contiene una gran variedad de acciones disponibles sólo mediante la ejecución de comando en la consola del servidor además no se ha configurado para el servicio de visualización en su totalidad las funciones disponibles en Portainer.

Con respecto a la optimización de los contenedores desplegados aún no se han explorado algunas configuraciones que permiten aumentar o restringir el acceso del contenedor a la memoria RAM y ancho de banda del Servidor Docker.

Por último, debido a que el prototipo de servicio de visualización de aplicaciones se encuentra funcional, se puede sugerir una implementación a mayor escala que abarque un grupo en específico de la Universidad Industrial de Santander pues así esta nueva implementación podrá ser utilizada continuamente de tal forma que obtengan datos de rendimiento e informes de errores por parte los usuarios llevando a que el desarrollo sea optimizado en las secciones específicas o críticas.

9. LIMITACIONES

Las número de aplicaciones científicas a implementar en SVA son restringidas.

No es posible implementar aplicaciones Windows.

La visualización de las aplicaciones del SVA por medio de Sistema de ventanas X11, necesita configuraciones estrictas en la parte del Cliente.

REFERENCIAS

- [1] L. M. Moreno, Computacion paralela y entornos heterogeneos, Tenerife: Soportes Audiovisuales e Informaticos, 2004.
- [2] J. Pedro, «Sistema Heterogéneos,» 2014. [En línea]. Disponible en: <http://sistemas-eterogeneos.blogspot.com.co>. [Último acceso: 9 2016].
- [3] O. A. Mejia, «Computación en la nube,» Universidad Autónoma de México, [En línea]. Disponible en: <http://www.izt.uam.mx/newpage/contactos/anterior/n80ne/nube.pdf>.
- [4] VMware, «Virtualizacion,» 21 5 2014. [En línea]. Disponible en: <http://www.vmware.com/latam/solutions/virtualization.html>. [Último acceso: 21 10 2016].
- [5] Nvidia, «Nvidia Grid,» [En línea]. Disponible en: <http://www.nvidia.es/object/nvidia-grid-es.html>.
- [6] Microsoft, «Virtualización,» [En línea]. Disponible en: <https://www.microsoft.com/spain/virtualizacion/products/application/default.mspix>.
- [7] A. Plasencia, «Virtualización de Aplicaciones,» [En línea]. Disponible en: <https://aprendizajeubicuo.wordpress.com/2011/05/19/virtualizacion-de-aplicaciones/>.
- [8] N. Toole, «Visualisation and Virtual Reality,» Open Chanel Foundation, Marzo 2004. [En línea]. Disponible en: http://www.openchanelfoundation.org/discipline/Visualization_and_Virtual_Reality/ . [Último acceso: Julio 2016].
- [9] N. Sundown, J. Byungil y R. Luc, «Remote Visualisation of Large Scale Data for Ultra-High Resolution Display Environments,» Electronic Visualisation Laboratory, [En línea]. Disponible en: https://www.evl.uic.edu/documents/sc09_ultravis_fullpaper_nam.pdf.
- [10] P. K. S. Roldán, «Animación y Visualización de Fenómenos Naturales,» Universidad de las Américas Puebla, 13 Enero 2003. [En línea]. Disponible en: http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/suarez_r_pk/capitulo4.pdf. [Último acceso: 9 2016].
- [11] Paraview, «ParaView Guide,» Kitware Inc., [En línea]. Disponible en: <http://www.paraview.org/documentation/>.
- [12] Kitware, «Visualization toolkit (VTK),» Kitware Inc., [En línea]. Disponible en: <http://www.vtk.org/> .
- [13] VMware, «vGPU Deployment Guide Citrix on vSphere,» 9 2016. [En línea]. Disponible en: http://images.nvidia.com/content/pdf/grid/guides/vGPU_Deployment_Guide_Citrix_on_vSphere_Tech_Pub_v01_final.pdf. [Último acceso: 2016].
- [14] Nvidia, «VMware Horizon 6,» [En línea]. Disponible en: <http://www.nvidia.es/object/vmware-horizon-6-es.html> .
- [15] Docker, «Get Docker for Debian,» 12 10 2016. [En línea]. Disponible en: <https://docs.docker.com/engine/installation/linux/debian/>. [Último acceso: 2016].
- [16] Hewlett Packard Enterprise, «HPE ProLiant ML350 Gen9 Server,» [En línea]. Disponible en: <https://www.hpe.com/us/en/product-catalog/servers/proliant-servers/pip.hpe-proliant-ml350-gen9-server.7271259.html> .
- [17] VMware, «Horizon Suite,» VMware, 23 10 2016. [En línea]. Disponible en: <http://www.vmware.com/products/horizon.html>. [Último acceso: 2016].
- [18] Portainer, «Portainer,» [En línea]. Disponible en: <https://portainer.readthedocs.io/en/stable/>. [Último acceso: 2016].
- [19] Debian, «1.1 Guía de instalación de Debian 8.0 'Jessie',» Debian, 2014. [En línea]. Disponible en: <https://servidordebian.org/es/jessie/install/guide>. [Último acceso: 2016].

- [20] Nvidia, «Nvidia Docker,» 11 2015. [En línea]. Disponible en: <https://github.com/NVIDIA/nvidia-docker>. [Último acceso: 2016].
- [21] Portainer.io, «Portainer Documentation,» 2016. [En línea]. Disponible en: <https://portainer.readthedocs.io/en/stable/index.html>. [Último acceso: 2017].
- [22] C. D. Parada, «Visualization Architecture For Scientific Applications As A Service,» Grupo de Supercomputacion y Calculo Cientifico, 2015. [En línea]. Disponible en: http://forge.sc3.uis.edu.co/redmine/projects/service_visualization. [Último acceso: 2017].
- [23] Beyond Linux®, «Introduction to Xorg-7,» Beyond Linux®, 19 Diciembre 2016. [En línea]. Disponible en: <http://www.linuxfromscratch.org/blfs/view/cvs/x/xorg7.html>. [Último acceso: Enero 2017].
- [24] A. Chambers, «Life Science Trends – Data Visualization as a Service (DVaaS),» MavenWave, [En línea]. Disponible en: <http://www.mavenwave.com/white-papers/life-science-trends-were-watching/>.
- [25] AMD, «Remote Visualisation of Large Scale Data for Ultra-High Resolution Display Environments,» [En línea]. Disponible en: <http://developer.amd.com/resources/heterogeneous-computing/what-is-heterogeneous-system-architecture-hsa/>.

BIBLIOGRAFÍA

- [1] A. Chambers, «Life Science Trends – Data Visualization as a Service (DVaaS),» MavenWave, [En línea]. Available: <http://www.mavenwave.com/white-papers/life-science-trends-were-watching/> .
- [2] A. Plasencia, «Virtualización de Aplicaciones,» [En línea]. Available: <https://aprendizajeubicuo.wordpress.com/2011/05/19/virtualizacion-de-aplicaciones/>.
- [3] AMD, «Remote Visualisation of Large Scale Data for Ultra-High Resolution Display Environments,» [En línea]. Available: <http://developer.amd.com/resources/heterogeneous-computing/what-is-heterogeneous-system-architecture-hsa/> .
- [4] Beyond Linux®, «Introduction to Xorg-7,» Beyond Linux®, 19 Diciembre 2016. [En línea]. Available: <http://www.linuxfromscratch.org/blfs/view/cvs/x/xorg7.html>. [Último acceso: Enero 2017].
- [5] C. D. Parada, «Visualization Architecture For Scientific Applications As A Service,» Grupo de Supercomputacion y Calculo Cientifico, 2015. [En línea]. Available: http://forge.sc3.uis.edu.co/redmine/projects/service_visualization. [Último acceso: 2017].
- [6] Debian, «1.1 Guía de instalación de Debian 8.0 'Jessie',» Debian, 2014. [En línea]. Available: <https://servidordebian.org/es/jessie/install/guide>. [Último acceso: 2016].
- [7] Docker, «Get Docker for Debian,» 12 10 2016. [En línea]. Available: <https://docs.docker.com/engine/installation/linux/debian/>. [Último acceso: 2016].
- [8] FreeBSD, «Capítulo 5. El sistema X Window,» 2016. [En línea]. Available: <https://www.freebsd.org/doc/es/books/handbook/x11.html>.
- [9] Hewlett Packard Enterprise, «HPE ProLiant ML350 Gen9 Server,» [En línea]. Available: <https://www.hpe.com/us/en/product-catalog/servers/proliant-servers/pip.hpe-proliant-ml350-gen9-server.7271259.html> .
- [10] J. Pedro, «Sistema Heterogéneos,» 2014. [En línea]. Available: <http://sistemas-eterogeneos.blogspot.com.co>. [Último acceso: 9 2016].
- [11] Kitware, «Visualization toolkit (VTK),» Kitware Inc., [En línea]. Available: <http://www.vtk.org/> .
- [12] L. M. Moreno, Computacion paralela y entornos heterogeneos, Tenerife: Soportes Audiovisuales e Informaticos, 2004.
- [13] Microsoft, «Virtualización,» [En línea]. Available: <https://www.microsoft.com/spain/virtualizacion/products/application/default.aspx>.
- [14] N. Sundown, J. Byungil y R. Luc, «Remote Visualisation of Large Scale Data for Ultra-High Resolution Display Environments,» Electronic Visalization Laboratory, [En línea]. Available: https://www.evl.uic.edu/documents/sc09_ultravis_fullpaper_nam.pdf.
- [15] N. Toole, «Visualisation and Virtual Reality,» Open Chanel Foundation, Marzo 2004. [En línea]. Available: http://www.openchannelfoundation.org/discipline/Visualization_and_Virtual_Reality/ . [Último acceso: Julio 2016].
- [16] Nvidia, «Nvidia Docker,» 11 2015. [En línea]. Available: <https://github.com/NVIDIA/nvidia-docker>. [Último acceso: 2016].
- [17] Nvidia, «Nvidia Grid,» [En línea]. Available: <http://www.nvidia.es/object/nvidia-grid-es.html>.
- [18] Nvidia, «VMware Horizon 6,» [En línea]. Available: <http://www.nvidia.es/object/vmware-horizon-6-es.html> .
- [19] O. A. Mejia, «Computación en la nube,» Universidad Autónoma de México, [En línea]. Available: <http://www.izt.uam.mx/newpage/contactos/anterior/n80ne/nube.pdf>.

- [20] P. K. S. Roldán, «Animación y Visualización de Fenómenos Naturales,» Universidad de las Américas Puebla, 13 Enero 2003. [En línea]. Available: http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/suarez_r_pk/capitulo4.pdf. [Último acceso: 9 2016].
- [21] Paraview, «ParaView Guide,» Kitware Inc., [En línea]. Available: <http://www.paraview.org/documentation/>.
- [22] Portainer, «Portanier,» [En línea]. Available: <https://portainer.readthedocs.io/en/stable/>. [Último acceso: 2016].
- [23] Portainer.io, «Portainer Documentation,» 2016. [En línea]. Available: <https://portainer.readthedocs.io/en/stable/index.html>. [Último acceso: 2017].
- [24] VMware, «Horizon Suite,» VMware, 23 10 2016. [En línea]. Available: <http://www.vmware.com/products/horizon.html>. [Último acceso: 2016].
- [25] VMware, «vGPU Deployment Guide Citrix on vSphere,» 9 2016. [En línea]. Available: http://images.nvidia.com/content/pdf/grid/guides/vGPU_Deployment_Guide_Citrix_on_vSphere_Tech_Pub_v01_final.pdf. [Último acceso: 2016].
- [26] VMware, «Virtualizacion,» 21 5 2014. [En línea]. Available: <http://www.vmware.com/latam/solutions/virtualization.html>. [Último acceso: 21 10 2016].