Universidad Autónoma Metropolitana
Unidad Azcapotzalco
División de Ciencias Básicas e Ingeniería

Licenciatura en Ingeniería en Computación

Reporte final del Proyecto de Integración

Aplicación software para administrar y monitorizar un clúster de alta disponibilidad con Servidores Linux

asociado al Proyecto de Investigación

Sistema de Computo Altamente Disponible Clave: EL001-16

David Contreras Tovar
206200359
al206200359@alumnos.azc.uam.mx

Asesor:
M. en C. José Ignacio Vega Luna
Titular C, Departamento de Electrónica
vlji@correo.azc.uam.mx

Trimestre 2018 Invierno - 11 de abril de 2018

Yo, M. en C. José Ignacio Vega Luna, *declaro que aprobé el contenido del presente Reporte de Proyecto de Integración y doy mi autorización para su publicación en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.*

Yo, David Contreras Tovar, *doy mi autorización a la Coordinación de Información de la Universidad Autónoma Metropolitana, Unidad Azcapotzalco, para publicar el presente documento en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.*

Resumen

En la sociedad actual, que valora y estima como una necesidad básica el acceso a la información de forma continua e inmediata, es vital para el mundo empresarial como para el mundo académico poder tener acceso a datos y aplicaciones de sus sistemas de cómputo en todo momento. Un clúster de alta disponibilidad permite alcanzar este objetivo. Ante cualquier fallo de hardware o software de un nodo, los datos persisten y las aplicaciones continúan su ejecución en los nodos restantes del clúster. El acceso a las aplicaciones críticas fue interrumpido brevemente y los usuarios pueden continuar sus actividades.

El siguiente proyecto de integración, en la modalidad de proyecto de investigación, presenta una estrategia para implementar un clúster de alta disponibilidad con servidores Linux y su aplicación de monitorización, enfocada a la reducción de costos en comparación con otras soluciones de este tipo ya existentes.

Índice de contenido

Índice de figuras

Índice de tablas

1. Introducción

Un clúster de computadoras se define como un sistema de procesamiento paralelo o distribuido. Consta de un conjunto de computadoras independientes, interconectadas entre sí, de tal manera que funcionan como un solo recurso computacional. A cada uno de los elementos del clúster se le conoce como nodo. Los nodos pueden estar contenidos e interconectados en un solo gabinete, o, como en muchos casos, acoplados a través de una red de área local. Otro componente básico de un clúster es la interfaz de red, la cual es responsable de transmitir y recibir los paquetes de datos que viajan a través de la red entre los nodos. El lograr que todos estos elementos funcionen como un solo sistema, es la meta a la que se quiere llegar para dar origen a un clúster. ¿Cuáles son las ventajas que ofrecen estos sistemas? La respuesta inmediata es: alto rendimiento, escalabilidad, capacidad de alta carga de trabajo y alta disponibilidad [1].

Un centro de procesamiento de datos (CPD), es una instalación que concentra recursos y equipos necesarios para el procesamiento y almacenamiento de información de empresas e instituciones, así como equipos de comunicaciones para acceder tanto local como remotamente dicha información. Con la rápida evolución de la Internet y la necesidad de tener disponible el acceso a sus sistemas cómputo en todo momento, las empresas se ven obligadas a contar con un alto nivel de confiabilidad y seguridad en su operación, ubicando sus equipos de cómputo, telecomunicaciones y de almacenamiento en un CPD para garantizar la continuidad de servicio a clientes, empleados, ciudadanos, proveedores y empresas colaboradoras [2].

Es vital que un Centro de Procesamiento de Datos (CPD) cuente con mecanismos seguros y eficientes para su funcionamiento óptimo, ya que contiene información con datos críticos y necesarios para las operaciones diarias de las empresas a fin de evitar poner en riesgo la productividad y el negocio mismo [3] – [4]. Periódicamente los centros de datos son auditados por organismos y empresas externas, para poder estar certificados y ofrecer servicios garantizados a sus clientes. Un punto importante que consideran estas auditorías es el monitoreo y control de variables ambientales y la disponibilidad de servicios al usuario o cliente. A pesar de ubicar los equipos de cómputo en un CPD, la operación del mismo y el acceso a la información y aplicaciones puede verse afectado por diferentes factores, por ejemplo: fenómenos naturales, como inundaciones, huracanes y sismos; fuego, falla en el suministro de energía eléctrica, terrorismo, fallas de hardware y software, errores humanos o asuntos legales [5] – [6]. Ante esto, lo importante es contar con una estrategia para la protección de datos, del hardware y del software crítico para el restablecimiento y continuidad de las operaciones del negocio, para responder lo más rápidamente posible a la interrupción de los servicios y continuar trabajando con las aplicaciones importantes o críticas. Estas aplicaciones, que son cada vez más orientadas a la multimedia, comercio electrónico y dispositivos móviles, demandan mayor capacidad de información, mayor velocidad de procesamiento, mejores tiempos de respuesta y mayor tiempo de disponibilidad.

Las empresas deban contar con esquemas de protección y recuperación que van desde los más básicos, que consisten en eliminar los llamados puntos de falla simples (*Single Point of Failure* – SPOF) [7] – [9], hasta implantar planes de recuperación de desastres (*Disaster Recovery Plan* – DRP), en los que se dispone de CPD redundantes y alternos [10] – [17]. Un SPOF es cualquier elemento hardware o software de un sistema de cómputo cuya falla cause que las aplicaciones de usuarios no se encuentren disponibles durante un cierto tiempo al cual se le llama *downtime*.

Un esquema de protección y recuperación inicia con tener redundancia en los servidores de cómputo y puertos de acceso a la red de datos. Para eliminar el SPOF que representan los servidores se instala y configura un clúster de computadoras con acceso compartido a la información y aplicaciones. Bajo este esquema, las aplicaciones se podrán ejecutar en cualquier computadora del clúster. Si alguna computadora del clúster falla, las aplicaciones que se encontraban ejecutándose en ella arrancarán en las computadoras restantes del clúster, con el objetivo de seguir prestando el servicio al usuario. Esto constituye un sistema de cómputo altamente disponible donde la falla de uno de sus componentes interrumpe el funcionamiento del sistema solo por un periodo de tiempo breve.

En este proyecto de integración se realizó un clúster altamente disponible y su aplicación software de monitorización con tres funcionalidades: monitorizar puertos de red, monitorizar la aplicación crítica y una interfaz gráfica. La utilidad y funcionalidades del proyecto surgen de necesidades reales y urgentes, ya que podrá usarse en un mercado, que en su mayoría son las pequeñas y medianas empresas, que necesiten prestar un servicio de manera continua. El clúster será útil tanto en casos de contingencia, cuando una computadora del mismo falle, como en eventos planeados, cuando se necesite liberar una computadora para realizar alguna actividad de mantenimiento preventivo o correctivo al hardware o software y mover la aplicación a la otra computadora del clúster.

2. Antecedentes

2.1. Proyectos de Integración o Terminales

- Clúster de alta disponibilidad bajo sistema operativo Linux [18].

El sistema propuesto en este proyecto terminal consiste en dos nodos, los cuales cooperan para contar con alta disponibilidad de las aplicaciones que estén corriendo en cada nodo, principalmente un Servidor Web. La implementación del clúster a nivel de hardware es muy semejante a la solución presentada en este proyecto. La diferencia es que el clúster implantado se administra y monitoriza mediante el uso de un software propietario de *Hewlett Packard Enterprise* (HPE) y no por una aplicación propia, como en este caso.

- Clúster para Alta Disponibilidad con Microcontroladores [19].

Este proyecto y el aquí propuesto tienen objetivos similares. Son diferentes por que los dispositivos tecnológicos utilizados para implementar el clúster varían (servidores

Linux y microcontroladores); además, las aplicaciones de los clústeres son completamente distintas.

2.2. Tesis

- Instalación y configuración de un clúster de alta disponibilidad con reparto de carga. Servidor Web y Máquinas Virtuales [20].

El objetivo de este proyecto consiste en la instalación y configuración de un clúster de computadoras, el cual aloja un servidor web de Alta Disponibilidad y Reparto de Carga. También, se ofrece a los usuarios un servicio de máquinas virtuales. El objetivo fundamental es parecido; sin embargo, el proyecto proporciona características (reparto de carga) y servicios (máquinas virtuales) adicionales en comparación con este proyecto. Además, existen diferencias en el diseño e implementación del clúster, como son el número de nodos conectados, el sistema operativo y el software de configuración y administración utilizados.

- Clúster Alta Disponibilidad sobre plataforma GNU/Linux (VERITAS) [21].

En este proyecto se presentan soluciones sobre como poder montar una arquitectura de software que incluye: aplicación, servidor de aplicaciones (jboss), balanceador (apache), base de datos (MySQL) y sistema de archivos compartido, sobre un clúster de alta disponibilidad en plataforma GNU/LINUX. Las soluciones presentadas son teóricas y el clúster no se implementó físicamente, sino virtualmente (se hace uso de Oracle VM VirtualBox), utilizando el software *Veritas Cluster Server*.

2.3. Software Comercial

- *High Availability Add-On* para Red Hat Enterprise Linux 7 [22].

El *High Availability Add-On* es un conjunto integrado de componentes de software que se puede implementar en una variedad de configuraciones para satisfacer necesidades de rendimiento, alta disponibilidad, balanceo de carga, escalabilidad, uso compartido de archivos y economía. Este software, que se adapta a la arquitectura de hardware que el cliente necesite, proporciona las soluciones que este proyecto realiza, excepto la interfaz gráfica de monitorización. Estas tareas se ejecutan a través de comandos.

- *HPE Serviceguard* for Linux A.12.10.00 [23].

HPE Serviceguard para Linux es un software de *clustering* de alta disponibilidad diseñado para proteger las aplicaciones y servicios de un tiempo de inactividad (*downtime*) planificado y no planificado. Serviceguard para Linux (SGLX) engloba todas las características de software de este proyecto, incluyendo la interfaz de monitorización, y más. La diferencia radica en que es un software de costo muy elevado, restringido a ciertas plataformas de servidores y dispositivos de almacenamiento compatibles, generalmente productos de HPE.

3. Justificación

Las pequeñas y medianas empresas tienen servidores de capacidad menor que generalmente usan el sistema operativo Linux. Este sistema operativo tiene características como robustez, confiabilidad y estabilidad, y cuenta con el soporte de los fabricantes de servidores, teniendo gran aceptación en la industria porque es de código abierto y es mucho más barato que otros sistemas operativos; inclusive algunas distribuciones son gratuitas. Los usuarios de Linux tienen necesidades de alta disponibilidad con funcionalidades similares a los grandes corporativos; también cuentan con aplicaciones de misión crítica. Una de estas necesidades es contar con un clúster de alta disponibilidad.

Actualmente existen productos de software de diferentes proveedores para configurar e implementar un clúster con servidores Linux. Su precio y servicios de consultoría son muy elevados, llegando a veces a ser más caros que el mismo sistema operativo. Estos productos solo permiten implantar el clúster, de manera tal que, para tener una aplicación altamente disponible, el usuario debe adquirir otro producto llamado *toolkit*, que solo sirve para una aplicación específica y pagar por la consultoría para instalarlo y configurarlo.

Las soluciones actuales son costosas y algunas requieren UNIX como sistema operativo. Siendo UNIX generalmente un software propietario, se debe pagar una licencia para utilizarlo, generando un costo adicional. Además, la mayoría de estas soluciones funcionan a través de línea de comandos, sin proporcionar una interfaz gráfica amigable con el usuario. Todo esto hace a las soluciones actuales prácticamente inaccesibles para pequeñas y medianas empresas, así como instituciones y organizaciones públicas, a las cuales va dirigida la solución desarrollada en este proyecto.

4. Objetivos

4.1. Objetivo general

Diseñar e implementar una interfaz gráfica para monitorizar un clúster de servidores Linux, con el fin de obtener alta disponibilidad de una aplicación crítica.

4.2. Objetivos específicos

- Diseñar e implementar el módulo gráfico, que despliega la información del estado del clúster en pantalla.
- Diseñar e implementar el módulo de comunicación y estado, que establece una conexión entre la interfaz gráfica y el clúster, de manera remota; y determina el estado del clúster.

5. Marco teórico

5.1. Alta Disponibilidad

Se entiende como disponibilidad de un sistema al tiempo durante el cual dicho sistema trabaja de forma normal y puede brindar servicio a sus usuarios [24].

Para medir el nivel de disponibilidad de un sistema se utilizan los siguientes parámetros:

5.1.1. Disponibilidad [24]

Se expresa la disponibilidad como un porcentaje, que permite determinar el tiempo en el que el sistema funcionará correctamente, durante un período de tiempo determinado. En la Tabla 1 se presentan algunos valores de disponibilidad y su equivalente en horas de funcionamiento.

Disponibilidad	Porcentaje de caída	Tiempo de caída por año	Tiempo de caída por semana
98%	2%	7,3 días	3,3 horas
99%	1%	3,65 días	1,6 horas
99,8%	0,2%	17,5 horas	20 min
99,9%	0,1%	8,7 horas	10 min
99,99%	0,01%	52,5 min	1 min
99,999%	0,001%	5,2 min	6 segundos

Tabla 1. Valores de disponibilidad y su equivalencia en tiempo.

A medida que aumenta el porcentaje de disponibilidad lo hace también el costo que el sistema tendrá.

Para muchas aplicaciones un porcentaje de disponibilidad del 99% es adecuado, es decir, existen sistemas que pueden tolerar una caída de aproximadamente dos horas cada semana, lo que depende de a qué hora del día suceda esa caída. Si la caída sucede en la madrugada de un domingo no genera los mismos inconvenientes que si sucede en un día hábil a la hora en que todos los usuarios se conectan.

5.1.2. Modelo de los nueves [24]

El expresar la disponibilidad usando porcentajes, como 99,99%, se conoce como el modelo de los nueves y deberá utilizarse solo para propósitos teóricos, ya que no es posible modelar todos los componentes o sistemas involucrados en la entrega de un recurso o servicio.

5.1.3. Tiempo de caída [24]

Es el tiempo durante el cual el usuario no puede acceder al servicio que el sistema brinda, debido a un fallo del sistema o uno de sus componentes.

5.1.4. Cálculo de disponibilidad [24]

Para determinar la disponibilidad de un sistema se utiliza la ecuación:

$$A = \left[\frac{MTBF}{(MTBF + MTTR)}\right] * 100$$

Donde ***A*** es el grado de disponibilidad expresada en porcentaje, ***MTBF*** *(Mean Time Between Failures)* es el tiempo medio entre fallas, y ***MTTR*** (*Maximum Time To Restore)* es el tiempo máximo que tomaría reparar o resolver un error en particular.

Por ejemplo, si se tiene un sistema que tiene un tiempo medio entre fallas igual a 50.000 horas y un tiempo de reparación promedio de 5 horas, la disponibilidad del sistema será 99,99%, mejorar ese porcentaje a 99,998% implicaría disminuir el tiempo de caída a menos de 10 minutos al año.

5.1.5. Alta disponibilidad [24]

Que un sistema esté disponible significa que el usuario puede acceder a los servicios que el sistema brinda, dentro del período de tiempo que se supone que el sistema está en funcionamiento.

Alta disponibilidad implica maximizar el tiempo de funcionamiento del sistema, es decir, disminuir el tiempo de caída y el tiempo de recuperación, mediante técnicas de software, redundancia de hardware, redundancia a nivel de red, planes de recuperación ante desastres, etc.

Un sistema de cómputo con alta disponibilidad es aquel que ha sido diseñado para minimizar los tiempos de caída en los servicios que el sistema brinda mediante la reducción de puntos de falla y el control de los posibles errores que puedan presentarse.

5.2. ¿Qué es un clúster?

Es un conjunto de computadoras construidas mediante la utilización de componentes de hardware que se comportan como si fuesen una única computadora [25].

La tecnología de *clustering* consiste en que dos o más computadoras trabajen de manera coordinada para ofrecer escalabilidad, alta disponibilidad o alto desempeño, y en caso de fallas la carga de trabajo pueda distribuirse entre las computadoras que forman parte del clúster [24].

La tecnología de clúster ha evolucionado gracias al apoyo de actividades que van desde aplicaciones de supercómputo, software de misiones críticas, servidores web y comercio electrónico, hasta bases de datos de alto rendimiento, entre otros usos.

El cómputo con clúster surge como resultado de la convergencia de varias tendencias actuales. Incluye disponibilidad de microprocesadores económicos de alto rendimiento y redes de alta velocidad, desarrollo de herramientas de software para cómputo distribuido de alto rendimiento y la creciente necesidad de potencia computacional para aplicaciones que la requieran [25].

Las simulaciones en computadora son vitales para el estudio de problemas que van desde el diseño en ingeniería hasta el estudio de procesos complejos en la naturaleza. Sin embargo, el alcance y la precisión de estas simulaciones están limitados por la potencia computacional de las supercomputadoras más potentes.

La historia de los clústeres computacionales en Linux comenzó cuando Donald Becker y Thomas Sterling construyeron un clúster para la NASA cuyo nombre fue Beowulf [25]. El modelo de clúster tipo Beowulf se basa en componentes y periféricos para la plataforma x86 común para obtener un rendimiento sin precedentes a un costo muy bajo. A partir de este proyecto, han surgido numerosas iniciativas en este sentido.

Estos clústeres se utilizan para cualquier tarea que requiera enormes cantidades de cómputo: *data mining*, simulaciones científicas, renderización de gráficos, modelado meteorológico, etc.

5.2.1. Clasificación de los clústeres [25]

Los clústeres pueden clasificarse con base en sus características. Hay clústeres de alto rendimiento o *High Performance* (HP), clústeres de alta disponibilidad o *High Availability* (HA), y clústeres de alta eficiencia o *High Throughput* (HT).

High Performance: Son clústeres en los cuales se ejecutan tareas que requieren una gran capacidad computacional, cantidades enormes de memoria o ambas a la vez. Llevar a cabo estas tareas puede comprometer los recursos del clúster por largos periodos de tiempo.

High Availability: Son clústeres cuyo objetivo es proveer disponibilidad y confiabilidad. Estos clústeres tratan de brindar la máxima disponibilidad de los servicios que ofrecen. La confiabilidad se provee mediante un software que detecta fallos y permite recuperarse frente a ellos, mientras que en hardware se evita tener un único punto de fallo (SPOF).

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

5.2.2. ¿Cómo funciona un clúster? [25]

Desde un punto de vista general, un clúster consta de dos partes. La primera es el software, un sistema operativo confeccionado especialmente para esta tarea (por ejemplo, un Kernel Linux modificado). Luego se tienen compiladores y aplicaciones especiales que permiten que los programas que se ejecuten en el sistema utilicen todas las ventajas del clúster. En el entorno de GNU/Linux hay que destacar la PVM

(*Paralell Virtual Machine*) y la MPI (*Message Passing* Interface), librerías que abstraen el componente hardware del componente software.

El segundo componente es la interconexión hardware entre las máquinas (nodos) del clúster. Se han desarrollado interfaces de interconexión especiales muy eficientes; sin embargo, es común realizar las interconexiones mediante una red Ethernet dedicada de alta velocidad. Gracias a esta red de interconexión los nodos del clúster intercambian entre sí las tareas, las actualizaciones de estado y los datos del programa. En un clúster abierto, existirá una interfaz de red que conecte al clúster con el mundo exterior (Internet). Cuando se trata de resolver un problema en paralelo, el software debe ser capaz de dividirlo en tareas más pequeñas, repartirlas entre los nodos y elaborar los resultados. Puesto que las sub-tareas van a ejecutarse en paralelo se consigue un aumento de velocidad, aunque hay que tener en cuenta el retardo en la división, el reparto y la transmisión de mensajes (resultado, coherencia y estados).

En el caso de los clústeres de balanceo de carga, el hardware y el software deben actuar conjuntamente para que el tráfico se distribuya entre los nodos del clúster. De esta forma, se pueden ofrecer los servicios a mayor velocidad o se realiza una tarea más rápidamente.

Los servidores de un clúster de alta disponibilidad normalmente no comparten la carga de procesamiento que tiene un clúster de alto rendimiento; tampoco comparten la carga de tráfico, como lo hace un clúster de balanceo de carga. Su función es la de estar preparados para entrar inmediatamente en funcionamiento, en caso de que falle algún otro servidor.

5.2.3. Puntos a considerar a la hora de configurar un clúster [25]

Por sus características especiales, hay varias cuestiones particulares asociadas a esta tecnología que deben tomarse en cuenta.

Uno de los principales problemas a los que hay que hacerle frente cuando se construye un clúster es buscar y eliminar los puntos de fallo únicos (SPOF). Cuando se trabaja en un clúster de supercomputación que depende de un servidor central para repartir las tareas, y este servidor cae, todo el clúster quedará inservible. Igualmente, si se trata de un clúster de balanceo de carga o de alta disponibilidad, se deben establecer garantías de que los servidores seguirán funcionando; pero si estos servidores están conectados a una red corporativa o a internet mediante una sola interfaz, un fallo en ella dejaría aislado al sistema. Es importante perseguir la redundancia para evitar que el fallo de un solo componente hardware (recordemos que en un clúster van a integrarse gran número de elementos con lo que la probabilidad de fallo crece) anule la funcionalidad de todo el sistema.

Otra cuestión importante es elegir correctamente la tecnología que se utilizará en función de nuestras necesidades. Mantener un clúster sobre una red Ethernet de 10 Mbps, puede resultar una buena decisión si el clúster sólo tiene unos cuantos nodos; pero en el momento en que se inserten más nodos, la red se convertirá en un cuello de botella que obligaría a los servidores a estar desocupados en espera de los datos durante demasiado tiempo.

5.2.4. Clúster de alta disponibilidad [24]

Un clúster de alta disponibilidad es un conjunto de nodos que se comunican entre sí y trabajan con un objetivo común que consiste en ofrecer acceso ininterrumpido a datos o servicios.

Si uno de los servidores que forma parte del clúster pierde conexión con la red o experimenta un fallo de hardware, e incluso si el error se presenta en la aplicación que brinda el servicio, el clúster debe ser capaz de continuar con su funcionamiento con un tiempo de caída mínimo.

Un clúster de alta disponibilidad básico puede verse en la Figura 1; el mismo consta de dos nodos o servidores que comparten un almacenamiento común, en donde se guardan los datos que los clientes necesitan.

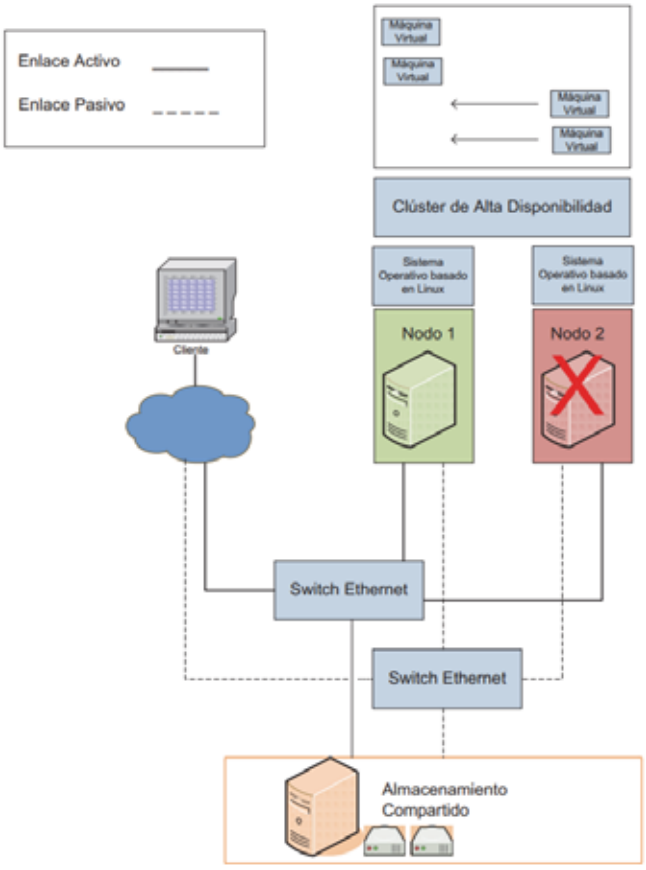


Figura 1. Clúster de alta disponibilidad básico.

Si uno de los nodos no pudiera acceder a los datos, el otro tomaría su lugar respondiendo a las peticiones de los clientes.

5.3. Elementos que forman parte de un clúster

Los elementos que conforman un clúster son [25]:

- ✓ Un nodo activo, donde corren los servicios.
- ✓ Un nodo pasivo que funciona como respaldo (*backup*).
- ✓ Servidores reales.
- ✓ Software de administración.
- ✓ Protocolos de comunicación y servicios.
- ✓ Conexiones de red.
- ✓ Ambientes de programación paralela.
- ✓ Middleware.

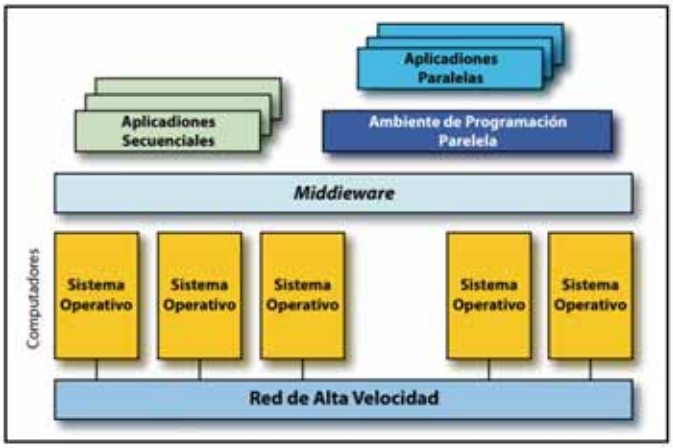


Figura 2. Elementos de un clúster.

5.3.1. Componentes lógicos de un clúster de alta disponibilidad [24]

A nivel lógico un clúster de alta disponibilidad puede dividirse en 4 capas:

1. Almacenamiento compartido del clúster.
2. Software de administración del clúster.
3. Software de administración de recursos del clúster.
4. Agente de recursos.

En la Figura 3 se presenta como ejemplo el modelo de cuatro capas que implementa la solución de alta disponibilidad de *SUSE Linux Enterprise Server* (SLES), en la cual se puede apreciar las herramientas que se emplean en cada una de las capas.

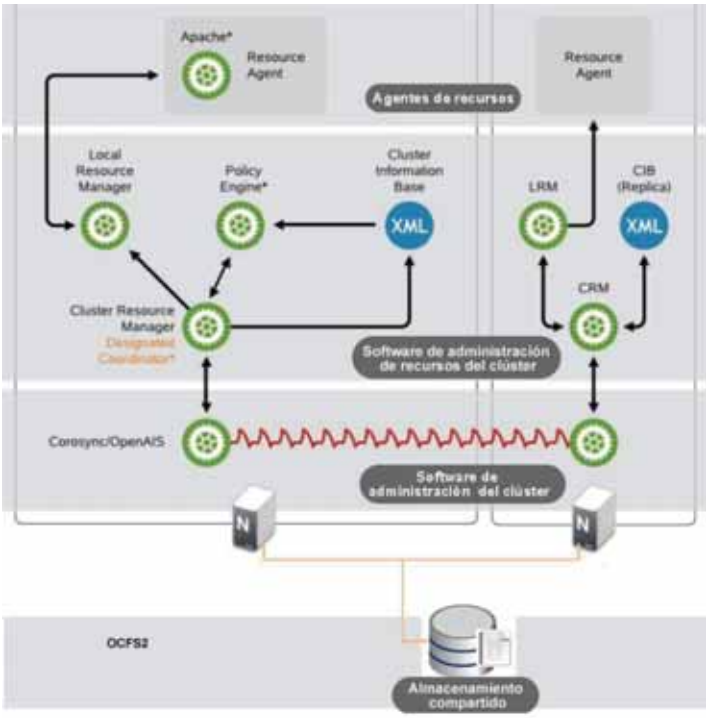


Figura 3. Componentes lógicos de un clúster de alta disponibilidad.

5.3.1.1. Almacenamiento compartido [24]

En un clúster de alta disponibilidad es necesario que cada nodo tenga acceso a un almacenamiento común. En caso de falla de un nodo que esté ejecutando alguna aplicación, está arrancará en otro nodo y tendrá acceso a los mismos datos.

Así, por ejemplo, en un clúster con tres nodos denominados A, B, C, si falla el nodo A en el que se ejecutaba la aplicación MySQL, la aplicación puede levantarse en el nodo B, siempre y cuando los archivos que forman la base de datos se encuentren en el almacenamiento compartido.

Existen dos tipos de almacenamiento que se emplean en un clúster:

- *Single – Instance Storage (SIS)* [24]. En este tipo de almacenamiento el clúster almacena todos sus datos en una instancia centralizada, como por ejemplo una SAN *(Storage Area Network)*.

  El acceso al almacenamiento puede ser activo/pasivo, es decir un nodo a la vez o activo/activo en donde múltiples nodos acceden simultáneamente.

  Si se emplea la configuración activa/activa es necesario utilizar un sistema de archivos del tipo GFS2 *(Global File System 2)* u OCFS2 *(Oracle Cluster File System 2)*.

  Cualquiera sea el método de acceso empleado en SIS, es necesario que exista un mecanismo para coordinar el acceso de los nodos al almacenamiento compartido; Linux emplea DLM *(Distributed Lock Manager)* a nivel de kernel para realizar esa tarea.

La implementación de SIS es sencilla, pero tiene un grave inconveniente; si por algún motivo la SAN falla o se vuelve inaccesible, se perderán los datos que necesitan las aplicaciones para su ejecución en el clúster, y no estarán disponibles para el usuario.

En la Figura 4 se presenta un clúster con almacenamiento del tipo SIS; si por algún motivo los enlaces o el conmutador fallaran, de nada serviría tener redundancia en los servidores físicos, ya que estos no podrían acceder a los datos que se encuentran almacenados en la SAN, con lo que se perdería la alta disponibilidad.

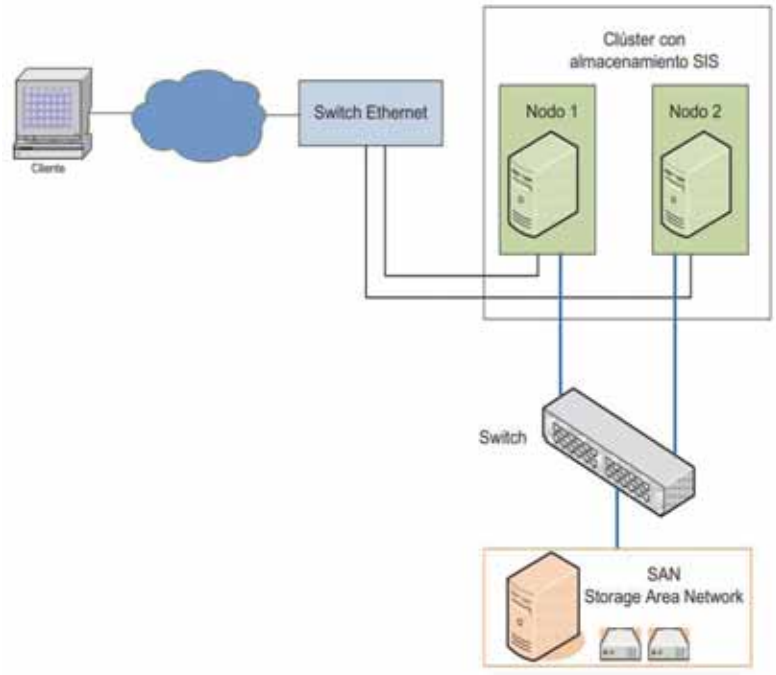


Figura 4. Clúster con almacenamiento del tipo SIS.

- *Almacenamiento Replicado* [24]. Permite mediante hardware o software que los datos escritos en un almacenamiento primario se repliquen de manera síncrona (escribe al mismo tiempo) o asíncrona (escribe primero en el almacenamiento local) en un almacenamiento secundario.

  Existen varias opciones para implementar la replicación de datos vía software. La forma estándar en Linux se denomina DRBD *(Distributed Replicated Block Device)*.

  DRBD consiste de un módulo kernel y aplicaciones para configurar y administrar un dispositivo de bloque denominado `/dev/drbd0`, cuya información se sincroniza entre dos servidores vía red.

  En la Figura 5 se indica el esquema de un clúster con almacenamiento replicado del tipo DRBD; si el servidor DRBD principal falla, el servidor secundario toma su lugar, y los nodos del clúster no pierden conectividad con el almacenamiento compartido.

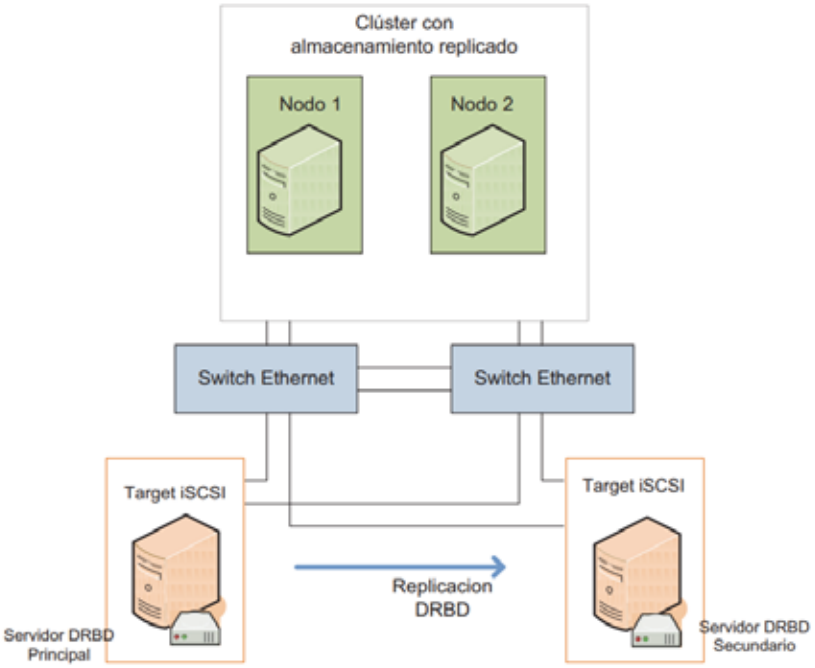


Figura 5. Clúster de alta disponibilidad con almacenamiento replicado.

Las soluciones de replicación mediante hardware dependen de la empresa que suministre el almacenamiento; por ejemplo, IBM ofrece la solución PPRC *(Peer to Peer Remote Copy)* que funciona solamente con los sistemas de almacenamiento de IBM, como *IBM System Storage DS8000* o *IBM System Storewize*.

Por lo general la replicación por hardware tiene costos bastante elevados.

5.3.1.2. Software de administración [24]

Permite que un grupo de computadoras o nodos funcionen como un clúster. Las tareas del software de administración son enviar mensajes entre los nodos que son parte del clúster, determinar si el número de nodos de los que dispone el clúster es suficiente para funcionar *(quorum)* y determinar si un nodo pertenece o no al clúster *(membership)*.

En los clústeres de HA basados en Linux se utiliza *Heartbeat* (demonio que implementa comunicaciones y pertenencia en un clúster Linux) para realizar estas tareas. Sin embargo, en los últimos años en la mayoría de las distribuciones se emplea *Corosync*, el cual es un software que implementa el protocolo *Totem* de membresía y envío de mensajes ordenados. *Corosync* además cifra los mensajes y autentica a los nodos del clúster, lo que permite el envío de mensajes de forma segura y ordenada utilizando UDP/IP sobre Ethernet.

Al software de administración del clúster se le conoce también como *cluster stack*.

5.3.1.3. Software de administración de recursos [24]

Como su nombre lo indica es el software encargado de activar, detener y monitorizar el estado de los servicios que un clúster ejecuta. En base al estado de los nodos del clúster,

y políticas y reglas configuradas previamente, el software de administración de recursos toma decisiones para garantizar el funcionamiento continuo de los servicios.

El administrador de recursos por defecto en Linux es *Pacemaker*, aunque distribuciones como *Red Hat Enterprise Linux* (RHEL) y CentOS pueden utilizar el administrador *RGManager*.

5.3.1.4. Agente de recursos [24]

Es un *script* o programa que actúa como interfaz entre el recurso y el administrador de recursos.

Pueden escribirse en cualquier lenguaje de programación, pero lo común es que sean *scripts* que se ejecutan en la consola de Linux, y que permitan realizar operaciones como *start, stop, status, validate-all* sobre un recurso, servicio o programa:

- *Start*: Esta operación inicia el servicio.
- *Stop*: Detiene el servicio o programa.
- *Status*: Retorna el estado del servicio.
- *Validate-all*: Valida los parámetros con los que se configuró el recurso.

El administrador de recursos *Pacemaker* es compatible con varias clases de agentes:

- *LSB Resource Agents* (Linux *Standard Base*): Son los *scripts* que se encuentran en la carpeta `/etc/init.d/`; se crean por defecto al instalar un programa que funciona como un servicio, tal como Apache, SSH, MySQL, etc. Varían dependiendo de la versión de Linux.
- *OCF Resource Agents* (Oracle *Cluster Framework*): Son *scripts* basados en LSB que tienen características adicionales.
- Systemd son los *scripts* de inicio que utiliza el programa `systemd`, el cual es el encargado de arrancar un programa como un servicio del sistema, por ejemplo: programas de correo, firewall, etc.

5.3.2. Componentes físicos de un clúster de alta disponibilidad

5.3.2.1. Nodos [24]

Deben cumplir ciertos requisitos de RAM y procesador; en particular para el caso de la virtualización, el procesador debe soportarla.

A fin de tener redundancia a nivel de red se recomienda que los nodos cuenten con al menos dos tarjetas Ethernet, así el nodo podrá continuar comunicándose con el resto del clúster si una de ellas falla.

Si para el almacenamiento compartido se emplea una SAN los nodos deben tener tarjetas de conexión de fibra, o Gigabit Ethernet si el almacenamiento compartido es mediante un servidor iSCSI *(Internet Small Computer System Interface)*.

5.3.2.2. Red de comunicaciones del clúster [24]

Es recomendable implementar redundancia en la red de comunicaciones ya sea mediante el enlazado de interfaces o *network bonding*, o definiendo en el software de administrador del clúster un canal de comunicación de respaldo. Sin embargo, la misma red que se utiliza para las comunicaciones del clúster puede emplearse para las comunicaciones con los clientes (red pública) e incluso para las comunicaciones con el almacenamiento compartido.

5.3.2.3. Red pública [24]

Es la red a la que los clientes se conectan para requerir los servicios de alta disponibilidad que el clúster brinda.

5.3.2.4. Sistema de almacenamiento compartido [24]

Se refiere a la capa física sobre la que se implementa la capa de almacenamiento lógico, puede tratarse de un servidor iSCSI, un almacenamiento SAN con conexión de fibra o inclusive un clúster iSCSI con alta disponibilidad.

5.4. Ventajas y desventajas de los clústeres [25]

5.4.1. Ventajas

- Disponibilidad: Capacidad para continuar operando ante la caída de alguno de los ordenadores del clúster.
- Distribución en paralelo.
- Flexibilidad: Los balanceadores de carga no están amarrados a ninguna arquitectura específica, en lo que respecta a hardware.
- Escalabilidad: Capacidad para hacer frente a volúmenes de trabajo cada vez mayores, prestando así un nivel de rendimiento óptimo.
- Expansibilidad: Capacidad de aumentar sus capacidades a través de mejores técnicas.
- Transferencia de información y todo tipo de servicio por internet de forma rápida, a bajo costo e ininterrumpidamente.
- Incremento de velocidad de procesamiento ofrecido por los clústeres de alto rendimiento.
- Incremento del número de transacciones o velocidad de respuesta ofrecido por los clústeres de balanceo de carga.
- Incremento de la confiabilidad y la robustez ofrecido por los clústeres de alta disponibilidad.

5.4.2. Desventajas

- Empresas y entidades prefieren seguir utilizando el modelo cliente/servidor tradicional debido al espacio físico o a nuevos problemas que no se daban en la arquitectura tradicional.
- Costo elevado.
- Espacio físico para el montaje del clúster.

6. Desarrollo del proyecto

6.1. Diseño del clúster

El clúster está compuesto por dos servidores o nodos y un arreglo de discos conectado a ellos como se muestra en la Figura 6. Cada nodo cuenta con dos puertos de red y dos puertos de acceso a disco.

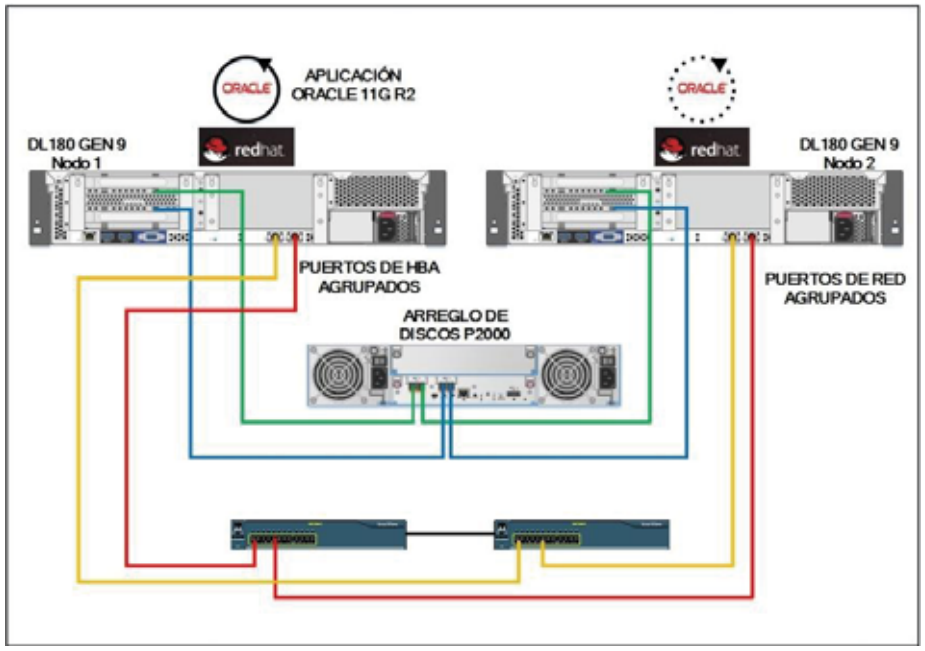


Figura 6. Arquitectura del clúster implementado.

6.2. Especificación Técnica

- Para la implementación de la interfaz gráfica se utilizó Visual Studio 2017 [26], con C# y Visual C# [27]; plataforma de desarrollo *(developer platform)* .NET [28]; y .NET Framework 4.7.1 [27].
- Requerimiento de ejecutar la interfaz gráfica en cualquier computadora con sistema operativo Windows (7 o posterior).
- La distribución de Linux utilizada como sistema operativo en los nodos del clúster es Red Hat Enterprise Linux 6.4 [29].
- Se utiliza el manejador Oracle 12c [30] como aplicación crítica, con una base de datos de 1 GB.

A continuación, se muestra la Tabla 2, donde se listan las características de los elementos que conforman el clúster implementado:

Equipo	Características
Dos servidores HP modelo Proliant DL180 Gen9.	1 Procesador Intel Xeon E5 - 2609v3, de 6 núcleos a 1.9 GHz, 8 GB de RAM, 1 controladora de disco duro HP H240 FIO de 12 Gb/s, 6 ranuras para disco duro SAS/SATA y 3 ranuras de expansión PCI – Express 3.0. Alimentación 100 – 240 VAC, 50/60 Hz., 3.3 A.
Arreglo de discos HP P2000 Modular Smart Array 3.5 – inch Drive Bay Chassis.	Dos ranuras para tarjetas controladoras y 12 ranuras para discos duros SAS/SATA. Alimentación 100 – 240 VAC, 50/60 Hz., 3 A.
Dos tarjetas de red Ethernet.	Cada tarjeta de 2 puertos de 1 Gb/s.
Dos tarjetas HPE modelo H241 Smart Host Bus Adapter.	Cada tarjeta de 2 puertos SAS de 12 Gb/s o SATA de 6 Gb/s, conexión a slot PCIe3 x 8.
Sistema operativo Red Hat Enterprise Linux (RHEL) 6.	Red Hat Enterprise Linux 6.4 x86_64, ambiente gráfico Anaconda, Firewall Lokkit, sistema de codificación UTF-8.
Software de Oracle Versión 12.	Oracle Database 12c for Linux x86_64.

Tabla 2. Elementos del clúster implementado.

6.3. Configuración de *Network Bonding* del clúster

En esta sección se presenta la lista total de *scripts* que conforman la configuración del *Network Bonding*, así como una lista detallada de los 4 *scripts* principales.

```
[root@digitales1 /]# cd /etc/sysconfig/network-scripts/
[root@digitales1 network-scripts]# ll
total 220
-rw-r--r--. 1 root root   138 Feb 21 14:23 ifcfg-bond0
-rw-r--r--. 1 root root   122 Feb 21 14:34 ifcfg-bond0.10
-rw-r--r--. 1 root root   111 Feb 16 19:32 ifcfg-eth0
-rw-r--r--. 1 root root   110 Feb 16 19:33 ifcfg-eth1
-rw-r--r--. 1 root root   254 Jan  9  2013 ifcfg-lo
lrwxrwxrwx. 1 root root    20 Oct 10  2016 ifdown -> ../../../sbin/ifdown
-rwxr-xr-x. 1 root root   627 Jan  9  2013 ifdown-bnep
-rwxr-xr-x. 1 root root  5397 Jan  9  2013 ifdown-eth
-rwxr-xr-x. 1 root root   781 Jan  9  2013 ifdown-ippp
-rwxr-xr-x. 1 root root  4168 Jan  9  2013 ifdown-ipv6
lrwxrwxrwx. 1 root root    11 Oct 10  2016 ifdown-isdn -> ifdown-ippp
-rwxr-xr-x. 1 root root  1481 Jan  9  2013 ifdown-post
-rwxr-xr-x. 1 root root  1064 Jan  9  2013 ifdown-ppp
-rwxr-xr-x. 1 root root   835 Jan  9  2013 ifdown-routes
-rwxr-xr-x. 1 root root  1370 Jan  9  2013 ifdown-sit
-rwxr-xr-x. 1 root root  1434 Jan  9  2013 ifdown-tunnel
lrwxrwxrwx. 1 root root    18 Oct 10  2016 ifup -> ../../../sbin/ifup
-rwxr-xr-x. 1 root root 12365 Jan  9  2013 ifup-aliases
-rwxr-xr-x. 1 root root   859 Jan  9  2013 ifup-bnep
-rwxr-xr-x. 1 root root 10157 Jan  9  2013 ifup-eth
-rwxr-xr-x. 1 root root 11971 Jan  9  2013 ifup-ippp
-rwxr-xr-x. 1 root root 10401 Jan  9  2013 ifup-ipv6
lrwxrwxrwx. 1 root root     9 Oct 10  2016 ifup-isdn -> ifup-ippp
```

```
-rwxr-xr-x. 1 root root   727 Jan  9  2013 ifup-plip
-rwxr-xr-x. 1 root root   954 Jan  9  2013 ifup-plusb
-rwxr-xr-x. 1 root root  2364 Jan  9  2013 ifup-post
-rwxr-xr-x. 1 root root  4154 Jan  9  2013 ifup-ppp
-rwxr-xr-x. 1 root root  1925 Jan  9  2013 ifup-routes
-rwxr-xr-x. 1 root root  3499 Jan  9  2013 ifup-sit
-rwxr-xr-x. 1 root root  2488 Jan  9  2013 ifup-tunnel
-rwxr-xr-x. 1 root root  3770 Jan  9  2013 ifup-wireless
-rwxr-xr-x. 1 root root  4623 Jan  9  2013 init.ipv6-global
-rwxr-xr-x. 1 root root  1125 Jan  9  2013 net.hotplug
-rw-r--r--. 1 root root 13079 Jan  9  2013 network-functions
-rw-r--r--. 1 root root 29853 Jan  9  2013 network-functions-ipv6
drwxr-xr-x. 2 root root  4096 Feb 21 14:15 origs
[root@digitales1 network-scripts]# pwd
/etc/sysconfig/network-scripts
[root@digitales1 network-scripts]#
[root@digitales1 network-scripts]# more ifcfg-bond0
NAME=bond0
DEVICE=bond0
BONDING_MASTER=yes
TYPE=Bond
ONBOOT=yes
BOOTPROTO=dhcp
BONDING_OPTS="mode=balance-rr miimon=100"
NM_CONTROLLED=no
[root@digitales1 network-scripts]# more ifcfg-bond0.10
DEVICE=bond0.10
NAME=bond0.10
BOOTPROTO=none
ONPARENT=yes
IPADDR=10.10.10.160
NETMASK=255.0.0.0
VLAN=yes
NM_CONTROLLED=no
[root@digitales1 network-scripts]# more ifcfg-eth0
NAME=bond0-slave0
DEVICE=eth0
TYPE=Ethernet
BOOTPROTO=none
ONBOOT=yes
MASTER=bond0
SLAVE=yes
NM_CONTROLLED=no
[root@digitales1 network-scripts]# more ifcfg-eth1
NAME=bond0-slave1
DEVICE=eth1
TYPE=Ethernet
BOOTPROTO=none
ONBOOT=yes
MASTER=bond0
SLAVE=yes
NM_CONTROLLED=no
[root@digitales1 network-scripts]#
```

6.4. Módulo gráfico

Este módulo es el encargado de visualizar en pantalla el estado del clúster. La vista general de la interfaz gráfica "Monitor de Cluster" se muestra en la Figura 7:

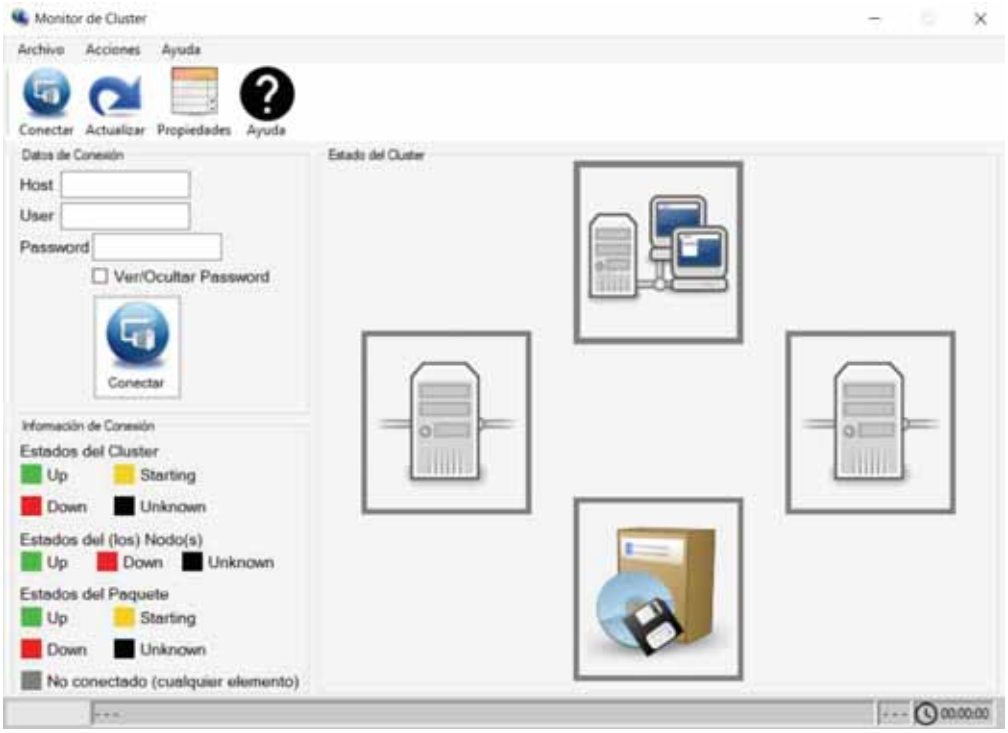


Figura 7. Vista general de la interfaz gráfica "Monitor de Cluster".

El módulo gráfico se divide en 3 secciones:

1. **Estado del Cluster**: Sección principal de la interfaz gráfica. Está conformada por cuatro íconos que representan al clúster y sus elementos; Nodo 1 (principal), Nodo 2 (respaldo o *backup*), y Paquete (aplicación crítica).
2. **Datos de conexión**: En esta sección se introducen los datos necesarios para establecer una conexión remota entre la interfaz gráfica y el clúster. Los datos de conexión solicitados son los siguientes:
    a. Host. Dirección IPv4 en la red LAN donde se ubica el clúster.
    b. User. Nombre que identifica al usuario dentro del clúster (basado en servidores Linux). Para más información, consultar [31].
    c. Password. Contraseña de acceso para el usuario respectivo.

    Además, se incluyen dos elementos adicionales:

    - Ver/Ocultar Password. *Checkbox* que permite al usuario visualizar su contraseña para verificar posibles errores.
    - Conectar. Botón que realiza la conexión remota.
3. **Información de Conexión**: Muestra al usuario de la interfaz gráfica el código de colores utilizado para informar el estado del clúster y sus elementos.

El módulo gráfico proporciona, además, barra de menú, barra de herramientas y barra de estado.

La programación de este módulo se realizó con Visual C#.

6.5. Módulo de comunicación y estado

Establecer la conexión remota entre la interfaz gráfica y el clúster, y determinar y comunicar el estado del clúster al módulo gráfico; son las tareas que ejecuta el módulo de comunicación y estado.

6.5.1. Conexión remota

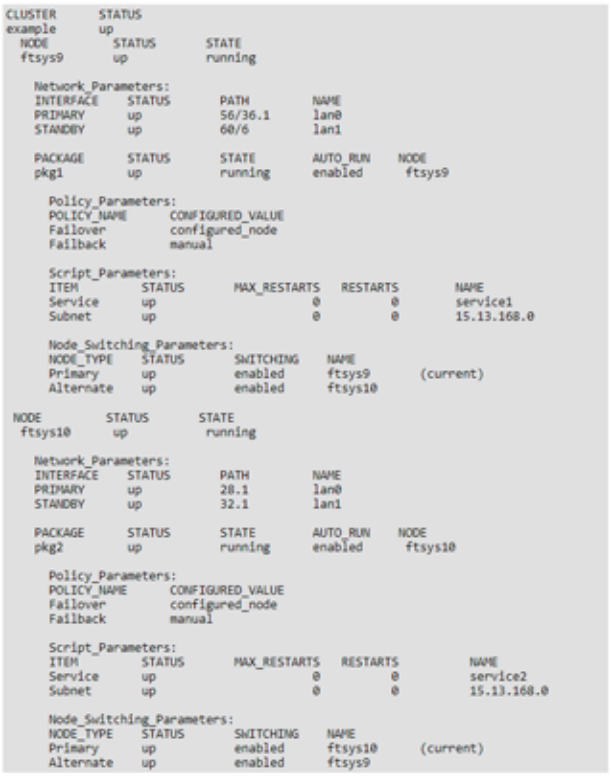
Un cliente SSH es un software que se utiliza para conectarse a un servidor SSH. El servidor puede estar en su universidad, trabajo o casa. Necesita un nombre de host (por ejemplo, **`students.example.edu`**) o una dirección IPv4 (por ejemplo, **`177.33.189.54`**) para conectarse.

La conexión remota entre la interfaz gráfica y el clúster se realiza mediante el protocolo SSH *(Secure Shell)* [32]. Dado que C# no soporta de manera nativa SSH, se utilizó un paquete NuGet [33] llamado SSH.NET [34]. Los términos de la licencia de uso se pueden consultar en [35].

6.5.2. Estado del clúster

cmviewcl [36] – [37]: Información sobre clústeres de Alta Disponibilidad

cmviewcl es un comando Linux (heredado de UNIX) que devuelve la información del estado actual de un clúster. La salida puede ser mostrada para todo el clúster o puede estar limitada a nodos o paquetes particulares. Para ver la información del clúster, un usuario debe ser superusuario (UID=0), o tener una política de acceso de MONITOR permitida en el archivo de configuración del clúster, o tener una política de PACKAGE_ADMIN en el archivo de configuración de un paquete.



```
CLUSTER      STATUS
example      up
  NODE         STATUS       STATE
  ftsys9       up           running

    Network_Parameters:
    INTERFACE    STATUS       PATH         NAME
    PRIMARY      up           56/36.1      lan0
    STANDBY      up           60/6         lan1

    PACKAGE      STATUS       STATE        AUTO_RUN     NODE
    pkg1         up           running      enabled      ftsys9

      Policy_Parameters:
      POLICY_NAME      CONFIGURED_VALUE
      Failover         configured_node
      Failback         manual

      Script_Parameters:
      ITEM         STATUS       MAX_RESTARTS   RESTARTS       NAME
      Service      up                      0          0       service1
      Subnet       up                      0          0       15.13.168.0

      Node_Switching_Parameters:
      NODE_TYPE    STATUS       SWITCHING    NAME
      Primary      up           enabled      ftsys9       (current)
      Alternate    up           enabled      ftsys10

NODE         STATUS       STATE
 ftsys10     up           running

    Network_Parameters:
    INTERFACE    STATUS       PATH         NAME
    PRIMARY      up           28.1         lan0
    STANDBY      up           32.1         lan1

    PACKAGE      STATUS       STATE        AUTO_RUN     NODE
    pkg2         up           running      enabled      ftsys10

      Policy_Parameters:
      POLICY_NAME      CONFIGURED_VALUE
      Failover         configured_node
      Failback         manual

      Script_Parameters:
      ITEM         STATUS       MAX_RESTARTS   RESTARTS         NAME
      Service      up                      0          0         service2
      Subnet       up                      0          0         15.13.168.0

      Node_Switching_Parameters:
      NODE_TYPE    STATUS       SWITCHING    NAME
      Primary      up           enabled      ftsys10      (current)
      Alternate    up           enabled      ftsys9
```

Figura 8. Ejemplo de la salida del comando *cmviewcl*.

Opciones

- -v: Se visualiza la salida Verbose.
- -c nombre_cluster: Nombre del clúster a visualizar. Si se omite esta opción, *cmviewcl* mostrará información sobre el clúster actual al que pertenece el nodo local.
- -n nombre_nodo: Ver información sólo sobre nombre_nodo, incluyendo información sobre los paquetes que se están ejecutando en este nodo.
- -p nombre_paquete: Ver información sólo sobre nombre_paquete específico.
- -l paquete|cluster|nodo|grupo: Limita la salida para mostrar sólo información específica de paquetes, clústeres, nodos o grupos.

Información de salida

- CLUSTER: Nombre del clúster. Este nombre se usará para identificar el clúster.
- STATUS: Estado del clúster, nodo, paquete, etc. El estado puede ser “up”, “down”, “starting”, “halting” o “unknown”. “unknown” es un estado especial que indica que no es posible recopilar suficiente información para determinar exactamente si el elemento está up o down, por ejemplo, cuando no se puede llegar a algunos de los nodos del clúster a través de la red.
- NODE: Nombre del nodo.

- STATE: La acción que el nodo o paquete está tomando actualmente (si existe). El estado normal es running. Sin embargo, un nodo puede estar reformándose, lo que significa que un nuevo nodo se está uniendo al cluster, o que uno o más nodos han perdido la conexión entre sí.

6.5.3. Programación

La Figura 9 muestra el diagrama de flujo del módulo de comunicación y estado. Este diagrama representa de manera gráfica el algoritmo que realiza las tareas requeridas.

La Figura 10 presenta un diagrama que visualiza los atributos y métodos de las clases desarrollas para éste módulo.

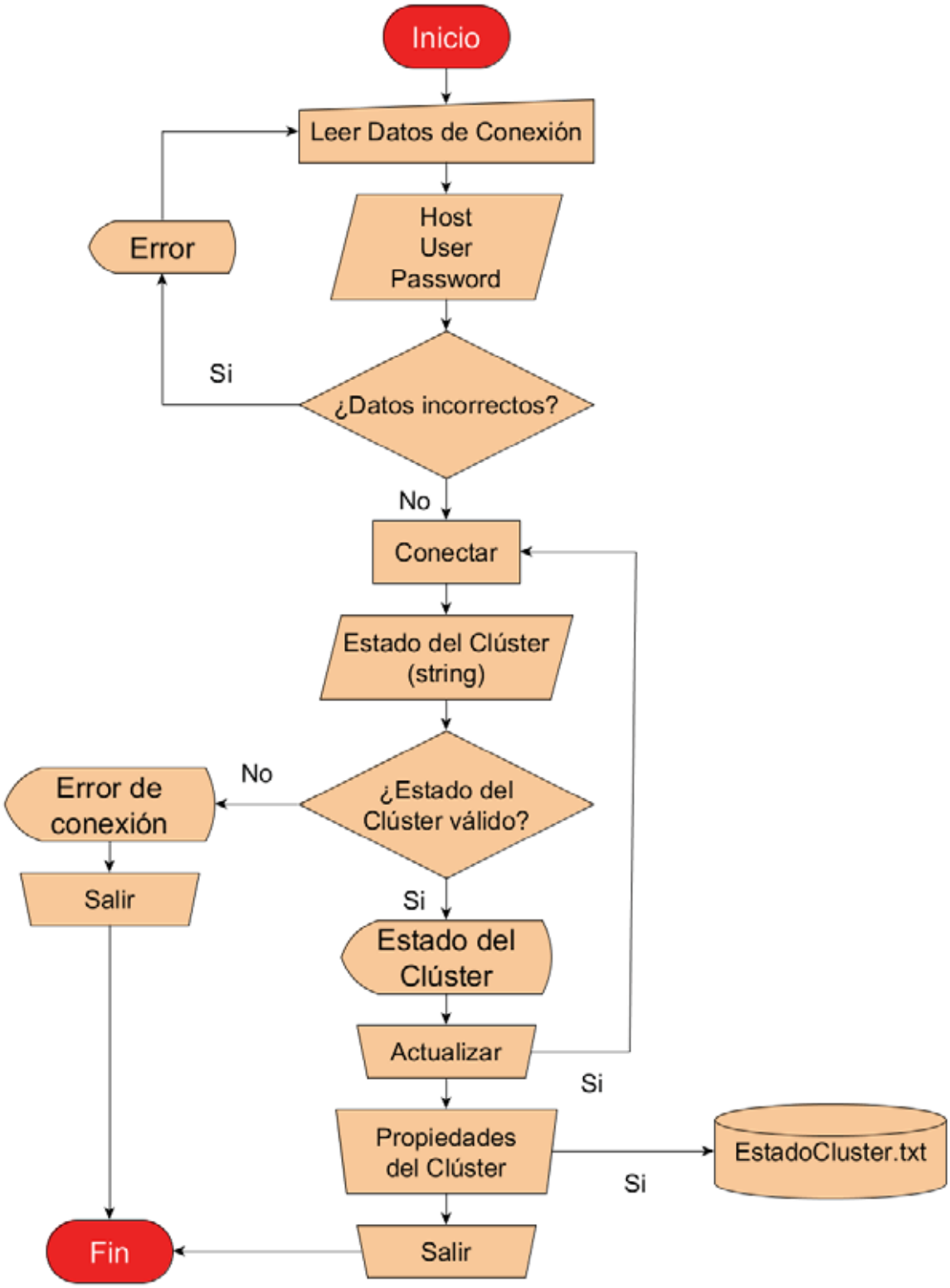
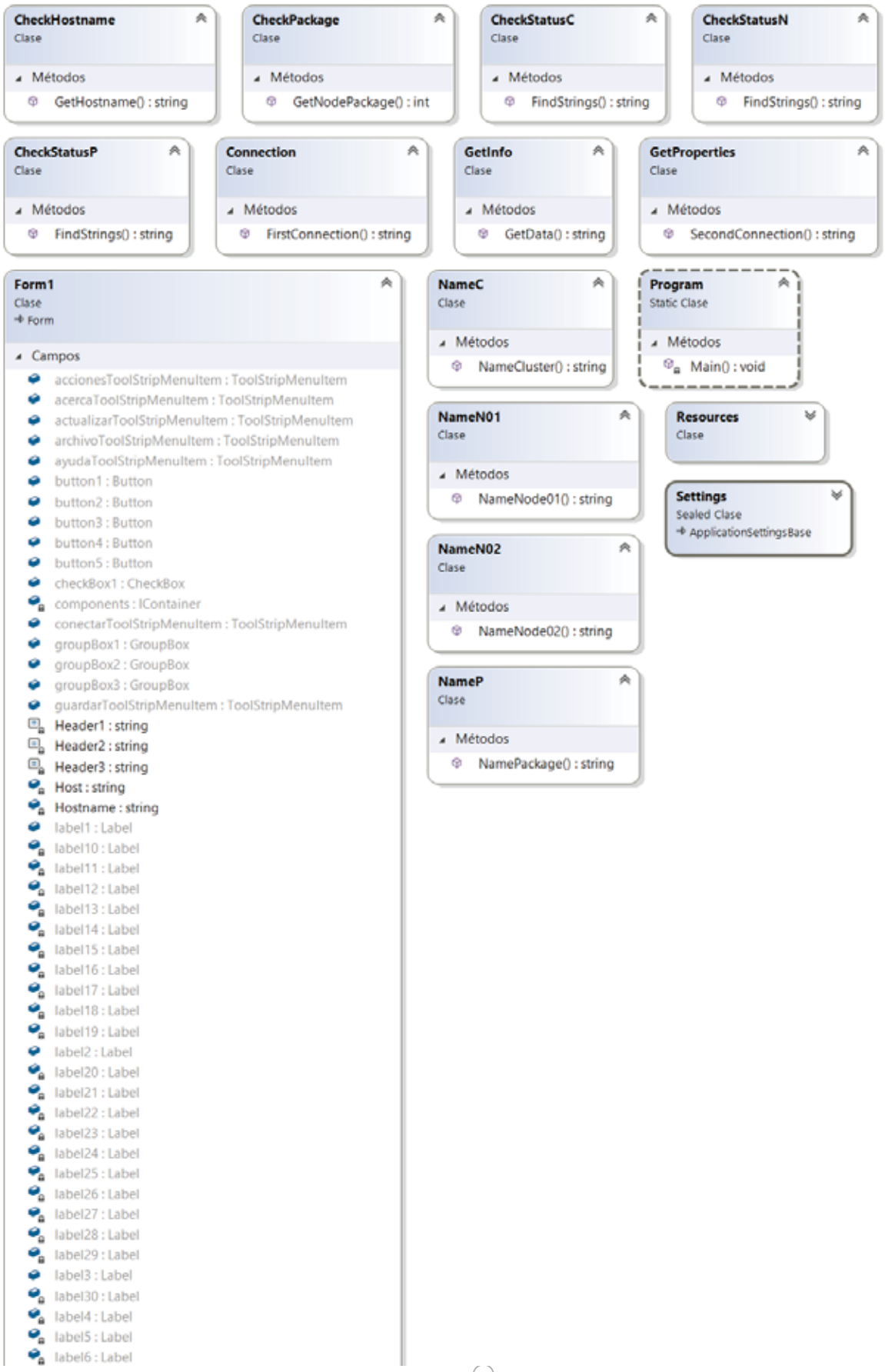


Figura 9. Diagrama de flujo del módulo de comunicación y estado.


CheckHostname
Clase
Métodos
GetHostname() : string
CheckPackage
Clase
Métodos
GetNodePackage() : int
CheckStatusC
Clase
Métodos
FindStrings() : string
CheckStatusN
Clase
Métodos
FindStrings() : string
CheckStatusP
Clase
Métodos
FindStrings() : string
Connection
Clase
Métodos
FirstConnection() : string
GetInfo
Clase
Métodos
GetData() : string
GetProperties
Clase
Métodos
SecondConnection() : string
Form1
Clase
Form
Campos
accionesToolStripMenuItem : ToolStripMenuItem
acercaToolStripMenuItem : ToolStripMenuItem
actualizarToolStripMenuItem : ToolStripMenuItem
archivoToolStripMenuItem : ToolStripMenuItem
ayudaToolStripMenuItem : ToolStripMenuItem
button1 : Button
button2 : Button
button3 : Button
button4 : Button
button5 : Button
checkBox1 : CheckBox
components : IContainer
conectarToolStripMenuItem : ToolStripMenuItem
groupBox1 : GroupBox
groupBox2 : GroupBox
groupBox3 : GroupBox
guardarToolStripMenuItem : ToolStripMenuItem
Header1 : string
Header2 : string
Header3 : string
Host : string
Hostname : string
label1 : Label
label10 : Label
label11 : Label
label12 : Label
label13 : Label
label14 : Label
label15 : Label
label16 : Label
label17 : Label
label18 : Label
label19 : Label
label2 : Label
label20 : Label
label21 : Label
label22 : Label
label23 : Label
label24 : Label
label25 : Label
label26 : Label
label27 : Label
label28 : Label
label29 : Label
label3 : Label
label30 : Label
label4 : Label
label5 : Label
label6 : Label
NameC
Clase
Métodos
NameCluster() : string
Program
Static Clase
Métodos
Main() : void
NameN01
Clase
Métodos
NameNode01() : string
Resources
Clase
Settings
Sealed Clase
ApplicationSettingsBase
NameN02
Clase
Métodos
NameNode02() : string
NameP
Clase
Métodos
NamePackage() : string

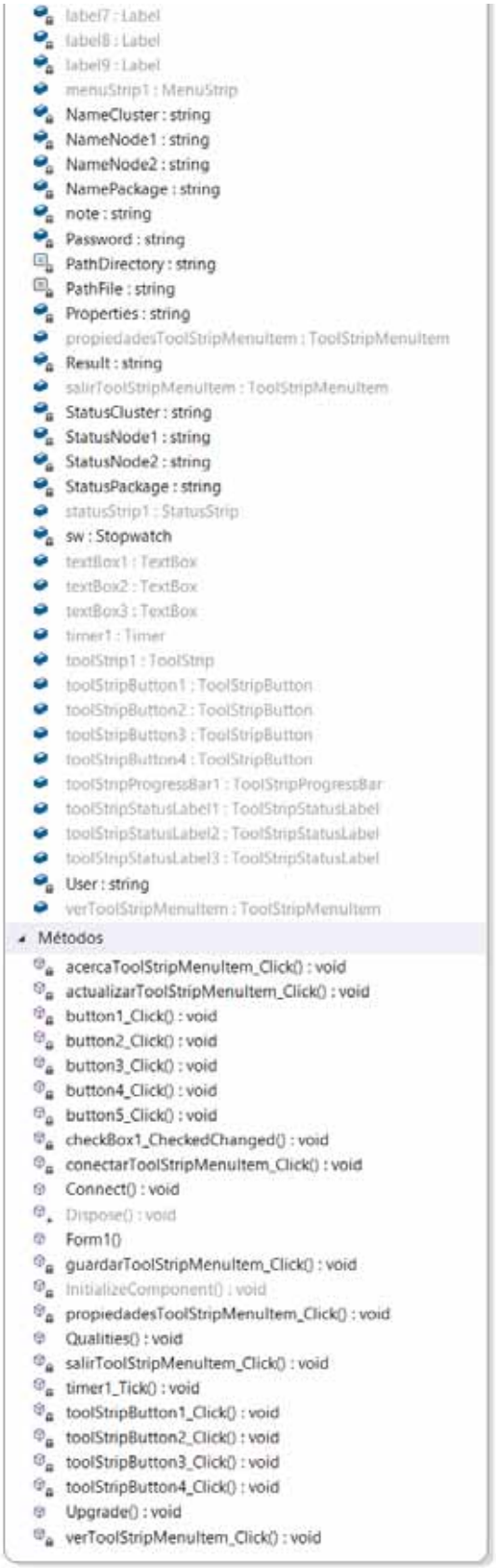


Figura 10. Clases del módulo de comunicación y estado.

La programación de este módulo se llevó a cabo con C#.

6.5.4. Código

La solución de Visual Studio 2017 que ejecuta la interfaz gráfica "Monitor de Cluster" recibe el nombre de ***ClusterGUI***. El **código fuente** del módulo gráfico es el archivo *Form1.Designer.cs*.

Los archivos:

- ✓ *CheckHostname.cs*
- ✓ *CheckPackage.cs*
- ✓ *CheckStatusC.cs*
- ✓ *CheckStatusN.cs*
- ✓ *CheckStatusP.cs*
- ✓ *Connection.cs*
- ✓ *Form1.cs*
- ✓ *GetInfo.cs*
- ✓ *GetProperties.cs*
- ✓ *NameC.cs*
- ✓ *NameN01.cs*
- ✓ *NameN02.cs*
- ✓ *NameP.cs*
- ✓ *Program.cs*

son el **código fuente** del módulo de comunicación y estado.

En el Anexo A se enlistan estos archivos.

7. Resultados

7.1. Prueba de conexión remota 1

Esta prueba se realizó el día 9 de abril de 2018. La prueba fue satisfactoria.

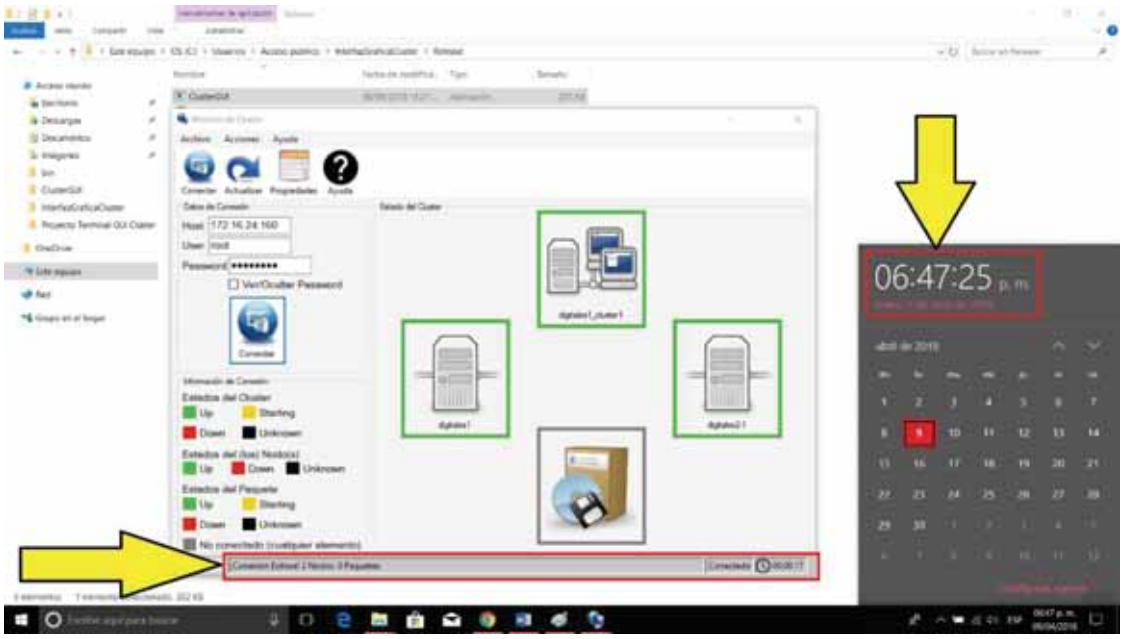


Figura 11. Resultado de la prueba de conexión remota 1.

7.2. Prueba de conexión remota 2

Esta prueba se realizó el día 10 de abril de 2018. La prueba fue satisfactoria.

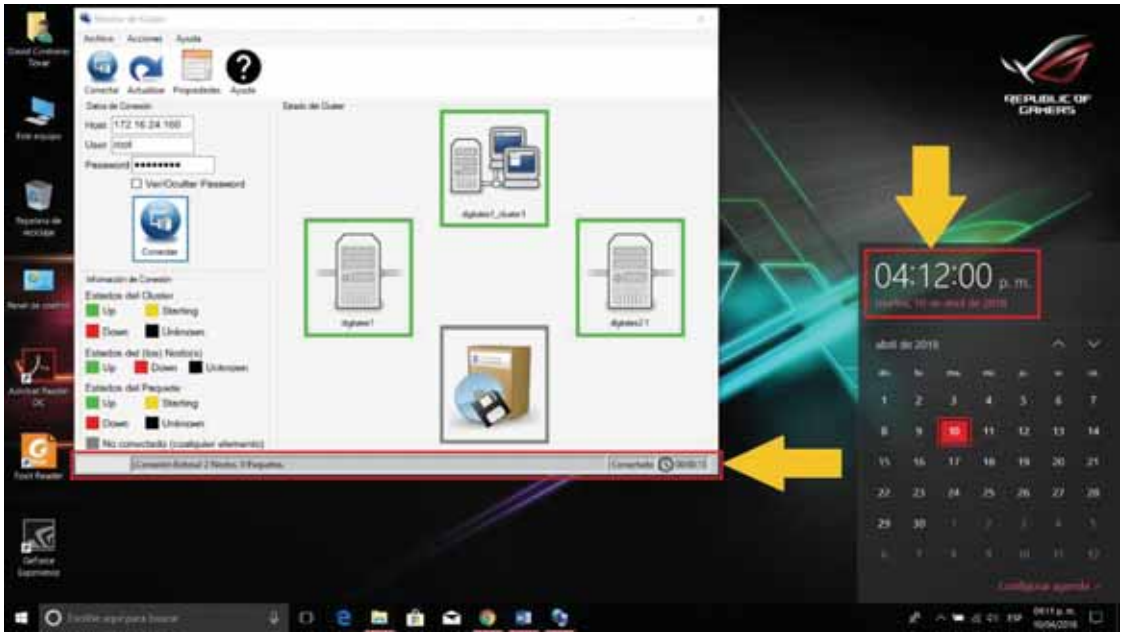


Figura 12. Resultado de la prueba de conexión remota 2.

8. Conclusiones

La redundancia natural que proveen los nodos de un clúster lo convierten en una plataforma confiable, tanto para la computación de alto rendimiento como para los servicios de alta disponibilidad. Así mismo, su alta escalabilidad, posibilidad de incorporación de recursos adicionales sin eliminar los ya existentes, y opciones de software libre para implementarlos, convierten a los clústeres Linux en una opción con un factor precio – rendimiento muy competitivo.

Los clústeres constituyen herramientas versátiles para enfrentar altas demandas en servicios de distinta índole. Sin embargo, el alto costo inicial de inversión que requieren para su implementación, sobre todo en el aspecto hardware (servidores, infraestructura de red, dispositivos de almacenamiento redundante, etc.), provoca que los usuarios de todos los niveles, en especial las pequeñas empresas, instituciones educativas y similares, posterguen o cancelen la decisión de instalar un clúster que cubra sus necesidades de alta disponiblidad y/o alto rendimiento. Además, la complejidad que supone implementar y administrar el clúster inhibe a los potenciales usuarios. Aunque Linux y UNIX cuentan con ambientes gráficos para el trabajo, la instalación, configuración y administración del clúster se realiza a través de línea de comandos. Las soluciones gráficas que están disponibles en el mercado son costosas, y generalmente los usuarios deben pagar además servicios de consultoría.

Este proyecto de integración ofrece una propuesta de solución de una interfaz gráfica para cubir la necesidad de obtener alta disponibilidad de una aplicación, con un clúster de servidores Linux. La interfaz gráfica desarrollada implementa la monitorización del clúster.

Como trabajos futuros, se pueden añadir comandos que permitan complementar la interfaz gráfica con funciones de administración del clúster.

Bibliografía

[1] Hernández Cervantes, Liliana, Santillán González, Alfredo J. y Caballero Cruz, Reyna E. (2003) "Maestros y Esclavos. Una aproximación de los Cúmulos de Computadoras" [En línea]. Revista Digital Universitaria. 30 de junio de 2003, vol. 4, no. 2. http://www.revista.unam.mx/vol.4/num2/art3/art3.htm

[2] Birke, R.; Chen, L.Y.; Smirni, E. (2012). Data Centers in the Cloud: A Large Scale Performance Study. 2012 IEEE 5th International Conference on Cloud Computing (CLOUD). 24 – 29 de junio de 2012. Páginas: 336 – 343.

[3] Janosepah, S.; Modiri, N.; Malakooti, M.V.; PahlavanTafti, A. (2011). A multi – tiered model approach for monitoring and control of data center entrance and exit scenarios. 2011 4th International Conference on Interaction Sciences (ICIS). 16 – 18 de agosto de 2011. Páginas: 129 – 135.

[4] Seymour, M.; Ikemoto, S. (2012). Design and management of data center effectiveness, risks and costs. 2012 28th Annual IEEE Semiconductor Thermal Measurement and Management Symposium (SEMI-THERM). 18 – 22 de marzo de 2012. Páginas: 64 – 68.

[5] Hackenberg, D. (2014). The Plenum concept: Improving scalability, security, and efficiency for data centers. 2014 IEEE Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems (ITherm). 27 – 30 de mayo de 2014. Páginas: 1137 – 1144.

[6] Takeuchi, Y.; Nakamura, T.; Tamura, H.; Kagawa, K.; Takahashi, K. (2011). The proposal of geographically distributed OSS against a great earthquake – A study on macroscale disaster recovery. 2011 13th Asia – Pacific Network Operations and Management Symposium (APNOMS). 21 – 23 de septiembre de 2011. Páginas: 1 – 8.

[7] Varghese, B.; McKee, G.; Alexandrov, V. (2010). Handling single node failures using agents in computer clusters. 2010 International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS). 1 – 14 de julio de 2010. Páginas: 96 – 101.

[8] Liu, Z.; Yang, X.; Sha, J. (2010). Fault detection for high availability RAID system. 2010 Sixth International Conference on Networked Computing and Advanced Information Management (NCM). 16 – 18 de agosto de 2010. Páginas: 27 – 32.

[9] Jian, S.; Weifeng, G.; Xiaoshe, D. (2014). High availability analysis and evaluation of heterogeneous dual computer fault-tolerant system. 2014 5th IEEE International Conference on Software Engineering and Service Science (ICSESS). 27 – 29 de junio de 2014. Páginas: 460 – 464.

[10] Alhazmi, O.H.; Malaiya, Y.K. (2013). Evaluating disaster recovery plans using the cloud. 2013 Proceedings – Annual Reliability and Maintainability Symposium (RAMS). 28 – 31 de enero de 2013. Páginas: 1 – 6.

[11] Zhiyang, G.; Yuanyuan, Y. (2015). On Nonblocking Multicast Fat-Tree Data Center Networks with Server Redundancy. 2015 IEEE Transactions on Computers. Volumen: 64. Número: 4. Páginas: 1058 – 1073.

[12] Al-Enezi, K.A.; Al-Shaikhli, I.F.; Al-Kandari, A.R.; Al-Mutairi, H.M. (2013). E-business Continuity and Disaster Recovery Plan Case Study Kuwait: Kuwait Government Entities (GEs). 2013 International Conference on Advanced Computer Science Applications and Technologies (ACSAT). 23 – 24 de diciembre de 2013. Páginas: 504 – 509.

[13] Tanner, A.L.; Pendleton, A.; Robinson, D.; Wicks, L. (2015). On modeling the emergency management process: Conceptualizing disaster planning and recovery. 2015 IEEE International Symposium on Technologies for Homeland Security (HST). 14 – 16 de abril de 2015. Páginas: 1 – 6.

[14] Costello, T. (2012). Business Continuity: Beyond Disaster Recovery IT Professional. Volumen: 14. Número: 5. Páginas: 64 – 64.

[15] Honglin, H.; Lin, Li.; Dehai, Z. (2012). Research and Implementation on Remote Disaster Recovery System. 2012 International Conference on Computer Science & Service System (CSSS). 11 – 13 de agosto de 2012. Páginas: 875 – 879.

[16] Chidambaram, J.; Prabhu, C.; Narasimha Rao, P.A.; Wankar, R.; Aneesh, C.S.; Agarwal, A. (2008). A methodology for high availability of data for business continuity planning / disaster recovery in a grid using replication in a distributed database. 2008 IEEE Region 10 Conference TENCON 2008. 19 – 21 de noviembre de 2008. Páginas: 1 – 6.

[17] Fei-fei, L.; Xiang-zhan, Y.; Gang, W. (2009). Design and Implementation of High Availability Distributed System Based on Multi-level Heartbeat Protocol. CASE 2009. IITA International Conference on Control, Automation and Systems Engineering. 11 – 12 de julio de 2009. Páginas: 83 – 87.

[18] A. Y. Perabeles Ríos, “Clúster de alta disponibilidad bajo sistema operativo Linux”, proyecto terminal (Lic. en Ingeniería Electrónica), División de Ciencias Básicas e Ingeniería, Universidad Autónoma Metropolitana, Unidad Azcapotzalco, México, 2008.

[19] A. U. Patiño González, "Clúster para Alta Disponibilidad con Microcontroladores", proyecto terminal (Lic. en Ingeniería Electrónica), División de Ciencias Básicas e Ingeniería, Universidad Autónoma Metropolitana, Unidad Azcapotzalco, México, 2011.

[20] Lenin Alcántara Roa, "Instalación y configuración de un clúster de alta disponibilidad con reparto de carga. Servidor Web y Máquinas Virtuales", tesis de maestría, Universidad Politécnica de Valencia, Valencia, febrero 2014.

[21] Santiago Barrio González, "Clúster Alta Disponibilidad sobre plataforma GNU/Linux (VERITAS)", tesis de licenciatura, Universitat Oberta de Catalunya, Cataluña, enero 2014.

[22] Introduction, 1. (2018). 1.2. High Availability Add-On Introduction - Red Hat Customer Portal. https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/high_availability_add-on_overview/s1-rhcs-intro-haao Acceso: 8 de abril de 2018.

[23] Hpe.com. (2018). HPE Serviceguard for Linux A.12.20.00. https://www.hpe.com/h20195/v2/GetPDF.aspx/c04154488.pdf Acceso: 8 de abril de 2018.

[24] Acero Quilumbaquín, W. A. (2016). Diseño e implementación de un clúster de alta disponibilidad para virtualizar los servidores de adquisición y procesamiento de datos del Instituto Geofísico. 226 hojas. Quito: EPN. http://bibdigital.epn.edu.ec/handle/15000/14756

[25] Sinisterra, M., Díaz Henao, T., y Ruiz López, E. (2012). Clúster de balanceo de carga y alta disponibilidad para servicios web y mail. *Informador Técnico, 76*, 93. https://doi.org/10.23850/22565035.34

[26] Visual Studio. (2018). IDE de Visual Studio, editor de código, Team Services y Mobile Center. https://www.visualstudio.com/es/ Acceso: 7 de abril de 2018.

[27] Docs.microsoft.com. (2018). Introducción al lenguaje C# y .NET Framework. https://docs.microsoft.com/es-mx/dotnet/csharp/getting-started/introduction-to-the-csharp-language-and-the-net-framework Acceso: 7 de abril de 2018.

[28] Microsoft. (2018). What is .NET?. https://www.microsoft.com/net/learn/what-is-dotnet Acceso: 7 de abril de 2018.

[29] Access.redhat.com. (2018). Product Documentation for Red Hat Enterprise Linux - Red Hat Customer Portal. https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/?version=6 Acceso: 7 de abril de 2018.

[30] Oracle.com. (2018). Oracle Database 12c | Oracle. http://www.oracle.com/us/corporate/features/database-12c/index.html Acceso: 7 de abril de 2018.

[31] Linuxtotal.com.mx. (2018). Administración de usuarios en Linux. https://www.linuxtotal.com.mx/index.php?cont=info_admon_008 Acceso: 7 de abril de 2018.

[32] Ssh.com. (2018). SSH (Secure Shell) Home Page | SSH.COM. https://www.ssh.com/ssh/ Acceso: 8 de abril de 2018.

[33] Docs.microsoft.com. (2018). Guía de introducción al uso de paquetes NuGet en Visual Studio. https://docs.microsoft.com/es-mx/nuget/quickstart/install-and-use-a-package-in-visual-studio Acceso: 8 de abril de 2018.

[34] Nuget.org. (2018). SSH.NET 2016.1.0. https://www.nuget.org/packages/SSH.NET/ Acceso: 8 de abril de 2018.

[35] GitHub. (2018). sshnet/SSH.NET. https://github.com/sshnet/SSH.NET/blob/master/LICENSE Acceso: 8 de abril de 2018.

[36] Unix.com. (2018). man page for cmviewcl (all section 1m) - Unix & Linux Commands. https://www.unix.com/man-page/All/1m/cmviewcl Acceso: 8 de abril de 2018.

[37]https://h50146.www5.hpe.com/products/software/oe/linux/mainstream/support/doc/other/ha_cluster/pdfs/sg_201708_a00018693en_us.pdf Página: 252. Acceso: 8 de abril de 2018.

9. Anexo y Entregables

9.1. Anexo A: Código de la interfaz gráfica “Monitor de Cluster”.

9.2. Entregable 1: Manual de administración del clúster.

9.3. Entregable 2: Manual de usuario de la interfaz gráfica.

Anexo A

Archivo fuente correspondiente al módulo gráfico.

Form1.Designer.cs

```
namespace ClusterGUI
{
    partial class Form1
    {
        /// <summary>
        /// Variable del diseñador necesaria.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Limpiar los recursos que se estén usando.
        /// </summary>
        /// <param name="disposing">true si los recursos administrados se deben desechar; false en caso contrario.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region

        /// <summary>
        /// Método necesario para admitir el Diseñador. No se puede modificar
        /// el contenido de este método con el editor de código.
        /// </summary>
        private void InitializeComponent()
        {
            this.components = new System.ComponentModel.Container();
            System.ComponentModel.ComponentResourceManager resources = new
System.ComponentModel.ComponentResourceManager(typeof(Form1));
            this.menuStrip1 = new System.Windows.Forms.MenuStrip();
            this.archivoToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
            this.guardarToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
            this.salirToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
            this.accionesToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
            this.conectarToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
            this.actualizarToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
            this.propiedadesToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
            this.ayudaToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
            this.verToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
            this.acercaToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
            this.toolStrip1 = new System.Windows.Forms.ToolStrip();
            this.toolStripButton1 = new System.Windows.Forms.ToolStripButton();
            this.toolStripButton2 = new System.Windows.Forms.ToolStripButton();
            this.toolStripButton3 = new System.Windows.Forms.ToolStripButton();
            this.toolStripButton4 = new System.Windows.Forms.ToolStripButton();
            this.statusStrip1 = new System.Windows.Forms.StatusStrip();
            this.toolStripProgressBar1 = new System.Windows.Forms.ToolStripProgressBar();
            this.toolStripStatusLabel1 = new System.Windows.Forms.ToolStripStatusLabel();
            this.toolStripStatusLabel2 = new System.Windows.Forms.ToolStripStatusLabel();
            this.toolStripStatusLabel3 = new System.Windows.Forms.ToolStripStatusLabel();
            this.groupBox1 = new System.Windows.Forms.GroupBox();
            this.button4 = new System.Windows.Forms.Button();
            this.button3 = new System.Windows.Forms.Button();
            this.button2 = new System.Windows.Forms.Button();
            this.button1 = new System.Windows.Forms.Button();
            this.groupBox2 = new System.Windows.Forms.GroupBox();
            this.button5 = new System.Windows.Forms.Button();
            this.checkBox1 = new System.Windows.Forms.CheckBox();
            this.textBox3 = new System.Windows.Forms.TextBox();
            this.label3 = new System.Windows.Forms.Label();
            this.textBox2 = new System.Windows.Forms.TextBox();
            this.label2 = new System.Windows.Forms.Label();
            this.textBox1 = new System.Windows.Forms.TextBox();
```

```
this.label1 = new System.Windows.Forms.Label();
this.groupBox3 = new System.Windows.Forms.GroupBox();
this.label30 = new System.Windows.Forms.Label();
this.label29 = new System.Windows.Forms.Label();
this.label28 = new System.Windows.Forms.Label();
this.label27 = new System.Windows.Forms.Label();
this.label26 = new System.Windows.Forms.Label();
this.label25 = new System.Windows.Forms.Label();
this.label24 = new System.Windows.Forms.Label();
this.label23 = new System.Windows.Forms.Label();
this.label22 = new System.Windows.Forms.Label();
this.label21 = new System.Windows.Forms.Label();
this.label20 = new System.Windows.Forms.Label();
this.label19 = new System.Windows.Forms.Label();
this.label18 = new System.Windows.Forms.Label();
this.label17 = new System.Windows.Forms.Label();
this.label16 = new System.Windows.Forms.Label();
this.label15 = new System.Windows.Forms.Label();
this.label14 = new System.Windows.Forms.Label();
this.label13 = new System.Windows.Forms.Label();
this.label12 = new System.Windows.Forms.Label();
this.label11 = new System.Windows.Forms.Label();
this.label10 = new System.Windows.Forms.Label();
this.label9 = new System.Windows.Forms.Label();
this.label8 = new System.Windows.Forms.Label();
this.label7 = new System.Windows.Forms.Label();
this.label6 = new System.Windows.Forms.Label();
this.label5 = new System.Windows.Forms.Label();
this.label4 = new System.Windows.Forms.Label();
this.timer1 = new System.Windows.Forms.Timer(this.components);
this.menuStrip1.SuspendLayout();
this.toolStrip1.SuspendLayout();
this.statusStrip1.SuspendLayout();
this.groupBox1.SuspendLayout();
this.groupBox2.SuspendLayout();
this.groupBox3.SuspendLayout();
this.SuspendLayout();
//
// menuStrip1
//
this.menuStrip1.BackColor = System.Drawing.Color.WhiteSmoke;
this.menuStrip1.ImageScalingSize = new System.Drawing.Size(20, 20);
this.menuStrip1.Items.AddRange(new System.Windows.Forms.ToolStripItem[] {
this.archivoToolStripMenuItem,
this.accionesToolStripMenuItem,
this.ayudaToolStripMenuItem});
this.menuStrip1.Location = new System.Drawing.Point(0, 0);
this.menuStrip1.Name = "menuStrip1";
this.menuStrip1.Size = new System.Drawing.Size(1182, 28);
this.menuStrip1.TabIndex = 0;
this.menuStrip1.Text = "menuStrip1";
//
// archivoToolStripMenuItem
//
this.archivoToolStripMenuItem.DropDownItems.AddRange(new System.Windows.Forms.ToolStripItem[] {
this.guardarToolStripMenuItem,
this.salirToolStripMenuItem});
this.archivoToolStripMenuItem.Name = "archivoToolStripMenuItem";
this.archivoToolStripMenuItem.Size = new System.Drawing.Size(71, 24);
this.archivoToolStripMenuItem.Text = "Archivo";
//
// guardarToolStripMenuItem
//
this.guardarToolStripMenuItem.Name = "guardarToolStripMenuItem";
this.guardarToolStripMenuItem.Size = new System.Drawing.Size(224, 26);
this.guardarToolStripMenuItem.Text = "Guardar Propiedades";
this.guardarToolStripMenuItem.Click += new System.EventHandler(this.guardarToolStripMenuItem_Click);
//
// salirToolStripMenuItem
//
this.salirToolStripMenuItem.Name = "salirToolStripMenuItem";
this.salirToolStripMenuItem.Size = new System.Drawing.Size(224, 26);
this.salirToolStripMenuItem.Text = "Salir";
this.salirToolStripMenuItem.Click += new System.EventHandler(this.salirToolStripMenuItem_Click);
//
```

```
// accionesToolStripMenuItem
//
this.accionesToolStripMenuItem.DropDownItems.AddRange(new System.Windows.Forms.ToolStripItem[] {
this.conectarToolStripMenuItem,
this.actualizarToolStripMenuItem,
this.propiedadesToolStripMenuItem});
this.accionesToolStripMenuItem.Name = "accionesToolStripMenuItem";
this.accionesToolStripMenuItem.Size = new System.Drawing.Size(80, 24);
this.accionesToolStripMenuItem.Text = "Acciones";
//
// conectarToolStripMenuItem
//
this.conectarToolStripMenuItem.Name = "conectarToolStripMenuItem";
this.conectarToolStripMenuItem.Size = new System.Drawing.Size(167, 26);
this.conectarToolStripMenuItem.Text = "Conectar";
this.conectarToolStripMenuItem.Click += new System.EventHandler(this.conectarToolStripMenuItem_Click);
//
// actualizarToolStripMenuItem
//
this.actualizarToolStripMenuItem.Name = "actualizarToolStripMenuItem";
this.actualizarToolStripMenuItem.Size = new System.Drawing.Size(167, 26);
this.actualizarToolStripMenuItem.Text = "Actualizar";
this.actualizarToolStripMenuItem.Click += new System.EventHandler(this.actualizarToolStripMenuItem_Click);
//
// propiedadesToolStripMenuItem
//
this.propiedadesToolStripMenuItem.Name = "propiedadesToolStripMenuItem";
this.propiedadesToolStripMenuItem.Size = new System.Drawing.Size(167, 26);
this.propiedadesToolStripMenuItem.Text = "Propiedades";
this.propiedadesToolStripMenuItem.Click += new System.EventHandler(this.propiedadesToolStripMenuItem_Click);
//
// ayudaToolStripMenuItem
//
this.ayudaToolStripMenuItem.DropDownItems.AddRange(new System.Windows.Forms.ToolStripItem[] {
this.verToolStripMenuItem,
this.acercaToolStripMenuItem});
this.ayudaToolStripMenuItem.Name = "ayudaToolStripMenuItem";
this.ayudaToolStripMenuItem.Size = new System.Drawing.Size(63, 24);
this.ayudaToolStripMenuItem.Text = "Ayuda";
//
// verToolStripMenuItem
//
this.verToolStripMenuItem.Name = "verToolStripMenuItem";
this.verToolStripMenuItem.Size = new System.Drawing.Size(277, 26);
this.verToolStripMenuItem.Text = "Ver Ayuda";
this.verToolStripMenuItem.Click += new System.EventHandler(this.verToolStripMenuItem_Click);
//
// acercaToolStripMenuItem
//
this.acercaToolStripMenuItem.Name = "acercaToolStripMenuItem";
this.acercaToolStripMenuItem.Size = new System.Drawing.Size(277, 26);
this.acercaToolStripMenuItem.Text = "Acerca de Monitor de Cluster";
this.acercaToolStripMenuItem.Click += new System.EventHandler(this.acercaToolStripMenuItem_Click);
//
// toolStrip1
//
this.toolStrip1.BackColor = System.Drawing.Color.White;
this.toolStrip1.ImageScalingSize = new System.Drawing.Size(48, 48);
this.toolStrip1.Items.AddRange(new System.Windows.Forms.ToolStripItem[] {
this.toolStripButton1,
this.toolStripButton2,
this.toolStripButton3,
this.toolStripButton4});
this.toolStrip1.Location = new System.Drawing.Point(0, 28);
this.toolStrip1.Name = "toolStrip1";
this.toolStrip1.Size = new System.Drawing.Size(1182, 75);
this.toolStrip1.TabIndex = 1;
this.toolStrip1.Text = "toolStrip1";
//
// toolStripButton1
//
this.toolStripButton1.BackColor = System.Drawing.Color.White;
this.toolStripButton1.Image = ((System.Drawing.Image)(resources.GetObject("toolStripButton1.Image")));
this.toolStripButton1.ImageTransparentColor = System.Drawing.Color.Magenta;
this.toolStripButton1.Name = "toolStripButton1";
```

```
            this.toolStripButton1.Size = new System.Drawing.Size(72, 72);
            this.toolStripButton1.Text = "Conectar";
            this.toolStripButton1.TextImageRelation = System.Windows.Forms.TextImageRelation.ImageAboveText;
            this.toolStripButton1.Click += new System.EventHandler(this.toolStripButton1_Click);
            //
            // toolStripButton2
            //
            this.toolStripButton2.Image = ((System.Drawing.Image)(resources.GetObject("toolStripButton2.Image")));
            this.toolStripButton2.ImageTransparentColor = System.Drawing.Color.Magenta;
            this.toolStripButton2.Name = "toolStripButton2";
            this.toolStripButton2.Size = new System.Drawing.Size(79, 72);
            this.toolStripButton2.Text = "Actualizar";
            this.toolStripButton2.TextImageRelation = System.Windows.Forms.TextImageRelation.ImageAboveText;
            this.toolStripButton2.Click += new System.EventHandler(this.toolStripButton2_Click);
            //
            // toolStripButton3
            //
            this.toolStripButton3.Image = ((System.Drawing.Image)(resources.GetObject("toolStripButton3.Image")));
            this.toolStripButton3.ImageTransparentColor = System.Drawing.Color.Magenta;
            this.toolStripButton3.Name = "toolStripButton3";
            this.toolStripButton3.Size = new System.Drawing.Size(96, 72);
            this.toolStripButton3.Text = "Propiedades";
            this.toolStripButton3.TextImageRelation = System.Windows.Forms.TextImageRelation.ImageAboveText;
            this.toolStripButton3.Click += new System.EventHandler(this.toolStripButton3_Click);
            //
            // toolStripButton4
            //
            this.toolStripButton4.Image = ((System.Drawing.Image)(resources.GetObject("toolStripButton4.Image")));
            this.toolStripButton4.ImageTransparentColor = System.Drawing.Color.Magenta;
            this.toolStripButton4.Name = "toolStripButton4";
            this.toolStripButton4.Size = new System.Drawing.Size(55, 72);
            this.toolStripButton4.Text = "Ayuda";
            this.toolStripButton4.TextImageRelation = System.Windows.Forms.TextImageRelation.ImageAboveText;
            this.toolStripButton4.Click += new System.EventHandler(this.toolStripButton4_Click);
            //
            // statusStrip1
            //
            this.statusStrip1.BackColor = System.Drawing.SystemColors.ScrollBar;
            this.statusStrip1.ImageScalingSize = new System.Drawing.Size(20, 20);
            this.statusStrip1.Items.AddRange(new System.Windows.Forms.ToolStripItem[] {
            this.toolStripProgressBar1,
            this.toolStripStatusLabel1,
            this.toolStripStatusLabel2,
            this.toolStripStatusLabel3});
            this.statusStrip1.Location = new System.Drawing.Point(0, 723);
            this.statusStrip1.Name = "statusStrip1";
            this.statusStrip1.Size = new System.Drawing.Size(1182, 30);
            this.statusStrip1.SizingGrip = false;
            this.statusStrip1.TabIndex = 2;
            this.statusStrip1.Text = "statusStrip1";
            //
            // toolStripProgressBar1
            //
            this.toolStripProgressBar1.BackColor = System.Drawing.SystemColors.ScrollBar;
            this.toolStripProgressBar1.ForeColor = System.Drawing.SystemColors.HotTrack;
            this.toolStripProgressBar1.Name = "toolStripProgressBar1";
            this.toolStripProgressBar1.Size = new System.Drawing.Size(100, 24);
            this.toolStripProgressBar1.Style = System.Windows.Forms.ProgressBarStyle.Continuous;
            //
            // toolStripStatusLabel1
            //
            this.toolStripStatusLabel1.BorderSides =
((System.Windows.Forms.ToolStripStatusLabelBorderSides)((((System.Windows.Forms.ToolStripStatusLabelBorderSides.Left |
System.Windows.Forms.ToolStripStatusLabelBorderSides.Top)
            | System.Windows.Forms.ToolStripStatusLabelBorderSides.Right)
            | System.Windows.Forms.ToolStripStatusLabelBorderSides.Bottom)));
            this.toolStripStatusLabel1.BorderStyle = System.Windows.Forms.Border3DStyle.Sunken;
            this.toolStripStatusLabel1.DisplayStyle = System.Windows.Forms.ToolStripItemDisplayStyle.Text;
            this.toolStripStatusLabel1.Name = "toolStripStatusLabel1";
            this.toolStripStatusLabel1.Size = new System.Drawing.Size(967, 25);
            this.toolStripStatusLabel1.Spring = true;
            this.toolStripStatusLabel1.Text = "- - -";
            this.toolStripStatusLabel1.TextAlign = System.Drawing.ContentAlignment.MiddleLeft;
            this.toolStripStatusLabel1.TextImageRelation = System.Windows.Forms.TextImageRelation.TextBeforeImage;
            //
```

```
            // toolStripStatusLabel2
            //
            this.toolStripStatusLabel2.BorderSides =
((System.Windows.Forms.ToolStripStatusLabelBorderSides)((((System.Windows.Forms.ToolStripStatusLabelBorderSides.Left |
System.Windows.Forms.ToolStripStatusLabelBorderSides.Top)
            | System.Windows.Forms.ToolStripStatusLabelBorderSides.Right)
            | System.Windows.Forms.ToolStripStatusLabelBorderSides.Bottom)));
            this.toolStripStatusLabel2.BorderStyle = System.Windows.Forms.Border3DStyle.Sunken;
            this.toolStripStatusLabel2.DisplayStyle = System.Windows.Forms.ToolStripItemDisplayStyle.Text;
            this.toolStripStatusLabel2.Name = "toolStripStatusLabel2";
            this.toolStripStatusLabel2.Size = new System.Drawing.Size(39, 25);
            this.toolStripStatusLabel2.Text = "- - -";
            this.toolStripStatusLabel2.TextImageRelation = System.Windows.Forms.TextImageRelation.TextBeforeImage;
            //
            // toolStripStatusLabel3
            //
            this.toolStripStatusLabel3.BorderSides =
((System.Windows.Forms.ToolStripStatusLabelBorderSides)((((System.Windows.Forms.ToolStripStatusLabelBorderSides.Left |
System.Windows.Forms.ToolStripStatusLabelBorderSides.Top)
            | System.Windows.Forms.ToolStripStatusLabelBorderSides.Right)
            | System.Windows.Forms.ToolStripStatusLabelBorderSides.Bottom)));
            this.toolStripStatusLabel3.BorderStyle = System.Windows.Forms.Border3DStyle.Sunken;
            this.toolStripStatusLabel3.Image =
((System.Drawing.Image)(resources.GetObject("toolStripStatusLabel3.Image")));
            this.toolStripStatusLabel3.Name = "toolStripStatusLabel3";
            this.toolStripStatusLabel3.Size = new System.Drawing.Size(59, 25);
            this.toolStripStatusLabel3.Text = "- - -";
            //
            // groupBox1
            //
            this.groupBox1.Controls.Add(this.button4);
            this.groupBox1.Controls.Add(this.button3);
            this.groupBox1.Controls.Add(this.button2);
            this.groupBox1.Controls.Add(this.button1);
            this.groupBox1.Location = new System.Drawing.Point(370, 120);
            this.groupBox1.Name = "groupBox1";
            this.groupBox1.Size = new System.Drawing.Size(800, 595);
            this.groupBox1.TabIndex = 3;
            this.groupBox1.TabStop = false;
            this.groupBox1.Text = "Estado del Cluster";
            //
            // button4
            //
            this.button4.FlatAppearance.BorderColor = System.Drawing.Color.Gray;
            this.button4.FlatAppearance.BorderSize = 5;
            this.button4.FlatStyle = System.Windows.Forms.FlatStyle.Flat;
            this.button4.Image = ((System.Drawing.Image)(resources.GetObject("button4.Image")));
            this.button4.Location = new System.Drawing.Point(300, 380);
            this.button4.Name = "button4";
            this.button4.Size = new System.Drawing.Size(200, 200);
            this.button4.TabIndex = 3;
            this.button4.TextImageRelation = System.Windows.Forms.TextImageRelation.ImageAboveText;
            this.button4.UseVisualStyleBackColor = true;
            this.button4.Click += new System.EventHandler(this.button4_Click);
            //
            // button3
            //
            this.button3.FlatAppearance.BorderColor = System.Drawing.Color.Gray;
            this.button3.FlatAppearance.BorderSize = 5;
            this.button3.FlatStyle = System.Windows.Forms.FlatStyle.Flat;
            this.button3.Image = ((System.Drawing.Image)(resources.GetObject("button3.Image")));
            this.button3.Location = new System.Drawing.Point(550, 200);
            this.button3.Name = "button3";
            this.button3.Size = new System.Drawing.Size(200, 200);
            this.button3.TabIndex = 2;
            this.button3.TextImageRelation = System.Windows.Forms.TextImageRelation.ImageAboveText;
            this.button3.UseVisualStyleBackColor = true;
            this.button3.Click += new System.EventHandler(this.button3_Click);
            //
            // button2
            //
            this.button2.FlatAppearance.BorderColor = System.Drawing.Color.Gray;
            this.button2.FlatAppearance.BorderSize = 5;
            this.button2.FlatStyle = System.Windows.Forms.FlatStyle.Flat;
            this.button2.Image = ((System.Drawing.Image)(resources.GetObject("button2.Image")));
```

```
            this.button2.Location = new System.Drawing.Point(50, 200);
            this.button2.Name = "button2";
            this.button2.Size = new System.Drawing.Size(200, 200);
            this.button2.TabIndex = 1;
            this.button2.TextImageRelation = System.Windows.Forms.TextImageRelation.ImageAboveText;
            this.button2.UseVisualStyleBackColor = true;
            this.button2.Click += new System.EventHandler(this.button2_Click);
            //
            // button1
            //
            this.button1.FlatAppearance.BorderColor = System.Drawing.Color.Gray;
            this.button1.FlatAppearance.BorderSize = 5;
            this.button1.FlatStyle = System.Windows.Forms.FlatStyle.Flat;
            this.button1.Image = ((System.Drawing.Image)(resources.GetObject("button1.Image")));
            this.button1.Location = new System.Drawing.Point(300, 15);
            this.button1.Name = "button1";
            this.button1.Size = new System.Drawing.Size(200, 200);
            this.button1.TabIndex = 0;
            this.button1.TextImageRelation = System.Windows.Forms.TextImageRelation.ImageAboveText;
            this.button1.UseVisualStyleBackColor = true;
            this.button1.Click += new System.EventHandler(this.button1_Click);
            //
            // groupBox2
            //
            this.groupBox2.Controls.Add(this.button5);
            this.groupBox2.Controls.Add(this.checkBox1);
            this.groupBox2.Controls.Add(this.textBox3);
            this.groupBox2.Controls.Add(this.label3);
            this.groupBox2.Controls.Add(this.textBox2);
            this.groupBox2.Controls.Add(this.label2);
            this.groupBox2.Controls.Add(this.textBox1);
            this.groupBox2.Controls.Add(this.label1);
            this.groupBox2.Location = new System.Drawing.Point(10, 120);
            this.groupBox2.Name = "groupBox2";
            this.groupBox2.Size = new System.Drawing.Size(350, 290);
            this.groupBox2.TabIndex = 4;
            this.groupBox2.TabStop = false;
            this.groupBox2.Text = "Datos de Conexión";
            //
            // button5
            //
            this.button5.BackColor = System.Drawing.Color.White;
            this.button5.Font = new System.Drawing.Font("Microsoft Sans Serif", 9F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.button5.Image = ((System.Drawing.Image)(resources.GetObject("button5.Image")));
            this.button5.Location = new System.Drawing.Point(95, 160);
            this.button5.Name = "button5";
            this.button5.Size = new System.Drawing.Size(105, 115);
            this.button5.TabIndex = 8;
            this.button5.Text = "Conectar";
            this.button5.TextImageRelation = System.Windows.Forms.TextImageRelation.ImageAboveText;
            this.button5.UseVisualStyleBackColor = false;
            this.button5.Click += new System.EventHandler(this.button5_Click);
            //
            // checkBox1
            //
            this.checkBox1.AutoSize = true;
            this.checkBox1.Font = new System.Drawing.Font("Microsoft Sans Serif", 10.2F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.checkBox1.Location = new System.Drawing.Point(95, 128);
            this.checkBox1.Name = "checkBox1";
            this.checkBox1.Size = new System.Drawing.Size(196, 24);
            this.checkBox1.TabIndex = 7;
            this.checkBox1.Text = "Ver/Ocultar Password";
            this.checkBox1.UseVisualStyleBackColor = true;
            this.checkBox1.CheckedChanged += new System.EventHandler(this.checkBox1_CheckedChanged);
            //
            // textBox3
            //
            this.textBox3.Font = new System.Drawing.Font("Arial", 10.8F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.textBox3.Location = new System.Drawing.Point(95, 94);
            this.textBox3.Name = "textBox3";
            this.textBox3.Size = new System.Drawing.Size(150, 28);
            this.textBox3.TabIndex = 6;
```

```
            this.textBox3.UseSystemPasswordChar = true;
            //
            // label3
            //
            this.label3.AutoSize = true;
            this.label3.Font = new System.Drawing.Font("Microsoft Sans Serif", 10.2F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.label3.Location = new System.Drawing.Point(6, 97);
            this.label3.Name = "label3";
            this.label3.Size = new System.Drawing.Size(83, 20);
            this.label3.TabIndex = 5;
            this.label3.Text = "Password";
            //
            // textBox2
            //
            this.textBox2.Font = new System.Drawing.Font("Arial", 10.8F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.textBox2.Location = new System.Drawing.Point(57, 60);
            this.textBox2.Name = "textBox2";
            this.textBox2.Size = new System.Drawing.Size(150, 28);
            this.textBox2.TabIndex = 3;
            //
            // label2
            //
            this.label2.AutoSize = true;
            this.label2.Font = new System.Drawing.Font("Microsoft Sans Serif", 10.2F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.label2.Location = new System.Drawing.Point(6, 63);
            this.label2.Name = "label2";
            this.label2.Size = new System.Drawing.Size(45, 20);
            this.label2.TabIndex = 2;
            this.label2.Text = "User";
            //
            // textBox1
            //
            this.textBox1.Font = new System.Drawing.Font("Arial", 10.8F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.textBox1.Location = new System.Drawing.Point(57, 26);
            this.textBox1.Name = "textBox1";
            this.textBox1.Size = new System.Drawing.Size(150, 28);
            this.textBox1.TabIndex = 1;
            //
            // label1
            //
            this.label1.AutoSize = true;
            this.label1.Font = new System.Drawing.Font("Microsoft Sans Serif", 10.2F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.label1.Location = new System.Drawing.Point(6, 30);
            this.label1.Name = "label1";
            this.label1.Size = new System.Drawing.Size(45, 20);
            this.label1.TabIndex = 0;
            this.label1.Text = "Host";
            //
            // groupBox3
            //
            this.groupBox3.Controls.Add(this.label30);
            this.groupBox3.Controls.Add(this.label29);
            this.groupBox3.Controls.Add(this.label28);
            this.groupBox3.Controls.Add(this.label27);
            this.groupBox3.Controls.Add(this.label26);
            this.groupBox3.Controls.Add(this.label25);
            this.groupBox3.Controls.Add(this.label24);
            this.groupBox3.Controls.Add(this.label23);
            this.groupBox3.Controls.Add(this.label22);
            this.groupBox3.Controls.Add(this.label21);
            this.groupBox3.Controls.Add(this.label20);
            this.groupBox3.Controls.Add(this.label19);
            this.groupBox3.Controls.Add(this.label18);
            this.groupBox3.Controls.Add(this.label17);
            this.groupBox3.Controls.Add(this.label16);
            this.groupBox3.Controls.Add(this.label15);
            this.groupBox3.Controls.Add(this.label14);
            this.groupBox3.Controls.Add(this.label13);
            this.groupBox3.Controls.Add(this.label12);
            this.groupBox3.Controls.Add(this.label11);
```

```
            this.groupBox3.Controls.Add(this.label10);
            this.groupBox3.Controls.Add(this.label9);
            this.groupBox3.Controls.Add(this.label8);
            this.groupBox3.Controls.Add(this.label7);
            this.groupBox3.Controls.Add(this.label6);
            this.groupBox3.Controls.Add(this.label5);
            this.groupBox3.Controls.Add(this.label4);
            this.groupBox3.Location = new System.Drawing.Point(10, 415);
            this.groupBox3.Name = "groupBox3";
            this.groupBox3.Size = new System.Drawing.Size(350, 300);
            this.groupBox3.TabIndex = 5;
            this.groupBox3.TabStop = false;
            this.groupBox3.Text = "Información de Conexión";
            //
            // label30
            //
            this.label30.AutoSize = true;
            this.label30.Font = new System.Drawing.Font("Microsoft Sans Serif", 10.2F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.label30.Location = new System.Drawing.Point(37, 275);
            this.label30.Name = "label30";
            this.label30.Size = new System.Drawing.Size(270, 20);
            this.label30.TabIndex = 26;
            this.label30.Text = "No conectado (cualquier elemento)";
            //
            // label29
            //
            this.label29.AutoSize = true;
            this.label29.BackColor = System.Drawing.Color.Gray;
            this.label29.Font = new System.Drawing.Font("Microsoft Sans Serif", 10.2F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.label29.ForeColor = System.Drawing.Color.Gray;
            this.label29.Location = new System.Drawing.Point(10, 275);
            this.label29.Name = "label29";
            this.label29.Size = new System.Drawing.Size(21, 20);
            this.label29.TabIndex = 25;
            this.label29.Text = "--";
            //
            // label28
            //
            this.label28.AutoSize = true;
            this.label28.Font = new System.Drawing.Font("Microsoft Sans Serif", 10.2F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.label28.Location = new System.Drawing.Point(147, 240);
            this.label28.Name = "label28";
            this.label28.Size = new System.Drawing.Size(77, 20);
            this.label28.TabIndex = 24;
            this.label28.Text = "Unknown";
            //
            // label27
            //
            this.label27.AutoSize = true;
            this.label27.BackColor = System.Drawing.Color.Black;
            this.label27.Font = new System.Drawing.Font("Microsoft Sans Serif", 10.2F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.label27.ForeColor = System.Drawing.Color.Black;
            this.label27.Location = new System.Drawing.Point(120, 240);
            this.label27.Name = "label27";
            this.label27.Size = new System.Drawing.Size(21, 20);
            this.label27.TabIndex = 23;
            this.label27.Text = "--";
            //
            // label26
            //
            this.label26.AutoSize = true;
            this.label26.Font = new System.Drawing.Font("Microsoft Sans Serif", 10.2F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.label26.Location = new System.Drawing.Point(37, 240);
            this.label26.Name = "label26";
            this.label26.Size = new System.Drawing.Size(52, 20);
            this.label26.TabIndex = 22;
            this.label26.Text = "Down";
            //
            // label25
            //
```

```
            this.label25.AutoSize = true;
            this.label25.BackColor = System.Drawing.Color.Red;
            this.label25.Font = new System.Drawing.Font("Microsoft Sans Serif", 10.2F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.label25.ForeColor = System.Drawing.Color.Red;
            this.label25.Location = new System.Drawing.Point(10, 240);
            this.label25.Name = "label25";
            this.label25.Size = new System.Drawing.Size(21, 20);
            this.label25.TabIndex = 21;
            this.label25.Text = "--";
            //
            // label24
            //
            this.label24.AutoSize = true;
            this.label24.Font = new System.Drawing.Font("Microsoft Sans Serif", 10.2F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.label24.Location = new System.Drawing.Point(147, 205);
            this.label24.Name = "label24";
            this.label24.Size = new System.Drawing.Size(67, 20);
            this.label24.TabIndex = 20;
            this.label24.Text = "Starting";
            //
            // label23
            //
            this.label23.AutoSize = true;
            this.label23.BackColor = System.Drawing.Color.Gold;
            this.label23.Font = new System.Drawing.Font("Microsoft Sans Serif", 10.2F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.label23.ForeColor = System.Drawing.Color.Gold;
            this.label23.Location = new System.Drawing.Point(120, 205);
            this.label23.Name = "label23";
            this.label23.Size = new System.Drawing.Size(21, 20);
            this.label23.TabIndex = 19;
            this.label23.Text = "--";
            //
            // label22
            //
            this.label22.AutoSize = true;
            this.label22.Font = new System.Drawing.Font("Microsoft Sans Serif", 10.2F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.label22.Location = new System.Drawing.Point(37, 205);
            this.label22.Name = "label22";
            this.label22.Size = new System.Drawing.Size(30, 20);
            this.label22.TabIndex = 18;
            this.label22.Text = "Up";
            //
            // label21
            //
            this.label21.AutoSize = true;
            this.label21.BackColor = System.Drawing.Color.LimeGreen;
            this.label21.Font = new System.Drawing.Font("Microsoft Sans Serif", 10.2F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.label21.ForeColor = System.Drawing.Color.LimeGreen;
            this.label21.Location = new System.Drawing.Point(10, 205);
            this.label21.Name = "label21";
            this.label21.Size = new System.Drawing.Size(21, 20);
            this.label21.TabIndex = 17;
            this.label21.Text = "--";
            //
            // label20
            //
            this.label20.AutoSize = true;
            this.label20.Font = new System.Drawing.Font("Microsoft Sans Serif", 10.2F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.label20.Location = new System.Drawing.Point(6, 180);
            this.label20.Name = "label20";
            this.label20.Size = new System.Drawing.Size(163, 20);
            this.label20.TabIndex = 16;
            this.label20.Text = "Estados del Paquete";
            //
            // label19
            //
            this.label19.AutoSize = true;
            this.label19.Font = new System.Drawing.Font("Microsoft Sans Serif", 10.2F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
```

```
            this.label19.Location = new System.Drawing.Point(227, 145);
            this.label19.Name = "label19";
            this.label19.Size = new System.Drawing.Size(77, 20);
            this.label19.TabIndex = 15;
            this.label19.Text = "Unknown";
            //
            // label18
            //
            this.label18.AutoSize = true;
            this.label18.BackColor = System.Drawing.Color.Black;
            this.label18.Font = new System.Drawing.Font("Microsoft Sans Serif", 10.2F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.label18.ForeColor = System.Drawing.Color.Black;
            this.label18.Location = new System.Drawing.Point(200, 145);
            this.label18.Name = "label18";
            this.label18.Size = new System.Drawing.Size(21, 20);
            this.label18.TabIndex = 14;
            this.label18.Text = "--";
            //
            // label17
            //
            this.label17.AutoSize = true;
            this.label17.Font = new System.Drawing.Font("Microsoft Sans Serif", 10.2F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.label17.Location = new System.Drawing.Point(127, 145);
            this.label17.Name = "label17";
            this.label17.Size = new System.Drawing.Size(52, 20);
            this.label17.TabIndex = 13;
            this.label17.Text = "Down";
            //
            // label16
            //
            this.label16.AutoSize = true;
            this.label16.BackColor = System.Drawing.Color.Red;
            this.label16.Font = new System.Drawing.Font("Microsoft Sans Serif", 10.2F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.label16.ForeColor = System.Drawing.Color.Red;
            this.label16.Location = new System.Drawing.Point(100, 145);
            this.label16.Name = "label16";
            this.label16.Size = new System.Drawing.Size(21, 20);
            this.label16.TabIndex = 12;
            this.label16.Text = "--";
            //
            // label15
            //
            this.label15.AutoSize = true;
            this.label15.Font = new System.Drawing.Font("Microsoft Sans Serif", 10.2F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.label15.Location = new System.Drawing.Point(37, 145);
            this.label15.Name = "label15";
            this.label15.Size = new System.Drawing.Size(30, 20);
            this.label15.TabIndex = 11;
            this.label15.Text = "Up";
            //
            // label14
            //
            this.label14.AutoSize = true;
            this.label14.BackColor = System.Drawing.Color.LimeGreen;
            this.label14.Font = new System.Drawing.Font("Microsoft Sans Serif", 10.2F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.label14.ForeColor = System.Drawing.Color.LimeGreen;
            this.label14.Location = new System.Drawing.Point(10, 145);
            this.label14.Name = "label14";
            this.label14.Size = new System.Drawing.Size(21, 20);
            this.label14.TabIndex = 10;
            this.label14.Text = "--";
            //
            // label13
            //
            this.label13.AutoSize = true;
            this.label13.Font = new System.Drawing.Font("Microsoft Sans Serif", 10.2F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.label13.Location = new System.Drawing.Point(6, 120);
            this.label13.Name = "label13";
            this.label13.Size = new System.Drawing.Size(201, 20);
```

```
            this.label13.TabIndex = 9;
            this.label13.Text = "Estados del (los) Nodo(s)";
            //
            // label12
            //
            this.label12.AutoSize = true;
            this.label12.Font = new System.Drawing.Font("Microsoft Sans Serif", 10.2F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.label12.Location = new System.Drawing.Point(147, 85);
            this.label12.Name = "label12";
            this.label12.Size = new System.Drawing.Size(77, 20);
            this.label12.TabIndex = 8;
            this.label12.Text = "Unknown";
            //
            // label11
            //
            this.label11.AutoSize = true;
            this.label11.BackColor = System.Drawing.Color.Black;
            this.label11.Font = new System.Drawing.Font("Microsoft Sans Serif", 10.2F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.label11.ForeColor = System.Drawing.Color.Black;
            this.label11.Location = new System.Drawing.Point(120, 85);
            this.label11.Name = "label11";
            this.label11.Size = new System.Drawing.Size(21, 20);
            this.label11.TabIndex = 7;
            this.label11.Text = "--";
            //
            // label10
            //
            this.label10.AutoSize = true;
            this.label10.Font = new System.Drawing.Font("Microsoft Sans Serif", 10.2F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.label10.Location = new System.Drawing.Point(37, 85);
            this.label10.Name = "label10";
            this.label10.Size = new System.Drawing.Size(52, 20);
            this.label10.TabIndex = 6;
            this.label10.Text = "Down";
            //
            // label9
            //
            this.label9.AutoSize = true;
            this.label9.BackColor = System.Drawing.Color.Red;
            this.label9.Font = new System.Drawing.Font("Microsoft Sans Serif", 10.2F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.label9.ForeColor = System.Drawing.Color.Red;
            this.label9.Location = new System.Drawing.Point(10, 85);
            this.label9.Name = "label9";
            this.label9.Size = new System.Drawing.Size(21, 20);
            this.label9.TabIndex = 5;
            this.label9.Text = "--";
            //
            // label8
            //
            this.label8.AutoSize = true;
            this.label8.Font = new System.Drawing.Font("Microsoft Sans Serif", 10.2F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.label8.Location = new System.Drawing.Point(147, 50);
            this.label8.Name = "label8";
            this.label8.Size = new System.Drawing.Size(67, 20);
            this.label8.TabIndex = 4;
            this.label8.Text = "Starting";
            //
            // label7
            //
            this.label7.AutoSize = true;
            this.label7.BackColor = System.Drawing.Color.Gold;
            this.label7.Font = new System.Drawing.Font("Microsoft Sans Serif", 10.2F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.label7.ForeColor = System.Drawing.Color.Gold;
            this.label7.Location = new System.Drawing.Point(120, 50);
            this.label7.Name = "label7";
            this.label7.Size = new System.Drawing.Size(21, 20);
            this.label7.TabIndex = 3;
            this.label7.Text = "--";
            //
```

```
            // label6
            //
            this.label6.AutoSize = true;
            this.label6.Font = new System.Drawing.Font("Microsoft Sans Serif", 10.2F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.label6.Location = new System.Drawing.Point(37, 50);
            this.label6.Name = "label6";
            this.label6.Size = new System.Drawing.Size(30, 20);
            this.label6.TabIndex = 2;
            this.label6.Text = "Up";
            //
            // label5
            //
            this.label5.AutoSize = true;
            this.label5.BackColor = System.Drawing.Color.LimeGreen;
            this.label5.Font = new System.Drawing.Font("Microsoft Sans Serif", 10.2F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.label5.ForeColor = System.Drawing.Color.LimeGreen;
            this.label5.Location = new System.Drawing.Point(10, 50);
            this.label5.Name = "label5";
            this.label5.Size = new System.Drawing.Size(21, 20);
            this.label5.TabIndex = 1;
            this.label5.Text = "--";
            //
            // label4
            //
            this.label4.AutoSize = true;
            this.label4.Font = new System.Drawing.Font("Microsoft Sans Serif", 10.2F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.label4.Location = new System.Drawing.Point(6, 25);
            this.label4.Name = "label4";
            this.label4.Size = new System.Drawing.Size(156, 20);
            this.label4.TabIndex = 0;
            this.label4.Text = "Estados del Cluster";
            //
            // timer1
            //
            this.timer1.Enabled = true;
            this.timer1.Interval = 1000;
            this.timer1.Tick += new System.EventHandler(this.timer1_Tick);
            //
            // Form1
            //
            this.AutoScaleDimensions = new System.Drawing.SizeF(8F, 16F);
            this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
            this.BackColor = System.Drawing.Color.WhiteSmoke;
            this.ClientSize = new System.Drawing.Size(1182, 753);
            this.Controls.Add(this.groupBox3);
            this.Controls.Add(this.groupBox2);
            this.Controls.Add(this.groupBox1);
            this.Controls.Add(this.statusStrip1);
            this.Controls.Add(this.toolStrip1);
            this.Controls.Add(this.menuStrip1);
            this.FormBorderStyle = System.Windows.Forms.FormBorderStyle.FixedSingle;
            this.HelpButton = true;
            this.Icon = ((System.Drawing.Icon)(resources.GetObject("$this.Icon")));
            this.MainMenuStrip = this.menuStrip1;
            this.MaximizeBox = false;
            this.Name = "Form1";
            this.Text = "Monitor de Cluster";
            this.menuStrip1.ResumeLayout(false);
            this.menuStrip1.PerformLayout();
            this.toolStrip1.ResumeLayout(false);
            this.toolStrip1.PerformLayout();
            this.statusStrip1.ResumeLayout(false);
            this.statusStrip1.PerformLayout();
            this.groupBox1.ResumeLayout(false);
            this.groupBox2.ResumeLayout(false);
            this.groupBox2.PerformLayout();
            this.groupBox3.ResumeLayout(false);
            this.groupBox3.PerformLayout();
            this.ResumeLayout(false);
            this.PerformLayout();

        }
```

```
        #endregion

        public System.Windows.Forms.MenuStrip menuStrip1;
        public System.Windows.Forms.ToolStripMenuItem archivoToolStripMenuItem;
        public System.Windows.Forms.ToolStripMenuItem guardarToolStripMenuItem;
        public System.Windows.Forms.ToolStripMenuItem salirToolStripMenuItem;
        public System.Windows.Forms.ToolStripMenuItem accionesToolStripMenuItem;
        public System.Windows.Forms.ToolStripMenuItem conectarToolStripMenuItem;
        public System.Windows.Forms.ToolStripMenuItem actualizarToolStripMenuItem;
        public System.Windows.Forms.ToolStripMenuItem propiedadesToolStripMenuItem;
        public System.Windows.Forms.ToolStripMenuItem ayudaToolStripMenuItem;
        public System.Windows.Forms.ToolStripMenuItem verToolStripMenuItem;
        public System.Windows.Forms.ToolStripMenuItem acercaToolStripMenuItem;
        public System.Windows.Forms.ToolStrip toolStrip1;
        public System.Windows.Forms.ToolStripButton toolStripButton1;
        public System.Windows.Forms.ToolStripButton toolStripButton2;
        public System.Windows.Forms.ToolStripButton toolStripButton3;
        public System.Windows.Forms.ToolStripButton toolStripButton4;
        public System.Windows.Forms.StatusStrip statusStrip1;
        public System.Windows.Forms.ToolStripProgressBar toolStripProgressBar1;
        public System.Windows.Forms.ToolStripStatusLabel toolStripStatusLabel1;
        public System.Windows.Forms.ToolStripStatusLabel toolStripStatusLabel2;
        public System.Windows.Forms.ToolStripStatusLabel toolStripStatusLabel3;
        public System.Windows.Forms.GroupBox groupBox1;
        public System.Windows.Forms.Button button1;
        public System.Windows.Forms.Button button2;
        public System.Windows.Forms.Button button3;
        public System.Windows.Forms.Button button4;
        public System.Windows.Forms.GroupBox groupBox2;
        public System.Windows.Forms.Label label1;
        public System.Windows.Forms.TextBox textBox1;
        public System.Windows.Forms.Label label2;
        public System.Windows.Forms.TextBox textBox2;
        public System.Windows.Forms.Label label3;
        public System.Windows.Forms.TextBox textBox3;
        public System.Windows.Forms.CheckBox checkBox1;
        public System.Windows.Forms.Button button5;
        public System.Windows.Forms.GroupBox groupBox3;
        private System.Windows.Forms.Label label4;
        private System.Windows.Forms.Label label5;
        private System.Windows.Forms.Label label6;
        private System.Windows.Forms.Label label7;
        private System.Windows.Forms.Label label8;
        private System.Windows.Forms.Label label9;
        private System.Windows.Forms.Label label10;
        private System.Windows.Forms.Label label11;
        private System.Windows.Forms.Label label12;
        private System.Windows.Forms.Label label13;
        private System.Windows.Forms.Label label14;
        private System.Windows.Forms.Label label15;
        private System.Windows.Forms.Label label16;
        private System.Windows.Forms.Label label17;
        private System.Windows.Forms.Label label18;
        private System.Windows.Forms.Label label19;
        private System.Windows.Forms.Label label20;
        private System.Windows.Forms.Label label21;
        private System.Windows.Forms.Label label22;
        private System.Windows.Forms.Label label23;
        private System.Windows.Forms.Label label24;
        private System.Windows.Forms.Label label25;
        private System.Windows.Forms.Label label26;
        private System.Windows.Forms.Label label27;
        private System.Windows.Forms.Label label28;
        private System.Windows.Forms.Label label29;
        private System.Windows.Forms.Label label30;
        public System.Windows.Forms.Timer timer1;
    }
}
```

Archivos fuente correspondientes al módulo de comunicación y estado.

CheckHostname.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.IO;
using Renci.SshNet;

namespace ClusterGUI
{
    class CheckHostname
    {
        public static string GetHostname(string host, string user, string password)
        {
            string hostname = string.Empty;

            try
            {
                using (var client = new SshClient(host, user, password))
                {
                    client.Connect();
                    var output = client.RunCommand("hostname");
                    client.Disconnect();
                    hostname = output.Result;
                }

                hostname = hostname.Replace("\n", string.Empty);
                hostname = hostname.Replace("\r", string.Empty);
                hostname = hostname.Replace("\t", string.Empty);
                hostname = hostname.Replace(" ", string.Empty);
            }
            catch (Exception e)
            {
                System.Windows.Forms.MessageBox.Show("Ha ocurrido un ERROR de Conexión."
                    + "\nNo fue posible verificar el estado del Nodo o Nodos, y/o Paquete.");
                System.Windows.Forms.MessageBox.Show(e.Message);
                hostname = "null";
            }

            return hostname;
        }
    }
}
```

CheckPackage.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.IO;
using Renci.SshNet;

namespace ClusterGUI
{
    class CheckPackage
    {
        public static int GetNodePackage(string host, string user, string password, string namepackage, string namenode1,
string namenode2)
        {
            string nNode1 = namenode1;
            string nNode2 = namenode2;
            string nPackage = namepackage;
            string result = string.Empty;
            int index = 0;

            try
            {
```

```
                using (var client = new SshClient(host, user, password))
                {
                    client.Connect();
                    var output = client.RunCommand("cmviewcl -p " + nPackage);
                    client.Disconnect();
                    result = output.Result;
                }

                result = result.Replace("\n", string.Empty);
                result = result.Replace("\r", string.Empty);
                result = result.Replace("\t", string.Empty);
                result = result.Replace(" ", string.Empty);
            }
            catch (Exception e)
            {
                System.Windows.Forms.MessageBox.Show("Ha ocurrido un ERROR de Conexión."
                    + "\nNo fue posible verificar el estado del Paquete.");
                System.Windows.Forms.MessageBox.Show(e.Message);
                index = -1;
            }

            if (index != -1)
            {
                if (result.Contains(nNode1) == true)
                    index = 1;
                else if (result.Contains(nNode2) == true)
                    index = 2;
                else
                    index = -1;
            }

            return index;
        }
    }
}
```

CheckStatusC.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.IO;
using Renci.SshNet;

namespace ClusterGUI
{
    class CheckStatusC
    {
        public static string FindStrings(string result, string namecluster)
        {
            string r = result;
            string nCluster = namecluster;
            string statusCluster = string.Empty;

            if (r.Contains(nCluster + "up") == true)
                statusCluster = "up";
            else if (r.Contains(nCluster + "down") == true)
                statusCluster = "down";
            else if (r.Contains(nCluster + "starting") == true)
                statusCluster = "starting";
            else if (r.Contains(nCluster + "unknown") == true)
                statusCluster = "unknown";

            return statusCluster;
        }
    }
}
```

CheckStatusN.cs

```
using System;
```

```
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.IO;
using Renci.SshNet;

namespace ClusterGUI
{
    class CheckStatusN
    {
        public static string FindStrings(string result, string namenode)
        {
            string r = result;
            string nNode = namenode;
            string statusNode = string.Empty;

            if (r.Contains(nNode + "up") == true)
                statusNode = "up";
            else if (r.Contains(nNode + "down") == true)
                statusNode = "down";

            return statusNode;
        }
    }
}
```

CheckStatusP.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.IO;
using Renci.SshNet;

namespace ClusterGUI
{
    class CheckStatusP
    {
        public static string FindStrings(string result, string namepackage)
        {
            string r = result;
            string nPackage = namepackage;
            string statusPackage = string.Empty;

            if (r.Contains(nPackage + "up") == true)
                statusPackage = "up";
            else if (r.Contains(nPackage + "down") == true)
                statusPackage = "down";
            else if (r.Contains(nPackage + "unknown") == true | r.Contains(nPackage + "detached") == true)
                statusPackage = "unknown";
            else
                statusPackage = "start";

            return statusPackage;
        }
    }
}
```

Connection.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.IO;
using Renci.SshNet;

namespace ClusterGUI
{
```

```
    class Connection
    {
        public static string FirstConnection(string host, string user, string password)
        {
            string result = string.Empty;

            try
            {
                using (var client = new SshClient(host, user, password))
                {
                    client.Connect();
                    var output = client.RunCommand("cmviewcl");
                    client.Disconnect();
                    result = output.Result;
                }
            }
            catch (Exception e)
            {
                System.Windows.Forms.MessageBox.Show("Ha ocurrido un ERROR de Conexión."
                    + "\nLos Datos de Conexión pueden ser incorrectos. Por favor, verificalos."
                    + "\nSi los Datos de Conexión son correctos, es un problema de Red. Intenta más tarde.");
                System.Windows.Forms.MessageBox.Show(e.Message);
                result = "null";
            }

            return result;
        }
    }
}
```

Form1.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Diagnostics;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.IO;
using System.Threading;
using Renci.SshNet;

namespace ClusterGUI
{
    public partial class Form1 : Form
    {
        string Host = string.Empty;
        string User = string.Empty;
        string Password = string.Empty;
        string Result = string.Empty;
        string Properties = string.Empty;
        string NameCluster = string.Empty;
        string Hostname = string.Empty;
        string NameNode1 = string.Empty;
        string NameNode2 = string.Empty;
        string NamePackage = string.Empty;
        string StatusCluster = string.Empty;
        string StatusNode1 = string.Empty;
        string StatusNode2 = string.Empty;
        string StatusPackage = string.Empty;
        string note = string.Empty;
        const string Header1 = "CLUSTERSTATUS";
        const string Header2 = "NODESTATUSSTATE";
        const string Header3 = "PACKAGESTATUSSTATEAUTO_RUNNODE";
        const string PathDirectory = @"C:\Users\Public\InterfazGraficaCluster";
        const string PathFile = @"C:\Users\Public\InterfazGraficaCluster\EstadoCluster.txt";
        Stopwatch sw = new Stopwatch();

        public Form1()
        {
```

```
            InitializeComponent();
        }

        public void Connect()
        {
            Host = this.textBox1.Text;
            User = this.textBox2.Text;
            Password = this.textBox3.Text;

            if (Host == string.Empty | User == string.Empty | Password == string.Empty)
            {
                System.Windows.Forms.MessageBox.Show("Por favor, escribe correctamente los Datos de Conexión."
                    + "\nEl formato del Host es: 148.206.71.50 "
                    + "\nUser y Password distinguen MAYÚSCULAS y minúsculas. Verifica los datos antes de presionar
'Conectar'.");
            }
            else
            {
                ///*
                Result = Connection.FirstConnection(Host, User, Password);

                if (Result != "null")
                {
                    Result = Result.Replace("\n", string.Empty);
                    Result = Result.Replace("\r", string.Empty);
                    Result = Result.Replace("\t", string.Empty);
                    Result = Result.Replace(" ", string.Empty);

                    if (Result.Contains(Header1) == true & Result.Contains(Header2) == true)
                    {
                        NameCluster = NameC.NameCluster(Result, Header1);
                        this.button1.Text = NameCluster;
                        StatusCluster = CheckStatusC.FindStrings(Result, NameCluster);
                        if (StatusCluster == "up")
                            this.button1.FlatAppearance.BorderColor = System.Drawing.Color.LimeGreen;
                        else if (StatusCluster == "down")
                            this.button1.FlatAppearance.BorderColor = System.Drawing.Color.Red;
                        else if (StatusCluster == "starting")
                            this.button1.FlatAppearance.BorderColor = System.Drawing.Color.Gold;
                        else if (StatusCluster == "unknown")
                            this.button1.FlatAppearance.BorderColor = System.Drawing.Color.Black;

                        Hostname = CheckHostname.GetHostname(Host, User, Password);
                        if (Hostname == "null")
                        {
                            this.button1.Text = string.Empty;
                            this.button2.Text = string.Empty;
                            this.button3.Text = string.Empty;
                            this.button4.Text = string.Empty;
                            this.button1.FlatAppearance.BorderColor = System.Drawing.Color.Black;
                            this.button2.FlatAppearance.BorderColor = System.Drawing.Color.Black;
                            this.button3.FlatAppearance.BorderColor = System.Drawing.Color.Black;
                            this.button4.FlatAppearance.BorderColor = System.Drawing.Color.Black;
                            this.toolStripStatusLabel1.Text = "Conexión interrumpida... ... ...";
                            this.toolStripStatusLabel2.Text = "Desconocido";

                            sw.Stop();
                            timer1.Stop();
                        }
                        else
                        {
                            if (Result.Contains(Header1 + NameCluster + StatusCluster + Header2 + Hostname) == true)
                            {
                                NameNode1 = Hostname;
                                this.button2.Text = NameNode1;
                                StatusNode1 = CheckStatusN.FindStrings(Result, NameNode1);
                                if (StatusNode1 == "up")
                                    this.button2.FlatAppearance.BorderColor = System.Drawing.Color.LimeGreen;
                                else if (StatusNode1 == "down")
                                    this.button2.FlatAppearance.BorderColor = System.Drawing.Color.Red;

                                NameNode2 = NameN02.NameNode02(Host, User, Password, Result, NameNode1, Header2, Header3);

                                if (NameNode2 != "" & NameNode2 != string.Empty)
                                {
```

```
                                this.button3.Text = NameNode2;
                                StatusNode2 = CheckStatusN.FindStrings(Result, NameNode2);
                                if (StatusNode2 == "up")
                                    this.button3.FlatAppearance.BorderColor = System.Drawing.Color.LimeGreen;
                                else if (StatusNode2 == "down")
                                    this.button3.FlatAppearance.BorderColor = System.Drawing.Color.Red;

                                note = "2 Nodos.";
                            }
                            else
                                note = "1 Nodo.";
                        }
                        else
                        {
                            NameNode2 = Hostname;
                            this.button3.Text = NameNode2;
                            StatusNode2 = CheckStatusN.FindStrings(Result, NameNode2);
                            if (StatusNode2 == "up")
                                this.button3.FlatAppearance.BorderColor = System.Drawing.Color.LimeGreen;
                            else if (StatusNode2 == "down")
                                this.button3.FlatAppearance.BorderColor = System.Drawing.Color.Red;

                            NameNode1 = NameN01.NameNode01(Host, User, Password, Result, NameNode2, Header1,
NameCluster, StatusCluster, Header2, Header3);

                            if (NameNode1 != "" & NameNode1 != string.Empty)
                            {
                                this.button2.Text = NameNode1;
                                StatusNode1 = CheckStatusN.FindStrings(Result, NameNode1);
                                if (StatusNode1 == "up")
                                    this.button2.FlatAppearance.BorderColor = System.Drawing.Color.LimeGreen;
                                else if (StatusNode1 == "down")
                                    this.button2.FlatAppearance.BorderColor = System.Drawing.Color.Red;

                                note = "2 Nodos.";
                            }
                            else
                                note = "1 Nodo.";
                        }

                        if (Result.Contains(Header3) == true)
                        {
                            NamePackage = NameP.NamePackage(Result, Header3, Header2, NameNode1, NameNode2);
                            this.button4.Text = NamePackage;
                            StatusPackage = CheckStatusP.FindStrings(Result, NamePackage);
                            if (StatusPackage == "up")
                                this.button4.FlatAppearance.BorderColor = System.Drawing.Color.LimeGreen;
                            else if (StatusPackage == "down")
                                this.button4.FlatAppearance.BorderColor = System.Drawing.Color.Red;
                            else if (StatusPackage == "start")
                                this.button4.FlatAppearance.BorderColor = System.Drawing.Color.Gold;
                            else if (StatusPackage == "unknown")
                                this.button4.FlatAppearance.BorderColor = System.Drawing.Color.Black;

                            if (CheckPackage.GetNodePackage(Host, User, Password, NamePackage, NameNode1, NameNode2)
== 1)
                                this.toolStripStatusLabel1.Text = "¡Conexión Exitosa! " + note + " 1 Paquete en
ejecución en: " + NameNode1 + ".";
                            else if (CheckPackage.GetNodePackage(Host, User, Password, NamePackage, NameNode1,
NameNode2) == 2)
                                this.toolStripStatusLabel1.Text = "¡Conexión Exitosa! " + note + " 1 Paquete en
ejecución en: " + NameNode2 + ".";
                            else if (CheckPackage.GetNodePackage(Host, User, Password, NamePackage, NameNode1,
NameNode2) == -1)
                            {
                                this.button4.Text = string.Empty;
                                this.button4.FlatAppearance.BorderColor = System.Drawing.Color.Gray;
                                this.toolStripStatusLabel1.Text = "¡Conexión Exitosa! " + note + " 0 Paquetes.";
                            }

                            this.toolStripStatusLabel2.Text = "Conectado";

                            timer1.Start();
                            sw.Start();
                        }
```

```
                        else
                        {
                            this.toolStripStatusLabel1.Text = "¡Conexión Exitosa! " + note + " 0 Paquetes.";
                            this.toolStripStatusLabel2.Text = "Conectado";

                            timer1.Start();
                            sw.Start();
                        }
                    }
                }
                else
                {
                    System.Windows.Forms.MessageBox.Show("Ha ocurrido un ERROR."
                        + "\nLa conexión fue exitosa, pero no fue posible verificar el estado del Clúster, Nodo(s) y/o
Paquete."
                        + "\nIntenta más tarde.");
                    this.button1.Text = string.Empty;
                    this.button2.Text = string.Empty;
                    this.button3.Text = string.Empty;
                    this.button4.Text = string.Empty;
                    this.button1.FlatAppearance.BorderColor = System.Drawing.Color.Black;
                    this.button2.FlatAppearance.BorderColor = System.Drawing.Color.Black;
                    this.button3.FlatAppearance.BorderColor = System.Drawing.Color.Black;
                    this.button4.FlatAppearance.BorderColor = System.Drawing.Color.Black;
                    this.toolStripStatusLabel1.Text = "¡Conexión Exitosa!";
                    this.toolStripStatusLabel2.Text = "Desconocido";

                    sw.Stop();
                    timer1.Stop();
                }
            }
            else
            {
                this.button1.Text = string.Empty;
                this.button2.Text = string.Empty;
                this.button3.Text = string.Empty;
                this.button4.Text = string.Empty;
                this.button1.FlatAppearance.BorderColor = System.Drawing.Color.Gray;
                this.button2.FlatAppearance.BorderColor = System.Drawing.Color.Gray;
                this.button3.FlatAppearance.BorderColor = System.Drawing.Color.Gray;
                this.button4.FlatAppearance.BorderColor = System.Drawing.Color.Gray;
                this.toolStripStatusLabel1.Text = "¡¡¡ Error de Conexión !!!";
                this.toolStripStatusLabel2.Text = "No conectado";

                sw.Stop();
                timer1.Stop();
            }
            //*/
        }
    }

    public void Upgrade()
    {
        if (Result == string.Empty | Result == "null")
            System.Windows.Forms.MessageBox.Show("¡ ERROR !"
                + "\nNO has establecido conexión con ningún Cluster."
                + "\nPara actualizar el Estado del Cluster, primero conéctate a él.");
        else
        {
            this.button1.Text = string.Empty;
            this.button2.Text = string.Empty;
            this.button3.Text = string.Empty;
            this.button4.Text = string.Empty;
            this.button1.FlatAppearance.BorderColor = System.Drawing.Color.Gray;
            this.button2.FlatAppearance.BorderColor = System.Drawing.Color.Gray;
            this.button3.FlatAppearance.BorderColor = System.Drawing.Color.Gray;
            this.button4.FlatAppearance.BorderColor = System.Drawing.Color.Gray;
            this.toolStripStatusLabel1.Text = "- - -";
            this.toolStripStatusLabel2.Text = "- - -";

            Connect();
        }
    }

    public void Qualities()
```

```
{
    if (Result == string.Empty | Result == "null")
        System.Windows.Forms.MessageBox.Show("¡ ERROR !"
            + "\nNO has establecido conexión con ningún Cluster."
            + "\nPara mostrar en pantalla las Propiedades de un Cluster, primero conéctate a él.");
    else
    {
        Properties = GetProperties.SecondConnection(Host, User, Password);

        if (Properties != "null")
        {
            if (Properties != string.Empty)
                System.Windows.Forms.MessageBox.Show(Properties);
            else
                System.Windows.Forms.MessageBox.Show("Ha ocurrido un ERROR."
                    + "\nLa conexión fue exitosa, pero no fue posible verificar las Propiedades del Cluster."
                    + "\nIntenta más tarde.");
        }
    }
}

private void checkBox1_CheckedChanged(object sender, EventArgs e)
{
    this.textBox3.UseSystemPasswordChar = !checkBox1.Checked;
}

private void button5_Click(object sender, EventArgs e)
{
    Host = this.textBox1.Text;
    User = this.textBox2.Text;
    Password = this.textBox3.Text;

    if (Host == string.Empty | User == string.Empty | Password == string.Empty)
    {
        System.Windows.Forms.MessageBox.Show("¡ ERROR !"
            + "\nNO has establecido conexión con ningún Cluster.");

        Connect();
    }
    else
        Connect();
}

private void toolStripButton1_Click(object sender, EventArgs e)
{
    Host = this.textBox1.Text;
    User = this.textBox2.Text;
    Password = this.textBox3.Text;

    if (Host == string.Empty | User == string.Empty | Password == string.Empty)
    {
        System.Windows.Forms.MessageBox.Show("¡ ERROR !"
            + "\nNO has establecido conexión con ningún Cluster.");

        Connect();
    }
    else
        Connect();
}

private void toolStripButton2_Click(object sender, EventArgs e)
{
    Upgrade();
}

private void toolStripButton3_Click(object sender, EventArgs e)
{
    Qualities();
}

private void toolStripButton4_Click(object sender, EventArgs e)
{
    Help.ShowHelp(this, "file://C:\\Users\\Public\\InterfazGraficaCluster\\Ayuda_Cluster_GUI.chm");
}
```

```
private void guardarToolStripMenuItem_Click(object sender, EventArgs e)
{
    if (Result == string.Empty | Result == "null")
        System.Windows.Forms.MessageBox.Show("¡ ERROR !"
            + "\nNO has establecido conexión con ningún Cluster."
            + "\nPara guardar las Propiedades de un Cluster, primero conéctate a él.");
    else
    {
        try
        {
            if (!Directory.Exists(PathDirectory))
            {
                DirectoryInfo DI = Directory.CreateDirectory(PathDirectory);
            }
        }
        catch (Exception ex)
        {
            System.Windows.Forms.MessageBox.Show("¡ ERROR !"
                + "\nLa carpeta 'InterfazGraficaCluster' no pudo ser creada.");
            System.Windows.Forms.MessageBox.Show(ex.Message);
        }

        if (!File.Exists(PathFile))
        {
            using (FileStream FS = File.Create(PathFile))
            {
                FS.Close();
            }
        }

        Properties = GetProperties.SecondConnection(Host, User, Password);

        if (Properties != "null")
        {
            if (Properties != string.Empty)
            {
                System.IO.File.WriteAllText(PathFile, Properties);
                System.Windows.Forms.MessageBox.Show("¡ Archivo de propiedades creado exitosamente !"
                    + "\nLa ruta del archivo es: " + PathFile);
            }
            else
                System.Windows.Forms.MessageBox.Show("Ha ocurrido un ERROR."
                    + "\nLa conexión fue exitosa, pero no fue posible verificar las Propiedades del Cluster."
                    + "\nIntenta más tarde.");
        }

    }
}

private void salirToolStripMenuItem_Click(object sender, EventArgs e)
{
    Application.Exit();
}

private void conectarToolStripMenuItem_Click(object sender, EventArgs e)
{
    Host = this.textBox1.Text;
    User = this.textBox2.Text;
    Password = this.textBox3.Text;

    if (Host == string.Empty | User == string.Empty | Password == string.Empty)
    {
        System.Windows.Forms.MessageBox.Show("¡ ERROR !"
            + "\nNO has establecido conexión con ningún Cluster.");

        Connect();
    }
    else
        Connect();
}

private void actualizarToolStripMenuItem_Click(object sender, EventArgs e)
{
    Upgrade();
}
```

```
private void propiedadesToolStripMenuItem_Click(object sender, EventArgs e)
{
    Qualities();
}

private void verToolStripMenuItem_Click(object sender, EventArgs e)
{
    Help.ShowHelp(this, "file://C:\\Users\\Public\\InterfazGraficaCluster\\Ayuda_Cluster_GUI.chm");
}

private void acercaToolStripMenuItem_Click(object sender, EventArgs e)
{
    System.Windows.Forms.MessageBox.Show("Monitor de Cluster v0.1.0"
        + "\nUniversidad Autónoma Metropolitana - Azcapotzalco"
        + "\nDivisión de Ciencias Básicas e Ingeniería"
        + "\nLicenciatura en Ingeniería en Computación"
        + "\nAlumno: David Contreras Tovar."
        + "\nAsesor: M. en C. José Ignacio Vega Luna.");
}

private void timer1_Tick(object sender, EventArgs e)
{
    this.toolStripStatusLabel3.Text = String.Format("{0:00}:{1:00}:{2:00}",
        sw.Elapsed.Hours, sw.Elapsed.Minutes, sw.Elapsed.Seconds);
}

private void button1_Click(object sender, EventArgs e)
{
    string tempC = string.Empty;

    if (NameCluster != string.Empty)
    {
        tempC = GetInfo.GetData(Host, User, Password, "cmviewcl -c " + NameCluster);

        if (tempC != "null")
        {
            if (tempC != string.Empty)
                System.Windows.Forms.MessageBox.Show(tempC);
            else
                System.Windows.Forms.MessageBox.Show("Ha ocurrido un ERROR."
                    + "\nLa conexión fue exitosa, pero no fue posible obtener la información requerida."
                    + "\nIntenta más tarde.");
        }
    }
    else
        System.Windows.Forms.MessageBox.Show("¡ ERROR !"
            + "\nNO has establecido conexión con ningún Cluster.");
}

private void button2_Click(object sender, EventArgs e)
{
    string tempN = string.Empty;

    if (NameNode1 != string.Empty)
    {
        tempN = GetInfo.GetData(Host, User, Password, "cmviewcl -n " + NameNode1);

        if (tempN != "null")
        {
            if (tempN != string.Empty)
                System.Windows.Forms.MessageBox.Show(tempN);
            else
                System.Windows.Forms.MessageBox.Show("Ha ocurrido un ERROR."
                    + "\nLa conexión fue exitosa, pero no fue posible obtener la información requerida."
                    + "\nIntenta más tarde.");
        }
    }
    else
        System.Windows.Forms.MessageBox.Show("¡ ERROR !"
            + "\nNO has establecido conexión con ningún Cluster.");
}

private void button3_Click(object sender, EventArgs e)
{
```

```
            string tempN = string.Empty;

            if (NameNode2 != string.Empty)
            {
                tempN = GetInfo.GetData(Host, User, Password, "cmviewcl -n " + NameNode2);

                if (tempN != "null")
                {
                    if (tempN != string.Empty)
                        System.Windows.Forms.MessageBox.Show(tempN);
                    else
                        System.Windows.Forms.MessageBox.Show("Ha ocurrido un ERROR."
                            + "\nLa conexión fue exitosa, pero no fue posible obtener la información requerida."
                            + "\nIntenta más tarde.");
                }
            }
            else
                System.Windows.Forms.MessageBox.Show("¡ ERROR !"
                    + "\nNO has establecido conexión con ningún Cluster."
                    + "\nO el Cluster seleccionado solo tiene 1 Nodo activo.");
        }

        private void button4_Click(object sender, EventArgs e)
        {
            string tempP = string.Empty;

            if (NamePackage != string.Empty)
            {
                tempP = GetInfo.GetData(Host, User, Password, "cmviewcl -p " + NamePackage);

                if (tempP != "null")
                {
                    if (tempP != string.Empty)
                        System.Windows.Forms.MessageBox.Show(tempP);
                    else
                        System.Windows.Forms.MessageBox.Show("Ha ocurrido un ERROR."
                            + "\nLa conexión fue exitosa, pero no fue posible obtener la información requerida."
                            + "\nIntenta más tarde.");
                }
            }
            else
                System.Windows.Forms.MessageBox.Show("¡ ERROR !"
                    + "\nNO has establecido conexión con ningún Cluster."
                    + "\nO el Cluster seleccionado NO tiene un Paquete activo.");
        }
    }
}
```

GetInfo.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using Renci.SshNet;

namespace ClusterGUI
{
    class GetInfo
    {
        public static string GetData(string host, string user, string password, string command)
        {
            string result = string.Empty;

            try
            {
                using (var client = new SshClient(host, user, password))
                {
                    client.Connect();
                    var output = client.RunCommand(command);
                    client.Disconnect();
                    result = output.Result;
                }
```

```
            }
            catch (Exception e)
            {
                System.Windows.Forms.MessageBox.Show("Ha ocurrido un ERROR de Conexión."
                    + "\nLa causa de este error es un problema de Red."
                    + "\nNo es posible en este momento mostrar la información requerida. Intenta más tarde.");
                System.Windows.Forms.MessageBox.Show(e.Message);
                result = "null";
            }

            return result;
        }
    }
}
```

GetProperties.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using Renci.SshNet;

namespace ClusterGUI
{
    class GetProperties
    {
        public static string SecondConnection(string host, string user, string password)
        {
            string result = string.Empty;

            try
            {
                using (var client = new SshClient(host, user, password))
                {
                    client.Connect();
                    var output = client.RunCommand("cmviewcl -v");
                    client.Disconnect();
                    result = output.Result;
                }
            }
            catch (Exception e)
            {
                System.Windows.Forms.MessageBox.Show("Ha ocurrido un ERROR de Conexión."
                    + "\nLa causa de este error es un problema de Red."
                    + "\nNo es posible en este momento mostrar las Propiedades del Cluster. Intenta más tarde.");
                System.Windows.Forms.MessageBox.Show(e.Message);
                result = "null";
            }

            return result;
        }
    }
}
```

NameC.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.IO;

namespace ClusterGUI
{
    class NameC
    {
        public static string NameCluster (string result, string header1)
        {
            string r = result;
            string firstString = header1;
```

```
            const string lastString1 = "upNODE";
            const string lastString2 = "downNODE";
            const string lastString3 = "startingNODE";
            const string lastString4 = "unknownNODE";
            string nameCluster = string.Empty;
            int option = 0, pos1 = 0, pos2 = 0;

            if (r.Contains(lastString1) == true)
                option = 1;
            else if (r.Contains(lastString2) == true)
                option = 2;
            else if (r.Contains(lastString3) == true)
                option = 3;
            else if (r.Contains(lastString4) == true)
                option = 4;

            switch (option)
            {
                case 1:
                    pos1 = r.IndexOf(firstString) + firstString.Length;
                    pos2 = r.IndexOf(lastString1);

                    nameCluster = r.Substring(pos1, pos2 - pos1);
                    break;
                case 2:
                    pos1 = r.IndexOf(firstString) + firstString.Length;
                    pos2 = r.IndexOf(lastString2);

                    nameCluster = r.Substring(pos1, pos2 - pos1);
                    break;
                case 3:
                    pos1 = r.IndexOf(firstString) + firstString.Length;
                    pos2 = r.IndexOf(lastString3);

                    nameCluster = r.Substring(pos1, pos2 - pos1);
                    break;
                case 4:
                    pos1 = r.IndexOf(firstString) + firstString.Length;
                    pos2 = r.IndexOf(lastString4);

                    nameCluster = r.Substring(pos1, pos2 - pos1);
                    break;
            }

            return nameCluster;
        }
    }
}
```

NameN01.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.IO;
using Renci.SshNet;

namespace ClusterGUI
{
    class NameN01
    {
        public static string NameNode01(string host, string user, string password, string result, string namenode2, string
h1, string namecluster, string statuscluster, string h2, string h3)
        {
            string r = result;
            string namenode1 = string.Empty;
            string nNode2 = namenode2;
            int pos1 = 0;
            int pos2 = 0;
            string temp = string.Empty;
            string header1 = h1;
            string header2 = h2;
```

```
            string nCluster = namecluster;
            string sCluster = statuscluster;
            string firststring = h1 + nCluster + sCluster + h2;
            string laststring1 = h3;
            string laststring2 = string.Empty;
            const string status1 = "up";
            const string status2 = "down";

            try
            {
                using (var client = new SshClient(host, user, password))
                {
                    client.Connect();
                    var output = client.RunCommand("cmviewcl -n " + nNode2);
                    client.Disconnect();
                    temp = output.Result;
                }

                temp = temp.Replace("\n", string.Empty);
                temp = temp.Replace("\r", string.Empty);
                temp = temp.Replace("\t", string.Empty);
                temp = temp.Replace(" ", string.Empty);
            }
            catch (Exception e)
            {
                System.Windows.Forms.MessageBox.Show("Ha ocurrido un ERROR de Conexión."
                    + "\nNo fue posible verificar el estado del Nodo 1.");
                System.Windows.Forms.MessageBox.Show(e.Message);
                temp = "null";
            }

            if (temp != "null")
            {
                pos1 = temp.IndexOf(header2) + header2.Length;
                pos2 = temp.Length;
                laststring2 = temp.Substring(pos1, pos2 - pos1);

                if (r.Contains(laststring1) == true)
                {
                    pos1 = r.IndexOf(firststring) + firststring.Length;
                    pos2 = r.IndexOf(laststring1);
                    temp = r.Substring(pos1, pos2 - pos1);
                }
                else if (r.Contains(laststring2) == true)
                {
                    pos1 = r.IndexOf(firststring) + firststring.Length;
                    pos2 = r.IndexOf(laststring2);
                    temp = r.Substring(pos1, pos2 - pos1);
                }

                if (temp.Contains(status1))
                {
                    pos1 = 0;
                    pos2 = temp.IndexOf(status1);
                    namenode1 = temp.Substring(pos1, pos2 - pos1);
                }
                else if (temp.Contains(status2))
                {
                    pos1 = 0;
                    pos2 = temp.IndexOf(status2);
                    namenode1 = temp.Substring(pos1, pos2 - pos1);
                }
            }
            else
                namenode1 = string.Empty;

            return namenode1;
        }
    }
}
```

NameN02.cs

```
using System;
```

```
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.IO;
using Renci.SshNet;

namespace ClusterGUI
{
    class NameN02
    {
        public static string NameNode02(string host, string user, string password, string result, string namenode1, string
h1, string h2)
        {
            string r = result;
            string nNode1 = namenode1;
            string namenode2 = string.Empty;
            int pos1 = 0;
            int pos2 = 0;
            string temp = string.Empty;
            string header1 = h1;
            string header2 = h2;
            string firststring1 = nNode1 + header1;
            string firststring2 = string.Empty;
            const string laststring1 = "up";
            const string laststring2 = "down";

            try
            {
                using (var client = new SshClient(host, user, password))
                {
                    client.Connect();
                    var output = client.RunCommand("cmviewcl -n " + nNode1);
                    client.Disconnect();
                    temp = output.Result;
                }

                temp = temp.Replace("\n", string.Empty);
                temp = temp.Replace("\r", string.Empty);
                temp = temp.Replace("\t", string.Empty);
                temp = temp.Replace(" ", string.Empty);
            }
            catch (Exception e)
            {
                System.Windows.Forms.MessageBox.Show("Ha ocurrido un ERROR de Conexión."
                    + "\nNo fue posible verificar el estado del Nodo 2.");
                System.Windows.Forms.MessageBox.Show(e.Message);
                temp = "null";
            }

            if (temp != "null")
            {
                pos1 = temp.IndexOf(header1) + header1.Length;
                pos2 = temp.Length;
                firststring2 = temp.Substring(pos1, pos2 - pos1);

                if (r.Contains(firststring1) == true)
                {
                    pos1 = r.IndexOf(firststring1) + firststring1.Length;
                    pos2 = r.Length;
                    temp = r.Substring(pos1, pos2 - pos1);
                    if (temp.Contains(laststring1) == true)
                    {
                        pos1 = 0;
                        pos2 = temp.IndexOf(laststring1);
                        namenode2 = temp.Substring(pos1, pos2 - pos1);
                    }
                    else if (temp.Contains(laststring2) == true)
                    {
                        pos1 = 0;
                        pos2 = temp.IndexOf(laststring2);
                        namenode2 = temp.Substring(pos1, pos2 - pos1);
                    }
                }
                else if (r.Contains(firststring2) == true)
```

```
                {
                    if (r.Contains(header2) == true)
                    {
                        pos1 = r.IndexOf(firststring2) + firststring2.Length;
                        pos2 = r.IndexOf(header2);
                        temp = r.Substring(pos1, pos2 - pos1);
                        if (temp.Contains(laststring1) == true)
                        {
                            pos1 = 0;
                            pos2 = temp.IndexOf(laststring1);
                            namenode2 = temp.Substring(pos1, pos2 - pos1);
                        }
                        else if (temp.Contains(laststring2) == true)
                        {
                            pos1 = 0;
                            pos2 = temp.IndexOf(laststring2);
                            namenode2 = temp.Substring(pos1, pos2 - pos1);
                        }
                    }
                    else
                    {
                        pos1 = r.IndexOf(firststring2) + firststring2.Length;
                        pos2 = r.Length;
                        temp = r.Substring(pos1, pos2 - pos1);
                        if (temp.Contains(laststring1) == true)
                        {
                            pos1 = 0;
                            pos2 = temp.IndexOf(laststring1);
                            namenode2 = temp.Substring(pos1, pos2 - pos1);
                        }
                        else if (temp.Contains(laststring2) == true)
                        {
                            pos1 = 0;
                            pos2 = temp.IndexOf(laststring2);
                            namenode2 = temp.Substring(pos1, pos2 - pos1);
                        }
                    }
                }
            }
            else
                namenode2 = string.Empty;

            return namenode2;
        }
    }
}
```

NameP.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.IO;
using Renci.SshNet;

namespace ClusterGUI
{
    class NameP
    {
        public static string NamePackage(string result, string h1, string h2, string namenode1, string namenode2)
        {
            string r = result;
            string firststring = h1;
            string laststring1 = namenode1 + h2;
            string laststring2 = "enabled" + namenode2;
            string laststring3 = "disabled" + namenode2;
            const string statusP1 = "up";
            const string statusP2 = "down";
            const string statusP3 = "start_wait";
            const string statusP4 = "starting";
            const string statusP5 = "halting";
            const string statusP6 = "halt_wait";
```

```
const string statusP7 = "failing";
const string statusP8 = "fail_wait";
const string statusP9 = "relocate_wait";
const string statusP10 = "reconfiguring";
const string statusP11 = "reconfigure_wait";
const string statusP12 = "detached";
const string statusP13 = "unknown";
string namePackage = string.Empty;
int option = 0, pos1 = 0, pos2 = 0;

if (r.Contains(laststring1) == true)
{
    pos1 = r.IndexOf(firststring) + firststring.Length;
    pos2 = r.IndexOf(laststring1);

    namePackage = r.Substring(pos1, pos2 - pos1);
}
else
{
    if (r.Contains(laststring2) == true)
    {
        pos1 = r.IndexOf(firststring) + firststring.Length;
        pos2 = r.IndexOf(laststring2);

        namePackage = r.Substring(pos1, pos2 - pos1);
    }
    else if (r.Contains(laststring3) == true)
    {
        pos1 = r.IndexOf(firststring) + firststring.Length;
        pos2 = r.IndexOf(laststring3);

        namePackage = r.Substring(pos1, pos2 - pos1);
    }
}

if (namePackage.Contains(statusP1) == true)
    option = 1;
else if (namePackage.Contains(statusP2) == true)
    option = 2;
else if (namePackage.Contains(statusP3) == true)
    option = 3;
else if (namePackage.Contains(statusP4) == true)
    option = 4;
else if (namePackage.Contains(statusP5) == true)
    option = 5;
else if (namePackage.Contains(statusP6) == true)
    option = 6;
else if (namePackage.Contains(statusP7) == true)
    option = 7;
else if (namePackage.Contains(statusP8) == true)
    option = 8;
else if (namePackage.Contains(statusP9) == true)
    option = 9;
else if (namePackage.Contains(statusP10) == true)
    option = 10;
else if (namePackage.Contains(statusP11) == true)
    option = 11;
else if (namePackage.Contains(statusP12) == true)
    option = 12;
else if (namePackage.Contains(statusP13) == true)
    option = 13;

switch(option)
{
    case 1:
        pos1 = 0;
        pos2 = namePackage.IndexOf(statusP1);

        namePackage = namePackage.Substring(pos1, pos2 - pos1);
        break;
    case 2:
        pos1 = 0;
        pos2 = namePackage.IndexOf(statusP2);

        namePackage = namePackage.Substring(pos1, pos2 - pos1);
```

```
                    break;
                case 3:
                    pos1 = 0;
                    pos2 = namePackage.IndexOf(statusP3);

                    namePackage = namePackage.Substring(pos1, pos2 - pos1);
                    break;
                case 4:
                    pos1 = 0;
                    pos2 = namePackage.IndexOf(statusP4);

                    namePackage = namePackage.Substring(pos1, pos2 - pos1);
                    break;
                case 5:
                    pos1 = 0;
                    pos2 = namePackage.IndexOf(statusP5);

                    namePackage = namePackage.Substring(pos1, pos2 - pos1);
                    break;
                case 6:
                    pos1 = 0;
                    pos2 = namePackage.IndexOf(statusP6);

                    namePackage = namePackage.Substring(pos1, pos2 - pos1);
                    break;
                case 7:
                    pos1 = 0;
                    pos2 = namePackage.IndexOf(statusP7);

                    namePackage = namePackage.Substring(pos1, pos2 - pos1);
                    break;
                case 8:
                    pos1 = 0;
                    pos2 = namePackage.IndexOf(statusP8);

                    namePackage = namePackage.Substring(pos1, pos2 - pos1);
                    break;
                case 9:
                    pos1 = 0;
                    pos2 = namePackage.IndexOf(statusP9);

                    namePackage = namePackage.Substring(pos1, pos2 - pos1);
                    break;
                case 10:
                    pos1 = 0;
                    pos2 = namePackage.IndexOf(statusP10);

                    namePackage = namePackage.Substring(pos1, pos2 - pos1);
                    break;
                case 11:
                    pos1 = 0;
                    pos2 = namePackage.IndexOf(statusP11);

                    namePackage = namePackage.Substring(pos1, pos2 - pos1);
                    break;
                case 12:
                    pos1 = 0;
                    pos2 = namePackage.IndexOf(statusP12);

                    namePackage = namePackage.Substring(pos1, pos2 - pos1);
                    break;
                case 13:
                    pos1 = 0;
                    pos2 = namePackage.IndexOf(statusP13);

                    namePackage = namePackage.Substring(pos1, pos2 - pos1);
                    break;
            }

            return namePackage;
        }
    }
}
```

Program.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.IO;

namespace ClusterGUI
{
    static class Program
    {
        /// <summary>
        /// Punto de entrada principal para la aplicación.
        /// </summary>
        [STAThread]
        private static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new Form1());
        }
    }
}
```

Entregable 1: Manual de administración del clúster

1. Instalar cliente SSH.

Dado que el acceso al clúster se realiza de manera remota, y dando por hecho que el usuario utilizá una PC con sistema operativo Windows 7 (o posterior), se requiere la instalación de un cliente SSH. Se recomienda descargar *PuTTY* [https://www.ssh.com/ssh/putty/download].
Aquí [https://www.ssh.com/ssh/putty/windows/install] se puede consultar la guía de instalación.

2. Conectarse a LAN de clúster.

Es requisito indispensable que la PC del usuario esté conectada a la LAN donde está ubicado el clúster. Es posible acceder al clúster desde una red externa, configurando una computadora en la LAN del clúster como un servidor VPN. Aquí [https://www.xataka.com/seguridad/que-es-una-conexion-vpn-para-que-sirve-y-que-ventajas-tiene] se puede consultar más información de qué es un servidor VPN y como implementar uno. Este manual asume que la PC del usuario está conectada a la LAN del clúster.

3. Acceder al clúster.

Desde el escritorio de Windows, presione el botón de Inicio y seleccione PuTTY.

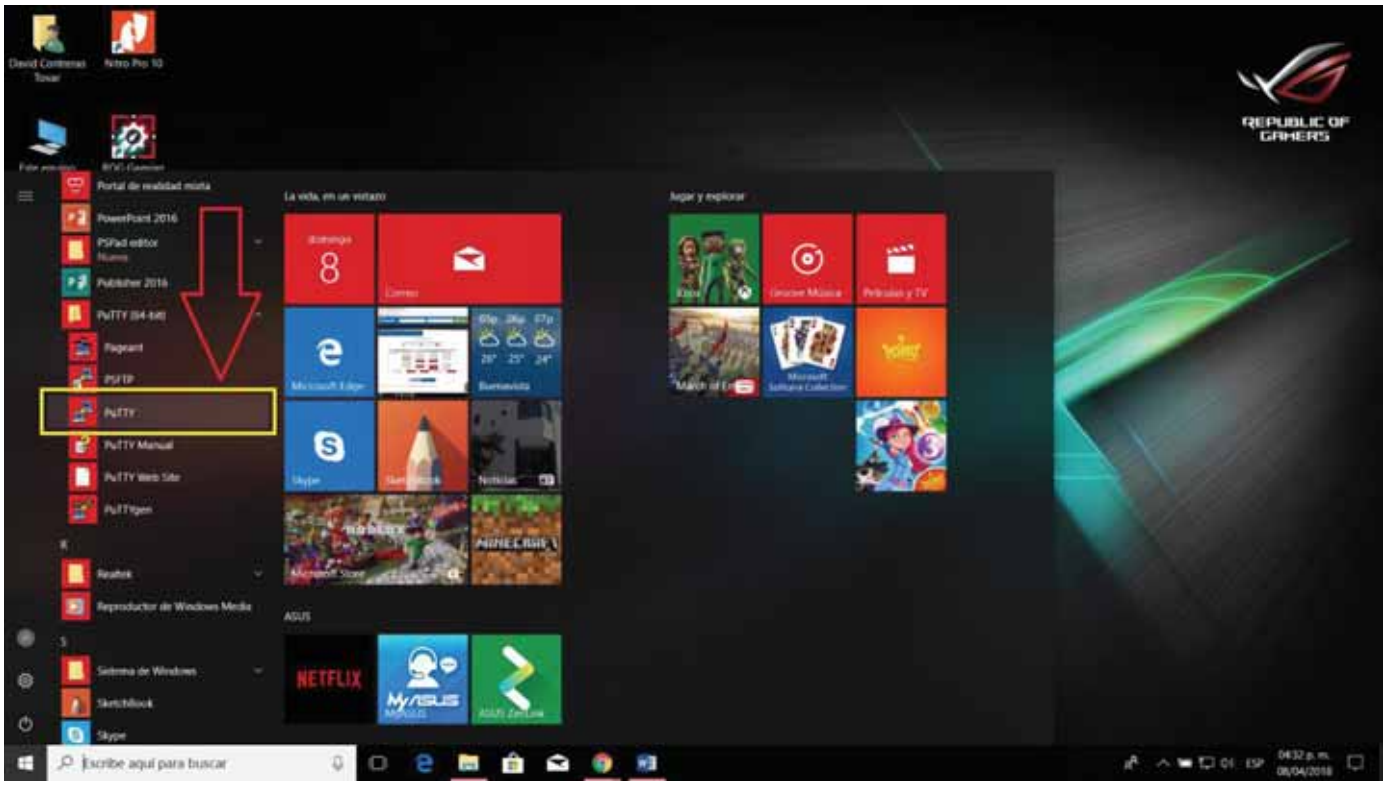


Figura M1.1. Iniciar cliente *PuTTY*.

Cuando se inicie el software, aparecerá una ventana con el título *PuTTY Configuration* con un campo *Host Name* en la parte superior intermedia. Introduzca el nombre de host al que desea conectarse (clúster) en ese campo y haga clic en *Open*.

El clúster implementado para el proyecto de investigación “Sistema de Computo Altamente Disponible”, con clave EL001-16, se ubica físicamente en la UAM Azcapotzalco. Su dirección IP es la **`172.16.24.160`**.

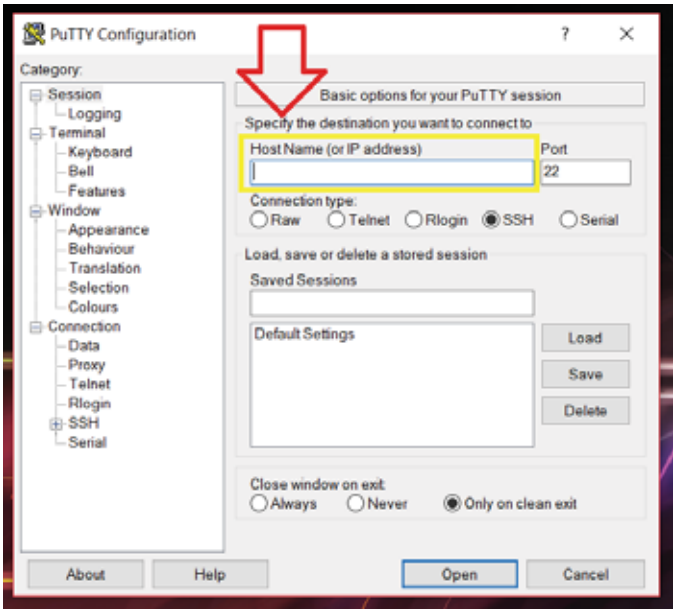


Figura M1.2. Campo Host Name en cliente *PuTTY*.

Si llegara a presentarse un error de conexión, la ventana que se obtiene es la siguiente:

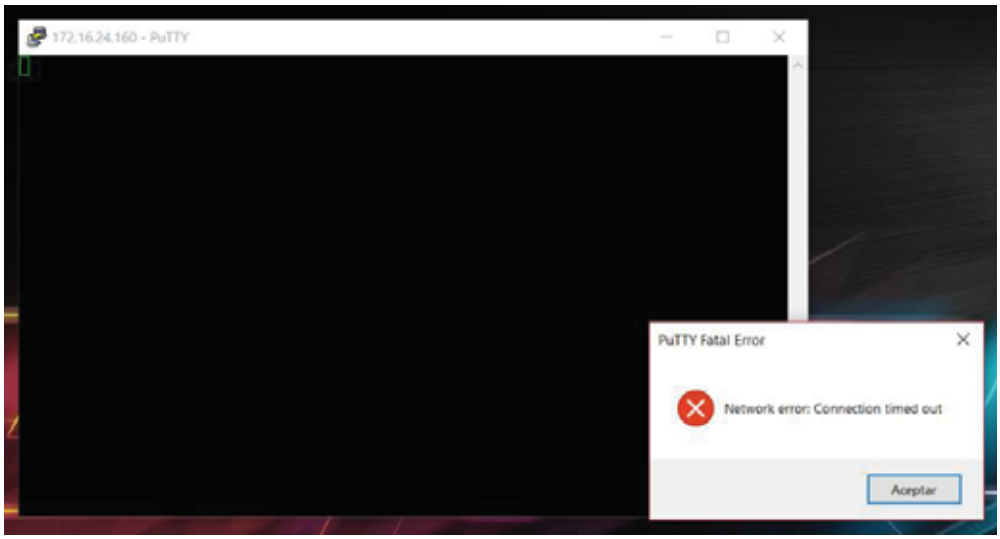


Figura M1.3. Error de conexión entre cliente *PuTTY* y el clúster.

En esta situación, espere algún tiempo para realizar otro intento de conexión.

Si la conexión fue exitosa, se despliega en pantalla la siguiente ventana:

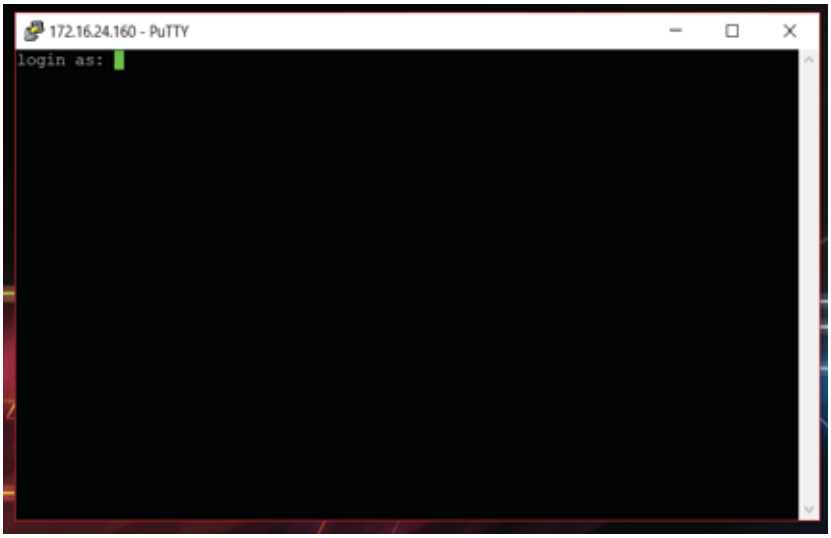


Figura M1.4. Conexión exitosa entre cliente *PuTTY* y el clúster.

Escribimos *root* como usuario, después el *password* y obtendremos acceso al clúster, como se muestra en la Figura M1.5. (Nota 1: por cuestiones de seguridad NO se proporciona el *password* del clúster. Acudir con el M. en C. José Ignacio Vega Luna, adscrito al Departamento de Electrónica, para más detalles).

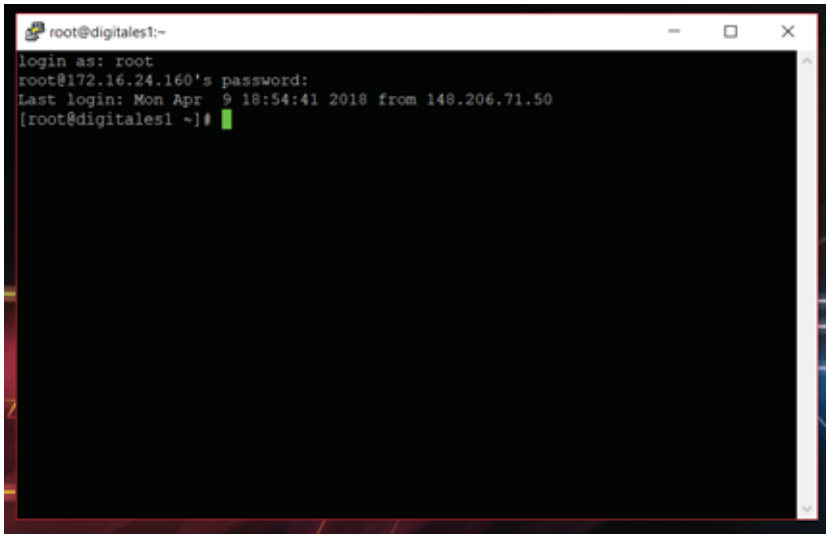


Figura M1.5. Acceso correcto al clúster.

4. Obtener estado del clúster.

Escribimos el comando *cmviewcl* para conocer el estado del clúster, como se exhibe a continuación:

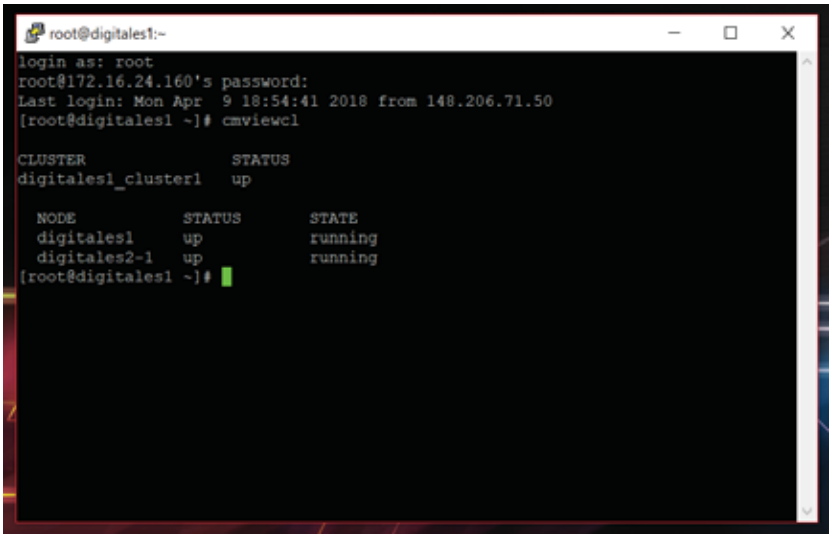


Figura M1.6. Estado del clúster.

Si deseamos obtener más detalles, o necesitamos verificar algún elemento del clúster en particular, podemos utilizar las opciones de *cmviewcl*.

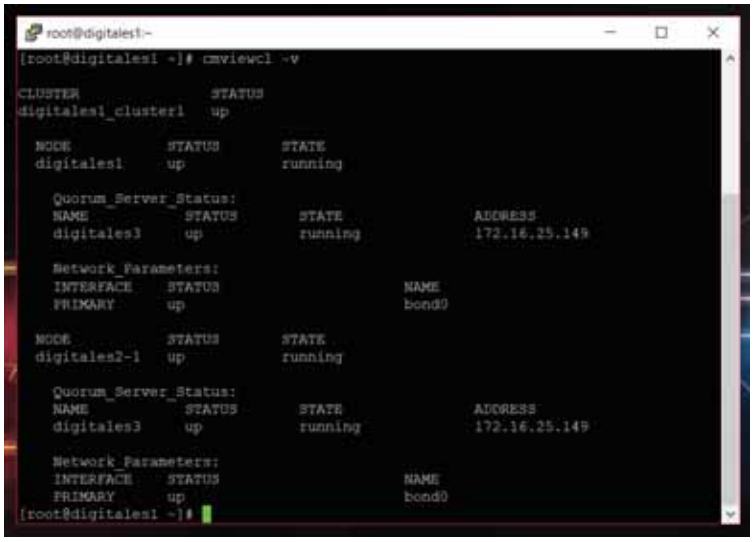


Figura M1.7. Salida del comando *cmviewcl –v*.

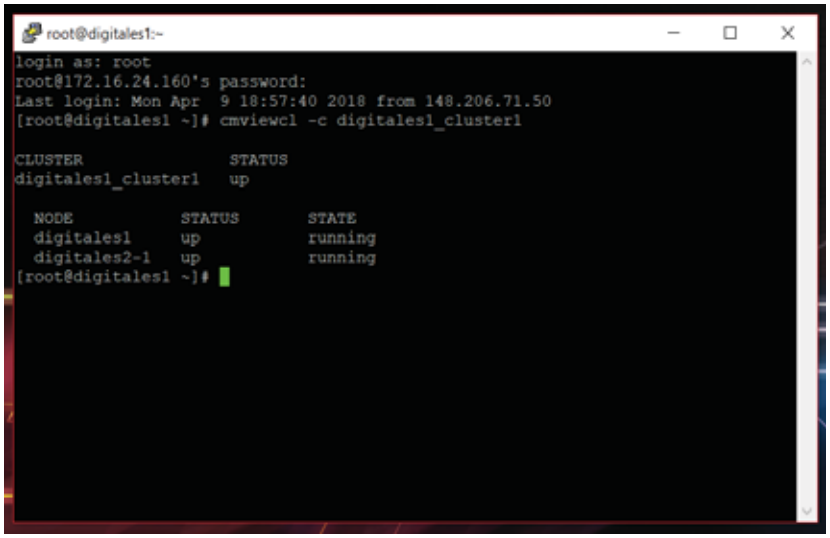


Figura M1.8. Salida del comando *cmviewcl –c*.

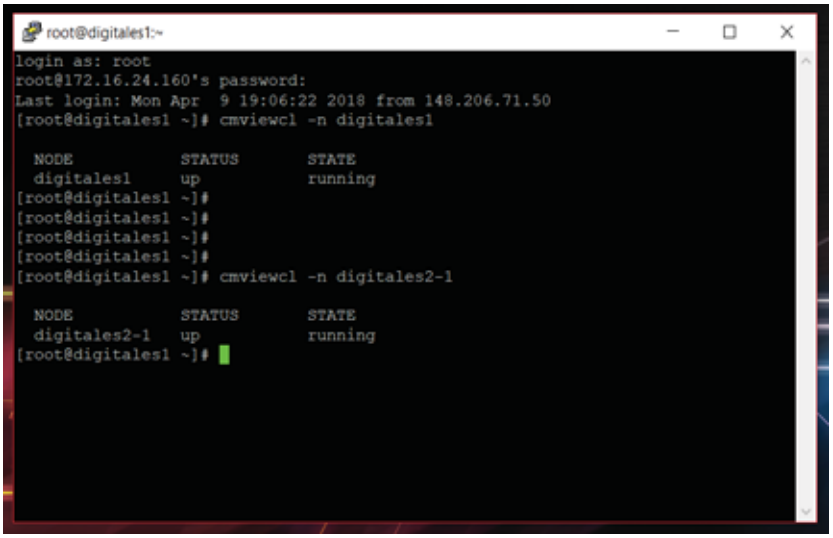


Figura M1.9. Salida del comando *cmviewcl –n.*

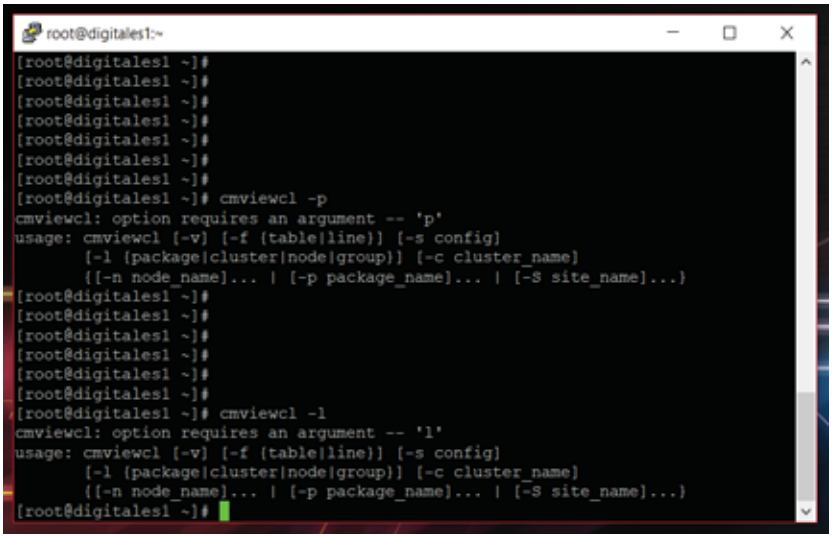


Figura M1.10. Salida del comando *cmviewcl –p* y *cmviewcl –l.*

Nota 2: Dado que en el momento de elaborar este manual, el clúster no tenía configurado un paquete como aplicación crítica, la salida de *cmviewcl –p* es la respuesta por default.

Más información de comandos para la administración del cluster, disponibles en la referencia [37].

Entregable 2: Manual de usuario de la interfaz gráfica

1. Instalar Microsoft .NET framework 4.7.1

Este framework permite la ejecución de aplicaciones .NET en Windows (7 o posterior). Descarga disponible en https://www.microsoft.com/es-mx/download/details.aspx?id=56115

Haga click en el botón Descargar, ubicado en la parte inferior de la ventana, tal como se muestra en la Figura M2.1.:

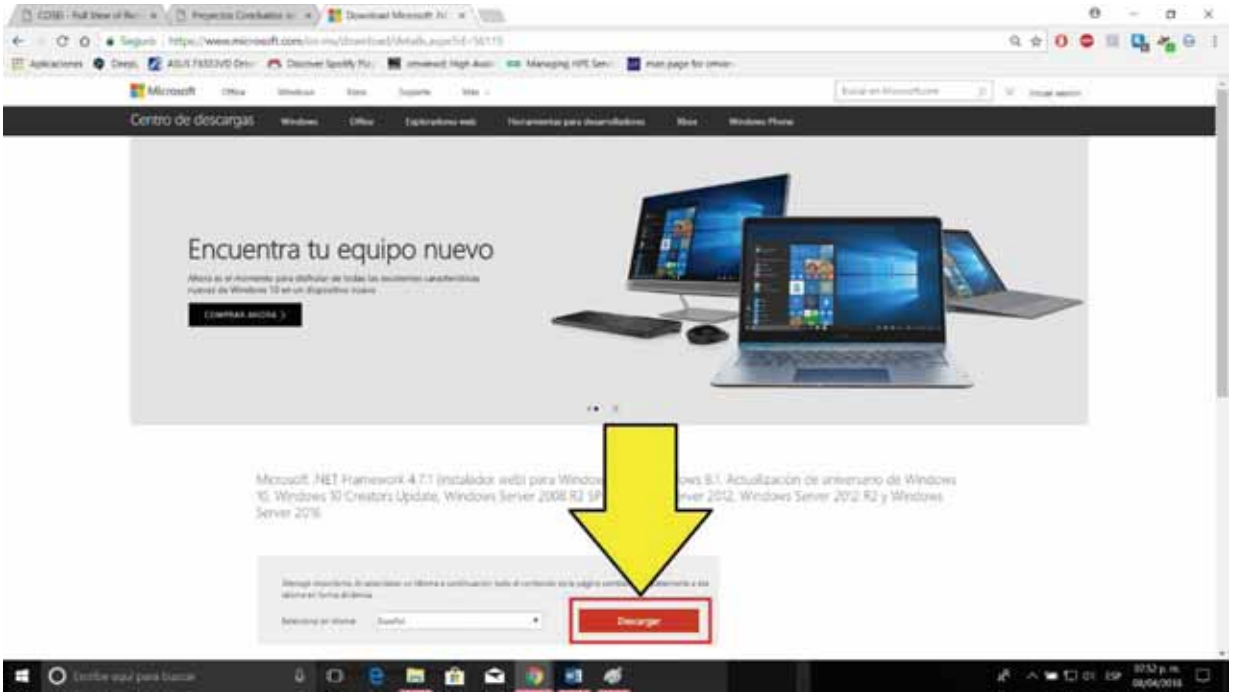


Figura M2.1. Descarga de Microsoft .NET framework 4.7.1

Seleccione la carpeta destino para el instalador y presione el botón aceptar. Una vez que la descarga haya concluido, haga doble click sobre el instalador. Espere que se abra la ventana de instalación, habilite la casilla *"He leído y acepto los términos de la licencia"* y presione el botón Instalar.

Figura M2.2. Ventana 1 de instalación de .NET framework 4.7.1

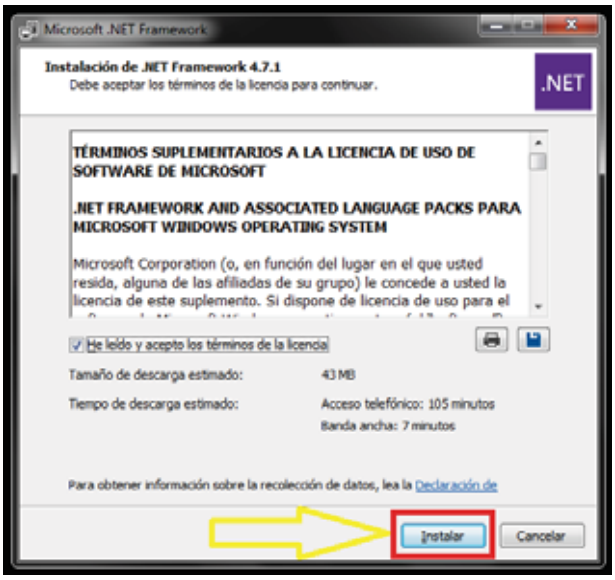


Figura M2.3. Ventana 2 de instalación de .NET framework 4.7.1

Espere a que la instalación se lleve a cabo, y cuando el proceso concluya, presione el botón finalizar.

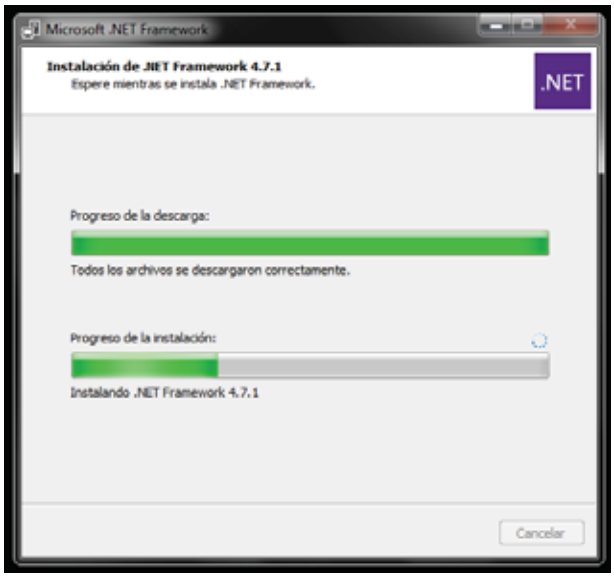


Figura M2.4. Instalación en proceso de .NET framework 4.7.1

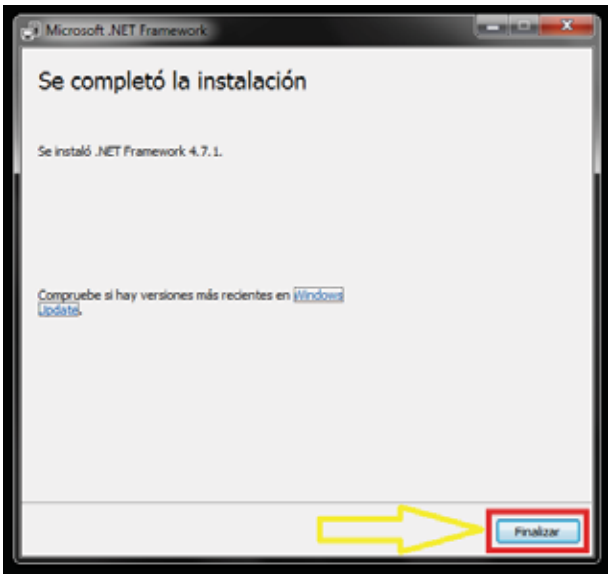


Figura M2.5. Instalación completa de .NET framework 4.7.1

2. Ejecutar la interfaz gráfica “Monitor de Cluster”.

Abra el archivo comprimido ***InterfazGraficaCluster.zip***, ubicado en la raíz del disco compacto entregado a la coordinación de la Licenciatura en Ingeniería en Computación de la UAM Azcapotzalco. Descomprima el contenido en la ruta **`C:\Users\Public\`**

Haga doble click en el archivo ejecutable ***ClusterGUI.exe***, ubicado en la carpeta *Release* (**`C:\Users\Public\InterfazGraficaCluster\Release`**), como se muestra en la Figura M2.6.

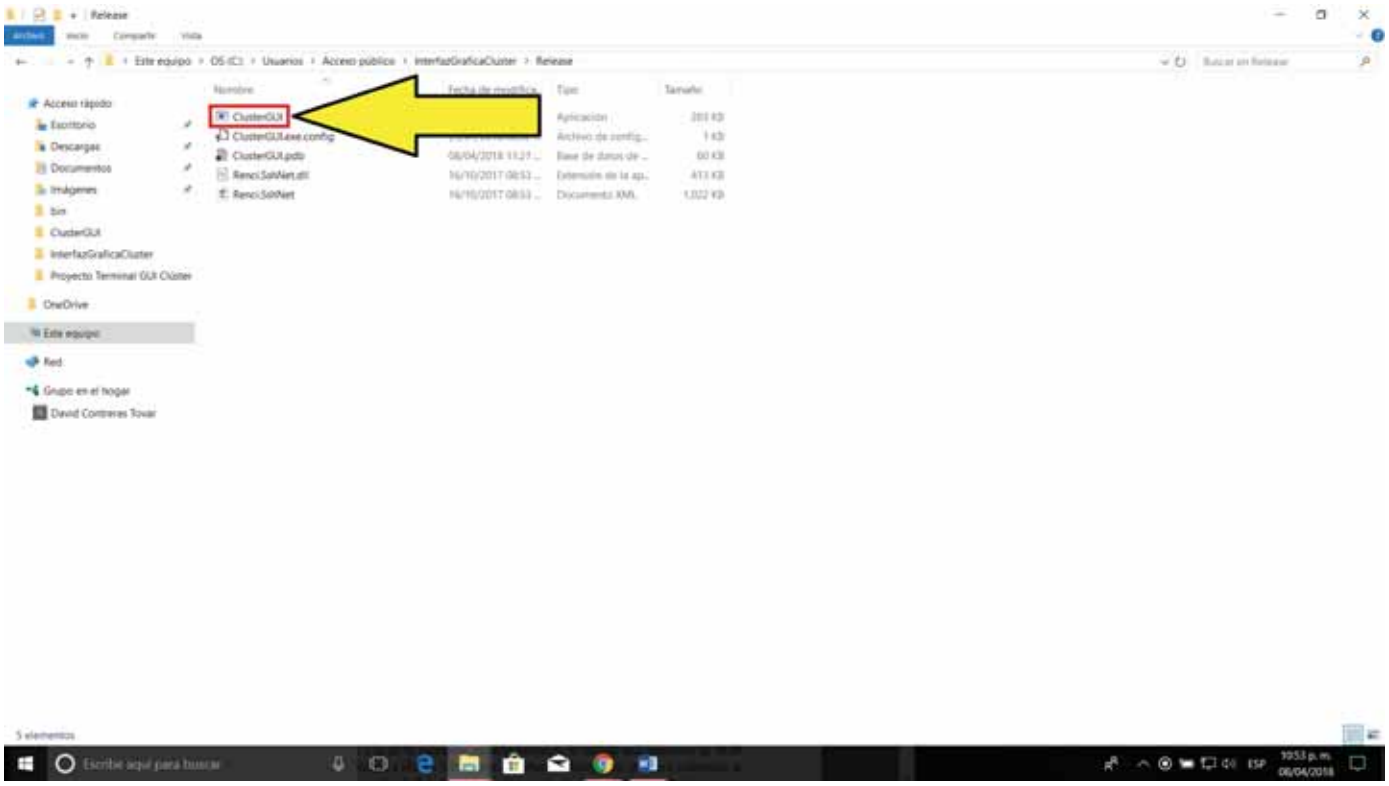


Figura M2.6. Ubicación del archivo ejecutable ***ClusterGUI.exe***

Nota 3: También puede encontrar el archivo ***ClusterGUI.exe*** y la carpeta *Release* en el archivo llamado ***CodigoFuenteClusterGUI.zip***, ubicado en la raíz del CD. Utilice la ruta de acceso **`CodigoFuenteClusterGUI.zip\ClusterGUI\ClusterGUI\bin`**

Se despliega en pantalla la ventana de la interfaz gráfica.

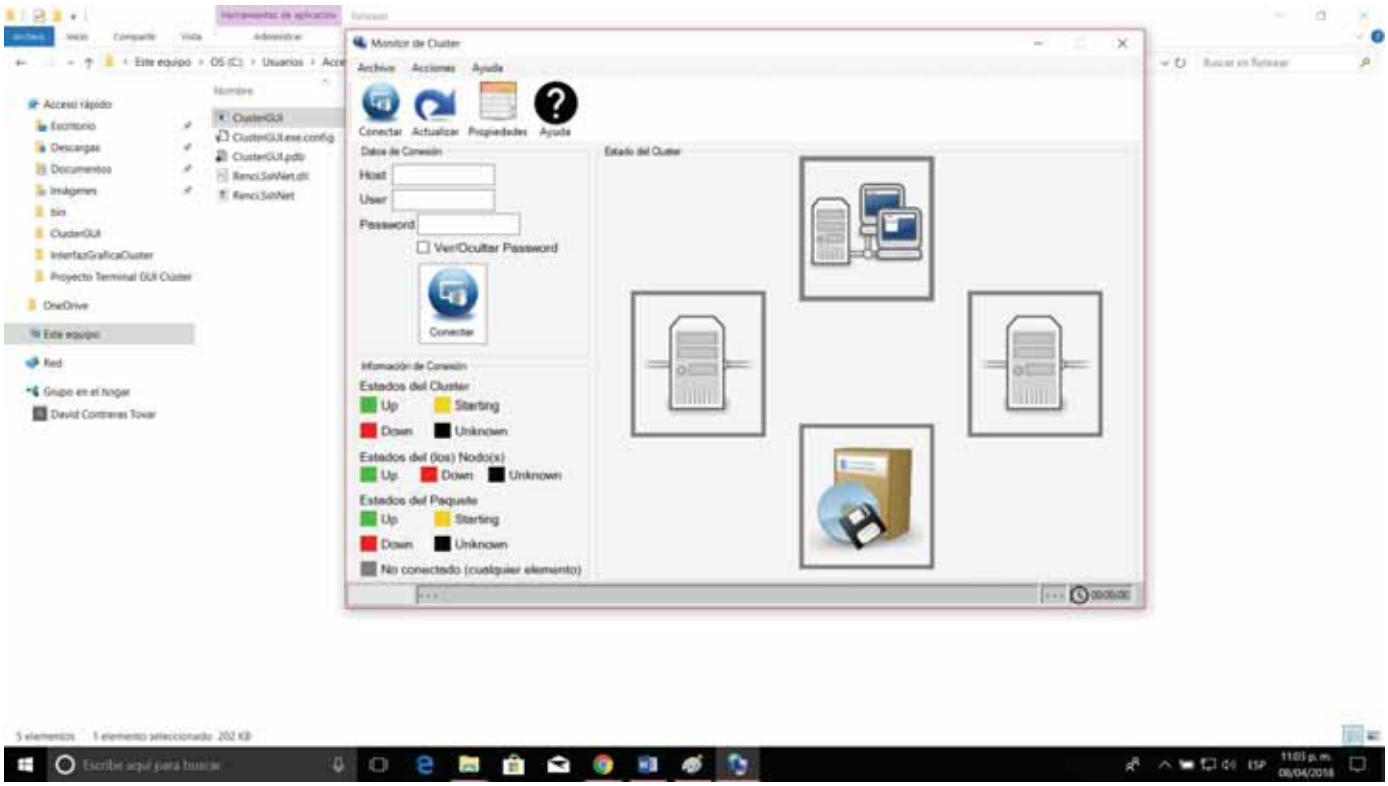


Figura M2.7. Aplicación de la interfaz gráfica "Monitor de Cluster" ejecutándose.

3. Funciones.
    a. Función principal: Conectar.

    Para iniciar una conexión remota con el clúster, escriba los datos necesarios en la sección llamada "Datos de Conexión" y presione el botón Conectar. Vea la Figura M2.8.

Figura M2.8. Iniciar conexión remota con el clúster.

Si la conexión remota se realizó de forma correcta, se visualiza el estado del clúster e información adicional. Vea la Figura M2.9.

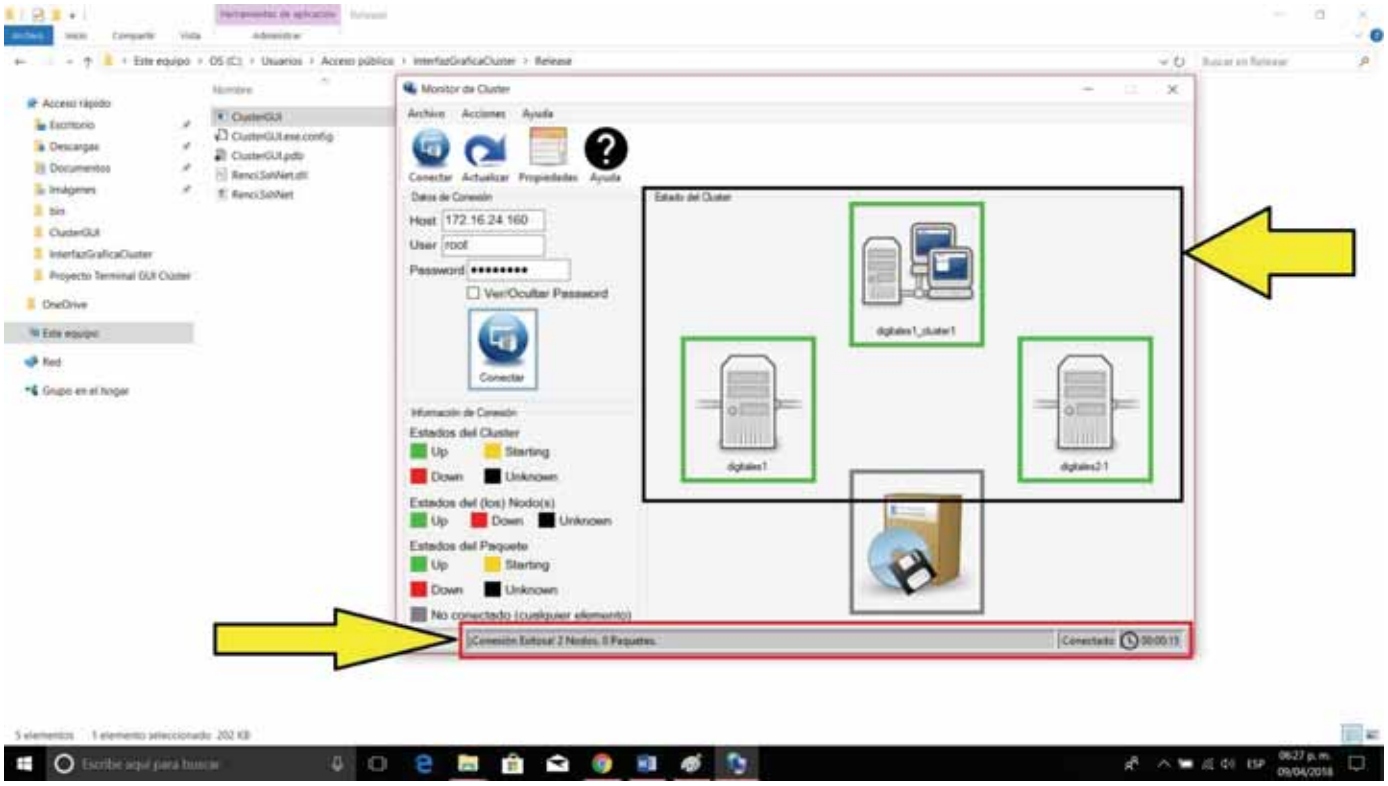


Figura M2.9. Conexión remota exitosa entre interfaz gráfica y clúster.

En caso de que se presente una conexión fallida, el resultado en pantalla sería el siguiente:

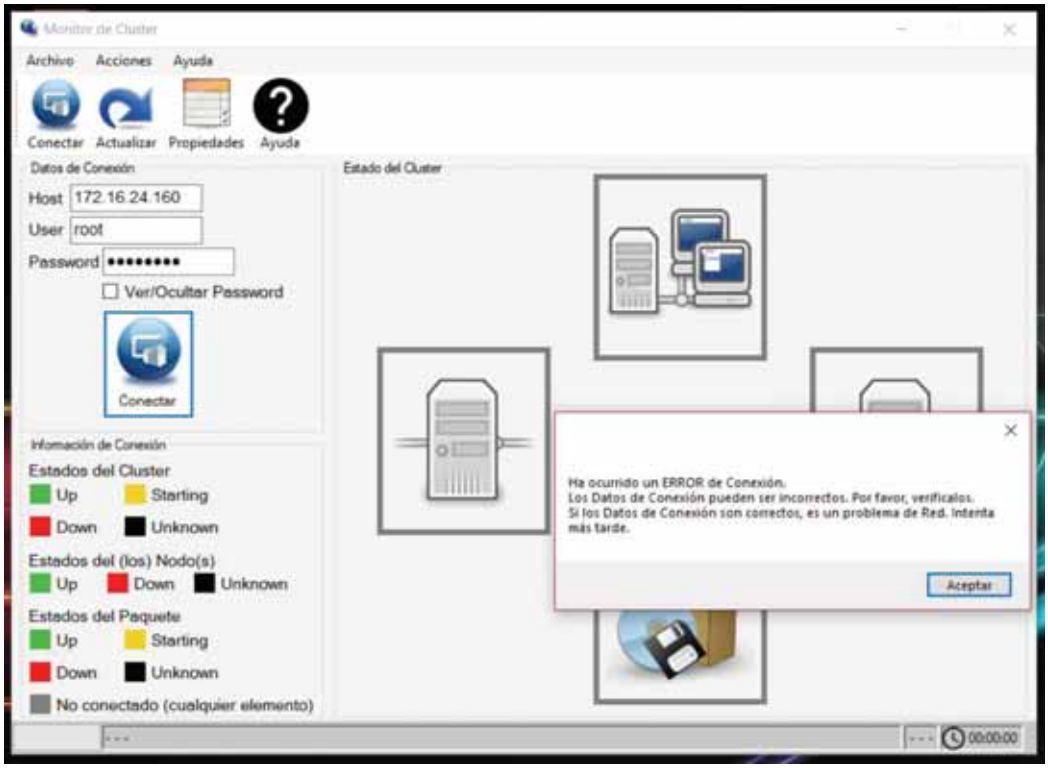


Figura M2.10. Mensaje 1 de conexión remota fallida entre interfaz gráfica y clúster.

Figura M2.11. Mensaje 2 de conexión remota fallida entre interfaz gráfica y clúster.

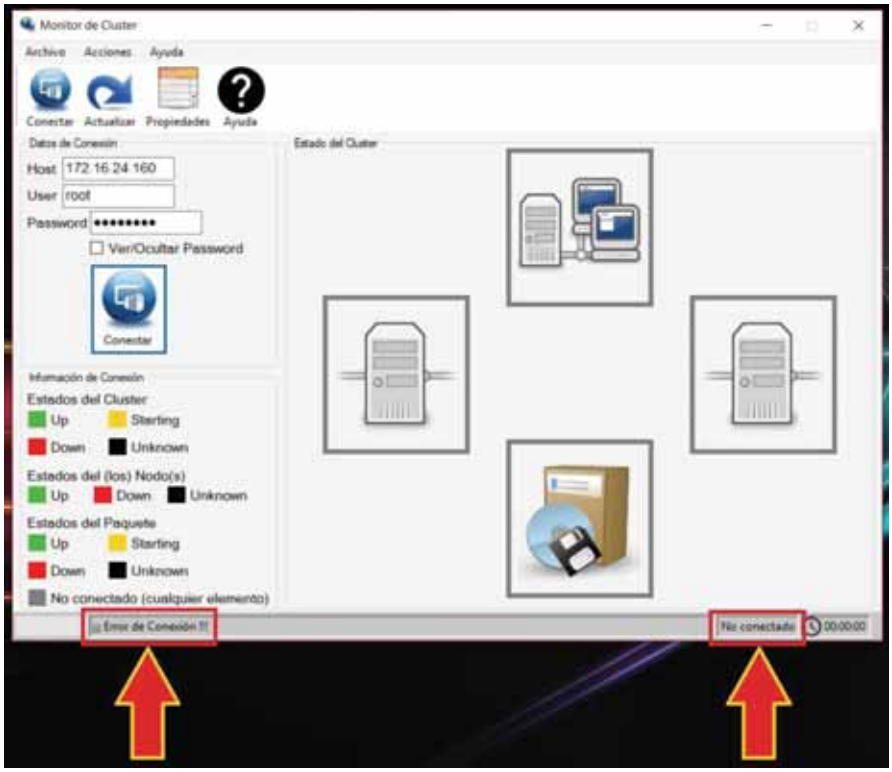


Figura M2.12. Notificación de conexión remota fallida entre interfaz gráfica y clúster.

b. Funciones adicionales.

Todas las siguientes funciones adicionales dependen de una conexión remota válida. En caso contrario, se despliega en pantalla un mensaje de error.

i. Actualizar. Si desea actualizar el estado del clúster, para verificar algún cambio, presione el botón Actualizar. Notará que los elementos del clúster, o algunos de ellos, cambiarán de color brevemente, para después volver a mostrar su estado. Mire la Figura M2.13.

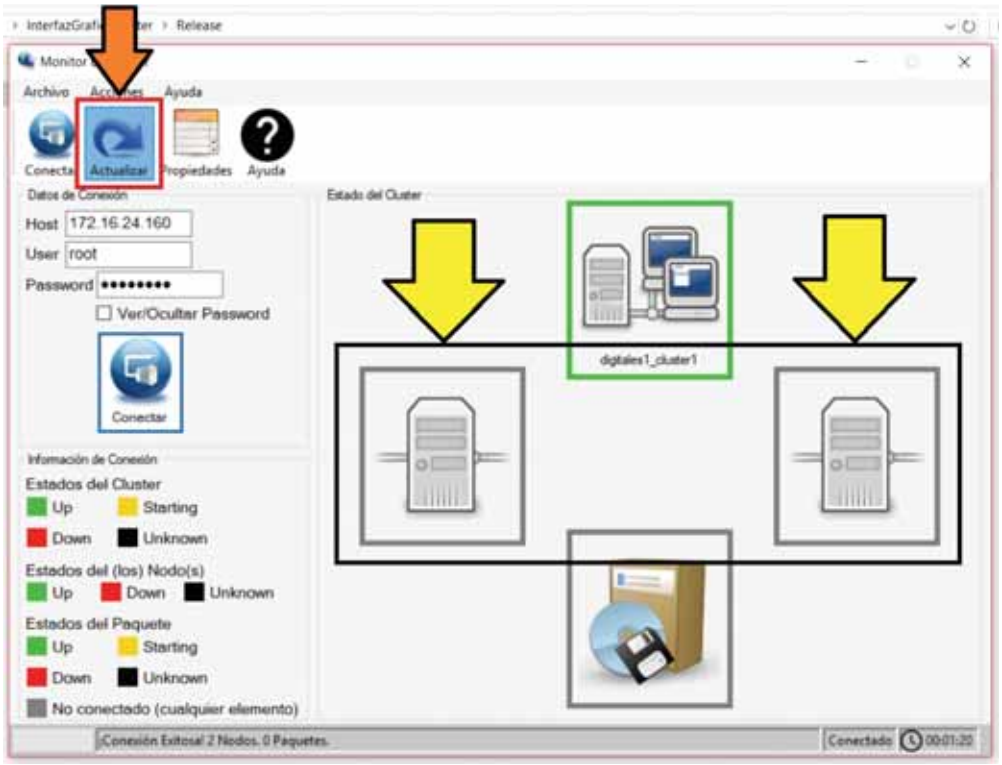


Figura M2.13. Actualizando el estado del clúster.

ii. Propiedades. Haga click en el botón Propiedades para visualizar las propiedades completas del estado del clúster en patalla. Mire la Figura M2.14.

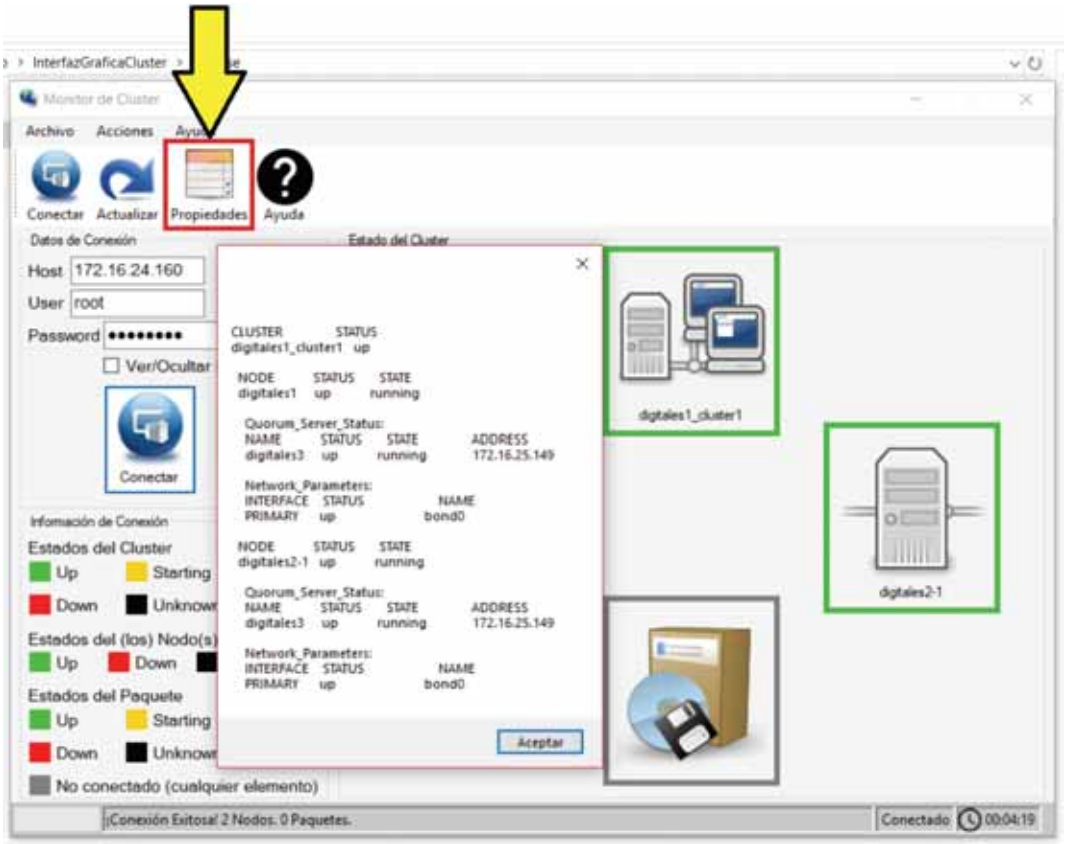


Figura M2.14. Propiedades del estado actual del clúster.

iii. Guardar propiedades. Si desea guardar las propiedades del estado del clúster en un archivo TXT, haga click en la opción Archivo de la barra de menú, y seleccione con otro click Guardar Propiedades.

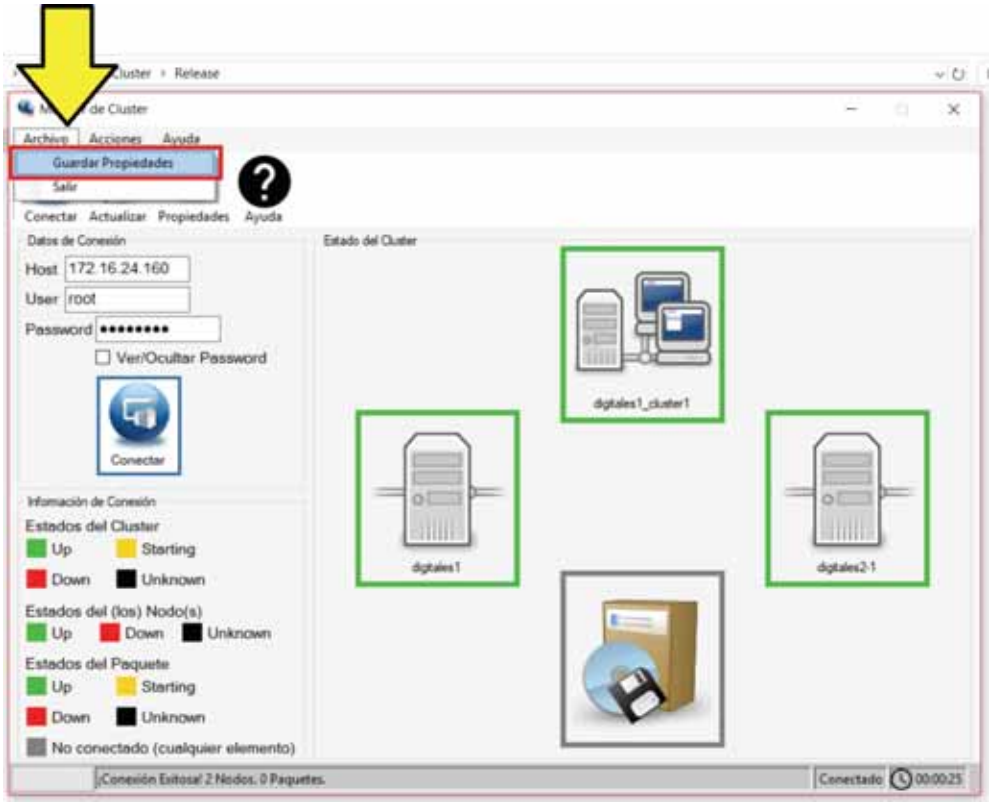


Figura M2.15. Guardar Propiedades del clúster en disco.

Si no ocurre ningún error durante la creación del archivo TXT, se muestra en pantalla un mensaje notificando la creación correcta del mismo y su ruta de ubicación.

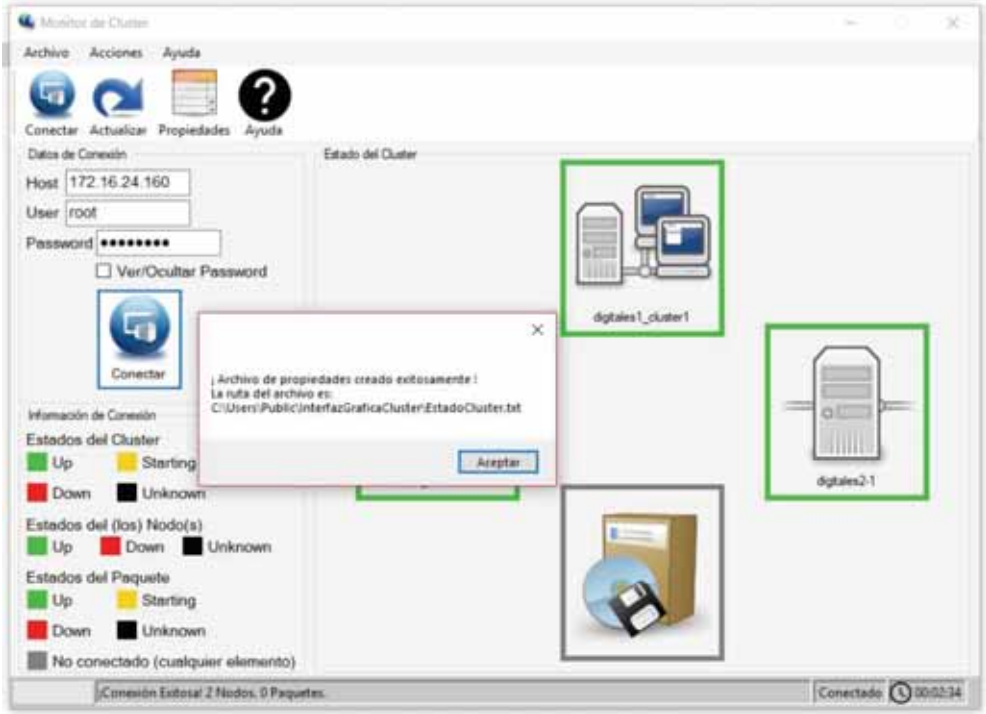


Figura M2.16. Creación exitosa del archivo ***EstadoCluster.txt***

El archivo ***EstadoCluster.txt*** es creado en la ruta
`C:\Users\Public\InterfazGraficaCluster`

Mire la Figura M2.17.

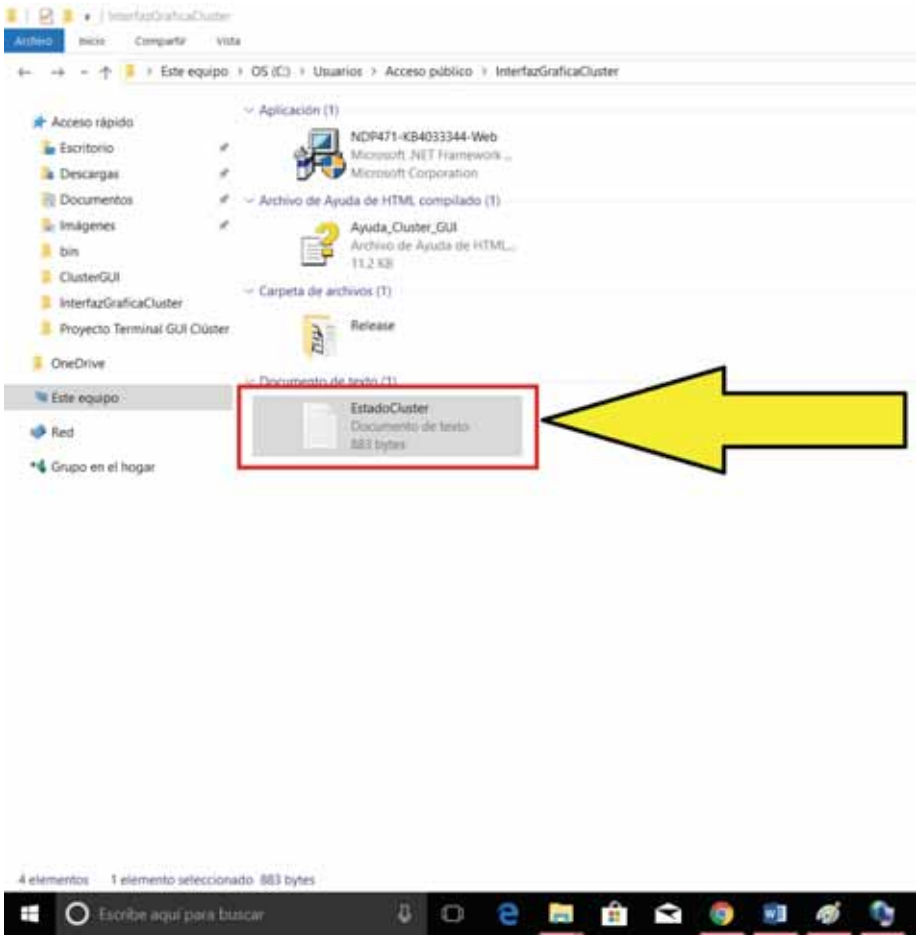


Figura M2.17. Ubicación del archivo ***EstadoCluster.txt***