

Universidad de las Ciencias Informáticas

Facultad 15



Título: Instalador del sistema de réplica Chronos.

Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas

Autor: Antonio Iván Aragón Leyva

Tutor: Ing. Lissuan Fadruga Artilles.

Mayo 2010

“Año del 50 Aniversario de la Revolución”



Los hombres son como los astros, que unos dan luz de sí y otros brillan con la que reciben.

José Martí

DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Facultad 15 de la Universidad de las Ciencias Informáticas; a ser uso del mismo en su beneficio.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Antonio Iván Aragón Leyva

Firma del Autor

Lissuan Fadruga Artilles

Firma del Tutor

DATOS DE CONTACTO

Antonio Iván Aragón Leyva

Email: aiaragon@estudiantes.uci.cu

Grupo: 1550

Sexo: M

AGRADECIMIENTOS

A la Revolución y a Fidel.

A mi tutor, a Arley gracias por su ayuda y apoyo incondicional.

Le doy muchas gracias a toda mi familia por estar siempre apoyándome y guiándome, especialmente a mi mamá, a mi papá, a mi hermana y a mis abuelos por su gran ayuda durante todos estos años., por confiar en mí, sin su apoyo y comprensión hoy no pudiera ser lo que soy.

Gracias a mi esposa Katia por su apoyo, su con fianza en mí, la fuerza que me dio muchas veces para seguir adelante y brindarme todo su amor y cariño.

A mis tíos, a mis primos por ayudarme siempre de forma incondicional.

A todos mis amigos por estar a mi lado en los momentos buenos y malos, especialmente, a Danny, Felipe, Koki, Leroy y a todos los demás, que son mucho pero todos sabes quienes son, gracias por su apoyo y comprensión.

A todas aquellas personas que de alguna forma pusieron su granito de arena en el desarrollo de este trabajo.

DEDICATORIA

A mi familia, gracias por:

Conseguirme la luz del sol a media noche y el número después del infinito, endulzar el agua del mar para mi sed y alquilarme el cuarto menguante de la luna.

A todos aquellos que hicieron todo lo posible para que yo no llegara hasta aquí, a ellos les debo gran parte de mi empeño.

RESUMEN

La réplica de datos ha alcanzado gran auge en la actualidad, pues es una necesidad de las empresas e instituciones científicas el almacenamiento de la información. En la Universidad de las Ciencias Informáticas (UCI) se desarrolla el sistema de réplica asíncrona multi-maestro Chronos, para el sistema gestor de base de datos PostgreSQL. Este sistema se desarrolla debido a la inexistencia de una herramienta para PostgreSQL que sea capaz de realizar una replicación de los datos entre nodos y la sincronización entre ellos, además de otros requisitos que esta herramienta va a proporcionar diferenciándola de las herramientas existentes. Se desean diseñar los componentes de la herramienta Chronos a partir de necesidades particulares de un escenario de réplica e implementando mejoras que el resto de estas herramientas no posee. Posibilitando una solución personalizada por el usuario que permita la programación de los eventos de sincronización. En el proceso de instalación de esta herramienta y su interfaz de administración en los nodos de un clúster de base de datos PostgreSQL, existe actualmente una alta complejidad y las posibilidades de errores se incrementan a medida que el clúster de base de datos es mayor. En el presente trabajo de diploma se arriba a la propuesta de una aplicación que facilitará la instalación de los componentes necesarios para el funcionamiento de Chronos en un clúster de base de datos y disminuirán las posibilidades de errores de instalación.

PALABRAS CLAVE

Réplica de datos, Base de datos, Gestor de base de datos, Chronos, Proceso de instalación, Clúster de base de datos.

TABLA DE CONTENIDOS

INTRODUCCIÓN	11
CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA	15
1.1. INTRODUCCIÓN	15
1.2. DEFINICIONES GENERALES.....	15
1.3. CHRONOS. COMPONENTES Y CARACTERÍSTICAS.....	17
1.3.1. Componentes del Sistema.....	18
1.3.1.1. Sistema Gestor de Base de Datos (SGBD) PostgreSql 1...N.....	19
1.3.1.2. LibTransaccionesSQL.so.....	19
1.3.1.3. Motor de Procesamiento de Sentencias (MPS).....	19
1.3.1.4. AdminChronos.....	19
1.3.1.5. BDChronos.....	20
1.3.2. Características del Sistema Chronos.....	20
1.3.2.1. Interfaz de administración, configuración y proceso de réplica (AdminChronos).....	20
1.3.2.1.1. Gestión de bases de datos involucradas en la réplica.....	21
1.3.2.1.2. Agrupamiento de fuentes de datos.....	22
1.3.2.1.3. Configuración básica del sistema para usuarios inexpertos.....	25
1.3.2.1.4. Configuración avanzada del sistema para usuarios expertos... 25	
1.3.2.1.5. Validación de la configuración.....	25
1.3.2.1.6. Módulo para la administración de Motores de Procesamiento de Sentencias.....	25
1.3.2.1.7. Sistema de seguridad y gestión de claves de usuario.....	26
1.3.2.2. Instalación gráfica y sencilla.....	26
1.3.2.3. Tipos de Replicación.....	26
1.3.2.4. Réplica de transacciones por filtros.....	27

1.3.2.5. Resolución de conflictos personalizados.	27
1.3.2.6. Métodos de Sincronizaciones.	27
1.3.2.7. Administración de conexiones.	28
1.3.2.8. Recuperación automática ante fallos.	28
1.3.2.9. Monitorización.	29
1.3.2.10. Módulo de ayuda.	29
1.3.2.11. Sistema de almacenamiento con bases de datos embebidas.	29
1.3.2.12. Administración del MPS de forma local.	30
1.4. HERRAMIENTAS PARA DESARROLLAR INSTALADORES DE SISTEMAS DE SOFTWARE EN PLATAFORMAS LINUX (DEBIAN).	30
1.4.1. BitRock InstallBuilder.	30
1.4.2. Launch4j.	31
1.4.3. InstallJammer.	32
1.4.4. CheckInstall.	32
1.4.5. Conclusiones del análisis de las herramientas.	33
1.5. INTERPRETES DE COMANDOS.	34
1.5.1. Shell.	35
1.5.2. Bourne Shell.	35
1.5.3. C Shell (csh).	36
1.5.4. Trusted C Shell (tcsh).	36
1.5.5. Korn Shell (ksh).	37
1.5.6. Bourne Again Shell (Bash).	38
1.5.7. Conclusiones del análisis de los intérpretes de comandos.	38
1.6. PROTOCOLOS DE TRANSMISIÓN DE DATOS.	39
1.6.1. Rsync.	39
1.8. CONCLUSIONES PARCIALES.	40
CAPÍTULO 2. GUÍA DE INSTALACIÓN DEL SISTEMA CHRONOS V1.0.	42

2.1. INTRODUCCIÓN.	42
2.2. PAQUETES DE INSTALACIÓN.	42
2.2.1. PostgreSql 8.4.....	43
1.2.2.AdminChronos.	45
.....	45
2.2.3. Chronos.....	46
2.3. TIPO DE SERVIDOR EN EL SISTEMA CHRONOS.	47
2.4. PASOS DE LA INSTALACIÓN.	47
2.5. CONCLUSIONES PARCIALES.....	50
CAPÍTULO 3: PRESENTACIÓN DEL PAQUETE DE INSTALACIÓN DE CHRONOS, CHRONOSINSTALL 0.3.	52
3.1 INTRODUCCIÓN.	52
3.2 DESCRIPCIÓN GENERAL DE LA HERRAMIENTA.	53
3.3 APLICACIÓN DE LA HERRAMIENTA CHECKINSTALL EN LA SOLUCIÓN PROPUESTA.	53
3.5 REALIZACIÓN DE LA INSTALACIÓN DEL PAQUETE DEBIAN (.DEB).	56
3.6. VALIDACIÓN DE LA SOLUCIÓN PROPUESTA MEDIANTE CASOS DE PRUEBA.	63
3.6.1. Casos de prueba.	63
3.7 CONCLUSIONES PARCIALES.....	65
CONCLUSIONES.....	66
RECOMENDACIONES.....	68
BIBLIOGRAFÍA.....	69
GLOSARIO DE TÉRMINOS:.....	71

Introducción.

Los procesos de negocio involucran actualmente grandes volúmenes de información y la complejidad de sus relaciones se incrementa cada día por lo que las empresas e instituciones científicas buscan alternativas eficientes y poco costosas que permitan el almacenamiento de la información. La réplica de datos surge como una de las soluciones que pretende fortalecer el tratamiento y almacenamiento de los datos, y que ha alcanzado gran auge en la actualidad. La réplica de bases de datos consiste en el proceso de copiar y mantener objetos de las base de datos, como por ejemplo relaciones, en múltiples bases de datos que forman un sistema de bases de datos distribuido. Las técnicas de réplica son manejadas por distintos sistemas gestores de base de datos, dígase Oracle, MySQL, PostgreSQL, siendo este último un potente gestor considerado como una de las alternativas de sistema de código abierto más utilizada. Existen numerosas herramientas desarrolladas para PostgreSQL enfocadas a la réplica de este tipo de bases de datos. Sus características varían en dependencia del tipo de réplica que implementan y algunas son muy utilizadas.

En la Universidad de las Ciencias Informáticas se desarrolla el sistema de réplica asíncrona multi-maestro, Chronos. Este sistema pretende complementar las características en el tratamiento de los datos como una solución de réplica para PostgreSQL, específicamente en la replicación asíncrona multi-maestro. Con esta solución se alcanza la implementación de una herramienta que responde a las necesidades para este tipo de entornos de réplica, que proporciona una resolución de conflictos personalizada por el usuario en dependencia de las características de su negocio. Garantiza, además, la configuración de diferentes tipos de sincronización que ofrecen mayor posibilidad de elección al administrador de bases de datos en el diseño de la réplica. Permite la programación de los eventos de sincronización, característica que no está presente en ninguna de las herramientas similares existentes. Posee una interfaz de usuario que facilita el trabajo de configuración y monitorización de todo el clúster de bases de datos.

Chronos está compuesto por tres componentes básicamente, el Motor de Procesamiento de Sentencias (MPS), la interfaz gráfica de usuario de administración y configuración y una librería dinámica que captura las transacciones que se realizan en el gestor de base de datos y establece la comunicación con el MPS para las operaciones de réplica. Todo el proceso de instalación y configuración del clúster de bases de datos debe realizarse teniendo en cuenta un conjunto bastante amplio de requisitos para el correcto funcionamiento del sistema. Es por eso que el proceso de instalación es bastante engorroso y las posibilidades de errores se incrementan a medida que el clúster de base de datos es mayor. Este proceso involucra copiar las librerías dinámicas utilizadas, monitorear la instalación de paquetes necesarios para el funcionamiento del sistema y asegurar el estado correcto del clúster de bases de datos para comenzar la configuración de la réplica.

A partir del análisis de esta **situación problemática** surge la presente investigación que identifica el **problema** de la existencia de una alta complejidad en el proceso de instalación de la herramienta de replicación Chronos y su interfaz de administración en los nodos de un clúster de bases de datos PostgreSQL.

Objeto de estudio:

Las herramientas para la instalación y configuración de sistemas de software sobre plataformas UNIX.

Campo de acción:

Herramientas para la instalación y configuración de sistemas de software sobre plataformas Linux Debian.

Objetivo General:

Implementar una aplicación para la instalación de los componentes necesarios para el funcionamiento de Chronos en un clúster de base de datos.

Objetivos Específicos:

1. Desarrollar el marco teórico de la investigación.
2. Identificar las herramientas adecuadas para el desarrollo de aplicaciones de este tipo.
3. Realizar la implementación de una aplicación informática que permita la instalación de las herramientas Chronos y AdminChronos.
4. Validar la aplicación utilizando las técnicas adecuadas.

Para guiar la presente investigación se plantea la siguiente idea a defender:

Si se realiza la implementación de un paquete para la instalación y configuración del sistema Chronos y su interfaz de administración en un clúster de bases de datos PostgreSQL se minimizará la complejidad existente en este proceso.

Tareas:

1. Realizar un estudio del estado del arte de las herramientas actuales que facilitan la instalación de sistemas en plataformas Linux.
2. Analizar el proceso de instalación y configuración del sistema Chronos.
3. Fundamentar el uso de las herramientas en la elaboración de la propuesta de solución.
4. Realizar la implementación del paquete de instalación y configuración del sistema Chronos y su interfaz de administración.
5. Validar la aplicación propuesta.

Para dar cumplimiento a las tareas propuestas anteriormente se van a utilizar los métodos científicos de la investigación: **teórico y empírico.**

Entre los métodos **teóricos** utilizados para la investigación se emplearon:

- El método **analítico-sintético**, para la realización del estudio teórico de la investigación y en la investigación previa sobre el funcionamiento del proceso de replicación de datos del sistema Chronos permitiendo extraer los elementos más importantes.
- El método **inductivo-deductivo**, para a través de un razonamiento llegar a un grupo de conocimientos particulares y generales.

De los métodos **empíricos** se utilizó: La **entrevista** para obtener información valiosa del proceso de replicación del sistema Chronos.

Capítulo 1. Fundamentación teórica.

1.1. Introducción

El presente capítulo incluye definiciones generales relacionadas con la réplica de datos, entorno multi-maestro, escenario de réplica y replicación asíncrona, posteriormente se abordan aspectos relacionados con el sistema Chronos, describiendo sus componentes, sus principales características y el proceso de réplica que realiza. Se describen además, las principales herramientas necesarias para la creación de paquetes de instalación, específicamente para software libre. Además se especifican las herramientas utilizadas en la elaboración de la propuesta de solución. Finalmente se incluye un estudio de los principales intérpretes de comandos y el protocolo de transmisión de datos Rsync.

1.2. Definiciones generales.

Muchas han sido las personalidades e instituciones que dedican gran parte de sus estudios a la réplica de datos, por lo que a lo largo de la historia del desarrollo del software se pueden encontrar infinidad de documentos y definiciones que hacen referencia a este tan importante tema. Se introduce diferentes definiciones necesarias.

1.2.1. Réplica de datos.

La réplica es un conjunto de tecnologías destinadas a la copia y distribución de datos y objetos desde una base de datos a otra, para luego sincronizar ambas bases de datos y mantener su coherencia. La réplica permite distribuir la información entre diferentes ubicaciones y entre usuarios remotos o móviles mediante redes locales y de área extensa, conexiones de acceso telefónico, conexiones inalámbricas e Internet. (2008)

La replicación de datos es el envío de información entre dos o más nodos, permitiendo que ciertos datos se almacenen en más de un sitio, aumentando la disponibilidad de los mismos y mejorando la eficiencia del acceso a dicha información. La réplica es una técnica que permite copiar y distribuir idénticamente las tablas de una base de datos en múltiples bases de datos ubicadas en diferentes nodos de la red. La replicación asegura que los datos correctos estén siempre disponibles en el momento y en el lugar necesario. Posee las siguientes características:

- **Alta Disponibilidad:** La información se encuentra en varios servidores, lo que implica que si por algún fallo uno de estos servidores no está activo la información continúa estando disponible.
- **Tolerancia a fallos:** Garantiza un comportamiento correcto, donde efectivamente pueden existir un número finito de fallos y tipos de fallos, sin afectar el estado y la disponibilidad de la información.
- **Rendimiento:** Provee un rápido acceso local a datos compartidos ya que balancea la actividad sobre múltiples sitios. Algunos usuarios pueden acceder a un servidor mientras que otros acceden a diferentes servidores, reduciendo la carga de todos los servidores. También, los usuarios pueden acceder a datos de un sitio replicado que tiene menos costo de acceso, típicamente un sitio que se encuentra geográficamente más cerca de ellos.
- **Reducción de la carga:** Es utilizada para distribuir datos sobre lugares en múltiples regiones. Así, las aplicaciones pueden acceder a varios servidores regionales en vez de acceder a un servidor central. Esta configuración puede reducir la carga de la red.
- **Reducción de costo:** Es menos costoso acceder a uno o más servidores locales o geográficamente más cercanos, que a uno o unos que se encuentre demasiado lejos.

1.2.2 Entorno multi-maestro.

El entorno multi-maestro ofrece la capacidad de replicar los datos en las tablas a través de bases de datos separadas. En él los datos se ingresan, se leen y modifican en cualquier nodo del sistema.

1.2.3 Replicación asíncrona.

La replicación asíncrona proporciona tolerancia a fallos en servidores y almacenamiento en redes, trabaja capturando los cambios que se producen en cualquier fichero gestionado por el sistema operativo del servidor. Esto se logra mediante la instalación de un driver de filtrado, cuya función consiste en discriminar todas las transacciones enviadas a un determinado sistema de ficheros. Utilizando un pequeño número de reglas, el driver captura una copia de cada transacción y la envía a un servicio de sistema, el cual entonces la transmite vía TCP/IP al servidor determinado previamente. (Net)

1.2.4 Escenario de réplica.

Escenario de réplica es el entorno donde se realiza la réplica de los datos. Principalmente los escenarios de réplica se dividen en dos categorías generales:

- Replicar datos en un entorno de servidor a servidor.
- Replicar datos entre un servidor y los clientes.

1.3. Chronos. Componentes y características.

Chronos es una herramienta de réplica asíncrona para entornos multi-maestro que pretende satisfacer la mayor parte de los requerimientos. Ha sido diseñada a partir de necesidades

particulares de un escenario de réplica, asumiendo las principales potencialidades de las soluciones similares en la actualidad para PostgreSQL e implementando nuevas mejoras que la distinguen del resto. Chronos surge a partir de la necesidad de establecer la sincronización de varios servidores de Bases de Datos en un modelo de datos correspondiente a un negocio con características particulares. (Chronos) La adaptación de herramientas de réplica existentes para PostgreSQL que cubran este escenario, provocaría mayor utilización de recursos, escasa personalización de la solución y partes incompletas que no podrían resolverse. De ahí que surge Chronos como una herramienta cuyas características permiten su aplicación en muchas situaciones con estos requerimientos.

1.3.1. Componentes del Sistema.

Chronos está integrado por varios componentes que permiten el funcionamiento de la réplica y la integración con el Sistema de Gestión de Bases de Datos (SGBD) PostgreSQL. En la Figura 1.1 se representa un diagrama de componentes generales del mismo.

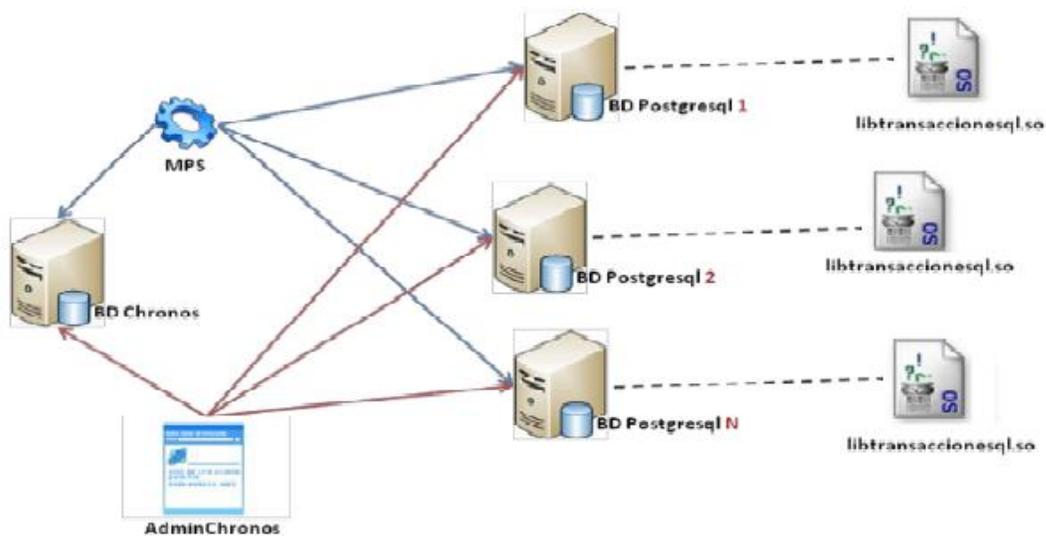


Figura 1.1 Componentes del Sistema Chronos

1.3.1.1. Sistema Gestor de Base de Datos (SGBD) PostgreSql 1...N.

Versiones liberadas de PostgreSql que constituyen los Sistemas de Gestión de Bases de Datos. Estos no forman parte de Chronos, se ponen en el diagrama para entender su interacción con el resto de los componentes. (Chronos)

1.3.1.2. LibTransaccionesSQL.so.

Librería de enlace dinámico que extiende la funcionalidad del gestor capturando los cambios que se producen en cualquiera de las tablas contenidas, los cuales deben ser replicados según el tipo de sincronización que se configure. (Chronos)

1.3.1.3. Motor de Procesamiento de Sentencias (MPS).

Constituye el núcleo de la aplicación. Se ejecuta como un proceso independiente capturando, planificando y replicando los cambios ocurridos en los SGBD PostgreSql. Posee programación concurrente para aprovechar las características de multiprocesamiento. Hace uso inteligente de conexiones a los SGBD (pool de conexiones) maximizando el rendimiento y la eficiencia total del sistema. (Chronos)

1.3.1.4. AdminChronos.

Aplicación web de administración, configuración y monitorización en línea del sistema. Posee una interfaz gráfica de usuario amigable y fácil de usar, lo que le confiere un valor agregado a la aplicación. A través de esta es posible realizar todas las configuraciones de sincronización para todos los nodos del clúster de manera sencilla y rápida. Permite monitorizar el estado de la réplica haciendo interactuar al usuario con este como parte del sistema. No interviene en el rendimiento de los nodos del clúster de bases de datos puesto que puede ser utilizada desde una estación de trabajo remota que tenga acceso a la base de datos del sistema Chronos. (Chronos)

1.3.1.5. BDChronos.

Almacena toda la configuración de las sincronizaciones gestionada por AdminChronos y consultada por el MPS para realizar la réplica de datos.

1.3.2. Características del Sistema Chronos.

La implementación y funcionamiento de estos componentes permiten garantizar un conjunto de características en el sistema, entre ellas se encuentran las que lo diferencian del resto de las herramientas de su tipo. Estas características son:

- Interfaz de administración, configuración y proceso de réplica (AdminChronos).
- Instalación gráfica y sencilla.
- Tipos de replicación.
- Réplica de transacciones por filtros.
- Resolución de conflictos personalizados.
- Métodos de sincronizaciones.
- Administración de conexiones.
- Recuperación automática ante fallos.
- Monitorización.
- Módulo de ayuda.
- Sistema de almacenamiento con bases de datos embebidas.
- Administración del MPS de forma local.

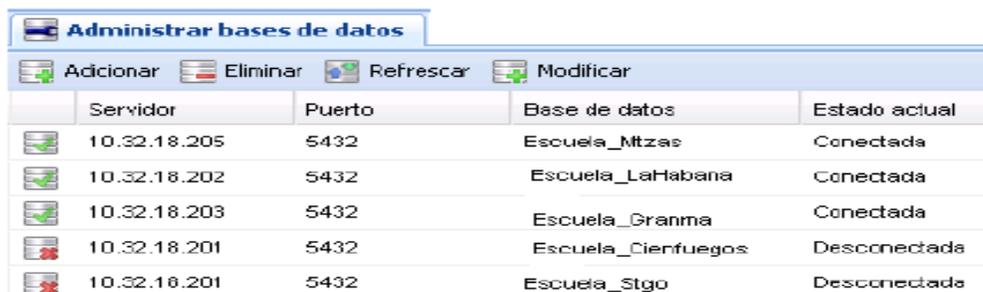
1.3.2.1. Interfaz de administración, configuración y proceso de réplica (AdminChronos).

La interfaz de administración y configuración de todo el proceso de réplica garantiza la configuración remota de todo el sistema, así como las actividades de monitorización y administración. Permite las siguientes funcionalidades.

- Gestión de bases de datos
- Agrupamiento de fuentes de datos.
- Configuración básica del sistema para usuarios inexpertos.
- Configuración avanzada del sistema para usuarios expertos.
- Validación de la configuración.
- Módulo para la administración de motores de procesamiento de sentencias.
- Sistema de seguridad y gestión de claves de usuario.
- Configuración de fuentes de datos ad-hoc.

1.3.2.1.1. Gestión de bases de datos involucradas en la réplica.

Al iniciar el proceso de configuración, el usuario debe establecer cuáles serán las bases de datos involucradas en la réplica. AdminChronos garantiza la gestión de estas bases de datos permitiendo adicionar nuevas bases de datos, eliminar, modificar su información, y ver el estado de conexión de las mismas, como se muestra a continuación en la Figura 1.2.



	Servidor	Puerto	Base de datos	Estado actual
	10.32.18.205	5432	Escuela_Mtzas	Conectada
	10.32.18.202	5432	Escuela_LaHabana	Conectada
	10.32.18.203	5432	Escuela_Granma	Conectada
	10.32.18.201	5432	Escuela_Cienfuegos	Desconectada
	10.32.18.201	5432	Escuela_Stgo	Desconectada

Figura 1.2. Módulo para la gestión de bases de datos.

1.3.2.1.2. Configuración de fuentes de datos *ad-hoc* o en caliente.

Para Chronos, la unidad más pequeña de la réplica de datos es una fuente, equivalente a una tabla de la base de datos. AdminChronos permite la adición de fuentes de datos *ad-hoc* a una sincronización ya configurada, lo que indica que esta se restablecerá desde cero.

1.3.2.1.2. Agrupamiento de fuentes de datos.

Para facilitar el proceso, Chronos permite la unión de muchas fuentes de datos en grupos, estos se corresponden como orígenes y destinos de una sincronización, lo que ahorraría el tedioso proceso de crear sincronizaciones para cada par de fuente de datos involucradas. Para establecer sincronizaciones entre grupos, los criterios de agrupación son:

- **De un origen a múltiples destinos.**

El grupo origen contiene una tabla de una base de datos y el grupo destino contiene todas las tablas que se sincronizarán con esta en la réplica. Por supuesto las tablas agrupadas deben cumplir con el criterio de igualdad entre fuentes de datos y corresponden a bases de datos diferentes. Estas tablas agrupadas deben tener la misma estructura que se muestra a continuación en la Figura 1.3.

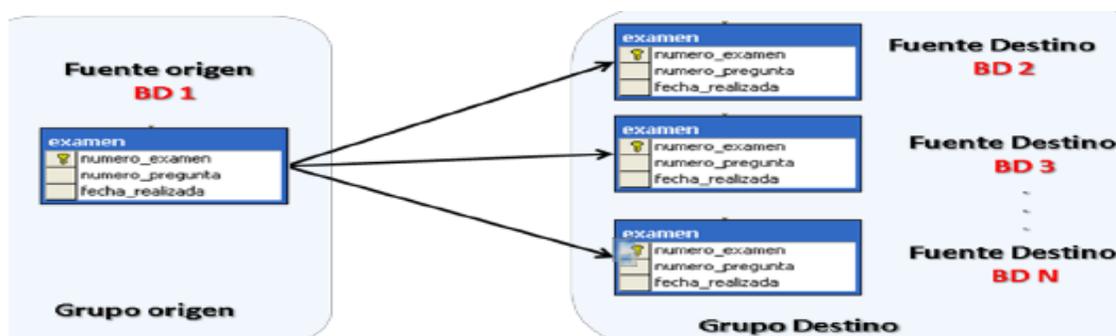


Figura 1.3. Agrupación de Fuentes desde un origen a uno o múltiples destinos.

- **De múltiples orígenes a múltiples destinos.**

El grupo origen contiene un conjunto de fuentes de datos relacionadas o no, entre sí y el grupo destino contiene otro conjunto de fuentes de datos de igual forma. Existe una función entre ambos grupos, a cada fuente origen le corresponde una fuente destino, de

manera que puedan sincronizarse entre sí homológamente. Este criterio es utilizado para configurar réplicas de bases de datos completas, en ese caso cada base de datos representaría un grupo de fuentes y se configuraría la réplica entre ellas. Esto se muestra a continuación en la Figura 1.4.

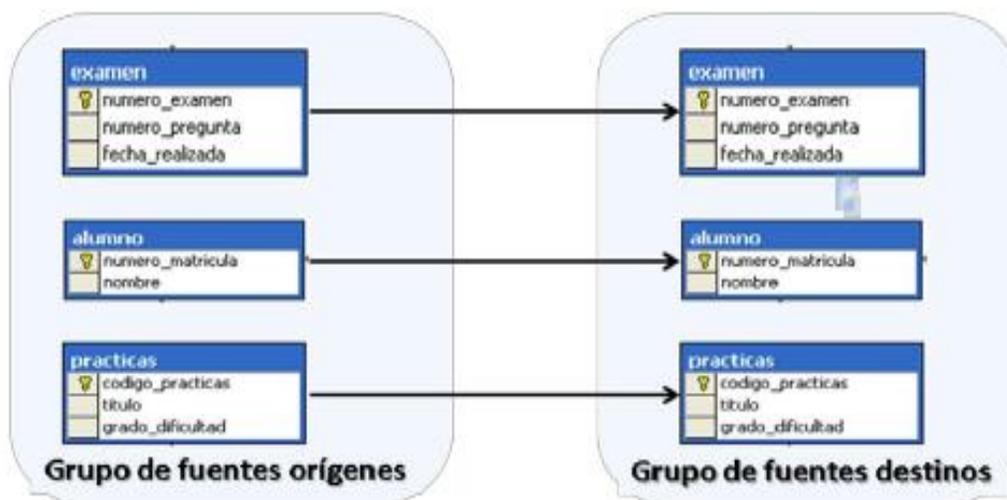


Figura 1.4. Agrupación de Fuentes desde múltiples orígenes a múltiples destinos.

Los grupos son una estructura lógica que se utiliza para facilitarle al usuario el proceso de configuración, lo que significa que agrupar varias fuentes de datos no genera cambios estructurales ni relacionales en estas. El usuario puede crear grupos, eliminarlos, modificar sus atributos, gestionar las fuentes de datos contenidas e incorporarlas a una sincronización cumpliendo la función de grupo origen o destino.

1.3.2.1.3. Configuración básica del sistema para usuarios inexpertos.

Permite la configuración básica de sincronizaciones de fuentes de datos, esta no incluye resolución de conflictos personalizados, solo se utilizan los propuestos por el sistema.

1.3.2.1.4. Configuración avanzada del sistema para usuarios expertos.

Garantiza la configuración total de procesos de réplica y permite, además, la personalización de soluciones de conflictos.

1.3.2.1.5. Validación de la configuración.

AdminChronos valida la configuración establecida, lo que reduce un gran volumen de errores que se pueden cometer en el diseño de la réplica de datos para un escenario determinado y que no necesariamente estén asociados al intelecto del administrador, sino a lo tedioso que puede resultar este trabajo.

1.3.2.1.6. Módulo para la administración de Motores de Procesamiento de Sentencias

Mediante AdminChronos el usuario tendrá la opción de administrar el comportamiento de los MPS que ha instalado en el clúster de bases de datos. Esta funcionalidad puede realizarse de forma remota garantizando que exista una conexión hasta el MPS en cuestión. Este módulo permite terminar, reiniciar o recargar la configuración durante el procesamiento de cualquiera de los MPS que existen funcionando en el clúster de bases de datos.

1.3.2.1.7. Sistema de seguridad y gestión de claves de usuario.

Para tener acceso a AdminChronos es únicamente con una cuenta de administración. Aquí el usuario va a tener la opción de gestionar su información personal, así como la cuenta de acceso a la aplicación.

1.3.2.2. Instalación gráfica y sencilla.

La instalación del sistema se realiza a partir de una interfaz gráfica que instala el MPS, copia las librerías dinámicas utilizadas, monitoriza la instalación de paquetes necesarios para el funcionamiento del sistema y asegura el estado correcto del clúster de bases de datos para comenzar la configuración.

Es completamente independiente a los componentes del sistema Chronos y permite la instalación de los mismos en todos los nodos de forma remota, sin necesidad de repetir el procedimiento manualmente en cada uno de ellos, lo que ahorra tiempo de instalación de la aplicación y disminuye el margen de errores.

1.3.2.3. Tipos de Replicación.

Chronos puede realizar replicación asíncrona multi-maestro de manera instantánea y programada:

1. La replicación instantánea ocurre cuando el intervalo de tiempo entre la confirmación (commit) de una transacción SQL y el momento en que se realiza la réplica hacia el resto de las fuentes de datos es muy pequeño. Es utilizada cuando se desea obtener la sincronización del sistema en el menor tiempo posible.
2. La replicación programada se utiliza para sistemas donde la sincronización de los datos no tiene que ser inmediata y puede realizarse en determinados momentos en los que el clúster de bases de datos no esté recibiendo tanto volumen de carga. Esta puede ser

configurada con una periodicidad horaria, diaria, semanal, mensual o cada N horas. Esta es una de las características del sistema que permite su flexibilidad para escenarios donde existan estaciones de trabajo locales que no necesiten de constante actividad con los servidores centrales.

1.3.2.4. Réplica de transacciones por filtros.

El usuario puede decidir qué información necesita replicar hacia los nodos destinos, estableciendo un filtro para la réplica de las transacciones desde una fuente origen hacia una fuente destino. En este caso debe especificar la condición de réplica para cada par de fuentes sincronizadas.

1.3.2.5. Resolución de conflictos personalizados.

El sistema ofrece un conjunto de soluciones estándares muy frecuentes para los conflictos que pueden surgir durante el proceso de sincronización de los datos. Sin embargo cuando estos no satisfacen las necesidades del usuario, este puede definir sus propios métodos a partir puntos de fusión que tiene el sistema. Esta característica permite la flexibilidad de Chronos para escenarios específicos de réplica.

1.3.2.6. Métodos de Sincronizaciones.

Las sincronizaciones entre fuentes de datos definen cómo se realizará la réplica entre una fuente origen y su destino. Pueden ser establecidas de tres tipos diferentes:

1. Adicionar Diferencias: Este método establece que se adicionarán sólo las diferencias (tuplas) de una fuente de datos origen hacia una fuente de datos destino. Se réplica desde el origen, solo los datos que no están en el destino.

2. Intercambio: Este método establece que se replicarán los datos de una fuente origen hacia una fuente destino y a su vez los datos desde la fuente destino hasta el origen. Cuando se realiza el Intercambio entre dos fuentes se requiere necesariamente de al menos un método de resolución de conflictos que permita a Chronos tomar una decisión en caso de que ocurra una anomalía.

3. Copia Total: Este método establece una copia total de una fuente de datos origen hacia una fuente destino. Los datos de la fuente destino se pierden y son reemplazados por la tabla origen. Suele ser bastante costoso para la réplica instantánea y es muy utilizado para recuperaciones de fuentes de datos que inician desde cero o adición de nuevas fuentes a sincronizaciones ya establecidas.

1.3.2.7. Administración de conexiones.

El MPS almacena las conexiones que está utilizando en un repositorio (pool). Cuando recibe una solicitud de conexión verifica la existencia de una con los mismos parámetros. En caso de que exista la reutiliza, sino la crea y almacena. Esta técnica es muy utilizada porque permite el uso eficiente de las mismas y evita que algún SGBD alcance el límite de conexiones posibles.

1.3.2.8. Recuperación automática ante fallos.

Ante la caída de un nodo el MPS es capaz de detectarlo. Cuando se restablece el servicio, Chronos inicia una recuperación automática de las bases de datos inconsistentes, sin necesidad de intervención del usuario, logrando la convergencia de los datos respecto a sus nodos orígenes.

1.3.2.9. Monitorización.

La interfaz de administración incluye un módulo de monitorización del clúster de bases de datos que aporta información al usuario sobre los eventos de réplica que tienen lugar. Esta información es recopilada por el MPS durante las actividades de réplica. El módulo de monitorización incluye el componente siguiente:

1. Buzón de mensajes para reportes y avisos del sistema que permite la visualización y filtrado de los eventos ocurridos durante la réplica, interactuando con el usuario. Los mensajes se clasifican en el orden de prioridad por la cual necesitan ser atendidos y contienen la información necesaria para la toma de decisiones al respecto por parte del administrador del sistema.

1.3.2.10. Módulo de ayuda.

El módulo de ayuda ofrece información de todas las características del sistema, sus configuraciones y buenas prácticas para proveer al usuario de la información suficiente para la manipulación de la herramienta.

1.3.2.11. Sistema de almacenamiento con bases de datos embebidas.

El MPS utiliza para la persistencia de la información que manipula el sistema de almacenamiento de la base de datos de Berkeley lo que garantiza la persistencia del estado de las transacciones cuando el MPS termina sus servicios. De esta forma puede comenzar la recuperación ante fallos exactamente en el punto en que se interrumpió la réplica. Las bases de datos de Berkeley permiten el almacenamiento y acceso a la información de forma rápida, simple y muy segura.

1.3.2.12. Administración del MPS de forma local.

El MPS inicia sus servicios automáticamente al reiniciarse el nodo servidor en el que reside. Existe un administrador local que interactúa con el MPS por Shell, para que los usuarios puedan además realizar tareas básicas de administración desde la consola, en caso de que el administrador en línea no esté disponible o sea un servidor dedicado carente de interfaz gráfica de usuario. El usuario puede reiniciar, terminar o recargar la configuración de los MPS residentes en el clúster de base de datos. (Chronos)

1.4. Herramientas para desarrollar instaladores de sistemas de software en plataformas Linux (Debian).

En la actualidad existen diversas herramientas capaces de desarrollar instaladores de sistemas de software en Linux Debian. Según un estudio detallado se escogieron las que se muestran a continuación:

1.4.1. BitRock InstallBuilder.

BitRock InstallBuilder crea fácilmente instaladores multiplataforma para los sistemas operativos Linux, Windows, Mac OS X, FreeBSD, OpenBSD, Solaris, IRIX, AIX, y HP-UX. La aplicación genera instaladores nativos que tienen un look-and-feel con el sistema operativo en el que se instala, es decir que se integrará en su entorno gráfico y funcional sin necesidad de dependencias. Adicionalmente, BitRock InstallBuilder contiene hasta 16 idiomas. Además incorpora instaladores autónomos, e incluye una herramienta de instalación que genera paquetes RPM independientes.

El programa genera ficheros de instalación ejecutables optimizados en tamaño y velocidad, independientemente de factores externos, que pueden ejecutarse con interfaz de usuario, texto o modo silencioso. Tiene una licencia libre totalmente funcional para los proyectos de código abierto.

De forma predeterminada, InstallBuilder realiza una copia de seguridad de todos los archivos sobrescritos durante la instalación, para que puedan ser recuperados, en caso de un error.

Está optimizada en tamaño y velocidad. Esto reduce la descarga, el inicio y tiempo de instalación.

Permite a los usuarios elegir los componentes a instalar, lo que garantiza la reutilización de componentes a través de instaladores para acelerar el desarrollo. (Inc., 2010)

BitRock InstallBuilder se distingue por:

- Ser independiente de plataformas o versiones.
- Brindar soporte de integración nativa con el escritorio.
- Tener un solo proyecto XML para todas las instalaciones.
- Proporcionar una herramienta de construcción de línea de comando.
- Permitir agregar páginas de guía en la instalación.
- Desplegar cuadros de diálogo durante el proceso de instalación.

1.4.2. Launch4j.

Launch4j es un instalador de código abierto que empaqueta los archivos jar en ejecutables nativos para los sistemas operativos Windows, MacOS, Linux y Solaris. Crea los lanzadores para los jars y los archivos de clase sin envoltorio. Soporta jars ejecutable y la resolución classpath dinámica utilizando variables de entorno y comodines. Posee un icono de la aplicación personalizada con múltiples resoluciones y

profundidades de color. Soporta interfaz gráfica de usuario y aplicaciones de consola. Permite establecer el inicial tamaño de almacenamiento dinámico máximo, también de forma dinámica en porcentaje de memoria libre. Contiene un tiempo de ejecución interruptores de línea de comandos para cambiar las opciones de compilado. El acceso a las variables de entorno, el registro y la ruta del archivo ejecutable a través de las propiedades del sistema. (Rasmussen., 2009)

- Establece variables de entorno.
- Tiene la opción para cambiar el directorio actual a la ubicación ejecutable.
- Un directorio bin del JRE se adjunta a la variable de entorno PATH.
- GUI y la interfaz de línea de comandos.
- Peso ligero: 26 KB
- Es gratis y puede ser utilizado para fines comerciales.

1.4.3. InstallJammer.

InstallJammer es una aplicación OpenSource y multiplataforma, destinada a crear interfaces gráficas de instalación para las aplicaciones que se desarrollen con cualquier lenguaje de programación o entorno de desarrollo como Java, C# y VB.Net o Eclipse y Visual Studio.NET.

Es multiplataforma y funciona bajo Windows y la mayor parte de las versiones de UNIX, además de soportar escasamente MacOS X. (Ingeman, 2009)

Entre sus múltiples características se puede destacar que permite una personalización de las instalaciones, así como la edición de los cuadros de diálogo que conformarán el asistente del instalador. Permite el control de temas. Proporciona capacidades para la desinstalación. Es una herramienta bastante adecuada que permite la conclusión de los proyectos de cualquier desarrollador o programador tanto independiente como empresarial. (Ins10)

1.4.4. CheckInstall.

CheckInstall es una brillante herramienta que puede hacer la administración de un equipo Linux un poco más fácil. Para su programación se debe compilar desde las fuentes, da la posibilidad de desinstalar limpiamente el programa sin el riesgo de un sistema inestable. (2004-2009) Además da la posibilidad de instalar estos paquetes también en otras máquinas sin tener que compilar el programa cada vez en cada una de ellas, de hecho se deben considerar posibles dependencias de paquetes, genera un paquete Slackware, Debian o RPM y lo instala con el gestor de paquetes por defecto de la distribución y deja una copia del paquete en el directorio fuente actual o en el directorio normal de almacenamiento. Además, es posible cambiar el directorio normal de almacenamiento dentro del archivo de configuración. La copia guardada puede ser instalada, de hecho bajo consideración de posibles dependencias de paquetes, en otras máquinas en la red sin su compilación. (Knopf, 2005)

1.4.5. Conclusiones del análisis de las herramientas.

El análisis de las herramientas BitRock InstallBuilder, Launch4j, InstallJammer y CheckInstall determinó que estas cumplen varias características que permiten una correcta instalación y configuración del sistema Chronos. BitRock InstallBuilder es independiente de plataformas o versiones, permite crear fácilmente instaladores multiplataforma, proporcionar una herramienta de construcción de línea de comando, agregar páginas de guía en la instalación y soporte de integración nativa con el escritorio. Posee un solo proyecto XML para todas las instalaciones.

Launch4j establece variables de entorno, tiene la opción para cambiar el directorio actual a la ubicación ejecutable. Posee un peso ligero: 26 KB.

InstallJammer es multiplataforma y funciona bajo Windows y la mayor parte de las versiones de UNIX, Proporciona capacidades para la desinstalación, así como la edición de los cuadros de

diálogo que conformarán el asistente del instalador.

CheckInstall administra un equipo Linux de forma sencilla. Instala paquetes en otras máquinas sin tener que compilar el programa, considera las dependencias de paquetes, genera un paquete Slackware, Debian o RPM, desinstala el programa sin el riesgo de un sistema inestable. Da la opción de cambiar el directorio normal de almacenamiento. La copia que este crea puede ser instalada, bajo consideración de posibles dependencias de paquetes, en otras máquinas en la red sin la necesidad de volver a compilar el paquete fuente cada vez.

La herramienta seleccionada fue CheckInstall debido a que satisface los requerimientos y se acomoda más a las necesidades para la instalación de los paquetes necesarios para el funcionamiento del sistema. La utilización de esta herramienta permitirá guardar una copia, teniendo en cuenta las dependencias de los paquetes existentes en el sistema y permitirá la copia de estos paquetes en otras máquinas que estén conectadas por la red. Además con la utilización de esta herramienta se podrá realizar la instalación sin necesidad de compilar el programa en cada máquina y se podrá desinstalar limpiamente el programa sin el riesgo de un sistema inestable.

1.5. Interpretes de comandos.

Parte fundamental de un sistema operativo encargada de ejecutar las órdenes básicas para el manejo del sistema. También se denomina shell. Suelen incorporar características tales como control de procesos, redirección de entrada/salida y un lenguaje de órdenes para escribir programas por lotes o scripts.

Es la interfaz entre el usuario y el sistema operativo. Por esta razón, se le da el nombre inglés "shell", que significa "caparazón".

Shell actúa como un intermediario entre el sistema operativo y el usuario gracias a líneas de comando que este último introduce. Su función es la de leer la línea de comandos, interpretar su

significado, llevar a cabo el comando y después arrojar el resultado por medio de las salidas. Los intérpretes de comandos que se abordarán son:

- Shell.
- Bourne Shell.
- C Shell (csh).
- Trusted C Shell (tcsh).
- Korn Shell (ksh).
- Bourne Again Shell (Bash).

1.5.1. Shell.

Shell o intérprete de comandos que por su utilidad y su gran aceptación, es el más usado tanto en Unix como en GNU/Linux.

La característica más destacable del shell es la versatilidad: facilidad de modificación y de adaptación a las necesidades y preferencias de cada usuario en particular. Además proporciona diversas facilidades al usuario. Redirecciona la salida o la entrada standard. Los metacaracteres, caracteres especiales con significado propio. Posee filtros capaces de gestionar la información de los ficheros. Conexión entre órdenes. Cuando se interconexiónan dos órdenes, la salida de un proceso se convierte en la entrada del siguiente. Así se elimina la necesidad de crear ficheros temporales intermedios. El lenguaje de programación del Shell, permite concatenar órdenes en un fichero de texto ejecutable con permiso de ejecución para automatizar procesos, lo que se conoce con el nombre de explotación Batch o por lotes. (Gite., 1999_2002)

1.5.2. Bourne Shell.

Bourne Shell (sh Bourne Shell) intérprete de comandos desarrollado por Stephen Bourne, es considerada la shell original de UNIX. A pesar de pretender ser un intérprete de comandos interactivo, ganó popularidad como lenguaje de guiones (script). La principal dificultad radica en el diseño de un depósito completamente programable, que puede servir para que los usuarios escriban comandos en una terminal interactiva como interfaz. Es invocado por el comando sh. Actualmente este shell no es el más utilizado, ya que las versiones más recientes de intérpretes de comandos ofrecen recursos de los que carece Bourne shell, como es el caso de la edición de las líneas de órdenes, la recuperación de órdenes emitidas previamente, y los alias para las órdenes de uso común. (1994-2005) Entre los recursos importantes que ofrece Bourne shell están los operadores para la ejecución en segundo plano, o ejecución condicional de órdenes, los enunciados para repetir la ejecución de órdenes, incluida la iteración a lo largo de una secuencia de valores que pueden asignarse a una variable de iteración, las variables sustituibles, posee tres formas de entrecomillado, además de realizar la captura de señales, y ejecución de órdenes específicas cuando ocurre una señal determinada.

1.5.3. C Shell (csh).

Lenguaje de programación conocido como csh, fue desarrollado por Bill Joy y está incluida en todas las versiones de UNIX más recientes. Este lenguaje ha sido considerado más idóneo que Bourne Shell, a pesar de que actualmente no presenta un amplio uso ya que ha sido superado por otros shell. Recibe su nombre debido a la semejanza de sus comandos con el lenguaje de programación C, lo cual hace a la shell más fácil de aprender para los programadores. Al poseer una sintaxis muy parecida al lenguaje C, hace que muchas veces los archivos elaborados para csh no pueden ser ejecutados bajo otro shell dígame bash o ksh. Su indicador es el signo de porcentaje (%). Brinda varias capacidades básicas entre las que se puede encontrar la ejecución de varios trabajos simultáneamente, con o sin la intervención del usuario, puede parar la

ejecución de un trabajo y empezar de nuevo, ejecuta comandos usados previamente, así como personalizar el ambiente a las necesidades personales de cada usuario.

1.5.4. Trusted C Shell (tcsh).

Shell o intérprete de comandos, el cual fue iniciado por Ken Greer en los años 70, continuado por Paul Placeway en los años 80, luego Wilfredo Sánchez (ex líder de ingenieros de Mac OS X) a principio de los años 90 y desde entonces miles de personas le ofrecen mantenimiento alrededor del mundo. Esta shell es una versión mejorada del intérprete de comandos C shell y es ejecutada bajo la orden `tcsh`. En 1984, `tcsh` reemplazó a `csh` como el intérprete de comandos por defecto en FreeBSD, y recientemente, en Mac OS X. Entre los recursos adicionales que ofrece se encuentra la capacidad para editar la línea de órdenes interactivamente, realiza llamadas sencillas de órdenes ejecutadas con anterioridad, las cuales se pueden editar, ofrece también la capacidad para programar la ejecución periódica de una orden, además posibilita la consulta de la documentación sobre una orden cuando es tecleada.

1.5.5. Korn Shell (ksh).

Este intérprete de línea de órdenes para sistemas Linux y Unix. Fue desarrollado por David Korn, es un lenguaje totalmente compatible con Bourne Shell además de incluir elementos del intérprete de comandos C Shell. Es uno de los intérpretes de comandos más notables que han venido surgiendo en los últimos años. Incorpora las mejores cualidades de Bourne Shell y de C Shell. Añade posibilidades de programación avanzada, facilidades aritméticas y mayor rapidez de ejecución. Es invocado por la orden `ksh`. Una de las ventajas más significativas que ofrece este intérprete de comandos es su capacidad de ejecutarse en muchos sistemas diferentes, esto hace posible que programas escritos para un sistema sean ejecutados sin tener que ser cambiados en otros muchos sistemas, además que les hace posible a los programadores que estén

familiarizados con ksh usar nuevos sistemas sin tener que pasar largos períodos de aprendizaje, y usar varios sistemas a la vez sin la confusión causada cuando cada sistema presenta una interfaz distinta. Comparado con otras Shell ksh tiene beneficios para ser usado como lenguaje de comandos y beneficios para ser usado como lenguaje de programación. Uno de los inconvenientes es que se distribuye de forma comercial y es por esta razón que otras Shell han tenido un mayor auge. Entre las características más significativas de ksh se encuentra la edición interactiva de la línea de órdenes, ofrece mejores definiciones de funciones que brindan variables locales y permiten escribir funciones recursivas, además posee la capacidad para cambiar con facilidad de un directorio a otro.

1.5.6. Bourne Again Shell (Bash).

Bash es un intérprete de comandos del proyecto GNU. Este es una versión de Bourne Shell (sh) que incluye nuevas funcionalidades. Es un lenguaje interpretado de programación, que facilita el trabajo a los administradores, para realizar la mayoría de las actividades en el sistema, por lo que es más que una simple consola. Su mayor utilización se ve en los sistemas Unix. Este intérprete incorpora algunas características que lo convierte en uno de los intérpretes de comandos más utilizados a nivel mundial.

Como lenguaje de alto nivel proporciona variables, control de flujo, funciones, control de procesos, redirección de entrada/salida y un lenguaje de orden para escribir programas por script. Incluye características de otros intérpretes de comandos como es el caso de la sintaxis Bourne Shell, el redireccionamiento, comandos y variables de Korn shell y la edición de comandos de C shell los cuales forman parte de la familia de intérpretes de comandos de la Unix. Siendo considerado como el intérprete de comandos más extendido en Linux, posee una sintaxis familiar y asequible para programas externos, propicio para sistemas de autocompilado. El soporte del mismo para funciones permite al sistema acoger un diseño modular fácil de entender, sacando

provechos del mismo permitiendo a los mantenedores de paquetes y a los desarrolladores reconfigurarlo inmediatamente (2000).

1.5.7. Conclusiones del análisis de los intérpretes de comandos.

Luego de realizar un estudio de los diferentes intérpretes de comandos y teniendo en cuenta las características y funcionalidades que brindan cada uno, se escogió Bash para la elaboración del paquete de instalación, ya que es el intérprete de comandos más extendido y utilizado en Linux, posee una sintaxis familiar y asequible al usuario, además de ser un lenguaje de alto nivel apropiado para la realización de operaciones por consola.

1.6. Protocolos de Transmisión de Datos.

Los protocolos de transferencia de datos son formatos estandarizados para transmitir datos entre dos dispositivos. El tipo de protocolo utilizado puede determinar variables como el método de comprobación de errores, el método de compresión de datos y la indicación de fin de archivo. Si todas las redes estuvieran organizadas de la misma forma y todo el software y equipos de las redes se comportaran de forma similar, sólo sería necesario un protocolo para todas las transmisiones de datos. Sin embargo, Internet está formada por millones de redes distintas con una amplia gama de combinaciones de hardware y software. Como resultado, la capacidad de transmitir de forma fiable contenido multimedia digital a los clientes depende de varios protocolos de gran complejidad. Uno de los protocolos empleados para transmitir contenido basado en Linux es rsync. (Int)

1.6.1. Rsync.

Rsync es un protocolo utilizado para respaldar y sincronizar archivos. Puede ejecutarse en muchas variedades de Unix. Una limitación de rsync es que no puede copiar de un sistema remoto a otro sistema remoto. Para poder hacer esto, se debe copiar uno de los remotos a un sistema local, y entonces copiarlo del local al otro sistema remoto. Con rsync se pueden hacer copias de seguridad muy fácil de utilizar, que puede respaldar archivos y directorios rápidamente. Esto se logra a través de una rutina muy inteligente para detectar cuándo los archivos han cambiado para que solamente estos archivos sean seleccionados para copiar. El rsync también utiliza una utilidad de compresión para agilizar el proceso de copiado. (Man10)

Rsync detecta archivos y carpetas que necesitan ser copiados porque uno o más de sus atributos han cambiado (por ejemplo, la fecha/hora de la última modificación, o el tamaño del archivo), en cualquier caso, algo ha cambiado desde el último respaldo. Este proceso de selección es muy rápido.

Cuando el rsync ha terminado de preparar la lista que va a utilizar, la copia de estos archivos cambiados es realizada mucho más rápidamente a causa de una rutina de compresión realizada durante el copiado. El rsync comprime antes de enviar y descomprime en el destino, sobre la marcha. También puede copiar archivos desde un sistema local a otro sistema local, sistema local a un sistema remoto y de un sistema remoto a un sistema local. (Jide)

1.8. Conclusiones parciales.

La alta complejidad y las posibilidades de errores se incrementan a medida que el clúster de base de datos es mayor, en el proceso de instalación de Chronos y su interfaz de administración en los nodos de un clúster de base de datos PostgreSQL, esto constituye una problemática hoy en día. Es por eso que en aras de desarrollar una aplicación informática que facilite este proceso de instalación, en este capítulo se realizó un estudio donde se incluyen diferentes conceptos generales sobre réplica de datos, entorno multi-maestro, replicación asíncrona y escenario de réplica, que ayudarán a interiorizar el tema tratado en el trabajo. Se abordan los componentes y las principales características del sistema de réplica de datos Chronos, además de realizarse un estudio del estado del arte de las diferentes herramientas para la creación de instaladores para plataformas linux, así como de los principales intérpretes de comandos existentes y el protocolo de transmisión de datos. De este estudio resultó que se utilizaría la herramienta CheckInstall y el lenguaje Bash para de esta forma lograr que el instalador logre solucionar el problema que inició esta investigación.

Capítulo 2. Guía de instalación del sistema Chronos V1.0.

2.1. Introducción.

Chronos está compuesto por tres componentes básicamente, el Motor de Procesamiento de Sentencias (MPS), la interfaz gráfica de usuario de administración y configuración y una librería dinámica que captura las transacciones que se realizan en el gestor de base de datos y establece la comunicación con el MPS para las operaciones de réplica. Debido a la necesidad de lograr la instalación viable de Chronos, en el capítulo dos se creó una guía de instalación capaz de resolver dicho problema la cual se divide en tres partes: Paquetes de instalación, tipo de servidores en el sistema Chronos, pasos de instalación.

2.2. Paquetes de instalación.

Debe darse la opción de instalar los paquetes desde un repositorio o desde la aplicación.

Los paquetes son:

PostgreSql 8.4.

- PostgreSql-dev.
- PostgreSql_contrib.
- PostgreSql_client.
- PostgreSql_common.

AdminChronos.

- Apache2.
- Php.

Chronos.

- uuid.
- libconfuse.
- libpq.

2.2.1. PostgreSql 8.4.

PostgreSql es un objeto completamente ofrecido de la gestión de bases de datos relacionales de sistemas. Soporta una gran parte del estándar SQL y está diseñado para que sea extensible por los usuarios en muchos aspectos. Algunas de las características: Transacciones ACID, claves ajenas, vistas, secuencias, subconsultas, disparadores, los tipos definidos por el usuario y funciones, combinaciones externas, multiversión de control de concurrencia. Interfaces de usuario gráficas y enlaces para muchos lenguajes de programación están disponibles también. Este paquete proporciona el servidor de base de datos PostgreSql 8.4.

PostgreSql-dev.

Archivos de desarrollo para PostgreSQL 8.4 del lado del servidor de programación. Archivos de cabecera para la compilación de código de SSI para ligarse a PostgreSQL, por ejemplo, para que desde SQL las funciones del lenguaje de programación C sean llamadas.

PostgreSql-contrib.

El paquete contrib de PostgreSQL proporciona varias características adicionales para la base de datos PostgreSQL. Esta versión está diseñada para trabajar con el paquete de servidor PostgreSQL-8.4. Contrib, a menudo sirve como banco de pruebas de características antes de que se adopten en PostgreSQL.

PostgreSql-client.

Este paquete contiene programas de cliente y administrativas para PostgreSQL: estos son el cliente de terminal interactivo psql y de programas para crear y eliminar usuarios y bases de datos.

Este es el paquete de cliente para PostgreSQL 8.4. Si instala PostgreSQL 8.4 en una máquina independiente, necesita el paquete del servidor PostgreSQL-8.4. En una red, usted puede instalar este paquete en muchas máquinas cliente, mientras que el paquete de servidor se puede instalar en una sola máquina.

PostgreSql-common.

El paquete PostgreSql-common proporciona una estructura en que múltiples versiones de PostgreSql puede ser instalado y / o múltiples grupos mantiene a la vez. Este paquete creará automáticamente un certificado para que se admitan las conexiones SSL.

2.2.2 AdminChronos.

Apache2.

El servidor HTTP Apache es un servidor web de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. El servidor Apache se desarrolla dentro del proyecto Servidor HTTP de la Fundación de Software Apache. Presenta entre otras características altamente configurables, bases de datos de autenticación y negociado de contenido, no posee una interfaz gráfica que ayude en su configuración.

El objetivo de la Fundación de Software Apache es crear un entorno seguro, eficiente y un servidor HTTP extensible como compatible con los estándares de software de código abierto. El resultado ha sido de largo el número uno de servidores web en Internet.

Cuenta con soporte para HTTPS, hosting virtual, CGI, SSI, IPv6, fácil de secuencias de comandos y la integración de bases de datos, solicitud respuesta de filtrado, esquemas de autenticación flexible, y más.

Php5.

Del lado del servidor, embebido en HTML lenguaje de scripting (metapaquete). Este paquete es un metapaquete que, cuando se instala, garantiza al menos un servidor de las tres versiones de parte del intérprete de PHP5 instalado. La eliminación de este paquete no eliminará PHP5 de su sistema, sin embargo, que puede eliminar otros paquetes que dependen de éste. PHP5 HTML es un lenguaje de scripting embebido. Gran parte de su sintaxis es tomada de C, Java y Perl con un par de PHP únicas características específicas. El objetivo del lenguaje es permitir a los desarrolladores web escriban de forma dinámica páginas generadas rápidamente.

2.2.3. Chronos.

Uuid.

UUID son números de 128 bits que van a tener una alta probabilidad de la singularidad en el espacio y el tiempo y son computacionalmente difíciles de adivinar. Ellos son los identificadores únicos globales que pueden ser localmente generados sin ponerse en contacto con una autoridad de registro global. UUID están concebidos como identificadores únicos para los objetos de masa de etiquetado con una vida muy breve y de manera fiable la identificación de muchos objetos persistentes a través de una red.

LibConfuse.

LibConfuse es una librería desarrollada en C que permite analizar de forma fácil archivos de configuración. Soporta las secciones y listas de valores (cadenas, enteros, reales, booleanos o de otras secciones), así como algunas otras características tales como: Una o dos cadenas entre

comillas, la expansión de variable de entorno, y declaraciones. Se utiliza para parsear el fichero de configuración Chronos.conf desde el MPS.

Libpq.

Libpq es la interfaz para los programadores de aplicaciones en C para PostgreSQL. Libpq es un conjunto de rutinas de biblioteca que permiten a los programas cliente trasladar consultas al servidor de Postgres y recibir el resultado de esas consultas. Libpq es también el mecanismo subyacente para muchas otras interfaces de aplicaciones de PostgreSQL, incluyendo libpq++ (C++), libpqtc (Tcl), Perl, y ecpg. Algunos aspectos del comportamiento de libpq. Los programas cliente que utilicen libpq deberán incluir el fichero de cabeceras libpq-fe.h, y deberán enlazarse con la biblioteca libpq.

2.3. Tipo de servidor en el sistema Chronos.

➤ En la réplica:

- **Origen:** Es donde se van a hacer las modificaciones en el sistema gestor de bases de datos, y las cuales serán capturadas por la librería libtransacciones.so y serán replicadas al servidor destino.
- **Destino:** Es el servidor donde se reciben los cambios o modificaciones que se hicieron en el servidor origen

2.4. Pasos de la Instalación.

Este procedimiento debe realizarse para cada uno de los nodos del clúster de base de datos.

1. Seleccionar la opción de Paquetes de Instalación por repositorio o desde la aplicación.

Para lograr la instalación de Chronos, Debe darse la opción de instalar los paquetes desde un repositorio o desde la aplicación. De un repositorio, la aplicación se conecta al repositorio y se utiliza el comando `apt-get install nombredelpaquete` para lograr la instalación y desde la aplicación, los paquetes deben descargarse en un directorio desde el repositorio de Debian. La carpeta que contiene los paquetes de instalación forma parte del instalador.

2. Instalar todos los paquetes de la sección 2.2.

3. Configuración.

Se necesita configurar los archivos **pg_hba.conf** y **PostgreSql.conf** ubicados en *etc/PostgreSql*. La configuración de estos archivos permite que el servidor de base de datos PostgreSQL recepcione peticiones solamente de las direcciones IP que se configuren en **pg_hba.conf**. Es necesario solicitarle al usuario la dirección IP y la máscara de subred de cada uno de los nodos del clúster de base de datos. Y así permitir copiar por cada nodo la línea que se muestra en la Figura 2.1 en el archivo `pg_hba.conf`.

	<i>host</i>	<i>all</i>	<i>all</i>	<i>IP/máscara</i>	<i>trust</i>
Ejemplo:	<i>host</i>	<i>all</i>	<i>all</i>	<i>10.32.12.202/32</i>	<i>trust</i>

Figura 2.1. Ejemplo de configuración del archivo `pg_hba.conf`.

Esto se resuelve con el script, pidiéndole al usuario por la consola el IP y la máscara de subred, se utiliza el comando **sed** que posibilita editar archivos de configuración, permitiendo al usuario una conexión completa del clúster de la réplica Chronos.

El archivo *Postgresql.conf* se cambia el parámetro de `listen_addresses = 'Localhost'` a `listen_addresses = '*'` para especificar que escuchará peticiones desde todas las direcciones IP existentes desde *pg_hba.conf*. Estos cambios requieren reiniciar el servicio PostgreSQL.

Este instalador también debe dar la opción de copiar los componentes para la correcta ejecución de Chronos donde entre estos aparece la librería `libTransaccionesSQL.so` que se debe copiar en el directorio `/usr/lib/PostgreSQL/8,4/lib`. `libTransaccionesSQL.so` es una librería de enlace dinámico que extiende la funcionalidad del gestor capturando los cambios que se producen en cualquiera de las tablas contenidas, los cuales deben ser replicados según el tipo de sincronización que se configure, captura las transacciones que se realizan en el gestor de base de datos y establece la comunicación con el MPS para las operaciones de réplica. Está programada en el lenguaje de programación C y permite la extensión de las funcionalidades del servidor para capturar campos.

Se le da permisos en el directorio a la librería con el comando:

`chown postgres /usr/lib/PostgreSQL/lib/libtransaccionesql.so`, luego de esto se reinicia PostgreSQL: **`/etc/init.d/PostgreSQL restart`**.

Para la correcta comunicación entre los nodos del clúster se debe copiar en el archivo hosts en etc/hosts para cada uno de los nodos implicados en el clúster de bases de datos la siguiente línea: (**10.32.20.201 eipad1.uci.cu eipad1**).

AdminChronos es la aplicación WEB de administración, configuración y monitorización la cual debe ser copiada en la dirección **/var/www/AdminChronos**.

Copiar en el directorio */etc/* la carpeta Chronos con el archivo de configuración chronos.conf. Esta carpeta Chronos con el archivo **chronos.conf** debe estar en el instalador también.

Se ejecuta en terminal de postgres, se instala el lenguaje de programación plpgsql con el comando (CREATE LANGUAGE 'plpgsql'), para las funciones de los procedimientos almacenados de las Bases de Datos. Se instala el lenguaje de programación C con el comando (CREATE LANGUAGE 'c'). Es necesario para la ejecución de la librería libtransaccionesql.so.

2.5. Conclusiones parciales.

El capítulo dos contiene la guía de instalación del sistema de réplica de datos Chronos. Se concluye que es necesaria la instalación de cada uno de los paquetes del sistema con sus respectivas dependencias. Se enuncia el tipo de servidor y se explican exhaustivamente cada uno de los pasos a seguir en el proceso de instalación, abordando la configuración de archivos como **pg_hba.conf** y **PostgreSql.conf** que permiten que el servidor de base de datos recepcione las peticiones de las direcciones IP configuradas por el usuario, entre otros permisos. También se describen como deben ser instalados los lenguajes de programación que Chronos necesita para su total ejecución. Además del directorio donde deben ser copiadas las librerías, la aplicación web y otros archivos que aparecen especificados en el capítulo.

Capítulo 3: Presentación del paquete de instalación de Chronos, ChronosInstall 0.3.

3.1 Introducción.

Una instalación exitosa es una condición necesaria para el funcionamiento de cualquier software. Mientras más complejo sea el software, más archivos contenga, mayor sea la dispersión de los archivos y la interdependencia con otros software, mayor es el riesgo de alguna falla durante la instalación. Si la instalación falla aunque sea solo parcialmente, el fin que persigue la instalación posiblemente no podrá ser alcanzado. Por esa razón, sobre todo en casos de software complejo, un proceso de instalación confiable y seguro es una parte fundamental del desarrollo del software. Con el fin precisamente, de realizar una aplicación informática que facilite el proceso de instalación del sistema de réplica de datos Chronos, es que se desarrolló el instalador ChronosInstall 3.0. A continuación, en el presente capítulo se expone una descripción general de la propuesta, se explica cómo fue el proceso mediante la utilización de la herramienta CheckInstall, luego se aborda la creación e instalación del paquete DEBIAN. Finalmente se procede a la validación de la aplicación mediante casos de prueba.

3.2 Descripción general de la herramienta.

El instalador que se realizó proporciona facilidad a la instalación y configuración del sistema de réplica de base de datos Chronos. Con este instalador se elimina el engorroso trabajo de instalar de forma manual dicha herramienta y se impide posibles errores que puedan ocasionarse durante la instalación. Chronos contiene un motor de procesamiento (MPS), una librería, la aplicación web y tres paquetes de instalación (PostgreSQL 8.4, AdminChronos y Chronos) cada uno con sus dependencias y necesita de un Script para la configuración de archivos. La herramienta debe realizar la instalación de dichas partes de Chronos de forma sencilla y fácil de usar para todos los usuarios que interactúen con el sistema de réplica de datos Chronos.

3.3 Aplicación de la herramienta CheckInstall en la solución propuesta.

Al utilizar la herramienta CheckInstall esta ofrece la opción de considerar posibles dependencias de paquetes, instala estos paquetes también en otras máquinas sin tener que compilar el programa. Dicha herramienta genera tres tipos de paquetes (Slackware, Debian o RPM), el paquete que se generó al hacer el instalador del sistema Chronos fue Debian (**.deb**). El paquete **.deb** da la opción de instalar y desinstalar fácilmente.

3.4 Creación del paquete DEBIAN.

1. Se establece una jerarquía de carpetas, donde la primera tendrá el nombre del instalador, dentro de esta se creará otra carpeta llamada **DEBIAN**, esto se muestra en la Figura 3.1. El nombre de la carpeta **DEBIAN** debe estar escrito con letras mayúscula obligatoriamente debido a que es una carpeta reconocida por la herramienta CheckInstall, dentro de **DEBIAN** se crean dos archivos, el primero archivo es **control**, dentro del cual aparecen una serie de datos que se deben llenar para lograr que se ejecute el instalador, los datos son los siguientes:

Source: Nombre de la aplicación.

Section: Tipo de sección (net).

Priority: Prioridad (extra).

Maintainer: Nombre y correo del creador.

Version: Versión.

Package: Nombre del paquete.

Architecture: Arquitectura (i386).

Depends: Posibles dependencias de los paquetes.

Description: insertar una pequeña descripción.

Insertar una descripción más detallada.

El más importante de los datos a llenar, es **Depends**, que es donde se van a poner las posibles dependencias que posea lo que se quiere instalar.

El otro archivo dentro de la carpeta es **postinst** que es el encargado de almacenar el script, esto da la posibilidad de que este se ejecute.

Para copiar los archivos necesarios de la aplicación a instalar, CheckInstall ofrece la opción de crear carpetas hermanas de DEBIAN, las cuales contendrán el contenido de lo que se desee instalar, pero de la siguiente forma:

Ejemplo. Si se desea copiar la librería libtransaccionesql.so en directorio /usr/lib/PostgreSql/8,4/lib, será como se muestra a continuación en la Figura 3.2 y dentro de la última carpeta se copia la librería y de igual forma con todo lo que se requiera copiar.



Figura 3.1 Jerarquía de carpetas de ChronosInstall 0.3.

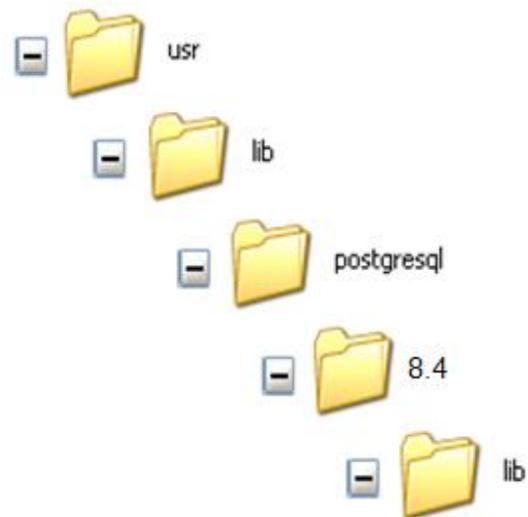


Figura 3.2 Copiar en un directorio.

8.4

2. Luego de haber realizado lo antes expuesto se prosigue a la creación del (.deb) .Se copia ChronosInstall 0.3 y en la consola de Linux se ejecuta el comando `dpkg-deb -b ChronosInstall 0.3` y a continuación se creará el paquete que se muestra en la Figura 3.3.



Figura 3.3 Paquete Debian.

3.5 Realización de la instalación del paquete Debian (.deb).

El primer paso para la instalación del paquete comienza cuando se ejecuta con la herramienta GDebi seguidamente se muestra la pantalla de la Figura 3.4 donde se observa una descripción de la herramienta a instalar.

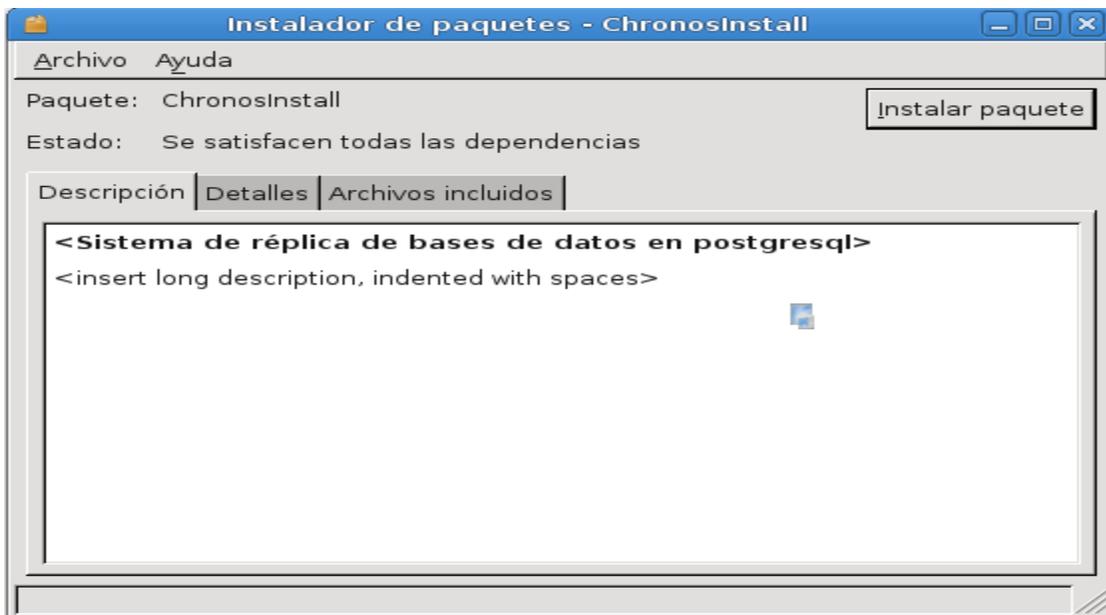


Figura 3.4 Instalador de paquetes. Vista de descripción.

Esta pantalla también muestra una vista de los detalles de la instalación esto se expone a continuación en la Figura 3.5.

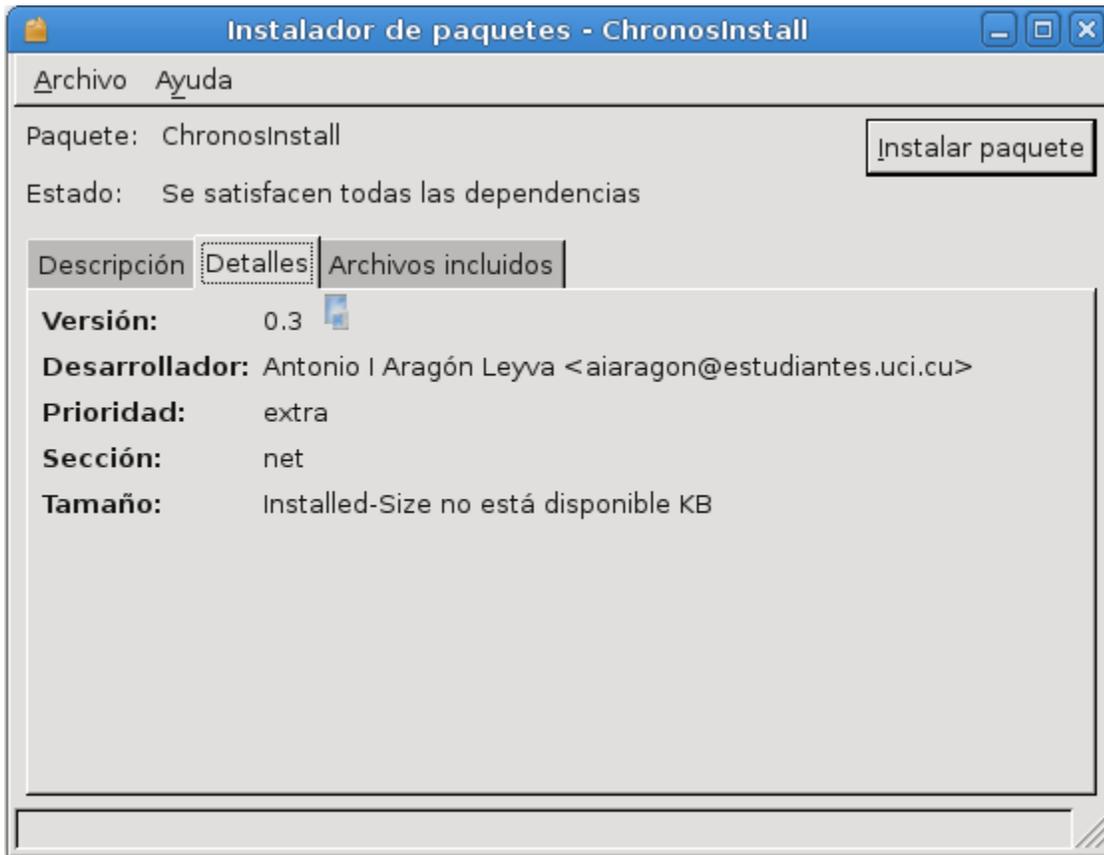


Figura 3.5 Instalador de paquetes. Vista de detalles.

En la Figura 3.6 se muestran los datos de los archivos que se incluyeron para la instalación, donde se observan las dependencias que se insertaron, la configuración del archivo **control** y **postinst**, además de los archivos que se desean copiar y en el directorio que se va a realizar dicha copia.

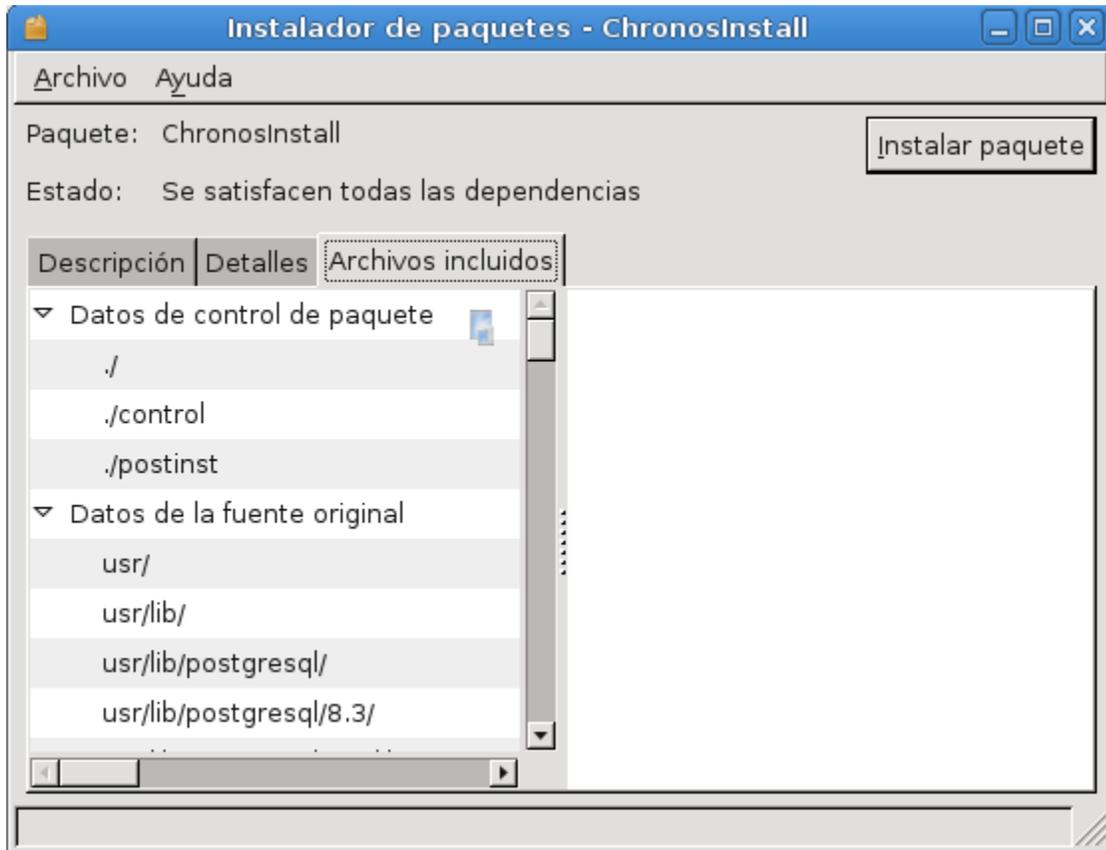


Figura 3.6 Pantalla de archivos incluidos.

A continuación se procede a instalar el paquete en la Figura 3.7 y comienza la instalación dónde lo primero que sucede es la verificación de las dependencias en el synaptic, en caso de no estar instalada ninguna dependencia se tendrá que instalar desde la consola y comenzar la instalación nuevamente. Se copian todos los archivos y se procede a abrir el terminal de la aplicación como se muestra en la Figura 3.8 donde aparece el script para la configuración de los archivos `pg_hba.conf`, `PostgreSql.conf`, en el cual se le solicita al usuario el IP (Figura 3.9) y la máscara de subred de las máquinas que se van a conectar en el clúster (Figura 3.10), además de crear los lenguajes (`plpgsql` y `C`). Después de haberse configurado, finalizado el script e instalado el paquete, el usuario puede salir oprimiendo el botón cerrar (Figura 3.11).



Figura 3.7 Pantalla de instalación del paquete.

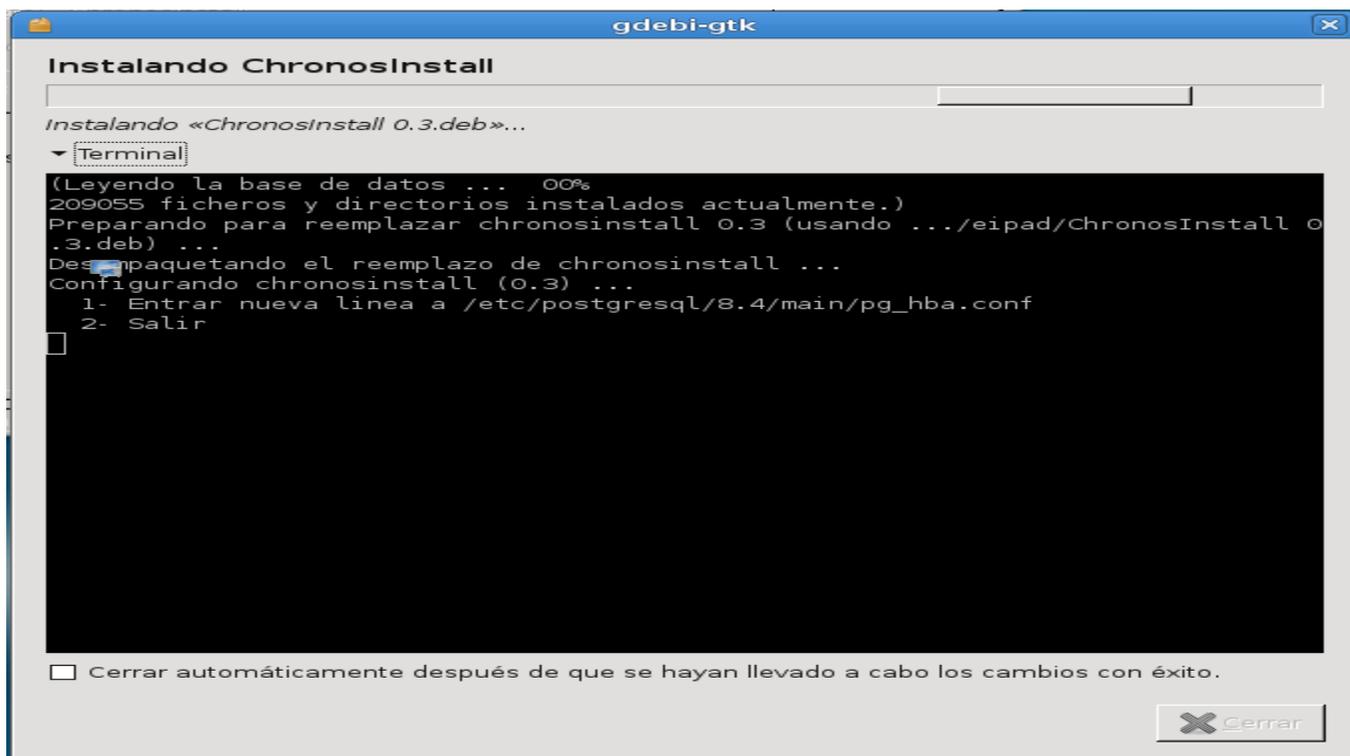


Figura 3.8 Representación del script en el terminal de la aplicación.

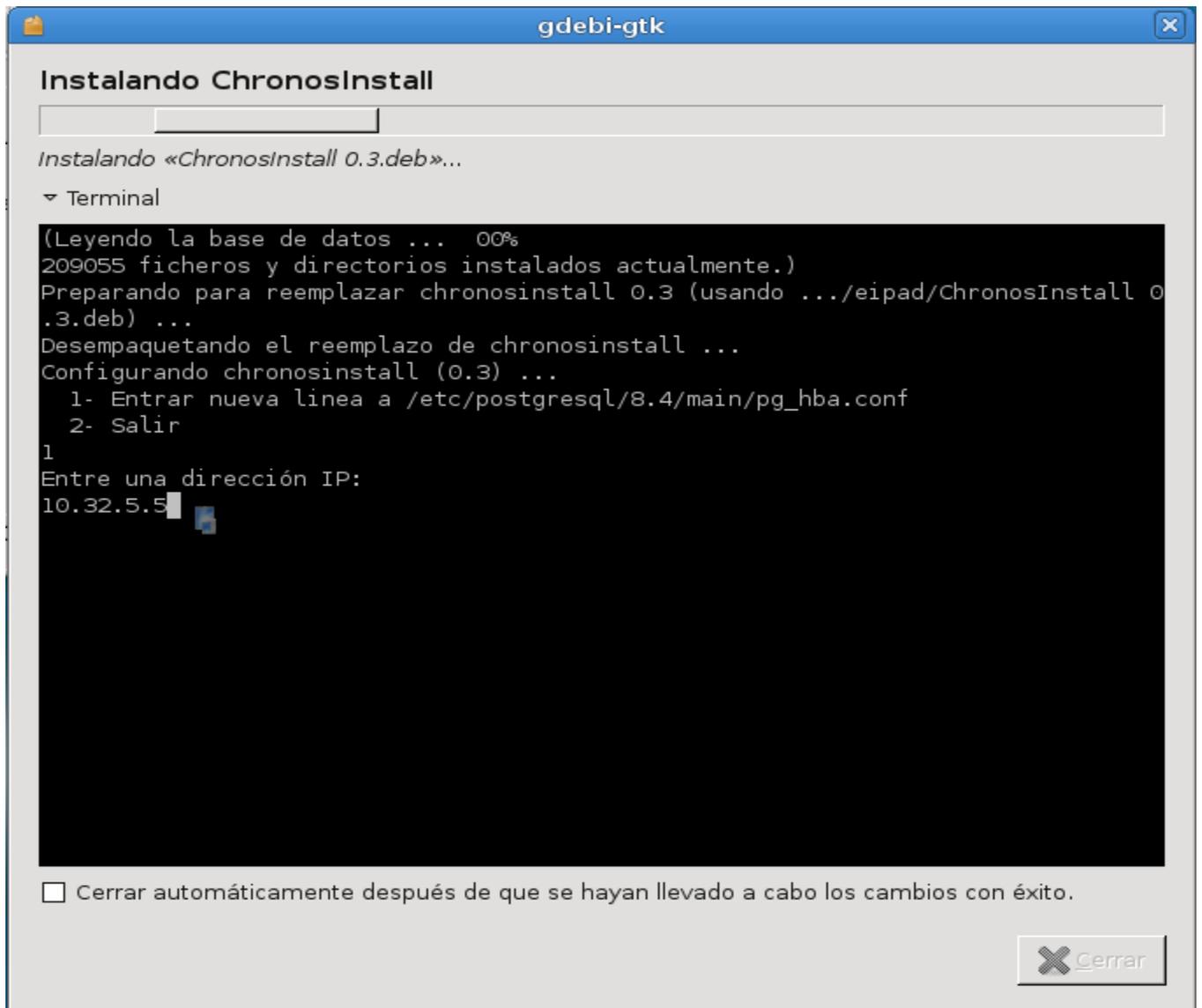


Figura 3.9 Entrada del usuario de la dirección IP.

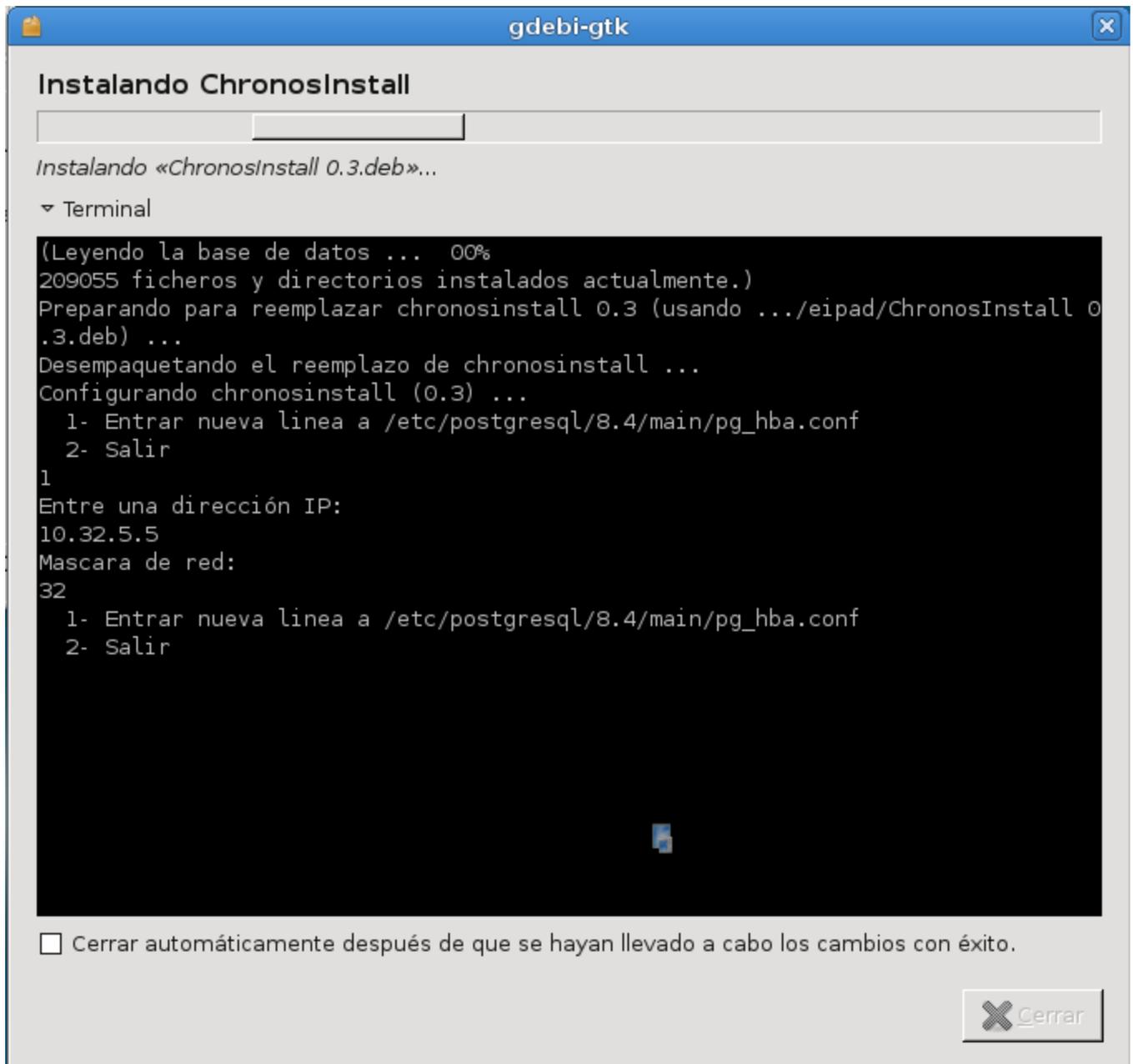


Figura 3.10 Entrada del usuario de la máscara de red.

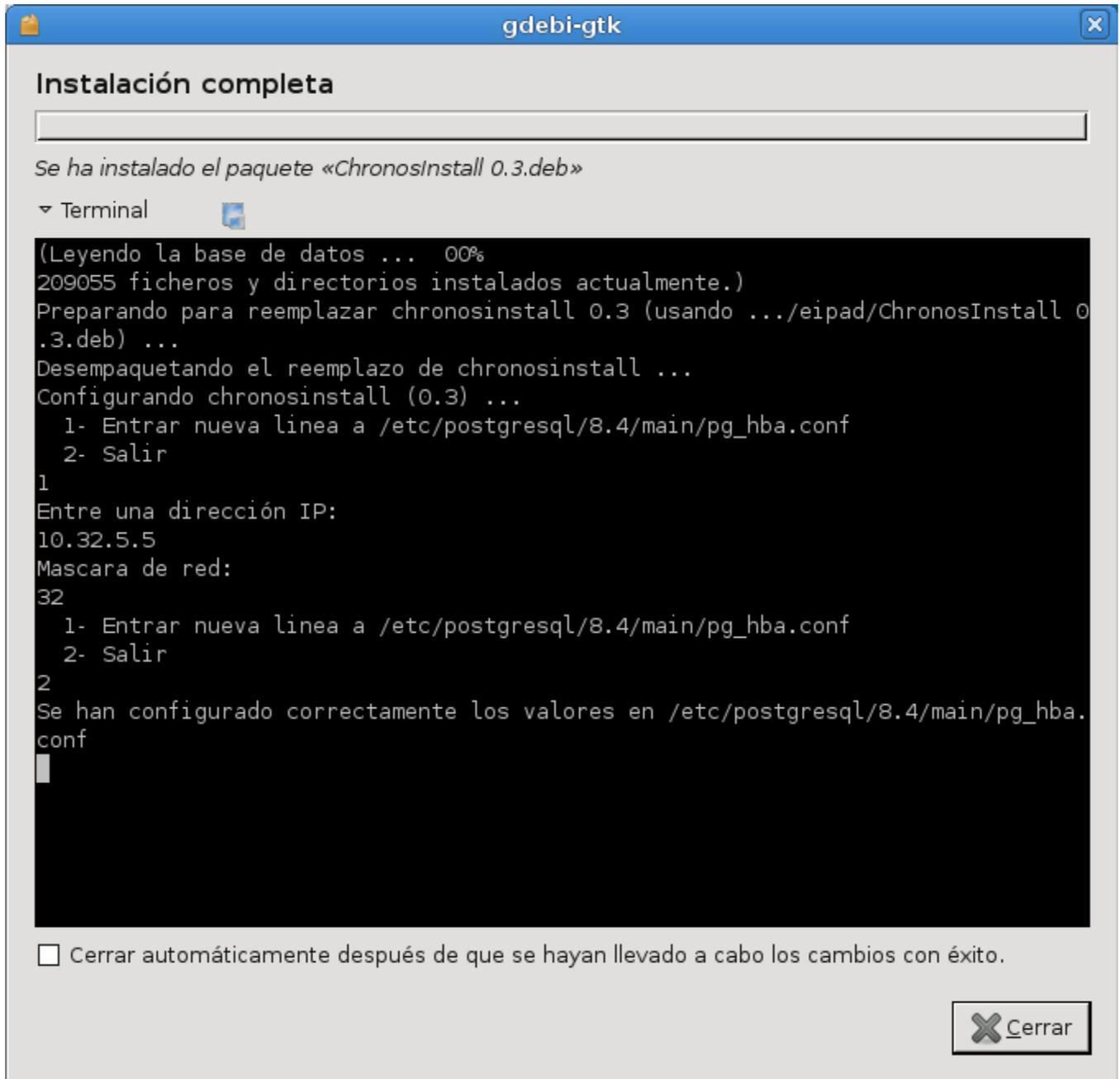


Figura 3.11 Fin de la configuración y ejecución del script dando lugar a la instalación completa del paquete.

3.6. Validación de la solución propuesta mediante casos de prueba.

La solución propuesta se validó mediante casos de prueba. Un caso de prueba es una especificación, usualmente formal, de un conjunto de entradas de prueba, condiciones de ejecución y resultados esperados, identificados con el propósito de hacer una evaluación de aspectos particulares de un elemento objeto de prueba.

- Los casos de prueba reflejan trazabilidad y funcionalidad, ya que estos muestran una secuencia ordenada de eventos, al describir flujos, precondiciones y postcondiciones.
- Las especificaciones suplementarias de requerimientos ya que existen otras características de calidad a evaluar, además de la funcionalidad, como usabilidad, confiabilidad, eficiencia, mantenibilidad y portabilidad.
- Las especificaciones de diseño del Sistema, ya que se debe verificar que el software fue implementado según el diseño y que los elementos arquitectónicos garantizan la calidad del software.

Esto garantiza que los procedimientos de pruebas sean compatibles con las necesidades de los usuarios. (Edumilis Méndez, 2007)

A continuación se muestran la aplicación de tres casos de pruebas del instalador ChronosInstall 0.3.

3.6.1. Casos de prueba.

Descripción general.

- Instalar el sistema de réplica de datos Chronos. En el cual se verifican todas las dependencias de paquetes, se instalan los componentes de la aplicación y configuran los archivos necesarios.

Condiciones de ejecución.

- Para la ejecución es necesario configurar las dependencias en el archivo **control**.
- Para la instalación de los componentes de Chronos se debe realizar mediante una jerarquía de carpetas con la dirección a la cual se desean copiar.
- Para que el Script se ejecute el usuario debe entrar los datos correctos (IP/máscara). Las otras funcionalidades del script son de forma interna.
- Crear el paquete Debian y debe ser ejecutado con GDebi de forma gráfica o con el comando `dpkg -i` (nombre del paquete).

Secciones.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo central
<i>Verificación de las dependencias 1</i>	<i>EC 1: Instalador de paquetes - ChronosInstall.</i>	<i>La herramienta verifica las dependencias de paquetes.</i>	<i>Ejecutar el paquete de instalación con GDebi</i>
<i>Instalación de los paquetes 2</i>	<i>EC 2: gdebi-gtk</i>	<i>La herramienta comienza la instalación de los paquetes</i>	<i>Ejecutar el botón Instalar paquete</i>
<i>Ejecución del script 3</i>	<i>EC 3: gdebi-gtk</i>	<i>Muestra en una pantalla el Script de configuración de ChronosInstall 0.3.</i>	<i>Ejecutar con el botón Terminal.</i>

Escenarios.

ID del escenario	Escenario	Variable 1	Variable 2	Variable n	Respuesta del sistema	Resultado de la prueba
[EC 1]	EC 1: <i>Instalador de paquetes - ChronosInstall.</i>	V	V	V	<i>El estado de la herramienta debe mostrar: Se satisfacen todas las dependencias.</i>	<i>Se satisfacen todas las dependencias.</i>
		<i>PostgreSQL</i>	<i>PostgreSQL-dev</i>	<i>Apache</i>		
[EC 2]	EC 2: <i>gdebi-gtk</i>	NA	NA	NA	<i>La herramienta emite el mensaje: Instalando ChronosInstall.</i>	<i>Comienza la instalación de ChronosInstall.</i>
[EC 3]	EC 3: <i>gdebi-gtk</i>	V	V	NA	<i>El sistema pide entrar el IP Y La máscara de subred.</i>	<i>Se configuraron correctamente los valores en los archivos que se realiza.</i>
		<i>10.5.5.5</i>	<i>25</i>			

Las celdas de la tabla contienen V o NA. V indica un valor válido y NA no es necesario proporcionar un valor del dato en este caso.

3.7 Conclusiones parciales.

En este capítulo se desea adiestrar al usuario para lograr una familiarización con el instalador, por lo que se describe la creación del paquete de instalación, mostrando las opciones de la herramienta y los pasos a seguir, especificando los aspectos más importantes, en pos de conseguir su correcta ejecución. Además se procede a la validación de la propuesta a través de casos de pruebas. Se puede concluir que ChronosInstall 0.3 proporcionará un proceso de instalación amigable y viable del sistema de réplica de datos Chronos.

Conclusiones.

Con el desarrollo del presente trabajo se logró facilitar el proceso de instalación del sistema de réplica de datos Chronos a partir del seguimiento de las siguientes estrategias:

- Se realizó un estudio detallado sobre el sistema replicación de datos Chronos, describiendo sus componentes, sus principales características y el proceso de réplica que realiza. Esto aportó una serie de definiciones y conceptos importantes a tener en cuenta en dicha instalación. Se realizó además una búsqueda exhaustiva de las herramientas existentes para crear instaladores a nivel mundial, se incluye un estudio de los principales intérpretes de comandos y el protocolo de transmisión de datos Rsync. Identificándose que: La herramienta utilizada en la realización del instalador es CheckInstall y el intérprete de comando usado es Bash.
- Se creó una guía, capaz de resolver la necesidad de hacer menos engorrosa la instalación de Chronos.
- Se validó la solución propuesta mediante casos de prueba y se verificó que la aplicación cumple con las funcionalidades necesarias para una fácil instalación de Chronos.

Se cumplió el objetivo general trazado para este Trabajo de Diploma: Implementar una aplicación para la instalación de los componentes necesarios para el funcionamiento de Chronos en un clúster de base de datos.

Recomendaciones.

Se recomienda:

- Utilizar esta herramienta no solo en este caso sino también para la instalación de otras aplicaciones, debido a la comodidad que proporciona la aplicación, además se puede instalar mediante un entorno gráfico o por la consola.

Bibliografía.

Calibre.org. [En línea] <http://www.cdlibre.org/index.html>.

The Apache Software Foundation. [En línea] <http://www.apache.org/foundation/licence-FAQ.html#Marks>.

The Apache Http Server Project. [En línea] <http://httpd.apache.org/>.

2009. *PostgreSQL Cluster Developers' Meeting, Tokyo, Japan*. [En línea] 19 de noviembre de 2009. http://www.postgresql.jp/events/pgcon09j/e/dev_mtg.

2009. *Petra Link Factory*. [En línea] Febrero de 2009. <http://www.petralinkfactory.com/tag/programas/>.

Osmosis Latina. [En línea] <http://javabasicos.osmosislatina.com/curso/progbasico/jars.htm>.

Manual de sidux. [En línea] <http://manual.sidux.com/es/sys-admin-rsync-es.htm..>

Intérprete de Comandos. [En línea]
http://www.linux10.com.ar/Glosario/terminos/interprete_de_comandos.htm.

Installjammer Multiplatform Installer. [En línea] <http://sourceforge.net/projects/installjammer/>.

2009. *Hypertext Transfer Protoco*. [En línea] diciembre de 2009. <http://www.osalt.com/installjammer>.

2008. *Libros en pantalla de SQL Server 2005*. [En línea] Noviembre de 2008. <http://msdn.microsoft.com>.

NetWork World. [En línea] <http://www.networkworld.es/Replicacion-de-datos-asincrona/seccion-/articulo-130592>.

1990-2010. *Zator Systems*. [En línea] 1990-2010. http://www.zator.com/Cpp/E1_4_4b.htm. Curso C++.

2006 2010. *Linperial International Systems S. A. de C. V.* [En línea] 2006 2010.
<http://www.linperial.com/communities/forums/developers/?q=node/15>.

Java.com. [En línea] <http://www.java.com/es/download/help/5000010500.xml>.

1994-2005. *LINUX Reviews*. [En línea] 1994-2005. http://linuxreviews.org/beginner/bash_GNU_Bourne-Again_SHell_Reference/.

2000. *LWP*. [En línea] 2000.

<http://www.lawebdelprogramador.com/cursos/enlace.php?idp=2982&id=186&texto=Linux/Unix+Shell+Scripting>.

"*Asynchronous Data Replication: A National Integration Strategy for Databases on Telemedicine Network*". **Douglas Dyllon Jeronimo de Macedo, Hilton Ganzo William Perantunes, Rafael Andrade, Aldo von Wangenheim, Mario Antonio Ribeiro Dantas.** 2008. 21st IEEE International Symposium on Computer-Based Medical Systems. págs. 638-643.

Chronos. **Yadisel Gálvez Velázquez, Yadira Lizama Mué, Lissuan Fadruga Artiles, Sergio Hernández Cisneros, Mario Concepción.** Vol. Sistema de Replicación asíncrona multimaestro para PostgreSQL.

D., Leffingwell D. and Widrig. 2006. *Managing Software Requirements, a Use*. s.l. : Second Edition , 2006.

Edumilis Méndez, María Pérez, Luis E. Mendoza. 2007. *Aplicación de un Método para Especificar Casos de Prueba de Software en la Administración Pública*. Caracas : Actas de Talleres de Ingeniería del Software y Bases de Datos, 2007. Vol. 1, No. 4.

Gite., Vivek G. 1999_2002. *Linux Shell Scripting Tutorial*. [En línea] 1999_2002. <http://www.freeos.com/guides/lst/>.

Ingeman, Anders. 2009. *osalt.com*. [En línea] 2009. <http://www.osalt.com/installjammer>.

Jide. *RSYNC*. [En línea] <http://magnifico.wordpress.com/2009/06/03/rsync-el-mejor-sistema-de-copias-de-seguridad/>.

Knopf, Mario. 2005. *Linux Focus.org*. [En línea] 26 de septiembre de 2005. http://linuxfocus.org/castellano/archives/lf-2004_12-0360.pdf. 2.51.

Kowal, Grzegorz. *Executable Wrapper*. [En línea] <http://www.altacracks.com/programas-gratis/launch4j-executable-wrapper>.

Lago, Ramiro. 2005. *Archivos jar*. [En línea] Enero de 2005. http://www.proactiva-calidad.com/java/archivos/leer_archivos_jar.html.

Paarmann, Daniel. 2004 . *Nongnu*. [En línea] 2004 . <http://www.nongnu.org/synaptic/> Synaptic.

2009. PostgreSQL Development Team, High Availability, Load Balancing, and Replication. [En línea] PostgreSQL 8.4.2 Documentation, 2009. <http://www.postgresql.org/docs/current/interactive/high-availability.html> .

Protocolos de transferencia de datos. *Windows Server TechCenter*. [En línea] <http://technet.microsoft.com/es-es/library/cc731604.aspx>.

2008. Ranade, Sandeep D. *Asynchronous Replication*. [En línea] LINUX Journal, 2008. <http://www.linuxjournal.com/article/7265> .

Rasmussen., Anders Ingeman. 2009. [En línea] 2009. <http://www.osalt.com/es/launch4j>.

Schneier, David Wagner and Bruce. 1996. *Analysis of the SSL 3.0 Protocol, The second USENIX Workshop on Electronic Commerce Proceedings* ,. 1996. págs. pp29–40.

SlackWARE. *Linux Argentina*. [En línea] <http://www.slackar.com.ar/>.

2004-2009. Steve Kemp. *Debian Administration*. [En línea] 2004-2009. <http://www.debian-administration.org/articles/147>.

Support, BitRock. 2009. *Bit Rock*. [En línea] 16 de noviembre de 2009. <http://support.bitrock.com/article/installbuilder-action-lists>.

Volkerding, Patrick. 1993. *Slackware*. [En línea] abril de 1993. Slackware: <http://www.slackware.com/info/>..

Glosario de Términos:

Slackware: Sistema operativo de software libre y abierto. Es una de las primeras distribuciones de GNU / Linux. Tiene como objetivo la estabilidad y la simplicidad de diseño, usa archivos de texto plano para la configuración y la toma como algunas modificaciones a los paquetes de software. (Volkerding, 1993) (SlackWARE)

Debian (deb): Es un paquete binario. Contienen código de máquina, no código fuente. El nombre de los paquetes de este tipo: nombre del paquete_version_plataforma.deb. (2004-2009)

RPM: Es un paquete binario. Contienen código de máquina, no código fuente. El nombre de los paquetes de este tipo: nombre del paquete_version_plataforma.rpm. (RPM)

Archivos jar: Los archivos JAR (Java Archive) son la forma sencilla y eficiente de transportar recursos. Con un archivo JAR se facilita la distribución e instalación de una gran variedad de archivos, applets, video, sonido, imágenes, texto, etc. Además un archivo JAR puede contener firmas digitales que aseguran la integridad y autenticidad de los datos. (Osm10) (Lago, 2005)

Classpath: Parecido a una variable de entorno de ruta en la que un cargador de clases debería buscar otras clases y recursos.

Secure Sockets Layer (SSL): Protocolo de Capa de Conexión Segura, es un protocolo criptográfico que proporciona comunicación segura por una red. (Schneier, 1996)

Librerías dinámicas: Ficheros independientes que pueden ser invocados desde cualquier ejecutable, de modo que su funcionalidad puede ser compartida por varios ejecutables. (1990-2010)

GDebi: Aplicación gráfica utilizada para instalar paquetes **‘.deb’**.

Synaptic: Es un programa informático, que es una interfaz gráfica, para el sistema de gestión de paquetes de Debian GNU/Linux. Generalmente se utiliza Synaptic para sistemas basados en paquetes **.deb** pero también puede ser usado en sistemas basados en paquetes RPM. Synaptic utiliza repositorios Debian. (Paarmann, 2004)

Clúster: Conjunto o agrupación de computadoras que están creados mediante el uso de componentes de Hardware comunes y que se comportan como si fuera una sola computadora.

Berkeley: Es un motor de bases de datos que provee a los desarrolladores una base de datos simple, rápida y segura, con cero administración, debido a que funciona como una biblioteca que se enlaza directamente en la aplicación eliminando la penalización en el rendimiento de los sistemas cliente-servidor y el procesamiento SQL, ideal para consultas estáticas sobre datos dinámicos.(2006-2010).

