

The logo for the Revista Cubana de Ciencias Informáticas (RCCI) features the letters 'RCCI' in a bold, blue, sans-serif font. The letters are contained within a white rectangular box with rounded corners. Below the box, there are faint, light blue horizontal lines.

Revista Cubana de Ciencias Informáticas

ISSN: 1994-1536

rcci@uci.cu

Universidad de las Ciencias Informáticas

Cuba

Anglada Martínez, Ramón Alexander; Garófalo Hernández, Alain Abel
Marco de trabajo para el desarrollo de herramientas orientadas a la gestión e integración
de servicios telemáticos de infraestructura en GNU/Linux
Revista Cubana de Ciencias Informáticas, vol. 7, núm. 2, junio, 2013, pp. 157-168
Universidad de las Ciencias Informáticas
Ciudad de la Habana, Cuba

Disponible en: <http://www.redalyc.org/articulo.oa?id=378334199006>

- Cómo citar el artículo
- Número completo
- Más información del artículo
- Página de la revista en redalyc.org

redalyc.org

Sistema de Información Científica

Red de Revistas Científicas de América Latina, el Caribe, España y Portugal

Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso abierto

Tipo de artículo: Artículo original
Temática: Tecnologías de la información y las telecomunicaciones
Recibido: 18/12/2012 | Aceptado: 28/05/2013

Marco de trabajo para el desarrollo de herramientas orientadas a la gestión e integración de servicios telemáticos de infraestructura en GNU/Linux

Framework for developing tools oriented to management and integration of infrastructure telematic services in GNU/Linux

Ramón Alexander Anglada Martínez¹, Alain Abel Garófalo Hernández²

¹ Centro Telemática. Universidad de las Ciencias Informáticas, Carretera a San Antonio de los Baños, km 2 ½, Torrens, Boyeros, La Habana, Cuba. CP.: 19370

² Departamento de Telecomunicaciones y Telemática. Instituto Superior Politécnico José Antonio Echeverría, Calle 114, No. 11901 e/ 119 y 127, Marianao, La Habana, Cuba. CP.: 19390

*Autor para la correspondencia: raanglada@uci.cu

Resumen

En la actualidad las Tecnologías de la Información y las Comunicaciones (TIC) juegan un rol fundamental en el desarrollo de cualquier organización. Una de las ventajas del uso de las TIC lo constituyen los servicios telemáticos, los cuales ofrecen muchas facilidades para los usuarios de las redes de datos actuales. Dentro de estos servicios telemáticos se encuentran un grupo de servicios telemáticos de infraestructura importantes para el buen funcionamiento de la red. Para alojar servicios telemáticos de infraestructura es muy utilizado el sistema operativo GNU/Linux por las ventajas que ofrece, sin embargo las herramientas empleadas para la administración de estos servicios de infraestructura no son del todo eficientes en la gestión de los parámetros de configuración de los servicios y la integración de los mismos. El presente artículo describe el diseño y arquitectura de un marco de trabajo que propicia el desarrollo de herramientas orientadas a la gestión e integración de servicios telemáticos de infraestructura,

reutilizando las bibliotecas de clases existentes especializadas en parte de la gestión de un determinado servicio de infraestructura, el artículo también muestra una herramienta implementada sobre este framework que facilita la gestión e integración de los servicios telemáticos de infraestructura en GNU/Linux disminuyendo las deficiencias de las herramientas actuales.

Palabras clave: Gestión, integración, marco de trabajo, servicios telemáticos, infraestructura.

Abstract

Today the Information Technology and Communications (ICT) play a fundamental role in the development of any organization. One of the advantages of using ICT constituent telematic services, which offer many facilities for users of existing data networks. Inside these telematics services are a set of infrastructure telematics services, important for the proper functioning of the network. To host telematics infrastructure services is widely used the GNU / Linux operating system for the benefits that its offers, but the tools used to manage these infrastructure services are not entirely efficient in managing the services configuration parameters and its integrations. This paper describes the design and architecture of a framework that enables the development of tools oriented to the management and integration of telematics infrastructure services, reusing existing class libraries specialized in management of a service infrastructure, the paper also shows a tool implemented with this framework that facilitates the management and integration of telematic infrastructure services in GNU / Linux reducing deficiencies of current tools.

Keywords: Framework, integration, infrastructure services, management, telematics.

Introducción

Un servicio telemático es una entidad geográficamente distribuida, que provee a un número de personas, un conjunto de facilidades para cubrir un rango de necesidades de información y comunicación, utilizando los recursos existentes de las redes de telecomunicaciones (Dionisis, 1998). Dentro de los servicios telemáticos existe un grupo de servicios de infraestructura, esenciales para el buen funcionamiento de la red, los cuales no son consumidos por el usuario final de manera directa, son servicios utilizados principalmente por otras aplicaciones o servicios las cuales dependen de estos para su funcionamiento (Peterson, 2011). Uno de los sistemas operativos utilizados con el objetivo de proveer servicios telemáticos de infraestructura es GNU/Linux, sistema operativo software libre y de código abierto el cual ofrece muchas ventajas. Para la administración de estos servicios en GNU/Linux el administrador de red utiliza regularmente herramientas informáticas las cuales ofrecen diversas funcionalidades, sin embargo, las mismas no son eficaces en la gestión de todos los parámetros de configuración del servidor, la integración de los servicios y la migración de los servicios alojados en plataformas propietarias hacia soluciones software libre de GNU/Linux.

Las herramientas informáticas de gestión de servicios de infraestructura son software cada vez más complejos, lo que impone la necesidad de una solución que permita reusar el diseño y el código fuente de los componentes existentes y aumentar la productividad de los desarrolladores en la generación de código y la flexibilidad para cambiar los requerimientos del software (Bosch, 1997; Fayad, 1997). Para desarrollar software y lograr reusabilidad de código fuente, productividad y flexibilidad se utilizan marcos de trabajo o frameworks. Un marco de trabajo es una aplicación reusable y semicompleta que puede ser especializada para producir aplicaciones personalizadas o específicas (Johnson, 1988). Con un framework se logra un alto grado de estandarización, homogenización y reutilización del código fuente, organización del esfuerzo de todos los miembros del proyecto y rapidez en el desarrollo del software.

El presente trabajo describe el diseño de un marco de trabajo que propicia el desarrollo de herramientas orientadas a la gestión de servicios telemáticos de infraestructura y muestra una herramienta desarrollada a partir de este marco de trabajo que disminuye las debilidades de las herramientas actuales utilizadas por los administradores para la administración de los servicios de infraestructura.

Materiales y métodos

En el diseño del marco de trabajo se utilizó un grupo de patrones de diseño (Larman, 1999) con el objetivo diseñar un marco sencillo, ligero, robusto y orientado a componentes que permitiera de forma coherente el desarrollo de herramientas para la gestión e integración de los servicios telemáticos de infraestructura y reutilizara la mayor cantidad de código fuente. Se utilizaron fundamentalmente los patrones Observador, Adaptador (Gamma, 1996), Fachada, Inversión del Control y varios Patrones Generales de Software para Asignar Responsabilidades (GRASP) (Larman, 1999).

Para la implementación del marco de trabajo se utilizó el lenguaje de programación Python. Python es un lenguaje de alto nivel orientado a objeto, de una sintaxis clara y limpia, posee una amplia biblioteca de clases que facilita el desarrollo de software (Lutz, 2008) y una comunidad de software libre que soporta el lenguaje y provee las actualizaciones correspondientes frecuentemente. Python posee varias bibliotecas de clases reutilizables que facilitan la gestión de varios servicios telemáticos de infraestructura, tales como python-ldap y python-dnspython que facilitan la gestión de los servicios DNS y Directorio respectivamente, python también posee bibliotecas de clases para la manipulación del contenido de los archivos de configuración de los servicios, en la implementación del marco de

trabajo se utilizó python-pyparsing la cual permitió representar los parámetros de configuración de los servicios de infraestructura de una manera orientada a objeto, adaptable y extensible. Se reutilizaron otras bibliotecas de clases para la conexión por SSH al servicio tales como python-paramiko y python-pexpect las cuales garantizan el acceso al servicio de una manera segura.

La herramienta desarrollada sobre este marco de trabajo es una aplicación de escritorio compuesta por 4 módulos y para el diseño de las interfaces gráficas se utilizó la herramienta Qt Designer la cual permite construir interfaces gráficas con la librería Qt. Qt es una librería multiplataforma mantenida por Qt-Project disponible en <http://qt-project.org/>. La librería Qt está desarrollada en C++ y tiene implementaciones ampliamente utilizadas en python, en el desarrollo de este software se utilizó la implementación PyQt lo cual está en correspondencia con la selección de python como lenguaje de programación para la implementación del marco de trabajo. Tanto la herramienta Qt Designer, la librería Qt, así como PyQt tienen licencias de software libre y código abierto lo cual permite el uso de estas tecnologías libre de costo siempre y cuando no se utilice para proyectos comerciales.

Para la codificación de todo el software en python se utilizó la herramienta EasyEclipse disponible en <http://www.easyeclipse.org>. Como su nombre lo indica está basado en Eclipse, Entorno de Desarrollo Integrado basado en plugins que permite el desarrollo de aplicaciones en varios lenguajes de programación. EasyEclipse es un Eclipse pre configurado para trabajar en python y otros lenguajes de programación, para el trabajo con python específicamente se utiliza el plugin pydev disponible en <http://pydev.org/> el cual brinda facilidades para desarrollar software con python. El marco de trabajo y la herramienta se nutren de configuraciones las cuales deben estar escritas en formato XML (del inglés, *eXtensible Markup Language*).

Resultados y discusión

GNU/Linux es uno de los sistemas operativos ampliamente utilizados para alojar servicios telemáticos de infraestructura. Para administrar correctamente estos servicios y obtener un resultado deseado es fundamental tener en cuenta todos los parámetros de configuración del servidor, así como las vías de integración con otros servicios. Para la administración de estos servicios se han desarrollado herramientas tales como Webmin, Phpldapmin, Zentyal, FreeIpa, entre otras. Estas herramientas no ofrecen la gestión de todos los parámetros de configuración de los servicios telemáticos de infraestructura, concentrándose, la mayoría de estas herramientas, en los parámetros fundamentales para la puesta en marcha del servicio, para los parámetros que no son gestionados por la herramienta y

son de interés para el administrador de red, el mismo se ve obligado a editar directamente los archivos de configuración del servicio lo cual aumenta la posibilidad de cometer errores afectando la disponibilidad del servicio y los usuarios finales. Las configuraciones de los servidores en GNU/Linux están en ficheros de texto y la sintaxis es heterogénea y cada servicio puede tener varios archivos de configuración en el servidor y dentro de cada uno definir disímiles parámetros con un alto grado de relación.

Otra de las dificultades que presentan las herramientas son las pocas opciones que ofrecen para la realizar las integraciones posibles entre los servicios, tales como las integraciones de los servicios a través de la gestión de los esquemas del servicio Directorio (Zeilenga, 2006), la integración del servicio DNS a una base de datos externa con el objetivo aumentar su rendimiento y reutilizar el mecanismo de resolución de nombres en otros escenarios (García, 2008), entre otras integraciones. En la mayoría de los casos el administrador de red tiene que acudir a una tercera herramienta que si brinde la posibilidad de integración que necesita implementar u otras variantes como la ejecución de comandos en consola y la edición de varios archivos de configuración afectando nuevamente la disponibilidad del servicio y haciendo más compleja su administración. También constituye una insuficiencia de estas herramientas de gestión de servicios telemáticos de infraestructura en GNU/Linux las pocas opciones para importar datos y configuraciones de los mismos servicios alojados en otros sistemas operativos lo que dificulta que se puedan trasladar los mismos a GNU/Linux, dejando de apoyar al proceso de migración a software libre que llevan a cabo muchas organizaciones y países de manera general.

En la siguiente tabla se muestra un resumen de las herramientas mencionadas y sus principales debilidades comparándolas con el marco de trabajo que se muestra en este artículo.

Tabla 1. Comparación entre el marco de trabajo y varias herramientas de gestión de servicios telemáticos de infraestructura.

Parámetros	Marco de trabajo	Webmin	PhpLadmin	Zentyal
Gestión de todos los parámetros de configuración.	La mayoría de los servicios LDAP, DNS, DHCP. Ofrece la posibilidad realizar extensiones.	La mayoría. No existen posibilidades de realizar extensiones.	Los parámetros básicos del servicio LDAP y no la mayoría.	Los parámetros básicos de la mayoría de los servicios. Ofrece posibilidades de extensión.
Integración de los servicios	DNS-DHCP, LDAP-DNS, DNS-BD, LDAP con otros servicios a través de la gestión de esquemas. Posibilidades de extensión.	DNS-DHCP.	Ninguna	DNS-DHCP
De código abierto	Si		Si	Solo una parte del proyecto, muchas de las mejoras que

				se realizan se ofrecen a través de una versión comercial.
Licencia propietaria	No	Sí. En algunos de sus módulos.	no	Sí.

La tabla anterior se puede apreciar que la mayoría de las herramientas que se comparan con el marco de trabajo no gestionan la mayoría de los parámetros de configuración de los servicios de infraestructura pero tampoco ofrecen posibilidades de extensión a excepción de Zentyal, pero este es un proyecto propietaria y comercial y para disfrutar de sus potencialidades se debe adquirir de una licencia. Las herramientas mencionadas tampoco ofrecen la posibilidad de ejecutar las integraciones posibles que se pueden configurar entre los servicios de infraestructura, una de las integraciones importantes para la integración de los servicios es la integración del servicio LDAP a través de sus esquemas, esta integración no se ofrece en las herramientas, en el caso de Webmin se ofrece directamente el texto de los archivos de configuración lo cual puede provocar errores en la configuración ya que se editan directamente estos archivos, sin embargo el marco de trabajo que se describe en este artículo ofrece la posibilidad de gestionar los esquemas de una manera orientado a objeto, de una manera sencilla, facilitando el desarrollo de herramientas para la gestión e integración de los servicios de infraestructura evitando la introducción de errores en la configuración de los servicios.

Cabe destacar que las herramientas mencionadas no reutilizan las bibliotecas de clases actuales de código abierto y software libre públicas y disponibles en Internet, empezando de cero en la mayoría de los casos con el desarrollo de las herramientas, implementando sus propias bibliotecas de clases, en varios casos poco reutilizables y acopladas a las herramientas desarrolladas. Con el objetivo de disminuir las deficiencias de las herramientas actuales se diseña e implementa este marco de trabajo que reutiliza e integra en un modelo o esquema las bibliotecas de clases existentes para la gestión e integración de los servicios telemáticos de infraestructura en GNU/Linux reutilizando código fuente y esfuerzo.

Diseño marco de trabajo

Un marco de trabajo o framework es un diseño abstracto orientado a objetos para un determinado tipo de aplicación, es un patrón arquitectónico que proporciona una plantilla extensible para un tipo específico de aplicaciones. Según (Larman, 1999) un framework o “esquema” es un subsistema expandible de un conjunto de servicios, es un conjunto cohesivo de interfaces y clases que colaboran para proporcionar los servicios de la parte central e invariable de un

subsistema lógico. El marco de trabajo diseñado posee una arquitectura orientada a componentes y está compuesto por cinco paquetes principales: **Núcleo**, **Servicios**, **Persistencia**, **Componentes** y **Aplicación de Usuario**. Cada paquete está compuesto por componentes y estos a su vez definen una interfaz de comportamiento con el objetivo de extender sus funcionalidades y adaptarse a posibles escenarios.

Núcleo es el paquete principal del modelo y en él se agrupan todos los componentes base del marco: Eventos, Conexión, Inversión del Control, Multiprocesamiento y Excepciones. Destacar en el núcleo el componente Conexión el cual garantiza el acceso al servicio telemático de infraestructura principalmente a través de SSH. El componente permite además la obtención de los archivos de configuración del servicio y ofrece un grupo de eventos para conocer el estado de la conexión con el servicio telemático de infraestructura. El resto de los componentes del núcleo son componentes horizontales a cualquier aplicación: Eventos es una implementación del patrón Observador, Multiprocesamiento define una implementación similar al patrón Piscina de Hilos (Garg, 2002) para garantizar un buen rendimiento de la aplicación a desarrollar, Inversión del control provee una implementación del término del mismo nombre también conocido como Inversión del Control e Inyección de dependencia (Larman, 1999), en el componente Excepciones se agrupan las clases que definen los tipos de error que pueden arrojar los métodos de todas las clases del marco de trabajo. A continuación a una vista de la arquitectura del marco de trabajo:

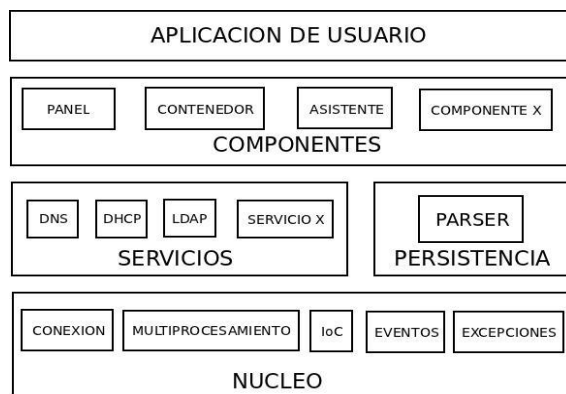


Figura 1. Diseño marco de trabajo.

En el paquete Persistencia se concentran los componentes y clases especializados en la manipulación del contenido de los archivos de configuración de los servicios de infraestructura. Se define un componente Parser el cual se especializa en la manipulación del contenido de los archivos de configuración del servicio. El uso de expresiones regulares para la lectura y escritura de los archivos de configuración de los servicios telemáticos de infraestructura hace muy compleja la obtención de la información contenida en los archivos de configuración cuya definición puede

variar de una versión a otra del software que implementa el servicio o con actualizaciones del mismo. Con el propósito de obtener la información contenida en los archivos de configuración del servicio de una forma orientada a objeto, adaptable, extensible y que permita representar la información del servicio de una manera cómoda para el desarrollador se define el componente Parser.

El componente Parser se nutre de plantillas que representan el patrón de información que puede estar contenido en los archivos de configuración del servicio de infraestructura y almacena directamente la información a las clases de dominio que representan al servicio. Por cada archivo de configuración del servicio se define esta plantilla en la cual se definen Literales, Palabras y Gramáticas. Los literales representan a los símbolos que denotan delimitación, puntuación, agrupación, entre otros, las palabras son todas las combinaciones de letras y números y una gramática es una agrupación de Literales, Palabras y Gramáticas, cuya función es definir la relación que existe entre los Literales, Palabras y Gramáticas de la plantilla, y como estos forman las clases que representan los objetos del servicio de infraestructura. De manera general el componente Parser garantiza una manipulación orientada a objeto de los parámetros de configuración del servicio de infraestructura, este componente es extensible y adaptable permitiendo al desarrollador abstraerse de los detalles de la sintaxis del servicio.

Otro de los paquetes fundamentales del marco de trabajo es el paquete Servicios, en este se encuentran los componentes y clases especializadas en la gestión de los servicios de infraestructura, como realizar sus posibles integraciones con otros servicios y se reutilizan los componentes Conexión y Parser principalmente y las bibliotecas de clases de terceros que facilitan parte de la gestión de los servicios telemáticos de infraestructura. Este paquete implementa el patrón Fachada y homogeniza a través de un grupo de interfaces la gestión de los servicios DHCP, DNS y Directorio de GNU/Linux, para cada servicio a gestionar se define un componente que se especializa en la gestión del mismo. En el paquete Componentes se definen los componentes gráficos para la aplicación de usuario utilizados con el objetivo de representar la información de los servicios y ejecutar acciones sobre los mismos. El paquete Aplicación de Usuario concentra la mayor parte de la lógica de la aplicación de usuario, donde interviene principalmente el desarrollador que utilice el marco de trabajo.

La definición en componentes del marco de trabajo permite especializar las áreas en la gestión de configuración e integración de los servicios concentrando al desarrollador en temas específicos de su aplicación. Los niveles permiten organizar los paquetes de componentes aumentando el grado de reutilización del marco de trabajo.

Herramienta desarrollada

A partir del marco de trabajo diseñado se desarrolló una herramienta compuesta por 4 módulos fundamentales, DNS; DHCP, Directorio e Integrador. Los tres primeros facilitan la gestión de los parámetros de configuración de los servicios telemáticos de infraestructura Bind9, DHCP de ISC version 4 y OpenLDAP en su version 2.3, y permiten gestionar simultáneamente más de un servidor del servicio correspondiente e importar los datos del mismo servicio ubicado en Windows Server 2003 contribuyendo a la migración paulatina a software libre. Los módulos también brindan opciones de parada, inicio y reinicio del servidor que se esté gestionando en caso de que los cambios realizados en la configuración de estos servicios lo necesiten. El módulo Integrador permite instalar los servicios mencionados y realizar varias integraciones posibles entre los servicios de infraestructura. Cada módulo posee una arquitectura basada en el patrón arquitectónico N capas (Larman, 1999) y se soporta sobre el marco de trabajo diseñado. En la siguiente Figura se muestra el diseño de la arquitectura de los módulos:

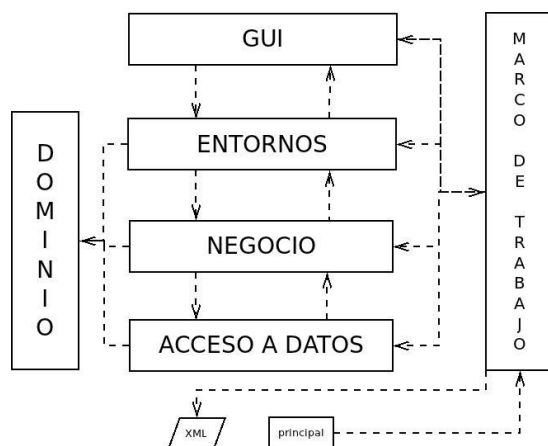


Figura 2. Arquitectura de los módulos de la herramienta.

En la capa GUI se agrupan todas las clases que representan las ventanas gráficas de usuario, Entornos enlaza las capas GUI y Negocio haciendo cada una de estas más reutilizables e implementa el patrón Acción (Gamma, 1996) separando en objetos cada uno de los flujos de acciones del usuario. Negocio concentra la lógica de gestión de los servicios apoyado en la capa Acceso a Datos la cual facilita el acceso y manipulación de las configuraciones de los servicios. En la capa Dominio se agrupan las clases que representan al servicio de infraestructura y es utilizada por la

mayoría de las capas de la arquitectura. Cada capa de esta arquitectura, excepto Dominio, reutiliza los componentes del marco de trabajo. Esta arquitectura define un punto de ejecución a través del cual se ejecuta la aplicación.

Una arquitectura multicapas permite aislar la lógica de las aplicaciones en componentes independientes que luego pueden reutilizarse, se alcanza una mejor distribución del sistema y se puede lograr una especialización por parte de los desarrolladores al estar asignado su trabajo a una sola capa. A continuación una imagen del módulo DNS:

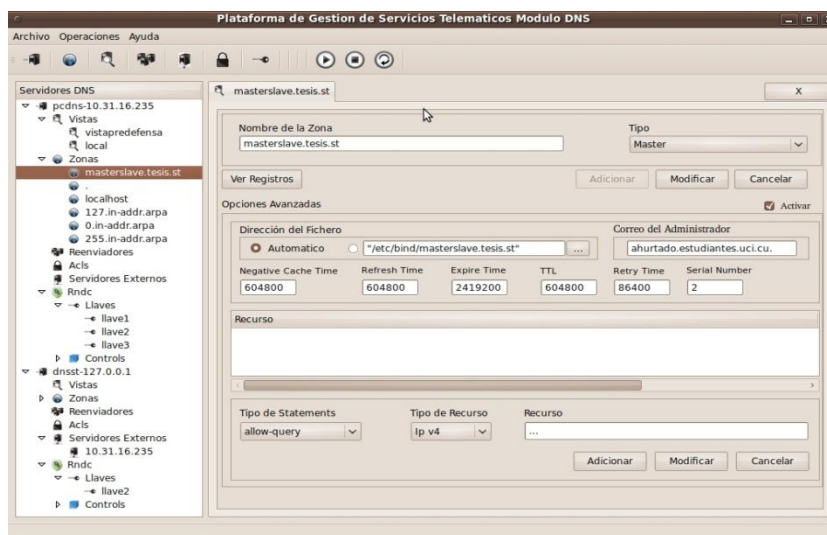


Figura 3. Módulo DNS.

El módulo DNS de la herramienta permite la gestión de los parámetros de configuración del Bind9, la integración del Bind con el DHCP de ISC o con una base de datos específicamente con Postgres, entre otras funcionalidades. Destacar en el módulo LDAP que el mismo permite la gestión de los esquemas y plantillas para el servidor OpenLdap facilitando la integración a este servicio telemático de infraestructura.

Conclusiones

Las herramientas actuales para la gestión de los servicios telemáticos de infraestructura en GNU/Linux poseen un grupo de debilidades en cuanto a gestión de los parámetros de configuración de los servicios y la integración de los mismos y no reutilizan muchas de las bibliotecas de clases de código abierto que pueden ser utilizados en el desarrollo de estas herramientas. El marco de trabajo diseñado facilita y propicia el desarrollo de herramientas orientadas a la gestión e integración de servicios telemáticos de infraestructura, agrupa en un modelo las bibliotecas de clases software libre y de código abierto especializadas en partes de la gestión de varios servicios de infraestructura

reutilizando código fuente y esfuerzo. La herramienta desarrollada, a partir de este marco de trabajo, disminuye las deficiencias de las herramientas actuales, ofrece una herramienta para la gestión e integración de los servicios Bind9, DHCP de ISC y OpenLdap y contribuye a la migración a software libre que llevan a cabo varias organizaciones y países.

Referencias

- ABOWD, GREGORY. Recommended Best Industrial Practice for Software Architecture Evaluation: Software Engineering Institute Carnegie Mellon University Pittsburgh. 15213. 1997.
- BIND DLZ, Dynamically Loadable Zones. Why is DLZ needed? [Consultado en: diciembre de 2012]. Disponible en: [<http://bind-dlz.sourceforge.net/>].
- BOSCH, JAN. Object-Oriented Frameworks - Problems & Experiences: s.n., 1997.
- DIONISIS, ADAMOPOULOS. Object-Oriented Development of Telematic Services: s.n., IEEE Symposium on Computers and Communications. 1998.
- FAYAD, MOHAMED E. Object-Oriented Application Frameworks: s.n., Communications of the ACM. 1997.
- GAMMA, ERICH. *Design Patterns. Elements of Reusable Object-Oriented Software*. s.l.: Addison Wesley, 1996.
- GARCÍA, MARCOS A. PÉREZ. Impacto del ENUM en las redes y los servicios: s.n., Ingeniería Electrónica, Automática y Comunicaciones. 2008.
- GARG, RAJAT. *Techniques for Optimizing Applications - High Performance Computing*. s.l.: Prentice-Hall, 2002.
- JOHNSON, RALPH. Designing Reusable Classes: s.n., Journal of Object-Oriented Programming. 1988.
- LARMAN, CRAIG. *UML y Patrones. Introducción al análisis y diseño orientado a objetos*. s.l.: Prentice Hall Hispanoamericana, 1999, p. 272-279.
- LUTZ, MARK. *Learning Python*. 3rd. s.l.: O'Reilly Media, 2008.
- PETERSON, LARRY L. *Computer Networks a Systems Approach*. 5th. 2011. p. 744-758.

- WEBSITE, Python Programming Language – Official. [Consultado en: marzo de 2013]. Disponible en [<http://python.org/>].
- ZEILENGA, Network Working Group K. Request for Comments: 4512. Lightweight Directory Access Protocol (LDAP): Directory Information Models. 2006.