Sistema de Notificaciones para la Plataforma de Desarrollo de Integración Continua de la distribución cubana de GNU/Linux, Nova.

Notification System for Development Platforms of Continuous Integration inside the Cuban Distribution Pattern of GNU/Linux Nova.

Ing Jesús Rabelo Pérez [1], Ing Marielis González Muño[2], Ing Héctor Pérez Baranda[3], Ing Mairim Delgado Muñiz[4]

[1]Universidad de las Ciencias Informáticas – (UCI)
Universidad de las Ciencias Informáticas. Carretera a San Antonio de los Baños, Km. 2 ½. Torrens, municipio de La Lisa. La Habana, Cuba.

{jrabelo, mmuno, hbaranda, mdelgado }@uci.cu

Abstract. *With the development of the current research it was possible to implement a notification system, within the Platform for Development and Continuous Integration relevant to the Cuban distribution of GNU / Linux, to notify users in an efficient and scalable way about the events related to the different tools involved in the construction process. A research of protocols and of middleware for message passing was conducted, with a deeper and more updated perspective of the present situation. Aspects of the proposed solution are also exposed, and the verification of their scalability and efficiency has taken place. Its main result is the obtaining of a functional system, made up by a generator of notifications and a client application that is embedded in the notification bar of Nova distributions, informing users about the platform by means of a publication/subscription mechanism. Furthermore, an architecture that easily supports the integration of different communication channels has been conceived, thus ensuring scalability and efficiency.*

Keywords: *development platforms and continuous integration, notification systems, Nova, messages middleware.*

Resumen. *Con el desarrollo de la presente investigación se logró implementar un sistema de notificaciones, en la Plataforma de Desarrollo e Integración Continua de la distribución cubana de GNU/Linux, para notificar a sus usuarios de manera eficaz y escalable de los eventos asociados a las herramientas que intervienen en el proceso de construcción. Se realizó un estudio sobre los protocolos y middleware para el paso de mensajes, con mayores perspectivas en la actualidad. Se expone los aspectos de la solución propuesta, y se realiza la verificación de su escalabilidad y eficacia. Su principal resultado lo constituye la obtención de un sistema funcional, compuesto por un generador de notificaciones y una aplicación cliente que se incrusta en la barra de notificación de las distribuciones Nova, manteniendo informado a los usuarios de la plataforma mediante un mecanismo de publicación/suscripción. Además de una arquitectura que soporta la*

incorporación de diferentes vías de comunicación fácilmente, garantizando escalabilidad y eficacia.

Palabras claves: *plataformas de desarrollo e integración continua, sistema de notificaciones, Nova, mensajes, middleware.*

1. Introducción

El desarrollo acelerado de las distribuciones de GNU/Linux se ha mantenido como uno de los temas más activos y en constante evolución. Hoy a más de 20 años del surgimiento de la primera de ellas, prácticamente no hay lugar donde no se encuentren presente. Ordenadores personales, teléfonos móviles, tabletas y electrodomésticos, son algunas de las formas más comunes en las que se manifiestan [Llamaret, 2013].

Actualmente Debian, Ubuntu, Fedora y openSUSE se encuentran entre las distribuciones más destacadas [Llamaret, 2013]. La calidad de su paquetería y la agilidad para obtener versiones nuevas o mejoradas de cada uno de sus productos, ha estado muy ligado a las infraestructuras tecnológicas de sus empresas. Gracias a los servidores que las componen y el software que corre en ellos, los desarrolladores solo tienen que preocuparse de tener el código fuente listo y enviarlo [Herrera, 2013]. Wana-Build, Soyuz, Koji y Open Build Service, son ejemplos respectivos de herramientas que conforman dichas infraestructuras [Hodek, 2013]. Sus diferentes componentes agilizan la construcción de paquetes y la gestión de repositorios. Automatizan todo el proceso de elaboración de la imagen y permiten detectar fallos de seguridad. Facilitan la liberación de aplicaciones, actualizaciones, extensiones, e incluso distribuciones completas. Todas en un amplio rango de arquitecturas de sistemas operativos y hardware al servicio de la comunidad [Herrera, 2013].

Un aspecto importante para dichas infraestructuras y la concepción de sus productos, ha sido la evolución de la informática y las comunicaciones. Sin el surgimiento del Internet, el correo electrónico, la telefonía móvil y los clientes de mensajería instantáneas, el éxito no sería el mismo. Es por eso que muchos componentes incluyen interfaces web o se apoyan en sus wiki, las cuales poseen sistemas de notificaciones por distintas vías, garantizando que los usuarios finales participen y estén al tanto de todo lo que acontece [Edwards, 2011]. Dichos sistemas se apoyan a su vez en diferentes productos que han buscado dominar el campo de la mensajería eficiente, contando con gran relevancia los middleware orientados a mensajes (MOM).

Cuba desde febrero del 2009 cuenta con Nova, una distribución cubana de GNU/Linux que busca minimizar la dependencia tecnológica [Pierra, 2013], desarrollada y mantenida por el Departamento de Sistema Operativo (DSO) del Centro de Soluciones Libres (CESOL) de la Universidad de las Ciencias Informáticas (UCI) [Oramas, 2013]. Desde su surgimiento ha pasado por diferentes etapas que han evidenciado la ineficacia de su infraestructura [Herrera, 2014]. Los problemas que han presentado las herramientas y métodos que componen su desarrollo base, han contribuido a que Nova sea reconocida ante el mundo como una distribución descontinuada [DistroWatch.com, 2014]. Considerando como las distribuciones exitosas integran sus herramientas, y como la interacción entre usuarios y desarrolladores repercute en su calidad. Los integrantes del DSO decidieron concebir su propia plataforma, en busca de una infraestructura más estable y eficiente.

La Plataforma de Desarrollo e Integración Continua de Nova, cuenta con un Sistema de

Gestión de Repositorios[1], un Sistema Distribuido de Compilación[2], un Sistema para la Generación Automática de Imágenes de CD/DVD[3] y una herramienta de escaneo automático de código fuente[4] [Herrera, 2013]. Sin embargo en un ambiente donde las operaciones pueden tardar horas en completarse, los usuario se ven obligados ha permanecer en línea con ella o estar pendiente con cierta frecuencia, para conocer si sus acciones han sido completadas o se han interrumpido por la ocurrencia de errores, lo cual genera molestos contratiempos que desmotivan y afectan el desarrollo de Nova. Dado que no se gestionan los eventos emitidos por dichas operaciones, se dificulta además el conocimiento de las actualizaciones, fallos detectados y mejoras realizadas a la distribución cubana o su paquetería por los demás miembros de las comunidad, provocando duplicación de esfuerzo y pérdida de tiempo en ajustes o desarrollos ya efectuados por otros. Teniendo en cuenta lo que significa para el éxito de las distribuciones actuales tener una comunidad que la respalde y contribuya constantemente con ideas nuevas, se hace urgente la búsqueda de una solución que permita mantenerla informada, con una efectividad dada esencialmente por los siguientes aspectos:

- Resistencia a fallos: El tratamiento de los mensajes al producirse un fallo técnico tanto en el servidor como en el cliente trae consigo menos mensajes/notificaciones perdidos, y por tanto los usuarios presentan una menor desinformación [Offut, 2002].
- Tiempo de respuesta: Cuanto menor sea el tiempo de respuesta al crear y enviar mensajes/notificaciones una vez que se produce un evento o acción, los usuarios se notifican más próximo a la culminación real del evento o acción [Offut, 2002].
- Escalabilidad: La forma en que se elimine las brechas de comunicación entre diferentes arquitecturas, plataformas y sistemas operativos, permite atender mayor o menor número de peticiones sin que se pierda o cese por completo la capacidad de respuesta [Pressman, 2009].

A raíz de estos problemas se plantea entonces, como objetivo de la presente investigación: Desarrollar un sistema de notificaciones, de manera efectiva, para la suscripción de los usuarios a temas de su interés, en la Plataforma de Desarrollo e Integración Continua de Nova.

2. Contenidos

Materiales y método

La solución que se propone se implementó en el lenguaje de programación Pyhton en su versión 2.7 y se basa en el modelo cliente-servidor y el patrón Broker. Las tecnologías utilizadas del lado del servidor fueron el marco de trabajo para aplicaciones web Django en su versión 1.3.1 y el gestor de bases de datos PostgreSQL. Como elemento fundamental en la creación de componentes visuales e interfaces de usuario del lado del

1 Sistema que gestiona los repositorios de paquetes tanto de código fuente como binario de las distintas variantes de la distribución cubana de GNU/Linux.
2 Se encarga de la generación de paquetes binarios a partir de código fuente mediante una granja de computadoras.
3 Automatiza el proceso de generación de imágenes en formato ISO 9660 para ser quemadas en CD/DVD.
4 Herramienta elaborada por los integrantes del proyecto Nova para la detección de fallas de seguridad en código fuente.

cliente, se utilizó la biblioteca gráfica PyQt en su versión 4.8.1. El entorno de desarrollo integrado fue la herramienta eclipse en su variante Indigo con el plugin PyDev 2.8.2. Para el intercambio de datos entre cliente y servidor se utilizó la notación de objetos de JavaScript (JSON, por sus siglas en Inglés). Su columna vertebral la constituye el empleo de un protocolo de colas de mensaje mediante un middleware orientado a mensajería para la gestión y envío de las notificaciones, que brinda cualidades imprescindibles para el logro de la efectividad deseada. A continuación se describen diferentes sistemas de notificación, protocolos y middlewares estudiados para el desarrollo de la solución.

Sistemas de Notificaciones e integración continua

La red social Twitter utiliza un sistema basado en la biblioteca de socket[5] ZeroMQ, para el envío eficiente de mensajes cortos a correo electrónico y teléfonos móviles. La plataforma colaborativa de software libre Launchpad, presenta un sistema de notificación configurable, con listas de correo electrónicos, a las cuales los usuarios se suscriben de acuerdo a su preferencia [Launchpad, 2014]. En el proyecto Fedora se utilizan además los canales IRC como comunicación inmediata [fedoraproject.org, 2014].

En el ámbito de la integración continua, Buildbot implementa un sistema de colas de espera, que ejecuta tareas en repositorios de código fuente reportando sus resultados mediante correo electrónico, canales IRC y clientes de estados con alta escalabilidad [buildbot.net, 2014]. Jenkins al igual que buildbot está diseñado para facilitar la integración de cambios en la construcción de proyectos [jenkins-ci.org, 2014], monitorizando los resultados por correo electrónico y la Sindicación Realmente Simple (RSS, por sus siglas en inglés). Considerando la amplia utilización por los usuarios de las diferentes vías de comunicación presentes en redes sociales y plataformas colaborativas, y su impacto en los sistemas de integración continua. Se plantea la concepción de una solución con una arquitectura que permita mantener al tanto de los proyectos a los usuarios de la plataforma, a la ves que comparten todo tipo de información asociada al proceso de desarrollo de Nova.

AMQP

El protocolo avanzado de espera de mensajes (del inglés AMQP) es un estándar abierto para la comunicación mediante un middleware orientado a mensajería. Su objetivo principal es alcanzar la interoperabilidad entre diferentes servicios. Proporciona un espacio de colas compartido, accesibles para todas las aplicaciones interesadas en el intercambio de mensajes. Permite tanto el enrutamiento punto a punto como el publicador/suscriptor [Videla, 2012]. Las características esenciales que lo diferencia de otros estándares de middleware son:

Seguridad: Proporciona una infraestructura para una red con transacciones seguras y de confianzas. Soporta la perdurabilidad de los mensajes independientemente de la conexión de los receptores. La entrega de mensajes es resistente a fallos técnicos.

Confiable: Capaz de eliminar las brechas y retrasos de diferentes plataformas, sistemas y componentes críticos de aplicaciones, tanto dentro como fuera de las empresas. Garantiza la entrega de los mensajes, implementando una semántica que abarca: al menos una vez, a lo sumo una vez y solo una vez, conocido como entrega fiable.

5 Constituyen un concepto abstracto por el cual dos programas pueden intercambiar cualquier flujo de datos.

Unificado: Provee un conjunto de patrones básicos de mensajería mediante un único protocolo manejable: mensaje dirigido asíncrono, publicador/suscriptor, solicitud/respuesta, almacenar y reenviar.

Los Middleware Orientados a Mensajes

Los MOM son aplicaciones que utilizan los mensajes como método de integración y provee mecanismos para crearlos, almacenarlos y trasmitirlos. Se ubican entre la capa más alta (usuarios y aplicaciones) y la más baja (mecanismos de comunicación básicos) [Pressman, 2009]. Utilizan diversos estándares de protocolos abiertos para la comunicación, entre los que destaca el protocolo orientado a texto simple o en flujo (del inglés STOMP), el servicio de mensajería de java (del inglés, JMS) y el protocolo avanzado de espera de Mensajes (del inglés, AMPQ).

RabbitMQ

RabbitMQ es un MOM de código abierto escrito en Erlang. Es especialmente adecuado para aplicaciones distribuidas y su instalación se ejecuta de forma rápida. Ofrece alto rendimiento, fiabilidad, escalabilidad y disponibilidad. Incluye la persistencia de los mensajes, así como su acuse de recibo. Permite la agrupación de varios intermediarios RabbitMQ, en una red local, formando un solo agente lógico. Tiene disponible diferentes bibliotecas para la programación en Java y ofrece un conjunto amplio de APIs de lenguajes de programación. Brinda una interfaz web atractiva que permite una fácil administración con un conjunto de estadísticas referente al rendimiento y el consumo de recursos.

RabbitMQ destaca sobre otras tecnologías punteras en la mensajería eficiente como ActiveMQ, ZeroMQ y Dbus dado que:

- Las bases de datos relacionales de ActiveMQ para la perdurabilidad de los mensajes, atenta contra el rendimiento de la plataforma [ActiveMQ.org, 2014].
- ZeroMQ no asegura la permanencia de los mensajes en la cola, dado el caso de que los usuarios no puedan recibirlos. Es compleja su implementación y futura extensión [ZeroMq.org, 2014].
- La comunicación por Dbus no es resistente a fallos, no admite el enrutamiento de los mensajes por patrones, no utiliza el paradigma publicador/suscriptor, requisito imprescindible de la solución [freedesktop.org, 2014].

RabbitMQ es ideal para la solución propuesta y su utilización se fundamenta en:

- La orientación de su núcleo al estándar AMQP, y la interoperabilidad, confiabilidad e integración que le confiere.
- La capacidad que posee para la gestión de las colas y de los mensajes, facilitando la concurrencia de usuarios.
- La garantía que ofrece, en caso de fallos en el sistema, brindando un lugar seguro, para la perdurabilidad de los mensajes, hasta que se restablezcan los servicios, respetando el orden de prioridad.
- La absoluta confianza de que los mensajes han sido recibidos por sus receptores, mediante la funcionalidad de entrega garantizada.

Para dar cumplimiento al objetivo propuesto, se han combinado diferentes métodos

teóricos y empíricos de la investigación científica, en la búsqueda y procesamiento de la información:

Métodos teóricos: El método histórico-lógico para el estudio de los trabajos anteriores que constituyen referentes teórico-prácticos en el tratamiento del problema planteado, así como el analítico-sintético para recopilar la información de las diferentes bibliografías relacionadas con los sistemas de notificaciones, y avanzar hacia la solución.

Métodos empíricos: El método experimental para elaborar los casos de pruebas y proceder a la comprobación de los indicadores que definen la efectividad de la solución.

3. Resultados y discusión

El sistema de notificaciones para la Plataforma de Desarrollo e Integración Continua de Nova posee una arquitectura genérica que permite al administrador elaborar notificaciones referentes a cada uno de los componentes de la plataforma a medida que se vayan incluyendo. Es capaz de adaptarse fácilmente a la incorporación de nuevos módulos y sus respectivas notificaciones, permitiendo a su vez la resistencia de la información a fallos técnicos tanto en el servidor como en el cliente, mediante las opciones de perdurabilidad brindadas por Rabbitmq y AMQP. El procedimiento asíncrono que emplea, posibilita que la plataforma pueda concentrar su poder de cómputo en las tareas más críticas del proceso de construcción de Linux, sin necesidad de esperar respuestas del envío de los mensajes, pero con un grado de confianza elevado referente a su recepción. En cuanto al cliente, se obtuvo una aplicación de escritorio, que se incrustan en la barra de notificación de los sistemas operativos brindando información a los usuarios en todo momento. La aplicación cliente utiliza el paradigma de comunicación publicador/suscriptor, permitiendo a los usuarios elegir sobre qué tema desean ser notificados. Como medidas de seguridad, solo los usuarios registrados en la plataforma pueden acceder a este servicio, y la información viaja por un canal cifrado.

La utilización del sistema de notificaciones agiliza el proceso de construcción de la distribución GNU/Linux Nova ya que elimina los contratiempos ocasionados por la puesta en marcha de operaciones de larga duración en la plataforma, cuya completitud o fallo solo era informado vía navegador web. Obligando a los usuarios a permanecer pendientes con cierta frecuencia y desviando la atención de sus otras tareas. La arquitectura de la solución propuesta resuelve por completo esta situación, el cliente de mensajería constituye solo un ejemplo cercano al equipo de desarrollo de nova y su comunidad. Disímiles son las vías de comunicación que pueden ser incluidas fácilmente sin necesidad de modificar en absoluto la solución, como el empleo de la telefonía móvil, el correo electrónico o RSS, posibilitando a sus usuarios compartir todo tipo de información referente a Nova. Esto trae consigo un impacto económico, político y social de suma importancia para el país cuando tenemos en cuenta el proceso de migración que se llevan a cabo en empresas civiles y militares, o los pedido de distribuciones a la medida donde el tiempo muchas veces es un factor crítico.

Así mismo garantiza que los usuarios de la Plataforma de Nova, ya sean desarrolladores o miembros de la comunidad puedan estar al tanto de todo lo que acontece y colaborar en aquellos temas que más les interesen lo cual repercute directamente en la evolución y calidad de Nova, puesto que la retroalimentación con la comunidad de usuarios ha sido por décadas el motor impulsor de las Distribuciones GNU/LInux [DistroWatch.com, 2014]. Por otra parte ayuda a formar en la comunidad cubana un sentido de pertenencia hacia la distribución cubana, que traza un invaluable camino hacia la soberanía

tecnológica. A continuación se muestran algunas ilustraciones del sistema:

Figura 1: Interfaz de la plataforma de desarrollo

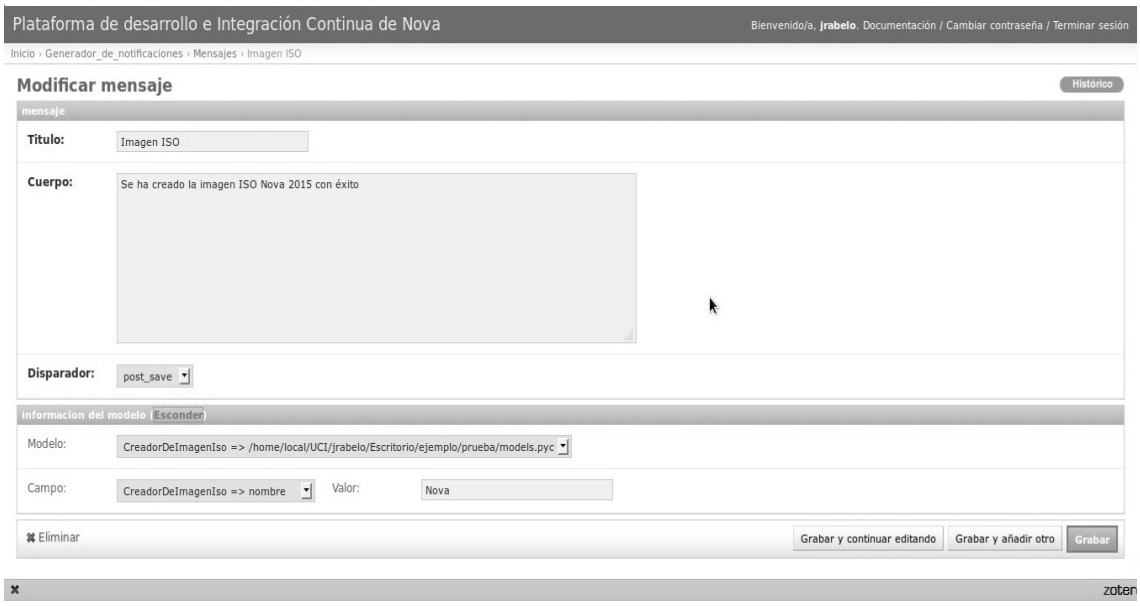

Figura 2: Cliente de notificaciones

4. Conclusiones

Al culminar la presente investigación se logró definir una arquitectura robusta, genérica y extensible, que permite al administrador de la Plataforma de Desarrollo e Integración Continua del Sistema Operativos Nova incorporar notificaciones personalizadas, así como nuevas vías de comunicación con alta escalabilidad. La utilización del protocolo asíncrono de colas de mensajes AMQP y del middleware orientado a mensajes

RabbitMQ permitió notificar a los usuarios de la Plataforma de Desarrollo e Integración Continua de Nova de manera efectiva.

5. Referencias Bibliográficas

[1] Apache ActiveMQ TM . [En línea]. [Accedido: 09-feb-2014]. Disponible en:

http://activemq.apache.org/.

[2] Django documentation — Django 1.3.1 documentation. [En línea]. [Accedido: 12-feb- 2014].Disponible en: file:///usr/share/doc/python-django-doc/html/index.html#.

[3] Edwards, J., 2011. Soyuz/TechnicalDetails - Launchpad Development. [En línea]. [Consultado el 17 de diciembre de 2012].Disponible en:<https://dev.launchpad.net/Soyuz/TechnicalDetails>

[4] Edwards, J., 2011. Soyuz/TechnicalDetails - Launchpad Development.[En línea]. [Consultado el 17 de diciembre de 2012].Disponible en:<https://dev.launchpad.net>

[5] Hodek, R., 2011. Debian -- Autobuilder network. [En línea],[Consultado el: 17 de Diciembre 2013]. Disponible en : <http://www.debian.org/devel/buildd/index.en.html>.

[6] Jacobson,I. ,Jacobson S. ,"Designing an integrated SEPSE", Object Magazine, September 1995, p.98-110.

[7] Llamaret, Heredia Maikel. SWL-X: Rompiendo el cascarón. [En línea]. Edición 0 (2013). [Consultado el: 12 de Noviembre del 2013]. Disponible en: <http://store.uci.cu/humanos/doc/revistas/CC/swlx0.pdf>.

[8] Mahmoud, Qusay H. Oracle.Getting Started with Java Message Service (JMS). Oracle.Getting Started with Java Message Service (JMS). [En línea] noviembre de 2004. [Consultado el: 2 de mayo de 2014.]. Disponible en: <http://java.sun.com/developer/technicalArticles/Ecommerce/jms/>.

[9] Marciniak, P. LINUX+ : Android en la era digital al alcance de tu dispositivo favorito.[En línea]. Edición 10 (2010)[Consultado el: 1 de Diciembre de 2013]

[10] Nova está poniendo a punto una nueva plataforma de desarrollo. [En línea].La Habana: blog de software libre, Dariem Pérez Herrera, 2 de Octubre del 2013. [Consultado el: 12 de Noviembre 2013] Disponible en: <http://humanos.uci.cu/2013/10/nova-esta-poniendo-a-punto-una-nueva- plataforma-de-desarrollo/>.

[11] Novell, 2011. openSUSE:Build Service Tools - openSUSE. [En línea], [Consultado: el 17 de Diciembre 13]. Disponible en: <http://en.opensuse.org/openSUSE:Build_Service_Tools>.

[12] Oramas, Goñi Angel . Nova 3.0, avances y expectativas de la distribución cubana de GNU/Linux. [En línea]. Vol. 4, No. 6 (2011). [Consultado el: 28 de Septiembre del 2013]. Disponible en: <http://publicaciones.uci.cu/index.php/ SC/article/view/695 >.

[13] Offut,J,"Quality Attributes of Web Software Applications", en IEEE Software, marzo-abril de 2002. pp. 25-32.

[14] Pierra, Fuentes Allan. Nova distribución Cubana de GNU/LINUX, Tesis (Maestría en informática aplicada). La Habana, Cuba: Universidad de la Ciencias Informáticas, 2013. pp. 29.

[15] Pressman, Roger S. Ingeniería del software. Un enfoque práctico. s.l: McGraw-Hill, 2009.

[16] Pérez, Herrera, Dariem. Sistema distribuido para automatizar la construcción de repositorios de paquetes binarios de la distribución cubana de GNU/Linux Nova. Tesis (Maestría en Ciencias de la Computación). Santa Clara, Cuba: Universidad Central "Marta Abreu" de Las Villas, 2013. pp 25-32.

[17] DistroWatch.com: Nova. [En línea]. [Accedido: 02-mar-2014]. Disponible en:

<http://distrowatch.com/table.php?distribution=Nova>.

[18] Holovaty, Adrian y Kaplan Jacob. The Definitive Guide to Django: Web Development Done Right. Apress, Diciembre de 2007.

[19] Gonzales, Duque Raúl. Python para todos. España, Mayo del 2007.

[20] Larman, Crai. UML y Patrones: Una introducción al análisis y diseño orientado a objeto y al proceso unificado 2da edición. España, 2003

[21] Videla, Alvaro y Williams Jason. RabbitMQ in Action. Estados unidos, 2012.