

Envío: 27-01-2013

Aceptación: 30-01-2013

Publicación: 19-02-2013

# SIRDOM. SISTEMA PARA LA GESTIÓN DEL SERVICIO DE RESOLUCIÓN DE NOMBRES DE DOMINIOS

---

**SIRDOM. MANAGEMENT SYSTEM FOR THE RESOLUTION  
NAMES DOMAINS SERVICE.**

Yoedusvany Hernández Mendoza<sup>1</sup>

Yordanis Arencibia López<sup>2</sup>

Yankier Crespo González<sup>3</sup>

1. Máster, Ingeniero Informático. Profesor del Departamento de Redes, UNICA.
2. Máster, Ingeniero Informático. Profesor del Departamento de Redes, UNICA.
3. Máster, Ingeniero Informático. Profesor del Departamento de Redes, UNICA.

## RESUMEN

Este artículo presenta un estudio del comportamiento del servicio DNS, su funcionamiento, herramientas y por último se propone un sistema informático que permite configurar y gestionar dicho servicio a través de una serie de prestaciones y facilidades que las aplicaciones actuales no posibilitan. Este sistema permitirá gestionar el servicio de resolución de nombres de dominio sobre BIND en su versión 9.

## ABSTRACT

This paper presents a study of the behavior of DNS, its operating principle, tools and finally proposes a computer system to configure and manage this service through a number of benefits and facilities that do not allow current applications. This system will manage the service of domain name resolution on BIND version 9.

## PALABRAS CLAVE

Bind, DNS, dominio, resolución, sistema.

## KEYWORDS

Bind, DNS, domain, resolution, system.

## INTRODUCCIÓN

Las diferentes instituciones y organizaciones, siendo los centros educacionales unos de los principales, han tenido que cambiar sus esquemas tradicionales para adaptarse a la actual era de la información. El correo electrónico, voz sobre IP (protocolo de Internet), la Web, videoconferencias y los servicios de mensajería instantánea (Chat) son algunas de las ventajas que han convertido al correo tradicional en algo casi obsoleto en lo que a comunicación se refiere.

A partir del auge alcanzado por estas tecnologías, los responsables de brindar y supervisar este servicio en las instituciones se enfrentan una serie de dificultades como es el caso de las direcciones IP y el abuso mental que provocaría tener que acceder a los distintos sitios por mediación de ellos. Esta fue la principal causa de la aparición del Sistema de Nombres de Dominio (DNS por sus siglas en inglés, Domain Name System), el servidor encargado de convertir las direcciones IP en nombres.

El presente trabajo se origina a partir de la necesidad de implementar una herramienta para la gestión del servicio DNS que posibilite elevar la calidad del mismo; además, permitir que los usuarios de los Sistemas Operativos (SO) Linux cuenten con un paquete para instalar y configurar de manera más amena y segura dicho servicio en las redes de datos. En la misma concurren varios procesos que se realizan de forma lenta, engorrosa y con riesgos de presentar fallas de seguridad y la no visibilidad de sitios Web. Esto se debe a que no existe una aplicación en específico que integre las funcionalidades necesarias y que contenga interfaces visuales que faciliten el proceso de gestión del servicio DNS, pues la puesta en marcha del mismo requiere conocimientos avanzados sobre el sistema operativo Linux y se realiza mediante la modificación directa de ficheros de configuración; estos factores negativos provocan que el proceso de gestión del servicio no se pueda realizar de forma eficaz.

El artículo se estructura de la siguiente manera: La sección 2 presenta un estudio de tendencia evolutiva del servicio DNS, seguidamente, en la sección 3, se brinda los fundamentos de funcionamiento del DNS, se presenta los tipos de servidores DNS que existen, los componentes, los registros que estos poseen y tipos de consultas que pueden realizarse. En la sección 4 se presentan las herramientas existentes para la gestión del servicio DNS.

## TENDENCIA EVOLUTIVA DEL SERVICIO DNS

El problema de convertir nombres a direcciones físicas es tan viejo como las redes de computadoras. La configuración simple de ficheros fue típicamente usada para ejecutar la traducción de las direcciones. El sistema de nombres de dominios nace de la necesidad de resolver los nombres de dominio a direcciones IP y viceversa. En los años 80 se crea el servicio que hoy se conoce como Sistema de Nombres de Dominio (DNS), creado por Paul Mockapetris en colaboración con Jon Postel de la Universidad del Sur de California y posteriormente, Paul Vixie.

Este servicio permite a los usuarios acceder a los sitios mediante nombres, así no se tiene que memorizar las direcciones IP de los servidores.

O sea, en Internet, la comunicación entre los equipos y los humanos se facilita por el hecho de que los primeros tienen asignado un nombre, de esta forma, se recuerda más fácil el nombre de una máquina ya que se puede asociar éste a la organización o lugar en el que se encuentra, sin tener que memorizar la dirección IP del equipo.

En [1] se presenta un estudio realizado sobre la cronología de los DNS, donde descubre que: originalmente, el uso del DNS involucra solamente instituciones académicas, de investigación y por supuesto, la milicia de los EEUU. Eran los tiempos en que las universidades empezaban a realizar su conexión a las múltiples redes. Algo empezaba a trascender y era importante establecer un orden en cuanto a los equipos que ingresaban a la red.

Sin embargo [2] menciona que compañías como IBM con la Arquitectura de sistemas de redes (SNA) que contenía un mainframe rudimentario con una base de datos para la traducción de nombres que fue originalmente publicado en 1974 constituyó una de los primeros intentos para tal fin. También el Modelo desarrollado por el International Organization for Standardization (ISO) definió los servicios de traducción de Direcciones/Nombres en la capa de transporte (Capa 4) publicado en 1978. Y por último NetBIOS proveyó el NetBIOS Name Server (NBNS) que fue definido en 1984 y que posteriormente se transformaría en el servicio de asignación de nombre de Internet de Microsoft Windows (WINS).

[3] menciona que en los principios de Internet, cuando todavía ni siquiera se le llamaba así, cada computadora tenía que estar al tanto de otras computadoras en la red y tener además una lista de todos los hosts importantes en un fichero llamado "hosts.txt". Este archivo era muy similar al archivo de /etc/hosts sobre computadoras de UNIX y al archivo de lmhosts sobre computadoras de Windows. Además, fue mantenido por una autoridad central, la InterNIC, donde las computadoras lo transferirían periódicamente. Los administradores enviaban por correo los cambios a la InterNIC, que incluiría los cambios luego. Entonces, cuando el tiempo avanzaba, las computadoras actualizaban este fichero con la más reciente versión.

Sin embargo existía un problema con este sistema tan simple; cuando el archivo y la red crecen, el mantenimiento y la distribución de un archivo único de un punto solo, era un trabajo bien engorroso. En 1983, la Internet estaba en crecimiento rápidamente, y la incompetencia de los ficheros hosts.txt era obvia ya que la red había crecido grandemente, y era fácil ver los problemas de actualización de este sistema.

Según [4] cuando el ARPANET se movió a TCP/IP, la población de la red estalló. Ahora había un montón de problemas con los archivos HOSTS.TXT. Por ejemplo, en términos de tráfico y carga de procesamiento involucrada en distribuir el archivo, esta se estaba haciendo muy costosa. Las colisiones de nombres (dos hosts con el mismo nombre en el fichero hosts.txt) y la no autorización del NIC sobre los nombres de los anfitriones impedían que este garantizara la singularidad en este fichero. Otro problema presentado fue la consistencia, ya que para mantener esta era bien difícil ya que en lo que un host de un extremo a otro se actualizaba podía haber surgido uno nuevo o cambiado otro.

El problema esencial según [4] fue que el mecanismo de los hosts.txt no se adaptó bien. Sucesivamente la directiva del ARPANET estableció los estatutos de una investigación para desarrollar un sucesor del sistema de hosts.txt Su objetivo fue crear un sistema que solucionara los problemas antes presentados, constituiría así el principal objetivo descentralizar la administración de esta información para así aliviar el tráfico, así los subnodos y la administración local realizarían las tareas de actualización más fácil. Según el autor anterior el sistema debía utilizar un sistema de nombres de dominio jerárquico para así asegurar la singularidad de los nombres.

Paul Mockapetris del USC's Information Sciences Institute, fue entonces el responsable de diseñar el nuevo sistema. En 1984, liberó el RFC 882 y 883 los cuáles describían el Sistema de Nombre del Dominio. [4]

En 1988, sin embargo, DARPA decidió que el experimento estaba terminado. El Ministerio de Defensa empezó a dismantelar el ARPANET. Otra red, el NSFNET, financiado por la fundación nacional de ciencia, reemplazaron el ARPANET como el nodo central de la Internet.

En la primavera de 1995, la Internet hizo una transición de usar el NSFNET públicamente financiado como un nodo central a usar nodos centrales de comercial múltiples, dirigido por compañías de telecomunicaciones, como SBC, MFS y UUNET.

Según [3] con la aparición de ARPANET, se definió un concepto de nombres de dominio que tuvo lugar en 1981 reflejado en el RFC 799 y la especificación definitiva para el Sistema de Nombres de Dominio de Internet tal como se conoce hoy en día fue publicado en 1987 en los RFC 1034 y 1035, aunque se han publicado subsecuentes RFCs que han modificados ciertos comportamientos del DNS, pero el núcleo de funcionalidades sigue intacto.

Se crearon entonces los nombres de dominio genéricos de primer nivel (GTLD, generic Top-level Domain o dominios de 1ra generación): .com, .net y .org, con el

fin de ubicar el tipo de entidades que buscaban tener presencia en Internet. Además de estos GTLD se empezó por delegar los sufijos nacionales (nTLD, national Top-level Domain o dominios nacionales) a los países que se fueran conectando a la red.

En el año 1995 empieza la gran expansión de Internet, desde entonces se han superado todas las expectativas. En este año la World Wide Web (www) se consolida como el primero de los servicios que ofrece la red.

Para mediados de 1996, Jon Postel, el director del Internet Assignet Numbers Authority (IANA), y el organismo administrador de las direcciones de IP y nombres de dominio, realizó una propuesta en la que contemplaba la creación de 150 nuevos nombres de dominios genéricos, GTLD, así como el .com, .net y .org. Esta propuesta pronto tuvo efectos importantes y finalizó en la formación de un grupo que se encargaría de discutir el re-diseño de los gTLD.

De esta forma, en Noviembre de 1996 nació el Internet-International Ad Hoc Committee (IAHC) impulsado por la Sociedad de Internet (ISOC), a los tres meses de haberse creado se generó el reporte final, en donde se planteaban las recomendaciones y requerimientos para un nuevo esquema de GTLD, este documento recibiría el nombre de “Memorando de Entendimiento para los Nombres de Dominio genéricos de Nivel Superior”. Y así dio pauta al establecimiento de las reglas que aplicarían a Internet en los próximos años.

## FUNDAMENTOS DE FUNCIONAMIENTO DEL DNS

Según [4] el sistema de nombre del dominio es básicamente una base de datos de información de hosts.

El DNS es una base de datos distribuida y jerárquica que almacena información asociada a nombres de dominio en redes como Internet. Aunque como base de datos, el DNS es capaz de asociar diferentes tipos de información a cada nombre, los usos más comunes son la asignación de nombres de dominio a direcciones IP y la localización de los servidores de correo electrónico de cada dominio. [5]

Cada nombre de dominio es esencialmente solo un camino en árbol extenso, llamado el espacio de nombres de dominio. Según [6] estos son separados por puntos (.), son además, procesados de izquierda a derecha. Los Top Level Domains (TLD) son definidos en la raíz del dominio. Existen dos clases de dominios principales, los Generic Top Level Domain (GTLD) y el Country Code Top Level Domain (ccTLD). La profundidad del árbol es de 127 niveles, es decir, un nombre de dominio como máximo puede contener 127 sub-dominios, y cada nodo puede contener hasta 63 caracteres sin puntos incluidos y puede contener letras, números, guiones y otro conjunto de caracteres que no son soportados por todas las aplicaciones, por esta razón no son frecuentemente utilizados. Los guiones no pueden estar al principio o al final de la cadena. Esta cadena puede contener como máximo 255 caracteres.

La estructura sería así: string.string.string.....string. , donde la primera cadena es el nombre de una computadora, y sucesivamente nombres de computadoras que representan un subdominio hasta llegar al último nombre que sería el host deseado. La siguiente figura (Fig. 1) muestra la estructura abstracta que tiene un DNS. Supongamos que se quiere acceder al host de la extrema derecha que está más abajo en la estructura, su cadena se conformaría de la siguiente forma: bc.ca.

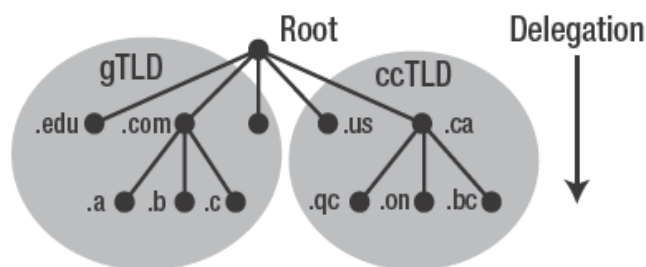


Fig. 1. Estructura arbórea y jerárquica del sistema de nombres de dominio. Fuente: [2]

Los nodos que son hijos del mismo padre tienen que tener nombres diferentes para garantizar la unicidad. Sin embargo esta condición no es restrictiva para nodos con distintos padres. Si se revisa el concepto de DNS podemos identificar a los nodos como los host a los cuáles se les asigna el nombre de dominio, ya que los nombres de dominio son sólo índices en la base de datos del DNS.

En la bibliografía consultada aparece el término autoridad en reiteradas ocasiones, el cual es definido en [2] como organización o persona responsable para la

administración y operación del dominio al cual pertenece. Al mismo tiempo esta puede ser delegada a un sub-dominio (segundo, tercer y demás niveles). Por ejemplo la autoridad para el dominio raíz la tiene el Internet Corporation for Assigned Numbers and Names (ICANN—www.icann.org/), para los GTLD, la ICANN y delegados a varios registradores acreditados; y para los ccTLD, la ICANN delega en los países. En el caso de los ccTLD, los países crean sus propias reglas de delegación.

Sin embargo no siempre las comunicación es entre usuario-máquina, existe entre máquina-máquina y esta es basada en direcciones IP, no con nombres de dominio. Por otra parte algunas aplicaciones necesitan encontrar un nombre para una dirección IP, en otras palabras, encontrar un “registro reverso”. Este proceso de traducción de un IP a un nombre de dominio es llamado “traducción reversa”. [6]

Para lograr esta funcionalidad es necesario crear una estructura de árbol de direcciones IP. Los dominios creados por direcciones IP son llamados “dominios reversos”, Los pseudo-dominios inaddr-arpa para IPv4 and ip6.arpa para IPv6 fueron creados para estos propósitos de traducción inversa, estos nombres tienen orígenes históricos, son acrónimos para la direcciones inversas de ARPANET. La siguiente figura (Fig. 2) muestra la estructura de estas direcciones almacenadas en el dominio inverso.

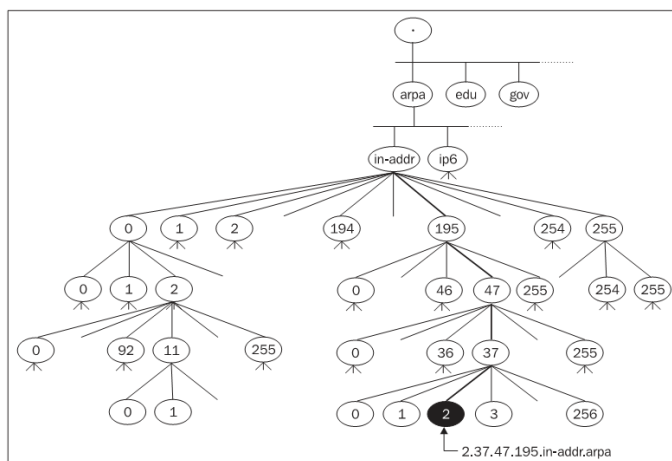


Fig. 2. Dominio inverso para la dirección IP 195.47.37.2. Fuente: [6]

Bajo el dominio in-addr.arpa. Hay varios dominios (de 0 a 255 y así sucesivamente para sus sub-dominios) con el primer número de la dirección IP. Por ejemplo el dominio para el nodo señalado en la figura sería 2.37.47.195.in-addr.arpa, hay que notar que estos son construidos igual que las direcciones de red lo que de atrás hacia delante.

Rápidamente se ha abordado los principales conceptos del principio de funcionamiento de DNS, para que este sea útil es necesario solicitarles información, estos reparten información en forma de preguntas o de respuestas a una computadora de un usuario o cualquier aplicación (cliente FTP, agente SMTP, etc.) que necesite resolver nombres a direcciones IP. La siguiente figura (Fig. 3) muestra como un navegador ejecutándose en una computadora usa y accede al DNS.



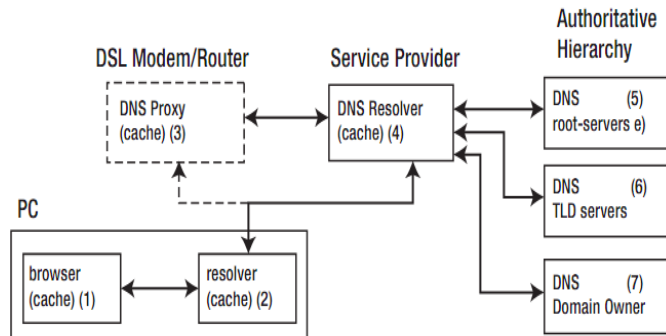


Fig. 3. Un ejemplo de funcionamiento del sistema DNS. Fuente: [2]

Para la operación práctica o gestión del sistema DNS se utilizan tres componentes principales según [5]:

- Los Clientes DNS: Un programa cliente DNS que se ejecuta en la computadora del usuario y que genera peticiones DNS de resolución de nombres a un servidor DNS.
- Los Servidores DNS: Que contestan las peticiones de los clientes. Los servidores recursivos tienen la capacidad de reenviar la petición a otro servidor si no disponen de la dirección solicitada.
- Zonas de autoridad: porciones del espacio de nombres de dominio que almacenan los datos. Cada zona de autoridad abarca al menos un dominio y posiblemente sus subdominios, si estos últimos no son delegados a otras zonas de autoridad.

Como se puede apreciar en la figura anterior el DNS hace uso extensivo de la cache, el cual puede jugar un rol significativo reduciendo la complejidad del sistema y el tiempo de respuesta del mismo. Según [2] la cache es el almacenamiento de cualquier resultado (de las respuestas a consultas) salvado temporalmente.

Una vez el usuario realiza la petición a través del navegador, ya que generalmente no se comunican directamente con el servidor DNS, la resolución de nombres se hace de forma transparente por las aplicaciones del cliente. Ej.: Navegadores Web., clientes de correo, etc. este último le entrega esta petición a un “resolver”, que puede estar contenido internamente por la aplicación, y el mismo es definido por [4] como clientes que acceden al servidor de nombres de dominio, pero que además son programas que se ejecutan en un host que necesita información desde el servidor de nombres de dominio, aunque estos verifican primero su cache en busca de la información solicitada por la aplicación que realizó la petición, impidiendo una comunicación externa y tráfico en la red. Estos tienen la responsabilidad de:

- Consultar al DNS.
- Interpretar las respuestas (las cuáles pueden ser registros de recursos o un error); y
- Retornar la información al programa que solicitó esta.

Al realizar una petición que requiere una búsqueda de DNS, la petición se envía al servidor DNS local del sistema operativo. El sistema operativo, antes de establecer alguna comunicación, comprueba si la respuesta se encuentra en la memoria caché.

En el caso de que no se encuentre, la petición se enviará a uno o más servidores DNS, de no encontrarse la respuesta adecuada se devuelve un error.

La información almacenada en los servidores de nombres de dominio se almacena en “zonas” que es definida por [6] como parte del espacio de nombres de dominio que administrado por un servidor de nombres en particular. Una zona contiene datos del dominio ubicado en el nivel más bajo de la estructura hereditaria que posee el sistema DNS y es usualmente llamada zona subordinada.

Debe señalarse que los servidores DNS pueden ser según [5]:

1. **Preferidos:** Guardan los datos de un espacio de nombres en sus ficheros.
2. **Alternativos:** Obtienen los datos de los servidores primarios a través de una transferencia de zona.
3. **Locales o caché:** Funcionan con el mismo software, pero no contienen la base de datos para la resolución de nombres. Cuando se les realiza una consulta, estos a su vez consultan a los servidores secundarios, almacenando la respuesta en su base de datos para agilizar la respuesta a las peticiones en el futuro.

En el caso de los clientes le pueden hacer las consultas a los servidores de dos maneras:

- **Recursiva:** que consisten en la mejor respuesta que el servidor de nombres pueda dar. El servidor de nombres consulta sus datos locales (incluyendo su caché) buscando los datos solicitados.
- **Iterativa:** el servidor no tiene la información en sus datos locales, por lo que busca un servidor raíz y repite el mismo proceso básico (consultar a un servidor remoto y seguir a la siguiente referencia) hasta que obtiene la respuesta a la pregunta.

El Servidor DNS en su gestión cuenta con una serie de registros, como son los publicados por [5]:

- **A = Address – (Dirección)** Este registro se usa para traducir nombres a direcciones IPv4.
- **AAAA = Address – (Dirección)** Este registro se usa en IPv6 para traducir nombres de hosts a direcciones IPv6.
- **CNAME = Canonical Name – (Nombre Canónico)** Se usa para crear nombres de hosts adicionales, o alias, para los hosts de un dominio. Es usado cuando se están corriendo múltiples servicios (como FTP y web server) en un servidor con una sola dirección IP. Cada servicio tiene su propia entrada de DNS (como ftp.ejemplo.com. y www.ejemplo.com.). esto también es usado cuando corre múltiples servidores http, con diferentes nombres, sobre el mismo host.
- **NS = Name Server – (Servidor de Nombres)** Define la asociación que existe entre un nombre de dominio y los servidores de nombres que almacenan la información de dicho dominio. Cada dominio se puede asociar a una cantidad cualquiera de servidores de nombres.
- **MX (registro) = Mail Exchange – (Registro de Intercambio de Correo)** Asocia un nombre de dominio a una lista de servidores de intercambio de correo para ese dominio.
- **PTR = Pointer – (Indicador)** También conocido como 'registro inverso', funciona

a la inversa del registro A, traduciendo IPs en nombres de dominio.

- SOA = Start of authority – (Autoridad de la zona) Proporciona información sobre el servidor DNS primario de la zona.
- HINFO = Host INfOrmation – (Información del sistema informático) Descripción del host, permite que las personas conozcan el tipo de máquina y sistema operativo al que corresponde un dominio.
- TXT = Text - (Información textual) Permite a los dominios identificarse de modos arbitrarios.
- LOC = Localización - Permite indicar las coordenadas del dominio. o WKS - Generalización del registro MX para indicar los servicios que ofrece el dominio. Obsoleto en favor de SRV.
- SRV = Servicios - Permite indicar los servicios que ofrece el dominio.
- SPF = Sender Policy Framework - En este registro se especifica cual o cuales hosts están autorizados a enviar correo desde el dominio dado.
- El servidor que recibe consulta el SPF para comparar la IP desde la cual le llega, con los datos de este registro.

## HERRAMIENTAS EXISTENTES PARA LA GESTIÓN DEL SERVICIO DNS.

Existen hoy día un número grande de utilidades para DNS, muchas de ellas son específicas para determinados DNS como BIND y otras que son utilizadas en varias plataformas. A continuación presentamos muchas de estas utilidades, las cuales cumplen diferentes roles, incluso muchas constituyen DNS.

- nslookup: Utilidad para interrogar un servidor DNS. Está disponibles en múltiples plataformas como Windows, \*nix (Linux, Unix, BSD, etc.).
- named-checkconf: Esta utilidad chequea la sintaxis del fichero de configuración named.conf en plataformas \*nix.
- named-checkzone: Chequea la sintaxis y consistencia del fichero zone.conf en ambientes \*nix.
- dnswalk: Es un script escrito en Perl para la depuración de DNS.
- dig: Utilidad para interrogar un servidor DNS. Está disponible para plataformas que soportan BIND.
- rndc: El programa rndc (control de servidor de nombres) es un programa que permite controlar un servidor de nombres remotamente de forma segura.
- nsupdate: Utilidad para actualizar dinámicamente los ficheros de zonas.
- named-compilezone: Utilidad para verificar zonas y convertirlas en formato binario para una rápida carga.
- named-checkzone: Utilidad para verificar ficheros de zonas.
- named-checkconf: Utilidad para chequear la sintaxis del fichero named.conf
- host: Herramienta simple para hacer búsquedas en Servidores DNS. Es utilizada para convertir nombres en direcciones IP y viceversa.[7]
- Cisco Network Register (CNR): Incluye un servidor DNS comercial de Cisco Systems, es compatible con las altas tasas de actualización dinámica.[8]
- Pdnsd: Es una caché de DNS del servidor proxy que almacena en caché los registros DNS en el disco para la retención a largo plazo. [9]
- Unbound: Es una validación, recursivo y el servidor de almacenamiento en caché de DNS diseñado para alto rendimiento. Fue lanzado el 20 de mayo 2008 (versión 1.0.0) en forma de programas de software libre licenciado bajo la licencia BSD por NLnet Labs, Inc. Verisign, Nominet, y Kirei.[10]
- Posadis: Es un programa gratuito de servidor DNS, escrito en C++, con el apoyo de actualización de DNS dinámico.[11]
- Nominum ANS: Es un servidor comercial autorizado de Nominum, una empresa fundada por Paul Mockapetris, el inventor del DNS. ANS ha sido diseñado para satisfacer las necesidades de los principales servidores de nivel de dominio, de alojamiento y las grandes empresas.[12]
- MaraDNS: Es un servidor DNS, desarrollado por Sam Trenholme que cuenta con buena seguridad y facilidad de uso, bajo licencia GNU.[13]
- PowerDNS: Es un programa gratuito de servidor DNS con una variedad de almacenamiento de datos back-end y las características de balanceo de carga.

Funciones de servidor con autoridad y recursiva se implementan como aplicaciones independientes. [14]

- NSD: Es un servidor de software libre de autoridad proporcionada por los laboratorios de NLnet. NSD. Alberga varios dominios de nivel superior, y cuenta con tres de los servidores de nombres raíz.[15]
- Simple DNS Plus, es un producto comercial de servidor DNS que se ejecuta en Microsoft Windows con un énfasis en una sencilla interfaz gráfica de usuario.[16]
- Djbdns: Es una colección de aplicaciones DNS, incluyendo tinydns, que fue el segundo más popular de software libre servidor DNS en el año 2004. Fue diseñado por Daniel J. Bernstein, autor de gmail, con énfasis en las consideraciones de seguridad. [17]
- Microsoft DNS: Es el servidor DNS que proporciona Windows Server, un componente de Active Directory de Microsoft.[18]
- Net::DNS: Es un DNS resolver implementado en Perl Permite ejecutar cualquier tipo de consulta DNS desde scripts de Perl.[19]
- Plataforma de gestión de servicios telemáticos en Gnu/Linux: Se desarrolló con el objetivo de proveer al administrador de red cubano de una herramienta potente y eficiente para la administración de los servicios telemáticos de una red de ordenadores. [20]
- DNS Manager: Una solución diseñada específicamente para ambientes de virtualización en nubes. Puede ser administrado por una interface Web. o a través de una interface de Línea de comandos.[21]
- NicTool: Otra solución para la administración de DNS. Utiliza varios componentes semejantes a la propuesta de este trabajo, por ejemplo almacenamiento de datos a través de SQL, tinydns, BIND, PowerDNS, Web Administration, etc.[22]
- DNS - Nixu Secure Name Server (SNS): Otra solución para virtualizar el servicio DNS que incluye administración vía Web. [23]
- DNS2Go: Herramienta de administración para DNS.[24]
- DnsBindEditor: Herramienta multiplataforma propietaria, sirve para configurar los ficheros name.conf y zone del BIND. Es una aplicación tipo Escritorio. [25]

## DESCRIPCIÓN DE LA HERRAMIENTA PROPUESTA EXISTENTES

Los servicios telemáticos en plataformas GNU/Linux tiene sus ventajas y desventajas.

Entre las ventajas se encuentra la seguridad que estos brindan, la cantidad de recursos que se dedican o queda disponible para que el sistema utilice estos ya que los ambientes se instalan sin un entorno de escritorio normalmente lo que posibilita un mejor aprovechamiento y disponibilidad de los recursos.

Entre las desventajas: no contar con una interfaz gráfica para la configuración de los servicios, necesidad de vastos conocimientos sobre los sistemas GNU/Linux, fallas de seguridad producto a mala configuración de los ficheros que posee un determinado servicio; tiempo prolongado en la configuración y puesta a punto del servicio, desconocimiento de las cláusulas de configuración, gran desaprovechamiento de las funcionalidades de este servicio.

Por todas estas razones se desarrolló la herramienta propuesta. La misma resuelve las desventajas que los sistemas analizados.

Esta herramienta depende de otros sistemas como son; BIND, Servidor Web. Apache, Gestor de Base de Datos MySql.

Además se utilizaron lenguajes de programación como son PHP para la programación del lado servidor y llamadas a comandos que se ejecutan en modo consola, Bash, PERL para la interacción con los ficheros de Linux, framework de desarrollo de lado cliente, ExtJS v3.2, este utiliza otros lenguajes y tecnologías como CSS, Javascript, DOM, JSON, Ajax, etc.

La metodología de desarrollo que se utilizó para la realización de esta herramienta fue la metodología Ágil extreme Programing (XP). Que según [26] es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en la retroalimentación continua entre el cliente y el equipo de desarrollo, la comunicación fluida entre todos los participantes, la simplicidad en las soluciones implementadas y el coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.

Esta metodología posee cuatro características principales que se mencionan en [27]:

- Los individuos e interacciones son más importantes que los procesos y las herramientas: dado que el proceso de desarrollo es creativo, no es posible pensar que las personas funcionen respondiendo a órdenes, a procesos rígidos.
- Que el software funcione es más importante que la documentación exhaustiva: puesto que si el software no funciona la documentación no vale de nada. A nivel interno puede haber documentación, pero solo la necesaria y a nivel externo lo que el cliente requiera.

- La colaboración con el cliente es más importante que la negociación de contratos: supone que la satisfacción del cliente con el producto será mayor, mientras exista una conversación y retroalimentación continua entre éste y la empresa.
- La respuesta ante el cambio es más importante que el seguimiento de un plan: puesto que si un proyecto de software no es capaz de adaptarse a los cambios fracasará, especialmente en productos de gran envergadura. La estrategia de planificación se basa en: planes detallados para las próximas semanas, planes aproximados para los próximos meses y muy generales para plazos mayores.

Los requisitos funcionales que posee esta herramienta son:

1. Gestionar Opciones Generales
2. Gestionar ACL.
3. Gestionar Rangos de IP.
4. Gestionar Registro SOA.
5. Gestionar Vistas.
6. Gestionar Zonas.
7. Gestionar Registros (records).
8. Gestionar la seguridad del sistema.

A continuación se muestra la Historia de usuarios (Tabla 1), uno de los principales artefactos que posee esta metodología

Iteración		Historia de Usuario	Pto. Est	Pto. Real
1	1	Gestionar Opciones Generales.	1	1
2	2	Gestionar ACL.	0.4	0.4
	3	Gestionar Rangos de IP.	1	1
3	4	Gestionar Registro SOA.	0.8	0.8
4	5	Gestionar Vistas.	1	1
5	6	Gestionar Zonas.	0.8	0.8
6	7	Gestionar Registros (records).	0.8	0.8
7	8	Gestionar seguridad del sistema	0.8	0.8
8	9	Diseño de la Base de Datos	0.4	0.4

**Tabla 1: Historias de usuarios. Fuente: Elaboración propia.**

## ANÁLISIS Y DISEÑO DEL SISTEMA

Según la metodología utilizada el principal artefacto que se presenta en esta fase son las tareas de ingeniería

Historias de Usuarios	Tareas de Ingeniería		Pto. Est.	Pto. Real
1	A-1	Elaboración del diseño de la interfaz Gestionar Opciones Generales.	0.4	0.4
	A-2	Implementación del módulo Gestionar Opciones Generales.	0.6	0.6
2	B-1	Elaboración del diseño de la interfaz de ACL.	0.4	0.4
	B-2	Elaboración del diseño de la interfaz de Rangos de IP.	0.4	0.4
	B-3	Implementación del módulo Gestionar Rangos de IP y ACL.	0.6	0.6
3	C-1	Elaboración del diseño de la interfaz de Registro SOA.	0.4	0.4
	C-2	Implementación del módulo Gestionar Registros SOA.	0.4	0.4
4	D-1	Elaboración del diseño de la interfaz de Vistas.	0.4	0.4
	D-2	Implementación del módulo Gestionar Vistas.	0.6	0.6
5	E-1	Elaboración del diseño de la interfaz de Zonas.	0.4	0.4
	E-2	Implementación del módulo Gestionar Zonas.	0.4	0.4
6	F-1	Elaboración del diseño de la interfaz de Registros (records).	0.4	0.4
	F-2	Implementación del módulo Gestionar Registros (records).	0.4	0.4
7	G-1	Implementación de la seguridad del sistema	0.8	0.8
8	H-1	Diseño de la Base de Datos	0.4	0.4

**Tabla 2: Tareas de Ingeniería. Fuente: Elaboración propia.**



## VALIDACIÓN DEL SISTEMA

Para la corroboración del sistema se analizaron varias técnicas, métodos y herramientas para verificar el comportamiento, seguridad y cumplimientos de los objetivos trazados. Para esto se utilizaron las pruebas de aceptación y unitarias que según la metodología seleccionada son necesarias, las mismas fueron satisfactorias.

Al sistema se le aplicó un Modelo de evaluación para software propuesto por [28]. El mismo plantea con el fin de entregar a los clientes productos satisfactorios, el software debe alcanzar ciertos niveles de calidad. Para alcanzar buenos niveles de calidad el número de defectos necesita mantenerse bajo mínimos. Según estos autores el modelo se basa en seis atributos que permiten definir o aproximarse a medir la calidad del software, estos atributos se dividen en varias características a evaluarse, dándole un peso numérico que sumado da un total de 100 puntos. Una vez ponderado cada característica de cada atributo el sistema puede evaluarse como aceptable ya que se obtuvo un puntaje promedio de 82.7 de 5 evaluaciones realizadas por especialistas en redes.

Puntaje	Evaluación de la propuesta	Calidad
0 - 33	Mala	Inaceptable
34 – 66	Regular	Dudosa
<b>67 - 100</b>	<b>Buena</b>	<b>Aceptable</b>

**Tabla 3: Rangos de puntos para la evaluación del software. Fuente: Elaboración propia.**

Además, al sistema se le aplicó una plataforma libre para verificar la seguridad de aplicaciones Web conocida como Subgraph Vega, la cual es desarrollada y atendida por Subgraph Technologies, Inc. y otros, 2011. La misma verifica posibles ataques de inyección SQL, posibles declaraciones del sistema de ficheros y otras vulnerabilidades frecuentes en el desarrollo de aplicaciones Web. Debe mencionarse que al ejecutarse la prueba correspondiente al sistema en cuestión no se encontraron vulnerabilidades en el mismo.

## CONCLUSIONES

En este artículo se analizó la tendencia evolutiva del sistema de nombres de dominio, su funcionamiento, las herramientas y sistemas que existen hoy día para la gestión del servicio DNS y el estado actual de la gestión de dicho servicio.

El sistema propuesto respondió positivamente a las pruebas realizadas. La herramienta SIRDOM permite mejorar de forma eficaz el proceso de gestión del servicio DNS de manera sencilla, respondiendo a las necesidades del grupo de administradores de red de la Universidad de Ciego de Ávila.

## REFERENCIAS BIBLIOGRÁFICAS

- [1] **OSCAR, R** "CRONOLOGÍA DEL DNS."
- [2] **AITCHISON, R.** *PRO DNS AND BIND 10*, 2ND ED. SPRINGER, 2011.
- [3] **LANGFELDT.** *CONCISE GUIDE TO DNS AND BIND, THE.* QUE, 2000.
- [4] **ALBITZ, P; AND LIU C,** *DNS AND BIND, 5TH EDITION*, FIFTH. UNITED STATES OF AMERICA: O'REILLY, 2006.
- [5] WIKIPEDIA, "DOMAIN NAME SYSTEM," WIKIPEDIA. 2011.
- [6] **DOSTÁLEK, L AND KABELOVÁ,** *DNS IN ACTION.* PACKT PUBLISHING LTD., 2006.
- [7] "SERVICIO DNS," *SISTEMAS OPERATIVOS III - LINUX*, 2012. [ONLINE]. AVAILABLE: [HTTP://DC233.4SHARED.COM/DOC/KPOJIDAZ/PREVIEW.HTML](http://dc233.4shared.com/doc/kPOJIDAZ/preview.html). [ACCESSED: 28-MAY-2012].
- [8] "CISCO NETWORK REGISTRAR (CNR)," 2012. [ONLINE]. AVAILABLE: [HTTPS://SUPPORTFORUMS.CISCO.COM/DOCS/DOC-1330](https://supportforums.cisco.com/docs/doc-1330). [ACCESSED: 28-MAY-2012].
- [9] "PDNSD," *PDNSD (ESPAÑOL)*, 2012. [ONLINE]. AVAILABLE: [HTTPS://WIKI.ARCHLINUX.ORG/INDEX.PHP/PDNSD\\_%28ESPA%C3%B1OL%29](https://wiki.archlinux.org/index.php/Pdnsd_%28Espa%C3%B1ol%29). [ACCESSED: 25-MAY-2012].
- [10] "ABOUT UNBOUND," (2012). [ONLINE]. AVAILABLE: [HTTP://UNBOUND.NET/](http://unbound.net/). [ACCESSED: 25-MAY-2012].
- [11] "POSADIS," *WELCOME TO POSADIS.ORG!*, 2012. [ONLINE]. AVAILABLE: [HTTP://POSADIS.SOURCEFORGE.NET/](http://posadis.sourceforge.net/). [ACCESSED: 25-MAY-2012].
- [12] "NOMINUM," *LOGICALIS NOW // NOVEDADES TECNOLÓGICAS // NOMINUM PRESENTA UNA NUEVA OFERTA COMBINADA PARA REDES EMPRESARIALES*, 2012. [ONLINE]. AVAILABLE: [HTTP://WWW.LA.LOGICALIS.COM/NOTICIAS-Y-EVENTOS/LOGICALIS-NOW/NOVEDADES-TECNOL%C3%B3GICAS/NOMINUM-PRESENTA-UNA-NUEVA-OFE.ASPX](http://www.la.logicalis.com/noticias-y-eventos/logicalis-now/novedades-tecnol%C3%B3gicas/nominum-presenta-una-nueva-ofe.aspx). [ACCESSED: 26-MAY-2012].
- [13] "MARADNS," *MARADNS A SMALL OPEN-SOURCE DNS SERVER*, 2012. [ONLINE]. AVAILABLE: [HTTP://WWW.MARADNS.ORG/](http://www.maradns.org/). [ACCESSED: 26-MAY-2012].
- [14] "POWERDNS," 2012. [ONLINE]. AVAILABLE: [WIKI.POWERDNS.COM](http://wiki.powerdns.com). [ACCESSED: 28-MAY-2012].
- [15] "NSD," *NSD*, 2012. [ONLINE]. AVAILABLE: [HTTP://EN.WIKIPEDIA.ORG/WIKI/NSD](http://en.wikipedia.org/wiki/NSD). [ACCESSED: 27-MAY-2012].
- [16] "SIMPLE DNS PLUS," *THE SIMPLE YET POWERFUL DNS SERVER FOR WINDOWS*, 2012. [ONLINE]. AVAILABLE: [HTTP://WWW.SIMPLEDNS.COM/](http://www.simpledns.com/). [ACCESSED: 27-MAY-2012].
- [17] "DJB DNS," 2012. [ONLINE]. AVAILABLE: [HTTP://EN.WIKIPEDIA.ORG/WIKI/DJBDNS](http://en.wikipedia.org/wiki/Djbdns). [ACCESSED: 27-MAY-2012].
- [18] "MICROSOFT DNS," 2012. [ONLINE]. AVAILABLE: [HTTP://ES.WIKIPEDIA.ORG/WIKI/MICROSOFT\\_DNS](http://es.wikipedia.org/wiki/Microsoft_DNS). [ACCESSED: 27-MAY-2012].
- [19] "NET::DNS," *NET::DNS AND NET::DNS::SEC*. 2012.
- [20] **ANGLADA, R A. LUIS, DE LA ROSA, J, HURTADO A, AND ZULUETA, E.** (2011) "PLATAFORMA DE GESTIÓN DE SERVICIOS TELEMÁTICOS EN GNU/LINUX. MÓDULOS DE DIRECTORIO Y DNS.," UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS, PP. 15.

- [21] "DNS MANAGER," *MULTITENANT SOFTWARE AUTOMATION SOLUTION, DESIGNED TO DELIVER ADVANCED DNS HOSTING TO SERVICE PROVIDERS AND BUSINESSES.*, 2012. [ONLINE]. AVAILABLE: [WWW.4PSA.COM](http://www.4psa.com).
- [22] "NICTOOL: A MANAGEMENT SOLUTION FOR DNS," 2012. [ONLINE]. AVAILABLE: [WWW.NICTOOL.COM](http://www.nictool.com). [ACCESSED: 14-MAY-2012].
- [23] "DNS - NIXU SECURE NAME SERVER (SNS)," 2012. [ONLINE]. AVAILABLE: [HTTP://WWW.NIXUSOFTWARE.COM/INDEX.HTML](http://www.nixusoftware.com/index.html). [ACCESSED: 14-MAY-2012].
- [24] "DNS2GO: A FEATURE-RICH DNS MANAGEMENT TOOL," 2012. [ONLINE]. AVAILABLE: [WWW.SERVERWATCH.COM](http://www.serverwatch.com). [ACCESSED: 14-MAY-2012].
- [25] "DNS BIND EDITOR," *ALL THE POWER OF ISC BIND - MADE EASY WITH THE DNSBINDEDITOR.*, 2011. [ONLINE]. AVAILABLE: [WWW.DNSBINDEDITOR.COM/HOME.XHTML](http://www.dnsbindeditor.com/home.xhtml). [ACCESSED: 16-MAY-2012].
- [26] **BECK, K.** (2000) "EXTREME PROGRAMMING EXPLAINED. EMBRACE CHANGE," ADDISON WESLEY.
- [27] **JOSKOWICZ, J.** (2008). "REGLAS Y PRÁCTICAS EN EXTREME PROGRAMING."
- [28] **MACÍAS RIVERO, M. V. GUZMÁN SÁNCHEZ, AND Y. MARTÍNEZ SUÁREZ.** (2009) "MODELO DE EVALUACIÓN PARA SOFTWARE QUE EMPLEAN INDICADORES MÉTRICOS EN LA VIGILANCIA CIENTÍFICO-TECNOLÓGICA," P. 16.