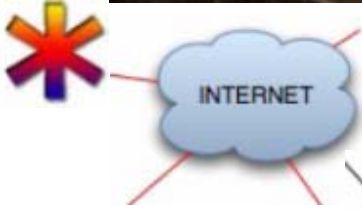




BOLETÍN ELECTRÓNICO



ENGLISH ZONE



RAMA DE ESTUDIANTES IEEE-UNED

30-October-2007 (BOLETÍN N°8)



RAMA DE ESTUDIANTES IEEE-UNED

30-Octubre-2007

COORDINADOR Y EDITOR

Gloria Murillo (gmcordero@indra.es)

REVISIÓN

Manuel Castro

Sergio Martín

Eugenio López

Gloria Murillo

DISEÑO PORTADA

Gloria Murillo

AUTORES

Sergio Martín, Eugenio López, José Antonio Cámara, Germán Carro, Guillermo F. Lafuente, Maria M. García, Gloria Murillo, Serafín Doblado.

**EN COLABORACIÓN CON EL CAPÍTULO ESPAÑOL DEL
IEEE EDUCATION SOCIETY**

AGRADECIMIENTOS

“Agradecemos a nuestro Catedrático de Tecnología Electrónica y profesor consejero de la Rama, Manuel Castro, todo el tiempo y la dedicación que nos presta, así como, el habernos dado la posibilidad de colaborar con el Capítulo Español del IEEE Education Society para la elaboración del mismo. Agradecemos a todos los autores, y a aquellos que han colaborado para hacer posible este Boletín Electrónico”.



ÍNDICE

CONTENIDO

PÁGINAS

SUMARIO.....	4
INFORMACIÓN Y URLS.....	6
PROMOCIÓN DE LA DIRECTIVA DE LA RAMA DE ESTUDIANTES IEEE-UNED PARA DEL AÑO 2007-08.....	10
PATRONES SOFTWARE: LA GRAN BOLA DE LODO.....	12
GALVANIZACIÓN EN CALIENTE.....	18
COMUNICACIONES MOVILES	21
KARAKURI, EL INICIO DE LOS AUTOMATAS EN EL PAÍS DEL SOL NACIENTE (Y II).....	25
CLUSTERS DE SERVIDORES.....	35
LA LENGUA INGLESA Y LA INGENIERIA.....	46
ENGLISH ZONE.....	51
ARTICULO DE EMPRESA ¿QUÉ ES LA VoIP?.....	56
INFORMACIÓN GENERAL RESUMIDA.....	68

SUMARIO

Para comenzar el boletín electrónico nº 8, se presenta como en ediciones anteriores un primer apartado de **Información** general de la Rama y **URLs** de interés general propuestas por miembros de la rama.

El primer artículo del Boletín trata sobre los "**Patrones Software: la gran Bola de lodo**"; Sergio Martín nos habla sobre como diseñar una aplicación parándonos primero a pensar en como diseñar las cosas en lugar de ponernos a "picar código" directamente. Este artículo trata de ayudarnos a programar y diseñar software de una manera mejor, con la explicación de algunas estructuras y arquitecturas que evitarán que se tenga que "reinventar la rueda" cada vez que hay que realizar una aplicación software. A estas estructuras y arquitecturas se les conoce con el nombre de "Patrones". En artículo detalla el patrón "Modelo-Vista-Controlador".

El siguiente artículo que encontramos es el de Eugenio López, "**Galvanización en caliente.**", este método es una técnica muy utilizada en la industria para conseguir resolver el problema de la corrosión en el hierro y acero. En el artículo podemos leer las ventajas que tiene el realizar el galvanizado en caliente así como apreciar la foto de una galvanizadora de la empresa Niedax Kleinhuis.

El artículo escrito por José Antonio Cámara sobre las "**Comunicaciones Móviles**" nos presenta un programa para calcular el radio celular en redes GSM basados en la fórmula de Okumura-Hata y la de Okumura-Hata modificada para 1800-1900 MHz, según el COST 231.

En el Boletín anterior Germán Carro realizó un análisis del nacimiento de los primeros autómatas con el artículo "**Karakuri, el inicio de los autómatas en el país del sol naciente**". Estos artilugios también impulsaron de manera incipiente una primitiva "inteligencia artificial" que fue recogida posteriormente por las formulaciones de Turing y su discurso sobre las máquinas de estados. Es precisamente esta vertiente y su implicación hasta nuestros días, la que va a centrar esta segunda y última parte del artículo sobre tan avanzados inventos mecánicos.

"**Clusters de servidores**", es un artículo escrito por Guillermo F. Lafuente en el cual se nos detallan las tecnologías que consiguen obtener más potencia de cálculo para los ordenadores. Para conseguir este objetivo la tecnología se basa en los clusters que son un grupo de equipos que trabajan de manera conjunta para ejecutar un grupo de aplicaciones comunes y ofrecer la imagen de un único sistema al cliente y a la aplicación.

El artículo escrito por la profesora de la UNED María M. García "**La lengua inglesa y la Ingeniería**", en el artículo de nos habla de como la lengua inglesa se está convirtiendo últimamente en una herramienta imprescindible sobre todo

para el desarrollo profesional de cualquier ingeniero, ya que cada vez nos enfrentamos a más proyectos internacionales. Un aspecto que está fomentando el aprendizaje del inglés es la convergencia al EEES, ya que las universidades españolas se están incorporando a la internacionalización de los estudios universitarios.

“**English Zone**” es una nueva sección que creamos en este Boletín con el objetivo de tratar de refrescar un poco ciertos aspectos de la lengua inglesa que hemos olvidado o así como adquirir nuevos conocimientos. Este artículo está en consonancia con el anterior en el cual hemos podido ver la importancia que tiene la lengua inglesa para nuestro desarrollo de la vida laboral. Esta nueva sección trataremos de ir ampliándola en posteriores boletines, para que tomemos conciencia y nos animemos todos un poco a no perder el inglés que tenemos y a ir aumentando nuestros conocimientos respecto a esta lengua.

Finalmente y para cerrar el Boletín tenemos el artículo de empresa “**¿Qué es la VoIP?**” escrito por Serafín Doblado Moreno, administrador de sistemas de VoIP de la empresa VOCARI. La VoIP, no es más que la evolución natural del sistema tradicional de telefonía. Su funcionamiento está basado en aplicaciones de entornos TCP/IP. En el artículo también se nos introduce el concepto de centralita de software (PBX) de código abierto, Asterisk es una centralita software (PBX) de código abierto que permite interconectar teléfonos y conectar dichos teléfonos a la red telefónica.

VOCARI es una empresa que ofrece tanto servicios avanzados de hosting como de VoIP; y que tiene como mercado principal las PYMES, aunque es totalmente válido para cualquier entidad que quiera sacar rendimiento a su ADSL y ha esta tecnología.

INFORMACIÓN Y URLS

En esta sección se pretende dar información general de la Rama y URLs de interés general propuestas por los distintos miembros que forman parte de la rama de estudiantes.

Este boletín número ocho, es sin duda un punto y seguido en la creación y desarrollo de la rama estudiantes del IEEE de la UNED. En este nuevo año se ha dado comienzo a una nueva etapa con la nueva junta, en la que una de las primeras cosas que hemos tratado ha sido refrescar un poco la imagen del Boletín, así como la creación de una nueva sección de colaboraciones externas dentro del Boletín en la cual se pueden ver artículos escritos por gente que no pertenece a la rama, pero pertenece a nuestro entorno laboral o universitario. Desde la creación en el año 2004, el número de miembros no ha dejado de crecer, y esperemos que para la próxima campaña se pueda consolidar aún más. Por un lado, se continuará intentando afiliar a nuevos miembros y de esta forma ir creciendo en número y actividades. Intentando atraer a todos aquellos estudiantes, tanto de las carreras de Ingeniería Informática como de Ingeniería Industrial de la UNED, con inquietudes y ganas de formar parte de la rama.

La clave del éxito de la Rama de la UNED, al igual que el resto de Ramas de todo el mundo, es el voluntariado. Por tanto, agradecer a todas las personas que hacen esto posible, y que sin su ayuda no se hubiera podido llevar a cabo. Agradecer a todos los miembros que están en la Rama, que han decidido formar parte de ella y agradecer a todos los voluntarios que han colaborado en las actividades realizándolas y llevándolas a cabo, así como, a todos los autores de los artículos.

Parte de la idea de ofrecer una diversidad cultural diferente entre los estudiantes donde nosotros mismos somos los que dedicamos los esfuerzos voluntarios en pro del bien común en cuanto a conocimientos, contactos y la posibilidad de compartir actividades técnicas, científicas y tecnológicas.

La información general sobre sus actividades e información de cómo hacerse miembro la hemos colocado en la página Web: www.ieec.uned.es/IEEE dentro del enlace de la Rama de Estudiantes.

Las actividades principales que se pretenden realizar son: charlas, cursos, congresos, concursos, actividades educativas, visitas a empresas y organizaciones, interrelación cultural y multidisciplinar y cualquier actividad que quiera desarrollar cada uno de sus miembros.



Para el próximo curso 2007/08, la nueva junta directiva, esta meditando plantear diferentes sugerencias, a los miembros, analizando su aceptación, así como estudiar previamente la viabilidad de éstas mediante los recursos de los que disponemos. Estas son a grandes rasgos las previstas:

A) Utilización de la Plataforma de e-learning aLF como instrumento de comunicación básico entre los componentes de la Rama:

El año pasado comenzamos a estructurar la posibilidad de utilizar la plataforma aLF para que; ante la característica de dispersión de la Universidad en la que estudiamos; nos sirviera de instrumento para acercar a todos los miembros y recabar, no sólo opiniones *on-line* de los mismos, sino también motivar su colaboración.

La experiencia ha sido fructífera hasta el punto de que no sólo hemos utilizado dicho canal para dar información sobre la IEEE en general, sino que hemos ofrecido la posibilidad de seguir *on-line* nuestro primer Taller de robótica virtual a través de este sistema. Todo ello nos ha motivado para profundizar aún más en esa línea de trabajo y de cara a este año académico tenemos previsto continuar con nuestro Taller de robótica *on-line*; profundizando en la línea de trabajo ya iniciada.

En este momento se ha habilitado un Taller de Control con la idea de colaborar con el ya citado Taller de robótica. Los objetivos son:

A.1) Mantener el citado Taller virtual de robótica y complementarlo con talleres presenciales, al menos uno en Madrid, y si existiera la posibilidad otro en A Coruña.

A.2) Iniciar las pruebas, prácticas y contactos necesarios para que nuestra Rama se vaya preparando para participar en uno de los concursos de robótica de competición que se celebran a nivel nacional. De hecho contamos con al menos 3 miembros que ya están trabajando activamente en este tema.

A.3) Iniciar un proyecto conjunto de robótica y control tanto a nivel virtual como presencial, tutorizado por alguno de los catedráticos que habitualmente colaboran con nosotros, para el diseño (sección virtual) y construcción (sección presencial) del hardware y software de un brazo robótico industrial.

A.4) Traducción de cursos del MIT OCW sobre control, que puedan ayudar a la comprensión de hispano-hablantes de dicha materia.

A.5) Desarrollo de una plataforma propia de simulación de procesos, se parte de una base desarrollada, Simucontrol.

A.6) Estudio de los sistemas híbridos, desarrollo de modelos y aplicaciones.

B) Cursos de formación en diferentes áreas en las que consideramos que tenemos y podemos mejorar nuestros conocimientos de cara al mercado laboral:

Hemos estado evaluando las posibles deficiencias formativas que pueden retrasar una adaptación rápida al mercado laboral una vez finalizada nuestra carrera, y este año queremos ofrecer a nuestros miembros la posibilidad de subsanarlas. Algunas de nuestras prioridades son las siguientes:

B.1) Potenciar nuestra capacidad de entender, expresarnos y escribir en inglés. Entendemos que es necesario no sólo tener un vocabulario fluido de cara a establecer contacto con ramas de fuera de nuestro país, sino que es algo fundamental saber manejar un vocabulario técnico para poder preparar alguna de las conferencias que se exponen en los diferentes congresos internacionales que se celebran cada año.

B.2) Iniciar un curso de Java que complemente nuestros conocimientos sobre este método de programación. Consideramos que como plataforma multisistema es necesario conocerla por su popularidad dentro de la actividad empresarial en cualquier ámbito de la ingeniería.

B.3) Creación de un curso a nivel usuario y administrador de aLF. De esta forma que todo aquel interesado podrá aprender a manejar esta plataforma.

B.4) Impartición de un curso de Instalación y Montaje de Redes inalámbricas (wi-fi).

C) Promocionar nuestra Rama para conseguir captar a nuevos miembros

Creemos que los esfuerzos que realizamos, tanto en actividades como en formación, no sirven de nada si los estudiantes de nuestra Universidad no los conocen y no pueden aprovecharse de ellas, por eso tenemos dos objetivos complementarios y de igual importancia estratégica:

C.1) Potenciar la sección WIE de nuestra Rama. Creemos que hay muchos estudiantes que aún desconocen la existencia de la citada sección o aunque la conozcan no saben que servicios les puede ofrecer. Nuestro propósito es empezar a promocionar el WIE desde el momento en que se explica a los estudiantes interesados en convertirse en miembros en que consiste la IEEE, y de esa forma promocionar en femenino nuestra Rama.

C.2) Dar a conocer lo que somos, qué hacemos, qué ofrecemos, y para qué puede servir formar parte de la Rama de Estudiantes de IEEE-UNED. Para ello nuestras acciones consisten en realizar las siguientes actividades de promoción:

- Presentaciones en la diferentes sedes de la UNED en que tengamos cierta representación de miembros; en concreto se realizarán en Madrid y en A Coruña, en esta última ya hay fechas concertadas en dos semanas de Octubre. La primera el día 24 de Octubre y la segunda el 31 de Octubre en la sede del centro Asociado de UNED en A Coruña.



- Establecer una adecuada red de información a través de las “news” de la UNED, dando cumplida cuenta de nuestras actividades y promociones para que nos conozcan.
- Publicidad a través de cartelería y de la realización de actos relacionados con nuestras actividades.

Uno de los objetivos que se planteó para este año y que ya es objetivo cumplido es el lanzamiento de la plataforma aLF, donde se ha incluido un foro para que los miembros de la rama podamos estar en contacto de una forma mucho más sencilla. Aquí también se irán publicando todos los cursos y actividades que se vayan organizando en la rama. Todos lo que todavía no lo halláis hecho debéis pulsar el siguiente link y solicitar el alta en la comunidad de la rama IEEE.

<https://www.innova.uned.es/register/?return%5furl=http%3a%2f%2fwww%2einnova%2euned%2ees%2fdotln%2f>

También, queremos informaros que ya se ha abierto el Call for Papers de la III Edición del congreso ONLINE del Observatorio para la CiberSociedad, cuyo objeto es debatir y reflexionar activamente en torno a las diversas temáticas relacionadas con las dimensiones sociales y culturales de las Tecnologías de la Información y la Comunicación.

<http://www.cibersociedad.net/congres2006/comuns/index.php?llengua=es>

El Observatorio para la CiberSociedad se define como [Comunidad Digital de Conocimiento](#) sobre las dimensiones sociales y culturales de las Tecnologías de la Información y la Comunicación. El OCS es un proyecto de carácter abierto, colaborativo y autogestionado, en constante construcción y crecimiento. Para ello, el OCS organiza [actividades](#), recopila [materiales](#), edita una [revista de contenidos específicos](#), difunde [trabajos de investigación](#) y divulgación, pretende ofrecer y construir una [red de investigación](#) y reflexión sobre lo ciberocial y, en definitiva, fomentar el debate y el trabajo sobre la nueva realidad interconectada y digital de nuestra sociedad.

Otro URL que puede ser de interés es sobre la Rama de Zaragoza en la que explican qué es el CNR (Congreso Nacional de Ramas), así como otras informaciones de interés y actualidad sobre nosotros.

<http://ieeesb.cps.unizar.es/cnr06>

Por último, se va a mostrar un mensaje enviado por Rob Reilly, dirigido a los miembros de la rama de estudiantes del IEEE. Éste decía:

“Hello IEEE Education Society Member,

The IEEE Education Society's Chapters Committee has created a Student Activities Committee (SAC). The SAC is a team of approximately ten EdSoc student-members. These student leaders are continuously involved in defining



different aspects of the SAC, under the guidance of an advisory board. One of the SAC's first initiatives is the establishment of a continuing serial student publication.

The Multidisciplinary Engineering Education Magazine (MEEM) is a student publication of the IEEE Education Society Student Activities Committee (EdSocSAC) whose primary objective is to facilitate the publication of students' papers and information relevant to students.

To access the EdSoc's student publication initiative (author information, submission information, etc.) go to:

<http://www.ewh.ieee.org/soc/es/MEE.html>

If you are a student, or have a student, that would like to be involved in this initiative (as reviewers, associate editors, etc.) or you are a student (or have a student) that has created (or can create) an exemplary manuscript, send email to the Editor In Chief, Emmanuel Gonzalez in Manila, Philippines (gonzaleze@dlsu.edu.ph).

Thank you for your support.

Please visit the IEEE Education Society's Web site at:

<http://www.ieee.org/edsoc>

-Rob- reilly@media.mit.edu"

Animaos a colaborar en esta iniciativa internacional de los estudiantes del IEEE.



PROMOCIÓN DE LA DIRECTIVA DE LA RAMA DE ESTUDIANTES IEEE-UNED AÑO 2007-08



Sergio Martín Gutiérrez: Nuevo Presidente de la rama de estudiantes IEEE-UNED. Ingeniero Informático y estudiante de doctorado en el DIEEC. Actualmente trabaja en el Departamento de Electricidad, Electrónica y Automática en proyectos de investigación. smartin@ieec.uned.es



Elio Sancristobal. Vicepresidente de la Rama de Estudiantes del IEEE-UNED. Ingeniero Informático y estudiante de Doctorado en el DIEEC de la ETSII de la UNED. Actualmente trabaja en el CSI de la UNED. En años anteriores ha colaborado con la junta directiva como secretario. elio@ieec.uned.es



Rosario Gil. Secretaria y Tesorera de la Rama de Estudiantes del IEEE-UNED. Coordinadora de Woman In Engineering. Ingeniera de Telecomunicaciones, actualmente trabaja como Becaria de Investigación en el DIEEC de la ETSII de la UNED. rgil@ieec.uned.es



Gloria Murillo. Coordinadora del Comité del Boletín Electrónico. Ingeniero Técnico en Telecomunicaciones, y estudiante de Ingeniería Industrial por la UNED. En estos momentos trabaja en Indra. gmcordero@indra.es



Germán Carro. Coordinador del Comité de Actividades Generales. Estudiante de Ingeniería Técnica en Informática de Sistemas por la UNED. germancf@eresmas.net



Guillermo Lafuente. Ing. téc. en informática de sistemas por la UNED. También estoy haciendo la certificación CCNA de Cisco Systems. quiye1984@hotmail.com



Francisco Javier Magán. Responsable del Comité de Calidad Interna de la rama. Ingeniero Técnico de Telecomunicación, especialidad en Sistemas Electrónicos, por la UPM. En la actualidad está cursando el segundo ciclo de Ingeniería Industrial, intensificación en Electrónica y Automática, y participa en proyectos de investigación en el DIEEC de la ETSII de la UNED. fjmagan@ieee.org





Ángel Iglesias. Responsable del Comité de Socios y Bienvenida. Estudiante de Informática de Gestión por la UNED. aiglesiascela@hotmail.com



Igor Chávez. Coordinador del Grupo de Robótica. Técnico en Electrónica en la National Schools. Estudiante de Ingeniería Electrónica en la Pontificia Universidad Católica del Perú (PUCP). Actualmente alumno de Ingeniería Técnica Industrial especialidad Electrónica Industrial de la UNED. Miembro de la Rama Estudiantil del IEEE-UNED y del Grupo de Robótica del mismo. igorchavez@ieee.org



Alejandro Díaz. Antiguo Coordinador del Boletín Electrónico de la Rama de Estudiantes del IEEE-UNED, a partir de ahora colaborará con el resto de la junta directiva, especialmente con el Comité de Actividades Generales. Ingeniero Industrial por la ETSII de la UNED, y estudiante de doctorado del DIEEC de la escuela. Trabaja en General de Servicios Integrales (Grupo Acciona) en instalaciones eléctricas. adiash@ieee.org



Jose Antonio Camara. Ingeniero técnico industrial, especialidad en electrónica industrial en la universidad de alcala, estudiante de ingeniería electrónica (uah) e ingeniería industrial (uned), he trabajado en diversos sectores, telecomunicación (siemens y actualmente en telefónica), automóvil, aerogeneradores y materiales en un proyecto en una institución alemana.



Manuel Castro. Profesor Consejero de la Rama de Estudiantes del IEEE-UNED. Catedrático de Tecnología Electrónica. Miembro Señor del IEEE y actual presidente del capítulo Español de la IEEE Education Society creada en España. mcastro@ieec.uned.es



Eugenio López. Mentor de la rama de estudiantes IEEE-UNED, y antiguo presidente de la rama de Estudiantes del IEEE-UNED. Ingeniero Industrial por ETSII de la UNED, y estudiante de Doctorado en el DIEEC de la Escuela. Actualmente trabaja en Niedax Kleinhuis. elopez@ieec.uned.es

Patrones software: La gran bola de lodo

Por: Sergio Martín Gutiérrez

Estudiante de Doctorado
Miembro de la Rama de Estudiantes IEEE – UNED
E-mail: sergio.martin@ieee.org



1. Introducción

Cuando uno está realizando prácticas en la Universidad, a menudo comete el error de no pararse demasiado a pensar en cómo diseñar la aplicación. Lo primero que se suele hacer es ponerse a “picar código” y a estructurarlo sobre la marcha. Esto no suele resultar muy grave dentro del ámbito de algunas asignaturas, pero cuando realmente es necesario programar aplicaciones que funcionen y que sean mantenibles la cosa cambia. El inconveniente es que al no tener costumbre de llevar a cabo ciertas “buenas prácticas” en ocasiones el resultado obtenido no es el esperado.

El presente artículo trata de arrojar algo de luz sobre cómo programar y diseñar software de una manera mejor, con la explicación de algunas estructuras y arquitecturas que evitarán que se tenga que “reinventar la rueda” cada vez que hay que realizar una aplicación software. A estas estructuras y arquitecturas se les conoce con el nombre de “Patrones”. En el presente artículo se comenta en detalle el patrón “Modelo-Vista-Controlador”.

Y como normalmente la mejor manera de aprender a hacer una cosa es ver que cosas no se deben hacer, una parte del artículo corresponde a lo que se conoce como “Antipatrones”, que son precisamente aquellos hábitos que todos en alguna ocasión hemos cometido y que se deberían evitar. Para empezar con los antipatrones describiremos los conocidos como “La gran bola de lodo”, “Poltergeist”, “Código espagueti” y “Reinventar la rueda”.

2. Patrones

2.1. Qué son los patrones

Hay patrones que describen soluciones para todo, desde el análisis hasta el diseño y desde la arquitectura hasta la implementación. Además, los patrones existen en diversas áreas de interés y tecnologías. Por ejemplo mostramos algunos:

- **Patrones organizativos:** Describen la estructura y prácticas de las organizaciones humanas, especialmente las productoras de software.
- **Patrones de análisis:** Describen un conjunto de prácticas destinadas a elaborar modelos de los conceptos principales de la aplicación que se va a construir. La intención principal de estos patrones es ayudar a las personas que realizan el trabajo de modelado, pues no siempre tienen experiencia al respecto y, en la mayoría de los casos, construyen sus modelos sin referencia alguna.
- **Patrones de arquitectura:** Expresan un paradigma fundamental para estructurar u organizar un sistema software. Proporcionan un conjunto de subsistemas o módulos predefinidos, con reglas y guías para organizar las relaciones entre ellos. Ejemplo:
 - Capas (Layers)
 - Aplicaciones: JVM, API, Windows NI
 - Pipes and Filters
 - Aplicaciones: UNIX
 - Pizarrón (Blackboard)
 - Aplicaciones: Hearsay, Inteligencia Artificial
- **Patrones de diseño:** Proporciona un esquema para refinar los subsistemas o componentes de un sistema software y las relaciones entre ellos. Describe estructuras recurrentes de comunicar componentes que resuelven un problema de diseño en un contexto particular. Son patrones de un nivel de abstracción menor que los patrones de arquitectura. Están por lo tanto más próximos a lo que sería el código fuente final. Su uso no se refleja en la estructura global del sistema.
- **Patrones de programación (Idioms patterns):** Un idioma es un patrón de bajo nivel, específico a un lenguaje de programación determinado. Describe como implementar aspectos particulares de los componentes de un patrón de diseño usando las características y potencialidades de un lenguaje de programación concreto.

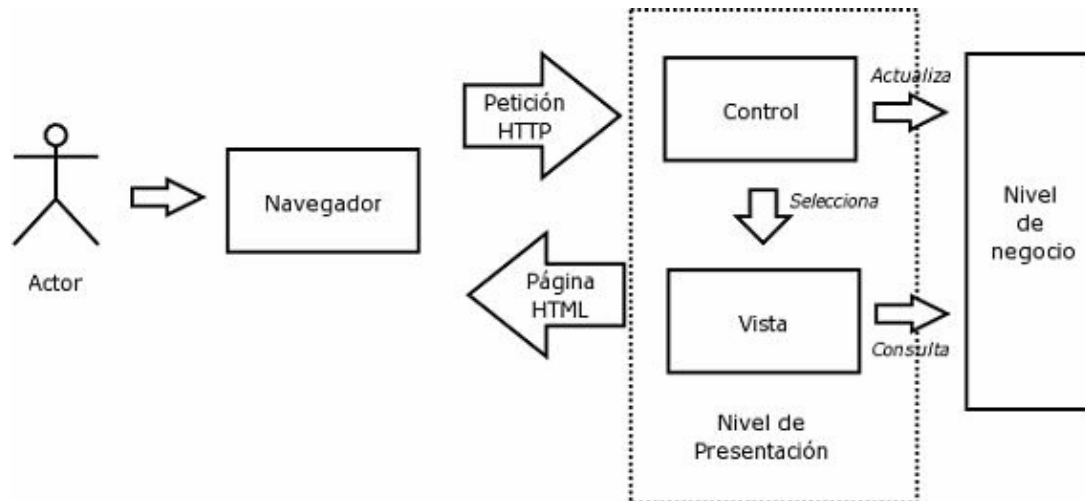
2.2 Modelo Vista Controlador

Modelo Vista Controlador (MVC) es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones Web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página, el modelo es el Sistema de Gestión de Base de Datos y el controlador representa la Lógica de negocio.

- **Modelo:** Esta es la representación específica de la información con la cual el sistema opera. La lógica de datos asegura la integridad de estos y permite derivar nuevos datos; por ejemplo, no permitiendo comprar un

número de unidades negativo, calculando si hoy es el cumpleaños del usuario o los totales, impuestos o portes en un carrito de la compra.

- **Vista:** Este presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario.
- **Controlador:** Este responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista.



Muchos sistemas informáticos utilizan un Sistema de Gestión de Base de Datos para gestionar los datos. En MVC corresponde al modelo. Aunque se pueden encontrar diferentes implementaciones de MVC, el flujo que sigue el control generalmente es el siguiente:

1. El usuario interactúa con la interfaz de usuario de alguna forma (por ejemplo, el usuario pulsa un botón, enlace)
2. El controlador recibe (por parte de los objetos de la interfaz-vista) la notificación de la acción solicitada por el usuario. El controlador gestiona el evento que llega, frecuentemente a través de un gestor de eventos (*handler*) o *callback*.
3. El controlador accede al modelo, actualizándolo, posiblemente modificándolo de forma adecuada a la acción solicitada por el usuario (por ejemplo, el controlador actualiza el carro de la compra del usuario). Los controladores complejos están a menudo estructurados usando un patrón de comando que encapsula las acciones y simplifica su extensión.
4. El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se refleja los cambios en el modelo (por ejemplo, produce un listado del contenido del carro de la compra). El modelo no debe tener conocimiento directo sobre la vista. Sin embargo, el patrón de observador puede ser utilizado para proveer cierta indirección entre el modelo y la vista, permitiendo al modelo notificar a los interesados de cualquier cambio. Un objeto vista puede registrarse con el modelo y esperar a los cambios, pero aun así el modelo en sí mismo sigue sin saber nada de la vista. El controlador no pasa objetos de dominio (el modelo) a la vista aunque puede dar la orden a la vista para que se actualice. En algunas implementaciones la

vista no tiene acceso directo al modelo, dejando que el controlador envíe los datos del modelo a la vista.

5. La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente.

3. Antipatrones

3.1. Qué son los antipatrones

Los patrones nos ofrecen una forma de resolver un problema típico, los antipatrones nos enseñan formas de enfrentarse a problemas con consecuencias negativas conocidas.

Los antipatrones se basan en la idea de que puede resultar más fácil detectar a priori fallos en el desarrollo del proyecto que elegir el camino correcto, o lo que es lo mismo, descartar las alternativas incorrectas nos puede ayudar a la elección de la mejor alternativa.

En la mayor parte de las disciplinas también existen malos usos que están más o menos extendidos, y que por lo tanto no aportan soluciones efectivas a la resolución de problemas. Si un patrón es una buena práctica, entonces un antipatrón es una mala práctica. El estudio de los antipatrones propugna la prevención de errores que pudieran incurrir profesionales inexpertos.

3.2. Antipatrón "La Gran bola de lodo"

Una Gran bola de lodo es una selva de código enrevesado, chapucero, caóticamente estructurado, que crece descontroladamente, que se mantiene como unido a base de cuerda y cinta aislante. Este tipo de sistemas presentan signos inconfundibles de crecimiento incontrolado y constantes necesidades de reparación. Elementos lejanos en el sistema comparten información profusamente, incluso hasta el punto de que prácticamente cualquier información importante se trata de manera global o se duplica. La estructura global del sistema puede no haber llegado a estar claramente definida nunca. Si alguna vez lo estuvo, es probable que se haya deteriorado hasta el punto de ser imposible reconocerla. Los programadores con un mínimo respeto por la estructuración huyen de esta clase de cenagales. Sólo a aquéllos a los que la arquitectura les trae sin cuidado y que tal vez se sienten cómodos programando por inercia parches día tras día para los interminables agujeros de estos diques que hacen aguas por todas partes, no les importa trabajar en tales condiciones.

Los sistemas que son grandes bolas de lodo han sido generalmente desarrollados en diferentes etapas por diferentes individuos que trabajaron en sus distintas piezas y partes. La conveniencia y la comodidad juegan un papel importante en su aparición. Es un antipatrón en el que caen habitualmente los

sistemas desarrollados por personas sin formación previa en análisis y diseño de sistemas ni programación.

No obstante, en ocasiones no se condena radicalmente la programación de grandes bolas de lodo, señalando que la predominancia de este antipatrón se debe a que funciona, al menos en un primer momento. La consecuencia, sin embargo, es la pesadilla de mantenimiento en que se convierte el software.

Los programadores a cargo de proyectos con grandes bolas de lodo deben estudiar su comportamiento para entender su función y usar este conocimiento como punto de partida para elaborar el conjunto formal de requisitos que un sistema nuevo y correctamente estructurado debería tener para reemplazar al existente. Cambios de tecnología (de una filosofía cliente-servidor a una basada en Web, de la utilización de ficheros al empleo de una base de datos, etc.) pueden proporcionar un buen motivo para empezar desde cero.

3.3. Antipatrón "Poltergeist"

Emplear objetos cuyo único propósito es pasar la información a terceros objetos. Esta mala práctica se refiere a objetos de un ciclo de vida corto cuya única función suele ser invocar métodos de otros objetos. Esto hace que crezca la dificultad para mantener el sistema. Suelen identificarse por su nombre: "start_", "manager_". La solución para evitarlos es eliminarlos y poner su funcionalidad en la clase a la que invocan

3.4. Antipatrón "Código Espagueti"

Construir sistemas cuya estructura es difícilmente comprensible, especialmente debido a la escasa utilización de estructuras de programación. Este antipatrón se refiere a un código mal estructurado, ausente de estructuras de control etc. Es frecuente cuando usamos GOTOs, programamos a base de interrupciones o excepciones etc. También es frecuentemente denominado "Código canguro" por lo contiguos saltos que hay que hacer en él para seguir el flujo de ejecución.

Cerca del 50% del tiempo de las operaciones de mantenimiento de este tipo de aplicaciones se invierte en descubrir como funcionan. En la programación orientada a objetos, este tipo de código es tiene clases con muchos métodos sin parámetros, aparecen clases "extrañas", variables globales etc.

3.4. Antipatrón "Reinventar la rueda"

Este es un antipatrón obvio; en muchas ocasiones por falta de un estudio de las tecnologías, diseños o posibilidades existentes se pierde mucho dinero y tiempo desarrollando cosas que ya estaban en el mercado o que ya se encontraban disponibles.

4. Conclusión

El autor espera que el presente artículo sirva de ayuda a todos aquellos que en mayor o menor medida tienen que enfrentarse con la programación o el diseño de aplicaciones software. Aunque dentro del ámbito universitario puede no tener mucha importancia si una aplicación es fácil de mantener o no, ya que suelen estar dentro del ámbito de una práctica, en el mercado real sí que adquiere un valor muy significativo. El coste de mantenimiento de una aplicación software mal programada o diseñada puede ser en algunos casos incluso mayor que el coste de desarrollar una nueva. Es por esta razón que el adquirir “buenos hábitos” de programación en las etapas de aprendizaje puede ser crucial en un futuro no muy lejano.

5. Referencias

- Antipatrón ‘Gran bola de lodo’ en Wikipedia. [última visita 01-10-2007]. http://es.wikipedia.org/wiki/Gran_bola_de_lodo
- Antipatrones de diseño. [última visita 01-10-2007]. <http://complejidad.wordpress.com/2007/05/25/anti-patrones-de-diseno/>
- Jará, O.H. “Evolución y Orientaciones de Patrones” [última visita 01-10-2007]. <http://www.monografias.com/trabajos27/evolucion-patrones/evolucion-patrones.shtml>
- Gamma, E. Helm, R. Jonson, R. & Vlissides, J.. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison Wesley, (1995).
- Buschmann, F. Meunier, R. Rohnert, H. Sommerlad, P. Stal, M. *Pattern-Oriented Software Architecture. A System of Patterns*. Wiley, (1996).
- Brown, W. Malveau, R. "Skip" McCormick III, H. Mowbray, T. *Anti Patterns. Refactoring software, Architectures and Projects in Crisis*. Wiley, (1998).
- van Duyn, D. Landay, J. Hong, J. *The Design of Sites: Patterns, Principles, and Processes for Crafting a Customer-Centered Web Experience*. Addison-Wesley, (2002).
- Hohpe, G. Woolf, B. *Enterprise Integration Patterns*. Addison-Wesley. (2003)
- Patrones, Ricardo Davis, 1996. [última visita 03-02-2005]. <http://www.arrakis.es/~devis/patrones.ppt>
- Palos, J.A. Catálogo de Patrones de Diseño J2EE. I, 1999 – 2005. [última visita 01-10-2007]. <http://www.programacion.com/java/tutorial/patrones/>

GALVANIZACIÓN EN CALIENTE

Por: Eugenio López Aldea

Estudiante de Doctorado en Ingeniería Industrial

Miembro de Student Branch IEEE – UNED

E-mail: elopez@ieee.uned.es



La galvanización en caliente es una técnica utilizada en la industria para conseguir resolver el problema de la corrosión en el hierro y acero. Las ventajas de realizar el galvanizado son la protección eficaz de las piezas que en servicio están expuestas a la intemperie o atacadas por cualquier agente que provoca problemas tempranos ante la funcionalidad provocando un deterioro de la instalación. El material empleado para la galvanización es principalmente el zinc. Prácticamente la mitad de la cantidad producida al año es para este fin. Aunque la producción de zinc requiere una alta inversión de capital, la galvanización contribuye también de otro modo a la economía al proteger el acero de la corrosión. Si las piezas no están protegidas, conllevaría un gasto añadido al tener que sustituir la pieza cada cierto tiempo. Si contamos con la necesidad de disponibilidad de las piezas que se han de sustituir podemos encontrar un abundante espectro donde el cambio implica la paralización de algún proceso industrial que conlleva unos gastos todavía mayores.

De este modo, la selección del material a utilizar en una instalación, como por ejemplo, las bandejas portacables, debe ser estudiado cuidadosamente pensando en el servicio y ahorros futuros más que en el coste inicial.

Son muchas las ventajas que tiene realizar el galvanizado en caliente del material, aparte de destacar que no necesita mantenimiento. El coste del galvanizado es inferior a otras soluciones para detener la corrosión, posee una importante resistencia a los golpes y a la abrasión lo que le da mayor fiabilidad y asegura la mantenibilidad del recubrimiento en el transporte de los materiales y su posterior utilización. Las piezas quedan totalmente recubiertas y protegidas al ser un proceso de inmersión en un baño de cinc fundido. La película de zinc que se forma sobre el acero lo protege de dos maneras, protección de barrera y protección galvánica (catódica). Diversas empresas realizan el galvanizado pero hay que asegurarse de la calidad con la que se realiza este proceso. En la **foto** podemos observar un ejemplo de galvanizadora de la empresa Niedax Kleinhuis, empresa líder en Alemania que fabrica todo tipo de material para la gestión del cableado, donde se puede observar cómo realizan el galvanizado de una bandeja de escalera.

El galvanizado en caliente se utilizaba hace más de 100 años para proteger al acero de la corrosión. Goza de muchos años de experiencia en todo el mundo y su eficacia ha sido probada constantemente. Se trata de un proceso industrial que se ha perfeccionado a lo largo de muchos años disfrutando actualmente de un gran control y mecanización lo que implica la reducción de costos unitarios.

Por regla general y dependiendo del ambiente en el que se encuentre, la duración de la protección que proporcionan los recubrimientos galvanizados, es directamente proporcional al espesor de los mismos. La duración se puede estimar tomando los valores medios de pérdidas anuales de espesor que experimenta un recubrimiento. La norma ISO 9223 contiene estos valores para algunos de los ambientes más comunes. Reafirmando la elevada duración de los materiales galvanizados, se hace innecesario el mantenimiento y la sustitución, lo que en poco tiempo hace rentable cualquier inversión, manteniendo el lema: *“hazlo una vez y que valga para siempre”*. No obstante, en alguna ocasión a muy largo plazo y en ambientes muy agresivos, puede ser necesario reacondicionar el recubrimiento. Esto puede lograrse con un coste muy bajo mediante el galvanizado en frío, sin necesidad de un costoso tratamiento de preparación superficial.

La galvanización en caliente es un procedimiento de gran versatilidad. Sirve tanto para la protección toda clase de piezas o artículos de acero así como de productos siderúrgicos como láminas y alambres. Además protegen al acero tanto de la corrosión atmosférica como de la provocada por las aguas o el terreno, como ejemplo se utiliza para tuberías y depósitos de agua así como para la construcción de alcantarillas o túneles.

Finalmente, señalar que una protección de barrera como pueden ser las pinturas, puede conllevar una oxidación si se rompe la continuidad por algún golpe o cualquier acción mecánica. En el caso del galvanizado, esto no ocurriría, pues si en un momento dado sufre un golpe por el transporte, se raya o presenta alguna discontinuidad por cualquier acción mecánica, el zinc adyacente al acero formaría una sal insoluble de zinc en el acero protegiéndose de nuevo ante la corrosión.

Referencias

<http://www.niedax.com/>

<http://galvanizadocaliente.galvanizing.com.ar/>

<http://www.grupocobos.com.mx/metalyzinc/razones1.htm>

<http://www.unctad.org/infocomm/espagnol/zinc/calidad.htm>

<http://www.asimet.cl/galvanizado.htm>



Figura 1. Galvanizadora de la empresa Niedax Kleinhuis.
Imagen cedida por Niedax Kleinhuis

Comunicaciones móviles

Por: José Antonio Cámara Madrid

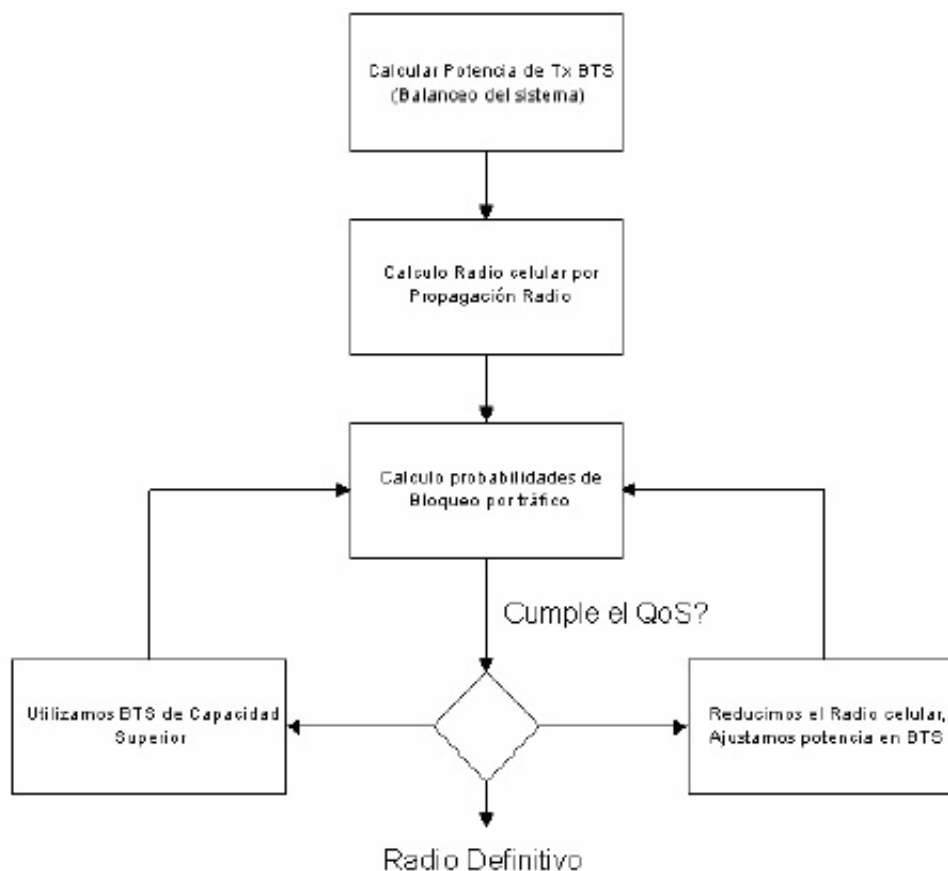


Estudiante de Ingeniería Industrial
Miembro de Student Branch IEEE – UNED
 E-mail: joseacamara@ieee.org

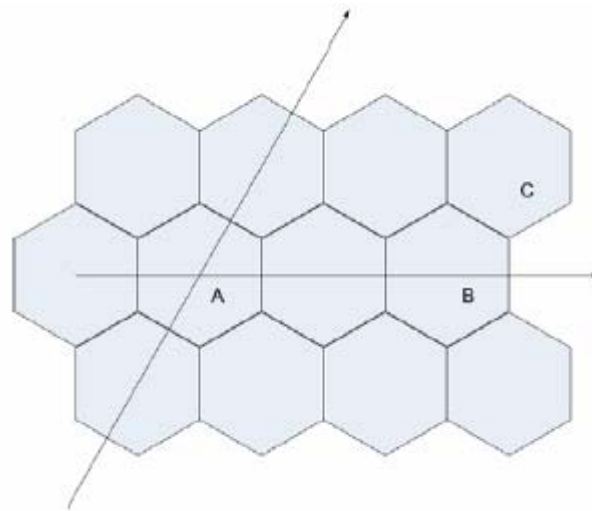
1. Introducción

En el presente artículo se presenta un programa para calcular el radio celular en redes GSM basados en la fórmula de Okumura-Hata y la de Okumura-Hata modificada para 1800-1900 MHz, según el COST 231.

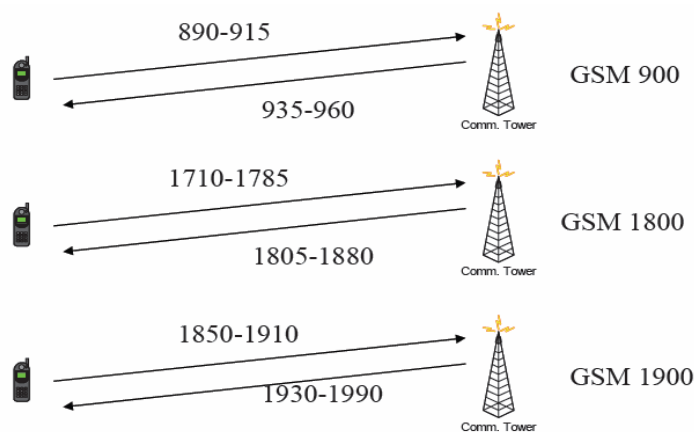
Se realiza el balanceo del sistema y el cálculo del tráfico para seleccionar el radio máximo, que será el menor, ya que es el que nos limita, se puede ver en el siguiente esquema:



Para planificar una estación radio, se divide el territorio en células, para poder distribuir las frecuencias disponibles en cada área, y ver cuándo se pueden reutilizar, dependiendo de los criterios de diseño.



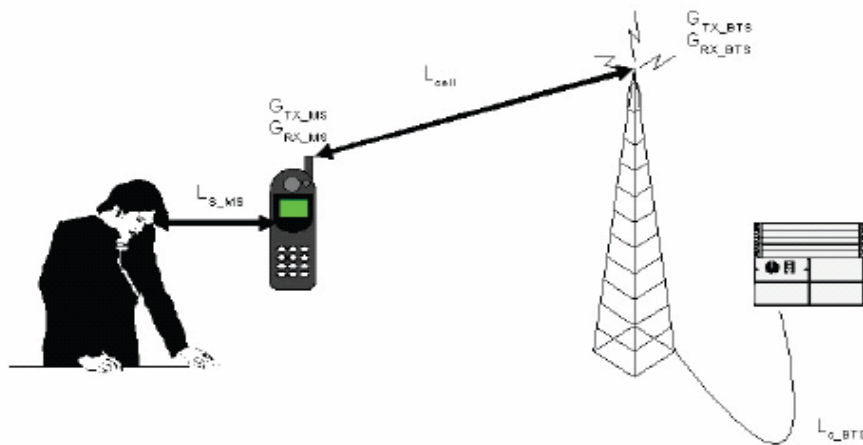
Las frecuencias más utilizadas son 900, 1800 y 1900 MHz, según sea Europa o Estados Unidos, depende bastante del ancho de banda.



2. Desarrollo

La aplicación es muy intuitiva, simplemente hay que introducir valores numéricos en las unidades internacionales en los espacios en blanco, y el programa nos calcula los valores sombreados.

Para el cálculo del radio celular se empieza realizando un balanceo de potencias, que cómo se puede ver en la imagen siguiente, es hacer un sumatorio de las potencias, ganancias y pérdidas del sistema de comunicación.



Planteando ecuaciones:

$$P_{RX_MS} = P_{TX_BTS} - L_{C_BTS} + G_{TX_BTS} - L_{cell} + G_{RX_MS} - L_{S_MS}$$

$$P_{RX_BTS} = P_{TX_MS} - L_{S_MS} + G_{TX_MS} - L_{cell} + G_{RX_BTS} - L_{C_BTS}$$

Para el modelo, las pérdidas se calculan:

$$L_{cell} = F + B \cdot \log(R) - E + G$$

siendo

- Fc : Frecuencia central
- R : Radio de la célula por propagación
- Hb : Altura de la antena de la estación base
- Hm : Altura del móvil

Dónde los parámetros son:

$$F = 69.55 + 26.16 \cdot \log(f_c) - 13.82 \cdot \log(h_b)$$

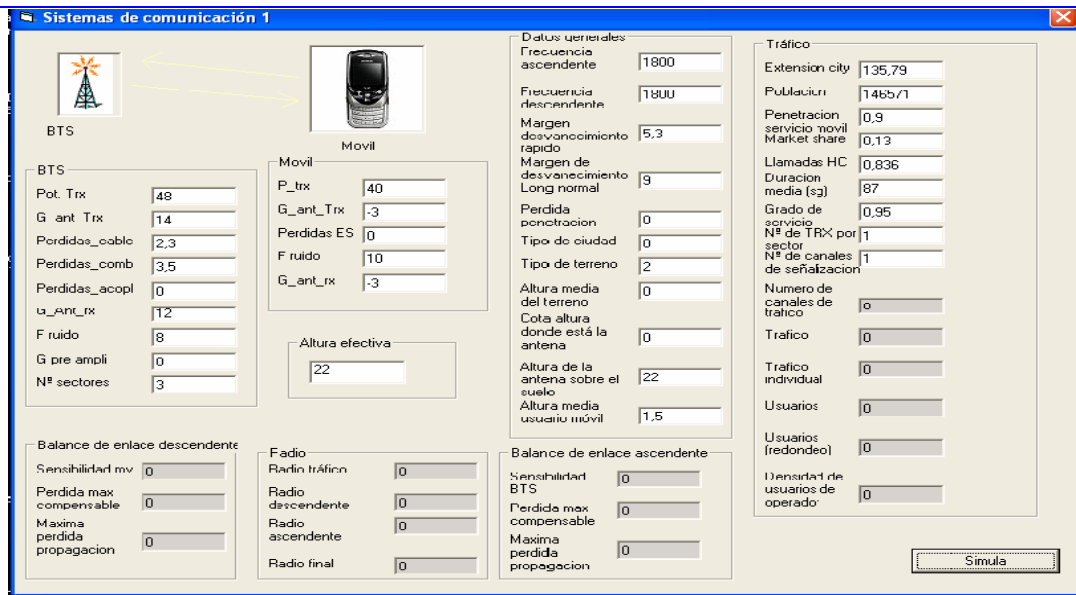
$$B = 44.9 - 6.55 \cdot \log(h_b)$$

$$E = 3.2 \cdot (\log(11.75 \cdot h_m))^2 - 4.97 \text{ Para ciudades grandes con } f_c \geq 300 \text{ MHz}$$

$$E = 8.29 \cdot (\log(1.54 \cdot h_m))^2 - 1.1 \text{ Para ciudades grandes } f_c < 300 \text{ MHz}$$

$$E = (1.1 \cdot \log(f_c) - 0.7) \cdot h_m - (1.56 \cdot \log(f_c) - 0.8) \text{ Para ciudades pequeñas o medianas}$$

$$G = 0$$



El programa de simulación del radio celular, el otro limitante es el tráfico, el modelo es poissoniano con cola limitada, teniendo en cuenta el modelo de velocidad en la ciudad. Se ha desarrollado según expresiones de la teoría de tráfico. Son expresiones que se salen del objetivo global de este texto.

3. Conclusión

Este programa es una aplicación sencilla y que ha sido probada con otras herramientas de simulación, sirve para establecer un radio máximo celular en GSM. Actualmente hay otras tecnologías más complejas, pero sirve de introducción a aquellos que se adentran en el mundo de las comunicaciones inalámbricas.

KARAKURI, EL INICIO DE LOS AUTÓMATAS EN EL PAÍS DEL SOL NACIENTE (y II)

Por: Germán Carro Fernández

*Estudiante de Ingeniería Técnica en Informática de Sistemas
Miembro de Student Branch IEEE – UNED*

E-mail: germancf@eresmas.net



1. Introducción

En nuestro artículo anterior afrontamos un sucinto análisis del nacimiento de los primeros autómatas; precursores de nuestros actuales robots; en los países orientales. Nos centramos en la figura del “karakuri” para ello. No obstante y a pesar del aspecto mecánico de los mismos, estos artilugios también impulsaron de manera incipiente una primitiva “inteligencia artificial” que fue recogida posteriormente por las formulaciones de Turing¹ y su discurso sobre las máquinas de estados. Es precisamente esta vertiente y su implicación hasta nuestros días, la que va a centrar esta segunda y última parte de nuestro artículo sobre tan avezados inventos mecánicos.

En definitiva, lo que se ha ido instaurando a lo largo de la historia de la robótica en oriente es; lo que algunos han calificado como; la filosofía de los karakuri. Una búsqueda de la humanización de las máquinas, cada vez más cercana a la actitud moral y sensitiva del hombre que las ha creado y del entorno que las rodea. Evidentemente esta humanización conlleva la capacidad de desarrollar un “pensamiento” que si bien en un principio era repetitivo y sólo respondía ante una determinada acción, posteriormente se ha ido refinando hasta desarrollar sistemas que imitan actividades cerebrales básicas, o bien diseñar herramientas destinadas al reconocimiento visual, ya sea de objetos, personas o; como la última cámara de fotos de la empresa Sony; gestos, que permitan al robot moderno reaccionar ante ciertos estímulos “naturales”. Es evidente que la tecnología vinculada a la inteligencia artificial no siempre tiene una cara humanoide, pero si es verdad que un aspecto cercano al del ser humano, ayuda a su integración en la sociedad que debe utilizarla, y por ende a su comercialización masiva.

A lo largo de este artículo intentaremos analizar esta evolución de forma que apreciemos; una vez más; que lo que nos parece moderno y mágico, tiene su origen hace varios siglos y ya en ese momento era moderno y mágico también, para las personas que lo desarrollaron. Una vez más veremos que la

innovación es un proceso de perfeccionamiento de una idea antigua pero científicamente revolucionaria.

2. Repetición vs. Inteligencia artificial

En la actualidad se podría decir que; sintetizando de una manera muy superficial y simplista; existen dos corrientes de pensamiento adscritas a la inteligencia artificial:

- *Inteligencia artificial convencional*: Aquella rama basada en el diseño y aplicación de algoritmos estructurados.
- *Inteligencia artificial computacional o simbólica*: Basada en un aprendizaje iterativo que está relacionado de manera directa con las experiencias prácticas que influyen y transforman los algoritmos de base que sirven para activar la máquina que los acoge. Esta última intenta imitar sistemas conexionistas similares a los utilizados por el cerebro de un ser humano al aplicar lo que denominamos trinomio inteligencia-aprendizaje-recuerdo.

Cuando en la primera parte de este artículo describíamos y hablábamos de los karakuri, y citábamos alguna de sus peculiaridades mecánicas, vimos que su respuesta por decirlo de alguna manera; estaba ligada a un proceso de causa-efecto. Es decir, a partir de una causa (colocación de una taza de té en la bandeja del “*Chahakobi Ningyo*”; o “Muñeco para servir el té”²) se producía un efecto “programado” (acercarse hasta el invitado, ofrecerle la taza, esperar a que éste la recogiera de la bandeja, tomara el té y la volviera a depositar vacía, y posteriormente darse la vuelta y regresar a su lugar de origen). Éste comportamiento está completamente controlado a través de la sucesión de varios estados (esperar, avanzar, girar, regresar) motivados por diferentes acontecimientos (colocar la taza, recogerla), y se va a producir esa misma sucesión, en esa misma secuencia hasta el punto de que si alguno de los pasos no se cumpliera, nuestro autómatas permanecería en un estado de espera ilimitado hasta que se diera el suceso necesario para que saliera de él y pudiera continuar el ciclo para el que esta programado finalizando así el mismo.

Este tipo de comportamiento podría englobarse perfectamente en el primer tipo de inteligencia artificial que citamos al principio de este punto. De alguna forma podemos apreciar que; como afirmaba Turing; no es necesario un soporte eléctrico para sustentar un proceso intelectual. En el caso de los karakuri es simplemente una acumulación de engranajes de madera o metal debidamente articulados, los que conforman el algoritmo que va a demostrar esa incipiente inteligencia.

Esta programación fue poco a poco complicándose a lo largo de la historia, diseño y creación de los diferentes karakuri o autómatas que se fueron desarrollando posteriormente. Respuestas como las ya vistas en nuestro anterior artículo del *Yumihiki Doji (Muñeco Arquero)* que estaba diseñado para fallar uno de cada diez disparos a propósito, demuestran el afán de los diseñadores por imbuir características humanas (como puede ser el error o la capacidad de equivocarse) a estos compañeros mecánicos.

En cualquier caso y a pesar de estas aplicaciones no podemos afirmar; evidentemente; que existía una inteligencia artificial de tipo computacional o basada en el aprendizaje. Los primeros autómatas no eran capaces de reconstruir nuevos algoritmos y reaccionar a raíz de la elaboración de nuevas conclusiones. Cualquiera de ellos actuaba de manera predeterminada desarrollando las mismas respuestas a los mismos efectos. No obstante y bajo mi punto de vista dichas respuestas se encontrarían más cerca de esa primitiva inteligencia artificial convencional que definíamos al principio de este apartado, que de la mera repetición de un movimiento constante y sin sentido. De ahí que podamos considerar a los karakuri los precursores de esa ansia centenaria que a los hombres les ha llevado a la búsqueda de la imitación de su capacidad intelectual en otros seres inanimados, dotando de expresividad y sentimientos a sus obras a través de la pintura o la escultura o desarrollando artilugios como los aquí tratados para un uso lúdico o religioso. En definitiva la búsqueda de la humanización de su entorno.

3. ¿Inteligencia artificial es sinónimo de humanidad?

Partiendo de la llamada “Prueba de Turing”; si una máquina se comporta a todos los efectos como inteligente, entonces debe ser inteligente. Aunque esto ocurriera, no podemos dar por sentado que dicha máquina a partir de ese momento se humanizara, ya que ese término engloba antropológicamente conceptos más amplios que la simple inteligencia. No obstante es evidente que a lo largo de la historia de la robótica ha habido una creciente necesidad de intentar humanizar a todos los autómatas que han ido apareciendo a través de los siglos. No debemos olvidar la primitiva expresión que se utilizaba con algunos computadores en sus primeros años de vida al referirse a ellas como “cerebros electrónicos”, o incluso la referencia a algo tan implicado en el mundo de la algoritmia como son los “lenguajes de programación”. Pero dónde mejor se aprecia esta necesidad de vernos reflejados en el desarrollo de una inteligencia artificial creada de la materia inerte, es precisamente en el diseño de los karakuri que hemos visto aquí. Su parecido o imitación de la figura humana no deja lugar a dudas. Y es precisamente en esta vinculación de la tecnología a la sociedad en el mundo oriental, lo que hace que en sus diseños siempre esté presente el parecido con el ser humano o con un entorno familiar a él. Como decíamos en nuestro artículo anterior, esto contrasta de frente con el aspecto utilitarista y meramente material del que Occidente hace gala a la hora de pensar en la robótica tradicional.

En el año 2005 la empresa japonesa **Kokoro Company Ltd.** Lanzó al mercado un androide nacido precisamente de las experiencias previas con los karakuri. El citado androide llevaba como nombre Actroid-DER y estaba dedicado en exclusiva a los museos. Concretamente su labor era la de hacer las veces de cicerone, o guía, en los museos en que trabajase. Para ello se le dotó de un software diseñado para recopilar información a través de una base de datos actualizable, y un rudimentario proceso de aprendizaje interactivo. Pero no quedó todo ahí. Se le dotó de una imagen lo más humana posible, y unos gestos que; aunque primitivos; denotaban también un comportamiento humano. En la foto inferior podéis apreciar perfectamente a este primer prototipo comercial.



Actroid-DER

Pero este éxito se gestó a raíz de los primeros modelos básicos diseñados en el 2003. Y ya desde antes; concretamente desde 1984; esta empresa comenzó a especializarse en el diseño de animatronics para parques, ferias, museos o exposiciones, y posteriormente de humanoides para trabajos relacionados con la información. Lo curioso de todo esto es que los mismos responsables de la empresa reconocen que su mayor dificultad y esfuerzo está precisamente en hacer creíbles a sus creaciones, en darles un hálito de inteligencia viva que permita que su interacción con los seres humanos sea lo más natural posible, en definitiva, en intentar humanizarlas.

El último paso en este sentido ha sido el diseño de un segundo modelo denominado Actroid-DER2 al que se le ha añadido una completa gama de expresiones faciales que harán aún más creíble su capacidad humana para la comunicación.



Actroid-DER2

Este último modelo fue presentado en el año 2006. Esta empresa trabaja, actualmente, en colaboración con la Universidad de Osaka.

Pero no sólo en Japón se dedican a humanizar este tipo de artilugios. Esta otra imagen que representa al profesor Zou Renti junto a su autómata “gemelo” en la exposición que la IEEE organizó en Beijing, China en el 2006 sobre robótica e inteligencia artificial, pone de manifiesto que todos los países orientales trabajan a fondo para instaurar la humanización de sus creaciones a la par que adaptar la capacidad intelectual de las mismas a las del hombre.



Prof. Zou Renti (a la izquierda) junto a su autómata “gemelo.”

Es fácil entender la razón de este mimetismo. El ser humano siempre tenderá a apreciar como más “inteligente” aquello que tenga un mayor parecido consigo mismo, aunque a nivel científico esto no tenga ninguna base, a nivel social es una prueba evidente del rechazo que provoca en el hombre enfrentarse a algo que desconoce o que es distinto a él, y eso; como adelantábamos al inicio de este artículo; no vende. Así que lo mejor es intentar aprovechar la vinculación inteligencia-ser humano; o mejor aún, inteligencia-aspecto humano; para garantizar una buena aceptación del producto de cara al consumidor final.

Conclusión, inteligencia no es sinónimo de humanidad, pero en el mercado internacional “humanidad” si implica inteligencia; aunque esta apenas exista. Con lo que si la aderezamos con una favorable imitación de emociones estudiadas y una base de datos con cierta capacidad de memoria interactiva, tendremos el paso necesario para transformar nuestros iniciales karakuri con una inteligencia artificial convencional, en los modernos androides con una básica inteligencia artificial simbólica, pero mucho más parecidos a los humanos que los primeros.

4. Los Adan y Eva de la Inteligencia Artificial

En esta línea y profundizando en el diseño de una inteligencia lo más parecida al del ser humano, el KIST (*Korea Institue of Science and Tecnology*) está desarrollando; desde el año 2005; lo que han venido en llamar Mahru y Ahra; algo así como “macho” y “hembra”; dos autómatas cuya característica diferencial se sustenta en que su inteligencia se conforma a través de varias áreas sensoriales:

- Reconocimiento de voz (sentido del oído).
- Reconocimiento de gestos y movimientos (sentido de la vista a través del movimiento)
- Reconocimiento de objetos (sentido de la vista a través de la forma).
- Reconocimiento de obstáculos (sentido del tacto).
- Reconocimiento de rostros o caras (sentido de la vista a través de la asociación de imágenes y la memoria).

La unión de todos estos factores sensoriales conforma una red de datos que les permiten ir desarrollando progresivamente una inteligencia basada en el entorno que los rodea. Es precisamente este uso de elementos sensoriales variados lo que hace que estos prototipos den un paso más hacia la imitación de la inteligencia humana. El funcionamiento de los mismos se basa precisamente en la recogida de los citados datos y el envío de los mismos a un servidor que tras analizar la información recibida y procesarla, les envía las

órdenes necesarias para interactuar con su entorno. Este sistema permitiría agrupar las “sensaciones” recogidas tanto por Mahru como por Ahra, combinando así las mismas y posibilitando la actuación conjunta de ellos si la situación lo requiriera. Un sistema similar al que utilizan muchos insectos como las hormigas o las abejas que, aprovechando esta memoria gregaria, pueden actuar como un único individuo si la situación así lo exige.



*Mahru y Ahra en su presentación en Seul
el 28 de marzo de 2005*

Es precisamente el avance en el desarrollo de estos y otros prototipos lo que hace que cada vez sea más fácil desarrollar nuevas aplicaciones sociales para estos autómatas. Según se incrementa su interacción, reconocimiento y comunicación con el entorno en el que se van a utilizar se abren las posibilidades de uso. Actividades relacionadas con cuidados de niños o ancianos, asistencia en establecimientos de comidas o cocinas, utilidades médicas en áreas de difícil acceso, rescate de víctimas tras catástrofes naturales, guías o cicerones interactivos, o simples juguetes o instrumentos lúdicos, todo vale para buscar una aplicación social a estos artilugios que nacieron de madera, alambre y mucha imaginación y que ahora; cada vez más; comienzan a llegar a nuestras vidas para ayudarnos en las tareas cotidianas.

5. Conclusiones finales

Aunque alguna de las futuras aplicaciones que se han citado en este artículo puedan parecer descabelladas o relacionadas más con la ficción que con la ciencia, lo cierto es que puede que estén más cerca de nosotros de lo que pensamos. El 19 de Abril del 2007 el periódico de tirada nacional *El País*, se hacía eco de una noticia publicada días antes por *The Times*. En ella se explicaba que tanto japoneses como coreanos estaban desarrollando las leyes que permitieran una interacción segura entre humanos y robots. La asistencia a ancianos era; entre otras; la utilidad que había generado el debate. Si bien se estimaba en ese artículo que el volumen de negocio de robots destinados a usos sociales relacionados con la asistencia, rondaba en aquel momento la nada despreciable cifra de diez mil millones de yenes (más de 60 millones de euros), se espera que en los próximos quince años las cifras se disparen de manera astronómica en este sector. El riesgo de posibles problemas de diseño y de análisis a través de implementaciones de inteligencia artificial, fundamentalmente vinculados al desarrollo de nuevas técnicas de reconocimiento formuladas a través de algoritmos basados en la visión artificial

para identificación de rostros y gestos, hace necesario que se tomen ciertas precauciones para evitar accidentes ocasionados por problemas de la técnica. De ahí que aunque en ese momento a muchos el artículo les pareciera sacado directamente de las novelas de Isaac Asimov⁴, en realidad ya es necesario que se empiecen a sentar las bases de un nuevo tipo de artilugios que; desbancando a los antiguos karakuri de las primeras épocas de la robótica; estarán mucho más implicados que ellos en actividades relacionadas con la ayuda o el soporte a los seres humanos.

Es precisamente la búsqueda de una mayor perfección en los sistemas de reconocimiento visual que citábamos anteriormente, lo que ha llevado a Sony a continuar trabajando en esta línea y sacar al mercado su nueva cámara digital, la serie *Ciber Shot T* (modelos DSC-T200 y DSC-T170), cuya utilidad radica en que únicamente se disparará cuando detecte que la persona a la que se está fotografiando está sonriendo. Para ello el sensor de visión que incorpora detecta treinta parámetros diferentes del rostro de la persona que se va a fotografiar analizando puntos como la sonrisa y las arrugas que se generan en la comisura de los labios y que se producen al sonreír. La comparación de estos parámetros con los que tiene almacenados de fábrica hace que reconozca ciertos movimientos como una sonrisa y que en ese momento se dispare el obturador. El proceso dura apenas 0,044 segundos. Quizá tiempo suficiente para que un robot con este sistema modificado perciba que el anciano al que cuida se encuentra en problemas y pueda así activar la alerta necesaria para socorrerlo.

En la novela de Philip K. Dick⁵, “¿Sueñan los androides con ovejas eléctricas?” escrita en 1968 (llevada al cine como “Blade Runner” por Ridley Scott en 1982), se especulaba con la dificultad que en un futuro lejano tendría la identificación de androides y de verdaderos seres humanos, debido precisamente a esa búsqueda de la humanización de la que hemos hablado a lo largo de este artículo. Es evidente que a pesar de la velocidad a la que avanza la ciencia, aún estamos lejos de cometer ese tipo de equivocaciones. Aunque visualmente puede que alguno de los últimos modelos evolucionados de karakuri nos pueda llevar a engaño; caso de la fotografía del profesor Zou Renti y de su robot gemelo; en cuanto nos paramos a ver como se mueve o habla el autómatas en cuestión la confusión desaparece y queda relegada a una mera anécdota. En cualquier caso resulta evidente que los pasos seguidos desde los primeros y automatizados karakuri basados en la inteligencia artificial convencional o en complejos mecanismos de repetición predecibles, hasta los nuevos modelos de autómatas con interacciones en forma de primitivas redes neuronales o receptores de impulsos sensoriales y su nueva y humanizada figura exterior, se ha dado un paso de gigante.

Sin duda alguna los avances continuarán en esta línea, pero no debemos olvidar que han sido esos pequeños artilugios de madera los que, animados por la imaginación de sus creadores, sentaron las bases de nuestro presente y a la vez del futuro que nos depara la robótica.

6. Notas aclaratorias

1.- **Alan Mathison Turing** (23 de junio de 1912- 7 de junio de 1954): Científico, matemático de la informática y la criptografía, y filósofo inglés. Considerado uno de los padres de la computación. Su mayor aportación a la ciencia informática fue su Máquina de Turing, una máquina de estados base de lo que posteriormente sería el desarrollo matemático de los estudios sobre computación e inteligencia artificial.

2.- **“Chahakobi Ningyo”**; o “Muñeco para servir el té”: Ver artículo anterior “Karakuri, el inicio de los autómatas en el país del sol naciente”; del que éste es continuación; en el Boletín nº 7 de 9 de mayo de 2007 editado por la *Student Branch IEEE-UNED*.

3.- **Animatronic**: Es el nombre genérico que reciben todos los autómatas o robots con forma de animal con capacidades individuales para comportarse en el torno como los animales reales en que están basados. Un ejemplo de ellos serían los dinosaurios diseñados para películas o parques de atracciones y otros animales como leones y gorilas entre otros. El término fue acuñado inicialmente por la factoría *Disney* para denominar a los autómatas que utilizaba en sus parques temáticos y posteriormente ha terminado por utilizarse de forma genérica.

4.- **Isaac Asimov** (2 de enero de 1920 – 6 de abril de 1992): Escritor y bioquímico estadounidense nacido en Rusia, exitoso y excepcionalmente prolífico autor de obras de ciencia ficción y divulgación científica.

5.- **Philip Kindred Dick** (16 de diciembre de 1928, Chicago, Illinois, Estados Unidos - 2 de marzo de 1982, Santa Ana, California, Estados Unidos): Más conocido como Philip K. Dick, fue un prolífico escritor y novelista estadounidense de ciencia ficción, que influyó notablemente en dicho género.

7. Bibliografía e imágenes

De estas páginas se ha sacado la mayor parte de la información necesaria para escribir este artículo, así como las fotografías que lo ilustran. Algunas de ellas están en japonés, por lo que para su correcta visualización se requerirá instalar las fuentes correspondientes en el navegador que se utilice para ello.

<http://www.karakuri.info/shobei/index.html>

<http://www.kokoro-dreams.co.jp/english/index.html>

<http://www.androidworld.com/prod01es.htm>

http://english.people.com.cn/200610/12/eng20061012_311174.html

<http://www.kist.re.kr/index.jsp>

<http://humanoid.kist.re.kr/eng/index.php>

<http://www.ittimes.co.kr/en/node.asp?em=M&mcode=200504>

http://www.elpais.com/articulo/tecnologia/Japon/Corea/Sur/preparan/leyes/regul ar/conducta/robots/elpeputec/20070419elpcbtec_1/Tes

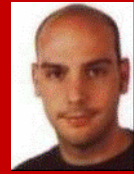
<http://www.lavanguardia.es/lv24h/20070910/53392795980.html>

<http://www.sony.jp/products/Consumer/DSC/DSC-T200/>

Clusters de servidores

Por: Guillermo F. Lafuente Tejero

Estudiante de Ingeniería Informática
Miembro de Student Branch IEEE – UNED
E-mail: guiye1984@hotmail.com



1. Introducción

El ansia de obtener más potencia de cálculo ha sido una de las principales causas del gran desarrollo de los ordenadores en los últimos años. Antes del amanecer de la era de las computadoras, científicos e ingenieros pensaban que realizar cientos de operaciones aritméticas por segundo, era algo inimaginable. Sin embargo, tan pronto como se encontraron con ese enorme poder computacional, empezaron a pedir más. Quisieron ser capaces de realizar miles de operaciones por segundo. Después de miles, fueron millones, después billones, y el actual objetivo son trillones (aunque muchos estén hablando ya de cuatrillones de operaciones por segundo). Para lograr llegar a estos objetivos se han desarrollado varias tecnologías que consiguen dichos propósitos. Dos ejemplos:

- El estudio de los sistemas paralelos, y de las aplicaciones paralelas desarrolladas sobre ellos, es un punto de alto intereses profesional. Actualmente tienen un alto grado de implantación en todo tipo de infraestructuras profesionales donde existan bases de datos, servidores Web, servidores de ficheros, etc.
- Los clusteres de servidores, de los que hablaré durante el artículo y que además emplean conceptos del ejemplo anterior.

Para terminar esta breve introducción, voy a exponer la definición de cluster de servidor:

Según la Wikipedia el término **cluster** se aplica a los conjuntos o conglomerados de computadoras construidos mediante la utilización de componentes de hardware comunes y que se comportan como si fuesen una única computadora. Juegan hoy en día un papel importante en la solución de problemas de las ciencias, las ingenierías y del comercio moderno. La tecnología de clusters ha evolucionado en apoyo de actividades que van desde aplicaciones de súper cómputo y software de misiones críticas, servidores Web y comercio electrónico, hasta bases de datos de alto rendimiento, entre otros usos.

Según Microsoft un cluster es un grupo de equipos que trabajan de manera conjunta para ejecutar un grupo de aplicaciones comunes y ofrecer la imagen de un único sistema al cliente y a la aplicación. Los equipos están conectados físicamente mediante cables y programáticamente mediante software para clusteres. Estas conexiones permiten que los equipos utilicen la conmutación por error y el equilibrio de carga, que no resultan posibles en equipos individuales.

2. Definición, tipos y componentes de un clúster

¿Qué es un Cluster?

Simplemente, **cluster** es un grupo de múltiples ordenadores unidos mediante una red de alta velocidad, de tal forma que el conjunto es visto como un único ordenador, más potente que los comunes de escritorio.

Tipos de clusters:

- Cluster con equilibrio de carga de red.

Proporciona soporte de escalabilidad y disponibilidad para aplicaciones y servicios basados en IP

Da soporte al equilibrio de carga del tráfico TCP, UDP, y GRE

- Cluster de cómputo.

Divide tareas complejas en partes más pequeñas para obtener una ejecución paralela y agregar los resultados

Las aplicaciones se deben escribir para explotar explícitamente el paralelismo

- Cluster de servidor.

Una plataforma altamente disponible para aplicaciones de host tales como SQL Server y Exchange Server

Es útil para las aplicaciones que deben mantener un estado entre solicitudes

Componentes de un cluster:

A nivel Hardware:

Se suelen utilizar ordenadores servidores de alta-disponibilidad que son los que se denominan nodos del cluster, los cuales se dimensionaran según las necesidades de nuestra organización y el tipo de cluster.

Una posible composición de un cluster es la siguiente:

Utilizaremos dos servidores conectados por fibra óptica a un cajón de disco. Dispondremos de una Red LAN externa la cual dará servicio a todos los clientes, una Red LAN intra-cluster privada (se aconseja hardware de red independiente para las dos redes) para la comunicación entre los nodos del cluster, y una Red SAN de almacenamiento compuesta por un cajón de discos externos de alta capacidad y en Raid 5, todo esto conectado por Fibra Óptica y switch de Fibra.

A nivel Software:

El sistema operativo establecido nos ofrecerá la opción de instalar y montar los servicios de cluster, como ejemplo Windows Server 2003 o software aparte como veritas cluster.

3. Ventajas dentro de la organización

Un cluster es un conjunto de ordenadores conectados por red que trabajan juntos para resolver un problema común. Estos sistemas paralelos cada vez tienen mayor relevancia y comienzan a desplazar a las potentes estaciones de trabajo, tanto en empresas, universidades, instituciones científicas, etc. Son numerosas las ventajas que presenta un cluster respecto a un mainframe:

Económica con el mismo precio que cuesta una estación de trabajo, se pueden comprar decenas de ordenadores personales bastante potentes que juntos resulten muchísimo más potentes.

Velocidad Alcanzan mayor nivel de computo que las estaciones de trabajo gracias al gran número de ordenadores que forman el cluster.

Escalabilidad Un sistema de ordenadores en red es fácilmente escalable, simplemente hay que añadir una nueva CPU.

Fiabilidad, Alta Disponibilidad si una CPU falla, el sistema sigue funcionando. Un cluster está diseñado para evitar un único punto de error. Las aplicaciones pueden distribuirse en más de un equipo, consiguiendo un grado de paralelismo y recuperación de errores y proporcionando más disponibilidad.

Capacidad de administración El cluster aparece como la imagen de un único sistema para los usuarios finales, las aplicaciones y la red, proporcionando a la vez un único punto de control para los administradores, local o remotamente si una CPU falla, el sistema sigue funcionando.

Balanceo de carga: Al usar múltiples equipos con procesamiento paralelo tenemos la posibilidad de balancear la carga de trabajo, de forma que se distribuya de manera equitativa, evitando así posibles cuellos de botella.

4. Problemas para la administración

Antes de evaluar los problemas de su administración, procederemos a entender las ventajas que aporta el uso de un cluster de servidores desde el punto de vista de un administrador, las cuales son las siguientes:

- Disponibilidad mejorada.
- Permiten que los servicios y aplicaciones sigan operando después de un fallo del hardware o software, o durante el mantenimiento.
- Mayor escalabilidad.
- Brindan soporte a servidores que se pueden ampliar.
- Procesadores adicionales: la potencia de los procesadores será la suma de todos los servidores que formen parte del cluster, con la ventaja añadida de la potencia de emplear procesamiento paralelo.
- Memoria adicional.
- Capacidad de administración mejorada.
- Permiten que los administradores manejen los dispositivos y recursos de cluster de la misma forma en que manejan un solo PC.

Una vez hemos evaluado las ventajas, podremos definir los inconvenientes que nos encontraremos por el uso de esta tecnología:

La operación de *clusters* requiere de un manejo adecuado de los recursos asociados. Los recursos del *cluster* deben ser administrados adecuadamente para que el administrador invierta la menor cantidad de tiempo en detectar, investigar y recuperar fallos de hardware y software, y de este modo definir posibles medidas de contingencia y tratar que el sistema esté libre de errores. A su vez, estos pasos permiten la adaptabilidad a los requerimientos y cambios constantes que se presentan en la manipulación de tecnologías *cluster*.

El administrador de un *cluster* debe tomar en cuenta algunos aspectos. Estos aspectos incluyen la configuración e instalación de un sistema de archivos universal, la configuración y administración de recursos mediante herramientas implementadas en software; el monitorizar sus actividades y el registro de cada uno de los eventos generados por la ejecución de cálculos computacionales.

Varios de los sistemas más importantes para la instalación automática de clusters, incluyen herramientas de monitorización, administración y registro de eventos mediante paquetes de distribución para sistemas Windows y Linux. Estos sistemas facilitan enormemente la administración al incluir software con propósitos específicos como:

- Definición y administración de nodos.
- Administración de colas por lotes (*Batch Queue Management*).
- Administración de recursos: grupos NIS (Network Information Service), cuotas de disco y CPU.
- Administración de servicios de resolución de nombres: DNS (*Domain Name System* para *clusters*).
- Registro de usuarios para *clusters* de dimensiones superiores a los 100 nodos.
- Monitorización de carga.

La administración de *clusters*, implica tomar medidas preventivas y planificar tareas. La administración implica los siguientes aspectos:

- Registro de eventos.
- Monitorización o medida del estado de los recursos del *cluster*.
- Recuperación ante fallos de hardware, software, incluyendo el sistema de archivos.
- Administración del registro de usuarios y grupos de usuarios, de los servicios del *cluster* (*accounting*).
- Planificación de tareas y balanceo de carga.

A continuación se explica con más detalle los anteriores puntos sobre la administración de un cluster.

Registro de eventos:

El registro de eventos supone la manipulación de los ficheros log del sistema operativo para registrar la actividad que se produce sobre la red y, en caso de ataque, poder recurrir a ellos para averiguar qué ha pasado en nuestro sistema.

Monitorización o medida del estado de los recursos del cluster.

La monitorización permite conocer si todos los componentes de hardware y software están disponibles, operando de acuerdo a lo esperado. Ejemplos de herramientas de monitorización serían: Big brother, cluemon, ganglio, etc.

Recuperación ante fallos:

La recuperación ante fallos es una de las cosas más importantes a la hora de considerar la administración de un cluster, ya que en él seguramente tendremos información crítica de la empresa.

La administración del *cluster* implica resolver problemas provocados por fallos de hardware y/o software. Los fallos causados por hardware pueden ocasionar que el *cluster* quede inutilizable.

La recuperación ante fallos a nivel de hardware implica:

1. Aislar los componentes que fallaron para asegurar que no causen un considerable impacto en las actividades del *cluster*.
2. Manejar las copias de seguridad (*backup*) que se realicen sobre el cluster, para poder hacer reemplazos y minimizar los efectos del fallo.

Los fallos de componentes de software muchas veces no tienen solución o forma de recuperación. Si se considera que el sistema operativo está basado en Linux, la mayor parte de aplicaciones requieren de parches o nuevas versiones para mejorar o recuperarse de errores; sin embargo, este proceso es muy complejo y conlleva mucho tiempo. Por tal motivo, si un componente de software falla lo único que resta por hacer es informar al vendedor, diseñador o desarrollador de la aplicación y esperar por las mejoras.

Accounting:

Los ambientes Linux nos dan la posibilidad de mantener copias de un conjunto de archivos en varios equipos. La forma más común y fácil de hacerlo es la utilización de una red basada en servicios para la administración de cuentas o registros de usuario. Cuando se emplea esta alternativa, cada computador realiza consultas a un servicio central, el cual maneja la autorización, la autenticación y la información de los usuarios dentro del sistema.

Planificación de Tareas y Balanceo de Carga:

Las actividades de administración y balanceo de carga que son críticas para un entorno *cluster* son:

- Administrar la disponibilidad de los nodos.
- Configurar atributos de los nodos que sean importantes para balanceo de carga.
- Administrar usuarios y grupos mediante cuotas de disco.
- Configurar y diseñar políticas.
- Administrar reservaciones y recursos dedicados.
- Monitorizar y generar un historial de utilización de recursos para usuarios y grupos.

5. Problemas de seguridad

Existen una serie de recomendaciones a seguir para poder proteger a nuestro cluster de servidores de ataques externos:

- Limitar el acceso físico al cluster de servidores: sólo el personal autorizado debe tener acceso al lugar elegido para ubicar el cluster.
- Se debe proteger la red en la que opere el cluster: una forma de realizar esto, en el caso de considerar a una organización con una red e infraestructura privada, sería el poner en una red aparte al cluster de forma que sólo se tenga acceso a él a través de la intranet y nunca a partir de tráfico exterior. Esto debe hacerse mediante cortafuegos que regulen el tráfico de nuestra red y refuercen la seguridad ante posibles ataques.
- Se debe evitar en la medida de lo posible la administración remota.
- Limitar los permisos de usuario: se debe de minimizar al máximo los permisos administrativos de forma que ningún usuario pueda causar daños de forma accidental o intencionada.
- Si se tiene más de un cluster, se deben de utilizar cuentas diferentes para cada uno en su administración: si por alguna razón una cuenta deja de ser segura esto no afectará a todos los clusters y, si por ejemplo, un atacante descubre una contraseña, limitaremos sus acciones dañinas a un solo cluster y no a todos los disponibles en la organización.
- En caso de ataque se debe realizar una auditoria de los sucesos ocurridos sobre el cluster: para ello disponemos, por ejemplo, de los ficheros log. Se debe de denunciar el hecho a las autoridades pertinentes.
- Se debe limitar el acceso a los datos compartidos
- Se debe limitar el acceso al disco de quórum y asegurarse de que siempre dispone de espacio libre: para reducir el riesgo de lecturas y escrituras no autorizadas sobre él se recomienda limitar al máximo los permisos de acceso a este recurso.
- Se debe asegurar que las aplicaciones ejecutadas sobre el cluster tienen un origen de confianza y que el acceso a los discos del cluster está limitado sólo a aplicaciones que se administran como recurso de cluster.
- Se deben proteger los archivos de secuencias de comandos.
- Se deben proteger los archivos de registro de la configuración del cluster, especialmente los creados al administrarlo remotamente.

6. Facilidad de integración con sistemas implantados

La implantación de un cluster dentro de una estructura de servidores, es una tarea fácil, porque todas las aplicaciones ven el cluster como una sola máquina, ya que inicialmente tiene una única IP, por la cual se identifica. Todos los nodos que componen el cluster toman dicha IP si se produce una conmutación por error o porque se produzca una conmutación manual. Por este motivo se podría sustituir un servidor de bases de datos en una organización dándole al cluster la misma IP que tenía este y restableciendo las bases de datos en el cluster.

Los clusters de servidores mejoran la escalabilidad del sistema. La escalabilidad es la medida de la capacidad de crecimiento de un servicio o de una aplicación para satisfacer demandas de rendimiento cada vez mayores. Al utilizar la tecnología de clusters de servidores, la escalabilidad que se obtiene nos da la posibilidad de añadir equipos a un cluster de manera incremental, de forma que cuando la carga global de dicho cluster supere su capacidad, se consiga mejorar su rendimiento hasta un nivel adecuado.

7. Programación de aplicaciones sobre clusters

Un cluster puede utilizarse como una máquina multiprocesador de memoria distribuida, para la ejecución de aplicaciones con unos grandes requisitos de cómputo. Esta máquina está formada por un conjunto de nodos, cada uno de los cuales tiene unas capacidades de cómputo, memoria, Entrada/Salida y comunicaciones que en el caso de los sistemas heterogéneos puede variar sustancialmente de unos nodos a otros.

Por otro lado, un programa paralelo está formado por un conjunto de procesos independientes y sus correspondientes conjuntos de datos, que se sincronizan e intercambian información a través del paradigma de paso de mensajes.

Estos procesos en principio son independientes y se pueden ejecutar en cualquiera de los nodos que forman el sistema, consumiendo una parte de los recursos de cómputo de dicho nodo.

Generalmente al realizar la fase de partición obtenemos más tareas que el número de procesadores disponibles para su ejecución. Por lo tanto, a la hora de ejecutar un programa se debe realizar una asignación o distribución de las tareas sobre el conjunto de procesadores; lo mismo ocurre con los datos. Dado que al utilizar una máquina paralela para resolver una aplicación lo que buscamos principalmente es mejorar los tiempos de cómputo, un objetivo principal a la hora de realizar esta asignación, debe ser utilizar todos los recursos del sistema de la mejor forma posible.

Para ello es importante distribuir la carga de trabajo que tenemos (tanto el conjunto de procesos como los datos) entre los nodos disponibles de forma que

el trabajo esté equilibrado. Esta correcta distribución mejora la eficiencia global de la aplicación y reduce el tiempo de ejecución, aprovechando al máximo los recursos disponibles en el sistema. Formalmente el problema del equilibrio de carga en un cluster puede definirse como:

Dado un conjunto de tareas y de datos a procesar, y un conjunto de procesadores en los cuales se puede ejecutar, encontrar una asignación de tareas y datos a los procesadores que consiga que cada procesador tenga aproximadamente la misma cantidad de trabajo.

En los sistemas heterogéneos esto se complica aún más, pues las capacidades de los nodos pueden variar mucho de unos a otros. Por otro lado, en sistemas homogéneos pero no dedicados, las capacidades de los nodos pueden variar mucho a lo largo del tiempo, en función de la carga de trabajo local de cada uno en un instante de tiempo dado.

Hay distintas maneras de resolver el problema de equilibrio de carga en un cluster.

a) Atendiendo a la situación de la información y a la estructura de los procesos se encuentran los algoritmos centralizados frente a los distribuidos.

b) Atendiendo a la toma de decisiones se encuentran los algoritmos estáticos frente a los dinámicos.

De esta manera surgen los siguientes algoritmos:

Centralizados

Son aquellos en los que un nodo central se encarga de todo el peso del algoritmo, es el que tiene toda la información de los demás nodos y el que toma las decisiones pertinentes. Los demás nodos por tanto sólo tienen información de ellos mismos, y solo se comunican con el nodo central. Son como esclavos que realizan el trabajo que les ordena el maestro.

Estos algoritmos tienen la ventaja de su gran sencillez y de su eficiencia con un número reducido de nodos, pero son mayores las desventajas, como la presencia de cuellos de botella en el lado del nodo central, y problemas al tener un límite en su grado de escalabilidad.

Distribuidos

Son aquellos en los que no existe un nodo central, si no que todos los nodos tienen información del estado del cluster, y toman sus propias decisiones dependiendo tanto de su propio estado como del estado del cluster en general.

Estos algoritmos son mas complicados que los centralizados y se producen mayor número de mensajes entre nodos, pero son muchísimo más eficientes

porque eliminan los problemas de escalabilidad y los cuellos de botella al no haber un nodo central que se ocupe de más trabajo que los demás. Otra dificultad añadida es que al tener todos los nodos información del cluster, puede haber problemas de coherencia al estar replicada la información.

Estáticos

Son aquellos en los que se toman decisiones a priori, de forma determinista o probabilística, antes de comenzar a ejecutar la aplicación y sin tener en cuenta el estado actual del sistema. Esta solución puede ser efectiva cuando la carga se puede caracterizar suficientemente bien antes de la ejecución, pero no es realmente eficiente si se producen numerosos cambios en el estado del sistema.

Dinámicos

Son aquellos en los que se utiliza la información del estado del sistema para tomar decisiones, por lo que potencialmente mejoran a los algoritmos estáticos al mejorar la calidad de las decisiones. Sin embargo, produce una pequeña penalización, ya que se produce una sobrecarga al tener que recoger información del estado del sistema en tiempo real.

8. El futuro

Según un estudio de la Unión Europea, los clusters son considerados como prominentes vehículos de innovación y competitividad lo cual es fundamental para el desarrollo y producción económica. Además los clusters se están desarrollando en un ambiente de competitividad superior al promedio. Esto provoca que las empresas que están metidas en el terreno del desarrollo y mejora de los clusters de servidores intenten buscar innovaciones que hagan que sus productos de cluster sean los más potentes y eficientes, con sistemas de seguridad cada vez más sofisticados.

Otro factor a tener en cuenta y, que a la vez es el que ha proporcionado un buen ambiente para la competitividad existente, es la posibilidad computacional de futuro de este tipo de sistemas. Esta tecnología tiene un alto atractivo para las empresas ya que les permite, por un costo inferior al de comprar una nueva máquina más potente, reutilizar los servidores de los que disponen actualmente aumentando el rendimiento del sistema y unificando la distribución de la información, evitando que esta se duplique.

A raíz de la importancia que tiene evitar la duplicidad de la información, existen proyectos en diferentes países y entre diferentes organizaciones, que pretenden formar clusters mediante los cuales compartir información con los siguientes objetivos:

- Mejorar la comunicación
- Evitar duplicidad de la información
- Evitar desperdicio de recursos humanos y financieros.

Los clusters también tienen una alta demanda del sector de la investigación científica. La posibilidad que brindan de hacer operaciones de súper cómputo hace que en el futuro, cálculos que hoy no son imposibles de realizar, puedan ser resueltos en un tiempo razonable de espera o se consiga la posibilidad de hacer simulaciones de mayor tamaño.

9. Bibliografía

Para terminar os dejo bibliografía y enlaces web en donde podréis ampliar la información publicada en este artículo:

Clusters: GREGORY F. PFISTER, In Search of Clusters. Prentice-Hall, 1997.

Loadindices: D. FERRARI y S. ZHOU, An empirical investigation of load indices for load balancing applications. Proceedings of Performance'87, 1987.

Iniciacion-receptor; L. M. NI, C. W. XU y T. B. GENDREAU, A distributed drafting algorithm for load balancing. IEEE Transactions on Computers, 1988.

Migracion: P. KRUEGER y M. LIVNY, A comparison of preemptive and non-preemptive load distribution. Proceedings of International Conference on Distribute Computing Systems, 1998.

Enlaces Web:

http://es.wikipedia.org/wiki/Cluster_de_computadores

<http://technet.microsoft.com/es-es/default.aspx>

<http://www.microsoft.com/technet/prodtechnol/windowsserver2003/es/library/ServerHelp/d577afb5-9f3a-47e8-a744-2546d330cb17.msp?mfr=true>)

<http://www.bs.com.ar/bsweb/RevistaBSknow/PDFs/n17/Exchange.pdf>

<http://clusterfie.epn.edu.ec/clusters/Definiciones/definiciones4.html>

La lengua inglesa y la Ingeniería

Por: María M. García Lorenzo

Doctora en Filología Inglesa

Profesora de la UNED

E-mail: mgarcia@flog.uned.es

Todos hablamos, leemos, y entendemos un poco de inglés. Resulta difícil no manejarse mínimamente, pues la expansión de la lengua inglesa llega a todos los rincones de nuestro día a día: desde títulos de películas que quedan sin traducir hasta la red de redes, que habla un inglés un tanto entrecortado pero sumamente efectivo. La sociedad de la información, globalizada y anglizada, apea a los que no se atreven ni con ese poco de inglés, que se ha convertido en lengua comodín a pesar de no ser la lengua nativa más hablada del mundo. De hecho, los hablantes de inglés como segunda lengua superan en número a los hablantes nativos, y se prevé que la diferencia aumente considerablemente en las próximas décadas. Así las cosas, el inglés ha echado raíces como **idioma internacional de la investigación y el desarrollo, y por tanto del intercambio científico, del comercio y de la comunicación**. No resulta extraño, por tanto, que las publicaciones académicas de mayor prestigio y difusión sean en lengua inglesa.

Resulta fácil comprobar que muchas ofertas de empleo nacionales e internacionales relacionadas con la ingeniería solicitan, además de unos sólidos conocimientos teóricos y prácticos de las materias propias del campo, un buen nivel de inglés. La vida profesional del ingeniero se vuelve progresivamente más internacional y comprende proyectos de colaboración, intercambio de ideas en foros académicos o laborales, e incluso diferentes roles empresariales que hacen del inglés una herramienta casi imprescindible. La globalización, por añadidura, trae consigo la necesidad de conocer otras estructuras culturales, legislaciones, prácticas empresariales y comerciales que tal vez nos sean extrañas, pero a las que accederemos gracias a la lengua inglesa.

“Un poco de inglés” sirve para muchas cosas, pero no para todo. Hay momentos y circunstancias que requieren cierta especialización del instrumento que utilizamos para comunicarnos. El **inglés para fines específicos** (profesionales o académicos) nos permite establecer relaciones dialécticas funcionales y recurrir a términos o expresiones precisas e inequívocas. Las universidades españolas, encargadas de la formación de ingenieros a todos los niveles, no responden a esta seria demanda social de igual modo a la hora de ofertar sus carreras de ingeniería. Hay universidades que retiran o simplemente no ofertan la asignatura por considerarla demasiado gramatical (legado de la

romanización, nada menos), y que por ello no aporta nada a la transversalidad en los planes de estudio. Otras, en cambio, ofertan cursos bilingües (español-inglés) para adecuarse al Espacio Europeo de Educación Superior. En esta misma línea, el Departamento de Ingeniería Eléctrica, Electrónica y de Control de la UNED puso en marcha el año pasado un curso de doctorado denominado “Academic English for Engineers”, con el objetivo de ofrecer a sus estudiantes una serie de pautas que les permita desenvolverse en diversas situaciones académicas y profesionales en inglés.

La convergencia al EEES implica un proceso de internacionalización que está fomentando un aprendizaje del inglés basado en competencias instrumentales (comunicación por medio de la lengua, más que conocimiento de la misma) e interpersonales (intercambio de información, desarrollo de la capacidad crítica en la recepción de textos, conocimiento del contexto internacional, etc). Así, la precisión gramatical tiene una importancia relativa si lo que se pretende es un uso funcional, pero sí es importante que un ingeniero, por ejemplo, utilice con fluidez las estructuras pasivas (habituales en los textos científico-técnicos en inglés) o las estrategias lingüísticas necesarias para definir, clasificar, contrastar o argumentar.

Existen buenos diccionarios técnicos, y páginas *web* que ofrecen glosarios relacionados con las diversas especialidades de la ingeniería. Están diseñados para salir del paso con la unidad mínima de significado, la palabra, designando términos, propiedades o procesos. “Aprender palabras” es, cómo no, un buen comienzo. Sin embargo, las palabras adquieren todo su significado dentro de oraciones, y las oraciones dentro de párrafos, que a su vez constituyen textos. La descontextualización y la falta de relaciones asociativas sustraen gran parte de la utilidad semántica de las palabras. El uso de cada una de las destrezas lingüísticas (escribir, leer, hablar y entender) siempre depende del propósito comunicativo y el foro a los que van destinadas. Presentar un proyecto no requiere los mismos giros o registro formal que un currículum *vítæ*, y un ensayo académico recurre a ciertas estructuras que sobran en un correo electrónico solicitando información. Cuestiones de estilo aparte (que abarcarían incluso aspectos socioculturales), el lenguaje científico y técnico codifica la información de forma particular, y por tanto es necesario aprender estos códigos para poder producir y recibir textos de ingeniería en lengua inglesa [1].

No obstante, el trabajo científico y técnico, especialmente el que se realiza en un entorno colaborativo como el de la ingeniería, necesita **buena preparación comunicativa** en cualquier idioma. Indudablemente, aprenderlo en inglés contribuye a aplicarlo también en la lengua nativa. A las destrezas lingüísticas antes mencionadas hay que sumar una más, tan importante como las anteriores: la social. Un informe de la Leicester Polytechnic alertaba sobre las carencias comunicativas de los titulados en ingeniería en el Reino Unido [2]. Según sus autores, los ingenieros británicos no estaban adecuadamente preparados para desarrollar ciertas actividades comunicativas demandadas por su profesión (expresión de ideas concisa y organizada, análisis de información, valoraciones críticas, capacidad de síntesis...). Monk, por su parte, reconoce en un estudio similar la importancia de las lenguas para desenvolverse en el

mercado globalizado, pero también apunta que los ingenieros, en primer lugar, deben desarrollar su competencia en lengua nativa tanto en forma oral como escrita, de modo que se les ofrezca la oportunidad de ejercitarse en la producción de las explicaciones, argumentos y especificaciones que encontrarán en su entorno profesional [3].

“The tech sector is a marvellous linguistic factory that ships out truckloads of new words and phrases every year”

(P. McFedries – IEEE Spectrum, abril 2005)



La idea que subyace tras estas propuestas es que, en ocasiones, un conocimiento gramaticalmente impecable del inglés no implica necesariamente una buena interacción con el interlocutor. Se hace preciso **combinar las destrezas lingüísticas con la social**, de modo que seamos capaces de generar textos, hablados o escritos, bien estructurados y comprensibles. Del mismo modo, en el momento de la recepción de textos en inglés, el conocimiento de esas mismas estrategias retóricas de la lengua puede desarrollar nuestra capacidad analítica y crítica.

A pesar de que el inglés es una lengua relativamente asequible (en comparación con el español, por ejemplo) lo que los filólogos llamamos lingüística contrastiva nos puede ayudar a perfilar cuáles son las dificultades con las que puede encontrarse un ingeniero español a la hora de desarrollar su competencia comunicativa en inglés. Es evidente que una de las primeras lecciones debe consistir en detectar las diferencias del lenguaje numérico: lo que en español escribimos “15.500 kW” en inglés se escribe “15,500 kW”, y “42,3° C” pasa a ser “42.3° C” en un texto en lengua inglesa. Sin embargo, para alguien cuya lengua madre es el español, la rigidez sintáctica del inglés y la brevedad de sus oraciones —propiedades que contribuyen enormemente a la claridad expositiva del lenguaje científico— presentan un problemático contraste con la flexibilidad y la verbosidad propias de nuestra lengua. En otras palabras: aunque podamos aprender más inglés, tal vez nos interese **aprender un uso más efectivo del inglés que ya sabemos**. Si lo expresamos en términos ingenieriles, el inglés es un instrumento del que conocemos su utilidad, primer paso para obtener un óptimo rendimiento del mismo.

En conclusión (que, por cierto, es una excelente práctica comunicativa), el inglés ha afianzado su condición de lengua comodín en el contexto de la globalización, y su utilidad no es ajena a la ingeniería. Las universidades españolas se están incorporando a la internacionalización de los estudios universitarios, con el ánimo de formar a sus titulados para que utilicen la lengua inglesa de forma efectiva en su vida profesional. Sin embargo, no conviene pasar por alto que las habilidades lingüísticas que les ofrezca su aprendizaje del inglés pueden ser aplicadas a su lengua nativa, a fin de desarrollar la destreza social necesaria en un entorno tan colaborativo como es el de la ingeniería.

Referencias

- [1] P. Zhihong y F. Yuejin. "Language and Engineers." *Engineering Science and Education Journal*. Vol. 10, nº 1, Feb. 2001, p. 13.
- [2] M. C. Pérez-Llantada. "Formación en competencias instrumentales, interpersonales y sistémicas para el aprendizaje de lenguas extranjeras en el marco del EEES". http://www.unizar.es/eees/innovacion06/COMUNIC_PUBLI/BLOQUE_II/CAP_II_10.pdf. 2 de octubre de 2007.
- [3] J. Monk. "The Role of Language in Engineering Competence". *Exploring Innovation in Education and Research*. <http://oro.open.ac.uk/5584/01/14-003.pdf>. 2 de octubre de 2007

ENGLISH ZONE

Por: Gloria Murillo Cordero

Coordinadora del Boletín de la rama IEEE – UNED
E-mail: gmcordero@indra.es



Debido a la importancia que tiene la lengua inglesa en nuestros días tal y como se ha detallado en el anterior artículo escrito por la profesora Maria M. García, hemos decidido crear en este Boletín una nueva sección enfocada, a recordar o aprender ciertos aspectos de la lengua inglesa.

1. Verb Tenses

Present Perfect vs Present Simple

Strangers on a Plane

When does the present perfect change to the simple past?

Nicole Morgan is sitting on a plane. She is flying back to Australia, her native land. She has been living in Spain for 10 years but now she intends to leave the country for good. She has a job offer in Melbourne and she has accepted it. She is sitting next to a Spanish man who is travelling to Australia on business. The departure of the flight is scheduled for 11:00 a.m.

In this conversation, the time is 10:55 a.m.		In this conversation, the time is 11:25 a.m.	
Spaniard:	How long have you been in Spain?	Spaniard:	How long were you in Spain?
Nicole:	I've been here for 10 years.	Nicole:	I was there for 10 years.
Spaniard:	Have you been in Madrid the whole time?	Spaniard:	Were you in Madrid the whole time?
Nicole:	Well, really I've never lived in Madrid. I've been living the entire 10 years in Guadalajara.	Nicole:	Well, really I never lived in Madrid. I lived the entire 10 years in Guadalajara.
Spaniard:	Has it been a good experience for you?	Spaniard:	Was it a good experience for you?
Nicole:	Yes, it's been a wonderful experience. I've met a lot of people and I've made a lot of friends.	Nicole:	Yes, it was a wonderful experience. I met a lot of people and I made a lot of friends.

Fijaos que nada más despegar el avión, el tiempo del verbo cambia. Cinco minutos antes del despegue se emplea lo que en inglés se llama el presente perfecto y, nada más despegar, se abandona este tiempo en busca del pasado simple.

El pasado simple entra en escena en el momento en que las acciones en cuestión dejan de tener validez en el presente. Por eso a los angloparlantes les os extraña cuando nos oyen en la oficina, decir “No he dormido bien esta noche”. En inglés decimos “I didn’t sleep well last night” (No dormí bien anoche). Lo decimos así porque ya estamos en la oficina y ya es por la mañana. La noche ya no es vigente.

2. One phrasal verb

To pick up – aprender

I picked up a lot of useful tips on the course.

Aprendí muchos trucos útiles en el curso.

3. One adjective

Heaving - Atestado de gente

It was absolutely heaving on the train this morning.

4. One expression

If you can’t beat them, join them.

Si no puedes con ellos, únete a ellos.

5. False Cognates

One of the **great** things about learning Spanish or English is that many words have the same roots in the Romance languages and English. However, there are also a great many **falsos amigos**, or false cognates, which look similar but are in fact very different. This is one of the biggest pitfalls for students of Spanish. There are also **semi-false cognates**: words that can only sometimes be translated by the similar word in the other language.

Because some of the words are identical in the two languages, the Spanish word appears in **blue** and the English in **purple**.

Alterado vs Altered

Alterado can mean *changed* or *altered* as well as *angry* or *upset*.

Altered = *modificado, cambiado, alterado*.

Echar vs Echo

Echar has numerous meanings, including to *throw*, to *put*, to *pour*, to *give*, to *cut*, and to *push*.

Echo means *resonar, repetir, or hacer eco*.

Fastidioso vs Fastidious

Fastidioso means *annoying* or *boring*. It can mean *fastidious* in Latin America.

Fastidious means *escrupuloso*.

Fiscal vs Fiscal

Fiscal means *fiscal* or *tax-related* as an adjective. As a noun, however, it refers to a *district attorney* or *public prosecutor*.

Fiscal = *fiscal*.

Mango vs Mango

Mango can mean *mango* the fruit as well as a *handle* (as of a knife).

Mango = *mango*.

6. Work Interview

¿Darías una buena impresión si tuvieras que realizar una entrevista de trabajo en inglés?
¿A veces no entiendes bien las preguntas o no sabes qué contestar? A continuación podemos ver una lista de preguntas que nos pueden realizar en una entrevista de trabajo.

Tell us about yourself

Esta pregunta es muy amplia ya que incluye tu carácter, educación, ambiciones, situación familiar, experiencia laboral...etc

Why should we hire you?

La persona que te entrevista quiere saber cuáles son tus cualidades y puntos fuertes.

Why did you leave your last job?

La persona que te entrevista quiere saber por qué quieres cambiar de empleo: si renunciaste al trabajo, te echaron o bien te quedaste sin empleo por reducción de la plantilla.

Are you willing to relocate?

Significa que estás dispuesto/a a vivir en otra localidad o país si fuera necesario.

Tell me about your scholastic record.

Se refiere a las notas que obtuviste en el colegio y en la universidad.

Tell me about your extra-curricular activities and interests.

Quiere saber qué tipo de actividades te interesaban fuera de clase y si tienes algún hobby o afición.

How would your last boss describe you?

Quiere saber qué opinaba tu jefe anterior sobre tí.

Where do you see yourself in 5 years time?

Qué planes tienes para el futuro y dónde crees que estarás en 5 años

How do you work under pressure?

Esta pregunta se refiere a si te las arreglas bien cuando hay mucho trabajo, límites de tiempo o situaciones de estrés.

What salary are you expecting?

La persona que te entrevista quiere saber el salario mínimo que aceptarías.

7. Crossword Puzzles (Antonyms)

The object of the game is to fill-in the 12 blank boxes with words that are related to the given clues.

	A	B	C	D	E	F	G	H	I	J	K	L
1												
2												
3												
4												
5												
6												
7												
8												
9												
10												
11												
12												

Across :

G3 exciting (8)
A4 listen (5)
B6 gloomy (8)
B10 slender (5)
H11 entrance (4)
A12 follow (4)

Down :

B3 crowded (8)
D1 destroy (6)
D8 drought (5)
F5 dull (6)
I2 solution (7)
K2 divide (5)

¹SOLVE

4. Links:

<http://www.englishtown.es/sp/article.aspx?articleName=9&Otag=O00314>

<http://www.vocabulary.co.il/games2/crosswords/index.php#main>

<http://www.vausys.com/>

<http://www.vaughanradio.com/reproducer/player2.htm>

<http://www.elearnspanishlanguage.com/vocabulary/falsosamigos.html>

¹ ACROSS → G3: boring, A4: speak, B6: cheerful, B10: stout, H11: exit, A12: lead.
 DOWN → B3: spacious, D1: create, D8: flood, F5: bright, I2: problem, K2: unite.

¿Qué es la VoIP?

Por: Serafín Doblado Moreno

Administrador de Servicios de VoIP de VOCARI
E-mail: serafin_doblado@hotmail.com



1. Introducción

Mucho se habla en los últimos tiempos de la evolución de las TIC², no cabe la menor duda que estamos ante tiempos de cambios, donde el denominador común es hacer mejor y más práctico nuestro mundo. Y se ha eso le añadimos el ahorro de dinero, la cosa empieza a pintar mejor.

Sea como fuere, los conceptos actuales que tenemos de “comunicación e intercambio de información”, van a dar en no mucho tiempo unos cambios serios. El crecimiento constante de la demanda de ordenadores, redes, banda ancha....va a ser el percusor para la expansión y comercialización de la tecnología que está empezando a despuntar en los sectores más innovadores: la VoIP. Así se le ha bautizado.

2. ¿Qué es la VoIP?

Antes de ponernos a definir este concepto habría que tener claro otros, como por ejemplo IP³ y OSI⁴. Pero este artículo, lejos de ser una engorrosa explicación del funcionamiento de los sistemas de interconexión y comunicación, intentaré arrojar un poco de luz sobre el mundo de VoIP⁵ de una forma sencilla y clara.

Si a Bell⁶, le hubieran dicho que su invento seria la base para todo el desarrollo de las redes telemáticas, se lo hubiera creído –no era tonto. La telefonía es el sistema base, a partir de él se forman todos los sistemas necesarios para hacer nuestra vida diaria mas cómoda, económica y por supuesto, engorrosa. Todo el mundo habla de Internet, pero no muchos saben que es, se habla de VoIP, ¿pero saben qué es esto? Esta claro, que al usuario solo le interesa levantar la el teléfono y escuchar el tono de línea. Por otro lado, es lo normal, vivimos en

² TIC: Tecnologías de la Información y la Comunicación.

³ IP: Internet Protocol. Es el protocolo “defacto” en el nivel Red en TCP/IP.

⁴ OSI: Open System Interconnection. Modelo de referencia para las redes telemáticas.

⁵ VoIP: Voz Over IP. Tecnología que usa las redes telemáticas para dar soporte de Voz.

⁶ Bell: Graham Bell. Inventor del Teléfono.

una sociedad que tecnológicamente avanza a pasos agigantados, y estar al día es algo impensable.

La VoIP, no es más que la evolución natural del sistema tradicional de telefonía. Su funcionamiento está basado en aplicaciones de entornos TCP/IP⁷, es decir, que “corre” bajo el mundo de las redes de datos. Al ser así, su principal característica es que funciona bajo la “conmutación de paquetes”, y no bajo la “conmutación de circuitos” (funcionamiento de la RTC⁸). Esto ya es una clara ventaja, para establecer una llamada ya no se tiene que reservar de forma explícita un circuito dedicado. Con la VoIP, las llamadas se “paquetizan” y son tratados del mismo modo que cualquier trama IP.

3. Elementos de un sistema de VoIP

Definido de forma llana el concepto, vamos a ver que nos hace falta para hablar a través de esta tecnología.

El propio estándar define tres elementos básicos en su arquitectura de red:

- **Terminales:** Son los sustitutos de los actuales teléfonos. Existen tanto en software (SJPhone⁹ y Kapanga), como en hardware (Linksys¹⁰ y GrandStream). Estos últimos tienen todo el aspecto físico de los tradicionales. Se configuran como un elemento más de la red, es decir, hay que indicarle su dirección IP, máscara, puerta de enlace,... y todos los parámetros necesarios para establecer el tono. Dicho así, parece algo para gurús, pero os aseguro que llevan una interfaz web sencilla y muy intuitiva.
- **Gatekeepers:** Son el core de toda la organización VoIP, son llamados a sustituir a las actuales centrales. Son implementadas en software, de existir, todas las comunicaciones pasarían por él. (Asterisk¹¹)
- **Gateways:** se trata de un enlace con la red telefónica tradicional, las llamadas que tengan que ir a la RTC, serán encaminadas por este enlace, el resto seguirá su curso por la red de datos hasta llegar al destino. Estos elementos proveen de las interfaces TDM¹² genéricas, como los E1¹³ (primarios).

Con estos tres elementos tendríamos cubiertas las necesidades de arquitectura para soporte de VoIP en dos delegaciones de una misma empresa. La ventaja es inmediata: todas las comunicaciones entre las dos delegaciones serían completamente gratuitas.

⁷ TCP/IP: Estandar “defacto” en las redes telemáticas.

⁸ RTC: Red Telefónica Conmutada.

⁹ SJPhone: Terminal Software para la VoIP. Es de libre distribución. www.sjphone.com

¹⁰ LinkSys: Terminal de Sobremesa para la VoIP. Es una marca blanca de CISCO.

¹¹ Asterisk: Software distribuido bajo licencia GPL. Despliega todos los servicios necesarios para dar soporte a VoIP.

¹² TDM: Multiplexión por División en el Tiempo. Forma de trabajar la RTC.

¹³ E1: Es un acceso Primario. Puede soportar hasta 30 canales de Voz.

4. Arquitectura de Protocolos

El estándar TCP/IP no fue pensado para el transporte de tráfico multimedia, su funcionalidad nativa es dar soporte a la transmisión de datos. Con toda la nueva convergencia de tecnologías y servicios, se ha tenido que adaptar la red a las nuevas necesidades. Con este fin, se crean los protocolos necesarios para dar soporte a este tipo de tráfico.

Hay que entender que las redes de datos solo transmiten información cuando es necesario, aprovechando al máximo el ancho de banda. El retardo, la alteración de llegada de los paquetes o la pérdida de paquetes no son un inconveniente, ya que en el sistema final se tienen una serie de procedimientos de recuperación; pero en la voz, estos factores son altamente influyentes. Por lo tanto se requieren redes y protocolos que ofrezcan alto grado de QoS¹⁴.

La VoIP está soportada por los siguientes protocolos:

- SIP¹⁵: Es el protocolo de señalización a nivel de aplicación, para la gestión y establecimiento de la llamadas. Su descripción viene en la RFC 3261¹⁶. Se encarga de negociar los parámetros de la comunicación de extremo a extremo. Se basa en mensajes de petición/respuesta y reutiliza muchos conceptos de los estándares HTTP¹⁷ ó SMTP¹⁸. Existen seis métodos básicos (definidos en la RFC 254) que describen las peticiones de clientes:
 - INVITE: Permite invitar a un usuario ó servicio para participar en una llamada.
 - ACK: Confirma el establecimiento de una llamada.
 - OPTION: Solicita información sobre las capacidades de un servidor.
 - BYE: Indica terminación de llamada.
 - CANCEL: Cancela una petición pendiente.
 - REGISTER: Registrar un usuario.



¹⁴ QoS: Quality of Service. Se define como la Calidad del Servicio. Parámetro muy importante en VoIP.

¹⁵ SIP: Protocolo de Inicio de Sesión. La señalización en Asterisk se hace por el puerto 5060 en UDP.

¹⁶ RFC 3261: Documento de referencia estandarizado por la ITU-T, donde se define el servicio.

¹⁷ HTTP: Hiper Text Transport Protocol. Es el protocolo “defacto” del servicio WEB.

¹⁸ SMTP: Simple Mail Transport Protocol. Es el protocolo “defacto” del servicio de MAIL.

A continuación un ejemplo real del método *REGISTER*:

```
Via: SIP/2.0/UDP
192.168.0.100:5060;rport;branch=z9hG4bK64646410000000b43c52d6
c00000d1200000f03
Content-Length: 0
Contact: <sip:20000@192.168.0.100:5060>
Call-ID: ED9A8038-A29D-40AB-95B1-0F5F5E905574@192.168.0.100
CSeq: 36 REGISTER
From: <sip:20000@192.168.0.101>;tag=910033437093
Max-Forwards: 70
To: <sip:20000@192.168.0.101>
User-Agent: SJphone/1.60.289a (SJ Labs)
Authorization: Digest
username="20000",realm="192.168.0.101",nonce="43c52e9d29317c0bf
1f885b9aaff1522d93c7692"
,uri="192.168.0.101",response="f69463b8d3efdb87c388efa9be1a1e63"
```

- *H.323*¹⁹: Fue diseñado con una finalidad clara: proveer a los usuarios con capacidad de tráfico multimedia, de voz y video a través de la red de conmutación de paquetes. Su principal problema frente a SIP es su falta de flexibilidad. SIP está menos definido, pero es más fácil de integrar. Por este y otros motivos (no da lugar en este artículo), la batalla por estandarizar la VoIP, la ha ganado con amplio margen SIP.
- *RTP*²⁰: Son las siglas de *Real-Time Transport Protocol*, es un protocolo de nivel de aplicación (no de transporte como su nombre podría hacer pensar). Es el protocolo utilizado para la transmisión en tiempo real de la voz. Este servicio corre bajo los puertos del 10000 al 2000 en UDP²¹. La pregunta es evidente, ¿tengo que abrir en mi Router²² todos estos puertos?... No, claro que no, sería una brutalidad en cuestión de seguridad. Para que este tráfico fluya de forma óptima, la mayoría de los Routers actuales se proveen del parámetro uPNP; solo hay que ponerlo a "enabled", y la gestión de apertura/cierre de puertos necesario para el RTP de VoIP, lo hará el Router de forma dinámica.

¹⁹ H.323: Estandar diseñado explícitamente para tráfico multimedia. Ha quedado obsoleto.

²⁰ RTP: Real-Time Transport Protocol. Diseñado para el transporte del tráfico multimedia, por ejemplo la Voz.

²¹ UDP: Unid Datagram Protocol. Protocolo NO ORIENTADO a conexión en la capa 4 del estándar TCP/IP.

²² Router: Elemento de Interconexión que trabaja en la capa 3 estándar TCP/IP.

5. Calidad del Servicio

El auge de la telefonía IP es algo evidente y la principal razón es el reaprovechamiento de los recursos y la disminución *en el coste de llamadas a través de Internet*.

Sin embargo, si de algo adolece todavía la VoIP es de la calidad de los sistemas telefónicos tradicionales. Los problemas de esta calidad son muchas veces inherentes a la utilización de la red (Internet y su velocidad y ancho de banda) y podrán irse solventando en el futuro. Mientras tanto, cuanto mejor conozcamos los problemas que se producen y sus posibles soluciones mayor calidad disfrutaremos (no vamos a entrar en detalles en cada uno de ellos, ya que esto nos llevaría toda una literatura).

Los principales problemas en cuanto a la calidad del servicio (QoS) de una red de VoIP, son la *Latencia*²³, el *Jitter*²⁴, la *pérdida de paquetes* y el *Eco*²⁵. En VoIP estos problemas pueden ser resueltos mediante diversas técnicas tanto hardware como software.

Hay que quedar claro que tanto las redes telemáticas actuales como el protocolo TCP/IP, no fueron diseñados para dar soporte eficiente de QoS.

6. Codecs

El mundo de lo digital, por desgracia no entiende de lo analógico. Para ello están los CODificadores-DECodificadores. En la VoIP, los codecs pasan la señal de voz (analógica) a señal digital para poder ser empaquetada y transportada. En el remoto, la señal es decodificada y pasada a analógica, para poder ser oída por nosotros.

Entre los CODECS más estandarizados tenemos: el iLBC²⁶, GSM, G.723 y G.729. Todos son soportados por la mayoría de terminales y sistemas de VoIP estándares, siendo los tres primeros “libres” y el último, bajo licencia para su uso.

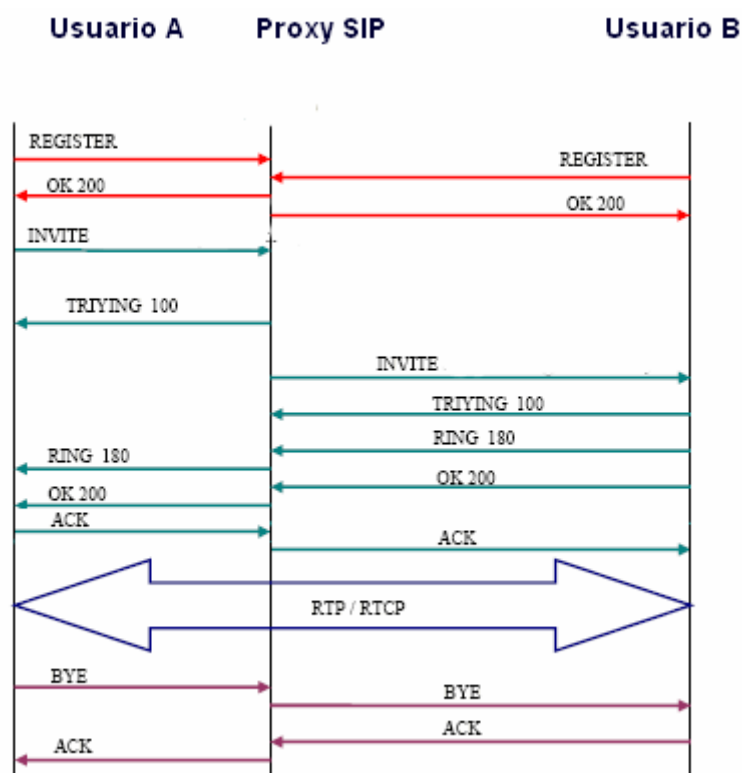
Para terminar este apartado dedicado a definir de manera muy simple la tecnología de VoIP, se muestra una típica secuencia de una llamada de teléfono. En esta imagen se resume de forma gráfica, todo lo dicho anteriormente:

²³ Latencia: Tiempo que tarda en llegar un paquete al destino.

²⁴ Jitter: Se define como la variación de la Latencia. Este parámetro se requiere que sea constante.

²⁵ Eco: Fenómeno que técnicamente se define como retorno de la señal. Es bastante incomodo, ya que oyes tu propia voz.

²⁶ iLBC, GSM, G.723 y G.729: Son CODECS diseñados para digitalizar la voz humana.



- **Las dos primeras transacciones corresponden al registro de los usuarios.** Los usuarios deben registrarse para poder ser encontrados por otros usuarios. En este caso, los terminales envían una petición REGISTER. El servidor Proxy (Asterisk), consulta si el usuario puede ser autenticado y envía un mensaje de OK en caso positivo.

- **La siguiente transacción corresponde a un establecimiento de sesión.** Esta sesión consiste en una petición INVITE del usuario al proxy. Inmediatamente, el proxy envía un TRYING 100 para parar las retransmisiones y reenvía la petición al usuario B. El usuario B envía un Ringing 180 cuando el teléfono empieza a sonar y también es reenviado por el proxy hacia el usuario A. Por ultimo, el OK 200 corresponde a aceptar la llamada (el usuario B descuelga).

- **En este momento la llamada está establecida,** pasa a funcionar el protocolo de transporte RTP con los parámetros (puertos, direcciones, codecs, etc.)

- **La última transacción corresponde a una finalización de sesión.** Esta finalización se lleva a cabo con una única petición BYE enviada al Proxy, y posteriormente reenviada al usuario B. Este usuario contesta con un OK 200 para confirmar que se ha recibido el mensaje final correctamente.

En esta sección del artículo, trataré de llevar a la práctica un modelo simple de sistema basado en VoIP. El core del sistema será el Asterisk. Explicaré su

instalación y puesta a punto en Linux, así como la configuración básica de sus ficheros principales. Se tratará un caso real: dos terminales de una misma LAN quieren hablar entre ellos.

No olvidemos que VoIP es una tecnología basada en IP, y que Asterisk es una aplicación que da soporte a VoIP. Comencemos.

7. ¿Qué es Asterisk?

Asterisk es una centralita software (PBX) de código abierto. Como cualquier centralita PBX permite interconectar teléfonos y conectar dichos teléfonos a la red telefónica conmutada (RTC). Su nombre viene del símbolo asterisco (*) en inglés.

El creador original de esta centralita es Mark Spencer de la compañía Digium que sigue siendo el principal desarrollador de las versiones estables. Pero al ser de código libre, existen multitud de desarrolladores que han aportado funciones y nuevas aplicaciones. Originalmente fue creada para sistemas Linux pero hoy en día funciona también en sistemas OpenBSD²⁷, FreeBSD, Solaris Sun y Windows. Pero como no podía ser de otra manera, Linux es su principal y mejor soporte.

Asterisk soporta todos los protocolos estándares de VoIP, es decir, H.232, SIP, RTP, así como los codecs (G.723, iLBC, GSM, G.729...).

El paquete básico de Asterisk incluye muchas características que antes sólo estaban disponibles en caros sistemas propietarios como *creación de extensiones, envío de mensajes de voz a e-mail, llamadas en conferencia, menús de voz interactivos, distribución automática de llamadas...* es decir, todas las prestaciones de cualquier centralita, en un solo software y además gratis!!

Requisitos para Instalar Asterisk

Para los que estéis relacionados con el mundo de Linux, de sobra sabréis que con este sistema operativo se aprovechan al máximo todos los recursos. Por lo tanto, si tenéis un viejo 486 ó PI, este os valdrá para montar un Proxy Asterisk.

En mi viejo portátil he montado una Debian²⁸, cada cual lo puede hacer en la versión que mas le guste, tarjeta de red, consola y conexión a Internet.

²⁷ OpenBSD, FreeBSD... Son Sistemas Operativos basado en UNIX.

²⁸ Debian: Una de la versiones de Linux. La última versión es la ETCH. con Kernel 2.6.x

Pasos Previos

Antes de la instalación de Asterisk tenemos que asegurarnos de las dependencias del sistema. Para saber que versión del Kernel estamos usando ejecutamos el comando `# uname -a`, averiguado esto, instalamos lo necesario:

- `# apt-get29 update` –para actualizar el repositorio
- `# apt-get install kernel-headers-VERSION_DEL_KERNEL`
- `# apt-get install libssl-dev`
- `# apt-get install libncurses5-dev`
- `# apt-get install zlib1g-dev`
- `# apt-get install gcc`

Esto son las librerías básicas para la compilación de Asterisk, si os hace falta alguna mas, cuando hagáis el make del Asterisk os lo dirá; y ya sabéis: `apt-get install` (Debian es una maravilla, nos ahorra mucho trabajo).

Acto seguido nos vamos bajamos el Asterisk y nos ponemos manos a la obra. Actualmente están por la 1.4, yo la que tengo corriendo en mi portátil es la 1.2.24 (última versión de la serie 1.2).

- `wget http://ftp.digium.com/pub/asterisk/releases/asterisk-1.2.24.tar.gz`
- `# mv asterisk-1.2.24.tar.gz /usr/src` – movemos el tar.gz
- `# cd /usr/src/`
- Descomprimes: `# tar -xvzf asterisk-1.2.24.tar.gz`
- `# cd asterisk-1.2.24`
- `# ./configure`
- `# make`
- Si todo ha ido bien: `# make install`
- Si te ha dado error, no habrás llegado al make install, lo mas seguro es que te falte alguna librería, fíjate en el debug del make y sabrás de que librería se trata, luego, `apt-get.....`
- Si es la primera vez que instalas la centralita, te recomiendo que instales los ejemplos: `# make samples`

Si hemos llegado ha este punto, tendremos nuestro Asterisk listo para “correr”, finalmente arrancamos el Asterisk con el comando:

- `# asterisk -vvvc`

Verás mucho verbose (debugging) en la pantalla cuando Asterisk se inicializa. (las vvv pertenecen al modo " very very verbose" y la c a que nos mostrará al final una línea de comandos en forma consola)

- `*CLI>`

²⁹ APT-GET: Herramienta nativa en Debian para la actualización de paquetes.

A partir de este momento tenemos el Asterisk funcionando, de aquí en adelante ya es lo que cada cual se quiera complicar. En la red hay infinidad de sitios donde se habla de cómo y de que manera explotar este sistema. Un Ejemplo:

Vamos a poner en práctica el ejemplo mas sencillo: voy a crear dos extensiones: usuarioA y usuarioB dentro de un mismo contexto "pruebaContexto" y voy a configurar Asterisk para que A pueda llamar a B y viceversa.

Como todo en Linux, las configuraciones son a través de ficheros planos, por defecto los ficheros principales de Asterisk son extensions.conf y sip.conf. Se encuentran dentro de /etc/Asterisk.

sip.conf

Aquí se configuran los parámetros generales del servidor, así como todo lo relacionado con sip y la creación de usuarios:

Un ejemplo básico,

[general]

context=default

port=5060 ; Puerto UDP en el que responderá el Asterisk

bindaddr=0.0.0.0 ; Si queremos especificar que Asterisk esté en una IP (si un equipo tiene 3 IPs por ej.) 0.0.0.0 vale para cualquiera

srvlookup=yes ; Habilita servidor DNS SRV

allow=ilbc ; Se habilita el codec iLBC

allow=gsm ; Se habilita el codec GSM

[pruebaContexto] ; Se define el contexto del usuario

type=friend

username=usuarioA

password=1234 ;por ejemplo(puedes poner la que quieras)

[pruebaContexto] ; Se define el contexto del usuario

type=friend

username=usuarioB

password=1234 ;por ejemplo(puedes poner la que quieras)

De este modo tan sencillo hemos dado de Alta en la Centralita Asterisk a dos usuarios, puedes parametrizarlo mucho más...así que... a empollar.

extensions.conf (DialPlan)

El archivo extensions.conf es el más importante del Asterisk y tiene como misión principal definir el dialplan o plan de numeración que seguirá la centralita para cada contexto y por tanto para cada usuario.

El fichero extensions.conf se compone de secciones o contextos entre corchetes []. Hay dos contextos especiales que están siempre presentes que son [general] y [globals]

A continuación definimos el contexto de los usuarios, es decir, voy a definir que “quiero que haga la llamada, cuando A marca a B y viceversa”

- *[pruebaContexto]*
- *exten => usuarioA,1,Dial(SIP/usuarioA,10,m)*
- *exten => usuarioA,2,Handup*
- *exten => usuarioB,1,Dial(SIP/usuarioB,10,m)*
- *exten => usuarioB,2,Handup*

Significado: “Cuando entre una llamada a usuarioA, con prioridad 1 ejecuta un Dial (llamada) al usuarioA, haz ring durante 10 sg y si no contesta, con prioridad 2, cuelga”

Igual para el usuarioB.

De este modo, hemos indicado la ruta que ha de seguir la llamada.

Ya tenemos definido y listo todo en el Asterisk, ahora solo nos quedaría configurar los clientes, en este caso, podemos configurar dos SJPhone, uno con cada extensión y probar a llamarnos.

De este modo tan sencillo, hemos montado un proxy Asterisk y lo hemos “echado” a andar. Vuelvo a insistir, en que lo hecho aquí ha sido una configuración básica. Cada caso puede tener unas necesidades muy concretas. Pero solo es cuestión de tiempo y paciencia.

Una visión general de la tecnología VoIP con Asterisk sería la siguiente (2 casos muy comunes):

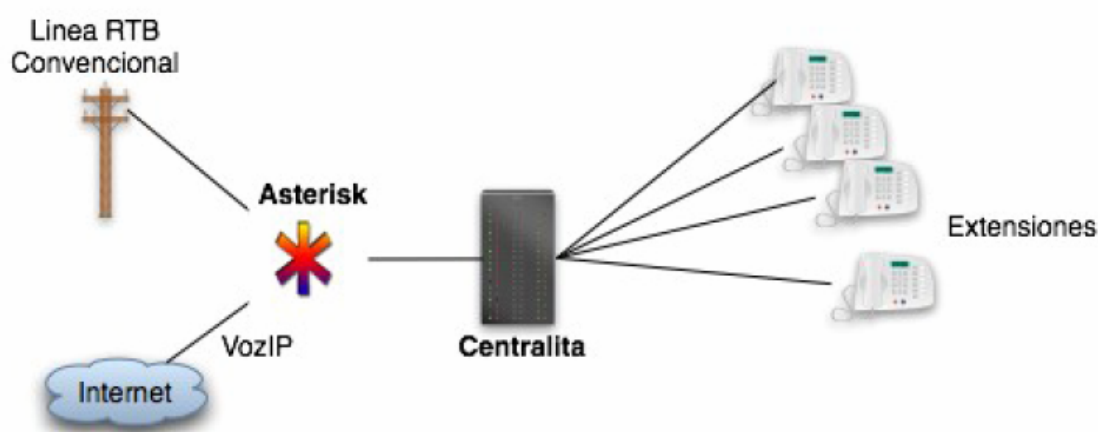


Figura 1: Reutilización de centralita analógica a través de Asterisk.

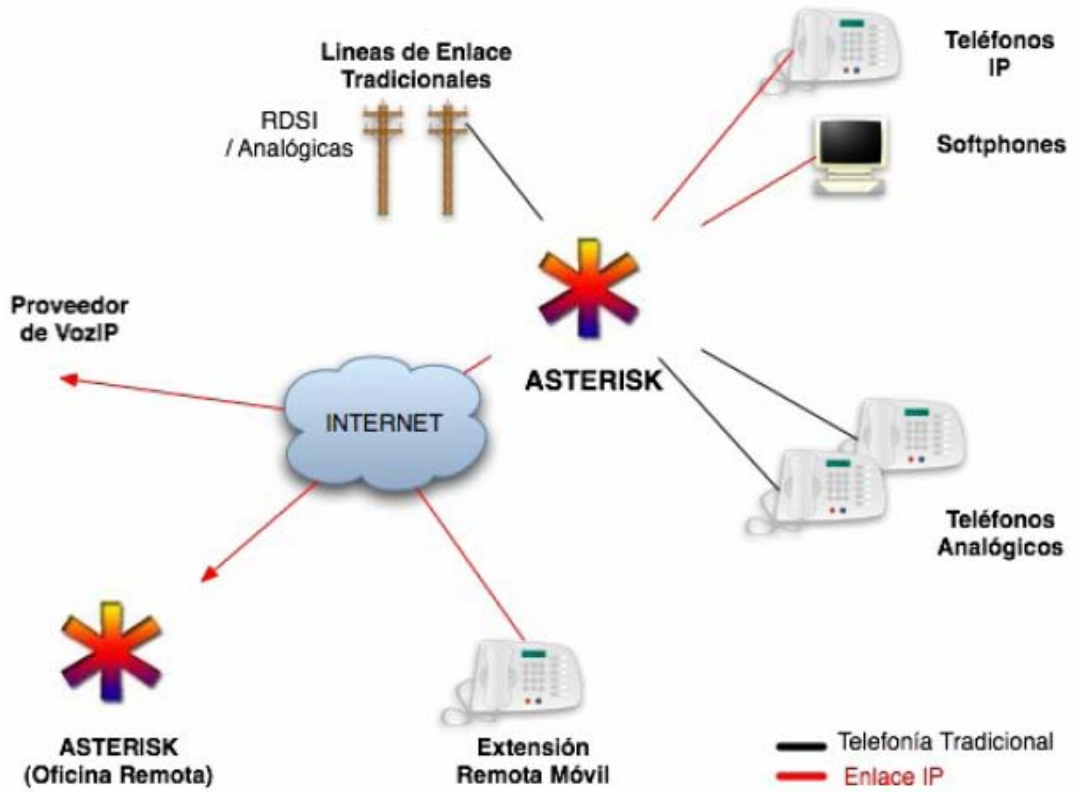


Figura 2: Escenario típico del sistema VoIP con Asterisk.

8. ¿Quién explota Asterisk?

Pues como podréis imaginar, trabajo como administrador de Sistemas de VoIP para una operadora, su nombre des VOCARI. Ofrecemos tanto servicios avanzados de hosting como de VoIP. Nuestro mercado principal lo hemos centrado en la PYMES, aunque es totalmente valido para cualquier entidad que quieras sacar rendimiento a su ADSL y ha esta tecnología.

Si quieres consultarnos, puedes hacerlo en www.vocari.com.

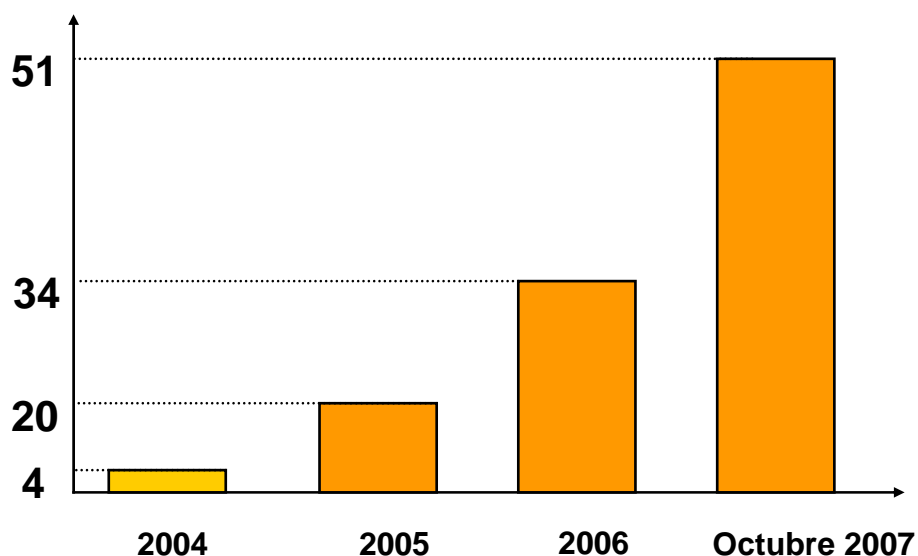


9. Referencias

- www.asterisk.org/
- es.wikipedia.org/wiki/Asterisk
- www.asterisk-es.org/
- www.voip-info.org/
- www.digium.com/
- www.digium.com/handbook-draft.pdf
- www.voipforo.com/

INFORMACIÓN GENERAL RESUMIDA

La Rama de Estudiantes creada en la Universidad Nacional de Educación a Distancia (UNED) tiene por objetivo principal **la difusión de la ciencia y la tecnología**. Actualmente hay 52 miembros dentro de la rama, lo que supone un crecimiento significativo en el último año. En la siguiente figura se puede contemplar de manera gráfica dicha evolución:



La información general sobre sus actividades e información de cómo hacerse miembro se puede ver en la página Web:

[http:// www.ieec.uned.es/IEEE/](http://www.ieec.uned.es/IEEE/)

dentro del enlace de la Rama de Estudiantes.

Las actividades principales que las Ramas de España realizan son: charlas, cursos, congresos, concursos, actividades educativas, visitas a empresas y organizaciones, interrelación cultural y multidisciplinar y cualquier actividad que quiera desarrollar cada uno de sus miembros.

Actualmente puede participar cualquier estudiante de las carreras de Informática y de Industriales de la UNED. Para conocer más información sobre el IEEE, las Ramas de España y sus posibilidades se recomienda leer los primeros artículos de éste Boletín y visitar la página Web para ver los boletines previos. De todas formas cualquier información o consulta puede dirigirse a Sergio Martín:

smartin@ieec.uned.es



Esperamos que os haya gustado a todos éste octavo Boletín y agradecer una vez más a todos los autores el haber participado en el mismo haciéndolo posible.

UN SALUDO

Sergio Martín

Presidente de la Rama de Estudiantes IEEE-UNED





**Hazte socio
De la Rama de Estudiantes
del IEEE en la UNED**

Web IEEE-UNED

<http://www.ieec.uned.es/IEEE/>

Charlas, conferencias,
cursos, visitas, empresa,
Boletín Electrónico, etc.



**RAMA DE ESTUDIANTES IEEE-UNED
30-OCTUBRE-2007 (BOLETÍN N° 8)**