



**BOLETÍN ELECTRÓNICO
DE LA RAMA DE ESTUDIANTES DEL IEEE DE LA UNED**
IEEE ELECTRONIC JOURNAL OF THE UNED STUDENT BRANCH
ISSN: 1989-2195



Número XI
Edición Abril 2009

© 2009 Rama de Estudiantes IEEE de la UNED
© 2009 IEEE Student Branch of UNED

Boletín Electrónico Rama de Estudiantes de la UNED

Abril-2009

EDITOR

Sergio Martín
(sergio.martin@ieee.org)

EDITOR

Gloria Murillo
(lorycordero151@hotmail.com)

REVISORES

Manuel Castro
Sergio Martín
Elio San Cristóbal
Gloria Murillo

DISEÑO PORTADA

Sergio Martín

AUTORES

Sergio Martín
Pablo Losada
Nuria Oliva
Eduardo Collado
José Antonio Cámara
Gloria Murillo.

AGRADECIMIENTOS

Vicerrectorado de Investigación UNED
Vicerrectorado de Estudiantes y Desarrollo
Profesional UNED
Escuela Técnica Superior de Ingenieros
Industriales UNED
Escuela Técnica Superior de Ingenieros
Informáticos UNED
Sección Española del IEEE
Departamento de Ingeniería Eléctrica,
Electrónica y de Control (DIEEC) UNED
IEEE Women In Engineering (WIE)

AGRADECIMIENTO ESPECIAL

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 de la Sociedad de Educación del IEEE para la elaboración del mismo.

Agradecemos a todos los autores, y a aquellos que han colaborado para hacer posible este Boletín Electrónico.

**BOLETÍN DESARROLLADO EN COLABORACIÓN CON EL CAPÍTULO
ESPAÑOL DE LA SOCIEDAD DE EDUCACIÓN DEL IEEE**



Nueva Junta Directiva 2009



Elio San Cristóbal. Nuevo Presidente de la rama de estudiantes IEEE-UNED. Ingeniero Informático y estudiante de doctorado en el DIEEC. Actualmente trabaja en el Departamento de Ingeniería Eléctrica, Electrónica, y Control en proyectos de investigación.
elio@ieec.uned.es



Germán Carro. Vicepresidente de la Rama de Estudiantes del IEEE-UNED. Economista y Estudiante de Ingeniería Técnica en Informática de Sistemas por la UNED. En años anteriores ha colaborado con la junta directiva como Coordinador de Actividades Generales.
germancf@ieee.org



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.
rgil@ieec.uned.es



Gloria Murillo. Coordinadora del Comité del Boletín Electrónico. Ingeniero Técnico en Telecomunicaciones, y estudiante de Ingeniería Industrial por la UNED.
lorycordero151@hotmail.com



Ramón Carrasco. Coordinador de Actividades Generales. Licenciado en Ciencias Físicas, especialidad Electrónica. Director de Colegio Karbo de la Coruña centro de Educación Infantil, Primaria y Formación Profesional de Grado Medio y Superior.
moncho@warningcorp.com



Guillermo Lafuente. Coordinador de Actividades Generales. Ingeniero Técnico en informática de sistemas por la UNED. También estoy haciendo la certificación CCNA de Cisco Systems.
guiye1984@hotmail.com



Igor Chávez. Coordinador del Grupo de Robótica. Técnico en Electrónica en la National Schools. Actualmente alumno de Ingeniería T. Industrial especialidad Electrónica Industrial de la UNED.
igorchavez@ieee.org



Carlos Conde. Estudiante de Ingeniería Técnica en Informática de Sistemas por la UNED. Miembro de la Rama Estudiantil del IEEE-UNED y Coordinador del Grupo de Robótica del mismo.
carlosch@mundo-r.com



Rubén Alonso Paredero. Coordinador del Grupo de Robótica. Estudiando 5º de Física. Empecé en la UPV-EHU y continué en la UCM. Trabajo en Barcelona como Técnico Especialista en Robótica, desarrollando el proyecto SEAT-EXEO.
rubenalonso78@hotmail.com



José Antonio Cámara. Coordinador Grupo de Control de Procesos. Ingeniero T. industrial, especialidad en electrónica industrial en la universidad de Alcalá, estudiante de ingeniería electrónica (UAH) e ingeniería industrial (UNED), he trabajado en diversos sectores, telecomunicación, automóvil, aerogeneradores y materiales.
jcm92251@alu.uah.es



Alberto Dopico. Coordinador Grupo de Software Libre. Estudiante de Ingeniería Técnica Industrial Electrónica en la UNED.
alberto.dopico@ieee.org

Pablo Calviño. Coordinador Grupo Diseño Web. Experto en desarrollo



web y arquitectura de la información. Técnico Superior en Informática de Gestión y estudiante de Ingeniería de Sistemas por la UNED. Miembro del IAI (Information Architecture Institute).

kemosade@gmail.com

Sergio Martín Gutiérrez. Anterior presidente de la Rama. Actual Coordinador de Publicaciones y del Boletín Electrónico. Ingeniero Informático. Estudiante de doctorado en el DIEEC en temas de Computación Ubicua, Entornos Inteligentes y Aprendizaje con Dispositivos Móviles.

sergio.martin@ieee.org



Manuel Castro. Profesor Consejero de la Rama de Estudiantes del IEEE-UNED. Catedrático de Tecnología Electrónica. Fellow del IEEE y primer presidente del Capítulo Español de la Sociedad de Educación del IEEE.

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

Índice

Últimas Noticias de la Rama de Estudiantes IEEE-UNED	Sergio Martín	6
Servicios Sensibles al Contexto Orientados al Aprendizaje Móvil	Sergio Martín	9
ANDROID - Plataforma libre para desarrollo de aplicaciones para móviles	Pablo Losada Nuria Oliva	13
Retos en la Transición y el Futuro de la Industria Fotovoltaica Española	Eduardo Collado	19
Introducción a los algoritmos Cordic	José A. Cámara	24
English Zone	Gloria Murillo	26

Últimas Noticias Rama Estudiantes IEEE-UNED

Sergio Martín y Germán Carro

sergio.martin@ieee.org, germancf@ieee.org

I. ORGANIZACIÓN CAMPEONATO INTERNACIONAL Y TALLERES DE ROBÓTICA EN LA XGN'09

Desde la delegación de A Coruña de la Rama de Estudiantes del IEEE de la UNED se ha organizado el I Campeonato Internacional de Robótica enmarcado dentro de la XGN'09 (la versión gallega de la Campus Party). Además del campeonato al que han acudido participantes tanto nacionales como internacionales, se organizaron unos talleres de iniciación y unas ponencias sobre robótica.



Figura 1. Cartel publicitario del evento

Día 3 de Abril.- Recepción, alojamiento y presentación del campeonato:

- Puesta en marcha de talleres para aquellos que estén interesados (10 equipos Home Boe Bot)
- Inscripción en las pruebas para aquellos equipos que ya vengan pertrechados con Robots.
- Primer tanteo en la prueba de velocidad (circuito de Malasia), lo que en fórmula 1 se llama entrenamientos libres. Sí, hemos diseñado el mismo circuito en el que

ese fin de semana corre Fernando Alonso. Y vamos a realizar las mismas sesiones para el circuito de velocidad.

- Inicio de Taller en JAVA de "robocode" Impartido por Mateo Soto Ramos (Rama de Sevilla). Para aquellos que prefieren lo virtual frente a lo real.

Día 4 de Abril.

- Sesiones clasificatorias del circuito de velocidad (lo que serían las Q1, Q2 y Q3) de ellas saldrán los participantes en la prueba final del último día.
- Inicio de Pruebas de Sumo Bot.
- Continuación de los talleres de Home Boe Bot
- Continuación del taller virtual de Robocode.
- Inicio campeonato robocode.
- Ponencia sobre robótica (Manel Frigola, Pasado Presidente del IEEE Robotics and Automation y Profesor de la Universidad Politécnica de Catalunya)

Día 5 de abril.

- Prueba del Laberinto: Dos secciones, una con laberinto conocido y otra con laberinto desconocido.
- Finalización de Pruebas de Sumo Bot.
- Fin talleres Boe Bot.
- Fin taller virtual Robocode.
- Fin campeonato robocode.
- Prueba final de velocidad.
- Votación "Robot más destacado" en las tres jornadas (atendiendo a diseño, funcionalidad, belleza o imaginación).
- Entrega de Premios de las tres pruebas clasificatorias (Velocidad, Sumo, Laberinto)
- Entrega de Premio "Robot Destacado".
- *Fin del campeonato y agradecimiento a los asistentes.

Respecto a las Ramas con las que hemos contado: La de UPM, La de la Complutense, la de Sevilla, al menos una Rama de la sección Portuguesa, y ajenos al IEEE contamos con un equipo de ARDE, Y respecto a sponsors del evento: Sección España IEEE, UNED, Cisco, LEGO, Xunta de Galicia.

Y finalmente respecto a medios de comunicación: La Voz de Galicia, Ideal Gallego, La opinión de A Coruña, Televisión Gallega, Tele5 y seguramente Antena3, también la Sexta tiene delegación en Coruña y esperamos que se hubiera pasado en algún momento del certamen.

Podéis consultar todos los premios, pruebas y concursos en la página oficial de la XGN09, en la sección de “Robótica”: www.xuventudegaliza.net.

Estos son algunos de los enlaces en prensa e Internet en los que se puede ver la importancia que ah tenido, para la publicidad de nuestra Rama este evento:

Artículo sobre el Campeonato de robótica que acabamos de organizar en la XGN, visto por Eurpa Press:

<http://www.europapress.es/galicia/a-coruna-00382/noticia-total-1800-jovenes-participan-hoy-domingo-decima-edicion-xuventude-galiza-net-20090403190719.html>

Otro artículo más, esta vez en el correo gallego:

<http://www.elcorreoallego.es/index.php?idMenu=11&idNoticia=412995&idEdicion=1207>

Otro más en la Voz de Galicia:

<http://www.lavozdeg Galicia.es/tecnologia/2009/03/27/00031238155687278427188.htm>

Apariciones en TVE; minutos del 16:00 en adelante de la transmisión:

<http://www.rtve.es/mediateca/videos/20090406/telexornal-informativo-galicia-06-04-09/469879.shtml>

<http://www.rtve.es/mediateca/videos/20090403/telexornal-informativo-galicia-03-04-09/467009.shtml?s1=noticias&s2=informativos-territoriales&s3=telexornal-galicia>

Nota: El nombre correcto es Germán Carro, los de TVE cometieron una errata.

Debido al éxito de participación y público, esperamos poder preparar la segunda edición del certamen para el año que viene y contamos con todos vosotros para ello.

II. PRESENTACIÓN DE LA RAMA EN EL CENTRO ASOCIADO DE TERRASSA

El pasado 11 de Marzo de 2009 la Rama de Estudiantes de la UNED hizo una sesión de presentación en el C. A. de Terrassa, con el objeto de darse a conocer y poder extender los grupos de trabajo a dicho centro.

El acto contó con la participación de Manel Frigola, Profesor de la Universidad Politécnica de Catalunya, que dio una magnífica ponencia sobre el Pasado, Presente y Futuro de la Robótica. También participó Federico González, un tutor del Centro de Terrassa que también está involucrado en la Rama; Carlos E. Jiménez, que nos contó por videoconferencia su punto de vista como ex-alumno de la UNED, ex-miembro de la Rama y actual VP Members del capítulo español del Technology Management Council (TMC) del IEEE, explicando en qué consiste este Council y dando una introducción a la Gestión de la Tecnología, la Ingeniería y la Innovación; y Ramón Carrasco (Moncho), que nos contó también por videoconferencia las actividades en las que la Rama está inmersa últimamente.

Terrassa es el segundo centro más grande de la UNED en cuanto a número de alumnos. En concreto, según cifras facilitadas por la dirección del Centro, el número de alumnos matriculados en Ingenierías, Física o Matemáticas ronda los 1000.

La agenda llevada a cabo fue:

- Presentación de la asociación (Sergio Martín)
- Videoconferencia ex-alumno Terrassa y VP Members del Technology Management Council (TMC) (Carlos Jiménez)
- Videoconferencia Coordinador Actividades (Ramón Carrasco – A Coruña)
- Punto de Vista de un Tutor Terrassa (Federico González)
- Conferencia sobre Robótica (Manel Frigola - Profesor Titular Robòtica i Visió UPC)



Figura 2. Ponencia profesor Manel Frigola



Figura 3. Charla Federico González y Sergio Martín

La acogida tanto de los asistentes (alrededor de 30) como del Centro fue remarcable, ofreciendo este último una total colaboración en todo momento para que sea posible la creación de un grupo de trabajo allí. Actualmente ya está en marcha la organización de un taller de robótica en el centro.

III. ESTRENAMOS ISSN Y NUEVO FORMATO EN EL BOLETÍN ELECTRÓNICO

En la presente edición del Boletín estrenamos el recién adquirido ISSN.

¿No sabes qué es el ISSN?

El ISSN (International Standard Serial Number / Número Internacional Normalizado de Publicaciones Seriadas) es un código numérico reconocido internacionalmente para la identificación de las publicaciones seriadas. El ISSN puede utilizarse siempre que haya que recoger o comunicar información referente a las publicaciones seriadas, evitando el trabajo y posibles errores de transcribir el título o la información bibliográfica pertinente.

Ventajas

- El ISSN permite identificar en todo el mundo, de una forma unívoca y sin ambigüedades, una publicación seriada, cualquiera que sea el país de edición o la lengua de la misma y sin importar que otras publicaciones seriadas lleven un título igual o parecido. Por ello el ISSN es un elemento básico en todos los procesos de información, comunicación, control y gestión referentes a las publicaciones seriadas
- Asegura un medio de identificación preciso e inmediato cuando se hace un pedido.
- Permite un método de comunicación rápido y eficaz entre editores, distribuidores, librerías y agencias de suscripción, mejorando sus circuitos de venta.
- El ISSN puede servir para la construcción de los códigos de barras de las publicaciones seriadas.
- La asignación de un ISSN comporta también la inclusión de los datos de la publicación en la Base de datos internacional del ISSN. Esta base de datos es, por su volumen (más de un millón de registros), cobertura (mundial) y fiabilidad de sus datos, un recurso informativo esencial sobre las publicaciones seriadas.
- En las bibliotecas y centros de documentación facilita las operaciones de identificación, adquisición y préstamo. Asimismo la base de datos del ISSN es la fuente más exhaustiva y autorizada para la catalogación de las publicaciones seriadas.
- Su asignación es completamente gratuita.

Nuevo formato

Además también estrenamos nuevo formato, siguiendo las directrices marcadas por IEEE. De esta manera nos acostumbraremos a utilizar dicho formato y así nos resultará

más sencillo a la hora de publicar en los congresos y revistas internacionales del IEEE.

IV. ¿EN QUÉ ESTAMOS TRABAJANDO ÚLTIMAMENTE?

A continuación os detallamos una lista de los proyectos en los que estamos trabajando últimamente, no están todos los que son, pero sí son todos los que están:

- **Grupo Robótica:** Impartición de cursos y talleres (tanto on-line como presenciales) con el objetivo de crear distintos tipos de robots:
 - Humanoides, creados por nosotros mismos desde cero.
 - SUMO-BOTS: Robots creados para llevar a cabo peleas de sumo
 - Sigue-líneas: Robots capaces de seguir una línea en el suelo gracias a sensores de infrarrojos
 - Además organización de concursos de robótica tanto a nivel nacional como internacional
- **Proyecto SSETI:** Participación en un proyecto de la Agencia Aeroespacial Europea (ESA). Proyecto de comunicaciones de un satélite nodriza con dos subsatélites de investigación y de éstos con el satélite nodriza, y del satélite nodriza con Tierra a través de una red de comunicaciones de seguimiento de radioaficionados. Podéis consultar una entrevista realizada al coordinador de este proyecto; Germán Carro; en este enlace: <http://www.vieiros.com/nova/73271/galegos-na-carreira-espacial>
La entrevista está en gallego, pero es lo suficientemente comprensible como para conocer los entresijos del proyecto.
- **Grupo de Programación:** Impartición de cursos y talleres de programación como en Java (tanto on-line como presenciales).
- **Grupo de Redes:** Impartición de cursos y talleres de Redes Inalámbricas.
- **Grupo de Software Libre:** Todo sobre Linux y herramientas de software libre tienen cabida en este grupo.
- **Grupo de Diseño Web:** Impartición de cursos sobre diseño Web, utilizando plataformas como Joomla.
- **Publicaciones:** Posibilidad de publicar en nuestro Boletín Electrónico interno o en otras revistas y conferencias nacionales o internacionales.

Servicios Sensibles al Contexto Orientados al Aprendizaje Móvil

Sergio Martín

Dep. Ingeniería Eléctrica, Electrónica y de Control
 Universidad Nacional de Educación a Distancia
 Madrid, España
 sergio.martin@ieee.org

Abstract—Las tecnologías de localización están convirtiéndose en una parte fundamental del paradigma de Aprendizaje Móvil, también conocido como M-Learning, debido a que permite a los estudiantes trabajar y colaborar en entornos fuera de las aulas. Además, están apareciendo nuevas arquitecturas y sistemas que ofrecen una información mejor y más personalizada a las necesidades de cada usuario, gracias a un mayor conocimiento de su contexto. El presente artículo muestra un sistema orientado al M-Learning que permite crear interoperabilidad entre distintas tecnologías de localización (GPS, redes de telefonía móvil, WLAN, RFID, NFC, etc.) de una manera transparente para las aplicaciones de alto nivel y usuarios.

Keywords: *context-awareness; location-based technologies, M-Learning*

I. INTRODUCCIÓN

Las tecnologías de mayor calado en la sociedad son las que pasan inadvertidas, adelantó hace más de una década el científico estadounidense Mark Weiser. En este sentido, una nueva tendencia conocida como inteligencia ambiental proclama la creación de servicios informáticos y telemáticos basados en la captura de información del contexto (preferencias, localización, histórico, condiciones ambientales, etc.) del usuario de manera transparente y automática. Estos servicios pueden ser integrados en el ámbito educativo de manera satisfactoria mejorando la atención proporcionada al estudiante mediante la creación de aplicaciones más conscientes de las necesidades de la persona, reaccionando de manera inteligente y autónoma a las mismas, mejorando así la experiencia del estudiante dentro del ámbito educativo.

Como ejemplo de las capacidades que adquiere un sistema consciente del contexto se podría imaginar una aplicación en una universidad que supiera quién es cada persona, de manera que cuando el profesor llegara al aula, el sistema ya habría encendido el ordenador, habría preparado las transparencias y se las habría pasado a los terminales de los alumnos. Otro ejemplo de aplicación podría ser que cuando un alumno entrara en una sala, por ejemplo, una biblioteca, el sistema detectara quién es el estudiante y de qué asignaturas está matriculado, y sabiendo que está en una biblioteca, le ofreciera el catálogo de

libros disponibles para poder realizar una búsqueda “inteligente”, acotando los resultados únicamente a las materias en las que el estudiante tiene interés.

El desarrollo de este tipo de aplicaciones es técnicamente posible con las actuales tecnologías (GPS, Wi-Fi, 3G, RFID, etc.), por lo que poco a poco irán calando más y más dentro de una sociedad cada vez más acostumbrada a fusionarse con sistemas electrónicos.

II. OBJETIVO

Todas estas tecnologías dentro de un dispositivo tan pequeño y portable como son hoy en día los teléfonos móviles están dando lugar a una nueva generación de aplicaciones en todo tipo de entornos, que permiten el aprendizaje y la colaboración en cualquier momento, cualquier lugar y con cualquier tipo de dispositivo.

Los sistemas basados en localización pueden aportar un interesante valor añadido a estas aplicaciones de M-Learning, debido a que crean una forma totalmente nueva de interacción entre los estudiantes y el entorno. Por ejemplo, un estudiante en una cafetería tiene diferentes necesidades que si está en un laboratorio o en la secretaría de la escuela; o un profesor en un aula necesitará información y servicios diferentes que si está en su despacho. Sabiendo donde está el usuario en cada momento es posible ofrecer aprendizaje personalizado a través de dispositivos móviles dependiendo no solo de su perfil, sino también del lugar donde se encuentre y el momento en el que se lleve a cabo la acción.

III. TECNOLOGÍAS DE LOCALIZACIÓN

Es un hecho que el principal y más extendido método de localización hoy en día es GPS (*Global Positioning System*). Esta tecnología ofrece una exactitud bastante razonable en exteriores. Su funcionamiento consiste en que cada satélite transmite ciertos datos que indican su situación (localización espacial) y la hora actual. Las señales, moviéndose a la velocidad de la luz llegan al receptor del GPS con unas pequeñas diferencias de tiempos debido a que unos satélites están más lejos que otros (Figura 1). La distancia hasta los

satélites puede ser determinada estimando el tiempo que les toma a las señales alcanzar el receptor. Una vez que determine la posición de al menos cuatro satélites GPS será capaz de calcular su posición en tres dimensiones.

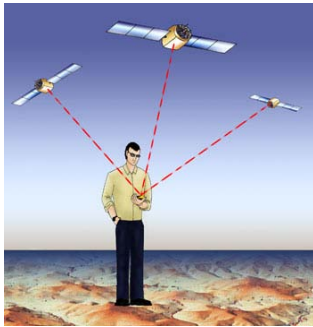


Figura 1. Receptor GPS determinando distancias a los satélites para estimar la posición en la superficie terrestre.

Sin embargo, el uso de esta tecnología dentro de edificios acarrea mayores dificultades debido a que el receptor necesita tener contacto directo con los satélites. Por esa razón, están surgiendo tecnologías como Wi-Fi o RFID (Identificación por Radiofrecuencia) como sistemas de localización para interiores. En la localización basada en Wi-Fi el dispositivo móvil recolecta los diferentes niveles de ruido y potencia que el punto de acceso Wi-Fi emite en el entorno cada pocos milisegundos (al menos tres puntos de acceso son requeridos para poder triangular) (Figura 2).



Figura 2. Mapa de cobertura Wi-Fi en un área con 5 puntos de acceso.

Esta información es procesada para obtener las coordenadas del usuario [1].

Esta misma filosofía es aplicada con las torres de telefonía móvil en lugar de con Wi-Fi [2]. Este es el caso de dispositivos como en nuevo iPod o los nuevos modelos de HTC que están revolucionando el mercado de dispositivos móviles gracias a su

gran interfaz y a sus capacidades de localización y conectividad.

Por otro lado, RFID (*Radio Frequency IDentification*, en español Identificación por radiofrecuencia), es un sistema de almacenamiento y recuperación de datos remoto que usa dispositivos denominados etiquetas, transpondedores o *tags* RFID (Figuras 3). El propósito fundamental de la tecnología RFID es transmitir la identidad de un objeto (similar a un número de serie único) mediante ondas de radio. Las tecnologías RFID se agrupan dentro de las denominadas

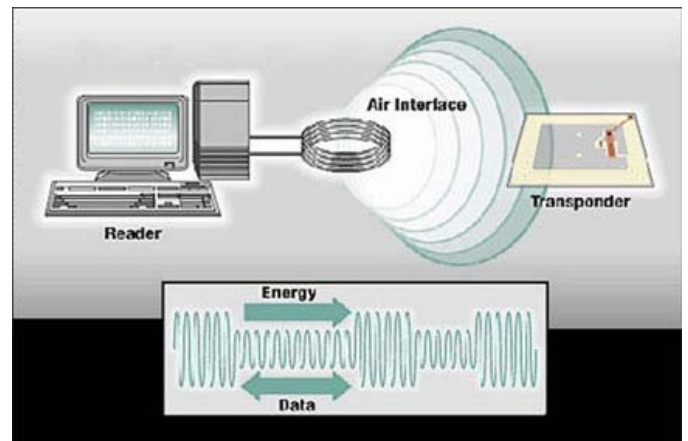


Figura 3. Lector RFID incorporado en un teléfono móvil Nokia leyendo información de etiquetas RFID.

Una etiqueta RFID es un dispositivo pequeño, similar a una pegatina, que puede ser adherida o incorporada a un producto, animal o persona. Contienen antenas para permitirles recibir y responder a peticiones por radiofrecuencia desde un emisor-receptor RFID. Las etiquetas pasivas no necesitan alimentación eléctrica interna, mientras que las activas sí lo requieren. Una de las ventajas del uso de radiofrecuencia (en lugar, por ejemplo, de infrarrojos) es que no se requiere visión directa entre emisor y receptor.

Dependiendo de las frecuencias utilizadas en los sistemas RFID, el coste, el alcance y las aplicaciones son diferentes. Los sistemas que emplean frecuencias bajas tienen igualmente costes bajos, pero también baja distancia de uso. Los que emplean frecuencias más altas proporcionan distancias mayores de lectura y velocidades de lectura más rápidas. Así, las de baja frecuencia se utilizan comúnmente para la identificación de animales, seguimiento de barricas de cerveza, o como llave de automóviles con sistema antirrobo. En ocasiones se insertan en pequeños chips en mascotas, para que puedan ser devueltas a su dueño en caso de pérdida. Las etiquetas RFID de alta frecuencia se utilizan en bibliotecas y seguimiento de libros, seguimiento de palés, control de acceso en edificios, seguimiento de equipaje en aerolíneas, seguimiento de artículos de ropa y ahora último en pacientes de centros hospitalarios para hacer un seguimiento de su historia clínica. Un uso extendido de las etiquetas de alta frecuencia como identificación de acreditaciones, substituyendo a las anteriores tarjetas de banda magnética. Sólo es necesario

acercar estas etiquetas a un lector para autenticar al portador [3].

Esta identificación es normalmente empleada en la identificación, como sustituto del código de barras, realizando el seguimiento de productos. Aunque también puede realizar identificación de personas dentro de un área que portan uno de las etiquetas RFID.

IV. M-LEARNING BASADO EN LOCALIZACIÓN

Obviamente, los servicios basados en localización son una parte fundamental del paradigma M-Learning [Saha, D. et al] creando un nuevo abanico de aplicaciones a integrar en el entorno educativo. Como ejemplos de estas aplicaciones, los autores han estado trabajando los últimos años en varios prototipos. El primero de ellos proporcionaba información personalizada a los estudiantes dentro de un edificio, detectando la habitación en la que se encontraba en cada momento y ofreciendo servicios usando su información contextual [5]. En este sentido, si un estudiante entra en la biblioteca el sistema detectará este evento, y recolectará información de su contexto a cerca de las asignaturas en las que está matriculado y su bibliografía asociada, filtrando de manera automática los resultados de las búsquedas que este usuario realice en el catálogo (Figura 4).

En este campo, la mayor contribución es la integración con plataformas de e-learning como dotLRN (una plataforma de código abierto basada en OpenACS).

Otras aplicaciones desarrolladas están más orientadas al aprendizaje informal en entornos culturales, como por ejemplo museos o lugares históricos [6].



Figura 6. Mapa con el usuario localizado en un despacho y la aplicación de búsqueda bibliográfica que utiliza el contexto educativo del alumno.

En este caso, el sistema ofrecería información a cerca de las obras de arte con solo acercarse a las mismas.

V. OTRAS SERVICIOS DE M-LEARNING

Por otro lado, actualmente, muchos servicios educativos tradicionalmente asociados con el e-learning, como son los laboratorios virtuales o remotos (es decir, la realización de prácticas a distancia a través de laboratorios software, o bien

hardware manipulados a distancia y con una cámara Web para controlar su comportamiento), están siendo también trasladados al mundo móvil.

Esto supone un gran avance, ya que permite el acceso a material didáctico destinado a mejorar las actitudes prácticas de los alumnos no solo en cualquier momento, sino también en cualquier lugar y con casi cualquier dispositivo. Generalmente, el único requisito es que el dispositivo móvil tenga un navegador Web que sea capaz de ejecutar Javascript o Flash, ya que así es cómo están implementados la mayoría de los interfaces de este tipo de laboratorios.

Como ejemplo de laboratorios empleados por los autores podríamos citar un laboratorio remoto de FPGAs y otro de Microcontroladores. A estos laboratorios, los estudiantes pueden acceder tanto a través de ordenadores de sobremesa como a través de dispositivos móviles, según sus circunstancias. Esta es la ventaja del empleo de las tecnologías Web en el desarrollo de interfaces para los laboratorios: una mayor versatilidad.

VI. CONCLUSIONES

Las aplicaciones sensibles al entorno y al contexto son el futuro de la computación, ya que permiten ofrecer un servicio mejor a los usuarios. Dichas aplicaciones se basan en parte en el concepto "Internet of Things", que promulga un cierto grado de inteligencia en todos los objetos que nos rodean, para lo cual el usuario debe estar dotado de nuevos "sentidos" artificiales que le permitan relacionarse con el nuevo entorno [7]. Siguiendo esta idea, el entorno propuesto detecta quién es el usuario, donde está, donde ha estado, las condiciones ambientales del entorno (clima, temperatura, previsión futura, etc.), así cómo que otros individuos hay en el mismo entorno, de manera que pueda adaptarse de manera inteligente a las necesidades de cada momento.

Gracias al desarrollo de este tipo de sistemas, una aplicación basada en localización, como por ejemplo un navegador GPS, podrá seguir dando servicio aunque ya no reciba información de satélites GPS, ya que, de manera transparente, empezará a utilizar la información recogida por otros sensores (RFID, GSM, UHF, WLAN, etc.).

En cuanto al M-Learning, estas tecnologías están suponiendo una verdadera revolución en la forma en que los estudiantes aprenden, tanto de una manera formal como informal, y soportando trabajo colaborativo.

REFERENCES

- [1] Borenovic, M.N.; Simic, M.I.; Neskovic, A.M.; Petrovic, M.M., 2005. Enhanced Cell-ID + TA GSM Positioning Technique. *The International Conference on Computer as a Tool, 2005. EUROCON 2005*. Volume 2, pp. 1176 – 1179.
- [2] Youssef, M.A.; Agrawala, A.; Udaya Shankar, A., 2003. WLAN location determination via clustering and probability distributions. *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications (PerCom 2003)*, pp.143-150.
- [3] Su, W.; Lee, S.-J.; Gerla, M., 2000. Mobility prediction in wireless networks. *21st Century Military Communications Conference (MILCOM 2000)*. Volume 1, 22-25 Oct. 2000 pp. 491 – 495.

- [4] Polito, S.; Biondo, D.; Iera, A.; Mattei, M.; Molinaro, A., 2007. Performance Evaluation of Active RFID Location Systems based on RF Power Measures. *IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications, 2007 (PIMRC 2007)*. pp. 1-5.
- [5] Martín, S., Castro, M., Gil, R., Colmenar A., Peire J., 2006. Ubiquitous and biometric applications on distance education. An alternative to the traditional examination. *I International Conference on Ubiquitous Computing: Applications, Technology and Social Issues*, Alcalá de Henares (Spain) pp. 39-42.
- [6] Castro, M., Gil, R., Martin, S., Peire, J., "New Project on Secure Education Services for On-Line Learning". San Juan (Puerto Rico): 9th International Conference on Engineering Education. 2006.
- [7] Saha, D., and Mukherjee A., "Pervasive Computing: A Paradigm for the 21st Century", IEEE Computer Society, pp. 25-31. 2003.

ANDROID

Plataforma libre para desarrollo de aplicaciones para móviles

Pablo Losada de Dios

Dept. de Ingeniería Eléctrica, Electrónica y de Control
E.T.S.I. Industriales
UNED
Madrid, Spain

Nuria Oliva Alonso

Dept. de Ingeniería Eléctrica, Electrónica y de Control
E.T.S.I. Industriales
UNED
Madrid, Spain

Abstract — Este documento quiere presentar una introducción de la plataforma libre de desarrollo de aplicaciones para dispositivos móviles. Se trata de describir a grandes rasgos los aspectos más destacables de la plataforma y por que puede convertirse en un periodo años en la plataforma de referencia en el mundo de la telefonía móvil

Plataforma, Desarrollo, aplicaciones móvil, android, Google, SDK.

I. INTRODUCCIÓN

Siempre es saludable que en cualquier entorno informático o de comunicaciones se incorpore como alternativa a los desarrolladores una plataforma que permite desarrollar software con código libre para teléfonos y dispositivos móviles. Y lo más interesante, desde su presentación hace más de dos años, es que ya está consolidada y empieza a tener interés real y no como un mero proyecto dentro de las empresas del sector de las comunicaciones e Internet.

Este artículo pretende introducir esta tecnología, adentrarse en sus fundamentos y sobre todo mostrar en que estado se encuentra su implantación y que posibilidades brinda al desarrollador de aplicaciones para dispositivos móviles.

II. ¿QUÉ ES ANDROID?

Android es una plataforma para aplicaciones móviles, totalmente abierta tanto para los fabricantes de dispositivos móviles como para los desarrolladores de aplicaciones que funcionan en tales dispositivos. Android incluye básicamente un sistema operativo, middleware y aplicaciones base.



Figura 1. Logo de Android.

Conviene aclarar un par de conceptos que pueden ayudar a entender que es Android. Para empezar, un sistema operativo son un conjunto de programas destinados a permitir la

administración de los recursos de un dispositivo. Su finalidad es gestionar el hardware de la máquina desde los niveles más básicos, permitiendo también la interacción con el usuario. Middleware, es un tipo de software que se sitúa entre el sistema operativo y las funciones de red del dispositivo en sí, proporcionando una Interfaz de Programación de Aplicaciones, API, que facilita la programación y el manejo de las aplicaciones distribuidas.

Inicialmente desarrollado por Google y luego por la Open Handset Alliance. Android permite a los desarrolladores escribir código gestionado en lenguaje de programación Java y controlar los dispositivos por medio de bibliotecas desarrolladas o adaptadas por Google.

La plataforma está apoyada por la alianza Open Handset Alliance [1], un consorcio de 48 compañías de hardware, software y telecomunicaciones comprometidas con la promoción de estándares abiertos para dispositivos móviles. Estas compañías distribuirán móviles basados en esta plataforma con el objetivo final de estandarizar, mejorar y abaratar el uso y desarrollo de aplicaciones para móviles.

Forman parte de esta alianza compañías como:

- Operadora: Telefónica, Telecom, Chine Mobile...
- Fabricantes: HTC Corporation, LG, Motorola y Samsung
- Desarrollo de software: Ascender Corp., Ebay, Google...
- Semiconductores: NVIDIA, Intel, Audience, Broadcom...
- Comercialización: Aplix Corporation, Wind River, TAT...

El núcleo de Android es Linux kernel 2.6. A este nivel están implementados todos los drivers de los dispositivos primarios, como son cámara, puertos USB, teclado, etc. El fabricante de un dispositivo puede agregar sus propios drivers.

La empresa Google ha publicado el Kit de Desarrollo de Software (SDK), que provee de herramientas y API's necesarios para que expertos, o simples usuarios con ganas de aprender nuevas tecnologías, puedan empezar a desarrollar

aplicaciones en la plataforma Android, usando el lenguaje de programación Java.

III. CARACTERÍSTICAS DE LA PLATAFORMA

Las características que definen esta plataforma y que permiten vislumbrar sus posibilidades son las siguientes:

- La plataforma Android viene con una variedad de aplicaciones escritas usando el lenguaje de programación Java [2]:
 - *Página de inicio*, muestra las aplicaciones, widgets, y accesos directos. También apoya personalizable de papel de pared.
 - *Teléfono*, admite regular las funciones de telefonía, así como controles de llamada, llamadas de conferencia, los servicios suplementarios, y fácil integración con los contactos.
 - *Web Browser*, es un completo navegador basado en WebKit que soporte HTML y XHTML.
 - *Correo electrónico*, proporciona acceso a servidores de correo electrónico se encuentran en Internet y apoya POP3, IMAP4 y SMTP.
 - *Media Player*, permite la gestión, la importación, y la reproducción de contenido que se ha codificado en diversas formas.
 - Alarma, Calculadora, Calendario, Cámara, Contactos, mensajería instantánea, MMS, Ajustes, Marcador de voz, y muchas otras aplicaciones se incluyen también en la versión 1.0.
- Android posee un *Framework* o marco de aplicaciones que permite la reutilización y reemplazo de componentes, típico en los lenguajes de programación Orientados a Objeto como Java. La arquitectura de las aplicaciones está diseñada para simplificar la reutilización de componentes, cualquier aplicación puede publicar sus funcionalidades permitiendo a cualquier otra aplicación a hacer uso de ellas (sujeto a reglas de seguridad impuestas por el *framework*). El mismo mecanismo permite a los componentes ser reemplazados por el usuario. Las más destacables son:
 - *Administrador de Notificaciones*, que permite a todas las aplicaciones mostrar alertas personalizables en la barra de estado
 - *Administrador de Actividades*, que maneja el ciclo de vida de las aplicaciones y provee un comportamiento común en la navegación, manteniendo un *Backstack* para que el usuario pueda utilizar unas aplicaciones mientras otras siguen ejecutándose.
- *Administrador de Recursos*, que provee acceso a recursos como cadenas, gráficos, y archivos. Controlador de Acceso a datos guardados de otras aplicaciones
- Un completo y extensible conjunto de vistas que pueden ser utilizadas para desarrollar una aplicación: listas, grillas, cajas de texto, botones e incluso un web browser
- Contiene la máquina virtual Dalvik, diseñada por Dan Bornstein con contribuciones de otros ingenieros de Google. Una máquina virtual es un software que emula a un ordenador y puede ejecutar programas como si fuera un ordenador real. Dalvik está optimizada para su uso en móviles y por ello requiere poca memoria y está diseñada para permitir ejecutar varias instancias de la máquina virtual simultáneamente, delegando el control y la gestión de memoria al sistema operativo subyacente.
- También posee un navegador integrado, basado en el motor de código abierto WebKit, que es un marco para las aplicaciones que facilita a los desarrolladores incluir gran parte de las funcionalidades de Safari, que es el navegador web desarrollado por Apple que incluye en sus propias aplicaciones.
- Gráficos optimizados, con una librería de gráficos 2D y gráficos 3D basados en la especificación estándar OpenGL (Librería de Gráficos Abierta), que define una API multilenguaje y multiplataforma para escribir aplicaciones que produzcan dichos gráficos.
- Sistema Gestor de Base de Datos SQLite, para almacenamiento de datos estructurados. Sus principales características:
 - Sistema de base de datos relacional compatible con ACID (transacciones seguras)
 - Contenida en una pequeña librería en C.
 - Se integra en la aplicación (no es un proceso independiente, ni diferencia entre cliente-servidor).
 - Todos los datos se almacenan en un único fichero.
- Soporte para medios con formatos comunes de audio, video e imágenes (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF).
- Telefonía GSM cuatribanda
- Tecnologías de conectividad mediante USB 2.0, Bluetooth 2.0 EDR, EDGE (o evolución del GPRS), 3.5G y 802.11 b / g WiFi .
- Cámara, GPS (Sistema de Posicionamiento Global), brújula y acelerómetro.

- Además, posee un entorno de desarrollo que incluye un emulador de dispositivo, herramientas para depurar, perfiles de memoria y performance, y un Plugin para el Entorno de Desarrollo Integrado (IDE) Eclipse.
- En cuanto al Hardware soportado por la plataforma, la plataforma funcionará en casi cualquier ARM basado en kernel de Linux. Exige un mínimo de 128 MB de RAM y 256 MB de memoria Flash. Si es necesario se puede apoyar en unidades de memoria Flash para poder descargar más aplicaciones de terceros para dispositivos de usuario.

IV. ARQUITECTURA

Android presenta una arquitectura de capas que se van apoyando en la inmediatamente inferior y que queda reflejado en el siguiente esquema [3]:



Figura 2. Arquitectura.

Como se ha comentado antes en la base de la arquitectura de la plataforma se encuentra el núcleo Linux, en concreto el kernel 2.6. A este nivel están implementados todos los drivers de los dispositivos primarios, como son cámara, puertos USB, teclado, vídeo, comunicación WIFI y Bluetooth, audio. El fabricante de un dispositivo puede también agregar sus propios drivers, como por ejemplo un driver para una pantalla táctil al estilo de iPhone o HTC touch.

Sobre este núcleo están las Librerías Nativas, que ofrecen funcionalidad básica al subsistema, como son SQLite (base de datos integrada con soporte para SQL), OpenGL ES (gráficos avanzados en 3D), SSL (cifrado de datos), CODECs de medios (audio, video), etc.

Compartiendo toda esta infraestructura vemos la Máquina Virtual Dalvik, que es en donde ejecutan las aplicaciones programadas.

Cada aplicación Android corre su propio proceso, con su propia instancia de la máquina virtual Dalvik. Dalvik ha sido escrito de forma que un dispositivo puede ejecutar múltiples

máquinas virtuales de forma más eficiente. La Máquina Virtual está basada en registros, y ejecuta clases compiladas por el compilador de Java que han sido transformadas al formato .dex por la herramienta incluida en la plataforma denominada DX.

Como se ha comentado, los ejecutables no son clases de Java, sino que las clases de Java son post-procesadas a un formato llamado .dex, el cual está optimizado para que tenga un uso de memoria mínimo.

Esta Máquina Virtual tiene acceso tanto a las *Core Libraries* (que son básicamente las librerías de funciones estándar de la plataforma Java, como son objetos *Object*, *String*, etc). En otras palabras, el equivalente a un JDK en Java.

Por otro lado, la Máquina Virtual tiene acceso a lo que se ha denominado *Application Framework* (o un Marco de Trabajo para Aplicaciones), que contienen gran número de módulos escritos en Java para el consumo de las aplicaciones. El *Framework* para el desarrollo de aplicaciones está dividido, como se ha comentado antes, en subsistemas para gestión del sistema como son los administradores de Notificaciones, actividades, recursos y Telefonía. También dentro de estos subsistemas se encuentran el Servicio de Vistas que es un conjunto de elementos gráficos para crear interfaces o el Administrador de ventanas para realizar el control de las diversas ventanas de una aplicación.

El nivel superior a todos estos niveles, son las Aplicaciones que ejecutan sobre toda esta infraestructura. Todas estas aplicaciones son 100% lenguaje Java y se puede encontrar un Navegador, una agenda de contactos, un cliente de correo y una herramienta de gestión del teléfono entre otras.

V. ANATOMÍA DE UNA APLICACIÓN

De una manera simplificada, una aplicación Android podría ser como una colección de componentes de varios tipos. Estos componentes en su mayoría tienen un bajo nivel de acoplamiento entre ellos, de tal manera, que una aplicación podría ser como una federación de todos ellos.

Por lo general, todos estos componentes se ejecutan en el mismo proceso del sistema. Se pueden crear múltiples *threads* o hilos dentro de tal proceso, y además es posible crear procesos hijos completamente separados del proceso padre, si se necesitan. En cuanto a esto último, Android trata, en lo posible, de evitarlo y conseguir procesos con un código lo más transparente posible.

Estas son las partes más importantes de las APIs (Application Programming Interface) de Android [4]:

A. *AndroidManifest.xml*

El archivo "AndroidManifest.xml" es el archivo de control que le dice al sistema qué es lo que debe hacer con los componentes de más alto nivel que se han creado (específicamente "Actividad", "Servicio", "Receptor de Intentos" y "Proveedor de Contenidos" descritos a continuación). Debido a esto, este archivo es considerado el "pegamento" que indica qué Intentos serán recibidos por las Actividades. Una vez que se han seleccionado los componentes

que se van a utilizar en la aplicación, es necesario que se listen en el "AndroidManifest.xml".

B. Actividades

Una Actividad o *Activity* es, fundamentalmente, un objeto que tiene un ciclo de vida. Una *Activity* es un trozo de código que desarrolla algún trabajo en la aplicación. Existen *Activities* que despliegan un UI (*User Interface*) al usuario y algunas *Activities* que nunca despliegan una UI. En general, se designará una *Activity* como el punto de inicio de la aplicación.

C. Interfaz de Usuario (UI) y las Vistas

Una actividad hace multitud de cosas, pero ella misma no presenta nada en la pantalla. Para conseguir que aparezca algo en la pantalla es necesario diseñar el UI, con *Views* y *Viewgroups*, que son las clases que se usan para crear la interfaz entre el usuario y la plataforma Android.

Una Vista o *View* es un objeto que sabe cómo dibujarse asimismo en la pantalla. Las UI o interfaces de usuario de Android están conformadas por un árbol de *Views* o Vistas.

Las *Views* son estructuras de datos cuyas propiedades contienen los datos de la capa y la información específica del área rectangular de la pantalla. Una Vista tiene: layout, drawing, focus change, scrolling, etc..

Un *Viewgroup* es un objeto especial de *View* cuya función es contener y controlar la lista de *views* y de otros *viewgroups*. Los *viewgroups* permiten añadir estructuras a la interfaz y acumular complejos elementos en la pantalla.

En la plataforma Android se define una *Activity* del UI usando un árbol de nodos *view*'y *viewgroups*, como se puede ver en la imagen de abajo.

partir de ese momento cada nodo *viewgroup* del árbol es responsable de pintar sus hijos directos.

El árbol puede ser tan simple o complejo como se necesite y se puede desarrollar usando como ayuda los *widgets* y *layouts* que Android proporciona.

Los *Widgets* son un objeto *View* que ayuda a definir al desarrollador el Interfaz de Usuario ya que existen muchos pre-implimentados como son Botones, casillas de selección, campos de texto o incluso más complejos como Relojes, calendarios, controles de zoom, etc..

La mejor forma de expresar la jerarquía de vistas de una aplicación Android es a través de la posibilidad de definir *Layouts* mediante ficheros XML. Cada elemento XML puede ser un *view* o *viewgroup*. Android proporciona una serie de *Layouts* predefinidos para ayudar al desarrollo de UI como pueden ser *LinearLayout*, *RelativeLayout*, *AbsoluteLayout*, *TableLayout*, *GridLayout*, etc..

Por últimos, en el UI de la aplicación se definirán los eventos de UI para poder interactuar con el usuario. Básicamente se trata de definir a que eventos de usuario van a reaccionar cada uno de los elementos o *Widgets* del UI.

D. Intentos

Un *Intent* es un simple mensaje que representa "la intención" de hacer algo. Por ejemplo, si una aplicación quiere abrir una página web, debe expresar su "Intento" para ver la URL a través de la creación de una instancia de *Intent* y luego entregársela al sistema. Posteriormente, el sistema ubicará una pieza de código que sepa cómo manipular aquel "Intento" y la ejecuta (en este caso un browser). Los *Intent* también pueden ser usados para difundir eventos interesantes a través de todo el sistema (tales como notificaciones).

E. Servicios

Un Servicio es un cuerpo de código que se ejecuta en segundo plano. Se ejecuta dentro de su propio proceso o en el contexto del proceso de otra aplicación, dependiendo de sus necesidades. Otros componentes se unen a un Servicio e invocan sus métodos a través de los llamados Controles de Procedimiento Remoto (RPC)".

F. Notificaciones

Una notificación es un pequeño ícono que aparece en la barra de estado. Los usuarios pueden interactuar con este ícono para recibir información. Las notificaciones más conocidas son los mensajes SMS, correo de voz; pero las aplicaciones pueden crear sus propias notificaciones. Las notificaciones son el mecanismo preferido para alertar al usuario que algo requiere de su atención.

G. ContentProvider

Un Proveedor de Contenidos o *ContentProvider* es un almacén de datos que provee acceso a datos en el dispositivo. Un ejemplo clásico es el *ContentProvider* que es usado para tomar datos desde la lista de contactos. Las aplicaciones pueden utilizar datos que otras aplicaciones han expuesto vía un *ContentProvider*. También se pueden definir *ContentProvider* específicos para exponer datos propios de una aplicación.

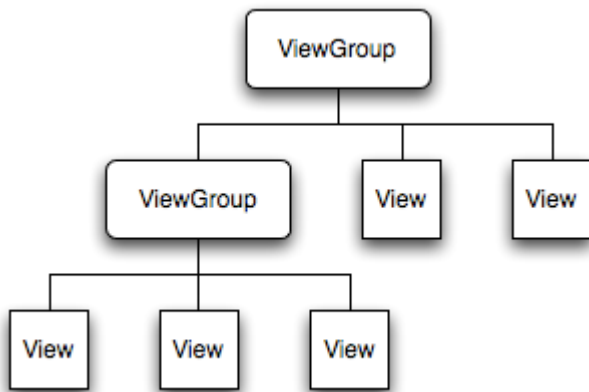


Figura 3. Árbol de nodos.

Para añadir el árbol a la pantalla, la Actividad llama al método *setContentview()* y pasa una referencia al objeto nodo principal. Una vez que el sistema Android ha referenciado el objeto nodo principal ya puede trabajar directamente con el nodo para anular, medir y dibujar el árbol. Cuando la Actividad está activa y recibe el foco, el sistema lo notifica y le pide al nodo principal medidas y dibuja el árbol. El nodo principal entonces pide que sus nodos hijos se dibujen a sí mismos, a

VI. BLOQUES DE CONSTRUCCIÓN DE UNA APLICACIÓN ANDROID

Como se ha adelantado en el punto anterior, en cualquier aplicación Android se van a utilizar alguno, varios o los cuatro bloques de construcción que se pasan a describir con algo más de profundidad:

- Activity
- Intent (Receiver)
- Service
- Content Provider

Activity es básicamente la parte visual de una aplicación. Una "actividad" es el bloque más usado en las aplicaciones Android. Generalmente, una "actividad" es una pantalla individual en cada aplicación. Cada "actividad" se implementa como una clase que hereda de *Activity*, lo que hará que la clase despliegue un Interfaz de Usuario (UI) compuesto de *Views* y responda a eventos. Lo común es que una aplicación consista de múltiples pantallas.

Por ejemplo, una aplicación de mensajería podría usar una pantalla para mostrar el listado de contactos y en una segunda pantalla para escribir el mensaje al contacto seleccionado y otras pantallas para ver cambiar la configuración. Cada una de estas pantallas debe ser implementada como una *Activity*.

Android usa una clase especial llamada *Intent* para moverse de una pantalla a otra. Un *Intent* describe lo que una aplicación desea hacer. Las dos partes más importantes de la estructura de datos de un *Intent* son la acción y los datos sobre los cuales se actuará. Los valores típicos para la acción son *MAIN*, *VIEW*, *PICK*, *EDIT*, etc. Por ejemplo, para ver la información de contacto de una persona podría ser necesario crear un *Intent* con la acción *VIEW* y los datos definidos como una URI (Uniform Resource Identifier) que representa a esa persona.

Se puede utilizar la clase *IntentReceiver* cuando se quiera programar una aplicación para que se ejecute como respuesta a un evento.

Intent Receiver es un bloque importante en una aplicación por los siguientes aspectos:

- Los *IntentReceiver* no despliegan un UI, sin embargo ellos podrían utilizar el Administrador de Notificaciones para avisar al usuario que algo interesante está pasando.
- No es necesario que esté ejecutándose una aplicación para que sus *IntentReceivers* puedan ser llamados. Si una aplicación no estuviera ejecutándose, el sistema la puede activar cuando uno de sus *IntentReceivers* sea activado.
- Las aplicaciones también pueden transmitir sus propios *Intents* a otras aplicaciones

Service es una aplicación que se mantiene activada por un largo tiempo y no despliega un UI. Un ejemplo es un programa que reproduce archivos mp3 desde una lista de música, mientras el usuario realiza otras actividades. En este caso, el

programa podría iniciar un servicio y de esa forma reproducir la música sin necesidad de usar la pantalla.

El sistema mantendrá el servicio ejecutándose hasta que finalice. Es posible conectarse a Servicio y activarlo si no estuviera ejecutándose. Una vez que la aplicación está conectada al servicio, la comunicación entre la aplicación y el servicio se realiza a través de la interfaz que el Servicio dispone. Para el ejemplo de un reproductor mp3, está interfaz podría permitir hacer pausa o saltar a una nueva canción.

Content Provider, es un servicio que le ofrece la capacidad a las aplicaciones de comunicarse con otras de manera interna a Android. Las aplicaciones podrían almacenar sus datos en archivos, una base de datos "SQLite" o cualquier otro mecanismo. Sin embargo, un *Content Provider* es de utilidad cuando los datos de tu aplicación deben ser compartidos con otras aplicaciones. *Content Provider* es una clase de Java que implementa un conjunto estándar de métodos para que otras aplicaciones almacenen o recuperen el tipo de datos que el *Content Provider* manipula.

Android trae un paquete de *ContentProvider* con tipo de contenidos comunes como Audio, imágenes, vídeos o contactos.

Cada objeto alojado en un *ContentProvider* tiene un identificado un ID único. Cada búsqueda o *Query* devuelve un objeto cursor que se puede mover de ítem en ítem y dentro de ellos de campo a campo para obtener la información buscada de cada uno de ellos.

Cada *ContentProvider* puede estar compuesto de numerosas tablas de datos, cada una de ellas identificada con una URI.

Se necesitan tres aspectos para realizar búsquedas en un *ContentProvider*:

- La URI que identifica la Tabla dentro del Provider.
- Los nombres de los campos que se quieren recibir
- Los tipos de datos de esos campos

VII. KIT DE DESARROLLO (SDK)

En la actualidad Google tiene publicado el SDK versión 1.1.

Este paquete o kit de desarrollo incluye las APIs y herramientas necesarias para desarrollar las aplicaciones utilizando JAVA como lenguaje de programación y testear el código, respectivamente. Las librerías adjuntas son compatibles con los siguientes entornos de desarrollo: Eclipse, JDK5 o JDK6, Android Development Tool Plugin y Apache Ant.

Google ofrece tres versiones del SDK, una para Windows, una para Mac OSX (Intel) y una para Linux (x86).

Destacar que la manera de crear aplicaciones Android de forma más eficiente y rápida es a través del Plugin para Eclipse ADT. Permite crear aplicaciones y depurarlas fácil y rápidamente.

Este Plugin incluye un asistente para empezar con nuevos proyectos que ayuda a crear los ficheros básicos para comenzar el desarrollo.

También incluye un editor de código Android que sirve de ayuda para escribir archivos con código XML válido para crear el archivo *Manifest.xml* y otros recursos.

Por último se incluyen una serie de herramientas gráficas para probar las aplicaciones según se van desarrollando y entre las que se encuentra un emulador de teléfono Android que permite visualizar el funcionamiento de la aplicación de forma muy real sin necesidad de cargarla en un teléfono.

VIII. CONCLUSIONES. ¿QUÉ PUEDE SUPONER ANDROID?

Es muy probable el éxito definitivo de Android dependerá de la creatividad de los dispositivos que se construyan sobre él, y no se podrá juzgar hasta que dichos dispositivos estén ampliamente implantados en el mercado.

Esto es un reto para Microsoft, y sobre todo para Symbian, que vienen luchando por lograr que los desarrolladores trabajen con sus API nativas, que son muy complicadas. Cuanta más actividad de programación absorba Android, más les costará a las otras plataformas conseguir el apoyo de los desarrolladores.

El diseño de Android, como se ha podido esbozar en el documento, es mucho más claro y libre que el de plataformas más antiguas como Symbian, y el modelo de desarrollo por componentes podría impulsar la creación rápida de un gran número de aplicaciones interesantes.

Es indiscutible que, a día de hoy, la irrupción de la plataforma Android de la mano de Google en el mundo de las comunicaciones móviles va a tener a muy corto plazo las siguientes consecuencias:

- Para los fabricantes de móviles, los gastos en software se harán menores: en vez de crear un Sistema operativo casi para cada modelo, se podrá usar Android y simplemente personalizarlo para cada caso.
- Para los creadores de software la ventaja de una plataforma estándar está clarísima: mientras menos plataformas diferentes se soporten, más optimizadas estarán sus aplicaciones y más integradas con el sistema, por lo que tendrán menos fallos y su coste será menor.
- Los teléfonos se conectarán a la 'nube' de aplicaciones de la Web. El navegador Web que incorpora Android es primo hermano de Chrome, el programa que la misma Google acaba de lanzar para ordenadores Windows. Una de las características más destacadas de Chrome es la tecnología Gears, que te permite seguir trabajando con los datos de la red mientras estás desconectado.

- Terminales y servicios financiados con publicidad: AdSense incrustado en todo tipo de aplicaciones móviles
- Competencia de tarifas en tiempo real: Google tiene patentado un sistema de subasta inversa que busca la operadora más ventajosa entre las que nos ofrecen cobertura.

Y, actualmente se está asistiendo a una evolución de Android hacia otras plataformas que no son solo el teléfono móvil. En el modelo de desarrollo de Android no hay nada que lo limite a los teléfonos móviles, y de hecho Google asegura abiertamente que es apropiado para el uso en todo tipo de dispositivos. Esperemos unos cuantos años a que madure el parque de aplicaciones para Android. Si lo hace bien, se podrían acabar viendo dispositivos Android que aspiren a competir directamente con el PC.

REFERENCIAS

- [1] OPEN HANDSET ALLIANCE
http://www.openhandsetalliance.com/android_overview.html
- [2] Guía de desarrolladores de Android
<http://developer.android.com/guide/index.html>
- [3] ANDROID OPEN SOURCE
<http://source.android.com/>
- [4] ANDROIDE, BLOG blog sobre Android en Español
<http://celutron.blogspot.com/>

Retos en la Transición y el Futuro de la Industria Fotovoltaica Española

Eduardo Collado Fernández

Director Técnico Asociación de la Industria Fotovoltaica -
ASIF
Madrid, España
ecollado@asif.org

España tiene pues, una posición geográfica y una posición industrial que le posibilita ser uno de los motores del desarrollo de esta nueva tecnología.

II.LA TRANSICIÓN

Resumen—En este documento, se analiza la transición difícil que se está realizando por la industria solar fotovoltaica, como consecuencia del paso del RD 661/2007 al RD 1578/2008, debido a la disminución drástica de tarifas y de objetivos de potencia, para continuar con las perspectivas alentadoras de futuro, con una etapa intermedia cuando se alcance el Grid Parity, para terminar con los nuevos requisitos técnicos que deberán facilitar la integración en la red eléctrica, necesaria para el crecimiento de la energía fotovoltaica

I. INTRODUCCIÓN

En España la industria solar fotovoltaica, está realizando en los últimos años, importantes esfuerzos que la han colocado a nivel de mercados líderes. España es un país con una insolación privilegiada. Esta circunstancia hace que en España, la energía solar fotovoltaica resulte más rentable, ya que las diferencias de insolaciones medias entre España y los países de Centroeuropa, son del orden del 50%, llegándose comparativamente entre regiones soleadas de España y algunas regiones del centro de Europa, al 100% de diferencia. Esto hace que regiones en España, como por ejemplo Galicia o el País Vasco, que popularmente se creen poco adecuadas para la energía solar, reciban más radiación solar que la media en Alemania, país con el mayor mercado mundial. La industria en España está alcanzando un apreciable grado de madurez, sobre la que se están basando importantes estrategias empresariales de mejora y expansión que abarcan toda la geografía nacional, según se indica en la Figura 1.

La Industria manufacturera FV española

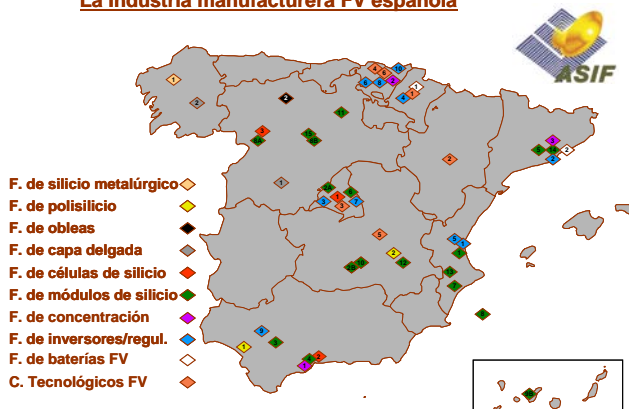


Figura 1. La industria manufacturera FV española.

Hagamos un poco de historia.

El Plan de Energías Renovables (PER) lo aprobó el Consejo de Ministros en agosto de 2005, y cubre el periodo 2005/2010, como revisión y continuación del Plan de Fomento de las Energías Renovables (PFER) de 1999. Estos planes han plasmado el compromiso de los gobiernos españoles con el aprovechamiento energético de los recursos renovables.

El objetivo que el PER fijó, en concreto para el sector de la energía solar fotovoltaica, es el de tener 400 MW instalados a finales de 2010, objetivo que se ha sobrepasado con creces puesto que el desarrollo de esta energía en España ha presentado una progresión constante, y de forma notable desde la aprobación del RD 436/2004, reforzándose el crecimiento con la aprobación del RD 661/2007, lo que ha hecho que antes de finales del año 2007 se alcanzaran los objetivos que tenía esta tecnología para finales del año 2010.

Se puede ver en la Figura 2, que el ritmo de instalaciones de los últimos años ha experimentado un súbito e importante crecimiento, así como su distribución por Comunidades Autónomas (Figura 3).

AÑO	MES	Potencia Instalada (MW)	Nº Instalaciones
2004	Diciembre	22	3.208
2005	Diciembre	46	5.300
2006	Diciembre	146	9.714
2007	Enero	165	10.449
	Febrero	189	11.122
	Marzo	223	11.599
	Abril	249	12.485
	Mayo	272	12.876
	Junio	335	13.989
	Julio	388	15.193
	Agosto	426	15.687
	Septiembre	463	16.374
	Octubre	525	17.277
	Noviembre	592	18.492
	Diciembre	692	19.967
2008	Enero	767	21.061
	Febrero	837	22.094
	Marzo	908	23.236
	Abril	1.027	25.115
	Mayo	1.190	28.251
	Junio	1.421	30.926
	Julio	1.903	36.310
	Agosto	2.427	41.113
	Septiembre	2.857	45.649
	Octubre	2.965	46.648
	Noviembre	2.972	46.729
	Diciembre	2.973	46.730

Figura 2. Distribución por Comunidades Autónomas.

Este crecimiento ha permitido alcanzar porcentajes apreciables de generación eléctrica fotovoltaica en España, ya que la media se situará a finales de del año 2008, en un 0,8% y en algunas comunidades autónomas, la electricidad fotovoltaica será responsable de más de un 3% de toda la electricidad que se consuma en esas comunidades (Figura 4).

todas las preasignaciones previstas en el RD 1578/2008 (Figura 5).

Potencia conectada a red (MW) y cuyas facturas constan en la CNE (por CCAA, datos a Diciembre-2008)

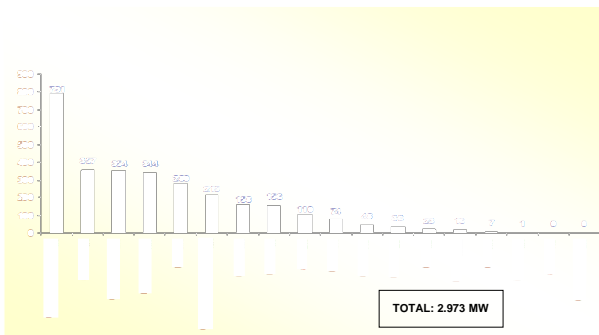


Figura 3. Potencia conectada a red.

Participación de la energía solar fotovoltaica (datos diciembre 2008) con respecto a la demanda eléctrica (por CC.AA a finales 2008)

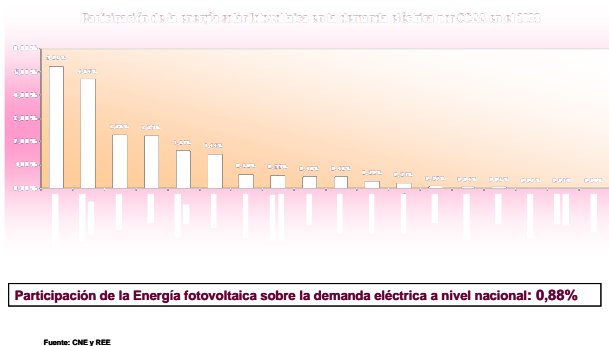


Figura 4. Participación de la Energía solar fotovoltaica con respecto a la demanda eléctrica

Sin embargo, ese súbito crecimiento que ha conllevado la implantación de significativas cantidades de instalaciones todavía con tarifas altas, está siendo frenado por la legislación actual de finales del año 2008 (Real Decreto 1578/2008), para minimizar el impacto que tiene su apoyo en la tarifa eléctrica. Esta legislación pretende que el crecimiento sea sostenible.

El ajuste está siendo dramático para el sector industrial fotovoltaico, al que se le pide que se adapte a los nuevos y más bajos volúmenes anuales de instalaciones con tarifas reducidas drásticamente, siendo los objetivos marcados por el Gobierno para los años 2009, 2010, y 2011, de aproximadamente 500 MW al año, con unas tarifas que han sufrido una bajada considerable, esperando además disminuciones adicionales en los próximos años, tal y como se puede ver en el precio de las instalaciones de Tipo II del siguiente gráfico, si se cumplen

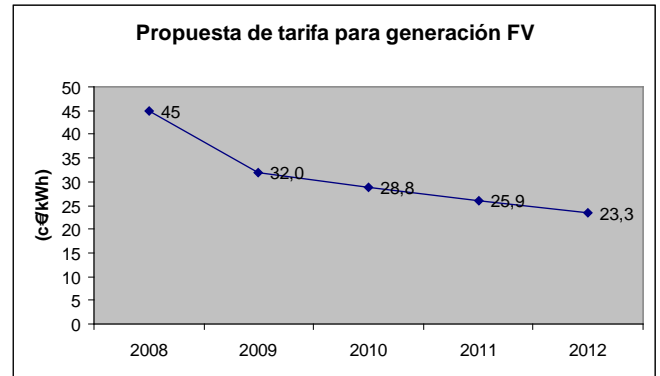


Figura 5. Propuesta de tarifa para generación FV

Es de esperar que el Sector supere los actuales momentos de transición, para que España siga siendo una de las potencias industriales líderes, y uno de los mercados de referencia mundial.

El auge surgido en la industria fotovoltaica, en el periodo de transición de los últimos 12 meses del antiguo RD 661/2007, hasta la puesta en marcha del nuevo RD 1587/2008, ha provocado un “efecto llamada”, en el que se ha pasado de una cantidad instalada de 690 MW a finales del año 2007, a una cantidad todavía sin confirmar de aproximadamente 3400 MW a finales de 2008 (aunque las cuentas de la CNE solo nos den, que a finales de Diciembre existen 2.973 MW, que ya están cobrando la tarifa correspondiente).

Esta transición va a ser traumática, ya que la industria fotovoltaica española va a sufrir el paso, desde un crecimiento exponencial, a un crecimiento totalmente planificado, muy inferior en cantidad y en cuantía de la tarifa (a menos de la cuarta parte de lo realizado en 2008, con respecto a lo aproximadamente 500 MW previstos para el 2009), y va a ser necesario el replanteamiento empresarial, con cambios estratégicos sobre el tipo de instalación a realizar (mas en tejados y menos en suelo), y sobre los márgenes del negocio, además del tiempo necesario para que puedan llevarse a cabo nuevos proyectos, ya que la mayoría de los que actualmente optan a los cupos de preasignación en suelo, exigidos por el RD 1578/2008, son proyectos “rezagados” de los lanzados durante los 12 últimos meses del RD 661/2007. A cambio de lo anterior, no es de esperar problemas en la preasignación de instalaciones en tejados, ya que de momento el primer cupo establecido para el año 2009, no ha sido cubierto en su totalidad.

III. PERSPECTIVAS DE FUTURO

Pero el sector fotovoltaico, como otros muchos, tiene carácter mundial, y ya ha demostrado su capacidad de ser una de las formas de generación de energía eléctrica, con un futuro mas prometedor.

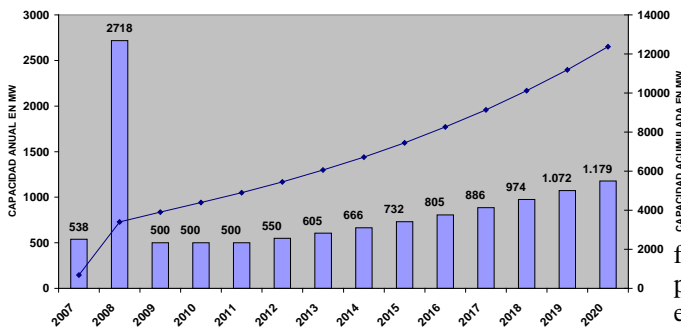
En la UE se espera que en el año 2020, las energías renovables supongan un 20% de la energía primaria, lo que

supondrá aproximadamente algo más del 40% de la energía eléctrica. En el mundo se espera que en 2050, el 50% aproximadamente de la energía eléctrica sea renovable, estando previsto por la Agencia Internacional de la Energía (AIE), que de esa cantidad total, aproximadamente la mitad sea eólica, y la otra mitad solar (fotovoltaica y térmica).

Por otra parte según EPIA (Asociación Europea de la Industria Fotovoltaica), la energía fotovoltaica podrá proveer el 12% de la demanda europea de electricidad para 2020, debido a que la evolución de la energía solar fotovoltaica será más rápida de lo anunciado anteriormente. En línea con esta idea, está el informe Solar Generation V - 2008 "Electricidad Solar para más de mil millones de personas y dos millones de puestos de trabajo en 2020", publicado por Greenpeace y EPIA, en el que se subraya el rápido crecimiento de esta industria, y afirma que podría cubrir un 8,9% de la demanda eléctrica mundial en 2030, a partir de energía FV, y del 13,8% según la previsión de Greenpeace, en su documento [R]evolución Energética.

En España actualmente y según el RD 1587/2008, si se cumplieran todas las expectativas de bajadas de costes de las instalaciones fotovoltaicas, con las consiguientes bajadas de tarifas en los cupos sucesivos, y el aumento de las cantidades anuales sucesivas (que están indexadas a las bajadas anuales de las tarifas), se podría llegar a algo más de 12 GW en España en el año 2020 (Figura 6), algo más de la mitad de lo ya previsto por ASIF en su Fotovoltaica 20, de llegar con 20GW en 2020, y casi la cuarta parte de lo indicado anteriormente por EPIA, que supondría un total en el caso español de 40GW.

ESCENARIO DEL GOBIERNO
CAPACIDAD ANUAL Y ACUMULADA DE LAS INSTALACIONES FV EN ESPAÑA



IV. GRID PARITY

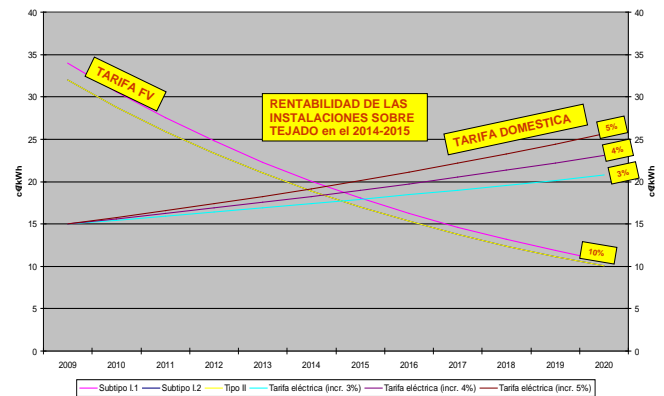
Pero la energía fotovoltaica tiene ventajas competitivas, y la necesidad de realizar la transición necesaria para disminuir costes, siendo las principales ventajas que:

- Es una energía distribuida y modular, que puede colocarse en cualquier lugar y en la cantidad precisa, para integrarse dentro de las redes eléctricas, sobre todo en instalaciones pequeñas en las redes de distribución eléctrica de media y baja tensión.
- Los costes para que las instalaciones pequeñas se integren en las redes anteriores, no necesitan competir con

respecto a los costes de generación, sino con respecto a los costes de la tarifa domestica (Grid Parity).

Las condiciones anteriores, hacen que las instalaciones fotovoltaicas sean ideales para un desarrollo sostenido a partir del momento en que se alcance el citado Grid Parity, previsto por la industria fotovoltaica aproximadamente para el 2015 (Figura 7), siendo ratificado especialmente por los fabricantes de las diferentes tecnologías actualmente en desarrollo, aunque en un informe, EPIA es mas optimista y cree que el coste de la electricidad fotovoltaica estará por debajo del precio para el consumidor, a partir de 2010 en varios mercados europeos. Italia y España, países con mayor radiación solar, podrán conseguirlo a partir de 2010 y 2012 respectivamente; en Alemania se alcanzará en 2015, y en la mayoría de los países europeos, en 2020.

Para que se produzca esta reducción de costes, la industria europea fotovoltaica se compromete a incrementar su inversión, pero pone como condición que exista un marco político apropiado. Un sistema de primas (feed in tariffs) adecuado hasta que se consiga la paridad, una simplificación de los trámites administrativos, obtener acceso prioritario a la red y la implementación de un ambicioso Plan Estratégico de Tecnología Energética (SET Plan) de ámbito europeo para incrementar los esfuerzos en investigación y desarrollo son sus demandas principales.



Para facilitar el desarrollo de las renovables en general, y la fotovoltaica en particular, ya está previsto en el caso español por el Operador del Sistema (REE), el desplazamiento de las energías de generación "ordinaria", por las "renovables", siempre y cuando no se vea comprometida la seguridad y la estabilidad del sistema eléctrico.

V. NUEVOS REQUISITOS TÉCNICOS

El RD 1578/2008, en su artículo 13, indica que, mediante una Orden del Ministro de Industria, Turismo y Comercio (MITyC), se establecerán requisitos técnicos y de calidad de las instalaciones fotovoltaicas para contribuir a la seguridad de suministro, lo que incluye la obligación de soportar los huecos de tensión; así mismo indica que obligará a cumplir estos requisitos a todas las instalaciones FV, a las futuras y a las instaladas en el pasado, aunque también indica que a éstas se les daría un periodo transitorio para adaptarse.

Hasta la publicación del RD 1578/2008, a las instalaciones FV no se les requería que cumplieran condiciones de huecos de tensión, al contrario, el RD 1663/2000 nos obliga a tener una protección de baja tensión como protección de funcionamiento en isla de la instalación, que desconecta la instalación si la tensión en barras cae por debajo de 85% de la tensión nominal, no indicando ninguna temporización lo que, de hecho, supone la desconexión inmediata de la instalación ante un hueco; así están diseñados todos los convertidos AC/DC (inversores) e instalaciones.

En el mencionado artículo 13 del RD 1578/2008, se indica explícitamente que la obligación de soportar los huecos de tensión, es condición necesaria para la percepción de la tarifa fotovoltaica. Otros de los condicionantes técnicos y de calidad de las instalaciones fotovoltaicas, van a exigir la necesidad obligatoria de normativas a los módulos fotovoltaicos, a los inversores, y al resto de los componentes de las instalaciones.

Además, para facilitar la integración en la red eléctrica, es necesario que las energías renovables sean capaces de aportar a medio plazo, los servicios complementarios que actualmente da la generación ordinaria (contribuir a la aportación de energía reactiva, y la estabilidad de la tensión y la frecuencia), además de poder soportar de una forma inmediata los condicionantes técnicos aludidos y poder soportar los huecos de tensión sin desconectar la instalación, no provocando de esta forma la desestabilización del sistema eléctrico.

VI. CREACIÓN DE EMPLEO

El importante crecimiento experimentado por el sector fotovoltaico durante los últimos cinco años ha resultado en una importante creación de empleo, pasando de unos 600 empleos directos en 2001 hasta los cerca de 24.000 empleos del año 2007, con el siguiente desglose:

OCTUBRE 2007	Directos	Indirectos	Total
Fabricantes	3355	3355	6710
Instaladores	7920	3960	11880
Otros	3960	1188	5148
TOTAL	15235	8503	23738
TOTAL	23.738		

El año 2008 fue como se ha visto por los datos, un año atípico e insostenible, en el que la cantidad de empleos ha podido oscilar en el entorno de 60.000 a 80.000.

La estabilidad planteada en el año 2009, como una posible continuación de lo que fue el pasado 2007, van a hacer que el empleo en este sector, se sitúe alrededor de 30.000 empleos, con una capacidad de crecimiento estimada que puede rondar en España los 60.000 empleos en el 2020

Hasta el momento, la creación de empleo se ha concentrado fundamentalmente en el segmento de fabricación, si bien según

aumente la base instalada se espera que el peso de este segmento disminuya, dando paso a un mayor protagonismo a los instaladores y a las empresas encargadas del mantenimiento de las instalaciones en operación. Los ambiciosos planes de inversión de las empresas fabricantes y su incursión en los segmentos de fabricación de polisilicio y lingotes contribuirán a mantener la creación de empleo en este segmento.

Un apoyo continuado a la solar fotovoltaica, permitiría conservar los puestos de trabajo existentes y mantener la creación de empleo a largo plazo.

Se ha realizado una estimación del nivel de empleo consecuente con el desarrollo de la energía fotovoltaica, y en línea con estimaciones realizadas por la EPIA (European Photovoltaic Industry Association), se tiene la siguiente previsión para el 2020, es un escenario sostenible (Figura 8).



Figura 8: Previsión del número de empleos directos generados por la industria solar fotovoltaica. Fuente: EPIA

En cuanto a la naturaleza del empleo creado, destacamos que, especialmente en las actividades de instalación y mantenimiento, el desarrollo de la generación fotovoltaica ha permitido crear nuevos puestos de trabajo estables en zonas rurales, generalmente desfavorecidas en términos de creación de empleo.

VII. CONCLUSIONES

Las perspectivas de la industria fotovoltaica en España a partir del 2010 son muy halagüeñas, una vez pasada la transición expansiva del año 2008, y la contracción y estabilización del sector en 2009, y van a cimentar una senda creciente, estable y continua, que la va a llevar a ser una energía con la que se empezó a contar por su cantidad y sus perspectivas de futuro en el año 2008, y que harán que esta industria sea competitiva sin necesidad de subvenciones en dos escalones, el primero aproximadamente a partir del 2015, con respecto a la tarifa doméstica, y en un futuro mas lejano, con respecto a la generación tradicional.

Esto se materializará en un crecimiento sostenible, tanto de las instalaciones en si, como del empleo generado por la industria fotovoltaica española.

Por lo tanto no está lejano el día, en el que la denominación de generación tradicional y la de generación fotovoltaica, formarán parte del mismo conjunto, corroborando que en el caso español, el futuro de la energía fotovoltaica en España va a ser brillante.

REFERENCIAS

- [1] ASOCIACIÓN DE LA INDUSTRIA FOTOVOLTAICA (2008).Informe, Hacia un suministro sostenible de electricidad
- [2] GREENPEACE (2007) Renovables 2050. Informe sobre el potencial de las Energías Renovables en la España peninsular. Greenpeace
- [3] GREENPEACE Y EPIA (2008) Solar Generation V - 2008 "Electricidad Solar para más de mil millones de personas y dos millones de puestos de trabajo en 2020"
- [4] GREENPEACE (2008) [R]evolución Energética
- [5] PLATAFORMA TECNOLÓGICA FOTOVOLTAICA (2007) Visión de la Tecnológica FV en España
- [6] REAL DECRETO 1578/2008. BOE del 27 de septiembre de 2008
- [7] WEO 2008 World Energy Outlook Noviembre 2008

Introducción a los algoritmos Cordic

Jose Antonio Cámara Madrid
 Grupo de IEEE de la UNED
 Ingeniería Industrial.
 Madrid, España
 joseacamara@ieee.org

Resumen —Esta publicación pretende dar a conocer de un modo básico un algoritmo que actualmente está siendo muy usado por la potencia del método en FPGA y en DSP.

I. INTRODUCCIÓN

Cordic es un acrónimo de COordinate Digital Computer que en castellano significa Computadora Digital para Rotación de Coordenadas. El Algoritmo original fue propuesto por Jack Volder en el año 1959, con el propósito de calcular funciones trigonométricas mediante la rotación de vectores. La rotación de vectores puede a su vez utilizarse para conversión de sistemas de coordenadas (cartesiano a polar y viceversa), magnitud de vectores y como parte de funciones matemáticas más complejas como la FFT y DCTE.

II. FUNDAMENTO TEÓRICO.

El algoritmo cordic es un método iterativo para rotar vectores ciertos ángulos determinados y se basa en sumas y multiplicaciones, describe la rotación de un vector bidimensional (cómo el R de la figura 1). La rotación de R a R', genera unas coordenadas que son x' e y' en función de x e y original.

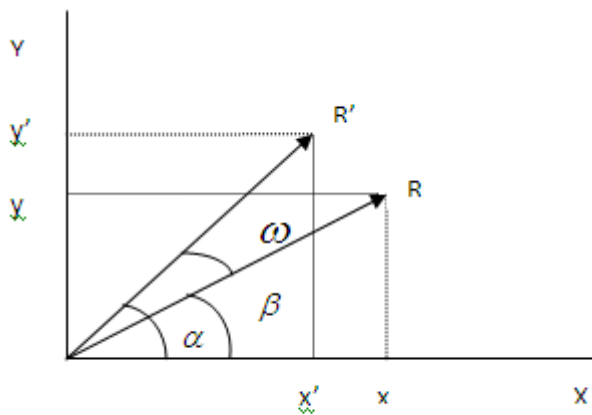


Figura 1

$$x' = x \cdot \cos(\omega) - y \cdot \text{sen}(\omega) \quad (1)$$

$$y' = y \cdot \cos(\omega) + x \cdot \text{sen}(\omega) \quad (2)$$

Según ecuación (1) y (2), obtenemos la rotación como multiplicaciones y sumas, pero se siguen aplicando relaciones trigonométricas para simplificar el método:

$$R: \begin{cases} x = r \cdot \cos \beta \\ y = r \cdot \text{sen} \beta \end{cases} \quad (3)$$

$$R': \begin{cases} x' = r \cdot \cos \alpha \\ y' = r \cdot \text{sen} \alpha \end{cases} \quad (4)$$

Conociendo que la tangente es:

$$\text{tg}(x) = \frac{\text{sen}(x)}{\cos(x)} \quad (5)$$

Con lo que se puede agrupar:

$$x' = \cos(\omega) \cdot (x - y \cdot \text{tg} \omega) \quad (\text{Eq. 6})$$

$$y' = \cos(\omega) \cdot (y + x \cdot \text{tg} \omega) \quad (\text{Eq. 7})$$

Conociendo que:

$$\cos \omega = \frac{1}{\sqrt{1 + \text{tg}^2 \omega}} \quad (\text{Eq. 8})$$

En el algoritmo cordic las rotaciones son reemplazadas por las pseudorotaciones que consiste en cambiar la magnitud del vector R una vez rotado, tal y cómo se muestra en la figura 2.

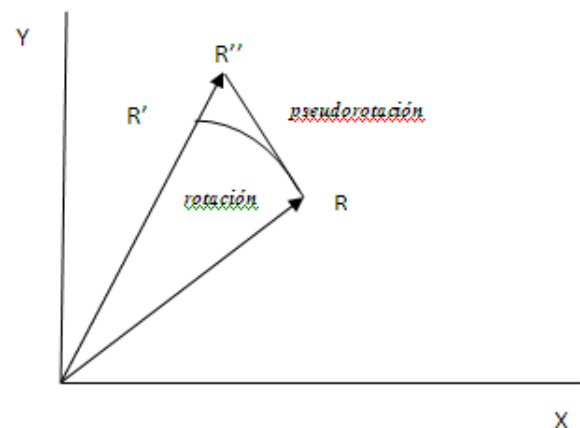


Figura 2.

El vector pseudorotado R'' incrementa su magnitud en respecto del vector R original, la pseudorotación está caracterizada por las siguientes ecuaciones:

$$x''_j = \sum_{i=0}^{j-1} \sqrt{1+tg^2 \omega_i} \left(x \cdot \cos \left(\sum_{i=0}^{j-1} \omega_i \right) - y \cdot \sin \left(\sum_{i=0}^{j-1} \omega_i \right) \right) \quad (\text{Eq. 9})$$

$$y''_j = \sum_{i=0}^{j-1} \sqrt{1+tg^2 \omega_i} \left(y \cdot \cos \left(\sum_{i=0}^{j-1} \omega_i \right) + x \cdot \sin \left(\sum_{i=0}^{j-1} \omega_i \right) \right) \quad (\text{Eq. 10})$$

$$z''_j = z - \sum_{i=0}^{j-1} \omega_i \quad (\text{Eq. 11})$$

Esta última componente se le llama acumulador angular porque incluye las rotaciones efectuadas hasta el momento.

Las pseudorotaciones se puede descomponer en una suma de pseudorotaciones más pequeñas, si se asume que $x=x_0$, $y=y_0$ y $z=z_0$, luego de j iteraciones para una pseudorotación se obtiene:

$$\begin{aligned} x_{j+1} &= K \cdot (x_j - y_j \cdot d_j \cdot 2^{-2j}) \\ y_{j+1} &= K \cdot (y_j + x_j \cdot d_j \cdot 2^{-2j}) \end{aligned} \quad (\text{Eq. 12})$$

Dónde $K=\cos \omega$ y $d=\pm 1$ dependiendo del sentido de rotación. El algoritmo Cordic se usa en dos formas, en modo rotación y en modo vectorización.

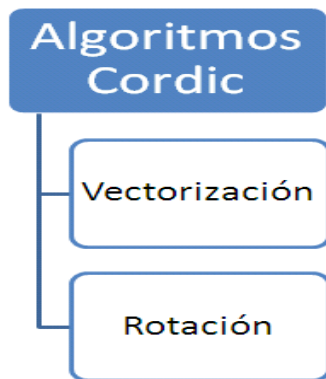


Diagrama 1.

El modo rotación que rota el vector de entrada en un ángulo específico que se introduce como parámetro, el acumulador angular se inicializa con el ángulo a rotar. La decisión sobre el sentido de rotación en cada paso de iteración, se efectúa para minimizar la magnitud del ángulo acumulado. Por ello, el signo que determina el sentido de rotación, se obtiene del valor de dicho ángulo en cada paso. Para el modo rotación, las ecuaciones son:

$$x_{j+1} = x_j - y_j \cdot d_j \cdot 2^{-j} \quad (\text{Eq. 13})$$

$$y_{j+1} = y_j + x_j \cdot d_j \cdot 2^{-j} \quad (\text{Eq. 14})$$

$$z_{j+1} = z_j - d_j \cdot \arctg(2^{-j}) \quad (\text{Eq. 15})$$

Para el modo vectorización, el ángulo introducido se rota para alinearlo con el eje X. Para obtener este resultado, en lugar de minimizar la magnitud del acumulador angular, se minimiza la magnitud de la componente y, ya que si $y=0$ entonces el vector se encuentra sobre el eje X. Asimismo se utiliza el signo del componente y para determinar la dirección de la rotación. Si el acumulador angular se inicializa a cero, al final del proceso contendrá el ángulo de rotación adecuado. Por lo tanto se pueden deducir las siguientes ecuaciones:

$$x_{j+1} = x_j - y_j \cdot d_j \cdot 2^{-j} \quad (\text{Eq. 16})$$

$$y_{j+1} = y_j + x_j \cdot d_j \cdot 2^{-j} \quad (\text{Eq. 17})$$

$$z_{j+1} = z_j - d_j \cdot \arctg(2^{-j}) \quad (\text{Eq. 18})$$

El Algoritmo Cordic está restringido al uso en el primer cuadrante, por lo tanto, se debe de reducir al primer cuadrante.

i	2 ^{-j}	arctg(2 ^{-j})	Ki
0	1	0,785398163	0,707106781
1	0,5	0,463647609	0,816496581
2	0,25	0,244978663	0,894427191
3	0,125	0,124354995	0,942809042
4	0,0625	0,06241881	0,9701425
5	0,03125	0,031239833	0,984731928
6	0,015625	0,015623729	0,992277877
7	0,0078125	0,007812341	0,99611649
8	0,00390625	0,00390623	0,998052578

Tabla 1

REFERENCIAS

[1] Algoritmos Cordicos. Tesis de Esther Leyva Suárez. Instituto Politécnico Nacional. Escuela Superior de Física y Matemáticas. México D. C.
 [2] J.E. Volder, "The Cordic Trigonometric Computing Technique". IRE Transactions on Electronics Computers, V.EC-8, No.3, pp330-334.1959.

English Zone

Gloria Murillo

Coordinadora Boletín Electrónico e English Zone
Universidad Nacional de Educación a Distancia
Madrid, España

I. HOW LONG?

Entre españoles es muy común oír la expresión "How much time?" para "¿Cuánto tiempo? Sin embargo entre angloparlantes la forma más natural es "**How long?**"

¿Cuánto tiempo llevas viviendo en España?	How long have you lived in Spain?
¿Desde cuando trabajas en tu empresa?	How long have you worked for you company?
¿Desde cuando se vende helado aquí?	How long have they been seelling icecream here?
¿Cuanto tiempo llevas esperándome?	How long have you been waiting for me?
¿Desde cuando estudias inglés?	How long have you been studying English?
¿Cuanto tiempo lleva lloviendo?	How long has it been raining?
¿Desde cuanto hace que la conoces?	How long have you known her?

Siempre usamos el presente perfecto para hablar de una acción empezada en el pasado pero todavía relevante en el presente.

II. APART FROM

Observamos la misma diferencia con la expresión equivalente a "aparte", "excepto" o "aparte de que". Cuando le sigue un sustantivo, simplemente decimos "**apart from**".

Aparte del trabajo, todo va bien.	Apart from work, everything is fine.
Excepto la cocina, el piso es precioso.	Apart from the kitchen, the flat is lovely.
Aparte de las moscas, comimos bien.	Apart from the flies, we had a good lunch.
Aparte de la lluvia, nos lo pasamos muy bien.	Apart from the rain, we had a great time.
Aparte de los nervios, la boda me hace mucha ilusión.	Apart from the nerves, I'm really looking forward to the wedding.

En cambio, cuando le sigue un verbo se dice "**apart from the fact that**".

Aparte de que tengo mucho estrés en el trabajo, todo va bien.	Apart from the fact that I'm under a lot of stress at work, everything is fine.
Aparte de que la cocina es muy pequeña, el piso es precioso.	Apart from the fact that the kitchen is very small, the flat is lovely.
Aparte de que soy alérgico al marisco, como de todo.	Apart from the fact that I'm allergic to seafood, I eat everything.
Aparte de que la pantalla me resulta un poco pequeña, me gusta mi portátil.	Apart from the fact that I find the screen a bit small, I like my

	laptop.
Aparte de que había muchos turistas, nos las arreglamos para ver muchas cosas.	Apart from the fact that there were a lot of tourists, we managed to see many things.

III. KEY LEARNING

“Plenty” nunca significa “mucho” sino “más que suficiente”

Don't worry. We have plenty of time.

IV. 4ONE EXPRESSION

He eats like a horse.

Come como una lima

V. ONE PHRASAL VERB

To get through – hacer entender

I'm trying to get it through to him how important this is!

Estoy intentando hacerle entender lo importante que es esto.

VI. FUNNY LESSON.

Un par de chistes en lengua inglesa que no harán aprender cierto vocabulario en inglés y nos harán esbozar alguna sonrisa.

An Order of Spaghetti

A doctor was having an affair with his nurse. Shortly afterward, she told him she was pregnant. Not wanting his wife to know, he gave the nurse a sum of money and asked her to go to Italy and have the baby there.

“But how will I let you know the baby is born?” she asked. He replied, “Just send me a postcard and write 'spaghetti' on the back. I'll take care of expenses.”

Not knowing what else to do, the nurse took the money and flew to Italy.

Six months went by and then one day the doctor's wife called him at the office and explained, “Dear, you received a

very strange postcard in the mail today from Europe, and I don't understand what it means.”

The doctor said, “Just wait until I get home and I will explain it to you”. Later that evening, the doctor came home, read the postcard, fell to the floor with a heart attack. Paramedics rushed him to the ER. The lead medic stayed back to comfort the wife. He asked what trauma had precipitated the cardiac arrest.

So the wife picked up the card and read, “Spaghetti, Spaghetti, Spaghetti, Spaghetti - Two with sausage and meatballs, two without”.

Vocabulary

affair (<i>aféer</i>) – aventura amorosa	afterward (<i>áfter ward</i>) - después
meatballs (<i>mítbols</i>) - albóndigas	attack (<i>aták</i>) - ataque
medic (<i>médic</i>) – médico	strange (<i>stréindch</i>) - extraño
order (<i>órder</i>) – orden, pedido	sum (<i>sam</i>) - suma
paramedics (<i>paramédics</i>) - paramédicos	cardiac arrest (<i>cárdiac arrést</i>) - paro cardíaco
sausage (<i>sósidch</i>) - salchicha	ER (<i>i ár</i>) – sala de emergencias
shortly (<i>shórtli</i>) - poco tiempo	Europe (<i>iúrop</i>) - Europa
nurse (<i>néers</i>) - enfermera	trauma (<i>tróoma</i>) - trauma
lead (<i>líid</i>) – principal	wife (<i>uáif</i>) - esposa

Husband in Sorrow

A woman awakes during the night, and her husband isn't in bed with her. She goes downstairs to look for him. She finds him sitting at the kitchen table with a cup of coffee in front of him. He appears to be in deep thought, just staring at the wall. She watches as he wipes a tear from his eye and takes a sip of his coffee. "What's the matter, dear?" she asks. "Why are you down here at this time of night?"

The husband looks up from his coffee, "Do you remember 20 years ago when we were dating, and you were only 16?" he asks solemnly.

"Yes, I do," she replies

"Do you remember when your father caught us in the back seat of my car making love?"

"Yes, I remember," says the wife, lowering herself into a chair beside him.

The husband continues, "Do you remember when he shoved the shotgun in my face and said, 'Either you marry my daughter, or I'll send you to jail for 20 years?'"

"I remember that, too," she replies softly.

He wipes another tear from his cheek and says, "I would have gotten out today".

Vocabulary

matter (*máter*) - asunto, problema **beside** (*bisáid*) - al lado

shotgun (*shótgan*) - escopeta

sip (*sip*) - sorbo

softly (*sóflty*) - suavemente

solemnly (*sólemnli*) - solemnemente

sorrow (*sórrou*) - pena

tear (*tíer*) - lágrima

thought (*zot*) - pensamiento

too (*tchúu*) - también

cheek (*chúik*) - mejilla

daughter (*dóter*) - hija

dear (*díar*) - querido

deep (*dúip*) - profundamente

downstairs (*dáunstérs*) - abajo

during (*diúring*) - durante

either (*íder*) - tampoco

jail (*dchéil*) - cárcel

VII. LINKS

- [1] <http://www.funylessons.com/>
- [2] <http://www.mansioningles.com/index.htm>
- [3] <http://www.vausys.com/>
- [4] <http://www.vaughanradio.com/reproductor/player2.htm>
- [5] <http://www.saberingles.com.ar/songs/315.html>



Rama de Estudiantes UNED
<http://www.ieec.uned.es/ieee-uned/>



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



Web IEEE-UNED

<http://www.ieec.uned.es/ieee-uned/>
Más info: elio@ieec.uned.es

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