

Pixels&Code

JULIO 2011 > N° 04 -

**ARMÁ TU PROPIO SITIO
DE VIDEO TUBE**

**CUIDANDO LA ACCESIBI-
LIDAD EN PORTALES WEB**



ANDROID Y WINDOWS PHONE 7

Secretos de la programación de mobile apps

Plataforma para Resellers

¡El negocio lo hacés vos!

Tu propio negocio de hosting con todo lo que necesitás.
Con más de 8 años de experiencia, entendemos tus necesidades. Por eso creamos una plataforma con productos y herramientas para que tu negocio crezca con la mayor rentabilidad.

www.dattatec.com/resellers



ENTERATEQUETENGO.COM

- ✓ Muchas Minutas
- ✓ Vacaciones en Montecarlo
- ✓ T.V. 75 pulpadas
- ✓ Velero con frigobar
- ✓ Loft vista al mar
- ✓ Loft vista al mar
- ✓ Colección completa de muñecos Jack
- ✓ Reabrir Studio 54 por una noche

Conocé el secreto de mi éxito
WWW.ENTERATEQUETENGO.COM





dattatec.com
Tu Hosting hecho Simple!

JEFA DE REDACCIÓN



Débora Orué

COLUMNISTAS



Adrián Ortiz



Cristian Hernán Gaitano Ornia



Gabriel Walter Gaitano Ornia



Juan Gutmann



Roxana Miguel



Natalia Cantero

DISEÑO EDITORIAL Y CREATIVIDAD

www.ampersandgroup.com.ar

REDACCIÓN

lectores@pixelscode.com

COMERCIALIZACIÓN

*Dattatec.com SRL - Córdoba 3753, Rosario, Santa Fe
www.dattatec.com*

DATOS DE CONTACTO

*Dirección Comercial:
publicidad@pixelscode.com*

Las opiniones expresadas en los artículos son exclusiva responsabilidad de sus autores y no coinciden necesariamente con la opinión de Dattatec.com SRL.

Sumario



NOTA CENTRAL

**30 // PROGRAMACIÓN
ANDROID...**

**42 // PROGRAMACIÓN
MOBILE PARA WINDOWS
PHONE 7**



6 // PROGRAMACIÓN

Accesibilidad Web

12 // BASES DE DATOS

Introducción a las bases de datos...

20 // PROMOCIÓN WEB

Cómo encauzar nuestra estrategia...

24 // PROGRAMACIÓN

Armá un sitio de video tube en 5 minutos

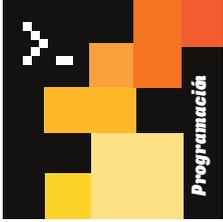
48 // ALGORITMOS

Algoritmos: Quick Sort

54 // TENDENCIAS

Tecnología NFC: La llave de la instantaneidad...





Accesibilidad Web: Pautas para crear un sitio navegable

El diseño y el buen funcionamiento no son los únicos aspectos a tener en cuenta a la hora de crear un sitio, sobre todo si deseamos que sea accesible para todos, incluso para aquellas personas con capacidades disminuidas. En este artículo conoceremos una serie de datos útiles para desarrollar un producto web sin dejar de lado el concepto de accesibilidad.

Por Roxana Miguel
roxana.miguel@pixelscode.com
@roxanamiguel





Diseños espectaculares pero que no pueden ser apreciados, accesos rápidos a páginas que son inalcanzables, sonidos para destacar botoneras que no se pueden oír y links interesantes, pero imposibles de cliquear. Es como construir un edificio espectacular sin escaleras o, más gráfico aún, una ciudad sin rampas para subir o bajar de las veredas. En muchos sentidos la Web puede convertirse en un espacio anegado para las personas que cuentan con algún tipo de capacidad reducida, sea física o mental. Lo cierto es que no todos accedemos del mismo modo a Internet. No será lo mismo navegar los anchos caminos de la Web para aquellas personas que no pueden utilizar sus manos y por ende manipular el mouse, tampoco para los que sufren cualquier grado de sordera, las personas con dificultades en la vista e incluso los que no pueden ver por completo. También puede ser una tarea compleja para disléxicos, y personas con dificultades cognitivas que afectan a su memoria, su atención y sus habilidades lógicas. El daltonismo y otras tantas patologías se convierten en grandes obstáculos, de una forma similar al que no conocer un idioma o no disponer de determinados conocimientos pueden transformarse en barreras para la comprensión y la habilidad en el mundo online. Imaginemos la siguiente escena: una persona que nunca se ha acercado a una computadora decide dar sus primeros pasos en Internet. Le han dicho que se trata de un excelente espacio para hacer negocios. Pero no conoce eBay, ni sabe lo que es un lector de feeds, no ha manejado nunca una cuenta de Twitter ni un perfil empresarial en Facebook, tampoco cuenta con buen manejo de inglés o chino, lengua

pujante en el cosmos empresarial. Este hombre, que en la jerga se conoce como “inmigrante web”, no cuenta con todas las herramientas que un “nativo web” sí posee. En este sentido se comprende que disponibilidad de herramientas es un factor clave en la experiencia online y que se trata de una situación que puede ser remediada al contemplar la diversidad. Retomando el cauce, todo aquel que cuente con alguna “herramienta” disminuida o acaso no posea una de ellas, también puede disfrutar, a su modo, de la información, el entretenimiento y toda la oferta de Internet, y esto demuestra una vez más la naturaleza de un medio democrático, plural y –término que nos convoca- ¡accesible! Ahora bien, bajo esta contemplación es válido notar que la integración en Internet comienza en el trabajo de los programadores y los diseñadores, responsables de poner frente a estos usuarios los instrumentos que suplan la carencia. En 1999, World Wide Web Consortium (W3C), entidad que promueve



Es como construir un edificio espectacular sin escaleras o, más gráfico aún, una ciudad sin rampas para subir o bajar de las veredas. En muchos sentidos la Web puede convertirse en un espacio anegado para las personas que cuentan con algún tipo de capacidad reducida.”



Las WCAG ofrecen indicaciones para los webmasters para armar contenidos viables, también hay pautas para herramientas de autor (ATAG) dirigidas para los desarrolladores de software y, como si fuera poco, las pautas para agentes de usuarios (UAAG) con indicaciones para navegadores, de manera que colaboren para que los programas sean asequibles para todos.”

la accesibilidad web por medio de su grupo de trabajo Web Accessibility Initiative publicó una primera versión, denominada 1.0, de las Pautas de Accesibilidad Web, que prontamente fueron aceptadas por todo el mundo y que en el año 2008 se convirtieron en una recomendación oficial. Las WCAG ofrecen indicaciones para los webmasters para armar contenidos viables, también hay pautas para herramientas de autor (ATAG) dirigidas para los desarrolladores de software y, como si fuera poco, las pautas para agentes de usuarios (UAAG) con indicaciones para navegadores, de manera que colaboren para que los programas sean asequibles para todos.

En definitiva, que un sitio web esté diseñado bajo estas pautas implica que todos los usuarios pueden acceder y por lo tanto cuentan con un código semánticamente correcto,

con textos alternativos y enlaces que permitan complementar con otras tecnologías de apoyo como lectores pantalla, líneas de Braille, magnificadores de pantalla o software de idioma para simplificar la usabilidad.

LEY DE ACCESIBILIDAD DE LA INFORMACIÓN EN LAS PÁGINAS WEB

El 3 de noviembre del 2010, la Cámara Nacional Argentina aprobó la ley con la que exige respetar las normas y requisitos de accesibilidad de la información. Aquí vemos un extracto de la misma:

El Estado nacional, entiéndanse los tres poderes que lo constituyen, sus organismos descentralizados o autárquicos, los entes públicos no estatales, las empresas del Estado y las empresas privadas concesionarias de servicios públicos, empresas prestadoras o contratistas de bienes y servicios, deberán respetar en los diseños de sus páginas Web las normas y requisitos sobre accesibilidad de la información que faciliten el acceso a sus contenidos, a todas las personas con discapacidad con el objeto de garantizarles la igualdad real de oportunidades y trato, evitando así todo tipo de discriminación. Se entiende por accesibilidad a los efectos de esta ley a la posibilidad de que la información de la página Web sea comprendida y consultada por personas con discapacidad y por usuarios que posean diversas configuraciones en su equipamiento o en sus programas.

¿CÓMO SABER SI MI WEB ES ACCESIBLE?

Si bien puedes disponer de un testing conformado por un equipo

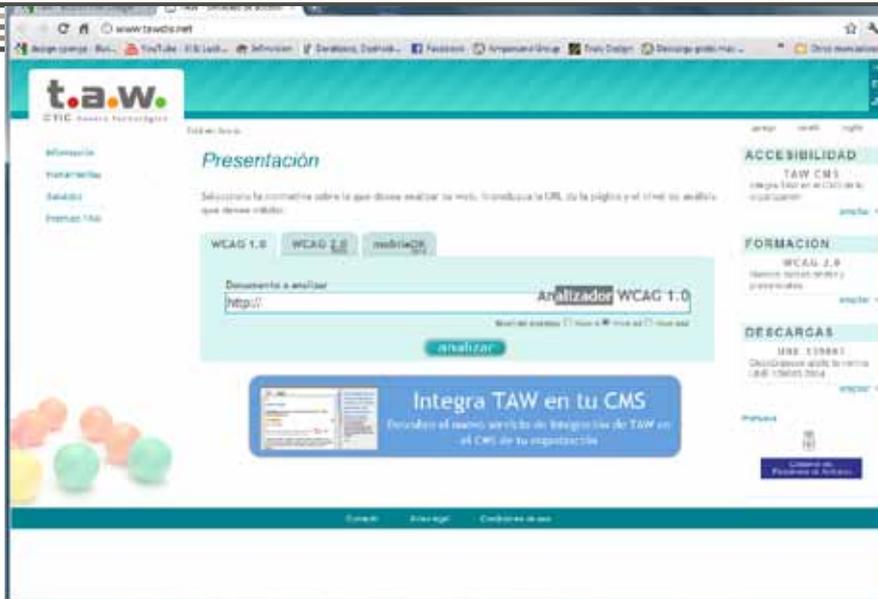
de personas con distintas discapacidades que marquen cuáles son las falencias del diseño y desarrollo de tu sitio, existen métodos más apropiados por medio de otras herramientas que funcionan como agentes de navegación pero que en definitiva tienen la misión de traducir el contenido que está a la vista pero al modo accesible. Estos son algunos ejemplos:

eDesigner: Creado por IBM Home Page Reader es una herramienta gratuita que sirve para evaluar la accesibilidad de un sitio web según las pautas W3C o Section 508. Muestra una simulación de lo que vería una persona ciega que usa lector de pantallas, con un tiempo marcado por segundos y código de colores que se requieren para navegarla. Las zonas oscuras marcan la demora de usabilidad que puede resolverse con enlaces intradocumentales (skip content).

Taw: Es un test desarrollado por la Fundación CTIC que está destinado para los diseñadores y desarrolladores web. Cuenta con dos modalidades de análisis: analizador online o aplicación instalable de forma local que estudian los resultados de acuerdo a las pautas de accesibilidad. Si el programa se ejecuta de forma local entonces se lograrán resultados más extensos y con opción de añadir comentarios.

Hera: Realiza un análisis automático de la página, hace un reporte de los errores y qué puntos de verificación de acuerdo a las Directrices de Accesibilidad para el Contenido Web 1.0 se deben resolver manualmente. Con este programa es necesario conocer las Directrices para solucionar los inconvenientes que presente la página.

Fangs: Se trata de una extensión



tadores por voz o línea de Braille.
 - Use el **elemento map** y texto para las zonas activas. Los mapas de imagen sirven para especificar regiones en una imagen u objeto y designar acciones para cada región, para lo cual hay que asociar un objeto con una especificación de áreas geométricas. Aunque los mapas de imágenes se pueden hacer en el cliente o en el servidor, lo recomendable es que sean en el lado del cliente ya que son accesibles a personas que utilizan agentes

para el navegador Mozilla Firefox que representa textualmente una página de la misma forma en que lo haría un lector de pantalla. El resultado que muestra es lo mismo que vería una persona con discapacidad que utiliza un lector de pantalla. Se presenta en tres muestras: Screen Reader output, donde se ve linealmente la página con anuncios marcados en gris, Heading List, que expone un listado de encabezados de página, y Links List, con otro listado de enlaces de la misma.

MANOS A LA OBRA

Pensar en un sitio web accesible implica preocuparnos por varios detalles que lo hacen apreciable para todas las personas, sea cual fuera la dificultad que imposibilite ver, leer o comprender el contenido y las formas de navegación del mismo. En este sentido la W3C estableció una guía para crearlos respondiendo a las pautas de comprobación:

- En cuanto al **uso de imágenes y animaciones**, lo conveniente es usar el atributo alt para describir la función de cada elemento visual. Este atributo se encarga de especi-



car el texto alternativo para agentes de usuario que no puedan mostrar imágenes, formularios o aplicaciones. Se utiliza sobre elementos IMG y AREA, aunque opcionalmente puede usarse para INPUT y APPLET. Sin embargo, para imágenes decorativas lo recomendable es usar hojas de estilo debido a que el atributo terminaría por hacer más lento el navegador y confundir los interpre-

de usuario no gráficos e informa sobre qué región está activa y cuál no. A este tipo de mapas se los conoce como etiquetas map y las zonas activas son etiquetas area, que a su vez poseen atributos shape coords, indicados para definir la forma geométrica de la zona activa. Finalmente, para que el mapa de imágenes sea accesible debe contar con un texto alternativo con atributo



“Pensar en un sitio web accesible implica preocuparnos por varios detalles que lo hacen apreciable para todas las personas, sea cual fuera la dificultad que imposibilite ver, leer o comprender el contenido y las formas de navegación del mismo.”

alt para cada etiqueta area.

- El **contenido multimedial** será más aprovechable si cuenta con subtítulos, transcripciones de sonido y descripciones de video.

Puedes valerte de tecnologías como Synchronized Multimedia Integration Language (SMIL) o Microsoft Synchronize Accessible Media Interchange (SAMI) que sirven para sincronizar los subtítulos o una transcripción con un video o grabación sonora. Programas de animaciones como Adobe Macromedia Flash y PDF han incorporado conversores online para facilitar la accesibilidad. De todas formas, el mejor consejo para un desarrollador es evitar el uso de archivos multimedia sin sentido.

- Los **enlaces de hipertextos** tienen que ser muy claros. Una buena forma de comprobarlo es quitarlos de contexto y verificar si realmente

aclaran hacia dónde conducen. Es común ver links como: “pinche aquí”, “entradas antiguas”, “ver resultados”. En estos casos la suma de estos links no aclaran a qué información concreta se dirigen. Lo correcto sería: “pinche aquí para leer más de (Titulo de la entrada)”. Hay que tener presente que un interpretador de texto a voz dejará más claro el camino a seguir por el usuario con dificultades visuales, en ese caso. También existen opciones mediante códigos CSS para utilizar enlaces poco descriptivos pero accesibles.

- La **organización de las páginas** siempre favorecerá a la navegación. Para ello hay que usar encabezados, listas, estructuras y CSS para la maquetación. Los consejos en este sentido son: utilizar las etiquetas de HTML como <title>, <h1>, <h2>, ..., , , <p>, <blockquote> que sirven para crear una estructura de página y no utilizar tablas, es preferible maquetar con CSS que ofrece los mismos resultados.

- Las **figuras y diagramas** tienen que tener una descripción en la página. Use el atributo longdesc porque complementa el atributo alt y se pueden manejar con etiquetas de XHTML como img, iframe y frame con valor URI (Uniform Resource Identifier). Este atributo no contiene un texto alternativo para cada imagen como alt, sino una dirección a otra página o a otra dentro de la misma - allí figurará la descripción de la imagen y se usa cuando las descripciones son muy largas. Pocos navegadores soportan este atributo, en el caso de Firefox se utiliza un complemento al que se



llega mediante el botón derecho del mouse y que permite navegar la URL.

- **Scripts, applets y plug-ins:** Ofrezca contenido alternativo si las funciones nuevas no son accesibles. Java Script o algunos elementos realizados en Java o Flash complican el acceso al site. Aunque se pudieran interpretar no se podría lograr una representación alternativa en texto o voz para una animación de un applet. Por esto es que se sabe que el HTML Dinámico (DHTML) no funcionará en sitios accesible. Existen estrategias para desvincular Java Script y lograr un contenido alternativo sin necesidad de quitar los beneficios que ofrecen las capas. Siempre es recomendable contar con un sitio plano y otro “con todos los chiches” no sólo por la accesibilidad que permite sino que además porque representa menores costos de mantenimiento.

- **Marcos.** Use el elemento noframes y títulos con sentido. La regla de oro para los desarrolladores web es no usar marcos, en este caso, principalmente porque pueden dar problemas para visualizar un site con navegadores antiguos, de sólo texto o agentes de usuarios no visuales. Pero si aún así quieres traerlos a tu página hay que tener en cuenta algunos consejos como proporcionar un texto equivalente para todo elemento no textual por medio de los atributos ya vistos, titular cada marco para facilitar la identificación al navegar y describir el objetivo de cada marco con la relación entre ellos. Los atributos de descripción como title y longdesc se emplearán en la etiqueta frame.

De lo contrario, la etiqueta a utilizar es <noframes> para crear versiones alternativas.

- **Tablas:** Facilite la lectura línea a línea. Resuma. Todos los datos son necesarios cuando hablamos de accesibilidad, pero tampoco hay que escribir una novela. Se trata de identificar bajo etiquetas todo el contenido en su conjunto. Los usuarios que emplean navegadores no visuales podrán recorrer la web sin problemas de forma lineal. En el caso del título se empleará la etiqueta caption, los encabezamientos usarán una etiqueta th y cuando se contengan encabezamientos complejos se usan los atributos scope y headers para definir la relación entre varias celdas. Finalmente, el resumen se define en HTML con el atributo summary de la etiqueta table.

LOS BENEFICIOS DE UNA WEB ACCESIBLE

En muchos casos los clientes serán quienes pedirán que sus contenidos sean apreciados por todos, sin embargo, aún existen quienes ignoran que una porción de la sociedad no puede navegar una web de la misma forma. Es misión de los diseñadores y desarrolladores web tener presente este aspecto y llevarlo a la práctica con cada proyecto. Para que esta fase del trabajo no sea vista negativamente por aquellos clientes neófitos del concepto de accesibilidad web, te proporcionamos tres razones por las cuales crear un sitio accesible:

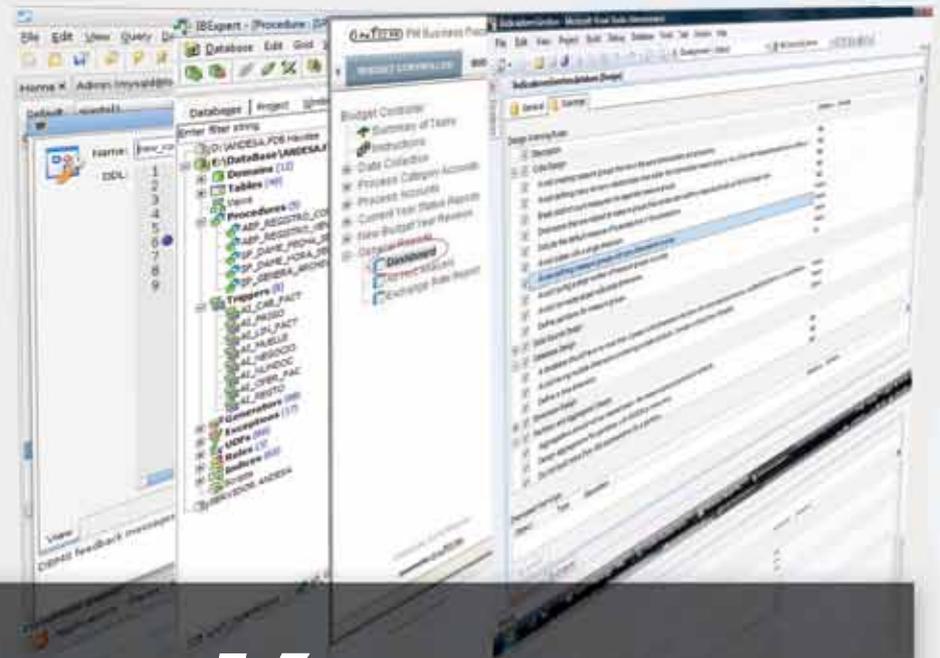
- **Más visitantes:** Las empresas siempre quieren captar nuevos clientes por lo que el armado de un sitio web

accesible logrará que sea visitado por distintas personas independientemente de sus capacidades. Evaluar si la página que estamos creando cumple con los estándares nos asegura que será visualizado por cualquier navegador y por tanto sumará más visitas y más tiempo de lectura en la página.

- **Costos:** Todos quieren lo mejor pero sin gastar tanto. Mientras que muchos clientes plantearán que este paso podría implicar un costo, la realidad es que no es tanto para ellos como para los desarrolladores que tendrán que capacitarse antes de ponerse a trabajar. Desarrollar y mantener una web accesible es menos costoso que una inaccesible, dado que la primera no contendrá tantos errores y será más fácil de actualizar.

- **Rapidez:** Actualmente la velocidad que ofrecen las nuevas tecnologías no dan lugar a esos sitios que demoran en cargar porque terminan por ahuyentar al usuario lector. Por eso un sitio accesible que presenta el contenido separado de su presentación mediante CSS reduce el tamaño de las páginas y el tiempo de carga haciéndolas más rápidas y fáciles de navegar.

Si existiera una herramienta que convirtiera a las ciudades, los edificios, museos o escuelas en espacios accesibles, entonces todos pediríamos a los arquitectos y decoradores que no dejen de lado esta posibilidad. De la misma forma, el universo de las webs cuenta con una opción para crear contenidos destinados para todas las personas por igual. Es cuestión de ponerlos en práctica y ser parte de un mundo sin diferencias. 



Introducción a las bases de datos relacionales (IV)

Juan Gutmann
juan.gutmann@pixelscode.com

En el número anterior de Pixels & Code iniciamos en esta sección las prácticas con PostgreSQL. Comenzamos por una corta guía para instalar este excelente motor de base de datos de código abierto, y detallamos cómo emplear psql, el cliente de línea de comandos. Ha llegado el momento de comenzar a trabajar con el motor propiamente dicho. Avanzaremos en la creación de las estructuras destinadas a contener los datos, que se definen con un sub-grupo de instrucciones del lenguaje SQL conocido como DDL, "Data Definition Language" (lenguaje de definición de datos). Como hemos visto en las primeras entregas de esta serie, cuando hablamos de los aspectos teóricos del mundo de las bases de datos relacionales, modelar adecuadamente las entidades con las que vamos a trabajar es fundamental para asegurar la posterior integridad de la información almacenada. Demos entonces los primeros pasos para aprender a trabajar con el lenguaje SQL, que es el corazón de prácticamente todos los motores relacionales.

UN BREVE RECORDATORIO

Asumimos que han hecho los pasos explicados en la tercera parte de este cursillo, y por lo tanto tienen instalado el motor PostgreSQL, y han creado exitosamente un cluster destinado a alojar nuestras bases de datos. De no ser así, pueden obtener en el sitio de la revista el PDF del número anterior, donde se detallan estos pasos, ya que son indispensables para poder realizar las prácticas que haremos de aquí en adelante. Recordemos también que bajo Windows, para poder ver sin inconveniente los caracteres propios de nuestro idioma español (como acentos y eñes), debe cambiarse el fuente (font) de la ventana de línea de comandos con la que trabajemos a “Lucida Console”. Pueden hacerlo desde las propiedades de dicha ventana (se acceden con ALT+Barra espaciadora). Antes de arrancar el cliente en modo texto, deben también cambiar el juego de caracteres a uno que soporte nuestro idioma, lo que se consigue ejecutando el comando “chcp 1252”. Finalmente, luego de ubicarse en el subdirectorío de binarios (bin) de la carpeta donde hayan instalado el motor (en nuestro caso lo hicimos en D:\Desa\pgsql\), se ejecuta el cliente psql de esta forma: “psql -U postgres”.

LA BASE DE PRÁCTICAS

Cumplido lo anterior, nos encontraremos dentro de la consola de PostgreSQL, que esperará el ingreso de órdenes mostrando el prompt “postgres=#”. Recordemos que escribiendo el comando “\list” o su apócope “\l” obtenemos una lista de las bases de datos que componen el cluster. En un cluster recién creado, contamos con tres bases de datos que son internas y necesarias para el correcto funcionamiento del motor: postgres, template0 y template1. En primer lugar, entonces, vamos a crear una nueva base de datos, que utilizaremos para realizar las prácticas. Escribimos:

```
CREATE DATABASE practicas;
```

El motor responde lacónicamente: “CREATE DATABASE”. Aunque no es muy “verbose” que digamos, nos está confirmando la correcta ejecución de la orden solicitada. Como observación al margen, aclaramos que es buena idea evitar los caracteres propios del español en los nombres de las entidades y sus atributos (es decir, las tablas y sus campos). Aunque PostgreSQL los soporta perfectamente, tanto en los valores de los datos almacenados como en los identificadores de los objetos de la base, en estos últimos es conveniente evitarlos, ya que en algún caso nos tocará trabajar con un equipo o motor cuya versión o configuración no los soporte, y si necesitamos implementar objetos ya creados de esta forma nos ganaremos un gran dolor de cabeza. Hecha esta aclaración, ejecutamos nuevamente la orden “\l” y veremos en el listado que ahora también figura la base “practicas”. Ahora debemos indicarle al cliente de consola que queremos trabajar con esta base, lo cual hacemos con el comando “\connect practicas” (o “\c practicas”). PostgreSQL responderá: “Ahora está conectado a la base de datos «practicas».” Observen que el prompt ha cambiado a: “practicas=#”, indicando que las órdenes que se ejecuten de aquí en adelante se aplicarán en el contexto de esta base.

EL MODELO DE DATOS DE PRÁCTICA

Vamos a definir un modelo de datos sencillo, que en un principio estará compuesto de tres tablas solamente: empleado, cargo y salario. Al momento de crearlas, aplicaremos las reglas de integridad referencial que hemos visto en las primeras dos entregas, al cubrir los aspectos teóricos del cursillo. Notarán que para los nombres de las tablas, hemos elegido una palabra en singular. Aunque lo corriente es emplear palabras en plural, como “empleados” (ya que esta tabla guardará información sobre los empleados de una hipotética empresa) lo académicamente correcto es emplear las palabras en singular, ya que estamos nombrando una entidad integrante de nuestro modelo de datos: el empleado. La orden del subconjunto de sentencias SQL perteneciente al DDL que utilizaremos para crear entidades (es decir, tablas) es “CREATE TABLE”. Debido a que estas tres tablas que vamos a generar tienen dependencias entre sí, existe un único orden posible en el que deben ser creadas para evitar errores. Comenzaremos entonces por la tabla “cargo”.

“Es buena idea evitar los caracteres propios del español en los nombres de las entidades y sus atributos (es decir, las tablas y sus campos). Aunque PostGreSQL los soporta perfectamente, tanto en los valores de los datos almacenados como en los identificadores de los objetos de la base, en estos últimos es conveniente evitarlos, ya que en algún caso nos tocará trabajar con un equipo o motor cuya versión o configuración no los soporte.”

CREANDO LA TABLA CARGO

Escribimos en el prompt de psql:

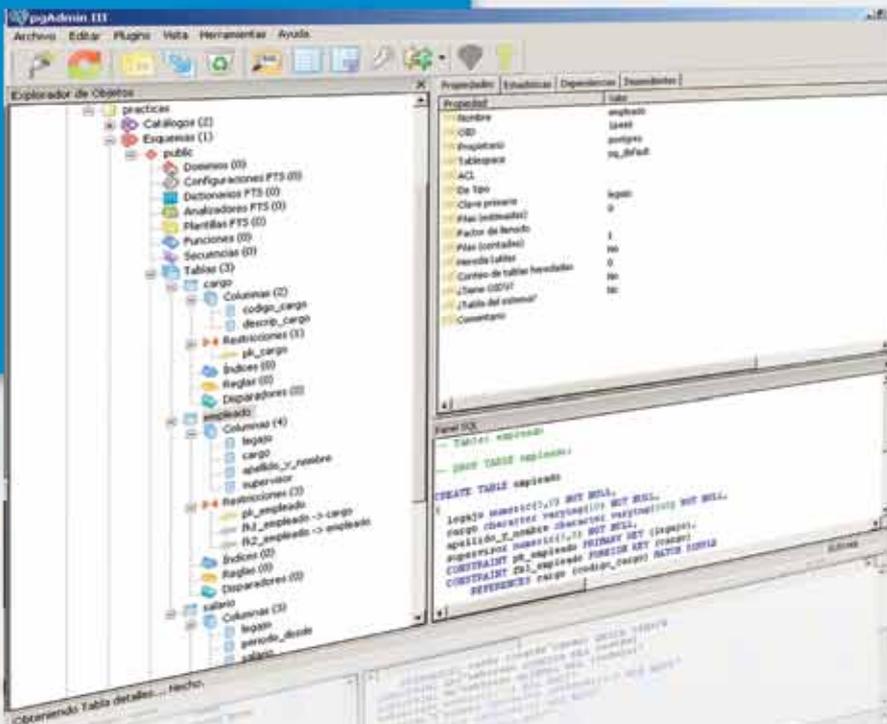
```
CREATE TABLE cargo (  
codigo_cargo VARCHAR(10) NOT NULL,  
descrip_cargo VARCHAR(50) NOT NULL,  
CONSTRAINT pk_cargo PRIMARY KEY (codigo_cargo)  
);
```

La respuesta de PostGres será:

```
NOTICE: CREATE TABLE / PRIMARY KEY creará el índice implícito «pk_cargo»  
para la tabla «cargo»  
CREATE TABLE
```

Analicemos la sentencia. Para la tabla “cargo” hemos especificado tres atributos: dos campos y una constraint (restricción). Para cada campo, debemos indicar su tipo de datos, la precisión del mismo, y si admite o no valores nulos (es decir, que en algunos registros dicho campo pueda permanecer vacío). Esta última característica (la admisión o no de los valores nulos para el campo) es también una constraint, pero a nivel de campo. Los dos campos de esta tabla son de tipo carácter (es decir, pueden contener una combinación de letras, números y símbolos). El primero, `codigo_cargo`, puede almacenar en cada registro hasta un máximo de 10 caracteres, mientras que el segundo, `descrip_cargo`, puede alojar hasta 50. Como el nombre de la tabla y sus campos

lo indican, aquí almacenaremos una lista de todos los puestos que puede tener un empleado en una empresa, nominándolos con un código breve, pensado para uso interno del sistema, y asociándole al mismo una descripción más larga, que será la que se le muestre al usuario. PostGres soporta tres tipos de datos distintos para almacenar cadenas de caracteres: CHAR, VARCHAR y TEXT. Por el momento nos limitaremos a trabajar con los primeros dos. La diferencia entre ambos es la siguiente: mientras que los valores almacenados en campos cuyo tipo de dato sea CHAR tienen una longitud fija, los que guardemos en tipos VARCHAR tendrán longitud variable. En ambos casos la máxima cantidad de caracteres posibles estará dada por el número indicado entre paréntesis luego del tipo de datos, conocido como “precisión” del tipo de datos. Si en un campo de tipo VARCHAR de precisión 10 guardamos



el dato "PEPE", se guardarán en dicho campo esos cuatro caracteres y nada más. A bajo nivel, algunas bases de datos emplean un byte extra para indicar la finalización del valor de longitud variable, pero esto es totalmente transparente para nosotros y sólo nos importa saber que se guarda el dato tal cual fue introducido. En cambio, si en un campo de tipo CHAR(10) guardamos el dato "PEPE", el valor que finalmente se almacena en realidad será "PEPE ", es decir, los cuatro caracteres de "PEPE" seguidos por seis espacios en blanco (ASCII 32), hasta completar el máximo de 10 caracteres indicado por la precisión. Esta es una diferencia sutil pero fundamental, ya que si posteriormente comparamos un campo de tipo VARCHAR con un campo de tipo CHAR que contengan el mismo dato, la base los considerará distintos. De hecho ni siquiera coincide la longitud de ambos valores: 4 en el primer caso, 10 en el segundo.

CLAVES PRIMARIAS

Habrán notado que luego de los campos, especificamos un atributo de tipo constraint. En este caso es una clave primaria, compuesta por el campo codigo_cargo. Esta restricción implica que no puede haber dos registros en la misma tabla con un mismo valor en el campo codigo_cargo, y su fin es identificar unívocamente cada uno de los registros de esta tabla. Como regla general (aunque por supuesto, toda regla tiene sus excepciones), toda tabla siempre debe tener clave primaria. Es condición necesaria para el o los campos integrantes de esta clave que no admitan valores nulos. Junto con la creación de la clave primaria, y tal como lo informó PostGres al confirmar la creación de la tabla, se crea en forma implícita para la tabla un índice único, que indexa la misma por el o los campos especificados como clave. La función de este índice es acelerar las búsquedas que realicemos sobre la tabla. Pero no nos adelantemos, ya que cubriremos con más detalle el funcionamiento de los índices en próximas entregas. Finalizamos el análisis observando que la cláusula empleada en esta orden para crear la clave primaria es una de las varias formas de hacerlo que hay en PostGres. También podríamos haberla creado especificando las palabras "PRIMARY KEY" en forma de constraint para el campo respectivo, luego de aclarar que el mismo es NOT NULL, pero elegimos esta otra forma por dos razones. La primera, es que hacerlo de esta forma nos permite asignarle un nombre a la constraint de clave primaria, lo que nos será de utilidad en un futuro. Con la otra sintaxis, la constraint también tendrá un nombre, pero generado automáticamente por el sistema, lo que es menos deseable. La segunda razón es que esta sintaxis es compatible con la empleada por el motor Oracle. Como aclaramos hace un tiempo atrás, elegimos para el cursillo a PostGres entre otros motivos por su alto grado de compatibilidad con este motor, probablemente el más poderoso del mercado, caballito de batalla de casi todas las grandes empresas del mundo para almacenar su información digital. Por ende, cuanto más familiarizados estemos con las características de Oracle, más allanado tendremos el camino el día que nos toque trabajar con él.

CONSULTAR OBJETOS EXISTENTES EN LA BASE

Si escribimos en la consola de PostGres el comando "\d", que lista las tablas, vistas y secuencias existentes en la base, veremos:

```
Listado de relaciones
Esquema | Nombre | Tipo | Dueño
-----+-----+-----+-----
public | cargo | tabla | postgres
(1 fila)
```

Para ver el detalle de una de las tablas listadas, ingresamos "\d nombretabla", en este caso "\d cargo", y la respuesta será:

```
Tabla «public.cargo»
Columna | Tipo | Modificadores
-----+-----+-----
codigo_cargo | character varying(10) | not null
descrip_cargo | character varying(50) | not null
Índices:
"pk_cargo" PRIMARY KEY, btree (codigo_cargo)
```

PostgreSQL Data Types	
BIGINT INT8	Integer, 8 byte, signed
BIGSERIAL SERIALS	Autoincrementing integer, 8 byte
BIT(n)	fixed-length bit string
BOOLEAN BOOL	Logical Boolean (true/false)
BOX	Rectangular box in the plane
BYTEA	Binary data (byte array)
CHAR(n)	Fixed-length character string
CIDR	IPv4 or IPv6 network address
CIRCLE	Circle in the plane
DATE	Calendar date (year, month, day)
DOUBLE PRECISION FLOAT8	Double precision floating-point number
INET	IPv4 or IPv6 host address
INTEGER INT INT4	Integer, 4 byte, signed
INTERVAL(n)	Time span
LINE	Infinite line in the plane
LSEG	Line segment in the plane
MACADDR	MAC address
MONEY	Currency amount
NUMERIC DECIMAL (precision, scale)	Exact numeric of selectable precision
PATH	Geometric path in the plane
POINT	Geometric point in the plane
POLYGON	Closed geometric path in the plane
REAL FLOAT4	Single precision floating-point number
SERIAL	Autoincrementing integer, 4 byte
SMALLINT INT2	Integer, 2 byte, signed
TEXT	Variable-length character string
TIME [with time zone]	Time of day
TIMESTAMP [with time zone]	Date and time
VARBIT(n)	Variable-length bit string
VARCHAR(n)	Variable-length character string

LA TABLA EMPLEADO

Ahora procedemos con la creación de la segunda tabla, que contendrá la información de los empleados de la empresa. Tipeamos lo siguiente:

```
CREATE TABLE empleado (
  legajo NUMERIC(5) NOT NULL,
  cargo VARCHAR(10) NOT NULL,
  apellido_y_nombre VARCHAR(100) NOT NULL,
  supervisor NUMERIC(5) NOT NULL,
  CONSTRAINT pk_empleado PRIMARY KEY (legajo),
  CONSTRAINT fk1_empleado FOREIGN KEY (cargo) REFERENCES cargo
  (codigo_cargo),
  CONSTRAINT fk2_empleado FOREIGN KEY (supervisor) REFERENCES empleado
  (legajo)
);
```

Con respecto a la tabla “cargo”, tenemos dos novedades. Por un lado, aparecen dos campos (legajo y supervisor) con un nuevo tipo de datos: NUMERIC. PostGres soporta diversos tipos de datos numéricos. Para un valor numérico entero, como los que vamos a emplear en estos atributos, y dependiendo de la cantidad máxima de empleados que consideramos que puede tener la empresa a modelar, podríamos usar el tipo “smallint”, que soporta números enteros con signo de hasta 2 bytes de capacidad (es decir, los valores posibles son todos los comprendidos entre -32768 y +32767). Como para representar números de legajo sólo nos sirven los valores positivos, 32767 parecen algo escasos. Si elegimos el tipo “integer”, de 4 bytes, el rango se amplía considerablemente: de -2147.483.648 a +2147.483.647. 2147 millones es algo excesivo, y estaríamos desaprovechando espacio de almacenamiento, ya que cada campo de cada registro que tenga este tipo de datos requiere 4 bytes para almacenar el valor, así almacenemos el valor “0”. Es mucho más eficiente, y también más compatible con Oracle, optar por el tipo de datos “numeric”, que nos permite elegir a gusto la precisión y escala de los valores numéricos a almacenar. La precisión, al igual que con los tipos de datos carácter, indica la máxima cantidad de dígitos que puede tener un valor numérico. La escala indica la cantidad de dígitos que se ubicarán a la derecha del punto decimal; de omitirlo, estamos definiendo que sólo se aceptarán números enteros. Indicando NUMERIC(5) el rango de valores posibles va de -99.999 a +99.999. Si hubiéramos asignado un valor a la escala, por ejemplo definiendo NUMERIC(5,2), los valores posibles irían de -999,99 a +999,99.

CLAVES FORÁNEAS

La otra novedad respecto a la tabla “cargo” es el empleo en la tabla “empleado” de dos claves foráneas. Las definimos a nivel teórico en la segunda entrega de la serie. Sin embargo, recordemos en pocas palabras que una clave foránea referencia a la clave primaria de otra tabla, impidiendo la carga del campo declarado como Foreign Key (FK) si el valor ingresado no existe como Primary Key (PK) en la tabla referenciada. Si así lo deseamos, esta restricción puede ser de dos vías. En ese caso, la primera FK que creamos, para el campo cargo, establece que si en la tabla “hija” (empleado) existe un registro con el valor de

cargo igual a "GTEGRAL", no podrá eliminarse de la tabla "padre" (a la que hace referencia la FK), en este caso "cargo", el registro cuya clave primaria (el campo codigo_cargo) tenga idéntico valor. De esta forma se impide que queden valores "huérfanos" en las tablas hijas, al desaparecer de la tabla padre el registro al que deben referenciar. La segunda FK de esta tabla es más particular, ya que referencia a otro campo de la misma tabla: el "supervisor" de un empleado debe contener el número de legajo de otro empleado ya existente en la tabla. Esto también nos asegura que esta tabla debe llenarse jerárquicamente: no puede cargarse un empleado si previamente no hemos cargado su supervisor. Para el primer empleado que se cargue, debe sí o sí ingresarse el mismo legajo tanto en "legajo" como en "supervisor" para que se admita el registro. Por lo tanto, el primer empleado en ser cargado en esta tabla será quien tenga el puesto de mayor jerarquía dentro de la estructura, y su supervisor será él mismo. Noten que si ahora pedimos una descripción de la tabla "cargo" con el comando "\d cargo", además de los atributos que vimos al ejecutar esta misma orden luego de crear dicha tabla, ahora aparecerá:

```
Referenciada por:  
TABLE "empleado" CONSTRAINT "fk1_empleado" FOREIGN KEY  
(cargo) REFERENCES cargo(codigo_cargo)
```

De esta forma, PostGres nos indica que la PK de esta tabla es referenciada por la FK "fk1_empleado" de la tabla "empleado".

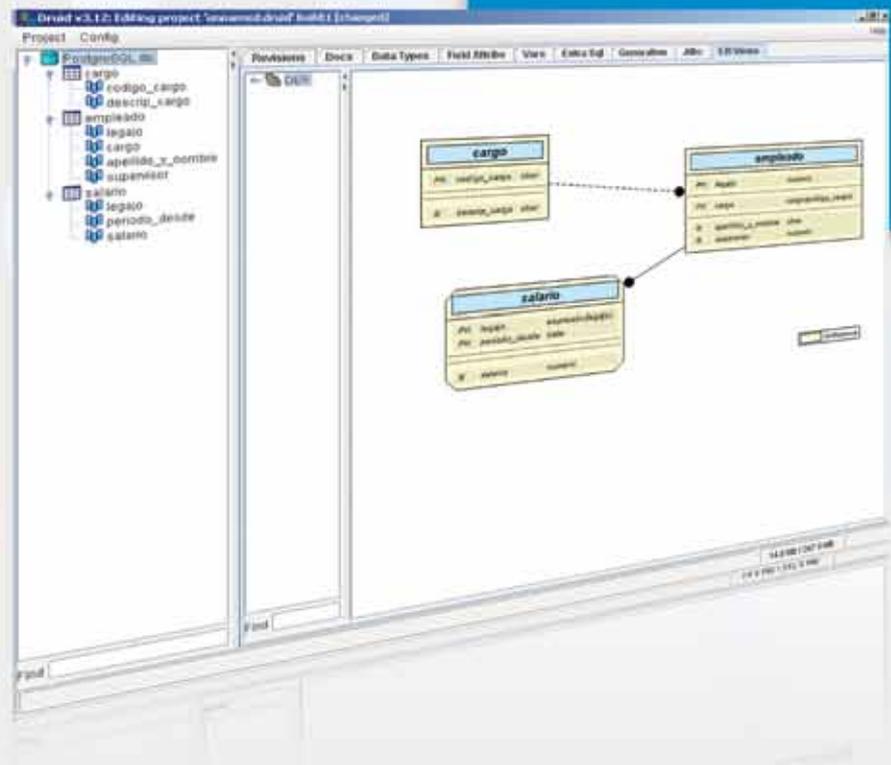
LA TABLA SALARIO

Vamos a crear la última tabla de la presente lección, mediante el siguiente comando:

```
CREATE TABLE salario (  
legajo NUMERIC(5) NOT NULL,  
periodo_desde DATE NOT NULL,  
salario NUMERIC(9,2) NOT NULL,  
CONSTRAINT pk_salario PRIMARY KEY (legajo, periodo_desde),  
CONSTRAINT fk1_salario FOREIGN KEY (legajo) REFERENCES empleado (legajo)  
);
```

La única novedad en este caso está en el campo "periodo_desde", cuyo tipo de datos es DATE. PostGres ofrece varios tipos de datos para almacenar horas y fechas. Para este caso, nos alcanza con uno que permita almacenar día, mes y año, ya que no nos interesa guardar la hora, minutos y segundos para establecer a partir de qué día un empleado en particular pasa a ganar un salario de valor X.

"Para la tabla "cargo" hemos especificado tres atributos: dos campos y una constraint (restricción). Para cada campo, debemos indicar su tipo de datos, la precisión del mismo, y si admite o no valores nulos (es decir, que en algunos registros dicho campo pueda permanecer vacío)."





“Pero hay una forma más práctica de hacerlo, una suerte de “CTRL-Z”, que permite deshacer todos los cambios realizados a partir de cierto punto, y es mediante el empleo de transacciones.”

CARGANDO LAS TABLAS

Para ingresar valores en las tablas, recurrimos a la sentencia INSERT, perteneciente al subconjunto de instrucciones de SQL conocido como DML: Data Modification Language. Ingresamos lo que sigue:

```
INSERT INTO cargo  
(codigo_cargo, descrip_cargo)  
VALUES  
(‘GTEGRAL’, ‘Gerente General’);
```

```
INSERT INTO cargo  
(codigo_cargo, descrip_cargo)  
VALUES  
(‘SUPERV’, ‘Supervisor’);
```

```
INSERT INTO cargo  
(codigo_cargo, descrip_cargo)  
VALUES  
(‘ADMIVO’, ‘Administrativo’);
```

Luego de cada uno de los Inserts, PostGres responde: “INSERT 0 1” confirmando la correcta ejecución de las sentencias. Podemos corroborarlo escribiendo el query “SELECT * FROM cargo” y la base responde:

```
codigo_cargo | descrip_cargo  
-----+-----  
GTEGRAL      | Gerente General
```

```
SUPERV      | Supervisor  
ADMIVO      | Administrativo  
(3 filas)
```

Ahora vamos a ingresar un registro a la tabla empleado, de esta manera:

```
INSERT INTO empleado  
(legajo, cargo, apellido_y_nombre, supervisor)  
VALUES  
(1, ‘GTEGRAL’, TO_DATE(‘01/01/2011’, ‘dd/mm/yyyy’), 1);
```

Si hubiéramos intentado ingresar al empleado antes de llenar la tabla cargo, obtendríamos el siguiente mensaje de error:

ERROR: inserción o actualización en la tabla «empleado» viola la llave foránea «fk1_empleado»
DETALLE: La llave (cargo)=(GTEGRAL) no está presente en la tabla «cargo».

Esto se debe al empleo de integridad referencial mediante claves foráneas, tal cual explicamos más arriba.

TRANSACCIONES

¿Qué ocurre si queremos deshacer algunos de los cambios hechos en los datos mediante las órdenes DML? Naturalmente, podríamos modificarlos mediante nuevas órdenes DML; por ejemplo, borrando los nuevos registros con la sentencia DELETE. Pero hay una forma más práctica de hacerlo, una suerte de “CTRL-Z”, que permite deshacer todos los cambios realizados a partir de cierto punto, y es mediante el em-

pleo de transacciones. En PostgreSQL, una transacción comienza cuando escribimos la sentencia BEGIN. Probemos lo siguiente:

```
BEGIN;  
  
INSERT INTO salario  
  (legajo, periodo_desde, salario)  
VALUES  
  (1, TO_DATE('01/01/2011', 'dd/mm/yyyy'), 8192.20);  
  
INSERT INTO salario  
  (legajo, periodo_desde, salario)  
VALUES  
  (1, TO_DATE('01/02/2011', 'dd/mm/yyyy'), 8500.60);
```

Si ahora con un query (“SELECT * FROM salario”) consultamos la tabla salario, veremos:

```
legajo | periodo_desde | salario  
-----+-----+-----  
      1 | 2011-01-01    | 8192.20  
      1 | 2011-02-01    | 8192.20  
(2 filas)
```

Pero probemos ahora escribir:

```
ROLLBACK;
```

PostGres responde “ROLLBACK”. Y al ejecutar nuevamente el mismo query sobre salario:

```
legajo | periodo_desde | salario  
-----+-----+-----  
(0 filas)
```

¡Sorpresa! Los inserts realizados previamente desaparecieron. Entonces, podemos decir que una transacción comienza cuando así lo especificamos mediante la orden “BEGIN”, y termina cuando escribimos las sentencias “ROLLBACK” o “COMMIT”. Si finalizamos la transacción con Rollback (“retrotraer” o “tirar para atrás”), todos los cambios a los datos realizados durante la transacción se deshacen. Por el contrario, si lo hacemos con Commit (“Confirmar”), los “grabamos” definitivamente. Si ahora ejecutamos nuevamente la transacción anterior, pero la finalizamos con un “COMMIT;” en lugar de con un ROLLBACK, al ejecutar nuevamente el query sobre salario veremos que las filas han permanecido. Aclaremos que todas las instrucciones DDL, como CREATE TABLE, tienen un COMMIT implícito y no pueden deshacerse. Esto es así en todos los motores de bases de datos relacionales. En cuanto a las órdenes DML (INSERT, UPDATE y DELETE), en PostGres, cuando no fueron escritas dentro de una transacción iniciada explícitamente con “BEGIN;”, también poseen un Commit implícito.

SE HACE CAMINO AL ANDAR

Ya inmersos de lleno en las prácticas con PostGreSQL, a partir de aquí iremos aplicando los conocimientos prácticos que deben manejar distintas ramas de expertos en informática: desde administradores y operadores de bases de datos, hasta desarrolladores, especialmente los que se dediquen a crear o mantener aplicativos basados en la arquitectura cliente/servidor. En las próximas entregas, seguiremos profundizando en el uso de sentencias DML, y comenzaremos a ver Stored Procedures (Procedimientos Almacenados), que nos permitirán trabajar con el lenguaje programático de PostGres: PL/pgSQL. Hasta entonces.



Cómo encauzar nuestra estrategia de negocios en las redes sociales

Cristian Hernán Gaitano Ornia

cristian.gaitanoornia@pixelscode.com
@cerge74



Es muy importante que podamos aprovechar al máximo esta inmejorable oportunidad que nos brinda la nueva corriente de negocios, para estar más cerca de las necesidades de nuestros clientes. A continuación se indican algunos aspectos a tener en cuenta.

Lo que se presenta como una excelente chance de recibir y catalogar la opinión de nuestra clientela en una red social, debe encauzarse y analizarse de la manera más óptima para no terminar siendo un vendaval de información que pasa a través nuestro y del cual no sólo no nos sentimos parte, sino

que tampoco lo podemos analizar y registrar en forma conveniente. En ese sentido, recibiremos tanto opiniones de nuestros clientes fieles y habituales, como también de personas interesadas en nuestros productos o servicios. Es muy importante darle especial trascendencia a este último segmento, por cuanto



esos compradores potenciales nos pueden ayudar muchísimo a través de sus comentarios y críticas a mejorar eso que deseamos comercializar o bien, ya estamos vendiendo. Ahora bien, ¿cómo podemos registrar y analizar ese flujo de información? Si bien es cierto que las redes sociales fueron pensadas para interactuar entre personas y no sobre marcas o empresas, es real que el crecimiento de las mismas en los últimos años cambió radicalmente el concepto y uso de ellas, y resulta imperativo poder analizar y utilizar esas nuevas corrientes de información no sólo para mejorar la relación con los clientes, sino también las utilidades de la empresa. En realidad, la medición de esa corriente de datos que surge de los nuevos ámbitos sociales representa a veces una tarea compleja, por cuanto los resultados suelen vislumbrarse a largo plazo y es por tanto complicado calcular en forma precisa el retorno de la inversión (o ROI), que esencialmente se calcula haciendo Beneficios - Costos.

EL IOR

Es allí donde surge un tipo de cálculo que hace foco en el impacto de las relaciones entre las marcas y sus seguidores, el cual se denomina IOR (Impact of Relationship) y es ahora usualmente muy utilizado para medir resultados de social media. El

IOR, creado por la experta Johana Calvanti, se define como un eficiente sistema de medición de nuestra presencia en los medios sociales y gestiona esencialmente cuatro elementos fundamentales, a saber:



La medición de esa corriente de datos que surge de los nuevos ámbitos sociales representa a veces una tarea compleja, por cuanto los resultados suelen vislumbrarse a largo plazo y es por tanto complicado calcular en forma precisa el retorno de la inversión (o ROI)."

-La AUTORIDAD: elemento representado por dos aspectos importantes, por un lado, el número de enlaces o menciones hacia nuestros contenidos y acciones. Por el otro, la relevancia de esos links o referencias y el valor que se les da. Quizás la más importante pero también, la más compleja

de medir. De todas formas, y para dar un ejemplo, pensemos que podemos valorar con una cierta cantidad de puntos la referencia de un blog, web o publicación importante, la cual será mucho menor si dicha cita parte de un sitio mucho menos conocido.

-Su INFLUENCIA: posiblemente la segunda en importancia, la influencia se centra en el número de seguidores, clientes o suscriptores de nuestros contenidos en las distintas redes sociales donde participamos. En función de la importancia que le demos a cada red o canal podemos calificar, por ejemplo, con más puntos a un seguidor de nuestro blog o website que a uno que sigue nuestros videos institucionales o novedades en YouTube.

-La PARTICIPACIÓN que ha generado nuestra presencia: se centra en los comentarios generados a partir de nuestras acciones y focalizados en nuestra plataforma, es decir, hace foco en la cantidad de veces que nuestros clientes virtuales interactúan en nuestra cuenta o grupo de red social. Es posiblemente la variable que más IOR genera, y si seguimos el ejemplo anterior podríamos decir que un comentario en nuestra página web tendría más puntos que un comentario en Facebook.

-El TRÁFICO generado: vale decir, la cantidad de visitas a nuestras cuentas

de redes sociales. Asimismo, en esta variable podemos englobar otras también importantes, como el número de usuarios únicos y repetidores y la media de tiempo en la web.

En ese sentido podemos monitorizar ese tráfico en las redes sociales mediante diversas herramientas que nos facilitan todo tipo de parámetros que podremos analizar con cierta periodicidad. En ese segmento, existen diversas herramientas online que permiten hacer un seguimiento de la red para encontrar opiniones sobre la reputación de nuestra empresa.

A la hora de calcular el IOR de la mejor forma, es menester poder fijar algunos pasos importantes que necesitamos verificar adecuadamente antes de entrar a una red social. Estos pasos pueden ser resumidos de la siguiente forma:

- Establecer fehacientemente por qué queremos estar en una red social con nuestra campaña, marca o empresa. Las respuestas en ese sentido podrían ser muchas: promocionar un nuevo producto, captar nuevos clientes, entrar en nuevos mercados, fidelización, etc. En ese sentido las redes sociales representan una plataforma inmejorable, pero a veces también, pueden resultar un arma de doble filo si no la sabemos utilizar adecuadamente.
- Medir apropiadamente el impacto analizando las cuatro variables antes explicadas.
- Cada valor cambia dependiendo del objetivo que tenemos, el sector al que nos dedicamos.

Así es que el IOR asoma como una invaluable herramienta de uso estratégico a nivel interno, ya que



permite diagramar el grado de importancia que cada empresa le da a esas variables. Es decir, destacar que cada uno de esos aspectos posee una valoración absolutamente individual que depende no sólo de cada empresa, sino de cada campaña, de cada estrategia, de los objetivos de cada marca y del segmento al que se dirige. Tal como vimos diferentes niveles de puntuación para cada variable, podemos pensar que así como una empresa puede hacer mayor hincapié en la presencia de su campaña invernal en Facebook, otra pone especial énfasis en su próxima liquidación en Twitter, por lo cual, cada compañía le da un lugar y valor diferente a cada variable dentro de su propia estrategia.

En estos casos de uso interno del IOR, y al no tener un punto de comparación externo, debemos tener en claro que pensaremos en esta herramienta como algo dinámico en el tiempo, y medir la evolución de nuestra presencia en esos medios

sociales en esos términos, de forma tal que podamos mensurar los resultados en forma objetiva, verificando en qué variables exponemos deficiencias y por ende, en cuáles debemos enfocarnos más. Vale decir, por ejemplo, de nada sirve tener un alto tráfico si nuestros contenidos no despiertan el interés y la participación de nuestros seguidores, accionar que se expande y se referencia a lo largo y ancho de dichas redes.

Como recién hablábamos de puntos de comparación externos, es importante destacar que con el IOR también podemos desarrollar acciones comparativas con otras marcas del mismo sector para así evaluar el alcance de nuestra estrategia en forma global. Esto se logra estableciendo precisas pautas de parametrización fija con cada indicador involucrado, indicadores que dependiendo del sector analizado y el nivel de segmentación buscado, permitirán obtener una visión estratégica más precisa del ámbito en el



El IOR asoma como una invaluable herramienta de uso estratégico a nivel interno, ya que permite diagramar el grado de importancia que cada empresa le da a esas variables.”

cual se mueve nuestra empresa.

HERRAMIENTAS

En ese ámbito, posiblemente una muy interesante propuesta sea la de Start PR (<http://startpr.com>), red

que se alimenta de medios como Twitter, WordPress, Flickr, YouTube, Technorati y Google Blog Search y si bien es paga, tiene una utilidad a todas luces muy conveniente. En base a esas herramientas de tracking y análisis de tráfico, podemos ir estudiando la variable de tráfico e ir asignando fácilmente diferentes puntajes según la visita proceda de una u otra red social.

Pero no todo es pago, y es bueno conocer que tenemos alternativas gratuitas en ese sentido ya que también podemos utilizar lectores de feeds y utilizar el RSS como una fuente de monitoreo importante. Netvibes (netvibes.com) ofrece un lector de feeds y también widgets (agregados) que nos permiten acelerar dicho procesamiento. Asimismo, tenemos diversas webs

de búsquedas orientadas justamente a búsquedas específicas dentro de redes sociales o blogs.

Entre las más importantes podemos destacar dos plataformas de búsquedas múltiples muy ágiles como www.socialmention.com, blogs.icerocket.com, así como otras más orientadas a ámbitos específicos (blogpulse.com o searchtwitter.com).

Como hemos visto, existen alternativas y variantes muy útiles a la hora de poder aprovechar esa genial herramienta que surge del encuentro del cliente con la empresa en el marco de una red social. Si deseamos estar un paso por delante de nuestra competencia es imperioso, pero también apasionante, poder saber utilizarlas de la manera más conveniente. **P**



El poder del scripting en GNU/Linux, al servicio de la web

Armá un sitio de video tube en cinco minutos

Juan Gutmann
juan.gutmann@pixelscode.com

*Ya no cabe ninguna duda que los Sistemas Operativos basados en UNIX, como GNU/Linux o los de la familia BSD, son la opción más adecuada para servidores. Su estabilidad y flexibilidad para estos fines superan por varios cuerpos a cualquier integrante de la familia Microsoft. Sin embargo, todavía hay un intenso debate sobre cuál es la plataforma más adecuada para ejecutar sobre estaciones de trabajo. La verdad es que en esta situación, el tema pasa a depender bastante del uso que se le va a dar a la workstation. A la hora de desarrollar, existen para Windows herramientas poderosas, que permiten un grado de productividad muy importante. No obstante, en nuestra opinión los descendientes de UNIX también aquí sacan ventaja, y nos proponemos dar una muestra en esta nota. Las herramientas "userland" de los S.O. herederos de *NIX son extremadamente poderosas. Particularmente, muchas herramientas de línea de comando, aunque no sean tan amigables, permiten hacer mucho en poco tiempo. A continuación, les entregamos un pequeño ejemplo que lo demuestra. Mediante un script, generaremos automáticamente y en escasos minutos las páginas HTML para crear un website sencillo de video-tube, una modalidad de entrega de contenidos extremadamente popular en estos tiempos.*

INGREDIENTES NECESARIOS

Vamos a necesitar para crear nuestro website de videos unos pocos elementos. Básicamente, como ya mencionamos, contar con un Sistema Operativo de la familia de clones de UNIX (lo más habitual, sin dudas, es alguna distribución de GNU/Linux) en la que deberemos tener apenas dos componentes extras a los que ya trae una instalación estándar. El primero es el lenguaje de programación Python, que en el caso de muchas distros ya se encuentra preinstalado, e inclusive suele ser un componente vital de las mismas. El otro es un servidor web. En nuestro caso, optamos por el más habitual: Apache. Pero cualquiera de ellos servirá a nuestros fines. Para reproducir los videos en el navegador de los visitantes a nuestro sitio web, optamos por la tecnología Flash, principalmente por su mayor grado de compatibilidad con los distintos web browsers, especialmente con versiones no muy recientes. Aunque el delivery de videos a través de la web mediante HTML 5 es considerado más estándar y efectivo, la realidad es que el soporte a esta nueva tecnología todavía no se ha popularizado. De hecho, todavía se encuentra en desarrollo, y hasta que no se publique su versión final, probablemente Flash siga siendo el "estándar de facto" para la difusión de contenidos en video a través de la web.

JW PLAYER

El primer paso es obtener un reproductor de videos en Flash para emplear en nuestro sitio. Nuestro elegido es el JW Player, que tiene múltiples virtudes. Primero y principal, es software libre. Pero además, es pequeño, eficiente, fácil de "embeber" en el código HTML de las páginas web, y compatible no solamente con videos en formato flash (".flv") sino también con el formato MP4. Se descarga desde su sitio oficial, ubicado en [<http://www.longtailvideo.com/players/jw-flv-player/>]. Una vez descargado el pequeñísimo archivo llamado "mediaplayer-viral.zip", vamos a crear un subdirectorio en el raíz del directorio "PUBLIC HTML", es decir, aquel que toma como raíz para publicar en la web nuestro web server. En el caso de Apache bajo GNU/Linux, suele ser /var/www. Si elegimos para el nuevo subdirectorio el nombre "minitube", entonces vamos a extraer en el directorio /var/www/minitube apenas dos ficheros del archivo .zip que descargamos: el reproductor propiamente dicho, llamado "player.swf", y una breve librería JavaScript que se emplea para "embeber" el reproductor, cuyo nombre es "jwplayer.js".

ARMANDO EL ÁRBOL DE SUBDIRECTORIOS

Dentro del directorio /var/www/minitube vamos a crear tantos subdirectorios como categorías queramos tener para clasificar los videos de nuestro sitio. Para nuestro ejemplo, un pequeño sitio de videos musicales, elegimos tres: Blues, Clásica y Heavy Metal. Entonces, creamos estos tres subdirectorios con el comando mkdir. Para evitar complejizar el script, los nombres de los subdirectorios no deben contener espacios:

```
mkdir /var/www/Blues
mkdir /var/www/Clasica
mkdir /var/www/Heavy
```

Si no están bien configurados los "locales" en el sistema, tendremos problemas también con los acentos y las eñes. Como en algunas distros no es

EN WINDOWS TAMBIÉN ES POSIBLE

*Aunque lo ideal es realizar esta práctica sobre GNU/Linux o *BSD, los que se vean obligados a trabajar únicamente con Windows también pueden hacerlo. Para ello, necesitarán descargar e instalar Cygwin, que no es otra cosa que una selección de paquetes de software para GNU/Linux recompilados para ejecutar en Windows. Mediante el administrador de paquetes que incorpora, pueden instalar todos los paquetes necesarios para realizar esta práctica, como bash, awk, apache y Python. La home page de Cygwin se encuentra en [http://www.cygwin.com/].*

un tema sencillo de solucionar, en ese caso podemos optar por omitirlos directamente, a fines de realizar la práctica con éxito y en poco tiempo.

Creados los subdirectorios, copiamos los videos en cada uno de ellos, según la categoría que les corresponda. Recordemos que tienen que estar codificados en formato FLV o MP4.

EL TEMPLATE

Para crear cada una de las páginas donde se mostrarán los videos, vamos a valernos de un “template” o plantilla. Esto les permitirá a los lectores “customizar” el diseño de la página a su gusto personal. Lo importante es respetar la ubicación y código de los scripts en JavaScript empleados para embeber el video, así como el DIV que será reemplazado por el reproductor flash al correr dichos scripts. También deben mantenerse los tags #TITULO#, #RUTA# y #ARCHIVO#, ya que nuestro script generador de páginas reemplazará esas cadenas literales por los valores correspondientes para cada uno de los videos.

El template que usamos para el ejemplo es bien básico, y siguiendo lo indicando arriba, pueden modificarlo de acuerdo a su propio criterio. Debe grabarse en el directorio “/var/www/minuteube” con el nombre “template.html”.

```
<html>
  <head>
    <title>#TITULO#</title>
    <script type="text/javascript" src="jwplayer.js"></script>
  </head>
  <body>
    <div id='mediaplayer' />
    <script type="text/javascript">
      jwplayer("mediaplayer").setup({
        'flashplayer': 'player.swf',
        'id': 'playerID',
        'width': '480',
        'height': '270',
        'file': '#RUTA#/#ARCHIVO#'
      });
    </script>
  </body>
</html>
```

Habiendo generado el template, si pedimos un listado del directorio raíz del sitio con el comando `ls -l /var/www/minuteube`, veremos algo como:

```
drwxr-xr-x 2 nobody nogroup 4096 jul  4 15:28 Blues
drwxr-xr-x 2 nobody nogroup 4096 jul  4 15:28 Clasica
drwxr-xr-x 2 nobody nogroup 4096 jul  4 15:28 Heavy
-rw-r--r-- 1 nobody nogroup 132411 jun 22 13:11 jwplayer.js
-rw-r--r-- 1 nobody nogroup 106416 jun 22 13:11 player.swf
-rw-r--r-- 1 nobody nogroup      382 jul  4 02:50 template.html
```

```
GNU nano 2.2.4          Archivo: mkhtml.sh
#!/bin/bash

#####
# Generador de HTML para sitio mini-tube v 0.01 #
#####

idx=1

echo "<html>" > index.html
echo "<body>" >> index.html

for dir in $(find . -type d -path "./.*")
do
echo Procesando directorio $dir

cd $dir

for file in ./*.flv ./*.mp4
do
if [ "$file" == "./*.mp4" ] || [ "$file" == "./*.flv" ]; then
continue
fi

echo " Procesando archivo" $file

new=$(echo $file | awk '{print("M0d", $0)}')

titulo=$(echo $file) | awk '{print substr($0,3,length($0)}'
ruta=$(echo $dir) | cut -c 3 |
ruta=$(python -c "import urllib;import sys;print urllib
archivo=$(echo $file) | cut -c 3" |
archivo=$(python -c "import urllib;import sys;print urllib
sed -e "s/#TITULO#/${titulo}/" -e "s/#RUTA#/${ruta}/"

echo "<a href='\"$new.html\"'>${ruta} - ${archivo}</a>" >>
echo "<br>" >> ./index.html

mv $new.html ..
```

“Para reproducir los videos en el navegador de los visitantes a nuestro sitio web, optamos por la tecnología Flash, principalmente por su mayor grado de compatibilidad con los distintos web browsers, especialmente con versiones no muy recientes.”



En lugar de los subdirectorios de nuestro ejemplo, deben ver los creados por ustedes. En cuanto a los demás archivos, si falta alguno, es momento de revisar si hemos completado satisfactoriamente los pasos anteriores. De lo contrario, no podremos completar exitosamente lo que sigue.

SCRIPT GENERADOR DE PÁGINAS

Veamos el script encargado de generar las páginas HTML. Las correspondientes a cada uno de los videos se generarán en base al template, una para cada video que hayamos copiado en los subdirectorios de las categorías que creamos previamente. En cambio, la página índice (index.html) se genera directamente desde el script, y el código HTML se encuentra “hardcodeado” en el mismo. Esto no es lo ideal, ya que es poco flexible y no muy práctico de modificar que digamos, pero decidimos implementarlo de esa manera para simplificar el funcionamiento del script, y también por razones de espacio. Con un poco de conocimientos de scripting en Bash, no es difícil modificarlo para emplear otro template para generar el índice, de forma similar a como se hace con cada una de las páginas de los videos. A continuación exponemos el código del script, el cual deben guardar en el raíz del sitio “tube” con el nombre que preferan (ej. /var/www/mini-tube/mkhtml.sh):

```
#!/bin/bash

#####
# Generador de HTML para sitio mini-tube v 0.01 #
#####

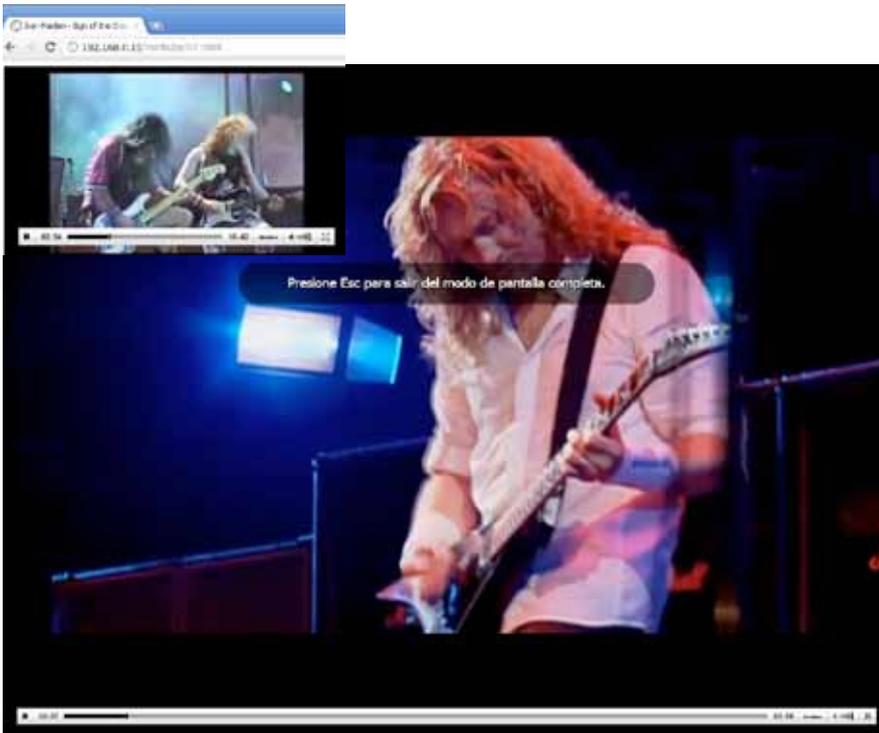
idx=1

echo "<html>" > index.html
echo "<body>" >> index.html

for dir in $(find . -type d -path "./.*")
do
echo Procesando directorio $dir

cd $dir

for file in ./*.flv ./*.mp4
```



```
do
if [ "$file" == "/*.mp4" ] || [ "$file" == "/*.flv" ]; then
    continue
fi
echo "  Procesando archivo" $file
newh=$(echo $(echo $idx | awk '{printf("%02d", $0)}'))
titulo=$(echo $file | awk '{print substr($0,3,length($0) -6)}')
ruta=$(echo $dir | cut -c 3-)
rutau=$(python -c "import urllib;import sys;print urllib.quote(sys.argv[1])" "$ruta")
archi=$(echo $file | cut -c 3-)
archiu=$(python -c "import urllib;import sys;print urllib.quote(sys.argv[1])" "$archi")
sed -e "s/#TITULO#/${titulo}/" -e "s/#RUTA#/${rutau}/" -e "s/#ARCHIVO#/${archiu}/" ../template.html > $newh.html # Nota: Este comando debe escribirse en una sola línea
echo "<a href=\"$newh.html\">\"$ruta - $archi\"</a>" >> ../index.html
echo "<br>" >> ../index.html
mv $newh.html ..
let idx+=1
done
cd ..
done
```

```
echo "</body>" >> index.html
echo "</html>" >> index.html
```

Procedamos a analizar su funcionamiento. Para generar los nombres de las páginas que tendrán embebidos los videos, usamos una variable secuenciadora "idx", que más adelante formatearemos a dos dígitos, aplicando el cero a la izquierda de ser necesario, con ayuda del minilenguaje de procesamiento de texto AWK. Esta variable idx se incrementará en uno tras cada iteración del bucle interno. Por cierto, tenemos dos bucles, uno externo, encargado de recorrer los directorios (cuya lista a iterar confeccionamos gracias al comando `find . -type d -path ".*"`) y otro interno, que procesa cada uno de los archivos de extensiones FLV y MP4 que residan en dichos directorios. La lista del bucle interno tiene una pequeña "pega": como le decimos al comando `for` que queremos iterar por `.*.flv` `.*.mp4`, si no existiese un archivo que `matchee` alguno de estos comodines, se interpreta que en lugar de un comodín es un literal, y se lo agrega a la lista como tal. Lo solucionamos salteando las iteraciones para los elementos literales `.*.flv` y `.*.mp4` con una sentencia condicional `if`. Luego de generar el nombre de la página html que contendrá el video actual que estamos procesando (que se guarda en la variable `newh`), nos valemos de los comandos `awk` y `cut` para formatear las asignaciones a las variables de reemplazo `titulo`, `rutau` y `archiu` en el `template`. Así eliminamos los prefijos `./` y también las extensiones, en el caso de `titulo`. Como `rutau` y `archiu` van a contener URLs, las procesamos con el lenguaje Python, ya que mediante su librería `urllib` podemos "encodear" los caracteres ASCII no estándar, de forma tal que el navegador los entienda sin inconvenientes. Por último, nos valemos de otro procesador de textos, `sed` -cuyo empleo es más veloz y sencillo que `awk`- para realizar el reemplazo en el `template` de los TAGS que hemos creado a tal efecto por sus valores definitivos. Finalizamos la iteración del bucle interno agregando los `hyperlinks` a cada una de las páginas HTML creadas en el `index.html`.

EJECUTAR EL SCRIPT

Habiéndonos ubicado en el directorio raíz del sitio (con `cd /var/www/miniube/mkhtml.sh`), ejecutamos el script escribiendo `./mkhtml.sh`, y luego pedimos un directorio con `ls -l`. Debe mostrarse algo como esto:

```
-rw-r--r-- 1 nobody nogroup 432 jul 4 15:28 01.html
-rw-r--r-- 1 nobody nogroup 432 jul 4 15:28 02.html
-rw-r--r-- 1 nobody nogroup 445 jul 4 15:28 03.html
-rw-r--r-- 1 nobody nogroup 446 jul 4 15:28 04.html
-rw-r--r-- 1 nobody nogroup 431 jul 4 15:28 05.html
-rw-r--r-- 1 nobody nogroup 439 jul 4 15:28 06.html
-rw-r--r-- 1 nobody nogroup 437 jul 4 15:28 07.html
drwxr-xr-x 2 nobody nogroup 4096 jul 4 15:28 Blues
drwxr-xr-x 2 nobody nogroup 4096 jul 4 15:28 Clasica
drwxr-xr-x 2 nobody nogroup 4096 jul 4 15:28 Heavy
-rw-r--r-- 1 nobody nogroup 646 jul 4 15:28 index.html
-rw-r--r-- 1 nobody nogroup 132411 jun 22 13:11 jwplayer.js
-rwxr--r-- 1 nobody nogroup 1132 jul 4 15:28 mkhtml.sh
-rw-r--r-- 1 nobody nogroup 106416 jun 22 13:11 player.swf
-rw-r--r-- 1 nobody nogroup 382 jul 4 02:50 template.html
```

Si todo está en orden, llegó el momento de probar el sitio. Vamos al navegador y escribimos en la barra de direcciones [`http://127.0.0.1/miniube/`], y veremos la página índice del sitio. Accediendo a cualquiera de los enlaces, ya estaremos disfrutando de uno de los videos que hayamos subido.

CONCLUSIONES

Por supuesto que hay muchísimo lugar para mejoras y customizaciones. El objetivo de esta práctica, como ya mencionamos, es demostrar el poder de las herramientas que acompañan a cualquier S.O. de la familia *NIX, y la gran productividad que podemos lograr con un dominio básico de las mismas. Esperamos que les haya sido de utilidad, y los alentamos a escribimos planteándonos cualquier duda o sugerencia que tuvieran respecto de lo aquí expuesto. ¡Hasta la próxima!



Por Gabriel Walter Gaitano Ornia
gabriel.gaitanoornia@pixelscode.com

1- ANDROID: ¿QUIÉN ERES Y DE DÓNDE VIENES?

La historia de Android se vincula a los comienzos mismos de la nueva revolución que sacude los cimientos de las industrias informáticas y de telecomunicaciones: la de los dispositivos móviles y crecientemente inteligentes de telefonía. Los más conocidos como “teléfonos celulares” en nuestro mercado tuvieron un vertiginoso crecimiento en capacidades y potencia, dando lugar (con la aparición de las tablets) a que se aproximen a destronar a las PCs en un futuro no muy lejano como el principal dispositivo informático en nuestras vidas.

Muchas empresas lucharon y luchan todavía por participar e idealmente alcanzar una posición de control en este mercado. Una de las que se sumó a esta abierta batalla es Google, cuyo papel dominante en Internet le abrió las puertas a notables recursos y perspectivas. Haciendo gala de su capacidad de previsión y análisis, sus directivos se dieron cuenta de que no existía una opción razonablemente abierta en un mercado dominado por soluciones propietarias y cerradas. Quizás pasó por sus mentes el recuerdo de los comienzos de la revolución de las computadoras personales en los 80, cuando IBM logró una notable victoria que llega a nuestros días utilizando una fórmula sencilla: combinar una gran capacidad económica y tecnológica con una arquitectura abierta que permitiera a muchos fabricantes de menor poderío aportar su propia producción y participar del mercado.

La arquitectura que Google creó tiene esos mismos cimientos y surgió de la base de una pequeña empre-

sa que adquirió en el año 2005, Android Inc ubicada en Palo Alto, California. Sobre lo allí desarrollado se decidió crear un sistema operativo basado en Linux sobre el cual se estructurarían todos los esfuerzos de la compañía y sus aliados, pues fiel a su política de apertura Google fundó una alianza de empresas para respaldar la nueva plataforma Android, la Open Handset Alliance. Algunos de estos socios, sólo para citar a los más conocidos, son Samsung Electronics, Sprint Nextel, Intel, Texas Instruments, Broadcom Corporation, Nvidia, Qualcomm, LG, Marvell Technology Group, T-Mobile y Motorola.

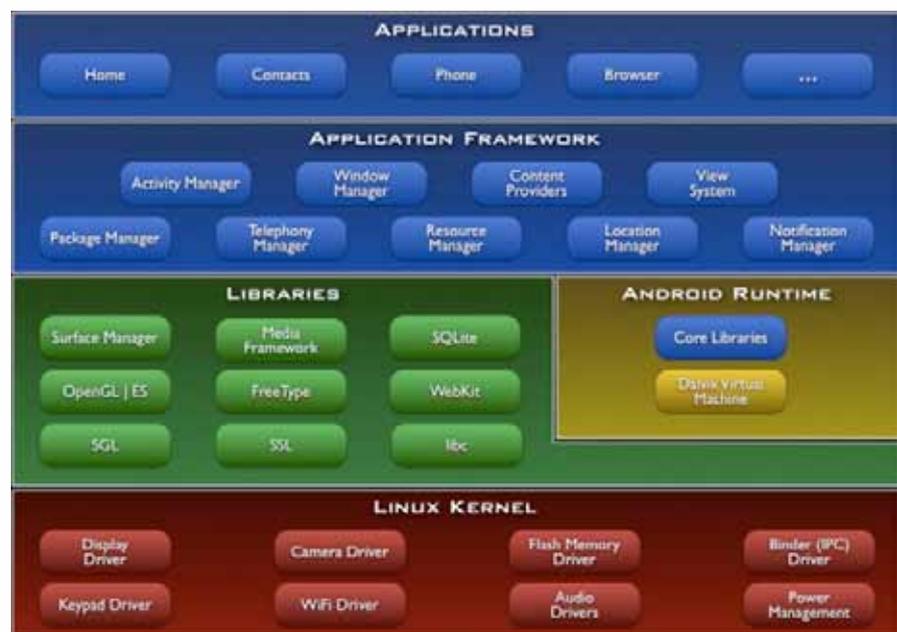
Tenemos así que Android no es simplemente un sistema operativo, sino una arquitectura basada en un core Linux con un set de librerías provistas para permitir a los desarrolladores escribir las propias. Es open source y utiliza una licencia Apache, bajo la cual las compañías

privadas pueden añadir sus propias aplicaciones y extensiones y venderlas sin tener que enviarlas a su vez a la comunidad open source.

En su desarrollo se tuvo en cuenta como valores fundamentales la portabilidad y escalabilidad a futuro. Existe todo un conjunto de herramientas conviviendo en simbiosis sobre el core, y una de las ideas rectoras del diseño es permitir al usuario una gran libertad y control sobre lo que tiene en su dispositivo a diferencia de soluciones cerradas que obligadamente abrevan en un único proveedor y estilo de uso. En la imagen a continuación podemos ver esquemáticamente la estructura interna de Android.

2- UNA FAMILIA NUMEROSA

Las primeras versiones de Android mostraban todavía falencias y fallos, demostrando claramente el hecho de ser una tecnología aún verde. A medida que su consolidación fue afirmán-



dose, se llegó a versiones más estables y con prestaciones más completas. La primera que logró alcanzar una importante popularidad fue la versión 1.6, conocida como Donut. Es bueno comentar que todas las versiones de Android tienen un nombre relacionado con algún dulce o postre, lo cual es probablemente una alusión a lo “delicioso” que resulta el producto final. La versión 1.6 surgió en el año 2009, siendo que es la primera en permitir, entre otras mejoras, la utilización de Google Maps como base de un GPS gratuito incorporado en el dispositivo móvil.

Las siguientes destacadas fueron la 2.0 y 2.1, nombre en clave Eclair (un postre de chocolate). Además de las mejoras de soporte a nivel de hardware y software, apuntadas a una mejora en la experiencia del usuario, esta versión destacó por una mejor integración con los chips de procesamiento de las CPU utilizadas en los dispositivos, permitiendo sustanciales mejoras de performance generales. En general a partir de esta versión se puede pensar en utilizar el celular como una auténtica minicomputadora personal.

Cuando surgen las versiones 2.2 y 2.3, conocidas como Froyo (“FROzen YOgurt”, yogur helado) y Gingerbread (pan de jengibre), Android es ya uno de los principales jugadores en el mercado móvil. Estas versiones amplían sobre todo la performance, producto de una especial revisión y optimización del código realizada con ese objetivo en mente. En Gingerbread se incorpora, además, una nueva versión del kernel de Linux, versión 2.6.35. Son consideradas versiones robustas y potentes, reflejo de la madurez de la plataforma.

Respondiendo a la creciente aparición de nuevos dispositivos con sus propias peculiaridades, Google aceleró el desarrollo de la siguiente versión, 3.0 Honeycomb (panal de miel). Los dispositivos hacia los que está apuntando esta nueva versión son las tablets, lo cual se percibe en el énfasis dado al soporte de pantallas de mayor tamaño y la aceleración gráfica por hardware y procesadores multinúcleo. También en la posibilidad de conexión con múltiples periféricos como hubs, cámaras digitales, mouses, etc. Con esto Google preanuncia un recrudecimiento de la batalla con sus rivales, especialmente con Apple y su tablet estrella iPad, líder e iniciadora de este segmento. Aunque en principio se criticó a Android por esta sobreabundancia de versiones, lo cierto es que fueron acompañando la evolución del hardware y mostrando compromiso para con la plataforma. Hoy por hoy se puede decir que existe una versión para cada necesidad o ambiente hardware móvil en el mercado.

3- LO QUE IMPORTA ES EL SOFTWARE

Uno de los puntos fundamentales del gran atractivo de Android y su meteórico crecimiento en el mercado es su universo de aplicaciones. Siendo muchas de ellas gratuitas o de costo muy reducido, presentan al usuario una gran variedad de posibilidades y permiten extraer el máximo beneficio posible de su tablet o celular.

El punto donde se encuentran principalmente estas aplicaciones es el Android Market, disponible en el URL <https://market.android.com>. Y decimos principalmente porque, a diferencia del modelo de negocio de Apple en

su Apple Store, no existe una fuente única de aplicaciones disponible. ¿Quiénes pueden desarrollar para Android? Todos nosotros nos hallamos en condiciones de hacerlo, cumplimentando unos pocos requisitos, ya que la idea de Google es apuntar a que la mayor cantidad posible de programadores pueda aportar al proyecto y que se logre cubrir la mayor cantidad de aplicaciones posibles en disponibilidad para los usuarios.

Las razones que motivan esta apertura son varias, principalmente que el mismo “espíritu” de Android está centrado en un modelo abierto y participativo. Y desde un punto de vista más prosaico y menos filosófico, pero sin dudas importante, es claro para Google que cuantas más aplicaciones haya disponibles, más atractiva será la plataforma como mencionábamos antes. Algo que sólo podrá lograrse realmente atrayendo a una masa importante de programadores a producir software para los usuarios Android.

Los principales requisitos para desarrollar en Android son conocimientos de la plataforma, y de los lenguajes a utilizar. También, por no dejar de mencionarlo aunque debería asumirse como básico para cualquier actividad, tiempo y dedicación. También un desarrollador podría preguntarse tranquilamente por qué es interesante desarrollar para Android. Aprender un lenguaje o tecnología nueva siempre representa una inversión, con frecuencia importante de tiempo y/o dinero. Naturalmente que todos queremos asegurarnos que la tecnología a aprender valga el esfuerzo realizado en su capacitación.

Sobre esto hay varios puntos a considerar. En primer lugar la perdurabilidad que se le ve a la tecnología que nos gustaría llegar a dominar. Por el caso de Android sus perspectivas son inmejorables: tiene el respaldo de una gran empresa como Google detrás y de, como mencionamos anteriormente, una importante alianza de proveedores de hard y soft. Un segundo punto es la difusión que haya alcanzado la susodicha tecnología. Si bien hay sin duda un lugar para los pioneros (y en ocasiones hacer punta en una arquitectura garantiza nuestra inversión como desarrolladores), también es cierto que se corre un riesgo de embarcarse en un ramal muerto que finalmente nos deje como únicos cultores de un lenguaje abandonado por el grueso del mercado. Viendo las actuales cifras de difusión y crecimiento entre los usuarios y desarrolladores de Android, no cabe la menor duda que no corremos riesgo alguno al respecto. Un elemento a considerar es, también, la complejidad o no de la nueva tecnología, y la existencia de herramientas de desarrollo. Como veremos más adelante, también por este flanco el desarrollo en Android se presenta como una excelente opción a fin de ampliar nuestras habilidades como desarrolladores. Una vez decididos a comenzar a explorar el mundo del desarrollo para Android, nos encontramos con que existen distintas aproximaciones a esta nueva tecnología. Dos son las más oficiales y reconocidas por Google, y la tercera es más independiente aunque se va asimilando a la corriente principal. Pero sea cual sea nuestra elección, necesitaremos

conocer a fondo la estructura de Android desde el punto de vista de un desarrollador.

4- TRAS EL VELO DE ANDROID

Una forma sencilla en que podemos intentar visualizar la estructura de Android es comparándola con una casa con 5 habitaciones. Esta casa representaría a Android como un conjunto, mientras que cada una de las habitaciones ubicadas en su interior representa uno de los 5 elementos clave de la mencionada estructura: las Aplicaciones, el framework de Aplicaciones, las Librerías, el Runtime de Android y el Kernel Linux.

Para seguir la analogía que estamos utilizando deberíamos imaginar que cada una de esas habitaciones contiene un cierto número de gente. Cada una de estas personas representa un elemento de esa habitación. Diferentes habitaciones contienen diversas cantidades de gente.

-APLICACIONES

Esta primera habitación es la más cercana al usuario, diríamos la de más "alto nivel". Su "gente" representa todas las aplicaciones de usuario que tenemos en Android: un cliente de e-mail, un navegador, calendario, mapas, lector de archivos pdf, etc.

-EL FRAMEWORK (MARCO) DE APLICACIONES

Como desarrollador, tendremos acceso completo a las APIs usadas por las aplicaciones del núcleo. Android fue diseñado de modo tal que cualquier aplicación pueda publicar sus capacidades. Y a su vez, cualquier otra aplicación puede usar

esas capacidades. Naturalmente hay algunas restricciones de seguridad, como podría esperarse por otra parte, pero igualmente la posibilidad está y es bastante sorprendente. Esto permite obtener para utilizar un enorme y rico conjunto de "vistas" que harán mucho más fácil codificar. Desde menús drop down, tablas, botones, hasta un navegador completo.

Otras cosas que podemos obtener son un Proveedor de Contenidos (utilizado para permitir a las aplicaciones compartir información), un Gestor de Recursos (que nos ayudará con los gráficos), un Gestor de Notificaciones y un Gestor de Actividades (quien gestiona el ciclo de vida de las aplicaciones). Así podríamos decir que la gente de esta habitación son los gestores y proveedores de nuestra casa virtual.

-LIBRERÍAS

Android tiene disponible un conjunto de librerías de las cuales se ejecutan las aplicaciones. Y como siempre, podemos como desarrolladores acceder a ellas. Estas son las pequeñas personas que pueblan nuestra tercera habitación. Algunas de las librerías de núcleo que podemos mencionar incluyen FreeType, SQLite, LibWebCore y SGL. No ahondaremos en profundidad en las librerías; estas requerirían un artículo entero en sí mismas.

-EL RUNTIME DE ANDROID

Comparada con las otras habitaciones de nuestro ejemplo, esta es bastante exclusiva, sólo contiene en su interior dos integrantes: la Máquina Virtual Dalvik y las librerías de núcleo. La mayor parte de la funcionalidad

disponible en las librerías Java es provista por el Runtime de Android, el cual es otro conjunto de librerías aparte de las anteriores. En Google Android, existe una herramienta llamada 'DX' que ejecuta los archivos en formato '.dex', los cuales son utilizados por la Máquina Virtual Dalvik, la cual es esencialmente el intérprete de las instrucciones Java de las aplicaciones. Este formato es también creado por sus huellas mínimas en la memoria, lo que lo hace especialmente ideal para teléfonos celulares y otros dispositivos móviles, cuyas capacidades de hardware y memoria están lejos de las existentes en equipos más potentes.

Esta máquina virtual fue escrita de modo tal que puede ejecutar múltiples procesos rápidamente y con suavidad. Y descansa fundamentalmente en el Kernel Linux para ejecutar su magia, el cual es nuestro siguiente componente a analizar.

-KERNEL LINUX

En la última habitación de nuestra casa Android imaginaria tenemos al componente base, el que está en la base del edificio. El Kernel Linux sostiene sobre sus anchas espaldas la estructura íntegra de Android.

Utiliza Linux versión 2.6.

Aquí podemos encontrarnos fundamentalmente con los drivers que actúan como una capa intermedia entre los niveles superiores y el hardware del dispositivo. Podríamos mencionar aquí los drivers que dan control al keypad, la conexión de red (WiFi o no), la cámara, y un largo y creciente etcétera.

5- LO CLÁSICO Y RECOMENDADO PARA PROGRAMAR ANDROID: EL SDK

De todas las modalidades de las que básicamente se puede elegir para programar aplicaciones en Android, Google da como principal y sancionada oficialmente la utilización de una mezcla de Java y XML, apoyándose en las librerías que mencionábamos para llevar adelante las tareas del desarrollo. Esto incluye la utilización del Software Development Kit (SDK) Android provisto por Google. Android está hecho pensando totalmente en Java+XML, siguiendo el patrón MVC (Model-View-Controller) que separa el Modelo (los datos) de los programas de administración o lógica de negocio (el Controlador), y a su vez separa ambos de la Vista (la interfaz de usuario).

Su principal inconveniente es que requiere un nivel de conocimiento mayor, especialmente de programación Java. Pero esto mismo es una ventaja para quien ya domina este lenguaje, y además es la modalidad con mayor cantidad de material disponible en la web (foros, tutoriales, etc.) para el aprendizaje.

A continuación mostraremos un ejemplo tomado de la documentación oficial de Google sobre el clásico primer programa "Hola Mundo," con el que solemos tener nuestro primer contacto con la programación de una nueva tecnología. En este caso el lenguaje es Java y la tecnología es Android.

Como programadores sabemos que realizar esta tarea con un nuevo

marco de desarrollo (o framework) suele ser fácil, a diferencia de tener que utilizar las herramientas puras del lenguaje. Con Android es muy fácil como iremos viendo gracias a todos los elementos de los que nos provee el framework. Y se hace más fácil aún si estamos usando Eclipse como nuestro IDE de programación, debido al uso de un plugin que Google provee como herramienta de desarrollo, para manejar la creación de nuestros proyectos y acelerar notablemente nuestros tiempos de desarrollo.

Veremos antes que nada que necesitamos configurar el entorno de desarrollo, para empezar a implementar nuestras aplicaciones Android. Como paso previo, necesitaremos una serie de herramientas instaladas en nuestro sistema, antes de empezar a programar. Como muchas herramientas y ejemplos están en inglés, dejaremos los nombres utilizados en este idioma sin traducir totalmente para facilitar la relación con el material disponible en la web.

ECLIPSE

Este será el IDE que vamos a utilizar. También sería posible programar en Netbeans, pero el plugin de Netbeans para Android no ha alcanzado aún el nivel de madurez que el de Eclipse. ¿Cómo lo instalamos? Para instalar este IDE debemos irnos a la página de descargas del proyecto Eclipse, ubicada en el URL:

<http://www.eclipse.org/downloads/>

y nos bajamos de allí la versión

definida para desarrolladores de lenguaje Java. Eclipse tiene como gran ventaja que no necesita ninguna instalación. Nos bajamos el programa entero, lo descomprimos y tal

descomprimirlo en cualquier carpeta disponible en nuestro disco rígido, recordando su ubicación porque vamos a utilizarlo nuevamente más adelante. En el SDK que Google nos provee

podemos encontrar todas las librerías y utilidades necesarias para poder trabajar con Android plenamente.

PLUGIN ECLIPSE

Para integrar Android con Eclipse, necesitamos un Plugin, que dependiendo de la versión del IDE que tengamos, se instalará de una forma concreta.

Para la versión de Eclipse 3.4 Ganymede

Acceder al Menú "Help->Software updates". Desde aquí, vamos a poder controlar todas las actualizaciones y plugins de Eclipse. Nosotros necesitaremos añadir una nueva dirección para que el sistema localice e instale el Plugin de Android.

Para ello, pulsamos la opción "Add Site...", y en la nueva ventana que se nos abre, tenemos que insertar la siguiente dirección URL: <https://dl-ssl.google.com/android/>

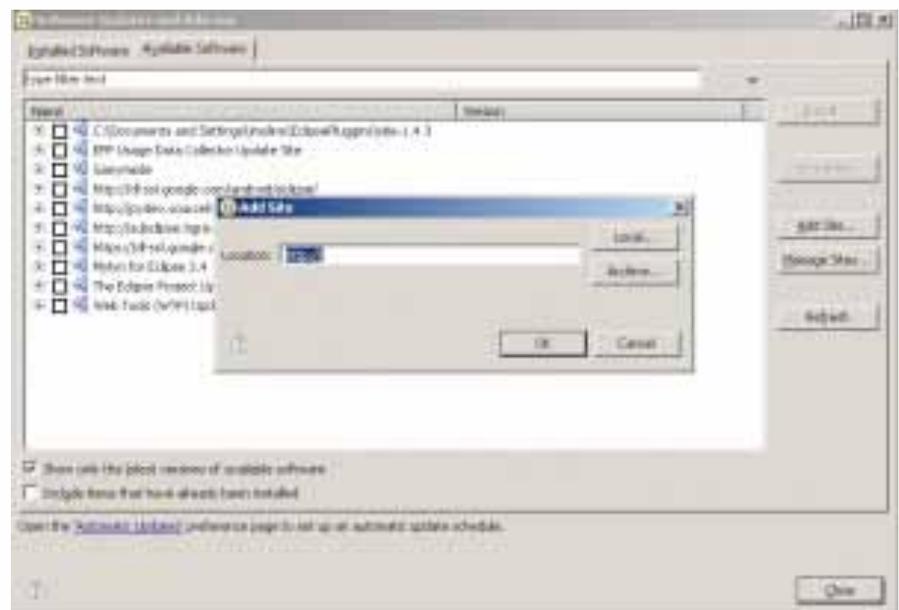
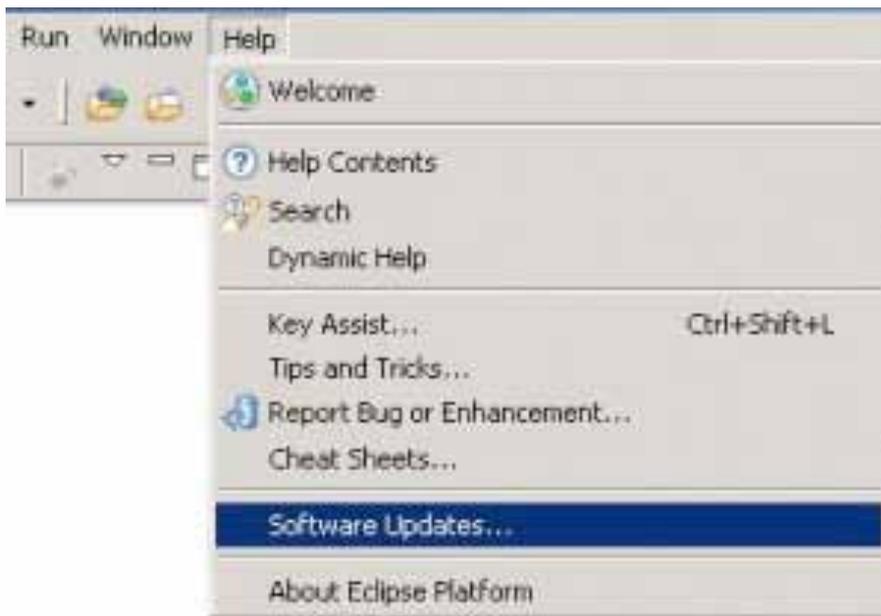
como está descomprimido, lo podemos usar. En esto se comporta similar a los programas denominados portables que carecen de instalador y conservan toda su configuración en las propias carpetas donde corren.

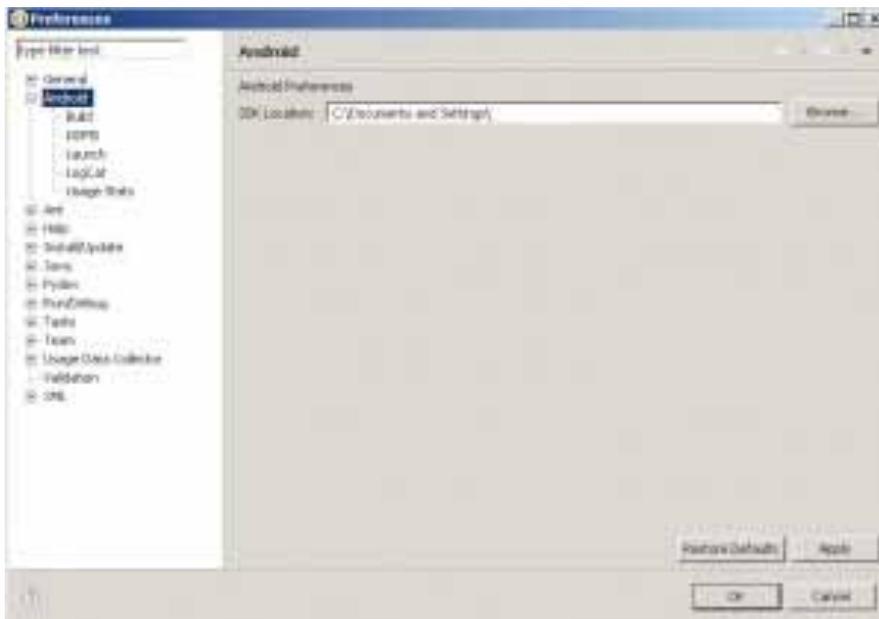
SDK ANDROID

El otro elemento muy importante para desarrollar aplicaciones en Android, es su SDK (Software Development Kit). Para conseguir y usar el SDK, tenemos que acceder a la siguiente página:

<http://code.google.com/intl/es-ES/android/download.html>

Una vez descargado, sólo hay que





eclipse/

Atención con esto, porque si no reconoce la dirección, vamos a tener que probar con “http” en vez de “https”. Una vez añadido el sitio, marcamos e instalamos el nuevo plugin que nos aparece en la lista desplegable. En caso que nos pida instalar otro plugin, vamos a buscarlo mediante el buscador de plugins y lo instalaremos de igual forma que empezamos a instalar el de Android. Una vez instalado todo, procedemos a reiniciar Eclipse para poder continuar.

EL TOQUE FINAL

Finalmente llegó el momento de juntar todo lo que estuvimos buscando e instalando. Para eso, tenemos que decirle a Eclipse, dónde se encuentra el SDK de Android. Para ello nos dirigimos al menú “Windows->Preferences”. Allí nos aparecerá una ventana como la siguiente

Marcamos la etiqueta de “Android” y en “SDK Location” vamos a buscar y añadir la dirección donde habíamos guardado el SDK original-

mente.

Una vez realizado todo esto, estamos en condiciones de poder empezar a crear nuestro propio proyecto para Android. Para lo cual lo primero será instalar una plataforma.

-INSTALAR UNA PLATAFORMA

Para poder correr la aplicación “Hola Mundo”, vamos a necesitar instalar al menos una plataforma Android en nuestro recién bajado SDK. Si no lo hicimos hasta ahora este es el momento. Y para instalar una plataforma en Eclipse debemos seguir los siguientes pasos:

1. En el Android SDK and AVD Manager, elegimos Available Packages en el panel izquierdo.
2. Clickeemos el checkbox del repositorio para mostrar los componentes disponibles para realizar una instalación nueva.
3. Tenemos que seleccionar al menos una plataforma para instalar y clickear Install Selected. Si no estamos seguros cuál es la plataforma adecuada para instalar, lo mejor

sería utilizar la más nueva.

-CREAR UN AVD (DISPOSITIVO VIRTUAL ANDROID)

Para aprender más sobre cómo utilizar AVDs, deberíamos buscar en la documentación de Google, aquí daremos un ejemplo somero de su utilización y posibilidades. Utilizaremos en este ejemplo para ejecutar nuestra aplicación el Android Emulator. Y antes que podamos lanzar el emulador, tendremos que crear un Android Virtual Device (AVD). Un AVD define la imagen de sistema y los seteos de dispositivo usados por el emulador.

PARA CREAR UN AVD:

1. En Eclipse, elegimos Window > Android SDK and AVD Manager.
2. Seleccionamos Virtual Devices en el panel ubicado del lado izquierdo.
3. Clickeamos la opción New.
4. El diálogo Create New AVD aparece.
5. Tipeamos el nombre del AVD, como ser “mi_avd”.
6. Elegimos un destino. El destino es la plataforma (o sea, la versión del SDK Android como 2.1), la cual queremos correr nosotros en el emulador
7. El resto de los campos pueden ser ignorados en este ejemplo en forma provisoria
8. Clickeamos la opción Create AVD

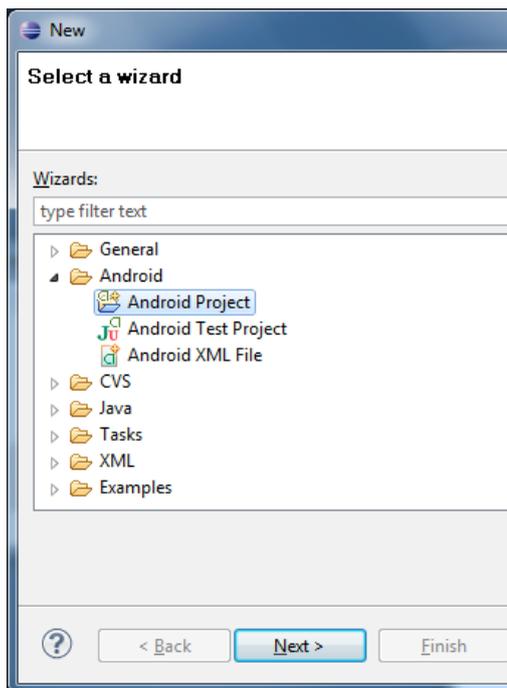
-CREAMOS UN NUEVO PROYECTO ANDROID

Después de haber creado el AVD, el próximo paso es comenzar un nuevo proyecto Android en Eclipse. Desde Eclipse entonces, seleccionamos File > New > Project.

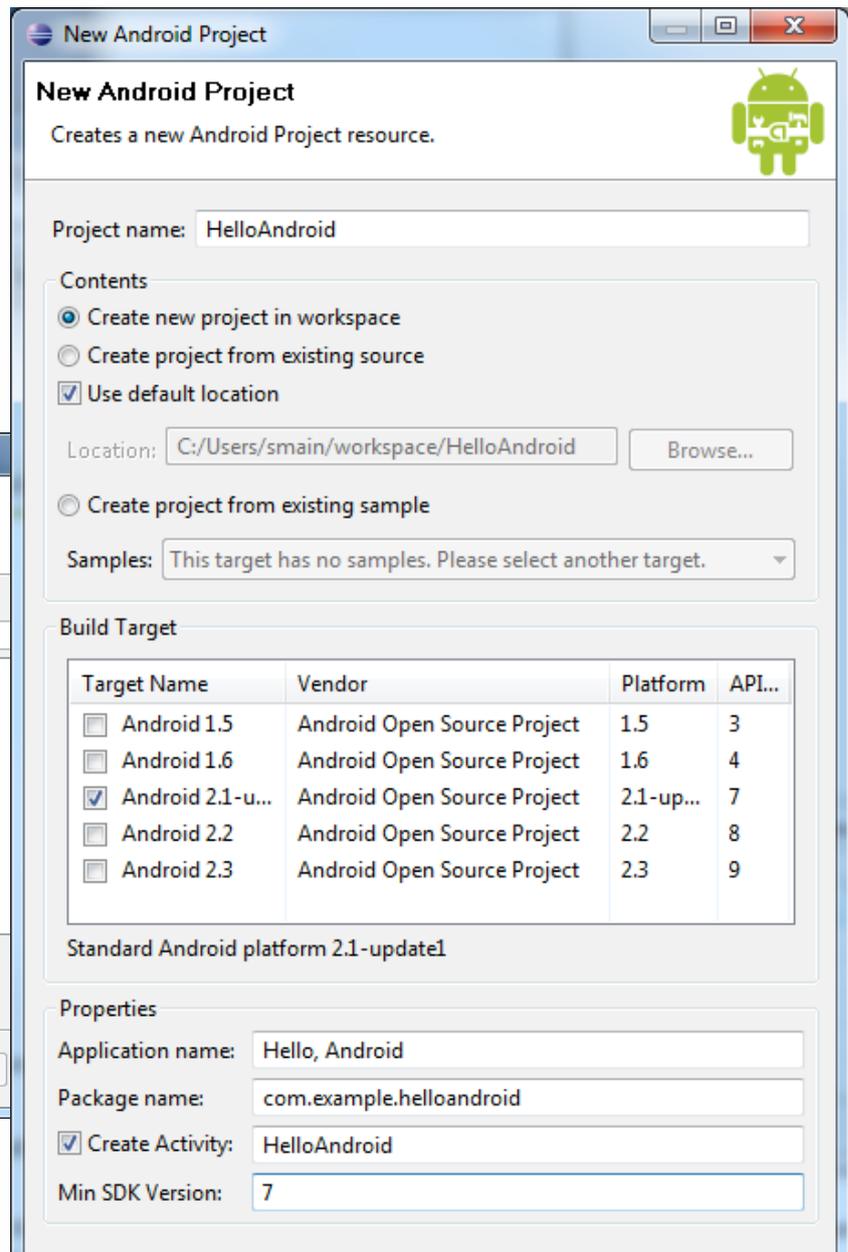
1. Si el plugin ADT para Eclipse

ha sido instalado exitosamente, el diálogo resultante debería tener una carpeta etiquetada como "Android", la cual debería contener una etiqueta "Android Project". Más tarde, cuando hayamos creado uno o más Proyectos Android aparecerá también una entrada para "Android XML File."

2. Seleccionamos "Android Project" y clickeamos la opción Next.



- 3.
4. Completamos los detalles del proyecto con los siguientes valores:
 Project name: HelloAndroid
 Application name: Hello, Android
 Package name: com.example.helloandroid (o nuestro espacio de nombres privado)
 Create Activity: HelloAndroid



5. Clickeamos la opción Finish.
- 6.
7. Acá tenemos una descripción de cada campo:
 - Project Name: Este es el nombre de proyecto para Eclipse — el nombre del directorio que contendrá los

- archivos del proyecto.
- Application Name: Este es el título legible por humanos para nuestra aplicación — es el nombre que aparecerá en el dispositivo Android.
 - Package Name: Este es el espacio de nombres (sigue la mismas reglas

utilizadas para los paquetes estándar en el lenguaje de programación Java), bajo el cual queremos que todo nuestro código fuente resida. También setea el nombre de paquete bajo el cual se generará el item Activity.

El nombre del paquete debe ser único en todos los paquetes instalados en el sistema Android, por esta razón, es importante utilizar un paquete con nombre estándar estilo dominio para nuestras aplicaciones. El ejemplo anterior utiliza el espacio de nombres "com.example", que es un espacio reservado para la documentación de ejemplo - a la hora de desarrollar nuestras propias aplicaciones, debe utilizar un espacio de nombres que sea adecuado al uso que le queramos dar

- Create Activity: Este es el nombre del item de clase que será generado por el plugin. Será una subclase de la clase Activity. Una clase Activity es simplemente una clase que puede ser ejecutada y realizar alguna tarea. Puede crear una Interfaz de Usuario si se decide hacerlo de ese modo, pero no es imprescindible. Como el checkbox sugiere, esto es opcional, pero una clase Activity es casi siempre usada como la base para una nueva aplicación Android.

- Min SDK Version: Este valor especifica el nivel mínimo de la API requerido. El Nivel API es un valor entero que identifica únicamente la revisión del framework que es ofrecida por la versión de la plataforma Android. Es un valor utilizado por las aplicaciones para esencialmente definir su ambiente y marcar en qué rango de niveles (o versiones) puede ser ejecutado. Al

hablarse de un valor mínimo lo que se está indicando es que se recomendará que la aplicación no sea ejecutada por Android por debajo de ese valor aquí indicado

- Otros campos: El checkbox para "Use default location" ("Usar ubicación por defecto") nos permitirá cambiar la ubicación en el disco donde los archivos del proyecto serán generados y almacenados. "Build Target" es la plataforma elegida como objetivo ("target") para compilar nuestra aplicación (esto debería ser seleccionado automáticamente, basado en el parámetro Min SDK Version que definimos antes). Notemos que el "Build Target" que nosotros elegimos usa la plataforma 1.1 Android. Esto significa que nuestra aplicación será compilada con la librería de plataforma Android versión 1.1. Si recordamos lo que realizamos antes, veremos que la AVD creada anteriormente se compilará utilizando la plataforma Android 1.5. ¿Tenemos un problema entonces? La respuesta es que no, debido a que las aplicaciones Android están diseñadas para ser compatibles hacia atrás. Estas versiones no tienen entonces por qué concordar, de modo que una aplicación compilada con la versión 1.1 correrá normalmente y sin mayores inconvenientes en una plataforma ejecutando la versión 1.5, por poner un ejemplo. Sin embargo no es posible ejecutar la opción inversa, o sea que si compilamos con (siguiendo nuestro ejemplo anterior) Android 1.5, no podremos ejecutarlo en un ambiente 1.1.

Ahora nuestro proyecto Android está listo. Debería ser visible en el Package Explorer sobre la izquierda.

Así abrimos entonces el archivo fuente java HelloAndroid.java, ubicado dentro del path HelloAndroid > src > com.example.helloandroid. Debería contener el siguiente código:

```
package com.example.helloandroid;

import android.app.Activity;
import android.os.Bundle;

public class HelloAndroid extends Activity {
    /** Llamado cuando es creada por primera vez la clase activity */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }
}
```

Como vemos, la clase extiende (hereda) de la clase Activity. Una Activity es una entidad de aplicación única que se utiliza para realizar acciones. Una aplicación podría tener muchas actividades separadas, pero el usuario sólo interactuará con una a la vez. Y el método onCreate() será llamado por el sistema Android en el mismo momento que la actividad sea iniciada — momento en el cual deberemos realizar toda la inicialización y configuración de la interfaz de usuario. No es necesario en lo más mínimo que una actividad tenga una interfaz de usuario, pero lo más usual es que la posea.

**CONSTRUYENDO
LA INTERFAZ DE USUARIO**

Miremos el código revisado que sigue a continuación y realicemos los mismo cambios a nuestra clase ejemplo HelloAndroid. Los elementos en negrita son las líneas que han sido añadidas en la expansión del ejemplo.

```
package com.example.helloandroid;

import android.app.Activity;
import android.os.Bundle;
import android.widget.TextView;

public class HelloAndroid extends
Activity {
    ** Llamado cuando es creada por
primera vez la clase activity */
    @Override
    public void onCreate(Bundle save-
dInstanceState) {
        super.
onCreate(savedInstanceState);
        TextView tv = new TextView(this);
tv.setText("Hola, Android");
setContentView(tv);
    }
}
```

Truco a recordar: Una forma muy fácil de añadir paquetes de importación a nuestro proyecto es apretar conjuntamente las teclas Ctrl-Shift-O (Cmd-Shift-O, en una computadora Mac). Este es un pequeño atajo de teclado de Eclipse que identifica paquetes faltantes basándose en lo que hayamos puesto en nuestro código y que los completa por nosotros. Una interfase de usuario Android está formada por jerarquías de objetos denominados Vistas (Views). Una Vista es un objeto arrastrable usado como elemento en nuestro diagrama de interfaz de usuario, como por

ejemplo un botón, imagen o (en este caso que estamos utilizando como ejemplo) una etiqueta de texto. Cada uno de estos objetos es una subclase de la clase Vista y la que se encarga de manejar texto es TextView.

En este cambio que realizamos, creamos nosotros un elemento textView con el constructor de la clase, el cual acepta una instancia de la clase Android Context como su parámetro. Un Context (Contexto en castellano) es un manejador (handle) del sistema; provee servicios como resolución de recursos, obteniendo acceso a bases de datos y preferencias, etc. Nuestra clase Activity hereda de Context, y debido a que nuestra clase HelloAndroid es en realidad una subclase de la clase Activity, lo es obviamente también de la clase Context. Así vemos de este modo que podemos pasarla como referencia de Context al textView.

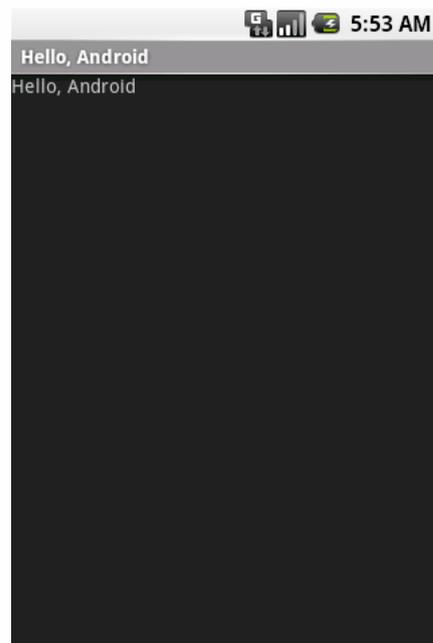
Luego definimos el contenido del texto con setText(). Y, finalmente, pasamos el TextView a setContentView() para mostrarlo como el contenido de la interfaz de usuario de la Activity. Si no lo hacemos así, o sea si la Activity no llama a este método, no habrá interfaz de usuario alguna y el sistema mostrará únicamente una pantalla en negro, lo cual no sería muy agradable de ver. Con esto tenemos lista una aplicación "hola mundo" clásica en Android utilizando Java. Veamos ahora cómo ejecutarla

CORRER LA APLICACIÓN CREADA

El plugin Eclipse hace muy fácil poder correr una aplicación que

hayamos realizado:

1. Seleccionamos la opción Run > Run.
 2. Elegimos "Android Application".
- Así veremos cómo el plugin para Eclipse crea una nueva configuración de ejecución en forma automática (es posible configurar nuestras propias configuraciones a mano si lo deseamos), y luego lanza el Emulador Android que vimos e instalamos anteriormente. Dependiendo del ambiente de trabajo que tengamos, el Emulador podría demorar varios minutos en arrancar por completo, lo cual significa que deberemos ser pacientes. Finalmente, cuando el Emulador haya arrancado, el plugin Eclipse instala nuestra aplicación y procede a instalar la



Activity por default. Deberíamos ver algo como lo que muestra la imagen a continuación:

El "Hola, Android" que veremos (la versión de la imagen esta en inglés)

en la barra grisada es realmente el título de la aplicación. Es creada en forma completamente automática por el plugin Eclipse (la cadena de texto es definida en el archivo `xml/res/values/strings.xml` y es referenciada por el archivo `AndroidManifest.xml`). En cambio, el texto visible por debajo del título es el texto que hemos creado en el objeto `TextView`

Ahora veamos entonces un poco la estructura de carpetas y archivos que suele tener una aplicación Android SDK:

- `/src` – Aquí se encuentran las actividades (controladores en el patrón MVC), clases y un archivo que se llama `R.java`. Este archivo no se debe manipular nunca, ya que en forma transparente al desarrollador Android lo actualiza automáticamente. Si lo miramos internamente nos encontraremos con variables y otros elementos.
- `/Android Library` – Eclipse referencia al `sdk` de Android. No deberíamos mover nunca el `sdk` de lugar, pero si lo hacemos será fundamental recordar la nueva ubicación.
- `/assets` – Aquí se puede incluir archivos de tipos varios (como ser música, rars, pdfs, zips, etc.).
- `/res/drawable` – Todas las imágenes y gráficos van almacenadas en este directorio.
- `/res/layout` – Todas las vistas que tengamos desarrolladas van en esta ubicación.
- `/res/values` – En el archivo `strings.xml` podemos poner variables que luego sustituiremos en las vistas, utilizando algo similar a un `template`.
- `AndroidManifest.xml` – Este archi-

vo es muy importante porque recoge todas las actividades y algunas configuraciones del proyecto.

RECURSOS PARA DESARROLLAR

Para poder acceder a posibilidades mayores, una serie de recursos interesantes para desarrolladores que podemos mencionar son los siguientes:

developer.android.com : Este es el sitio oficial de desarrolladores Android, con un nivel realmente bueno. A pesar del poco tiempo que tiene la tecnología Android en la calle, ya existe gran cantidad de developers que lo han elegido e intercambian información y experiencias. Además hay buena cantidad de material, como ser ejemplos y documentación disponibles para su uso.

www.androiddevelopment.org: Rebeldes entre los rebeldes, este grupo no oficial de programadores Android tiene gran cantidad de material sobre la plataforma para aquellos que ingresen a su sitio.

sgoliver.net (castellano): Un blog en el que se recopilan decenas de artículos para empezar desde lo más sencillo hasta lo más complicado en lo que a creación de apps Android se refiere. Existen aquí tutoriales para bases de datos en Android, XML y sobre todo de interfaz de usuario.

<http://and.roid.es/> (castellano): Es un pequeño blog español en donde se recogen las últimas noticias relacionadas con la plataforma Android, además de varios tutoriales para

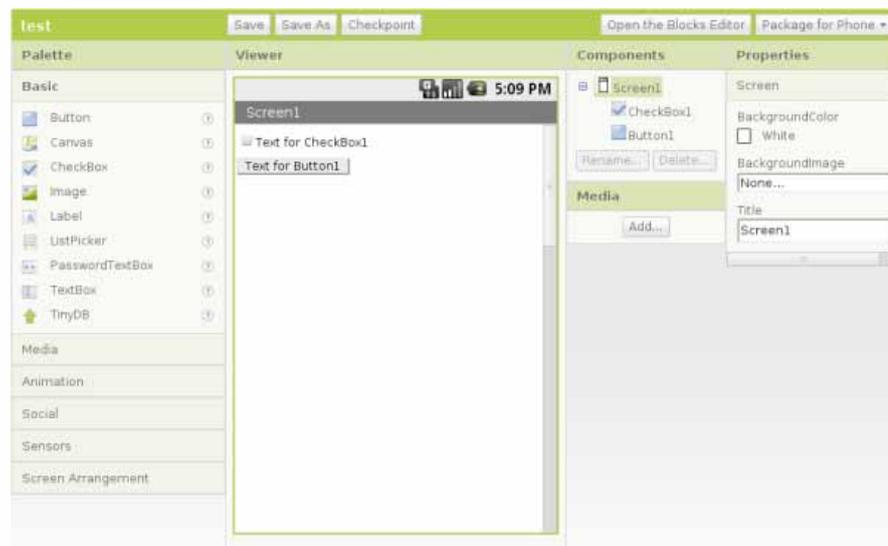
poder crear apps, entre otros contenidos que pueden considerarse bastante interesantes.

<http://www.maestrosdelweb.com> (castellano): Legendaria página y foro orientada al diseño y programación web, ha alcanzado ahora un nuevo nivel al agregar información referida a la programación de celulares y dispositivos móviles. Y, por supuesto, tiene abundante material sobre Android.

Esta es la forma de trabajar a través del SDK, aunque en su expresión más simple. Como hemos visto, es necesario poseer conocimientos de Java y de programación para poder utilizarla. Este modo es sin duda poderoso y versátil, pero no brinda las posibilidades que las herramientas RAD (Rapid Application Development, o sea Desarrollo Rápido de Aplicaciones) proporcionan al ambiente PC. Es por eso que Google decidió proveer una alternativa de más alto nivel al desarrollador, la cual consideraremos a continuación.

6- CUANDO PROGRAMAR ES UN JUEGO

Los ingenieros de Google se tomaron seriamente la necesidad de abrir la posibilidad de utilizar y por ende desarrollar en Android lo más posible. Debido a esto crearon un servicio pensado para que cualquier persona con o sin experiencia de programación pueda desarrollar para Android, con una interfaz de desarrollo que tiene fuertes reminiscencias de juego infantil. ¿Acaso un niño podría diseñar una aplica-



ción coherente para la plataforma Android? Bueno, de hecho niños fueron algunos de los betatesters en las pruebas que los ingenieros de Google realizaron para esta nueva aplicación: App Inventor, la cual es fundamentalmente un editor simplísimo que permite crear aplicaciones para Android en forma visual. Se trata de armar el diseño de cómo queremos que se vea nuestra app, y luego indicar el comportamiento de cada bloque lógico. Estos bloques pueden hacer “casi cualquier cosa que se puede hacer con un teléfono Android,” según explican los ingenieros de Google Labs.

Para poder acceder al App Inventor tenemos que anotarnos para que nos envíen una invitación. Este producto funciona en varios sistemas operativos como ser Windows, OS X y Linux. El sitio ya cuenta con varios tutoriales completos de aplicaciones simples para empezar a usar el software. Hay juegos, aplicaciones que usan infor-

mación del teléfono como nuestros contactos, una para dibujar, otras para hacer preguntas y respuestas y una para enviar mensajes de texto en forma totalmente automática. Esta aplicación permite tomar un bloque lógico visto en formato visual y arrastrarlo con un sencillo mecanismo de drag and drop al que nos tienen acostumbrados los IDE gráficos, interconectándolos luego como piezas de un rompecabezas.

¿Cuál es la motivación de este proyecto? Fundamentalmente parte de, en el espíritu abierto de Android, hacer que los usuarios puedan empezar a ser parte de sus dispositivos, como en algún momento pudieron apropiarse de Internet al crear sus propias páginas gracias a servicios como el recordado Geocities. Otra motivación que encontramos en este equipo es despertar la innovación a través del aprendizaje de crear aplicaciones simples, seguramente se tratará de la puerta

de entrada a muchos que luego terminen aprendiendo más para hacer mejores aplicaciones.

7- EL RETORNO DEL REY

Para concluir nuestro análisis de la programación Android no podemos dejar de mencionar una reciente alternativa. Hasta ahora teníamos una opción de nivel medio para programar con Java y XML al estilo oficial del SDK, sumamos como vimos la opción de programación de alto nivel que nos da el AppInventor. Pero ahora se agregó oficialmente la posibilidad de programar en el más tradicional y poderoso de los lenguajes: C y su descendiente C++. Si bien el Android Native Development Kit (NDK) permitirá a los desarrolladores construir componentes Android con estos lenguajes, no deja de venir con algunas limitaciones, orientándose más bien a la posibilidad de desarrollar partes que requieran una alta performance mientras que se mantiene el énfasis en la programación Java.

Como vemos esta plataforma Android se muestra rica en posibilidades para los interesados en sumarse al creciente número de desarrolladores que la han elegido, y se halla muy lejos de haber alcanzado su cenit. El tiempo nos mostrará sin duda sus alcances finales, pero queda claro que llegará muy lejos.

Programación mobile para Windows Phone 7



Por Fernando Luna
fernando.luna@pixelscode.com

En la última década hemos presenciado grandes cambios en la tecnología móvil. Los teléfonos celulares vivieron una gran transformación: de ser simples aparatos con línea telefónica a dispositivos inteligentes con una funcionalidad cada vez más extendida. Microsoft siguió esta carrera desde un puesto no tan privilegiado, aunque en los últimos años, una vez consolidado este mercado, la empresa oriunda de Redmond decidió dar un vuelco total y comenzar a apostar de lleno al desarrollo móvil.

Los teléfonos celulares juegan un papel importante en la vida de todos nosotros. Desde poder estar comunicados con la familia o con nuestro trabajo, ya sea mediante un llamado telefónico, o por mensajes de texto, hasta poder recibir los cambios de último momento en nuestros quehaceres diarios por e-mail o mediante el acceso a aplicaciones especialmente diseñadas para la tecnología móvil. Estamos viviendo la era de los tablets y telefonía móvil inteligente, y Microsoft decidió dar pelea por un lugar en este mercado con la llegada de Windows Phone 7. Este sistema operativo para teléfonos celulares inteligentes es la evolución al viejo Windows Mobile, tecnología adaptada por Redmond para correr en smartphones. Junto con la llegada de Windows Phone 7, a partir de ahora WP7, Microsoft preparó su terreno para poder disputar la administración íntegra de los smartphones desde este sistema operativo, sin descuidar las funciones básicas del mismo, el entretenimiento y las aplicaciones accesorias que nos brindan la información deseada en el momento preciso. Junto con WP7 y Visual Studio 2010, Microsoft preparó un SDK que permita a todo programador desarrollar aplicaciones móviles, ya sea web o propiamente instaladas en el teléfono. El SDK en un principio fue lanzado al mercado sólo para soportar programación bajo el lenguaje Visual C#, y debido al reclamo incesante de muchos programadores que siguen a Visual Basic desde sus inicios en la era del D.O.S., en marzo de 2011 fue lanzado el SDK para este lenguaje también, aunque por el momento limitado a las versiones Ultimate o Superior de Visual Studio 2010.



REQUISITOS BÁSICOS

Si ya tenemos instalado Visual Studio 2010 en cualquiera de sus versiones, podemos descargar el SDK WP7 para programar aplicaciones mobile desde la siguiente dirección: <http://create.msdn.com/en-us/resources/downloads>
 El SDK requiere que tengamos instalado Visual Studio 2010 y Service Pack 1, ya sea en su versión Express o superior. La página inicial para la descarga de las herramientas Express de Microsoft, sin ningún costo monetario es la siguiente: <http://www.microsoft.com/express>
 Los requisitos de instalación básicos comunes para todas estas herramientas son: Windows Vista SP2 o superior, 4 GB de espacio en disco, de 2 a 4 GB de memoria RAM (2 GB para la versión Express y 4 GB para la Professional o superior), DirectX 10 o superior. En esta nota haremos uso de la versión Express de dichas herramientas, las cuales permiten el desarrollo de aplicaciones móviles gratuitas, ideal para “Indie Developers”.

El punto de partida principal para la instalación de las herramientas básicas de desarrollo Visual Studio 2010

Para descargar el SDK de WP 7.1 (Codename Mango), aún en beta al momento de escribir esta nota, deben acceder al siguiente link: <http://create.msdn.com/en-US/>
 Una vez descargado e instalado todo, encontraremos en el menú de programas de Windows el acceso a Visual Studio 2010 Express for Windows Phone 7. Al iniciar el mismo nos encontraremos con la interfaz propia de Visual Studio 2010, y la posibilidad de iniciar el desarrollo de distintos proyectos. Este paquete no sólo se limita a desarrollar aplicaciones para WP7, sino también juegos de video para WP7 y para la plataforma XNA, la cual agrupa juegos de computadora y juegos para Xbox 360, entre otros proyectos no menos interesantes.

El lenguaje por defecto para desarrollar aplicaciones WP7 es Visual C#. Quienes conozcan alguna variante del lenguaje C, Java o PHP, no encontrarán dificultad alguna en crear un proyecto fácil.

mostrará todos los archivos vinculados a nuestro proyecto.

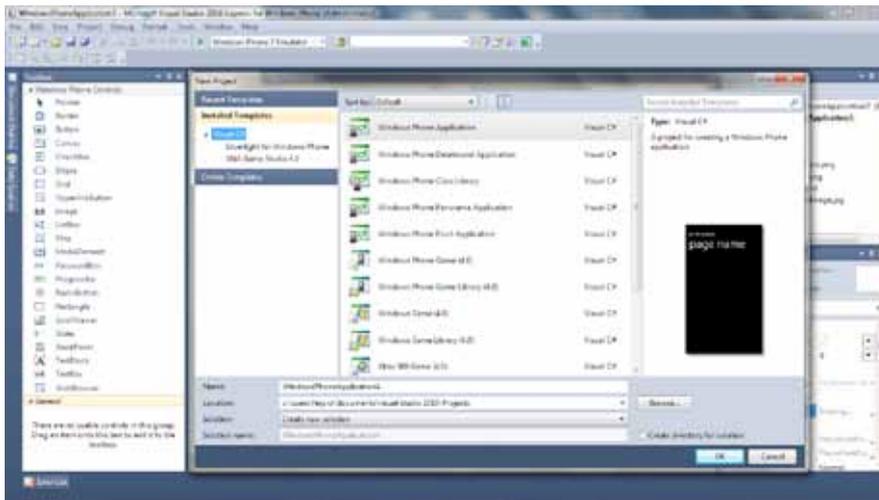
EL ALMA Y VIDA DE CUALQUIER PHONE APP

Las aplicaciones móviles desarrolla-

propiedades tal cual lo hacemos al desarrollar una aplicación Visual Studio para Windows o Web, o mediante XAML, el cual nos permite escribir mediante código la creación de cada uno de sus controles, con sus propiedades y demás características, que luego darán vida a nuestra aplicación.

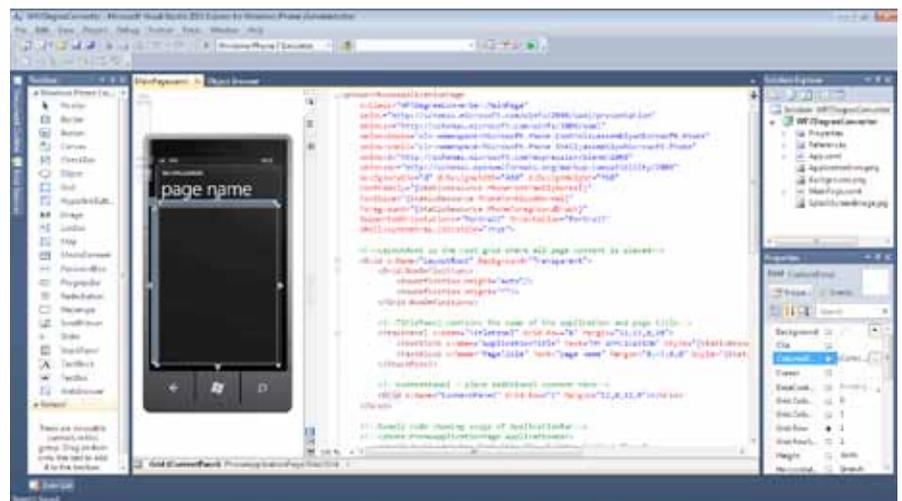
Toda aplicación WP7 está basada en un solo archivo distribuible para descargar e instalar en los teléfonos que corren este sistema operativo. Su extensión es XAP, que no es más que un archivo ZIP, el cual podrán abrir con cualquier compresor/ descompresor de archivos tan sólo renombrando su extensión.

El motor propio de Runtime de cada aplicación WP7 en el teléfono no es nada más ni nada menos que Inter-



El IDE de desarrollo de Visual Studio 2010 para Windows Phone 7 es similar al IDE propio de Visual Studio. Quienes hayan incurrido por alguna versión de éste último, se sentirán como en casa.

El IDE se divide en una barra de menú y barra de herramientas en la parte superior de la ventana. Sobre el lateral izquierdo encontramos el Panel de herramientas o Toolbox, el cual contiene los controles que darán vida visual a nuestras aplicaciones. En el panel central tenemos un simulador de un smartphone, lo que sería el equivalente a un WinForm. Sobre el lateral derecho encontraremos el Panel de propiedades, para ajustar cada propiedad de los controles que utilizaremos, y sobre el lateral derecho superior encontrarán el Explorador de Soluciones, el cual



das en WP7 tienen una particularidad respecto a las que son desarrolladas para la plataforma Android o iPhone. Las mismas contienen una estructura XAML, una variante de aplicaciones basadas en el lenguaje XML. Podemos desarrollar cada aplicación visualmente, arrastrando los controles y configurando sus

Al iniciar una nueva aplicación para WP7 nos encontraremos con un emulador de teléfono celular, en el cual podremos calcular con suma precisión lo que ocuparán los controles que comprenderán nuestra aplicación

net Explorer. El famoso navegador web de Microsoft se ejecuta en una instancia oculta para aprovechar su motor, el cual interpreta sin problemas la extensión XAP, descomprimiendo para utilizar los archivos que componen nuestra Mobile App. Dado el reciente lanzamiento mundial de WP7, aún es poco común encontrarse con teléfonos que

Application. La llamaremos WindowsPhoneLC. Esta aplicación simple nos permitirá transformar un número en millas o kilómetros, según lo que elijamos. Los elementos que incluiremos en el proyecto son básicos. 2 RadioButtons, 1 TextBox, 2 TextBlocks (equivalente al control Label), 1 Button y 1 control Image (este último

realizado este paso, setearemos el nombre de nuestra aplicación y su título. Sobre el TextBlock llamado PageTitle agregaremos a su propiedad Text = Conversor. Sobre el TextBlock llamado ApplicationTitle agregaremos a su propiedad Text = Longitude Converter. Agregamos sobre la pantalla un textBlock en el cual agregamos a su propiedad Text = Unidad. A su lado agregamos un textBox sin texto, que será en donde ingresaremos el valor a convertir. Debajo de estos controles agregamos un RadioButton1 cuya propiedad Content será Convertir a KM, y otro RadioButton2 cuya propiedad Content será Convertir a Millas. Debajo de estos agregamos un control Button, Button1, en el cual ingresamos la leyenda Convertir Ahora en su propiedad Text. Por último agregamos otro textBlock, TextBlock2, donde figurará su propiedad Text = resultado. Diagramemos nuestra aplicación para que quede tal cual la distribución de la siguiente imagen:



Al ejecutar nuestra aplicación en modo Debug, el emulador de WP7 cargará nuestra Mobile App, a través del motor de Internet Explorer, tal como vemos en esta imagen.

corran este sistema operativo. Por eso el IDE de desarrollo incluye un emulador del teléfono que permitirá que probemos nuestras aplicaciones en la computadora, y gracias al cual podremos apreciar el look&feel de la mismas aparte de su correcta operatividad.

CREAR UNA APLICACIÓN WP7
Iniciemos nuestra primera aplicación para WP7 desde el menú File > New Project > Windows Phone

mo opcional). El control Image es para agregar una imagen de fondo, la cual por supuesto puede ser obviada si no queremos incorporarla. Las resoluciones más comunes que podemos encontrar en Windows Phone 7 son: 480x800 y 320x480 pixeles (ancho por alto). Sobre estas bases debemos construir nuestras aplicaciones. Tenemos dentro de los namespaces propios del lenguaje comandos que nos permiten conocer la resolución del teléfono, y en base a esto ajustar la distribución de nuestros controles. Creemos a continuación la interfaz de nuestra aplicación. Para esto, agreguemos un control Image, y en su propiedad Source, naveguemos hasta poder seleccionar la imagen de fondo que deseamos visualizar. Una vez

Veremos al agregar cada control que simplemente arrastrando los mismos obtendremos las dimensiones de éstos y guías en pantalla que permitirán alinear y centrar los mismos respecto a los otros controles.

A CODIFICAR NUESTRA APP

Con sólo unas pocas líneas de código nuestra Mobile App cobrará vida. Por supuesto que la misma requiere de un análisis profundo mucho mayor, como ser validaciones de datos ingresados, control de errores y otras funcionalidades más que se le aplica a cualquier desarrollo, pero aún puliendo lo más posible nuestro desarrollo, no nos demandará mucho tiempo ni esfuerzo. Una vez armada la interfaz, sobre



A través de esta imagen podremos guiarnos para distribuir uniformemente los controles que darán vida a nuestra aplicación. Los controles serán agregados simplemente con arrastrar y soltar los mismos sobre el IDE de desarrollo.

el control Button1 debemos hacer doble clic. De esta manera se abrirá el editor de código, en este caso correspondiendo al evento Click() de dicho control. Allí debemos ingresar el siguiente texto:ç

```
private void button1_Click(object sender, RoutedEventArgs e)
{
    double ingresado;
    double resultado;
    double factoraKM;
    double factoraMillas;
    factoraKM = 1.6093;
    factoraMillas = 0.6214;
    ingresado = System.Convert.
    ToDouble(textBox1.Text);
    ...
}
```

En las primeras líneas estamos

declarando cuatro variables, que serán utilizadas para ejecutar una operación y así obtener el resultado que queremos mostrar. Dado que aparecerán decimales en nuestro sistema de conversión, utilizamos el tipo de dato Double, que nos dará mayor precisión en la respuesta. El siguiente código se encapsula en dos simples sentencias If, que realizarán el resto del trabajo antes de mostrar el resultado en pantalla:

```
...
if (radioButton1.IsChecked ==
true)
{
    ...
}
```



Si todo va bien y nuestra aplicación compila sin ningún inconveniente, se ejecutará en nuestra PC el emulador de WP7 para mostrarnos en acción nuestra primer MobileApp.

```
ingresado = System.Convert.
ToDouble(textBox1.Text);
resultado = ingresado *
factoraKM;
textBlock2.Text = Convert.
ToString(resultado);
}
```

```
if (radioButton2.IsChecked ==
true)
{
    ingresado = System.Convert.
ToDouble(textBox1.Text);
resultado = ingresado *
factoraMillas;
textBlock2.Text = Convert.
ToString(resultado);
}
resultado = 0;
ingresado = 0;
}
```

Las instrucciones del código son simples: Si RadioButton1 está chequeado, entonces suponemos que el valor ingresado es en millas y deberá ser convertido a Kilómetros. Por el contrario, si RadioButton2 fue marcado entonces la aplicación interpreta que se ingresó un valor en Kilómetros y deseamos convertirlo a millas.

Listo, ahora sólo nos queda ejecutar el proyecto en el Emulador.

El puntero del mouse simula ser nuestro dedo sobre la pantalla táctil. Hagamos un clic en el textBox1 y luego ingresemos un valor numérico. A continuación seleccionamos el destino de la unidad de conversión, y presionamos el botón. Veremos el resultado en el textBlock2.

Dentro de las propiedades de cada aplicación, también podemos especificar el copyright correspondiente de la misma y otra información importante, como ser el nombre y descripción. Las imágenes que se inclu-



En el primer teléfono vemos cómo se ingresa el valor con el teclado numérico automático provisto por el emulador. En la segunda imagen vemos claramente el resultado esperado.



como también recursos, videos, código de ejemplos, entre otra ayuda más para los programadores, más el convenio firmado con NOKIA, harán de este sistema operativo de teléfonos móviles un producto grandioso. También hay que destacar que este ejemplo es ínfimo respecto al alcance de WP7

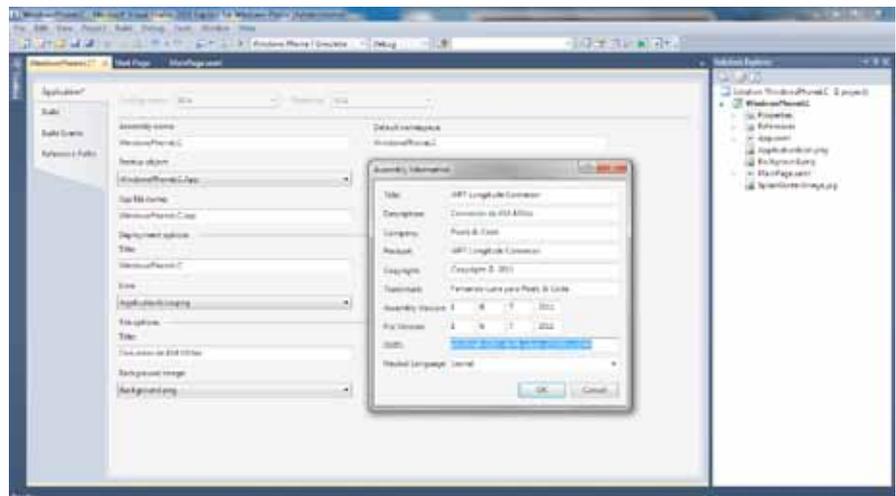
SDK. En este lenguaje de desarrollo podemos crear estas simples aplicaciones, como también juegos, aplicaciones basadas en GPS y Bing Maps, y también Apps con conexión a base de datos. En la próxima entrega podremos ver cómo desarrollar una aplicación Web con cualquier lenguaje de programación y encapsularla dentro de una aplicación instalable para WP 7.

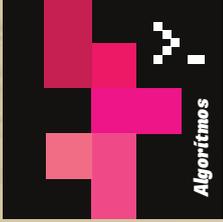
P

Al igual que ocurre con las aplicaciones instalables en Windows, Linux y Mac, VS2010 Express for Windows Phone nos permite incluir la información de autor, título de la app y versión de la misma.

yen en nuestro proyecto por defecto: ApplicationIcon.png, Background.png y SplashScreenImage.jpg sirven para crear un ícono de aplicación, establecer un fondo común a todas las pantallas y mostrar la pantalla de inicio respectivamente.

Conclusiones de Windows Phone 7 Este sistema operativo aún es nuevo en el mercado, pero con un trabajo bueno y ordenado, Microsoft ha logrado sacar un SDK de desarrollo poderoso y al alcance de cualquier usuario,

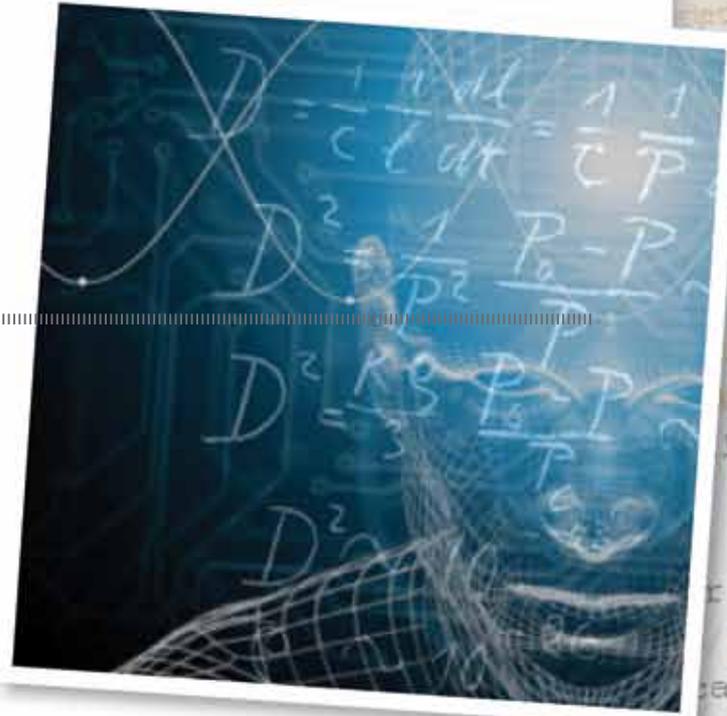




Algoritmos: quick sort

Por Juan Gutmann

juan.gutmann@pixelscode.com



Uno de los próceres de la informática es el británico Sir Charles Anthony Richard Hoare, conocido afectuosamente en la comunidad como “Tony”. Nacido en Sri Lanka en 1934, cuando todavía era colonia inglesa, el joven Tony se graduó en Ciencias Humanísticas en la Universidad de Oxford, en 1958. Dos años después, en 1960, el joven Hoare se encontraba aprendiendo ruso en la Unión Soviética, y simultáneamente estudiando cómo aplicar la informática en la traducción automatizada de idiomas. Allí desarrolló uno de los algoritmos de ordenamiento más simples y brillantes: QuickSort. Su objetivo era aplicarlo a un programa que traducía del ruso al inglés, ordenado las palabras a ser analizadas para compararlas más eficientemente contra un diccionario previamente almacenado en cinta magnética. Es sorprendente que un académico proveniente de la rama humanística desarrollara un algoritmo de ordenamiento que presenta tantas virtudes simultáneas: es fácil de comprender e implementar, y al mismo tiempo muy eficiente y veloz, lo que le da su nombre.

CARACTERÍSTICAS

Se trata de un miembro de la familia de algoritmos de ordenamiento comparativos, ya que recorre los elementos de la lista a ordenar, realizando comparaciones para hacer su trabajo. Al igual que Merge Sort, que analizamos el mes pasado, aplica la estrategia de “divide y conquistarás”, mediante el empleo de recursividad. La idea central es que en cada ejecución de Quicksort se elige un elemento cualquiera de la lista para utilizarlo como pivote. Luego se recorre la lista a ordenar, conformando dos nuevas sublistas: una con los elementos de valor menor al del pivote, y otra con los que sean mayores. Luego se repite el procedimiento con cada sublista, hasta que tengan uno o ningún elemento, en cuyo caso la lista ya está ordenada. En cada llamada, Quicksort finaliza concatenando la sublista conformada por elementos meno-

res al pivot con una sublista cuyo único elemento es el mismo pivot, y luego agregando al final de la unión de estas dos listas la sublista integrada por los elementos mayores al pivot. Tras una cantidad N de ejecuciones del algoritmo, la lista quedará ordenada.

En pseudocódigo, podemos expresarlo así:

Función QuickSort

Dada una lista L, de N elementos, y P, Q y R, listas vacías

```
si N <= 1
    devolver L
fin si
pivot = primer elemento de L
para cada elemento de L
    si elemento < pivot
        agregar elemento a lista P
    fin si
    si elemento > pivot
        agregar elemento a lista R
    fin si
fin para
agregar pivot a lista Q
lista ordenada = concatenar Quicksort(lista P) con lista Q con Quicksort(lista R)
devolver lista ordenada
```

Como podemos apreciar, el algoritmo es de una sencillez abrumadora, más aún teniendo en cuenta que cuando Tony Hoare lo concibió, tenía apenas 26 años de edad. Como de costumbre, veamos ahora una implementación en lenguaje Python, agregándole algunos "print" para poder visualizar mediante la salida en pantalla la forma de trabajo del algoritmo con mayor claridad.

```
def quicksort(L):
    N = len(L)
    P = []
    Q = []
    R = []

    print "Procesando Lista", L

    if N <= 1:
        return L
```

```

X = L[0]
for l in L:
    if l < X:
        P.append(l)
    elif l > X:
        R.append(l)
Q.append(X)
print "P =", P, "Q =", Q, "R =", R
return quicksort(P) + Q + quicksort(R)
Z = [1, 4, 8, 2, 5, 3, 9, 7, 10, 6]
print quicksort(Z)

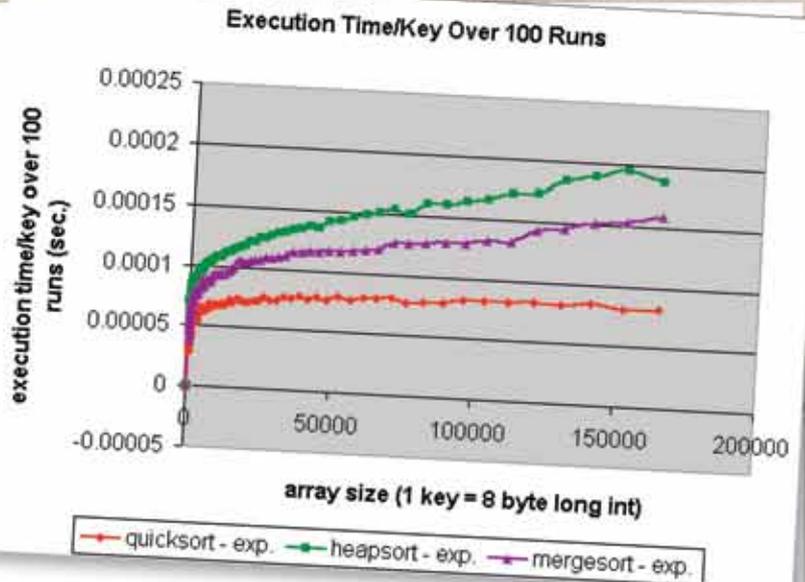
```

Como ya hemos mencionado en otras ocasiones, elegimos el lenguaje Python porque su sintaxis sencilla y elegante nos permite que la implementación sea muy similar a la versión en pseudocódigo. En este caso, dada la simpleza innata del algoritmo, el código resultante es casi idéntico al pseudocódigo, y comprenderlo es tan fácil que prácticamente carece de sentido analizarlo con mayor profundidad. Veamos el output que genera la implementación al ejecutarla:

```

Procesando Lista [1, 4, 8, 2, 5, 3, 9, 7, 10, 6]
P = [] Q = [1] R = [4, 8, 2, 5, 3, 9, 7, 10, 6]
Procesando Lista []
Procesando Lista [4, 8, 2, 5, 3, 9, 7, 10, 6]
P = [2, 3] Q = [4] R = [8, 5, 9, 7, 10, 6]
Procesando Lista [2, 3]
P = [] Q = [2] R = [3]
Procesando Lista []
Procesando Lista [3]
Procesando Lista [8, 5, 9, 7, 10, 6]
P = [5, 7, 6] Q = [8] R = [9, 10]
Procesando Lista [5, 7, 6]
P = [] Q = [5] R = [7, 6]
Procesando Lista []
Procesando Lista [7, 6]
P = [6] Q = [7] R = []
Procesando Lista [6]
Procesando Lista []
Procesando Lista [9, 10]
P = [] Q = [9] R = [10]
Procesando Lista []
Procesando Lista [10]

```



El gráfico demuestra la eficiencia de QuickSort. La curva resultante de representar tiempo de ejecución vs. tamaño de la lista es mucho más lineal que la de otros algoritmos muy eficientes, como HeapSort y MergeSort.

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

Como puede apreciarse, se resolvió el ordenamiento de una lista de diez elementos en apenas quince ejecuciones de la función Quicksort(), pero además, cada llamada se ejecuta con una gran velocidad, ya que en cada invocación a la función se aplica un único bucle iterativo, que recorre una sola vez la lista recibida como argumento. Además, hace un uso de recursos relativamente bajo. Es fácil detectar que este tipo de algoritmo puede beneficiarse enormemente con el empleo de caches, especialmente para trabajar con listas de grandes cantidades de elementos.

OPTIMIZACIONES

Otra ventaja de Quicksort sobre otros algoritmos de ordenamiento es que se pueden realizar diversas optimizaciones con gran facilidad. Dada la criticidad del valor pivot en el funcionamiento del algoritmo, podemos inferir con rapidez que trabajando sobre el criterio de elección del mismo, la cantidad de llamadas totales a la función pueden reducirse considerablemente. La implementación más común, como las demostradas hasta el momento, toman arbitrariamente un elemento cualquiera de la lista para emplearlo como pivot. Una típica elección es el primer elemento, aunque también es habitual inclinarse por el último. Una posibilidad es tomar como pivot un valor que no forme parte de la lista, cuyo valor sea el promedio entre el mínimo y máximo elemento. Este valor puede obtenerse con una sola recorrida adicional de la lista. Sin embargo, dependiendo de las características de la lista, en algunos casos no bajará la cantidad de ejecuciones, sino todo lo contrario. Además, la recorrida extra de la lista en cada corrida para determinar el valor del pivot ya de por sí alarga los tiempos de procesamiento. Una solución intermedia, muy utilizada, es evaluar tres elementos de la lista, por ejemplo el primero, el segundo y el último, y emplear como pivot al valor que se halle en el medio de los tres. De esta forma, no hace falta recorrer la lista completa para determinar el pivot, y se produce una mejora respecto de la implementación estándar en la cantidad total de ejecuciones, al margen de los valores que integren la lista y el grado de ordenamiento previo que esta tuviera.

Veamos una versión en Python que incorpora esta optimización:

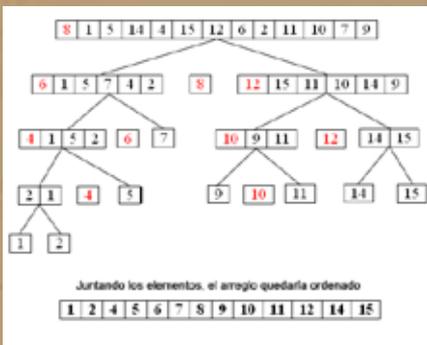
```
def quicksort(L):
    N = len(L)
    P = []
    Q = []
    R = []

    print "Procesando Lista", L

    if N <= 1:
        return L

    U = L[0]
    V = L[1]
    W = L[N - 1]

    if U >= V and U <= W:
        X = U
    elif V >= U and V <= W:
        X = V
    else:
        X = W
```



En este diagrama vemos la modalidad de funcionamiento de QuickSort. Pueden apreciarse claramente tanto el uso del valor pivot como la estructura de árbol, clásica de los algoritmos que aplican recursividad.

```
def quicksort(L):
    n = len(L)
    p = 1
    q = 1
    r = 1

    print "Procesando Lista", L

    if n <= 1:
        return L

    p = L[0]
    v = L[1]
    w = L[n-1]

    if v >= v and v <= w:
        x = v
    elif v >= v and v <= w:
        x = v
    else:
        x = w

    print "p, v, w, q, r, u, v, w, x, y, z"

    for i in L:
        if i < x:
            P.append(i)
        elif i > x:
            R.append(i)

    Q.append(x)

    print "p, v, w, q, r, u, v, w, x, y, z"

    return quicksort(P) + Q + quicksort(R)

z = [1, 4, 8, 2, 5, 3, 9, 7, 10, 6]
print quicksort(z)
```

Esta implementación de QuickSort incorpora una optimización muy simple en la elección del valor pivot, mejorando notoriamente los tiempos de ejecución.

```
print "U =", U, "V =", V, "W =", W, "X =", X
```

```
for l in L:
    if l < X:
        P.append(l)
    elif l > X:
        R.append(l)
```

```
Q.append(X)
```

```
print "P =", P, "Q =", Q, "R =", R
```

```
return quicksort(P) + Q + quicksort(R)
```

```
Z = [1, 4, 8, 2, 5, 3, 9, 7, 10, 6]
print quicksort(Z)
```

Y su salida por pantalla, mostrando también el criterio para elegir el pivot:

```
Procesando Lista [1, 4, 8, 2, 5, 3, 9, 7, 10, 6]
U = 1 V = 4 W = 6 X = 4
P = [1, 2, 3] Q = [4] R = [8, 5, 9, 7, 10, 6]
Procesando Lista [1, 2, 3]
U = 1 V = 2 W = 3 X = 2
P = [1] Q = [2] R = [3]
Procesando Lista [1]
Procesando Lista [3]
Procesando Lista [8, 5, 9, 7, 10, 6]
U = 8 V = 5 W = 6 X = 6
P = [5] Q = [6] R = [8, 9, 7, 10]
Procesando Lista [5]
Procesando Lista [8, 9, 7, 10]
U = 8 V = 9 W = 10 X = 9
P = [8, 7] Q = [9] R = [10]
Procesando Lista [8, 7]
U = 8 V = 7 W = 7 X = 7
P = [] Q = [7] R = [8]
Procesando Lista []
Procesando Lista [8]
Procesando Lista [10]
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

Observemos que para nuestra lista de prueba, la cantidad total de ejecuciones de redujo de quince a once, apenas una más que la cantidad total de elementos de la lista. Otra posibilidad para optimizar los tiempos de respuesta de QuickSort es, al igual que como se hace con otros algoritmos, combinarlo con un

segundo algoritmo para mejorar sus puntos débiles. La principal desventaja de Quicksort es que en sus últimas ejecuciones se ordenan listas de muy pocos elementos, y en la mayoría de los casos estas listas pequeñas estarán dispuestas de la forma equivalente al peor caso para el algoritmo, por lo que en estas pasadas finales baja mucho su eficiencia. Una alternativa propuesta por varios especialistas es aplicar Insertion Sort para las corridas que deben trabajar con listas de pocos elementos. Combinando ambos, se obtiene un algoritmo muy poderoso y veloz a la hora de trabajar con listas muy extensas.

ASPECTOS TEÓRICOS

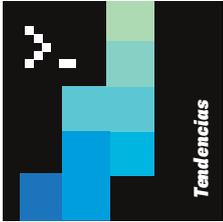
La complejidad promedio de Quicksort es de $O(n \log n)$, y dicho valor se degrada muy poco ante una lista cuyas características representen la peor situación posible para la modalidad de trabajo del algoritmo. Sus tiempos de ejecución suelen ser mejores que los de otros algoritmos de ordenamiento, independientemente de la longitud de la lista y el grado de ordenamiento previo que ésta tenga. Por esta razón, es un algoritmo ampliamente utilizado. Puede compararse por el procedimiento empleado para lograr el ordenamiento con el sistema de árbol binario, ya que ambos son muy similares en implementación, siendo la principal diferencia que en este último el árbol existe implícitamente, mientras que en Quicksort cada “hoja” se encuentra representada por la lista que recibe como argumento cada una de las llamadas recursivas. Comparándolo contra otros algoritmos clásicos de ordenamiento, suele contrastárselo con Heap Sort, cuya complejidad es siempre de $O(n \log n)$, inclusive ante la peor situación posible. Sin embargo, determinar cuál de ellos es más eficiente ante una lista promedio es una tarea difícil, y producto de un arduo debate entre los expertos. En general se sostiene que en esta batalla Quicksort resulta vencedor, ya que sus tiempos de ejecución suelen ser menores. Para paliar la deficiencia de este último ante una lista que represente el peor caso, se suele recurrir a un algoritmo combinado, conocido como “Introsort”, que detecta cuando el estado de la lista representa un caso poco ideal para Quicksort, recurriendo en esa instancia a un algoritmo de tipo HeapSort. Comparándolo con Merge Sort, el cual vimos en la edición anterior, es superado también ante la presencia de un caso poco propicio, ya que en esta situación la complejidad de Merge Sort también es de $O(n \log n)$. Además, la implementación de Merge Sort puede trabajar con muy pocos cambios con listas enlazadas, mientras que Quicksort requiere de cambios sustanciales. No obstante, para una lista tradicional, el algoritmo de Hoare realiza su trabajo más rápidamente y con una utilización de memoria considerablemente menor.

CONCLUSIÓN

Los invitamos a realizar pruebas y modificaciones a los algoritmos aquí publicados, ya que investigar por cuenta propia es la mejor forma de aprenderlos y comprender cuándo aplicarlos. Una herramienta útil para este tipo de tareas es el “List Randomizer” que se ofrece en el sitio web [<http://www.random.org/lists/>]. Es curioso pensar que el programador de esa página se enfrentó al problema opuesto al que nos ocupa en esta serie de notas: desordenar lo más posible una lista compuesta por una serie de elementos. Por cierto, a medida que seguimos adentrándonos en el fascinante mundo de los algoritmos de programación, queda cada vez más claro que los avances más importantes en este terreno se hicieron hace varias décadas, cuando todavía era reciente la invención de la computadora digital. Aunque los años nos fueron trayendo lenguajes cada vez más sencillos, flexibles y poderosos, los algoritmos que se emplean para resolver problemas usuales fueron concebidos por los pioneros originales de la informática. A los programadores modernos nos queda la humilde tarea de estudiar sus logros, para poder realizar nuestro trabajo de la forma más eficiente posible. En la próxima edición seguiremos recorriendo juntos este camino. **P**



En cada ejecución de Quicksort se elige un elemento cualquiera de la lista para utilizarlo como pivote. Luego se recorre la lista a ordenar, conformando dos nuevas sublistas: una con los elementos de valor menor al del pivote, y otra con los que sean mayores.”



Tecnología NFC

La llave de la instantaneidad y velocidad para móviles

Aunque existe desde hace muchos años, recién ahora empezaremos a ver los primeros productos que cuentan con la tecnología NFC. Esta plataforma sirve para transferencias de datos instantáneos, ideal para celulares o dispositivos móviles, y que abre las puertas a un universo de aplicaciones al alcance de tus manos. Aquí conocerás todo lo necesario para ofrecer tus propios desarrollos.

Por Roxana Miguel
roxana.miguel@pixelscode.com
@roxanamiguel

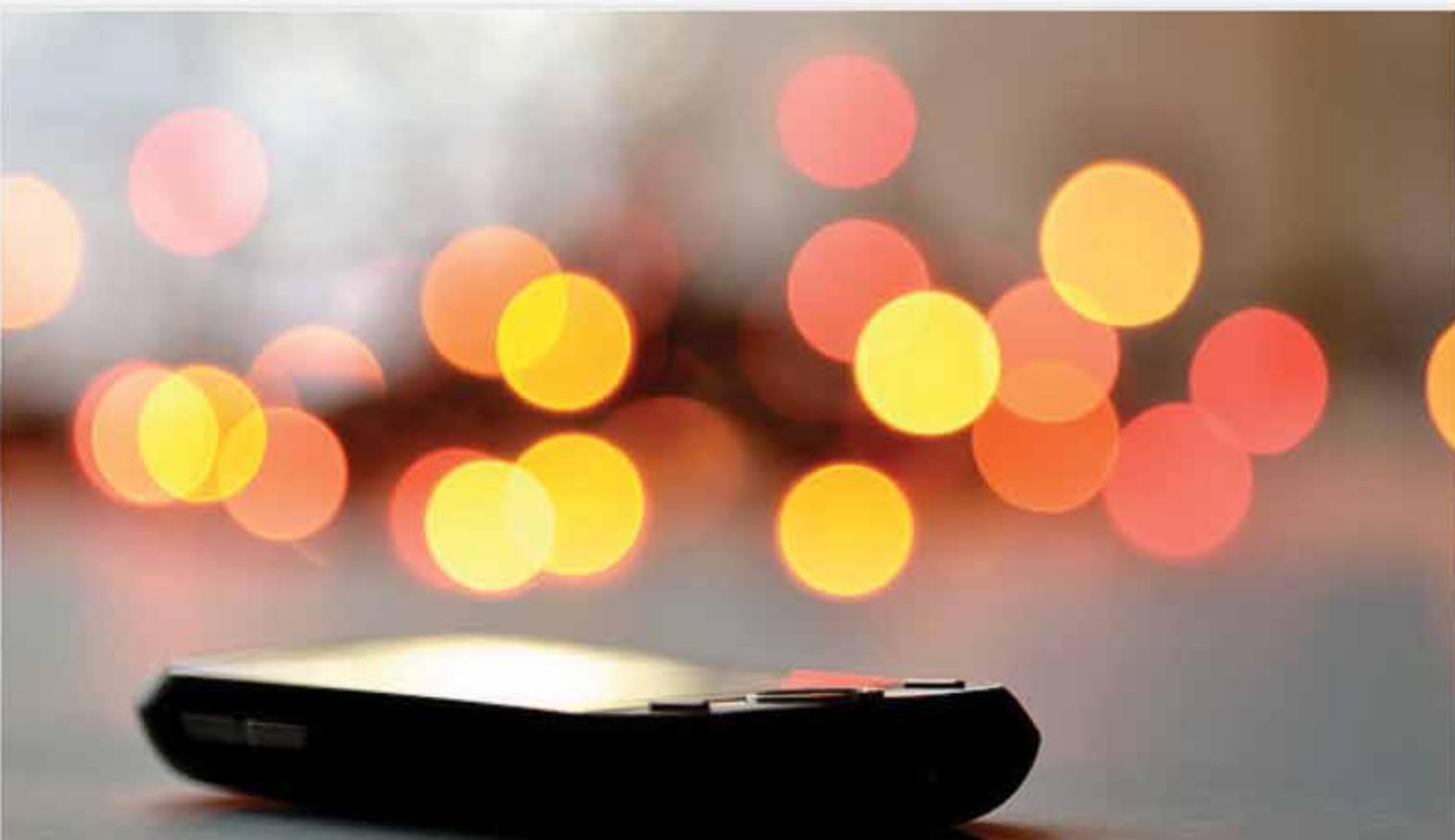


Atención desarrolladores, ha llegado el momento de poner el ojo y toda la creatividad sobre los pequeños y cada vez más reducidos dispositivos móviles, que además de comunicar a las personas, también ponen en juego todos sus datos en novedosos canales de información. Aquí no veremos cómo crear un portal web, sino cómo aprovechar las nuevas tendencias para acercarlas a las necesidades de nuestros clientes. En esta ocasión el protagonismo lo tiene NFC. Near Field Communication (de las siglas NFC) es una tecnología inalámbrica que funciona desde el año 2003 cuando fue aprobada como estándar ISO/IEC, y que recién en el año 2008 se incluye en los periféricos que además cuentan con Bluetooth. Hoy, compañías como Apple, Google y también varias operadoras, ponen a prueba sus funciones para dar el gran paso hacia una nueva era de las comunicaciones.

Básicamente se trata de una plataforma abierta inalámbrica para dispositivos móviles que utiliza una banda, de la cual no necesita licencia, de 13.56 MHz, además de la etiqueta RFID (Radio Frequency Identification) y que ocupa una tasa de transferencia que puede alcanzar los 424 kbits/s. Con todo esto, lo que sencillamente significa es que sirve para transmitir datos mediante una comunicación instantánea y de corto alcance ya que sólo funciona en un rango de 20 cm. Esta reducida transmisión entre un objeto emisor, que puede ser un celular, y un objeto receptor, que puede ser una terminal de cobro u otro celular, permite que un móvil se transforme en una tarjeta de presentación para registrarse en un lugar, una tarjeta de crédito para abonar un servicio o un canal directo para recibir y enviar pequeñas cuotas de datos de manera rápida y segura.

¿CUÁLES SON LAS APLICACIONES PARA NFC?

La tecnología ha ayudado y mucho a acortar las distancias, es así como con una webcam se pueden organizar reuniones con personas que están en distintas partes del mundo. Pero no todo requiere de tan amplias longitudes, hoy los desarrolladores tienen que poner el ojo sobre la creación de medios de comunicación de corto alcance por medio de etiquetas y chips. El desafío ya está en el mercado. Los primeros en poner a prueba los alcances de la tecnología NFC fueron Apple y Google. Actualmente, muchos locales de Starbucks de Estados Unidos pueden cobrar sus órdenes mediante NFC con tan sólo acercar el teléfono, disponer el pedido desde la app y abonarlo evitando las colas, el manejo de dinero en efectivo y ofreciendo más tiempo de disfrute. Lo mismo permite el sistema ope-



rativo Android. Con esta tecnología pronto será posible identificarnos, pagar, hacer el check in en un lugar, usarlo como parquímetro, llaves electrónicas, es decir, todas aquellas aplicaciones que puedas imaginar que se basten de dos objetos que transfieran datos. Al arrastrar información en espacios cortos y una buena velocidad, se logra eficacia y seguridad, dado que se puede utilizar en espacios abiertos sin riesgos de que un curioso pueda tomar información sin pedirla.

Se estima que en los próximos cinco años la tecnología NFC complementará a los teléfonos del mercado en todo el mundo garantizando utilidades como pagos, obtención de descuentos y ofertas por medio de un póster que posea una etiqueta NFC, administración de entradas de garaje, cine y conciertos, como así también tarjetas de contacto, acceso seguro a edificios y autos, hasta el traspaso inalámbrico

de archivos digitales a impresoras sin necesidad de Bluetooth.

LOS RUMORES DE NFC PARA EL 2011

Que no todas las compañías cuenten con este tipo de tecnología se debe a que está en constante observación. Sin embargo, se puede ver cómo las empresas se ven tentadas a implementarlas. Una de las telefónicas que lo incluirá en los desarrollos de este ciclo será Nokia, mientras que Apple anunció que también lo hará en sus próximos equipos, al igual que Blackberry. Google es, quizás, el más avanzado en tanto que el modelo Nexus S ya cuenta con NFC, y hasta ya se brinda soporte para el mismo en Android 2.3.

LOS RUMORES DE NFC PARA EL FUTURO

La promesa de esta tecnología puede sorprender a varios y para ello hemos



Near Field Communication (de las siglas NFC) es una tecnología inalámbrica que funciona desde el año 2003 cuando fue aprobada como estándar ISO/IEC, y que recién en el año 2008 se incluye en los periféricos que además cuentan con Bluetooth.”



“Se estima que en los próximos cinco años la tecnología NFC complementará a los teléfonos del mercado en todo el mundo garantizando utilidades como pagos, obtención de descuentos y ofertas.”

preparado algunas acciones cotidianas que quizás vivamos a futuro, con NFC en mano:

- Una familia podrá distribuir el dinero entre los integrantes encargados de pagar servicios o realizar compras con tan sólo apoyar los Smartphones entre sí y con la opción de transferir dinero. Luego, el que tiene que hacer las compras no deberá cargar con tarjetas, ni efectivo, sino que abonará el consumo con su celular pero sobre otra terminal de destino.
- Ya no tendrás que contar con manojos de llaves para abrir el auto, tu casa o la oficina. Simplemente

apoya tu teléfono en el destino, éste te solicitará una autenticación del pedido por medio de un control por reconocimiento facial (usando la cámara del celular) o voz (usando el micrófono). De esa forma, si pierdes el celular, no corres ningún riesgo.

- Cuando las medidas de seguridad sean las apropiadas, se estima que los portátiles con NFC podrán reemplazar tus documentos personales ya que estarán cargados en el chip permitiéndote identificarte si te para un policía de tránsito o acreditando tu identidad en una oficina municipal.

- Podrás dejar tus etiquetas RFID en los muros de Facebook o tuitearla para que otras personas que posean NFC decodifiquen esa información directamente desde las redes sociales.

- Los centros de salud podrán tener registros de sus pacientes mediante NFC y por cada visita ingresar los datos como presión, temperatura y pulso. Para los casos de emergencia, esta aplicación disminuirá con-

siderablemente el tiempo utilizado en registros administrativos.

¿CÓMO FUNCIONA?

Aunque no todos los dispositivos de la nueva generación cuentan con tecnología NFC, pronto estos estarán en las manos de los usuarios, entonces este es el momento de desarrollar nuestra creatividad para que los clientes potencien aún más su espacio en el mercado. Según la publicación Technology Review las aplicaciones móviles se han convertido en una mina de oro para los desarrolladores, convirtiéndolos en los obreros más requeridos para lograr que los teléfonos inteligentes sean cada vez más inteligentes.

El funcionamiento es sencillo. Dos dispositivos se comunican entre sí, uno tomará el rol de “iniciador” de la comunicación y el otro de “destino” debiendo responder como tales antes de recibir información. La comunicación puede ser pasiva (donde el destino recibe los datos de un campo electromagnético generado por el iniciador y confirma su recepción) o activa (que se da cuando ambos dispositivos –iniciador y destino– emiten un campo electromagnético y negocian el intercambio de datos).

La secuencia de transacción empieza con el descubrimiento del dispositivo NFC, luego de autenticarlo comienza la negociación, la transferencia de datos y finalmente la confirmación. Para que este paso sea seguro se pueden utilizar cifrados AES y triple DES, que brinda una seguridad similar a la de las tarjetas de crédito. A su vez, el protocolo permite definir durante la negociación, la velocidad, el identificador, tipo de aplicación, tamaño de

transferencia y acción solicitada. Quedará en manos de un futuro no muy lejano que el NFC Forum, entidad creadora del protocolo, facilite el desarrollo de aplicaciones estandarizando la comunicación entre el chip NFC y los procesadores. Mientras que el sistema de comunicaciones móviles (GSMA) deberá por su parte estandarizar la comunicación entre la tarjeta SIM y el chip por medio del protocolo Single Wire Protocol (SWP).

Sin embargo, por ser una plataforma abierta lo destacable es que hoy los desafíos son para los desarrolladores que pueden innovar con aplicaciones adicionales, pero de necesitar ayuda se puede entrar al foro de Nokia (www.forum.nokia.com) donde están disponibles los kits de desarrollo de software con recursos y recomendaciones sólo para registrados, que además podrán enriquecer tus conocimientos compartiendo experiencias piloto en el foro de la comunidad.

LOS ESTÁNDARES DE COMUNICACIÓN INALÁMBRICA COMPITEN

Aunque es sólo una forma de mencionar la convivencia entre Bluetooth, Wi-Fi y NFC, porque estas tecnologías se complementan, las características de cada uno presentan tres importantes diferencias: Cobertura: Bluetooth y Wi-Fi ofrecen un alcance que va desde 1 a 100 metros. Mientras que NFC sólo toma datos dentro de los 20 cm como máximo.

Seguridad: La diferencia de alcance pone en riesgo los datos de los usuarios que se conectan por medio de Wi-Fi o Bluetooth sin autenticación, dejando el acceso libre. Por las cor-

tas distancias necesarias para NFC el riesgo disminuye notablemente, además de que el protocolo exige un estadio de autenticación y otro de confirmación con opciones automáticas en pantalla.

Conveniencia: Para que las vías tradicionales sean seguras se puede requerir un password, pero esto genera una dificultad para las comunicaciones instantáneas, siendo esta una de las características básicas a favor de NFC.

TODO LO QUE HOY PUEDES HACER CON NFC

Creación de etiquetas: Como hemos podido ver hasta aquí, NFC no es más que un método de envío de datos de forma inalámbrica. Esto engloba infinitas posibilidades de aplicaciones para construir y poner a prueba, y todo mediante la creación de etiquetas que indiquen la acción o el objetivo de una comunicación. Cuando una etiqueta es descifrada por un dispositivo NFC se pueden lograr accesos más interesantes aún. Puedes acceder a ellas mediante el market de Android, por ejemplo, que dispone de apps para que puedas crear tus propias etiquetas de acuerdo a lo que te sea más útil: que al entrar a tu oficina el celular se configure automáticamente a modo silencioso, es un buen ejemplo. Terminales para fomentar el turismo: Diggia, emprendimiento de la empresa española Gamma Solutions, ofrece a los allegados a tierras españolas un dispositivo NFC con información de descuentos, museos, entre otros servicios que les permita un acceso a los espacios turísticos tanto para el disfrute como para el consumo. Los usuarios que lo deseen podrán configurarlo para pagar sus gastos del viaje mediante la terminal.

Jugar Angry Birds Magic con NFC: Se trata de un juego por demás sencillo pero que se ha convertido en uno de los más populares. Con la aplicación NFC se han asegurado que esto último sea así ya que si pasas tu Smartphone junto a otro con NFC podrás avanzar cinco niveles y desbloquearlos usando las etiquetas que encontrarás en los objetos del juego. Los equipos que Nokia lance durante el 2011 contarán con la tecnología y, por ende, la batalla en grupo contra los cerdos, ¡asegurada!

Escucha tu música: El Parrot Boombox es a simple vista un altavoz, sin embargo se destaca porque puedes escuchar música conectándolo a tu mp3, o mejor aún, apoyado tu móvil NFC. Escucharás con una potencia de 60 w y una respuesta de frecuencia de 80 Hz – 20 KHz.

Del cartel a tus auriculares con Smart Poster: Así se llama la aplicación creada por Philips, Visa y Universal Music Francia que colocó chips en los carteles callejeros presentando lanzamientos musicales. Quienes poseen un dispositivo NFC lo apoyan en el poster e inmediatamente cargan el álbum en su celular o bien, compran la canción y obtienen un pass para descargarla en Internet.

Para cerrar esta nota lo ideal sería que puedas apoyar tu dispositivo NFC en este párrafo y descargues tu Pixels&Code automáticamente. Sin embargo, para dar vuelta la página tendremos que esperar quizás unos meses, o unos segundos, el tiempo que la tecnología necesita para instalarse. Por lo pronto, ahora sabemos lo necesario para poner manos a la obra y sabemos que en estos campos el tiempo apremia. **P**

Email Marketing con tu marca

Ganá dinero ofreciendo a tus
clientes la mejor herramienta.

