



SCRUM

Desarrollando
software en equipo

ABRIL 2012 / N° 12



— Testing de aplicaciones Android

— Entrevista con los creadores del Panel Ferozo

SUMARIO

ENTRY POINT	4
INTRODUCCIÓN A LAS BASES DE DATOS RELACIONALES(XII)	6
PROYECTO #CODE: PANEL DE CONTROL FEROZO	14
DESARROLLO DE SOFTWARE ÁGIL CON SCRUM	22
TESTING DE APLICACIONES	30
LAS LEYES DE PROGRAMACIÓN DE TAYLOR (PARTE II)	40
CHOQUE DE GIGANTES: F ACEBOOK VS GOOGLE+	48

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

STAFF

DIRECTORA Y JEFA DE REDACCIÓN

Débora Orué

COLUMNISTAS

Gabriel Walter Gaitano Ornia

Juan Gutmann

Mariano Salvetti

Matías Iácono

Roxana Miguel

DISEÑO EDITORIAL Y CREATIVIDAD

www.americanbread.net

Arte de tapa: Esteban Paniagua

COMERCIALIZACIÓN

DATTATEC.COM SRL

Córdoba 3753, Rosario, Santa fe

www.dattatec.com

REDACCIÓN

lectores@pixelscode.com

DATOS DE CONTACTO

Dirección comercial:

publicidad@pixelscode.com

ENTRY POINT ENTRY POINT



GADGETS:

Y, finalmente, la empresa de la manzana lo hizo de nuevo: Apple presentó en sociedad una nueva versión de su popular tableta iPad, largamente esperada por el aficionado a la tecnología.

Equipada con un procesador más rápido, dotada con tecnología 4G, es considerada una verdadera revolución. Esta es la tercera versión que llega a las vitrinas, con la que espera conservar el liderazgo del mercado que le es intensamente disputado por empresas como Amazon y Samsung, entre otras. Aún no se la ha liberado en nuestro país, pero ya hay gran expectativa por su desembarco en estas playas.



SEGURIDAD E INTERNET

Luego de las películas le llega el turno a los libros. Un importante grupo de casas editoriales de EE.UU. y Europa ha apuntado su artillería contra el sitio Library.nu, que era considerado como un formidable reservorio de ebooks y que, con una modalidad similar a Megaupload, alcanzaba una recaudación en publicidad de más de 10 millones de dólares.

En este sitio los usuarios subían sus datos y recibían un link que podían compartir con sus otros contactos. Los servicios eran utilizados generalmente para el alojamiento de libros digitales que, de este modo, podían ser compartidos o leídos en formato PDF.

El sitio ha sido dejado inoperativo por requisitoria judicial, dejando en claro que la batalla por el modelo de comercialización y compartición de los medios por la red no ha hecho más que empezar.



WEB PARA CONOCER:

A pesar de estar fuertemente orientada a los productos Microsoft, si nuestro trabajo nos lleva a utilizarlos como developers tenemos un excelente recurso disponible para utilizar en el sitio <http://www.codeproject.com>.

Aquí vamos a encontrar una sabrosa combinación de artículos, Blogs técnicos con datos de consultas, trucos interesantes de programación y, como no podía faltar, un foro de discusión, así como un repositorio de preguntas donde podemos plantear nuestras propias dudas. C#, ASP.NET y otras tecnologías de la empresa de Redmond aparecen en estos medios con abundante información práctica y muy útil en nuestro trabajo con esas tecnologías.



IT NEWS:

Problemas con el nuevo Windows 8

Microsoft había, sin duda, imaginado que algunos usuarios encontrarían la transición y cambio desde el actual producto líder, Windows 7, a su planeado sucesor Windows 8 como algo desafiante. Pero es poco probable que estuviera preparada para la reacción de su propia comunidad base. Esta se ha manifestado en el Windows 8 Consumer Preview, o revisión inicial de consumidores. Especialmente denostadas en los comentarios de los usuarios han sido la nueva interfaz estilo tablet, llamada Metro y concebida para competir con el popular iPad de Apple; y la desaparición del viejo y querido botón de Inicio. De hecho, muchos usuarios han señalado que el producto entero es un “substantial paso hacia atrás”.

Las opciones para Microsoft son, ahora, demorar el producto para tratar de suavizar los cambios, por ejemplo, haciendo la nueva interfaz optativa (salvo para tabletas y teléfonos) o seguir adelante confiando que el usuario acabará por acostumbrarse a los cambios. Que esta vez son mucho más de lo acostumbrado.

INTRODUCCIÓN A LAS BASES DE DATOS RELACIONALES(XII)

Hace unas semanas, un lector nos escribía consultándonos acerca de la posibilidad de trabajar con una base de datos relacional en una aplicación para smartphones. Esta es una pregunta que se escucha cada vez más en los últimos tiempos. Cuando se desarrolla para dispositivos embebidos, ya sean equipos muy pequeños y específicos (como un controlador de maquinaria industrial) o más voluminosos y versátiles -smartphones o tablets- nos encontraremos con una situación en la que necesitamos hacer mucho con pocos recursos. La cantidad de memoria y el poder de procesamiento del CPU de este tipo de aparatos son más bien limitados, en comparación con una PC hogareña.

Ni hablar si lo comparamos contra un gran servidor corporativo, donde normalmente corre el software de base de datos relacional en las empresas medianas y grandes. ¿Esto significa que tenemos que renunciar a programar como más nos gusta, para complicarnos la existencia almacenando la información de otra manera? Afortunadamente, no. Aunque tendremos que prescindir de bases potentes y flexibles, como lo es PostgreSQL, existen alternativas pensadas para estos casos que permiten trabajar con un motor relacional lo suficientemente funcional como para satisfacer las necesidades de cualquier aplicación escrita para este tipo de plataformas. Aquí presentamos una de ellas: SQLite.





CARACTERÍSTICAS DE SQLITE



SQLite no es un motor de base de datos relacional, aunque trabaja de forma idéntica a uno. Ocurre que no se trata de un programa (o conjunto de programas) “servidor”, que se encuentra corriendo permanentemente (como un demonio), atendiendo peticiones en un puerto de red. En cambio, toma la forma de una pequeña librería desarrollada en lenguaje C. Sus principales virtudes están bien descritas en la página web del proyecto, [<http://sqlite.org/>]: es “self-contained” (no posee dependencias de otras librerías), “serverless” (según acabamos de describir, no requiere de un proceso ejecutándose en forma constante), “zero configuration” (no requiere realizar tareas de instalación ni configuración) y “transactional” (transaccional; como nuestros lectores ya saben, esto quiere decir que pueden agruparse una serie de cambios realizados a la información contenida para que sean confirmados o deshechos en conjunto en el caso de existir algún tipo de error). Al escribir una aplicación que trabaje con ella, podemos leer y escribir datos en un archivo “base”. Por ello es apropiada

para su empleo en dispositivos móviles y embebidos, ya que al no existir un motor propiamente dicho, el empleo de la base de datos tiene costo cero en recursos, ya que no existe un servidor que requiera de memoria RAM, espacio en el sistema de archivos y ciclos de CPU para funcionar. Los recursos consumidos dependerán exclusivamente de la forma en la que programemos nuestro aplicativo. Para su pequeño “footprint”, SQLite es notablemente flexible. Aunque debemos renunciar al empleo de Stored Functions (los Procedimientos Almacenados, que nos vienen dando tan buenos resultados con PostGres), sí cuenta con la posibilidad de crear Triggers, aunque su funcionalidad -claro está- es acotada. SQLite corre en prácticamente cualquier plataforma para la que exista un compilador de lenguaje C, y lo mejor de todo es que para programar software que la utilice no nos encontramos limitados a este mismo lenguaje, sino que podemos hacerlo en casi cualquier tecnología de desarrollo que se nos ocurra: C++, Java, JavaScript, Lisp, Perl, PHP, Python, Ruby y Tcl son apenas algunos de ellos.

ORÍGENES DE SQLITE

La librería SQLite fue creada por el programador norteamericano Richard Hipp, cuya labor ya era reconocida en el mundo del Software Libre y de Código Abierto por ser uno de los desarrolladores del lenguaje Tcl. Cuando fue empleado por la empresa General Dynamics para mantener el software encargado de controlar barcos de guerra de la U.S. Navy (la marina de guerra norteamericana), se le solicitó modificar los programas críticos para poder prescindir del pesado motor de base de datos relacional con el que trabajaban (Informix). Richard estaba muy familiarizado con el Proyecto GNU, el emprendimiento de Richard Stallman para crear un Sistema Operativo totalmente libre, gracias a su trabajo en el lenguaje Tcl. Sabía que uno de los paquetes que componen el proyecto, llamado gdbm (una versión no propietaria de la librería dbm para UNIX, que permite implementar un sencillo sistema de base de datos) cumplía con los requerimientos que le habían sido especificados. Tomando el código de gdbm, Hipp puso manos a la obra, y luego de algunas semanas de trabajo nació la versión 1.0 de SQLite. Esta primera versión era

una especie de “wrapper” de gdbm, que le permitía escribir en SQL los comandos para crear, modificar o borrar la información contenida en la base, delegando al código original del Proyecto GNU la tarea de realizar físicamente las lecturas y escrituras necesarias en el sistema de archivos. La segunda versión de SQLite (la 2.0) sustituyó definitivamente el código de gdbm, implementando una base propia, basada en B-trees.

A esta altura del proyecto, miles de usuarios estaban involucrados en su utilización y mejora, entre ellos un número importante de empresas. Una de las más grandes, AOL, puso una suma importante de dinero para acelerar el desarrollo de SQLite, tras lo cual fue liberada la siguiente versión (3.0), que incorporó un sinnúmero de mejoras. La última versión estable, la 3.7.10, publicada los primeros días del corriente año, se basa en dicha versión.



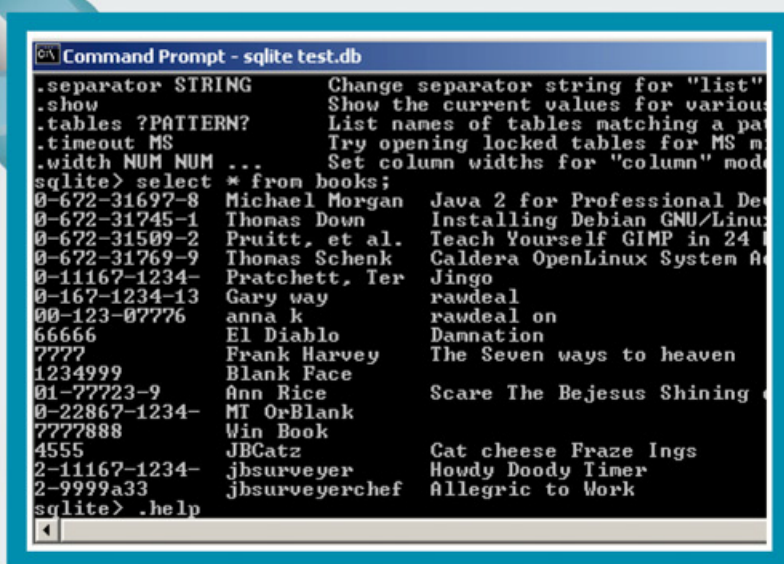
SQLite es una base de datos ideal para emplear en teléfonos móviles, a punto tal que tanto el Sistema Operativo del iPhone (iOS) y el de los teléfonos basados en Android incorporan esta librería.



LICENCIA DE SQLITE

Un factor determinante a la hora de elegir un sistema de base de datos es su licencia. La licencia GPL de MySQL, por ejemplo, impide incorporarla a una aplicación propietaria, salvo que se opte por recurrir a una versión privativa del motor, que sí permite este tipo de usos, pero no es gratuita. La licencia de SQLite es otro punto a favor en este sentido a la hora de desarrollar para dispositivos con escasos recursos.

Tanto las versiones binarias de la librería como el código fuente se encuentran liberadas en el “dominio público”, lo que permite su uso, copia, modificación, distribución e incluso su venta bajo cualquier circunstancia, incluyendo su incorporación a un software propietario.



```
Command Prompt - sqlite test.db
sqlite> .separator STRING      Change separator string for "list"
sqlite> .show                  Show the current values for various
sqlite> .tables ?PATTERN?     List names of tables matching a pa
sqlite> .timeout MS           Try opening locked tables for MS m
sqlite> .width NUM NUM ...    Set column widths for "column" mode
sqlite> select * from books;
0-672-31697-8  Michael Morgan  Java 2 for Professional Dev
0-672-31745-1  Thomas Down    Installing Debian GNU/Linu
0-672-31509-2  Pruitt, et al.  Teach Yourself GIMP in 24
0-672-31769-9  Thomas Schenk  Caldera OpenLinux System A
0-11167-1234- Pratchett, Ter  Jingo
0-167-1234-13  Gary way       rawdeal
00-123-07776   anna k         rawdeal on
66666         El Diablo      Damnation
7777         Frank Harvey   The Seven ways to heaven
1234999       Blank Face
01-77723-9    Ann Rice       Scare The Bejesus Shining
0-22867-1234- MT OrBlank
7777888       Win Book
4555         JBCatz        Cat cheese Fraze Ings
2-11167-1234- jbsurveyer     Howdy Doody Timer
2-9999a33     jbsurveyerchef Allegric to Work
sqlite> .help
```

El shell de SQLite permite acceder a las bases de datos sin necesidad de escribir programas en alguno de los lenguajes soportados.

DESCARGA E INSTALACIÓN

Algunos lenguajes, como PHP o Python, ya incluyen SQLite en sus paquetes de instalación. Para desarrollar en otros lenguajes con SQLite tenemos diversas alternativas. Bajo GNU/Linux, puede obtenerse mediante el administrador de paquetes correspondiente a cada distribución. Pueden descargarse del sitio oficial de SQLite los fuentes de la librería y compilarlos con el compilador de C de su preferencia, bajar directamente los binarios precompilados para diversas plataformas.

En el sitio se ofrecen binarios para Linux, Mac OS X, Windows y .NET. También se ofrecen junto con los binarios de las librerías un shell de comandos ejecutable, que permite ingresar directamente en la consola las órdenes SQL para manipular las bases de datos SQLite. Por último, debe obtenerse la librería de “binding” que corresponda a cada lenguaje. Para este último paso, se aconseja recurrir a la ayuda online de dicho lenguaje.

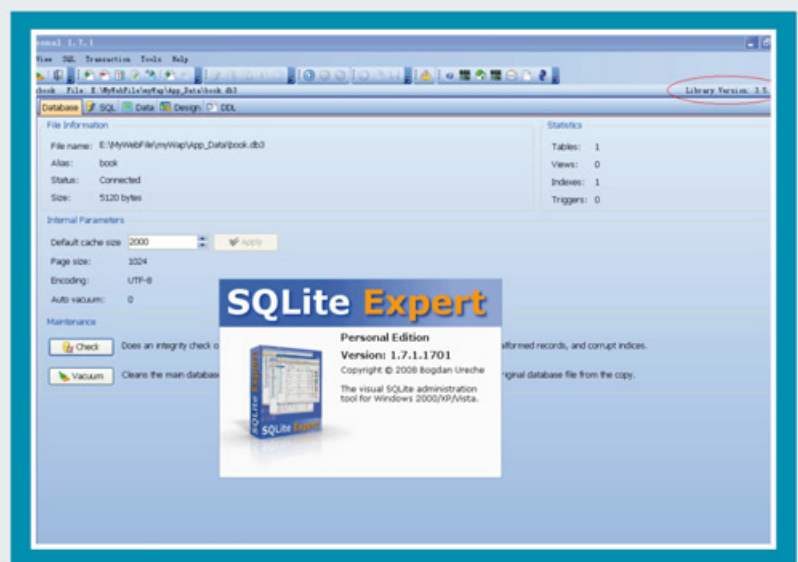
SOPORTE Y DOCUMENTACIÓN

En el sitio web de SQLite se encuentra publicada abundante documentación sobre sus características y utilización, así como una completa referencia de la librería, sus APIs desde distintos lenguajes (como C++ y Tcl), tutoriales y ejemplos, y también enlaces hacia libros (online y en papel) publicados por distintos autores sobre el desarrollo de aplicaciones en conjunto con esta base de datos relacional.

Además, su autor original, Richard Hipp, y su empresa (Hwaci) ofrecen soporte e inclusive la posibilidad de realizar modificaciones específicas a pedido, según ellos “por un precio

módico”. También figuran en el sitio algunas listas de correo a las que suscribirse para enterarse de las últimas novedades, intercambiar información y solicitar ayuda.

Por último, aunque SQLite es libre y gratuito, pueden conseguirse allí enlaces hacia extensiones propietarias (algunas de ellas pagas) que agregan funcionalidad no ofrecida por la versión estándar de la librería. La más interesante de ellas es SEE (SQLite Encryption Extension) que permite trabajar con una base de datos cifrada con potentes algoritmos AES de 128 y 256 bits.



Como casi todos los motores relacionales que se precien de tal, SQLite también cuenta con sofisticadas herramientas visuales desarrolladas por terceros que permiten administrar y mantener la base de datos.



UN EJEMPLO EN PYTHON

A continuación, un ejemplo de código Python que trabaja con el “binding” de SQLite. En este ejemplo, que se extiende por apenas unas pocas líneas de código (siguiendo la tradición “pythónica”) se comienza por crear una base permanente en el sistema de archivos raíz (puede crearse una base temporal en memoria, que desaparecerá cuando finalice la ejecución de la aplicación), prosigue creando una tabla y ejecutando SQL INSERTs para generar dos registros, y finaliza leyendo el contenido de la tabla y volcándolo a pantalla.

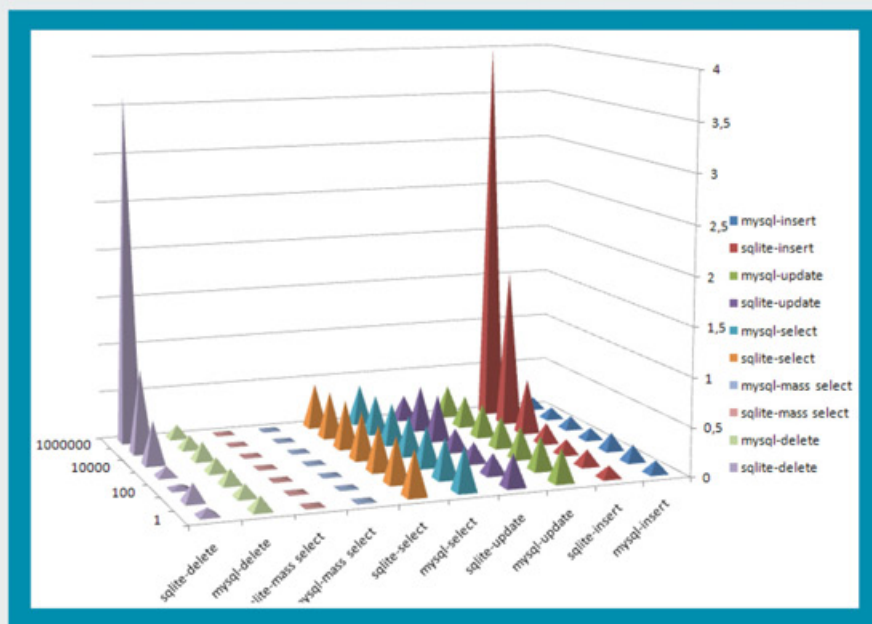


Gráfico comparativo de la performance de MySQL contra la de SQLite. Aunque MySQL gana por poco, su consumo de recursos muy superior inclina la balanza en favor de la librería que nos ocupa en esta edición.

CONCLUSIONES

```
from sqlite3 import dbapi2 as sqlite
connection = sqlite.connect(¿tmp/
prueba.db)
cursor = connection.cursor()
query = «CREATE TABLE empleado
(legajo INTEGER PRIMARY KEY, cargo
VARCHAR(10),»
query += «apellido_y_nombre
VARCHAR(100), supervisor
INTEGER)»
cursor.execute(query)
query = «INSERT INTO empleado
VALUES «
query += «(1, ¿GTEGRAL), ¿Carlos
Gomez), 1)»
cursor.execute(query)
query = «INSERT INTO empleado
VALUES «
query += «(2, ¿SUPERVISOR), ¿Jorge
Gonzalez), 1)»
cursor.execute(query)
connection.commit()
cursor.execute(«SELECT * FROM
empleado»)
resultset = cursor.fetchall()
for row in resultset:
    print row[0], row[1], row[2],
row[3]
connection.close()
```

Unos pocos conocimientos del trabajo con cualquier motor relacional, y un equipo informático con algo de RAM libre alcanzan para trabajar con SQLite.

El lenguaje SQL es una forma sencilla y eficaz de almacenar, modificar y recuperar información de cierta complejidad. No por nada entre los usuarios de esta excelente librería se cuentan gigantes de la industria informática, como Mozilla, Google, Apple, Nokia y Adobe. Por supuesto, el hecho de que esta base sea ideal para trabajar en dispositivos embebidos y móviles no lo limita a estas plataformas; emplearla en una aplicación hecha y derecha para las PC tradicionales es una buena idea, si tenemos estructuras complejas de datos con las que queremos interactuar, y preferimos obviar la dependencia de un motor tradicional, como MySQL.

En definitiva, contar a SQLite entre nuestro arsenal de conocimientos dentro del mundo de las bases de datos amplía nuestros horizontes en gran manera, y por lo tanto es imprescindible tener aunque sea nociones básicas de su empleo.





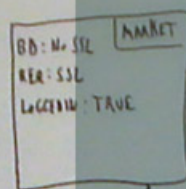
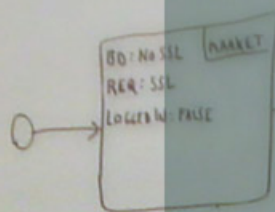
PROYECTO #CODE: PANEL DE CONTROL FERROZO

ENTREVISTA

Ing. Daniel Lázaro,
Coordinador
Área Desarrollo de Dattatec.

Bienvenidos a esta nueva sección de Pixels&Code que hemos decidido llamar “Proyecto #Code”. La idea de esta sección es que sea una puerta para compartir con ustedes los diferentes programas que revolucionan el mundo del software. Para ello nos comprometemos a involucrarnos de lleno en los diferentes proyectos que aquí destaquemos, yendo directamente a las fuentes y entrevistando a las personas que están directamente relacionadas con el proyecto del mes.

En esta primera entrega de Proyecto #Code, decidimos empezar por casa y reunirnos con Daniel Lázaro, quien es la persona encargada de una de las piedras basales de Dattatec: el panel de control Ferozo, un software que tiene como objetivo construir proyectos Web de fácil administración, con el cual podamos tener presencia en internet de manera inmediata, para volcar nuestras ideas y tener una amplia llegada hacia nuestros clientes.



Franco Rivero
franco.rivero@pixelscode.com

Converte tu sitio en un BLOG, FORO, y más con sólo un click!

Con Ferozo podrás instalar decenas de aplicaciones en tu hosting con sólo algunos clicks, desde una interfaz sumamente intuitiva y rápida.



Un aliado indispensable para tu empresa de hosting.

Ferozo permite convertir un servidor en una plataforma completamente automatizada, reemplazando tareas complejas por una interfaz de administración simple y muy intuitiva.

Pixels&Code: Daniel, contale a nuestros lectores de qué se trata Ferozo.

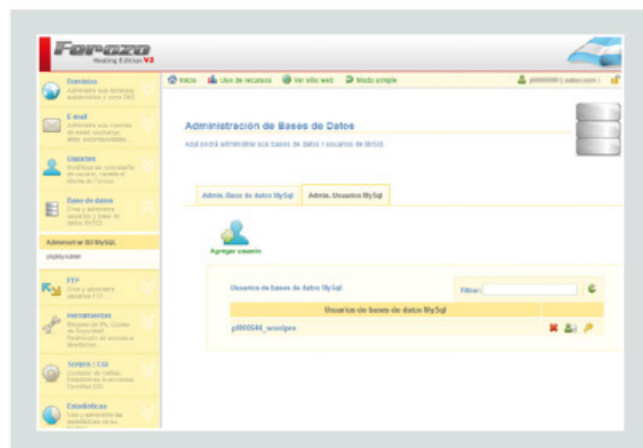
Daniel Lázaro: Ferozo es un panel de control para hosting, pionero a nivel nacional, 100% desarrollo propio, con versiones en las dos plataformas más importantes de hosting (Linux y Windows), y dividido convenientemente en módulos que abarcan todos los estratos de usuarios y necesidades (panel para usuarios de hosting, panel para revendedores de hosting y administradores de servidores, panel para usuarios de webmail y constructor on-line de sitios web).

Ferozo es un proyecto enteramente desarrollado en Rosario, que tiene alcance mundial a través del uso que los clientes de Dattatec le dan como panel de control para (auto) gestionar los recursos del servicio de hosting que la empresa brinda.

P&C: ¿Qué diferencias radicales distancian a Ferozo de otros sistemas de gestión de contenidos como Wordpress?

DL: Bueno, para empezar, las diferencias son muchas ya que Ferozo no es un Sistema de Gestión de Contenidos (CMS) como WordPress, sino un Panel de Control para Hosting.

Para explicarlo de una forma sencilla voy a hacer abuso de un ejemplo comparativo que me agrada. Las empresas que se dedican al servicio de hosting son como los hoteleros de la red: ofrecen espacio

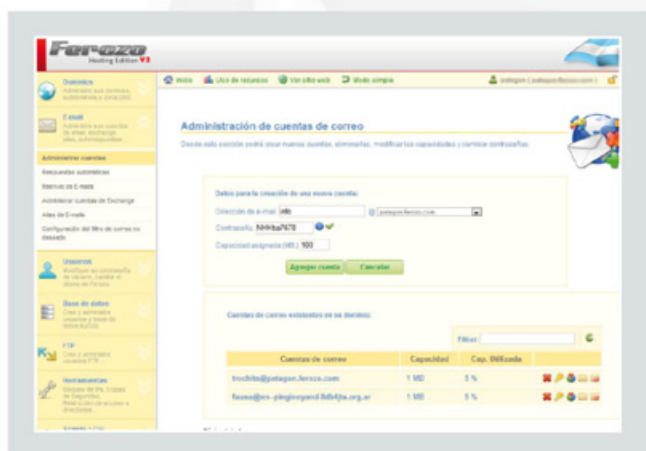


y servicios para que otras compañías almacenen cualquier información que quieran que sea accesible por una red, desde sus páginas web, hasta la información de su red interna o Intranet y sus casillas de correo.

Siguiendo con este ejemplo, los paneles de control de hosting vendrían a ser, en la web, algo así como los conserjes y el servicio de cuarto de nuestro hotel del ejemplo. En definitiva, una suerte de administradores, coordinadores y árbitros del servicio.

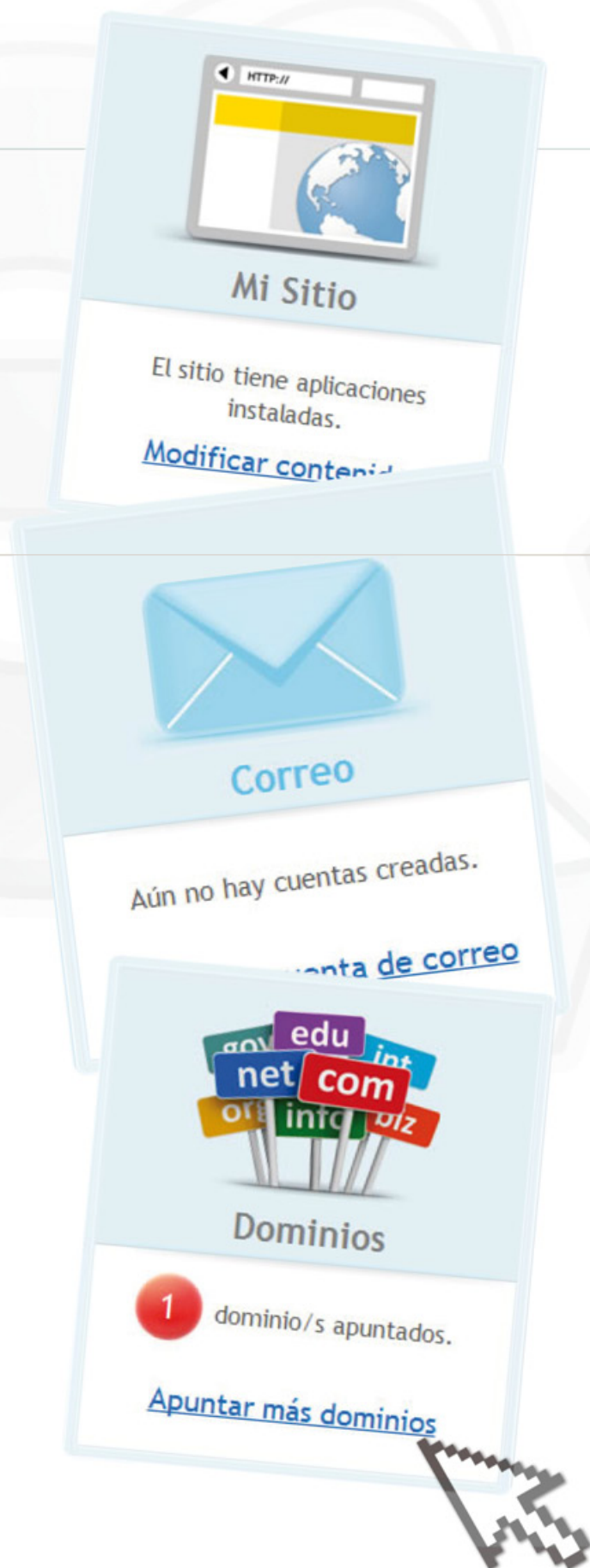
P&C: ¿Cuáles son las ventajas de operar con un panel de control?

DL: ¡Muchísimas! Hoy en día casi no se concibe administrar un servidor de hosting en forma manual. Además, eso sólo es posible en casos muy específicos.



Por lo general, los administradores de servidores de hosting prefieren dejar convenientemente en manos de los usuarios tareas como crear cuentas de correo, agregar dominios a un sitio, instalar una aplicación como Wordpress, etc. Creo que es muy clara la ventaja, por parte del usuario, de no tener que depender de un administrador de servidor para realizar tareas rutinarias como las mencionadas anteriormente. Y mucho más: por ejemplo, los revendedores de hosting pueden crear nuevas cuentas para sus clientes sin interactuar con ningún “Sysadmin”; y los usuarios de webmail pueden configurar su filtro antispam sin necesidad de depender del webmaster.

P&C: ¿Cuánto tiempo tardaron en desarrollarlo y cuál fue el capital humano involucrado en el proyecto?



Ferozo
Panel de control de Hosting

DL: Las primeras versiones de cada plataforma tardaron alrededor de un año cada una para desarrollarse e involucraron un equipo de un total de 4 personas entre las dos.

El proyecto Ferozo continúa hoy en día más vigente que nunca, con un equipo de más de 15 personas, entre arquitectos de software, desarrolladores, sysadmins y diseñadores web.

P&C: Teniendo en cuenta que con esta herramienta podemos generar y tomar el control rápidamente de sitio web, ¿cuál fue el aporte de Ferozo en el meteórico ascenso que ha tenido Dattatec en los últimos años?

DL: Puedo afirmar que el poder generar fácilmente un sitio desde una herramienta anexa al panel de control, fue un factor clave y estratégicamente

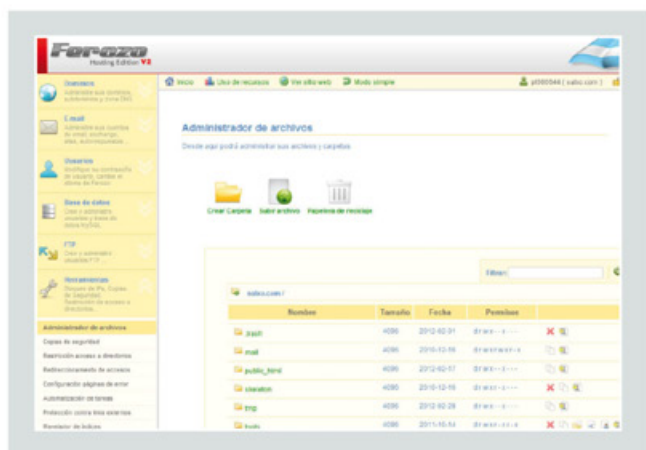
buscado desde edad temprana de Ferozo para alcanzar el éxito en materia del servicio de hosting. Por otro lado, a nivel estratégico empresarial, Ferozo significó la integración vertical y la independencia de la empresa Dattatec de proveedores externos, obteniendo de estos factores grandes ventajas competitivas.

P&C: ¿Qué devolución han obtenido de los usuarios finales?

DL: Los usuarios siempre se muestran complacidos por los avances y mejoras que Ferozo les va brindando. Recibimos de ellos una buena cantidad de feedback, casi siempre a modo de sugerencias y oportunidades de mejora. La realimentación que recibimos de parte de nuestros usuarios es siempre uno de los más valiosos motores que generan los cambios y los agregados de funcionalidades y tecnologías. La verdad, siempre hemos sentido esa sinergia con nuestros usuarios. Esa sensación de ir avanzando juntos en el camino. Siempre mejorando y haciéndonos mejorar mutuamente. Estamos muy agradecidos de haber logrado esa relación.

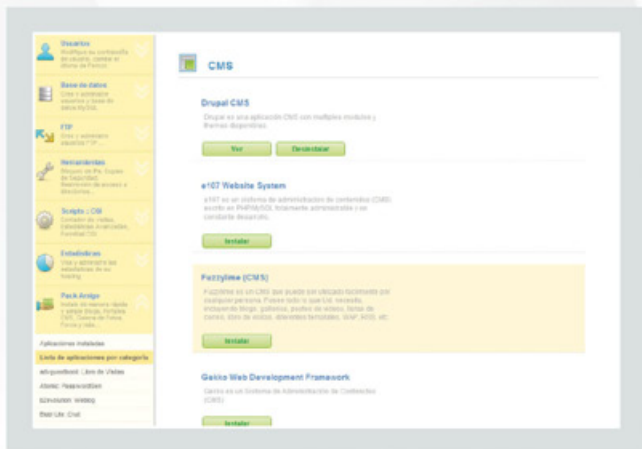
P&C: ¿Cuáles son las nuevas características en el desarrollo que incluyeron últimamente y cómo crecerá Ferozo en este año?

DL: Últimamente estuvimos trabajando más que nada en las interfaces y la usabilidad. Le agregamos un modo simple a la interfaz de Ferozo en el panel





SUBIR ARCHIVOS
DESDE MI PC



INSTALAR
APLICACIÓN



para usuarios de hosting, con las opciones mínimas y más usadas en la etapa de “set up” de un sitio, de forma que cada cuenta de hosting que es creada utiliza por defecto este modo simple que, por un lado, facilita al recién iniciado en materias de hosting la configuración inicial de su sitio, y por otro, le deja un cómodo acceso a las funcionalidades más requeridas a los más experimentados.

También hemos estrenado nuevas interfaces para el clásico “modo avanzado” de Ferozo para usuarios de hosting. Hemos re-hecho las interfaces resultando las nuevas con características totalmente Web 2.0: Ajax, recargas parciales, mejor usabilidad y funcionalidades ampliadas en general.

Con respecto a este año, podemos adelantar que pronto se viene el soporte de acceso seguro (SSL/HTTPS)

[illegible]



URL: <http://demopanel.dattatec.com:2082/>

Usuario: demo / Contraseña: demo

¿POR QUÉ FEROZO?

Mientras charlamos con Daniel, durante la entrevista no podía dejar de preguntarme el origen del nombre Ferozo. La respuesta vino de la mano de Diego Vitali (Marketing Manager de Dattatec), quien amablemente arrojó algo de luz sobre el tema.

“Durante los tiempos en que Guillermo Tornatore (CEO y fundador de Dattatec) toma la decisión de comenzar a desarrollar nuestro propio panel de hosting, él viaja con su familia unos días a Córdoba para descansar, y por supuesto, definir un nombre para el panel era una de las cosas que «viajaban en su cabeza». Por aquel entonces, su hija mayor tenía 3 años. En un momento del viaje, Guillermo manejaba por una zona de bosques frondosos al costado del camino y ante este paisaje, le comenta a su hija que en esos bosques vivía el Lobo Feroz del cuento de Caperucita Roja, con el objetivo de entretenerla y llamar su atención, a lo cual su hija le pregunta ¿Ahí vive el lobo Ferozo?... Esto quedó en la mente de Guillermo y al regresar a Rosario el tema del nombre del panel estaba resuelto. Quien en otro momento había sido motivo de la decisión de Guillermo de desarrollar Dattatec desde Argentina, puso también el nombre al panel de control”.

Ferozo
Site
 Su sitio web ya...!!!



DESARROLLO DE SOFTWARE ÁGIL CON SCRUM

Cuando hablamos de desarrollo de software muchas veces sólo nos quedamos en la superficie de lo que en realidad es el vasto, y muy profundo, mar de la ingeniería de software. Es común que hablemos de creación de líneas de código, técnicas y herramientas para mejorar nuestro rendimiento, velocidad o facilidad en la escritura de código, menos errores o una mejor calidad visual de nuestras aplicaciones, pero el desarrollo de software no se reduce sólo a esta tarea. Por detrás (o acompañando) a esta también se encuentra la gestión de los proyectos.

LA GESTIÓN


Como decíamos, el desarrollo de software no se reduce a la escritura de líneas de código, la gestión de los proyectos es tan importante como el producto, o en todo caso, la gestión, como la búsqueda y correcciones de errores, el diseño arquitectónico y muchos otros elementos hacen al todo del producto final de software. Si nos faltara alguna de estas patas simplemente estaríamos jugando a hacer software y no haciendo software realmente.

Pero en la gestión es especialmente compleja debido a que existen diferentes vertientes paradigmáticas que intentan sobresalir de otras. Debido a esto es que se han llegado a entablar peleas internas en empresas, separando a las personas en bandos. Aquellos que apoyan un modelo, otros que no están tan seguros y prefieren pivotear entre varios, y otros que prefieren los métodos ya conocidos y con más tiempo de implementación en el mundo del desarrollo.

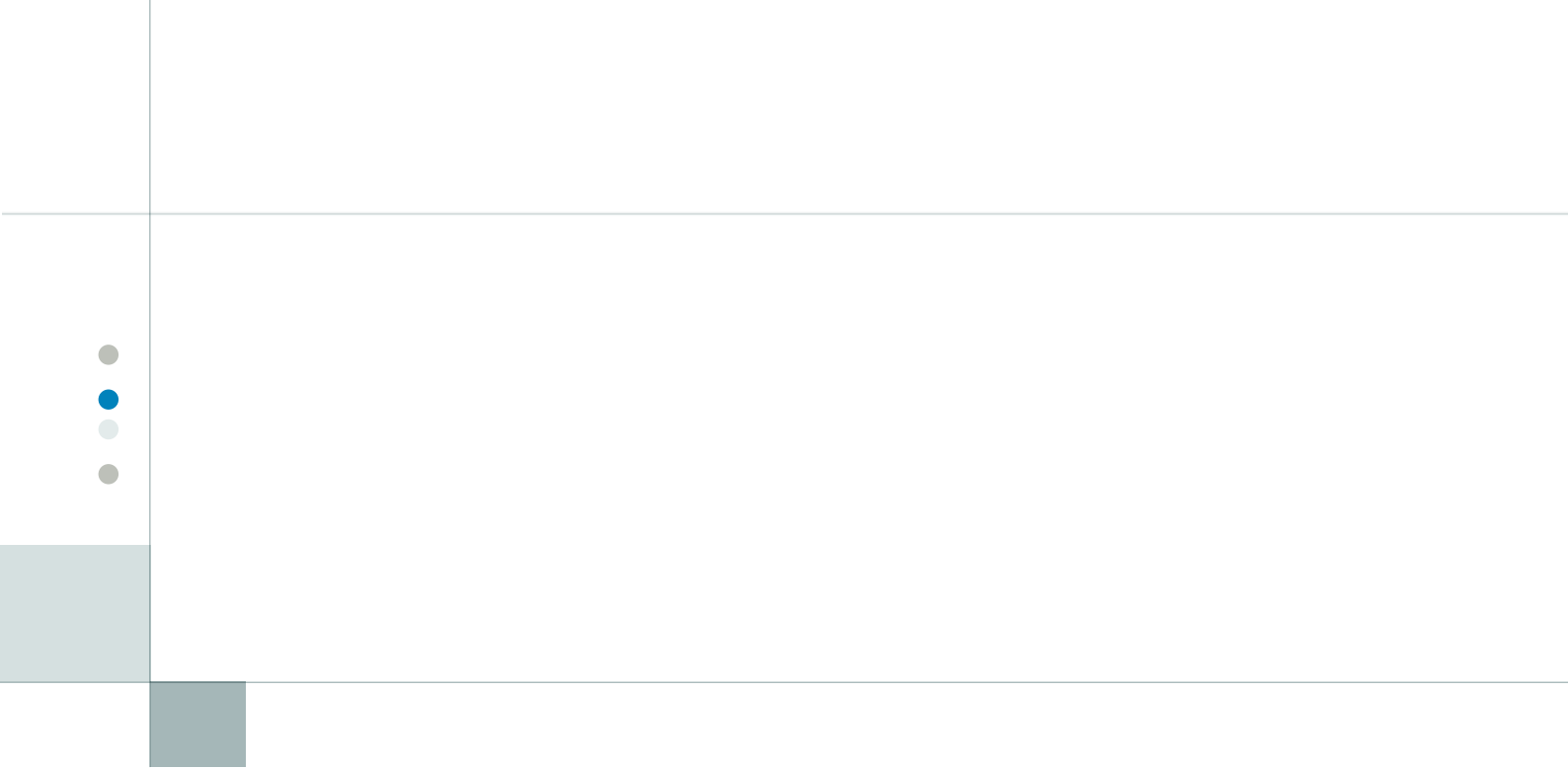
De cualquier manera podríamos colocar a la gestión de software en dos grandes áreas, aunque cada una de estas tendrá sus matices, una que llamaremos formal, y la otra ágil. Pudiendo ir de un lado a otro de estos extremos e incluso quedándonos parados en la ausencia de gestión por completo. Por supuesto, la ausencia de gestión sería un desastre seguro para nuestro proyecto.

Estos modelos de gestión tienen su historia, lo que le da inicio a los mismos y por lo tanto cada uno de ellos posee una filosofía que lo lleva a gestionar los proyectos de la forma en cómo lo hacen. Por

Matias Iacono
matias.iacono@pixelscode.com



ejemplo, los modelos de desarrollo de software más estrictos en sentido de documentación, seguimiento y control heredan su filosofía de procesos de gestión que provienen de otras industrias, poniendo, posiblemente, más el acento sobre la cosa que se produce y tratando de controlar las variables involucradas en dicho proceso más que en el valor del equipo que lo produce; es por este motivo que muchas veces hablamos de industria, ya que solemos imaginarnos las empresas que desarrollan software como aquellas que creaban automóviles en serie. De cualquier manera, y tratando de matizar esto, no debemos caer en extremismos, ya que este pensamiento extremo dependerá de la empresa que implemente este modelo y por otro lado, el tipo de proyecto que estemos creando. Esto quiere decir que muchos proyectos de carácter crítico, como software de alto rendimiento, para control de tráfico aéreo, manejo de redes y telecomunicaciones, entre otros, donde se necesita realizar la mayor cantidad de controles y verificaciones sobre el producto antes de que este llegue a ser utilizado.



En el área de la agilidad, por el contrario, el objetivo se centra principalmente en la reducción de costos innecesarios de la producción, en la entrega de productos con valor al cliente y en el valor que el equipo de producción puede otorgarle al producto que está construyendo. Por lo tanto este tipo de formas de gestión suele eliminar el papeleo excesivo al mismo tiempo que pretende interactuar más de cerca con el cliente, mostrándole avances del producto, o requiriéndole retroalimentación del producto de forma seguida.

Por supuesto, y al igual que el otro polo de gestión, aquí tendremos un abanico de posibilidades de gestión donde cada uno aplicará sus conceptos y prácticas, por lo que no debemos confundirnos y considerar que todos los modelos ágiles, así como los modelos orientados al control, aplican las mismas prácticas, aunque posiblemente sí, estén amparados por el mismo paraguas filosófico.

ESTO NO ES AGILIDAD

Es importante que destaquemos qué es y qué no es agilidad, o por lo menos, algunos de los vicios que se suelen encontrar cuando se habla de agilidad, los cuales han llevado a la mala interpretación y malas prácticas en proyectos de desarrollo de software. Como veíamos al principio, dividimos la gestión en dos grandes ramas: las orientadas al control y las

orientadas al valor entregado con el producto. Normalmente, la primera rama se distingue por la creación de documentación, fichas de seguimiento, procesos escalonados y otras actividades que, sin ser ni malas ni buenas, son parte de todo proceso que se oriente al control. Esto quiere decir que para poder saber qué pasa en cada momento dentro de un proyecto, conocer los estados de las diferentes materias, cómo pasan de uno a otro y demás, es necesario llevar controles exhaustivos, lo que deriva en el uso de herramientas y documentos constantes.

Por otro lado, cuando el desarrollador se topa con la agilidad, uno de los primeros atractivos será la sugerencia de no requerir la documentación que se proponía en los modelos anteriores. Esto es mal interpretado fácilmente y deriva en la ausencia absoluta de documentación, entre las que se incluirán la planificación de tareas y tiempos, documentos arquitectónicos, manuales de usuarios y muchos otros documentos inherentes al desarrollo de software. Pero esto es un error, un concepto de agilidad mal entendido y que es esgrimido por desarrolladores y gestores para no tener que realizar las prácticas de ingeniería que todo proyecto demanda. En este punto, la agilidad, y dependiendo del modelo que elijamos, nos sugerirá reducir la carga documental, en la medida que esto sea posible, pero no la abolición total de cualquier pieza de seguimiento en la construcción del producto. Por lo tanto, cuando hablamos de agilidad, también hablamos de estimaciones de tiempos, documentos y diagramas.



EL PROBLEMA DE SCRUM COMO MODELO ÁGIL

Los modelos ágiles son muchos, variados en su aplicación y con diferentes grados de dificultad en su entendimiento, de esta forma podemos encontrar nombres como Lean, Kanban, XP y otros, aunque Scrum posiblemente sea el que haya tenido mayor popularización y por lo tanto se haya inmiscuido en las empresas mucho más fácilmente.

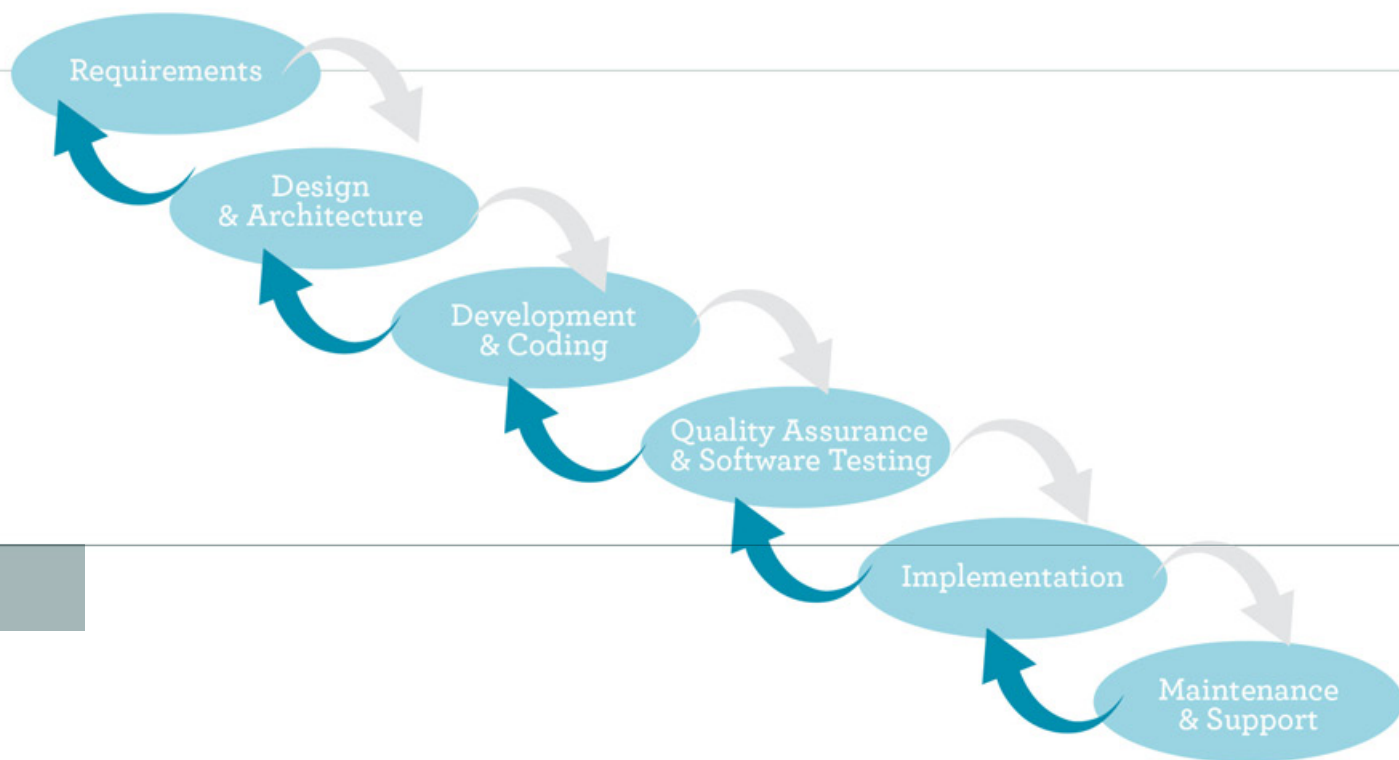
Lamentablemente, esta popularización no hizo, en muchos casos, más que dañar la credibilidad sobre esta herramienta y al mismo tiempo toda la idea que rodea a la agilidad y lo que tiene para ofrecernos en la gestión de nuestros proyectos. Este daño vino de la mano de malas aplicaciones del modelo o una falta de entendimiento del mismo y su aplicación. Posiblemente una de las grandes fallas de Scrum sea el hecho de parecer muy simple de aplicar debido a su simpleza conceptual; concepto que resulta simple e incluso elegante pero que encierra grandes complicaciones debido a la cantidad de ideas y conceptos que se deben manejar, así como paradigmas y prejuicios culturales, de gestión y personales.

Un claro ejemplo ha sido, en muchas empresas, la adopción de los nombres propuestos por Scrum para las diferentes ceremonias que éste presenta, pero la ejecución de la gestión del modo tradicional, jerarquizado, con gestiones realizadas desde las cúpulas, impidiendo que los equipos se estabilicen, entre otros factores de fracaso comunes. Por lo tanto, estas empresas terminarían cambiando

Scrum fomenta valores de profesionalismo, donde todos los integrantes del equipo deben trabajar en conjunto para conseguir un objetivo.

etiquetas, declarándose agilistas y dirigiéndose al fracaso garantizado y, con éste, generando mala fama al modelo.

De cualquier manera, aquellas empresas que pudieron entender la propuesta fueron las que equilibraron la balanza a favor de Scrum y la agilidad. No sólo aplicando los pasos propuestos sino generando un cambio interno en la estructura de la empresa, no esperando que la jerarquía de la misma tome las decisiones sino dejando a aquellos que estaban más inmiscuidos en los proyectos, que decidieran cuáles eran los pasos a seguir; dejando que los grupos de desarrollo se consolidaran, limaran sus asperezas y que cada miembro no fuera considerado un mero recurso, como una máquina más plausible de ser intercambiada en cualquier momento a otro equipo creyendo que la matemática de la gestión solucionaría el problema humano; educando al cliente para que entienda que el proyecto de software no es una mera pieza comprada en un supermercado y que para que el mismo tenga éxito él también deberá participar con sus ideas, con su visto bueno, con sugerencias y que su trabajo no se reduce a firmar cheques.



LOS PASOS DE SCRUM

Posiblemente lo más atractivo de Scrum es, como decíamos anteriormente (y esto se suele aplicar en todos los modelos ágiles), que no se dictaminan reglas metodológicas estrictas para seguir, sino que se pretende o se otorgan ciertas referencias para que al aplicar el modelo, se opte por la mejor solución. Posiblemente este es el punto más difícil de concebir, ya que al no contar con un manual lleno de lineamientos rápidamente se cae en las malas prácticas, olvidando incluso los pasos mínimos y necesarios para llevar adelante el modelo.

ROLES EN SCRUM

Normalmente, en el desarrollo de software los equipos de desarrollo suelen estar conformados por personas con diferentes especialidades, y cada especialidad actúa sobre el proyecto en determinado momento. Por ejemplo, los arquitectos podrán plantear el diseño del producto en el inicio del mismo, los desarrolladores crearán el código luego, y los encargados de realizar las pruebas sobre el producto y detectar errores al final. Esta delimitación por áreas, especialidades y procesos se puede ver más claramente en modelos de procesos de desarrollo llamados en cascada.

Una diferencia fundamental en Scrum es que no se realiza una distinción a nivel granular, limitándose a sólo tres roles: el Product Owner (o dueño del producto), el Scrum Master y el equipo propiamente dicho.



Requirements

Design
& Architecture

Development
& Coding

Proceso de desarrollo de software en cascada.
Este proceso es comúnmente utilizado en metodologías orientadas al control.

EL PRODUCT OWNER

El dueño del producto o Product Owner es el representante del producto y el conocimiento de qué es lo que dicho producto pretende hacer, por lo tanto, este rol está encargado de velar por el valor que ese producto generará, sabiendo qué partes del mismo pueden o no producir mayores beneficios si éste viera la luz.

Normalmente, se asocia a esta figura con el cliente en sí mismo, pero es posible contar con la ayuda de otra persona que actúe en nombre de él y se responsabilice de las decisiones tomadas a nivel de producto.

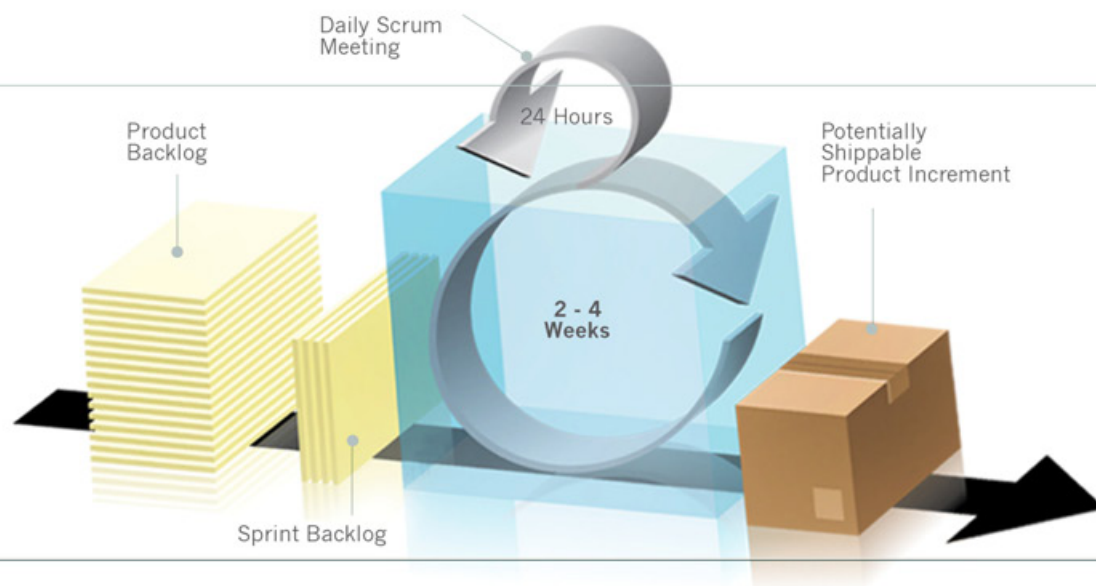
El Product Owner podrá, por ejemplo, elegir funcionalidades a quitar, o nuevas a añadir dentro de la lista de funcionalidades para ese producto. Este tipo de decisiones pueden ser tomadas en base a un mercado variable, donde lo pensado para un producto ya no sea válido con el correr del tiempo

EL SCRUM MASTER

El Scrum Master es una pieza fundamental dentro de Scrum. Su principal labor es la de convertirse en un facilitador para el equipo, intentando remover obstáculos del camino que impidan el normal avance del desarrollo del producto.

Comúnmente suelen surgir en el proceso de desarrollo problemas de comunicación entre el equipo y el Product Owner, o la estructura empresarial y el equipo, por lo que el Scrum Master podrá interceder para que el proceso productivo siga un flujo normal.

Por el contrario, el Scrum Master no es un líder de proyectos o gestor del mismo, rol que suele ser confundido fácilmente cuando este modelo es mal aplicado, por lo tanto, el Scrum Master no tiene permitido decidir sobre cómo realizar el producto, qué miembro del equipo hará determinada funcionalidad, realización de estimaciones, entre otras funciones.



EL EQUIPO

Por último y uno de los más importantes es el equipo de desarrollo. Decimos que es uno de los más importantes debido a que es éste el que se encargará de producir el producto, y sólo éste sabe cuál será la mejor forma de hacerlo así como cuánto tiempo tomará. Esto quiere decir, por un lado, que un equipo Scrum deberá estar formado por todas las personas que sean necesarias para poder crear este producto, las cuales deberán trabajar conjuntamente, tomando decisiones, eligiendo arquitecturas, probando herramientas, analizando el código y un largo etcétera.

Aquí es donde encontramos una diferencia fundamental y funcional sobre otras formas de gestión, ya que tanto arquitectos, desarrolladores, testers, diseñadores y demás, no esperan su turno para actuar y poner en práctica sus conocimientos sobre el producto, sino que trabajan de forma sinérgica entre ellos. Así, mientras el desarrollador escribe las líneas de código para una funcionalidad, el tester escribe casos de pruebas para la misma, el arquitecto plantea mejores formas de manejar los flujos de información y el gestor de bases de datos optimiza consultas. De esta forma todos tienen noción del proyecto como una unidad y la responsabilidad de su correcta finalización recae sobre todos.


LOS PASOS DE SCRUM

Por último, en Scrum contamos con algunos mecanismos que nos garantizarán la obtención de cierto ritmo en el proceso productivo. Este ritmo no se reduce sólo a la creación de código sino también a las estimaciones de tiempo, las interacciones con el equipo y con el cliente.

Todo proyecto de Scrum nace con una lista de tareas a desarrollar que representan el producto en su totalidad. Esta lista es comúnmente llamada Product Backlog. Además de tener la lista de tareas, esta debe estar priorizada en base a las necesidades del cliente, por lo que esta tarea de priorización suele ser realizada por el Product Owner.

Con nuestro Product Backlog completo, esto es, priorizado y estimado, iniciamos una ventana de desarrollo que suele durar entre dos a cuatro semanas – el tiempo que elijamos será una constante, por lo que es importante elegir correctamente. Esta ventana de tiempo recibe el nombre de Sprint y es un momento donde el equipo, seleccionando una cantidad de ítems del Product Backlog se pondrá en la tarea de crearlos. Por lo tanto, es vital que durante este tiempo el equipo no sea perturbado ya que su tarea es la de realizar las piezas elegidas para esta iteración.

Durante el tiempo que dure el Sprint, el equipo tendrá una reunión diaria, sólo entre ellos y el Scrum



El proceso de Scrum resulta simple, pero encierra en sí mismo una demanda de profesionalismo y prácticas de ingeniería que no muchos están dispuestos a llevar adelante.

Master si fuese necesario, donde se abordarán, por cada integrante, tres temas. Cada uno de los integrantes deberá contestar a las preguntas: - ¿Qué hice ayer? ¿Qué haré hoy? ¿Existe algún impedimento que no me permita trabajar?

Con estas preguntas en la mesa, todo el equipo puede saber cuál es el estado de cada uno de los integrantes del mismo y así poder solucionar problemas o colaborar en determinadas situaciones. Debemos recordar que la responsabilidad del producto recae sobre todos sus miembros, por lo que el mantener una comunicación fluida y clara es de vital importancia.

El último paso se lleva a cabo cuando el Sprint finaliza, esto es, al finalizar las dos o cuatro semanas de producción. En este momento es cuando se requiere mayor soporte del Product Owner, ya que el objetivo será mostrarle lo que se ha producido en base a los ítems por los cuales el equipo se comprometió al inicio del Sprint. En este punto, el Product Owner podrá validar o no cada uno de los elementos presentados, pero más importante será que en este momento se podrá saber si lo producido coincide con las ideas originales o con las demandas del mercado. Si tenemos en cuenta que el mercado fluctúa y las necesidades del mismo cambian, puede ser que en este punto el producto deba tomar otro rumbo. Estas iteraciones cortas y contactos con el cliente de forma seguida,

entonces, nos darán una oportunidad para poder hacer un viraje de timón sobre el proyecto y de esta forma alcanzar un producto que efectivamente genere algún valor para el cliente.

RESUMEN

Lo que hemos visto aquí es sólo la punta de Scrum como forma ágil de gestión. Como podemos notar, queda mucho recorrido, tanto a nivel de la aplicación del modelo, como de la interacción entre el cliente y el proveedor del servicio. Y es aquí donde se ven las fallas en la aplicación del modelo en la vida real, ya que muchas veces los que lo llevan a la práctica suelen quedarse con la historia corta sin ahondar en los detalles más importantes. De cualquier manera, las formas ágiles de gestión ya han podido torcer la media de desconfianza y se han probado como modelos válidos y útiles, por lo que se ha dejado de lado la experimentación para formar parte de una realidad en las empresas de desarrollo de software.

En siguientes artículos profundizaremos más en esta práctica, detallando cada uno de los artefactos, así como las responsabilidades y prácticas que deberá cumplir cada miembro del proyecto. Finalmente, aprenderemos sobre las herramientas de gestión que nos recomienda Scrum y cómo podemos saber del estado y evolución del proyecto.



TESTING DE APLICACIONES

No importa la cantidad de tiempo que se invierte en el diseño de una aplicación Android, o incluso qué tan cuidadosos somos en la programación, los errores son inevitables y los errores aparecerán. En esta nota, queremos ayudar a todos los lectores de Pixels&Code a conseguir rápidamente tener algunos test de unidad, y luego presentaremos un framework para testear de forma automática la interfaz de usuario, llamado Robotium.

COMENZANDO

Al momento de crear un proyecto Android podemos crear el Proyecto Android que realice el Testing. Pero en este caso, vamos a trabajar sobre un Proyecto Android ya creado anteriormente, y le vamos a crear su Proyecto de Testing asociado. Usaremos el «*Hola Mundo Android*», aquel que realizamos la primera vez que instalamos el Kit de Desarrollo de Android. Para quienes quieran tener ya un proyecto «*Hola Mundo Android*» andando de forma súper rápida (sin usar el wizard de Eclipse), en este link pueden descargar un «*Hello Pixel&Code*»: **<http://pixelscode.com/downloads/HelloPixelAndCode.zip>**



CREANDO EL PROYECTO DE TESTING

El testing de una aplicación Android es también una aplicación Android. El wizard de Eclipse para un «**Nuevo Proyecto Android de Test**» nos crea un proyecto de testing y el framework necesario, al mismo tiempo. Veamos cómo hacer esto:

1- En Eclipse, vamos a: **New > Project > Android > Android Test Project**

2- En la ventana del Wizard, le ingresamos los siguientes valores:

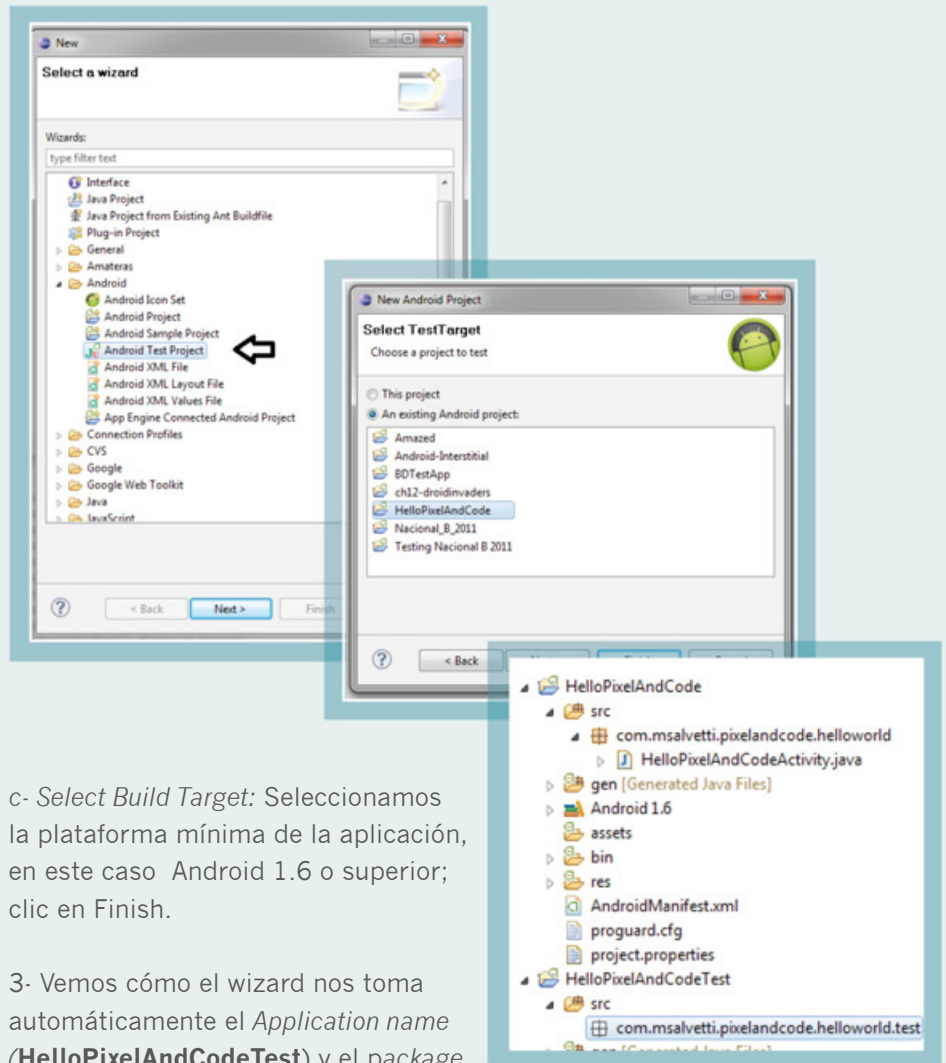
a- *Project*

Name: “**HelloPixelAndCodeTest**”, clic en Next.

b- *Select TestTarget*: seleccionamos “An existing Android project”, y luego seleccionamos “HelloPixelAndCode” de la lista de proyectos en nuestro Workspace; clic en Next.

c- *Select Build Target*: Seleccionamos la plataforma mínima de la aplicación, en este caso Android 1.6 o superior; clic en Finish.

3- Vemos cómo el wizard nos toma automáticamente el *Application name* (**HelloPixelAndCodeTest**) y el *package name*, todo reflejado en el nuevo proyecto que aparece en nuestro *Package Explorer*.



CREANDO LA CLASE CON EL CASO DE TEST

Ahora ya tenemos todos los archivos y directorios que necesitamos para crear y ejecutar una aplicación de testing, a excepción de la clase que contiene los tests (la clase de casos de prueba). Vamos a crear dicha clase Java que contenga el código que testea la Actividad y que extienda una de las clases de Test que trae Android, especialmente diseñada para Actividades. Nuestra clase va a contener la definición de cuatro métodos, de los cuales luego explicaremos su función. Con esto en mente, creamos la clase:

1- En Eclipse, realizamos «clic derecho» sobre el paquete «*com.msavetti.pixelandcode.helloworld.test*» del proyecto «**HelloPixelAndCodeTest**»: **New > Class**

2- En la ventana del Wizard, le ingresamos los siguientes valores:
a- *Name*: “**MiPrimerTest**”.
b- *Superclass*: “android.test.ActivityInstrumentationTestCase2 <HelloPixelAndCodeActivity>” la superclase está parametrizada por el

nombre de la Actividad a testear.
c- Sin cambiar ningún otro valor, damos clic en *Finish*.

3- Ahora, ya tenemos el archivo **MiPrimerTest.java** en el proyecto, que es una clase que extiende a la clase para Testear Actividades, llamada *ActivityInstrumentationTestCase2<T>*

4- Abrimos el archivo **MiPrimerTest.java** y veremos este código:

```
package com.msavetti.pixelandcode.helloworld.test;
```

```
import android.test.ActivityInstrumentationTestCase2;
```

```
public class MiPrimerTest extends ActivityInstrumentationTestCase2<HelloPixelAndCodeActivity> {  
}
```

ActivityInstrumentationTestCase2 nos permite realizar testing funcional para una sola Activity. También se puede obtener la Activity inyectada que se está probando, llamando al método *getActivity()*. Y como la actividad se pone en marcha antes de cada test,

nos permite ver cómo la interfaz de usuario está siendo probada.

5- Agregamos el constructor y el método `setUp()` a la clase, y luego tres variables de clase necesarias, que se muestran a continuación:

```
public MiPrimerTest() {
    super("com.msavetti.pixelandcode.helloworld",HelloPixelAndCodeActivity.class);
}
protected void setUp() throws Exception {
    super.setUp();
    actividadTestear = this.getActivity();
    mView = (TextView) actividadTestear.findViewById(com.msavetti.pixelandcode.helloworld.R.id.textTitulo);
    resourceString = actividadTestear.getString(com.msavetti.pixelandcode.helloworld.R.string.hello);
}
private HelloPixelAndCodeActivity actividadTestear;
private TextView mView;
private String resourceString;
```

6 - El método **setUp()** se ejecuta

antes de llamar a cada método de test, es por eso que inicializamos las variables y preparamos el entorno de prueba. Para este caso, inicializamos la aplicación Android a testear, recuperamos el texto que se muestra en pantalla y el texto que está en los archivos de recursos.

7 - En estos momentos, es necesario que importemos lo que nos pide Eclipse (el *TextView* y la Actividad *HelloPixelAndCodeActivity*).

8- Ahora, agregamos un método que testea las precondiciones:

```
public void testPrecondiciones() { //
    se ejecuta una sola vez este metodo.
    assertNotNull(mView);
}
```

Aunque una prueba de precondiciones puede comprobar diferentes condiciones, en esta aplicación sólo necesitamos comprobar si la aplicación sometida a prueba se inicializa adecuadamente y si existe el *TextView*. Para esto, se llama al método heredado *assertNotNull()*, pasando una referencia al *TextView*. La prueba tiene éxito sólo si la

referencia al objeto no es nula, es decir, si se inicializa correctamente.

9- Ahora, el método que valida el texto:

```
public void testText() {
    assertEquals(resourceString,(String) mView.getText());
}
```

Este método **testText()** llamará a un método *Assert* de JUnit para comprobar si el *TextView* se muestra con el texto esperado.

Para este ejemplo, la prueba espera que el *TextView* esté mostrando el recurso (la *String*) que se declaró originalmente en la aplicación a testear. La llamada a *assertEquals(String, String)* compara el valor esperado, el recurso *String*, con el texto mostrado por el *TextView*, obtenido a partir del *TextView*. El test tiene éxito sólo si los textos coinciden.

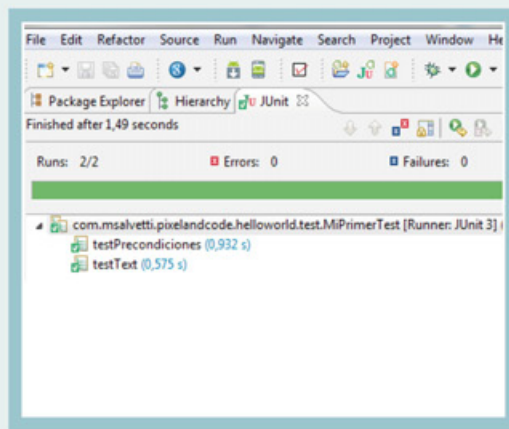
10- En este link pueden descargar el código fuente de este proyecto para testear: http://pixelscode.com/downloads/HelloPixelAndCodeTest_simple.zip

EJECUTANDO EL CASO DE TEST

En el *Package Explorer* hacemos clic en el proyecto de testing **HelloPixelAndCodeTest** y seleccionamos **Run As> Android JUnit Test**, para que a continuación Eclipse inicie la aplicación de testing y la aplicación sometida a prueba. Luego, el framework de pruebas ejecuta los tests y reporta los resultados a la vista JUnit de Eclipse, que muestra los resultados de las pruebas en un panel superior, con un resumen de los tests que se ejecutaron, y en un panel inferior presenta los rastros de los tests que fallaron. En este caso, los tests se han ejecutado correctamente, por lo que no hay ninguna falla reportada en la vista inferior. Ya estamos familiarizados con el proyecto de testing, continuemos con la automatización de pruebas.

ROBOTIUM, TEST AUTOMÁTICO DE INTERFAZ

Robotium es un framework creado para implementar test automáticos de aplicaciones Android. Con esta herramienta se puede testear cualquier tipo de aplicación visual como Activity o Dialogs. Comparándolo con el mundo del desarrollo web, se podría decir que Robotium es el Selenium de Android. La dinámica de trabajo con este framework es la programación de acciones que simulen al usuario. Por ejemplo, se puede realizar la acción de realizar *clic* en un texto específico, ingresar valores en una caja de texto concreta o buscar un texto en una pantalla. Realizando varias de estas acciones se puede comprobar si existe un texto, mediante un Assert por ejemplo, y de esta forma validar si la aplicación funciona bien y pasa las pruebas.



Nuestra aplicación a probar será ampliada; modificaremos la única Activity para que sea una simple calculadora. Ingreso dos valores, y al hacer clic en «*Multiplicar*» se mostrará el resultado de multiplicarlos entre sí. Por simplicidad se deja este ejercicio para el lector, y queda disponible el proyecto en el siguiente link:

http://pixelscode.com/downloads/HelloPixelAndCode_Calculadora.zip.

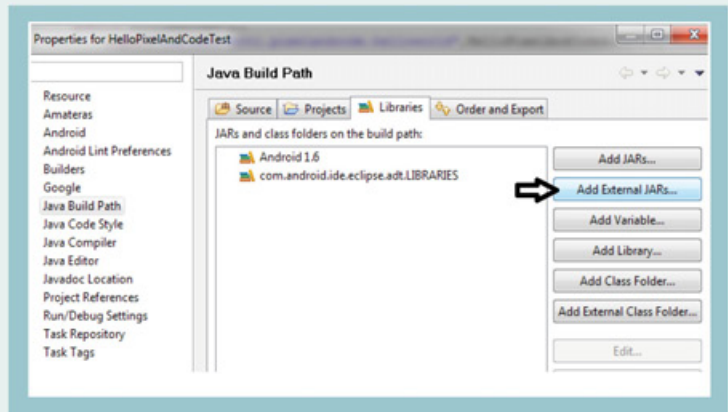
Así que ahora, vamos a probar esta mini-calculadora.

Para usar Robotium en un proyecto de testing, realizamos estos pasos:

- Descargar el archivo **robotium-solo-3-1.jar** que contiene el framework desde: <http://code.google.com/p/robotium/>

Será necesario que tengamos un proyecto Android con la aplicación que deseamos testear, y como en nuestro caso, ya creado un proyecto de **Test Android** (si no, crearlo como vimos anteriormente: **File->New->Project->Android->Android Test Project**).

- Necesitamos una referencia a Robotium desde nuestro proyecto. Para eso, vamos a agregar el archivo jar: clic derecho en el proyecto-



>**Build path->Configure Build Path** y en *Properties*, clic en la pestaña *Libraries*, para agregar el jar de Robotium, como se muestra en la siguiente imagen:

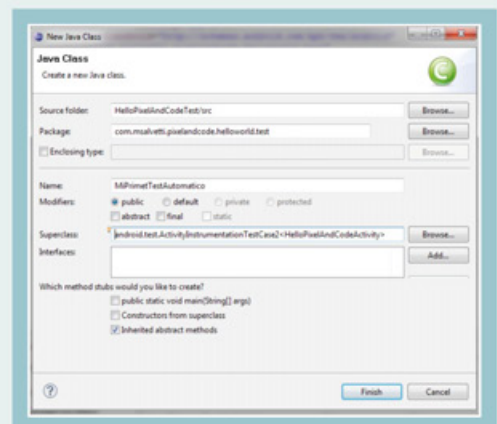
- Dentro de nuestro proyecto de test, creamos una nueva clase con un test de ejemplo: **New->Class**, derivamos esta clase de **ActivityInstrumentationTestCase2**, como se muestra en la captura de pantalla:

- Abrimos el archivo **MiPrimerTestAutomatico.java** y veremos este código:

```
package com.msalveti.  
pixelscode.helloworld.test;
```

```
import android.test.  
ActivityInstrumentationTestCase2;
```

```
public class MiPrimerTestAutomatico  
extends ActivityInstrumentationTestC
```



```
ase2<HelloPixelAndCodeActivity> {  
}
```

- Los métodos **tearDown()** y el **constructor** los creamos de la misma forma que en el test con JUnit.

- Tenemos que instanciar un objeto de Robotium en el método **setUp()**, esto lo realizaremos a través de la clase “Solo” del Framework. Para eso agregamos la siguiente variable de clase “Solo” a nivel de clase y modificamos el método **setUp()** así:

```
private Solo solo; //la instancia para  
trabajar con el Robotium
```

```
protected void setUp() throws  
Exception {  
    super.setUp();  
    solo = new Solo(getInstrumentation(),  
        getActivity());  
}
```

La clase “Solo” se utiliza para hacer las pruebas de instrumentación más fácil. Es compatible con los casos de prueba que se extienden sobre múltiples actividades. Todo es manejado automáticamente por Robotium-Solo, por eso tenemos que tener una instancia por test, e inicializarla al comenzar las pruebas, en el método **setUp()**.

Recordemos que, con este framework, los casos de prueba están escritos desde la perspectiva del usuario. En

nuestro caso de test, vamos a probar el contenido del formulario y su funcionalidad, realizando los siguientes cuatro pasos:

1. Manejar los controles de entrada (EditTexts)
2. Introducir los valores de nuestra propia elección
3. El acceso y clic al botón *Multiplicar*
4. Poner valores para verificar el resultado de la multiplicación, en el campo de resultado.

- ¿Y cómo usamos la variable privada “Solo”, dentro de nuestro Test Case? Fácil, acá tenemos unos ejemplos de sentencias de test, con Robotium:

```
solo.sendKey(solo.MENU);
```

```
solo.clickOnText(“Nuevo”);
```

```
Assert.assertTrue(solo.  
    searchText(“Crear nuevo archivo”));
```

- Ahora, en nuestro test, el primer método que ingresa valores en las cajas de texto, las limpia, reingresa valores y luego realiza clic en el botón:

```
public void testIngresoTexto() {  
    solo.enterText(0, “10”);  
    solo.enterText(1, “20”);  
    solo.clickOnButton(“Multiplicar”);  
    assertTrue(solo.searchText(“200”)); }  
}
```

• Ahora, vemos un método de prueba más completo:

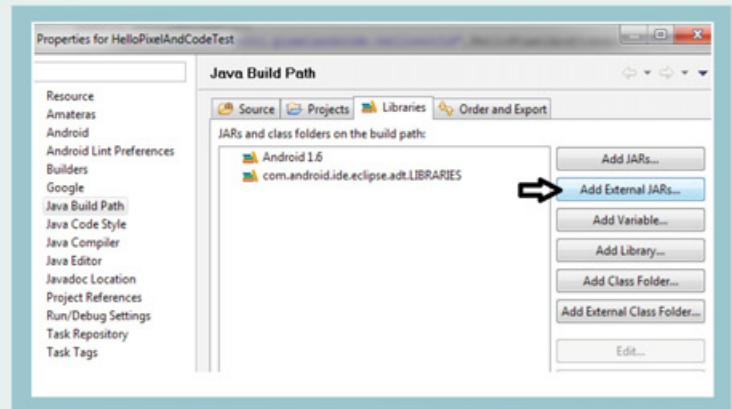
```
public void testMasCompleto() {  
    float primerNumero = 10;  
    float segundoNumero = 20;  
    float resultado = primerNumero *  
    segundoNumero;  
    EditText FirsteditText = (EditText)  
    solo.getView(R.id.EditText01);  
    solo.enterText(FirsteditText, String.  
    valueOf(primerNumero));  
    EditText SecondeditText = (EditText)  
    solo.getView(R.id.EditText02);  
    solo.enterText(SecondeditText,  
    String.valueOf(segundoNumero));  
    solo.clickOnButton(«Multiplicar»);  
    assertTrue(solo.searchText(String.  
    valueOf(resultado)));  
    TextView outputField = (TextView)  
    solo.getView(R.id.TextView01);  
    ArrayList currentTextViews = solo.ge  
    tCurrentTextViews(outputField);  
    assertFalse(currentTextViews.  
    isEmpty());  
    TextView salida = (TextView)  
    currentTextViews.get(0);  
    assertEquals(String.  
    valueOf(resultado), salida.getText().  
    toString());  
}
```

Ejecutamos con clic derecho en el proyecto, **Run As>Android JUnit Test**. Si las cosas salen bien, el Emulador carga la aplicación, y entrará automáticamente los valores

de los EditText, y realizará clic en el botón Multiplicar. Al terminar la ejecución, se mostrará la barra verde que muestra la ejecución exitosa.

En este link pueden descargar el código fuente de este proyecto para testear automáticamente, utilizando JUnit y Robotium: http://pixelscode.com/downloads/HelloPixelAndCodeTest_completo.zip

El principal problema que nos podemos encontrar es con la modificación de la interface de la aplicación, algo habitual cuando se está desarrollando. Si ocurriese alguna modificación en algún texto, los test afectados se deberían cambiar modificándolos a los nuevos textos.



CONCLUSIONES

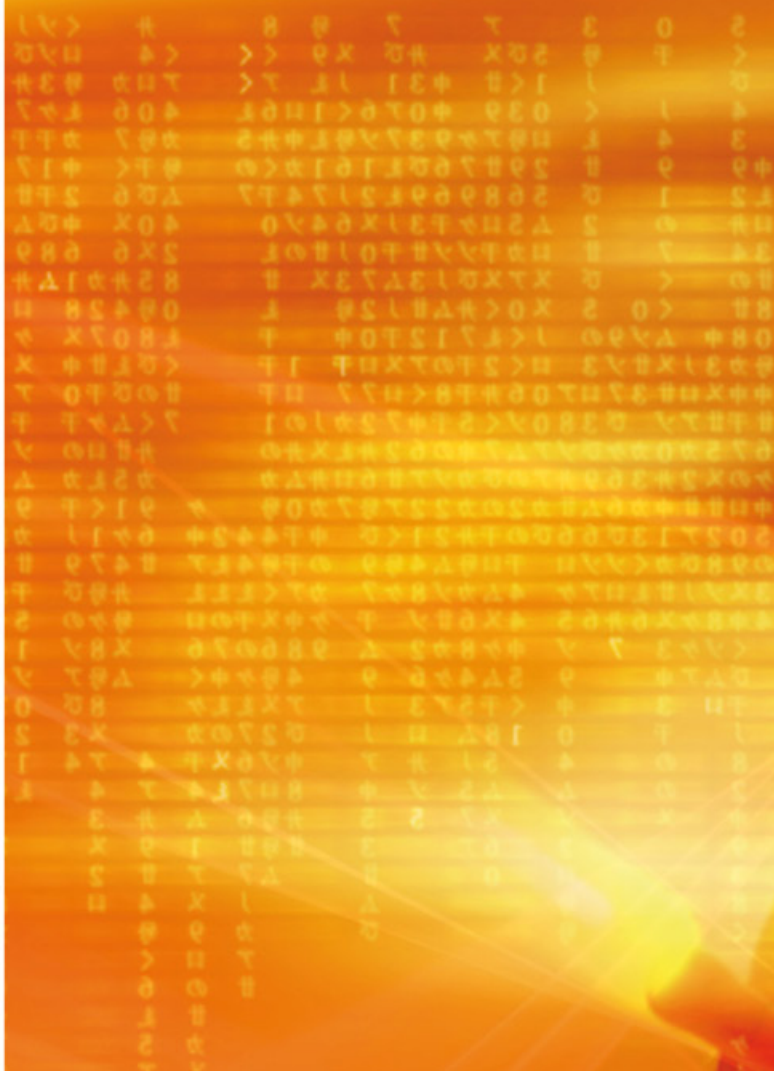
Android nos ofrece un framework muy poderoso y fácil de usar que está bien integrado con las herramientas de desarrollo que nos descargamos cuando comenzamos a desarrollar.

Cómo realizar testing es una parte importante de cualquier desarrollo; en esta nota realizamos una introducción al testing con JUnit y les ayudamos con el puntapié inicial, de forma rápida, para luego presentar una interesante forma de automatización de casos de test, mediante JUnit y Robotium.

Con este framework podemos automatizar nuestros tests y ejecutarlos diariamente a medida que vamos desarrollando.

LAS LEYES DE PROGRAMACIÓN DE TAYLOR (PARTE II)

Este mes continuamos por la senda iniciada en la edición anterior, tomando las leyes de Taylor como faro. El Sistema Operativo UNIX y sus herederos ofrecen a sus usuarios una importante cantidad de herramientas, que en mano de un programador que tenga un conocimiento mínimo de cómo trabajar con ellas se vuelven extremadamente poderosas, permitiendo hacer más en menos tiempo. En esta ocasión nos vamos a centrar en un programita muy interesante: “sed” (stream editor). Sed implementa un mini-lenguaje de programación, que permite automatizar transformaciones a un archivo de texto, de forma similar a como ya vimos con “tr”. Este último está pensado para realizar sustituciones sencillas de un caracter por otro, mediante una serie de reglas que se le establecen. En cambio, sed puede realizar acciones mucho más complejas, gracias a su soporte de expresiones regulares. Veamos cómo hacerlo.



UNIX

Where there is a shell, there is a way.





TAYLOR, SIEMPRE PRESENTE

Pero primero, recordemos el enunciado del misterioso Taylor, que como ya mencionamos en la primera parte, deben leerse de abajo hacia arriba, es decir, en orden de complejidad decreciente.

“Nunca programe en ‘C’ si puede hacerlo en ‘awk’;
Nunca lo haga en ‘awk’ si ‘sed’ puede manejarlo;
Nunca use ‘sed’ cuando ‘tr’ puede hacer el trabajo;
Nunca invoque ‘tr’ cuando ‘cat’ sea suficiente;
Evite usar ‘cat’ siempre que sea posible.”

En nuestro “episodio anterior” aprendimos las distintas posibilidades que nos ofrecen cat y tr, además de cubrir otras herramientas que permiten automatizar el tratamiento de textos, como cut. Siguiendo este orden, entonces, le toca el turno a sed. Escrito originalmente por Lee McMahon durante la década del 70, cuando se desempeñaba en los Bell Labs, cuna del UNIX original, ha sido portado a una infinidad de plataformas, aunque sigue siendo un componente central de los clones de UNIX, como GNU/Linux y *BSD. Inclusive, se dice que el gran Larry Wall, experto en sed, se inspiró en esta herramienta y en awk -que veremos en nuestra próxima entrega- al crear Perl, un lenguaje de scripting extremadamente poderoso y muy popular en nuestros tiempos, particularmente entre los sysadmins.

USOS SIMPLES DE SED

La mayoría de los casos en los que suele utilizarse sed son sencillos, y por lo tanto pueden entenderse a simple vista. No obstante, este minilenguaje es lo suficientemente complejo y poderoso como para que un script más largo y complicado sea lo suficientemente críptico para que sólo un usuario avanzado del mismo pueda interpretarlo rápidamente. Para colmo, la documentación más a mano (las páginas de man) no es de lo más accesible y completa. Por lo general, en el trabajo con sed, como dijo un gran poeta, “se hace camino al andar”.

El uso más común de sed es equivalente a la función “Buscar y Reemplazar” que ofrecen muchos editores de texto. La diferencia es que con ellos hay que hacerlo en forma interactiva, mientras que sed (como casi todo este tipo de herramientas de consola en UNIX) está pensado para operar en forma automatizada. Para hacer un “search & replace” de sed, entonces, escribimos:

```
sed s/cadena_a_reemplazar/nueva_cadena/  
archivo_original.txt > archivo_nuevo.txt
```

La “/” (barra, o en inglés “slash”), que se emplea como delimitador, es solamente una convención, ya que sed toma al carácter que sigue inmediatamente a la “s” (comando que ordena el reemplazo) como tal. Si se pretende reemplazar una cadena que contenga la



barra (como una ruta de archivos en UNIX), es buena idea optar por otro separador, aunque muchos optan por usar el caracter de escape “\”. Otras elecciones comunes como separador son el underscore “_”, el pipe “|”, o los dos puntos “:”.

La expresión del ejemplo anterior, tal como fue escrita, sólo reemplaza la primera ocurrencia de la cadena a reemplazar especificada. Si se quiere sustituir todas las ocurrencias, debe indicarse que la orden “s” es global, agregando luego de la cadena nueva un delimitador y la letra “g”, de esta manera:

```
sed 's/cadena_a_reemplazar/nueva_cadena/g'  
archivo_original.txt > archivo_nuevo.txt
```

Aquí también encerramos la expresión entre comillas simples, algo imprescindible si es que la cadena a reemplazar o la nueva contienen caracteres especiales, como el espacio en blanco. Para hacer una prueba sin necesidad de contar con un archivo de texto, podemos hacerlo con el comando “echo” y un pipe, tal como vimos la vez pasada:

```
echo «Hola Juan» | sed 's/Hola/Buen día/'
```

Da como resultado:

Buen día Juan

Entre los diversos caracteres especiales de las expresiones, como aquellos empleados para especificar las expresiones regulares (paréntesis, corchetes, asterisco, punto, etc.) se destaca el ampersand “&”, que representa la cadena encontrada en base a la expresión de búsqueda. Esto puede ser muy útil. Es fácil reemplazar “hola” por “_hola_” cuando conocemos exactamente la palabra que estamos buscando, pero ¿qué ocurre cuando desconocemos la totalidad de la cadena a procesar? Veamos:

```
echo «agrego underscores» | sed «s/[a-z]*/_&_/g  
_agreg_ _underscores_
```

Si alguno de los caracteres especiales que se emplean para especificar la expresión regular de búsqueda se encuentra dentro de la cadena a matchear, debemos “escaparlo” con una contrabarra “\” (backslash), como se hace en muchos lenguajes emparentados con UNIX (como los scripts de shell) para que sed comprenda que nos estamos refiriendo literalmente a ese caracter, y no lo interprete como un caracter especial:

```
echo «agrego [underscores]» | sed «s/\[[a-z]  
]*\]/_&_/»  
agreg_ _[underscores]_
```

Noten el uso de la contrabarra para escapar los corchetes, provocando que de las dos palabras que componen el string a procesar solamente sea



modificada aquella que está encerrada entre los caracteres “[” y “]”.

Ahora, si tenemos varias palabras entre corchetes, podemos recurrir al modificador global “g” para sustituirlas a todas. Pero si quisiéramos omitir el reemplazo de la primera ocurrencia, podemos indicarle a sed que lo haga sólo de la segunda ocurrencia en adelante, de esta forma:

```
echo ‘[agrego] [underscores] [parcial]’ | sed ‘s/\
[[a-z]*\]/_&_/2g’
[agrego] _[underscores]_ _[parcial]_
```

Podemos “encadenar” más de un reemplazo de dos maneras. La primera es la obvia para todos los usuarios de consola:

```
echo “Hola Juan. Chau.” | sed “s/Hola/Buen día”/
| sed “s/Chau/Hasta Luego/”
Buen día Juan. Hasta Luego.
```

En lugar de usar un pipe, podemos pasarle a sed múltiples patrones, aclarándole delante de cada uno que se trata de un juego de órdenes a través del switch -e:

```
echo “Hola Juan. Chau.” | sed -e “s/Hola/Buen
día”/ -e “s/Chau/Hasta Luego/”
Buen día Juan. Hasta Luego.
```

Por supuesto, si las órdenes a encadenar son muchas, esto ya es poco práctico. Hay una forma mejor de resolverlo.

SED SCRIPTS

Podemos escribir con el editor de nuestra preferencia un “sed script” y luego indicarle a sed con el switch -f que lea los comandos directamente de ese archivo. Veamos un ejemplo. Editamos un archivo que llamamos “descastellaniza.sed”, e ingresamos lo siguiente:

Este script elimina los acentos y la ñe

```
s/á/a/g
s/é/e/g
s/í/i/g
s/ó/o/g
s/ú/u/g
s/ñ/n/g
s/Á/A/g
s/É/E/g
s/Í/I/g
s/Ó/O/g
s/Ú/U/g
```

Cada comando sed debe escribirse en una línea separada. Si deseamos ingresar comentarios, los prefijamos con el caracter numeral “#”. Ahora editamos un archivo que bautizamos “prueba.txt” con el siguiente contenido:

“En la ortografía del idioma español, se denomina acento diacrítico a la tilde que se emplea para distinguir significados en pares de palabras, habitualmente monosílabas, de las cuales una es regularmente tónica, mientras que la otra átona en el habla, marcando el acento de la tónica.”



Lo probamos de esta forma:

```
sed -f descastellaniza.sed prueba.txt
```

“En la ortografía del idioma español, se denomina acento diacrítico a la tilde que se emplea para distinguir significados en pares de palabras, habitualmente monosílabas, de las cuales una es regularmente tónica, mientras que la otra atona en el habla, marcando el acento de la tónica.”

TRABAJANDO CON RANGOS

Una posibilidad útil es decirle a sed sobre qué líneas tiene que aplicar las órdenes recibidas. Esto puede hacerse escribiendo directamente el número de línea a procesar, en el caso de que sea una sola, o especificando un rango desde-hasta, en la forma “N,M”. Probemos esta funcionalidad en conjunción con el switch -n, que le dice a sed que por defecto no envíe a standard output los patrones “matcheados”, y de paso aprendemos un nuevo comando (parece mentira, pero hasta ahora usamos solamente el comando de búsqueda “s”, o “substitute”). La instrucción “p” de print, se usa en conjunto con el switch -n para indicar que el rango especificado sí debe imprimirse.

```
ls -l / | sed -n “1,5 p”
```

```
total 108
```

```
drwxr-xr-x 2 root root 4096 2012-03-15 18:04 bin
```

```
drwxr-xr-x 3 root root 4096 2011-09-30 02:31 boot
drwxr-xr-x 2 root root 4096 2010-12-14 21:53 cdrom
drwxr-xr-x 19 root root 3540 2012-03-15 17:54 dev
```

Cabe destacar que con esta línea estamos logrando lo mismo que si hubiéramos escrito:

```
ls -l / | head -n 5
```

Veamos otro comando de sed, “q” (quit) que nos permite obtener idéntico resultado:

```
ls -l / | sed “5 q”
```

Esto se interpreta como “luego de procesar la quinta línea, termine la ejecución.”

OTROS COMANDOS

Hay más instrucciones útiles. La “a” de add, agrega una línea con el texto que se le indique tras el patrón especificado:

```
echo “/* Comentario de código */” | sed “/\/* / a\\n”
```

Aquí se agrega una línea en blanco (recordemos que “\n” equivale a NEWLINE) luego de cada línea que contenga el principio de un comentario escrito en la forma “/* ... */”. Para ello, basta con especificar el

patrón del principio del comentario (/*), pero como tanto la barra como el asterisco son meta-caracteres, es necesario escaparlos con la contrabarra “\”. En vez de agregar texto al final del patrón encontrado, puede insertarse texto en una línea previa usando “i” (insert) en vez de “a”. Si quisiéramos agregar una línea antes y otra después del patrón, podemos agrupar los comandos, encerrándolos entre llaves “{” y “}”. Claro que aquí hay que tener en cuenta algo que aclaramos antes: cada comando sed debe estar en una única línea, y eso incluye tanto las llaves como los comandos agrupados en su interior. Para facilitar la escritura de este tipo de scripts sed, es conveniente editarlos y grabarlos, para invocarlos posteriormente con el switch -f, como hicimos antes.

Agrega líneas antes y después de un comentario

```
/\/*/ {  
a\  
\n  
i\  
\n  
}
```

**echo “/* Comentario de código */” | sed -f coment.
sed**

/* Comentario de código */

Otra posibilidad es cambiar la línea donde se encuentre el patrón con “c” (change) por el texto ingresado a continuación de la orden, o ninguno si queremos eliminarlo. Para eliminar todas las líneas de comentario prefijadas con una doble barra (//) podemos usar:

echo ‘// Comentario de código’ | sed ‘/\// c\’

Para esto mismo es también útil el comando “d” (delete) que borra las líneas que coincidan con la expresión especificada:

sed ‘/cadena_o_expresion_regular/d’

Es común el empleo de la orden “d” en conjunto con los rangos que vimos antes. Es posible borrar las primeras 10 líneas de un archivo de esta manera:

sed ‘1,10d’

ALGUNOS EJEMPLOS ÚTILES

Siempre es bueno ver la aplicación de lo aprendido en algo práctico. Aquí van algunos “one-liners” útiles con sed. Debe tenerse en cuenta que hacen un uso más avanzado de las expresiones regulares (un terreno complejo y fascinante por su utilidad, que amerita una nota en sí mismo) que las aquí mostradas. Por ejemplo, se incluyen los meta-caracteres “^” (que representa el comienzo de una línea) y “\$” (que representa el final de la misma).

Eliminar el espacio en blanco (espacios y tabuladores) al principio de una línea

sed ‘s/^[\t]*//’

Ídem, pero al final de una línea:

`sed 's/[\t]*$//'`

Ídem, pero tanto al principio como al final:

`sed 's/^[\t]*//;s/[\t]*$//'`

Imprimir las líneas que contengan una expresión regular (igual que “grep”)

`sed -n '/regexp/p'`

Eliminar todos los tags HTML de un archivo:

`sed -e 's/<[^>]*>//g' index.html`

... Y HASTA AQUÍ LLEGAMOS

Aunque parezca demasiada información para una sola herramienta, lo cierto es que apenas hemos logrado rascar la superficie de lo que sed es capaz de hacer. Utilizando los comandos que vimos en conjunto, más otras funciones que por razones de espacio tuvimos que dejar afuera, se pueden lograr cosas muy interesantes. Es que para cubrir sed en su totalidad, al igual que para awk (al que dedicaremos la tercera parte de esta serie) no alcanzan unas pocas páginas, e incluso hay libros enteros escritos sobre el tema, tanto digitales como en papel. Por lo tanto, dejamos en las manos de los lectores el seguir profundizando sus conocimientos. Como se dice en el ambiente UNIXERO, el mejor administrador o desarrollador es aquel que adquiere la mayoría de sus conocimientos por la vía de la práctica y la experimentación.



CHOQUE DE GIGANTES: FACEBOOK VS GOOGLE+

WEB 2.0 Y EL NUEVO CIBERESPACIO

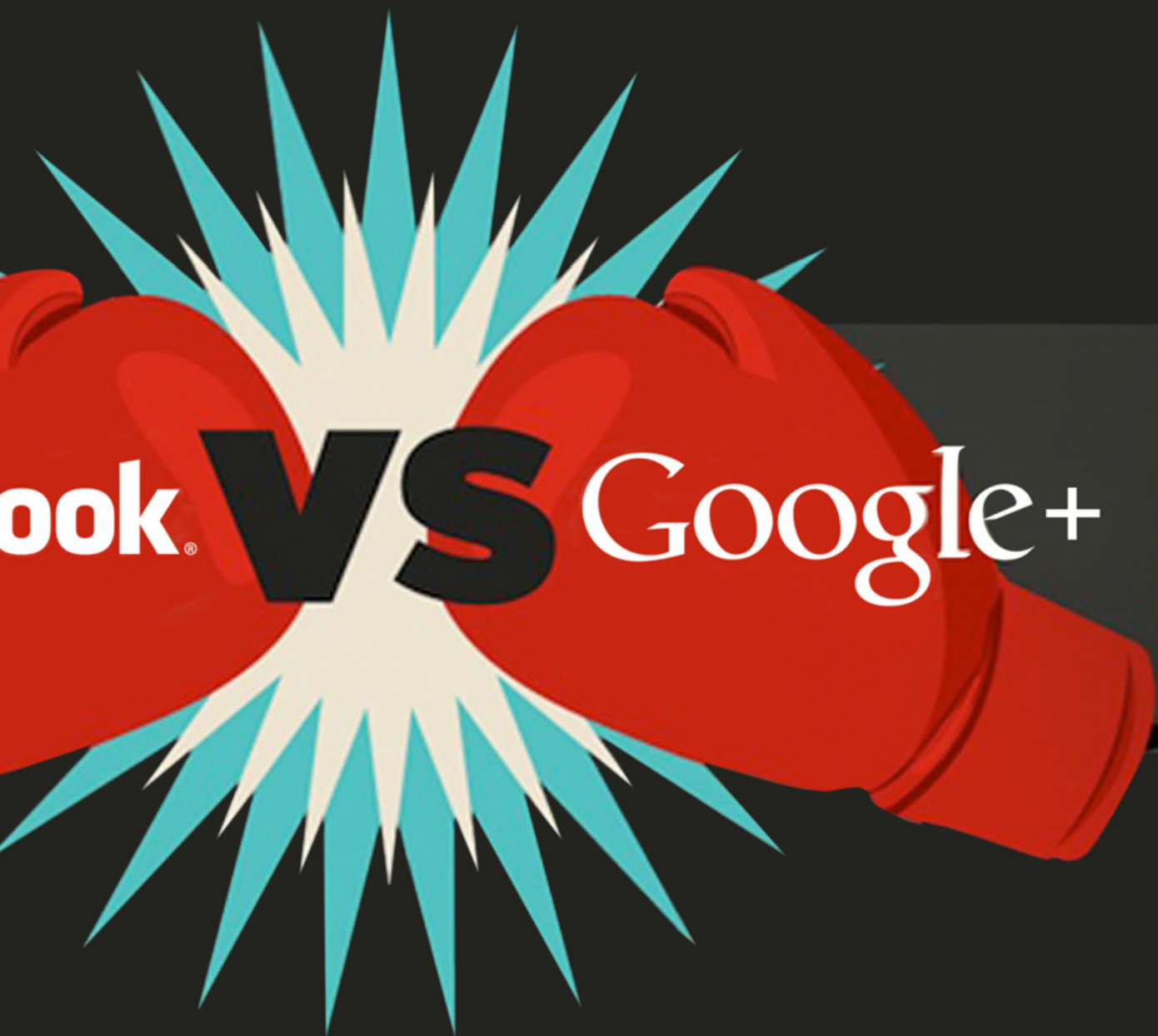
En los últimos tiempos se ha producido un notable cambio en el ciberespacio. A diferencia de los primeros tiempos de la web, ha comenzado a gestarse un fenómeno de masas al que suele referirse como Web 2.0. No más páginas que brinden información o servicios a usuarios pasivos, sino una transición que se ha dado de aplicaciones comunes y tradicionales hacia utilidades que funcionan a través de internet, específicamente la web, y todas ellas enfocadas al usuario final. Se trata de aplicaciones que generen colaboración y de servicios que reemplacen las aplicaciones de escritorio dando en todos los casos un papel activo a ese mismo usuario final.

Podemos mencionar como algunos ejemplos los blogs, las wikis (especialmente la más conocida del mundo, Wikipedia) y una vasta gama de sitios en los cuales la presencia del usuario y su feedback es y serán esenciales. Y de todas las clases de aplicaciones que podemos referir, probablemente, las que más encarnan este nuevo espíritu de la red son las llamadas redes sociales. YouTube, Twitter, Orkut: todas ellas se nutren de los usuarios y son ellos los que brindan el contenido a la red, no la red la que se los provee a ellos en su tradicional rol de consumidores pasivos de la información. Subiendo el video que realizó, comentando un suceso o actualizando su perfil, el usuario es el creador de su propio rostro de cara al mundo.



facebo

Gabriel Gaitano Ornia
gabriel.gaitanoornia@pixelscode.com



Es aquí cuando mencionamos al fin a la red social más conocida, Facebook. En ella se resume la web de hecho para muchos usuarios, y desde sus humildes orígenes a la fecha ha crecido exponencialmente en cantidad de usuarios y, sobre todo, en participación de los mismos. Y eso es la clave de su éxito, más allá de sus cualidades técnicas, ya que para toda red social la participación de sus usuarios es fundamental, la sangre que le da vida. Porque, ¿qué es una red social sin tráfico de información entre usuarios, sean estas opiniones, fotos, videos u otros recursos? La experiencia y la lógica nos dicen que nada en absoluto. Por eso, Facebook creció a pesar de que muchos usuarios se quejaban (y lo siguen haciendo) sobre aspectos fallidos como ser la interfaz, la privacidad, los frecuentes cambios de apariencia, etc. Simplemente “hay que estar ahí” a fin de mantener el contacto, y ese efecto de masa crítica constituye su bien máspreciado y, por cierto, más envidiado por el resto de los jugadores de la red.



Paul
Schlacter

The Mad Ones

10

facebook®



Ahora bien, la empresa líder de Internet es sin duda la omnipresente Google, dueña y señora de todos los caminos. Es muy sabido que si no está en Google o, mejor dicho, en sus búsquedas, no existe – como afirma la sabiduría popular de la red. Sin embargo, el crecimiento de Facebook ha sido tan firme que la importancia de su porción de torta se ha tornado, a la vez, atractiva y amenazante para Google, la cual con las compras de YouTube y Orkut inició su propia avanzada en el terreno de las redes sociales, con notable éxito en ambas. Sin embargo, nunca logró hasta este momento desbancar en su propio terreno a Facebook y los intentos de clonarla dentro del mundo Google han resultado sonoros fracasos.

En el mundo de los negocios, y especialmente de Internet y la tecnología, no avanzar es retroceder y retroceder es a la larga morir. Por eso está planteado el escenario para una lucha de gigantes entre Facebook y Google, de la mano de la más nueva creación de esta última: Google+, su gran apuesta por el control del ciberespacio.

Este es, probablemente, el momento para “triunfar o morir” para Google, al menos para las aspiraciones del gigante en los medios sociales. Facebook ya está suficientemente establecido en sí mismo como el líder sin disputa entre las redes sociales, y por cierto está bien encaminado en su meta de entretenerse a sí mismo en la mismísima “tela” de la red de redes. Si Google falla en este lance, es posible que ya no tenga una segunda oportunidad para volver a luchar. La pregunta en la mente de todos es si Google al fin se tornará “social” y tendrá éxito en desafiar a su rival.

Y EN ESTA ESQUINA...

Los contrincantes tienen al observarlos en detalle muchas similitudes producto, en muchos casos, de la observación del rival y de los comunes objetivos. Y también no menos significativas diferencias provenientes de las diferentes etapas de evolución y ambientes corporativos.

La historia de Facebook es en general bien conocida, habiendo tenido incluso su toque de épica con la película The Social Network. En ella se muestra a un grupo de jóvenes emprendedores que, tras muchas vicisitudes, logran quebrar su relativo ostracismo social

a través de la creación de un sitio donde vincular gente y sus historias. De todos modos, el principal fundador y actual líder de la empresa, Mark Zuckerberg, ha denostado la película afirmando: «Es divertido lo que se han empeñado en mostrar de forma acertada, porque todas las camisetas y todas las camperas que salen en la película, son camisetas y camperas que tengo. Hay otras cosas que hicieron bien pero otras, en cambio, que hicieron mal. Lo que peor está es el principio de la película. Empieza con una chica que nunca existió en la vida real y que en teoría me deja.

Algo que ha pasado en la vida real muchas veces,» agrega mordazmente. «Parece que la única razón por la que quería hacer Facebook era porque quería conocer chicas, o porque quería entrar en alguna institución social. Creo que eso es bastante significativo de cómo la gente que hace películas nos ve a los que estamos creando cosas en Silicon Valley. Parece que no se dan cuenta que hay gente que construye cosas porque les gusta construir cosas,» concluye su análisis el líder de Facebook. En definitiva, lo realmente cierto es que tras unos comienzos mínimos como sitio de encuentro entre estudiantes de la universidad estadounidense de Harvard, Facebook se expandió al mundo entero.

Con un formato en general fácil de entender y manejar por no expertos, Facebook es una forma sencilla de mantenerse en contacto y compartir toda clase de cosas con amigos, sean videos, comentarios o juegos. Con muy buena visión de negocio, los directivos de la red social han liberado una API que permite a terceros invertir en aplicaciones incluso con fines lucrativos, transformando así a Facebook en un vehículo donde muchos generadores de contenido se encuentran con usuarios de todo tipo.

Finalmente, en los últimos tiempos ha empezado a incrementarse el número de empresas deseosas de tener presencia en Internet a través de lo que Facebook puede proporcionarles: comunicación y una magnífica vidriera al mundo.

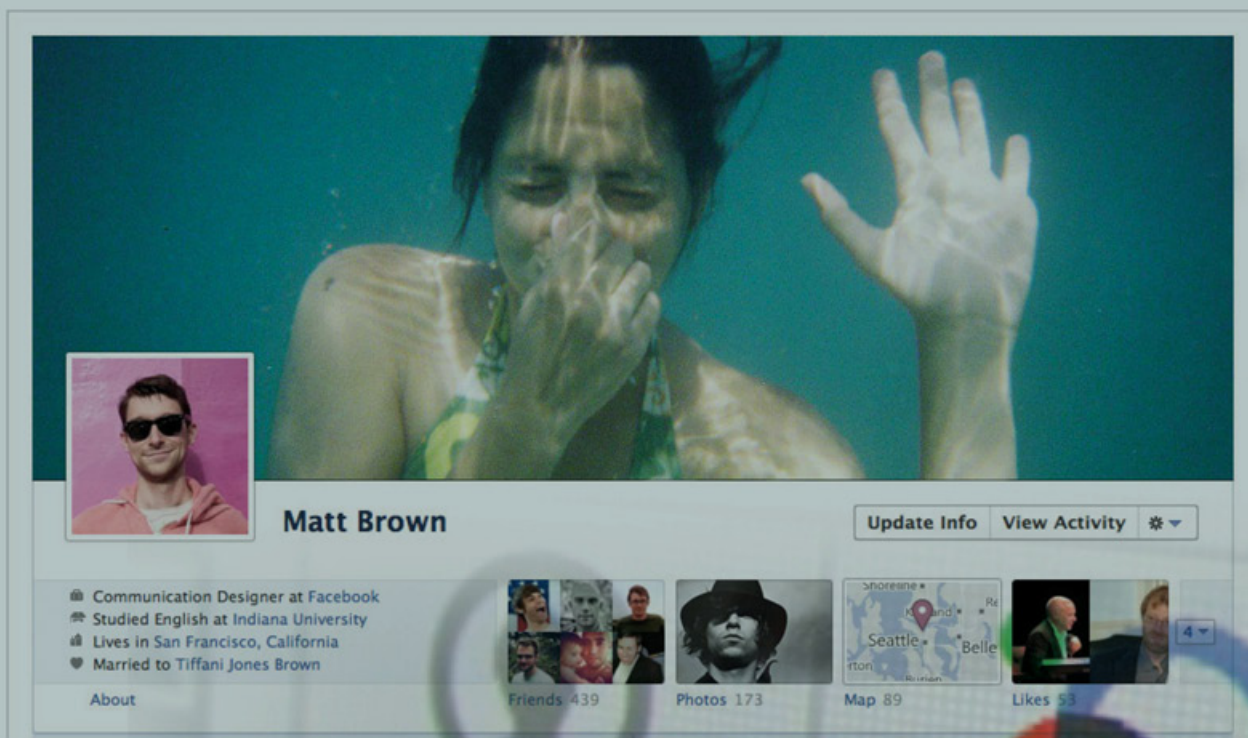
Google+



EL PODEROSO RETADOR

Aún después del lanzamiento por parte de Google de su candidato a destronar a los Zuckerberg's boys, han sido muchas las voces que han calificado a esta empresa como poco "social". Suele alegarse que los ingenieros de Google, apasionados por la matemática y creadores de formidables algoritmos de rastreo y búsqueda carecían del "toque" necesario para entender a los usuarios de una red social. Y los ejemplos estaban a la vista abonando esta forma de ver las cosas: en 2010 Google Wave y en el año siguiente Buzz fueron los primeros tanteos del gigante de las búsquedas por destronar a su

"masa crítica" que les permitiría enfrentar a Facebook y ponerlo de rodillas. Entonces fue que desde Google analizaron nuevamente lo sucedido, y realizaron los ajustes y toques que consideraron necesarios para poder competir con ventaja. Si lo que necesitaban era ofrecer algo nuevo y atractivo que Facebook no tuviera, se pondrían a buscarlo. Recursos por cierto no les faltaban, y así fue como surgió la iniciativa Google+, la cual a diferencia de sus fallidas hermanas ha tenido desde su aparición un notable suceso. Falta ahora ver si eso significará la victoria definitiva o no frente a su bien establecido rival.



adversario. Pero, como los primeros manotazos de un niño, fueron torpes y carentes de lo necesario para la ambiciosa tarea que se les requería. En principio, fueron problemas de difusión y seguridad/privacidad, pero más tarde quedó claro que eran demasiado parecidos a su rival. Si yo tengo algo que funciona y lo hace bien, ¿por qué cambiarlo? Enfrentados a esta sencilla lógica de los usuarios y padeciendo esas mencionadas torpezas, los productos mencionados jamás alcanzaron la ansiada



FORTALEZAS Y DEBILIDADES

A la hora de las comparaciones, notamos de inmediato que se mantiene una constante de los anteriores enfrentamientos: los productos de Google suelen atraer más a los usuarios con conocimientos de informática medianos o altos, o apasionados por la innovación tecnológica. Para desgracia de Google, el resto de los usuarios no comparte esta tendencia, al menos no por ahora, ni masivamente.

De acuerdo con estadísticas conocidas, un porcentaje alto de los usuarios de Google+ son desarrolladores, arquitectos de software, ingenieros y diseñadores. Por el contrario, su rival se haya muchísimo más difundido entre la masa de usuarios cuyo uso de la tecnología es más elemental.

Las innovaciones introducidas por Google+ esta vez sí son interesantes, y ameritan una prueba por parte del más recalcitrante usuario de Facebook: las más importantes son los círculos de amigos y los “hangouts” (traducidos en ocasiones como “quedadas” o “juntadas”).

Aprovechando los problemas de confidencialidad y seguridad que asolaron a Facebook, como quejas del hecho que muchas publicaciones realizadas en el perfil de un usuario podían llegar a ser perjudiciales para éste si eran vistas por el amigo o amiga inadecuados, Google+ introdujo el interesante concepto de los “círculos de amigos”, o sea una manera de agrupar por características comunes a ciertos usuarios y limitar lo que se publica al círculo por nosotros deseado. De este modo, una opinión de un usuario criticando el lugar de trabajo no tiene por qué ser vista por su jefe, el cual estará en un círculo distinto de amigos. Y este tipo de cosas sí atraería mucha atención sobre el producto.

Por otro lado, los Hangouts son una versión moderna de los tradicionales sitios de chat, por el momento mucho más reducidos en cantidad de integrantes pero pensados para dar una vía de conversación online ágil y práctica a sus usuarios.

Otra innovación muy interesante son las páginas de intereses personales que podemos crear (llamados “sparks”), donde veremos noticias y artículos sobre

los temas que seleccionemos como favoritos.

La difusión ha sido más ágil, las líneas de qué era y qué se podía hacer con el producto son mucho más claras en Google+ y queda demostrado que Google algo ha aprendido. Sin embargo, no faltan las voces que han señalado que estas ventajas son sólo temporarias. Facebook también aprende de sus errores y trabaja activamente para mejorar su apariencia y posibilidades, reforzar su seguridad y defender su espacio en el mundo de las redes sociales. Y conserva su atractivo intacto para el usuario menos técnico, así como para el más conservador que ya tiene mucha historia en el sitio. Justamente, para resaltar y aprovechar eso, Facebook sumó un nuevo aspecto -todavía resistido por algunos-, conocido como “Timeline” que permite ver en una línea de tiempo los post realizados; una especie de historial que apela a los recuerdos del usuario y su larga vinculación con Facebook. Y en esto, al menos, el recién nacido Google+ no puede seguir la iniciativa con algo similar.

Y AL FINAL SOLO PUEDE QUEDAR UNO

¿Quién será el definitivo ganador? Es claro que la lucha será enconada y que Facebook parte con mucha ventaja. Pero Google sigue siendo la más poderosa empresa de internet y tiene muchos recursos y capacidad de aprender de sus propios errores. Google+ ha incorporado mucha experiencia de los anteriores fracasos de las redes sociales fallidas de Google y su crecimiento en cantidad de usuarios ha sido vertiginoso. Es posible que el ganador se defina pronto o que persista durante un tiempo una dura pulseada entre ambos gigantes.

Pero de lo que no cabe duda es que en esta pelea los árbitros seremos, afortunadamente, nosotros, los usuarios, quienes con nuestro veredicto y uso decidiremos quién se queda con la corona de la red social más importante del mundo.

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*](http://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



Email Marketing con **tu marca**

Ganá dinero ofreciendo a tus
clientes la mejor herramienta.



EnvialoSencillo.com

dattatec.com