

SG

SOFTWARE GURU

Nº.51

CONOCIMIENTO EN PRÁCTICA
www.sg.com.mx

NOVEDADES

SQL SERVER

2016

PAG. 13

INTERNET DE LAS COSAS

Y SU IMPACTO EN LAS EMPRESAS

EL IMPACTO REAL DEL IOT EN LA EMPRESA
PAG. 18

EL MAYOR VALOR DEL IOT SON LOS DATOS
PAG. 21

HOLA MUNDO IOT
PAG. 40

"Software is eating the world, be part of this evolution."

SGNEXT

Conoce el nuevo paradigma
de desarrollo de software

SOLID DAY

30 de agosto



Internet of things +
Bots + VR Interfaces

LIQUID DAY

31 de agosto



DevOps + Testing +
Cloud-native architecture

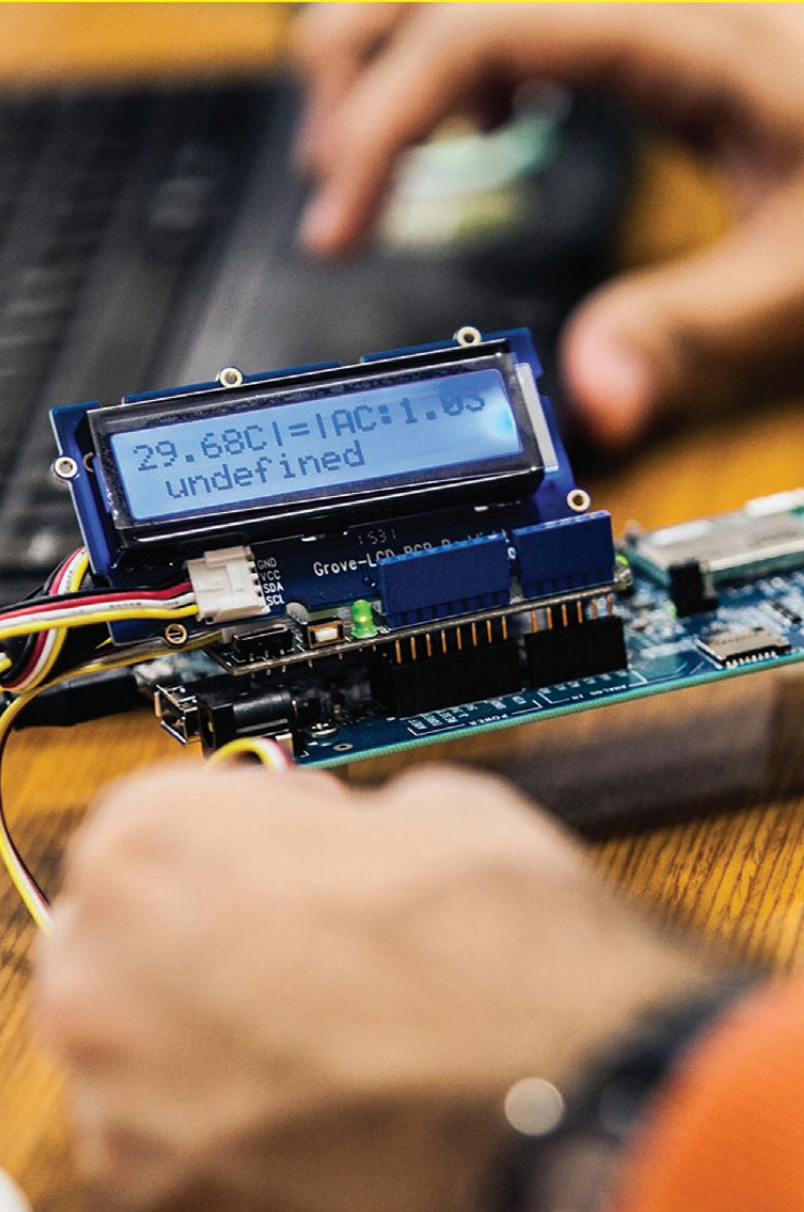
30 y 31 de agosto de 2016

Hotel Crowne Plaza WTC, Ciudad de México

<http://sg.com.mx/sgnext>



Asiste a SG NEXT y conoce las nuevas tecnologías, arquitecturas y métodos para construir software grandioso.



Descubre las mejores prácticas para el desarrollo, despliegue y testing para construir soluciones tecnológicas con alta calidad.



presentado por:



SG

SOFTWARE GURU

NO.51

CONOCIMIENTO EN PRÁCTICA
www.sg.com.mx

EN PORTADA

INTERNET DE LAS COSAS

016

Preparémonos para la disrupción que generará el internet de las cosas y su impacto en las empresas.

I INDUSTRIA
Y EMPRESAS

**5 Estrategias Clave para
Difundir Tu Startup** 014

T HERRAMIENTAS Y
TECNOLOGÍAS

Radar 012

Novedades 013

Zona Maker 040

V VOCES

**¿Estamos Listos en
QA para el IoT?** 010

P PRÁCTICAS

**UX También es Pensar
en Productos** 026

**Los Niveles de
Granularidad en
los Requerimientos
Funcionales** 028

**JavaScript y Herencia
Basada en Prototipos** 032

F FUNDAMENTOS

**Perdamos el Miedo al
RTFM** 050

C COLUMNAS

Tejiendo Nuestra Red 008

Prueba de Software 040

**Programar es un
Modo de Vida** 042

R REPORTAJES

Hackathon Cancún 006

Dev Day 4 Women 008

O EN CADA
NÚMERO

Noticias y eventos 005

Hardware 046

Humor 047

Biblioteca 048

EMPRENDIENDO

014

IoT y el nuevo paradigma



● **A través de los años**, en estas páginas hemos comentado de forma recurrente sobre cómo las fronteras del software cada vez son más difusas.

El software ya no es tan solo una herramienta para automatizar procesos departamentales, sino que se ha convertido en un elemento central de innovación en empresas de todos los sectores. Dentro de estas tendencias de innovación, una de las más importantes es el internet de las cosas.

El IoT está llevando el software a nuevos usos y fronteras. Así que si nos queremos mantener vigentes como profesionistas, necesitamos hacerlo parte de nuestro arsenal.

En SG estamos conscientes de este cambio de paradigma, y es por eso que hemos rediseñado nuestro congreso previamente llamado "SG Conference & Expo" como "SG Next", y estará enfocado en este nuevo paradigma de desarrollo de software, donde dominan temas como el internet de las cosas, bots, DevOps y arquitecturas en la nube.

Esperamos verte en SG Next, para que sigamos juntos en el camino de desarrollar mejores productos de software día con día.

El equipo de Software Guru

SG es posible gracias a la colaboración de

Dirección Editorial **Pedro Galván** | Dirección de Operaciones **Mara Ruvalcaba** | Dirección Comercial **Claudia Perea**
 Coordinación Editorial **Susana Tamayo** | Arte y Diseño **Oscar Sámano** | Suscripciones **Mariana Torres**
 Consejo Editorial: Luis Daniel Soto | Gunnar Wolf | Luis Vinicio León | Hanna Oktaba
 Ariel Jatuff | Emilio Osorio | Gloria Quintanilla | Jorge Valdés

COLABORADORES EN ESTA EDICIÓN

Roselyn Piñango, Edith Gómez, Curt Hall, Jorge Alvarado, Esteban Montelongo, Samuel Dos Reis, Eduardo Lugo, José Manuel Berruecos,
 Misael León, Guilherme Simoes, Carlos Vázquez, Erik Grueter.

EQUIPO SG

Coordinación de servicio **Yoloxochitl Juárez** | Marketing y Alianzas **Wendy Fabela**
 Developer Relations **Luis Sánchez** e **Hilda Ramírez** | Servicios online **Ivett Sánchez**
Contacto: info@sg.com.mx

SG Software Guru es una publicación trimestral editada por Brainworx, S.A. de C.V., San Francisco 238 Altos. Col. Del Valle. Los contenidos de esta publicación son propiedad intelectual de los autores y se hacen disponibles bajo licencia Creative Commons Attribution-NonCommercial 4.0 International. Todos los artículos son responsabilidad de sus propios autores y no necesariamente reflejan el punto de vista de la editorial.

Reserva de Derechos al Uso Exclusivo: En trámite. ISSN: 1870-0888. Registro Postal: PP15-5106. Distribuido por Sepomex.

Noticias

SG CTO FORUM

OSCON 2016

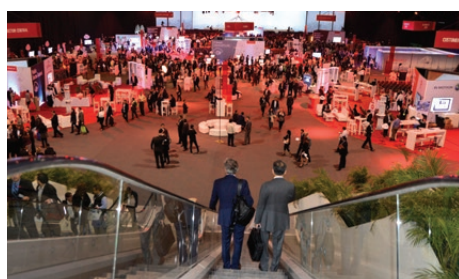
1

Del 16 al 19 de mayo se realizó en Austin, TX la edición 2016 del Open Source Convention (OSCON) producido por O'Reilly Media. OSCON es uno de los eventos de mayor tradición en la industria de TI, y es un espacio interesante porque se basa en valores de apertura y diversidad que van más allá de la tecnología. Hoy en día, hablar de las ventajas del open source ya no es necesario, porque esa batalla ha terminado y es el claro ganador. Es así que ahora las discusiones tienen que ver con hacer la industria más abierta hacia personas de distintos géneros, etnias y capacidades, y OSCON es reflejo de ello. En cuanto a tópicos técnicos, posiblemente el de mayor recurrencia fue el rol del open source en el ámbito del internet de las cosas.

Imagen cortesía de O'Reilly Media.



SG organizó una edición más de su meetup para líderes de tecnología empresarial, CTO Forum, en las instalaciones del Centro de Innovación BBVA Bancomer en la Ciudad de México. El evento se enfocó en “el nuevo paradigma de TI” derivado de la ola de transformación digital que estamos viviendo. Entre los conferencistas que compartieron opiniones y perspectivas estuvieron Domingo Suárez, CTO de HolaGus; Nicolas Gallegos, Gerente de Innovación de TI en Bayer de México; Mohamed Hassan, CTO de Segundamano; y Pedro Galván, Director de SG.



ORACLE CLOUDWORLD

3

Con la participación de más de 2,000 asistentes, se llevó a cabo Oracle CloudWorld México 2016 en Expo Bancomer Santa Fe. En este foro, Oracle dio a conocer casos de éxito en los que la tecnología jugó un papel fundamental en el desarrollo e implementación de nuevos negocios. También en el marco de este evento se anunciaron importantes lanzamientos, tales como la actualización de la familia Oracle Database Appliance; así como nuevos sistemas y un minicluster basados en el nuevo procesador SPARC S7 con capacidades de alto desempeño en seguridad, eficiencia y simplicidad de administración.



OPEN STACK DAY

4

La Fundación OpenStack y KIO Networks llevaron a cabo la tercera edición de OpenStack Day México en el World Trade Center de la Ciudad de México. En este espacio se dieron cita desarrolladores, administradores de sistemas de cómputo y usuarios, quienes intercambiaron ideas, conocimientos y experiencias alrededor de cómo utilizar la tecnología de código abierto OpenStack para administrar infraestructura de cómputo altamente escalable. En el evento se contó con cuatro escenarios alternos donde ponentes especializados platicaron sobre los beneficios de OpenStack y las ventajas competitivas que pueden brindarle a cada uno de los negocios que sepan y puedan sumarse a esta tecnología.

SG Organiza Hackathon en Reunión Ministerial de la OCDE



● Gracias a la colaboración entre Secretaría de Economía de México y el Comité de Asesoría Empresarial en la OCDE (BIAC), el pasado 20 y 21 de junio Software Guru tuvo el honor de organizar en Cancún el hackathon “Connected Lives, Connected Communities” como parte de las actividades de la Reunión Ministerial sobre Economía Digital de la OCDE.

El principal objetivo de realizar un hackathon fue permitir que líderes de gobierno y organismos industriales pudieran presenciar el proceso creativo de los jóvenes, colaborando para construir soluciones tecnológicas para enfrentar retos globales. Esta misión se cumplió con gran éxito, y los participantes de la reunión ministerial quedaron impresionados con la energía y creatividad de los participantes, quienes a lo largo de 24 horas construyeron soluciones innovadoras alrededor de los temas: inclusión social, ciudades inteligentes, herencia cultural y emprendimiento. El hackathon contó con la participación de 170 jóvenes procedentes de México, Colombia y Canadá. Adicionalmente, contó con el

apoyo de AT&T, Cisco, Disney, Google, Intel, Microsoft, Oracle, Verisign, y el Comité de Asesoría Técnica para Internet en OCDE (ITAC).

GANADORES

En #hackathoncancun se repartió una bolsa de más de 25 mil dólares en premios, siendo estos los ganadores:

- Nisi Vitae se llevó el premio como ganador absoluto además de ganar la categoría de ciudad inteligente, con una app que provee información sobre la salud del paciente de forma anticipada al equipo de atención médica antes de que llegue al lugar de una emergencia.
- Autonomi ganó la categoría de emprendimiento construyendo dispositivos wearable para asistir a personas con problemas de visión.
- VR-ehab ganó la categoría de inclusión social y el premio de Disney para el bienestar infantil con una app que aprovecha la realidad virtual para hacer la rehabilitación física más amigable para los niños.

- Time Stamps ganó la categoría de herencia cultural con una app que utiliza reconocimiento facial para brindar una experiencia inmersiva para estudiar historia.
- Oil Collection ganó el premio al mejor uso de la API de AT&T con un sistema de monitoreo para la recolección de contaminantes del medio ambiente.
- Here & Now ganó el premio a mejor uso de tecnología Google con una app para comunicar personas en caso de emergencias y desastres naturales.
- EcoFenix ganó el premio de IEEE con un sistema para fomentar el reuso y reciclaje de basura.
- Easy Lock ganó el premio Intel para hogar inteligente con el proyecto Easy Lock para una cerradura inteligente.
- Sufragia ganó el premio a mejor uso de tecnología Microsoft con una plataforma inteligente y descentralizada de participación ciudadana.
- Saverfid ganó el premio Oracle a mejor tecnología para la empresa con un sistema para reducción de energía eléctrica por medio del control automatizado de luces apoyado por detección RFID. ☺

Dev Day 4 Women

● Más de 200 profesionistas y amantes del código, se reunieron el pasado jueves 23 de junio, en las instalaciones de la Escuela Bancaria Comercial en Ciudad de México, para vivir la experiencia de la tercera edición del exitoso evento dedicado a mujeres desarrolladoras de software: Dev Day 4 Women.

Es una iniciativa aliada de todas aquellas mujeres que quieran compartir su conocimiento y experiencia, para apoyar y motivar a las nuevas generaciones en TI.

El evento fue posible gracias al patrocinio de Oracle, Luxoft, Accenture, Magma Labs, ThermoFisher Scientific, Escuela Bancaria Comercial, Praxis, Brainup Systems, GFT Technologies y Scio Consulting.

El evento comenzó con un fireside chat en el que Adriana Islas (CIO de Estafeta), Carolina Tatay (Directora General de Neta Systems) y Ana Briseño (consultora independiente), fueron las tres experimentadas “mujeres tech”, que compartieron experiencias y recomendaciones para las jóvenes mujeres profesionistas de TI.

Posteriormente se realizó un Ignite donde conocimos más sobre distintas iniciativas para mujeres en TI como Digirlz, Technovation Challenge, C Girls Code, GDG Wonder Coders, Women Who Code, Laboratoria México, WoMoz, y Código X.

El aprendizaje técnico también es parte de Dev Day 4 Women y en esta ocasión se concretó a través de talleres y pláticas. Anahí Salgado fue la instructora del

taller “Aprendiendo Java a través de los videojuegos” presentado por Luxoft. Por otro lado, Luz Gutierrez de Accenture nos compartió experiencias en el taller: “Agile: Transformando la cultura de trabajo en la industria de TI”. También tuvimos una plática sobre Go a cargo de Verónica López y otra sobre inteligencia artificial de Karla de la Loza.

Para cerrar esta edición de Dev Day 4 Women, Ophelia Pastrana experta en contenidos digitales, nos acompañó para presentar “Devotion by SG”, un nuevo videoblog para desarrolladores de software con las anfitrionas Maru Lango y Verónica López.

Pronto tendremos otra edición del Dev Day 4 Women. Para mayor información, visita <http://devday4w.com> ☺



Dime qué Desayunaste y Te Diré qué tan Productivo Estarás Hoy

Por Hanna Oktaba



La Dra. Hanna Oktaba es profesora de la UNAM y su objetivo principal es generar conocimiento a través de la creación y promoción de estándares. @hannaoktaba

● **Recientemente me topé en la red** con un artículo que, primero me llamó la atención por su título: “Productividad Ágil: Voluntad con enfoque de neurociencias” [1], y luego me cautivó por su contenido.

El enfoque al tema de productividad en Ingeniería de Software, abordado de manera similar como lo hacen las industrias “tangibles”, siempre me ha parecido inadecuado (con todo respeto para mis colegas “creyentes fervientes” en PSP, TSP y CMMI).

La forma en la que las empresas miden la productividad, es a través de un cálculo en el que se realiza una comparación entre los insumos y los productos, donde la eficiencia es lo que representa el costo por unidad de cada producto. [2]

El problema que percibo en la industria de software tiene dos aspectos: por un lado, producir software es un trabajo donde los insumos y la maquinaria se encuentran en el cerebro de un humano y los productos son abstractos, por lo tanto es una industria de lo intangible. Por otro lado, no se ha llegado a definir una unidad del tamaño de software de manera universalmente aceptada. En la prehistoria empezamos con líneas de código (LOCs), luego llegaron puntos por función (FPs), siguieron puntos por casos de uso (que nunca se estandarizaron) y ahora tenemos puntos COSMIC, que podrían ser buenos candidatos por ser estándar ISO, pero dependerá de la aceptación por parte de la industria. Sin la misma medida del tamaño, las definiciones de la productividad son incomparables. Para los que gustan de los albrures: en la Ingeniería de Software el tamaño sí importa :(.

¿Y QUÉ TIENE QUE VER LA VOLUNTAD CON TODO ESTO?

En el artículo mencionado al inicio, la autora nos explica que la fuerza de voluntad es la que nos permite tomar decisiones para hacer algo, a veces difícil y no necesariamente muy agradable, pero que al final nos traerá una gratificación.

Díganme, si participar en un proyecto de software no es hacer algo a veces difícil y no necesariamente muy agradable, pero que finalmente algún día nos pagarán. Pero si alguien, como yo, pensaba que la fuerza de voluntad dependía en gran medida de la “zanahoria y el látigo”, estaba muy equivocada. Para tener fuerza de voluntad necesitamos tener suficiente lisina (un aminoácido) que nutre nuestra capa prefrontal del cerebro (figura 1). Y, ¿de dónde viene la lisina? De las proteínas que comemos. Así que, si desayunaste huevos con frijolitos quédate tranquilo, vas a ser muy

productivo, pero si solo comiste pan dulce con café, tu concentración en el trabajo va ser muy cortita :(.

El consejo no es solamente desayunar y comer proteínas para ser productivo. La autora nos da unas pistas de cómo ahorrar la lisina para poder trabajar concentrados por más tiempo durante un día:

- Crear hábitos.
- Usar rituales.
- Trabajar en modo fluido.

Un hábito es algo que sabemos, podemos y queremos hacer. Ir a bañarse después de levantarse es un hábito que sabemos cómo hacerlo, si tenemos regadera, podemos hacerlo y si hay agua caliente, queremos hacerlo :) . La gratificación al final también se siente. Los hábitos requieren muy poca fuerza de voluntad, porque nuestro cerebro reconoce rápido que es algo habitual y lo hacemos sin mucho esfuerzo. Lo malo de adquirir nuevos hábitos es que para que se nos “peguen” necesitan repetirse diariamente por lo menos ¡70 días! Con este dato entendí porque mis propósitos de año nuevo se vuelven tan irreales.

Un ritual es una ceremonia o acción realizada como costumbre. Los rituales nos dotan de energía. Uno de los rituales más bonitos que he encontrado en México es el Día de Muertos, aunque cuando por primera vez vi en el altar del posgrado de la UNAM una calaverita de azúcar con mi nombre en la frente, me asusté. Los rituales y los hábitos son como caminos bien trazados en nuestro cerebro, que no cuesta esfuerzo volver a recorrerlos.

Un modo fluido de trabajo es cuando estás concentrado en una tarea única, interesante, compleja, pero no demasiado, desconectado de tu entorno y en el pico de tu concentración. La resistencia (física) y la paciencia (mental) también ayudan. El modo fluido de trabajar se dificulta cuando la tarea, que tenemos que hacer, no es ni interesante ni compleja, en este caso nos aburriríamos rápido y buscamos distracciones para no hacerla. Otro extremo es cuando la tarea es tal vez interesante, pero tan compleja que no sabemos ni cómo empezar. En este caso nos angustiamos, las cosas no salen y podemos caer en el estrés. El estado de estrés (figura 2) nos lleva a perder el foco en el trabajo, lo que causa la mala calidad de los resultados, cuyos defectos tenemos que corregir, lo que nos causa mayor estrés. Y de la gratificación ni hablar, puro regañón. ¿Quién no ha vivido este círculo vicioso?

¿Cómo esto se relaciona con el trabajo de desarrollo de software?

HÁBITOS

En el caso de las prácticas SCRUM, las reuniones diarias realizadas siempre al inicio del día, a la misma hora, en el mismo lugar, con asistencia de todo el equipo y con las mismas preguntas es un hábito que, bien llevado a cabo, carga a todos con energía necesaria para empezar con mucha voluntad el trabajo (suponiendo que todos desayunaron proteínas :). Otro ejemplo de hábito es cuando uno, al terminar una tarea, revisa el resultado antes de entregarlo y hace todo lo que se pide para que otros se enteren que ya la había terminado (por ejemplo, mover la tarjeta en el tablero y/o reportar las horas dedicadas). Avisar que uno había terminado bien una tarea es gratificante ¿verdad?

RITUALES

No se lo digan a nadie, pero todas las buenas prácticas de los modelos, estándares y libros de Ingeniería de Software son rituales. Qué tal el ritual del *Sprint* de SCRUM. Ya desde el nombre, carrera corta de atletismo, suena a misterio. Inicia con la reunión de la planificación a dos tiempos en la que participa activamente el equipo de desarrollo y el (igual de misterioso) *Product Owner*. El *sprint* tiene una duración corta de entre 2 a 4 semanas, que para el desarrollo de proyectos de software es efectivamente una carrera corta, y termina en la revisión de software con el *Product Owner* y en la retrospectiva del equipo. Al final del *sprint*, si todo salió bien, la satisfacción del equipo debería recargar sus energías para aguantar la siguiente carrera, igual de corta, que empieza la semana siguiente. Ni toman en cuenta, que en el atletismo los cuatro sprints de 100 metros se hacen por relevos :).

Otros ejemplos de rituales son las diferentes técnicas para comprender y especificar a los requerimientos como los casos de uso o historias de usuario. Empiezan con el peregrinar para entrevistar a los diferentes representantes del cliente para comprender qué diablos quieren y qué significa lo que quieren. Continúan con los intentos de expresarlo de alguna forma, que sea de utilidad posterior para guiar al desarrollo. Luego sigue la batalla con los involucrados por la confirmación de que lo que se expresó como requerimiento ellos lo entienden y quieren que sea implementado. Cuando termina este ritual, los desarrolladores se llenan de energía porque, por un rato, piensan que allí terminó la batalla por comprender los requerimientos. ¡Qué ingenuos!

La mala noticia es que en estos rituales los que ahorran lisina son solo aquellos que los han repetido muchas veces, es decir, ya se volvieron sus hábitos. Los que son novatos van a sufrir.

MODOS FLUIDOS DE TRABAJO

Para trabajar de esta manera en el desarrollo de software, es indispensable que el reto que tenemos que atacar sea acorde con nuestros conocimientos y habilidades. Si se deja una tarea habitual como por ejemplo, documentar un caso de uso o una historia de usuario a un principiante que apenas fue capacitado (o mintió que sabía como hacerlo), se va a sentir muy angustiado y no va a poder hacerlo bien. O al contrario, si dejamos a un analista experimentado una tarea de documentar el requerimiento de "iniciar la sesión", se va a sentir subutilizado. En ambos casos no van a sentir la satisfacción de su trabajo y la productividad va estar por los suelos.

El consejo final es: usa rituales (procesos) para iniciar el hábito de trabajar con fluidez. Y come proteínas :).

PD. Esta es mi columna número 50 para la revista Software Guru. Durante más de 11 años repito un ritual de escribirla, que ya se volvió mi hábito y lo hago de manera fluida. Cuando la releo ya publicada me siento muy bien :).

Referencias

[1] A. Obukhova. "Agile Productivity: Willpower and the Neuroscience Approach". <http://swgu.ru/rg>

[2] <http://definicion.de/productividad>



¿Su empresa está lista para los desafíos del Internet de las Cosas? Porque sus competidores ya lo están...



Desarrollo de Software a la Medida

Desarrollo de Apps Móviles



Personal de TI bajo Outsourcing +

Training y Certificaciones



- *Android básico
- *Android experto
- *IoT Android Wear
- *iOS-SWIFT básico



- *Redes y Servidores
- *Proyectos de SW

Consulta fechas en:
cursos.dwsoftware.mx

Partner
UNIFACE.COM
Advanced Development Technology

CONTÁCTENOS

✉ ventas@dwsoftware.mx

☎ (33) 3030 7100

🌐 www.dwsoftware.mx

🐦 @DWSsoftware

📘 /DWSsoftware

¿Estamos listos en QA para el IoT?

Por Roselyn C. Piñango

• **Las tendencias tecnológicas** son el titular principal de nuestro día a día, no ha sido suficiente que los sistemas hayan invadido las industrias de todos los sectores con soluciones para agilizar sus procesos operativos y de gestión; que los profesionales de todos los ámbitos tengan que interactuar con sistemas en su vida laboral y personal, desde hace un tiempo llegó el momento en el cual dispositivos de uso diario (en los automóviles, para el cuidado de la salud, para el hogar) también son protagonistas de esta “invasión” que sin duda, promete tener una envergadura mayor porque se trata de intervenir en la vida diaria de todos para seguir adaptándose al dinamismo reinante.

El Internet de las Cosas (IoT) supone la transformación del rol de Aseguramiento de Calidad (QA) porque involucra pruebas del hardware, software, la transmisión masiva de cantidades de información en tiempo real e incluso “signos de inteligencia” en los dispositivos dado que su enfoque es “no determinista” y que deben ser sometidos a pruebas en diferentes niveles y de diferentes tipos. Ahora bien, ¿estamos listos en QA?, ¿qué cambios requiere QA para incorporar en su portafolio de servicios las pruebas IoT?, ¿qué nuevas habilidades requiere un probador IoT?, ¿vamos en la dirección correcta?

Algunos de los retos a considerar en QA en aras de incorporar en un plazo razonable pruebas asociadas al IoT son:

- En cuanto a las necesidades del entorno no es suficiente un ambiente de pruebas, no solo deben contemplarse ambientes para las diferentes plataformas donde funcionan los sistemas sino en diferentes condiciones. Un caso extremo será por ejemplo probar el sistema a altas/bajas temperaturas. Es requerido incorporar laboratorios de prueba.
- En relación a la estrategia de QA toma importancia la ejecución de pruebas de interoperabilidad, rendimiento, mantenibilidad, seguridad (ataques de pruebas a dispositivos riesgosos, privacidad, autonomía y control), compatibilidad y usabilidad.
- Algunas de las herramientas de prueba que serán casi indispensables serían los simuladores, generadores de datos, virtualización, monitores, consolas y análisis estático en aras de garantizar la mantenibilidad del código que ahora debe estar optimizado en virtud a la característica de los dispositivos donde operará. Actividades que hoy en día hacemos manualmente ya no será una opción ante la cantidad de escenarios a probar.
- En particular sobre las pruebas de usabilidad, será importante cambiar de paradigma y pasar de validar requerimientos a asegurar la experiencia de usuario, ahora las acciones de usuario se amplían con la voz del usuario, su movimiento y su forma de tocar el dispositivo.

Roselyn C. Piñango Díaz es Ingeniero en Computación con especialización en Sistemas de Información. Gerente de QA Factory en Global R, Venezuela. Cuenta con 10 años de experiencia en desarrollo y pruebas de software; y certificaciones: OCE SQL, OCA PL/SQL Developer, CTFL ISTQB, ASE HP ALM v11, CAA SAPB1.

Somos **especialistas** en diferentes ramas relacionadas con el aseguramiento de la **calidad**. Más de **500 empresas atendidas** son parte de nuestros **proyectos implementados exitosamente**.

» Servicios «



Mejora de procesos



Herramientas tecnológicas



Aseguramiento del control interno de TI



Mejora de capacidad técnica



Estrategia de negocios

- Con el auge del IoT surge un híbrido entre un par de técnicas de diseño de casos de prueba (caja negra y blanca): las pruebas caja gris que consideran que para diseñar es necesario entender la arquitectura orientada a eventos, hardware, protocolos de conectividad, etc., además de los requisitos del cliente

- Algunos de los nuevos conocimientos requeridos a los probadores para convertirlos en profesionales más integrales: ambientes embebidos, entender sobre el hardware, redes, etc.

- Continúa la premisa que no todas las pruebas serán automatizadas, será valioso llegar al balance de pruebas automáticas y manuales según se requiera.

Tendencias como los enfoques ágiles, donde el probador pasa de ser actor de reparto a protagonista del ciclo de vida de desarrollo de software y a involucrarse con todas las actividades así como las habilidades del cliente y programadores, nos llevan hacia la dirección correcta en torno al IoT, si bien en algunos países de Latinoamérica parece estar lejos la llegada de IoT desde QA, deben replantearse visiones que hasta ahora habían funcionado: no es suficiente buscar un sistema que “funcione”, IoT requiere validaciones robustas en función de ratificar que la “experiencia de usuario” sea la esperada, debemos incorporar nuevas herramientas CAST como emuladores, generadores de datos y de virtualización. Se convierten en indispensables las pruebas de rendimiento, usabilidad, compatibilidad y seguridad, y definitivamente el probador debe involucrarse aún más con los procesos del negocio, conceptos de desarrollo, hardware, software, redes y protocolos de comunicación que le permita seguir detectando fallas en un entorno mucho más complejo e interrelacionado, porque IoT sugiere una visión 360 para QA, es el momento de trabajar en ello. ☺

¡Contáctanos!

Oficina Matriz Jalisco

Tel: +52.(33).3030.7348

Oficina Ciudad de México

Tel: +52.(55).5211.9757

info@qualtop.com
www.qualtop.com

1

YA VIENE SPARK 2.0



Con apenas un par de años de vida, Spark es una de las tecnologías más candentes actualmente en el ámbito del análisis de datos. La novedad es que Spark 2.0 ya se encuentra disponible en una versión previa, y promete bastante. Entre las mejoras introducidas en Spark 2.0 destaca el amplio soporte para SQL. Esto facilitará significativamente migrar datos de bases de datos relacionales hacia Spark. Las APIs de Spark 2.0 también incorporan mejoras significativas, orientadas especialmente a escenarios de machine learning. Otro campo de mejora importante es el desempeño. Spark 2.0 utilizará el nuevo motor Tungsten que implementa varios trucos, tales como emitir bytecode optimizado en tiempo de ejecución para colapsar queries completos en una sola función; esto hace que Spark 2.0 tenga mejoras de desempeño de alrededor de 10x comparado con Spark 1.6. Spark 2.0 todavía es una versión previa y no está listo para producción, pero te invitamos a descargarlo y experimentar.

<http://spark.apache.org>

2 FIREBASE SE
CONVIERTE EN
PLATAFORMA



A finales del 2014 Google anunció su adquisición de Firebase, un servicio de Backend as a Service para apps móviles (MBaaS). El conjunto de funcionalidad de Firebase era pequeño comparado con otros MBaaS, ya que sólo proveía base de datos en la nube, hospedaje (CDN) y autenticación de usuarios; pero lo hacía bastante bien y se hizo popular. Durante el Google I/O 2016 se anunciaron nuevas capacidades para Firebase, que lo expanden para no solo ser un MBaaS, sino una plataforma unificada para aplicaciones móviles y web avanzadas. Entre las nuevas capacidades de Firebase destacan: Analytics para fácilmente dar visibilidad del comportamiento de los usuarios de tus apps; Cloud Messaging para proveer mensajería y push notifications; Remote Config permite ajustar el comportamiento de tus apps (por ejemplo esconder o habilitar funcionalidad específica) inmediatamente en todos los usuarios sin necesidad de que tengan actualizar la app; Test Lab, un servicio automatizado de testing de apps en dispositivos reales.

Conoce más en <http://firebase.google.com>

3 PROJECT BLETCHLEY
= BLOCKCHAIN
SOBRE AZURE



Project Bletchley es un nuevo proyecto de Microsoft para incorporar capacidades de blockchain en la plataforma Azure. Ya desde hace meses Microsoft había lanzado Azure Blockchain as a Service, que permite generar rápidamente un ambiente de desarrollo en Azure con capacidades de crear y gestionar blockchain. Desde entonces, la empresa ha estado estudiando los requisitos y capacidades deseadas por las empresas que desean usar el blockchain y Project Bletchley busca soportar dichos escenarios de uso. Project Bletchley se basa en dos conceptos: blockchain middleware y cryptlets. El middleware es un mecanismo de interoperabilidad con otros servicios de Azure para resolver aspectos como gestión de identidad, reporte y analítica. Los cryptlets son un mecanismo de interoperación y extensibilidad para implementar servicios basados en blockchain que requieran información adicional o integración con sistemas terceros.

Conoce más sobre Project Bletchley en el whitepaper disponible en <http://swgu.ru/ri>

4 FUNDACIÓN ECLIPSE
TOMA PROTAGONISMO
EN IOT



Empresas como Apple, Google y Amazon están iniciando su batalla por el dominio del IoT para consumidor. La aspiración de todas ellas es lograr acorralar a los usuarios dentro de su propio ecosistema. En el caso de las empresas, dicha estrategia no es efectiva ya que la estandarización e interoperabilidad es clave. Es por ello que el grupo de trabajo para IoT en la Fundación Eclipse está cobrando especial importancia. Recientemente el grupo anunció nuevas versiones disponibles para varios de sus proyectos, entre ellos:

- Eclipse Paho 1.2, librerías e implementaciones abiertas de protocolos de comunicación IoT.
- Eclipse Kura 2.0, framework para gateways IoT.
- Eclipse SmartHome 0.8, framework para construir soluciones de automatización del hogar.
- Eclipse OM2M 1.0, plataforma para construir servicios M2M en un ambiente heterogéneo.

Adicionalmente anunció la creación del proyecto Eclipse Kapua, una plataforma de gestión de dispositivos periféricos IoT resolviendo aspectos de conectividad y configuración en ambientes heterogéneos. Te invitamos a conocer más sobre estos proyectos en <http://iot.eclipse.org>



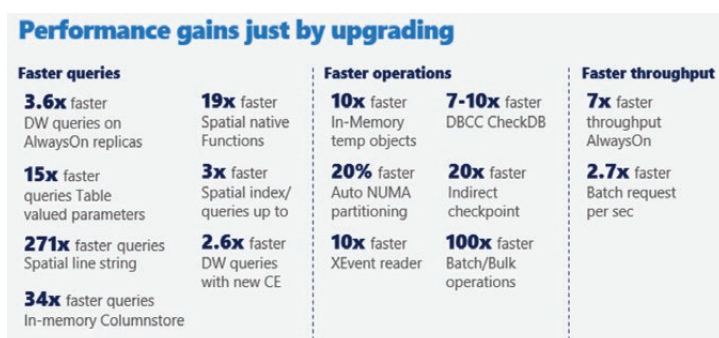
Novedades de SQL Server 2016

Por Pedro Galván

● **SQL Server 2016 ya está disponible de forma general**, y recientemente participé en un taller a fondo sobre las nuevas características que incorpora esta nueva versión y puedo decir que quedé gratamente sorprendido. A continuación comparto un resumen de las capacidades que más me han llamado la atención.

DESEMPEÑO

SQL Server 2016 tiene un gran enfoque en mejorar el desempeño, y mucho de esto se logra gracias a mejoras en el soporte de cómputo en memoria. SQL Server 2014 ya soportaba tablas en memoria, pero la implementación tenía muchas limitaciones (por ejemplo, no soportaba llaves foráneas y solo se podían tener tablas de máximo 256 GB). Esto ha mejorado drásticamente en SQL Server 2016; ya se soportan llaves foráneas, constraints únicos, operaciones paralelas, y ahora es posible tener tablas en memoria de hasta 2 TB. La siguiente figura ilustra las mejoras en desempeño.



SEGURIDAD

Una de las limitantes en términos de seguridad de versiones anteriores de SQL Server era que no soportaba seguridad a nivel de renglones (row-level security) para restringir que ciertos usuarios solo tuvieran acceso a un determinado rango de datos. Esto es clave en escenarios multi-tenant. SQL Server 2016 ya soporta esto.

Otra capacidad interesante es el cifrado transparente (transparent queryable encryption). Los datos se guardan cifrados en la base de datos, y se mantienen cifrados a través del canal de comunicación, y solo son descifrados hasta que llegan a la aplicación final (el driver se encarga de hacer el cifrado y descifrado). Esto tiene dos ventajas: si los datos son extraídos del servidor o captados en la red por un atacante, los datos estarán cifrados; la segunda ventaja es que al hacerse el cifrado y descifrado en el mismo punto, se pueden utilizar algoritmos de llave simétrica, que son mucho más rápidos. Este cifrado se puede configurar a nivel de columnas individuales.

COMPATIBILIDAD CON HERRAMIENTAS DE BIG DATA

SQL Server 2016 tiene muy buenas capacidades de interoperación con herramientas de analytics y big data. Entre ellas:

- **Soporte a R** - SQL Server 2016 incluye soporte para R, permitiendo que los científicos de datos puedan realizar experimentos de datos directamente sobre las bases de datos, sin necesidad de exportar hacia archivos de texto. De la misma manera, los desarrolladores pueden programar stored procedures que incluyan código R para hacer analítica compleja de forma estandarizada y parametrizada, lo cual fomenta su reutilización.
- **Polybase** - Un conector para integración con Hadoop. Permite interactuar con datos en Hadoop desde SQL Server de forma sencilla, sin necesidad de entrar en detalles de HDFS o MapReduce ni usar un ETL. Polybase no es una tecnología nueva, ya se ofrecía como parte del Microsoft Analytics Platform System, y ahora se incluye en SQL Server 2016.
- **Soporte a JSON** - SQL Server 2016 permite que los resultados de un query se exporten como JSON, y también permite tomar un JSON e importarlo como SQL hacia una tabla.

EXTENDER DATOS A LA NUBE

SQL Server 2016 está diseñada para aprovechar las ventajas de la nube. Un reflejo de esto es la capacidad "stretch". Esto permite que podamos elegir qué partes de nuestra base de datos pueden almacenarse en la nube, en lugar de nuestro data center local. Todo esto se hace de forma transparente para el usuario y el desarrollador. El administrador de base de datos simplemente define las políticas bajo las cuales los datos se pueden poner en la nube (ej. datos más viejos o con menor riesgo) y SQL Server gestiona esto de manera transparente. Al realizar un query, SQL Server se encarga de traer los datos donde quiera que estén, ya sea en la base de datos local o en la nube, y entregarlos como un solo conjunto de resultados. Esto se puede combinar con la capacidad de cifrado de punta a punta, de manera que los datos que se extienden hacia la nube siempre están cifrados.

FUTURO Y CONCLUSIONES

Para algunas empresas, el principal problema de SQL Server no es el producto como tal sino el hecho de que solo funcione sobre Windows. La buena noticia es que hace un par de meses Microsoft anunció que se está trabajando para crear una versión de SQL Server para Linux. Ese producto todavía no está listo y se espera que se lance hasta mediados de 2017. Aun así, vemos esto como una señal positiva de hacia dónde van las cosas.

En general, podemos decir que SQL Server tiene un conjunto muy atractivo de nuevas capacidades, y tiene un rumbo prometedor. ☺



5 Estrategias Clave para dar a Conocer tu Startup

Por Edith Gómez

● **De nada sirve tener la mejor startup**, con las mejores ideas y el mejor talento trabajando en ella si no encontramos la forma adecuada de contárselo al mundo. Como dueño de una startup debes ser capaz de explicar de una forma clara, sencilla y rápida a inversionistas y periodistas de qué va tu idea de negocio y captar su atención para crear en ellos la necesidad de querer saber más sobre tus proyectos y aspiraciones.

Sabemos que crear una startup no es fácil, pero darla a conocer después de todas las dificultades que hemos atravesado, resulta pan comido, se los aseguro. El problema es que si lo hacemos mal, pasaremos desapercibidos y nos sentiremos algo ignorados. Por eso, si queremos captar la atención de un periodista (que recibe al día cientos de peticiones similares), debemos centrarnos en elaborar el pitch perfecto para conectar con él y “engancharlo”. Así que atiende a estas 5 estrategias efectivas para conseguirlo:

CONOCER EL MEDIO AL QUE TE DIRIGES

Cuando hemos decidido contactar con algún medio de

comunicación para dar a conocer nuestra startup, es importante que seleccionemos el medio adecuado, ya que si nuestra empresa trata sobre videojuegos, no debemos perder tiempo en contactar con medios que hablan sobre salud o deporte. Una buena idea sería crear una lista de periodistas (de diferentes medios) que traten sobre los temas que abarca nuestra startup.

INVESTIGAR Y ESTUDIAR EL PERFIL DEL PERIODISTA

Si piensas que enviar miles de emails a periodistas de diferentes medios con la ayuda de una plantilla, te ayudará a captar su atención, estás equivocado. Hay que enviar emails personalizados y para ellos debemos conocer un poco más a la persona a la que nos dirigimos, conocer sus intereses, seguirla en Twitter y observar sobre qué escribe, cómo es su humor y cómo se relaciona con sus seguidores, por ejemplo.

Entonces será cuando le enviemos un email explicándole que hemos visto que escribe sobre un tema específico y resulta que nuestra startup trata también sobre ese tema. Por eso, le proponemos que escriba sobre nuestra startup, porque es un tema

interesante para su medio y para él, y le facilitaremos la búsqueda de información para escribir sobre la noticia.

CONTAR EN POCO TIEMPO LO MÁS IMPORTANTE

Primero, debemos presentarnos, decir cómo nos llamamos, el nombre de nuestra empresa, y justo después (en no más de 30 segundos) explicar brevemente qué hacemos. Nuestro objetivo en este paso será transmitirle que hemos detectado un problema o necesidad y que la empresa tiene la solución y va a satisfacer al cliente. Además de esto, deberemos señalar en qué nos diferenciamos del resto y cuál es nuestra propuesta de valor. Esto quizás sea lo más difícil de explicar, pero ese esfuerzo será recompensado si lo conseguimos.

CONTAR UNA HISTORIA

El periodista sabe que para enganchar a sus lectores debe contar una historia que emocione, divierta, llene de alguna forma. Por tanto, no debemos aburrirle con datos informativos de nuestra compañía, sino contarle una historia que provoque reacciones en el lector (que motiven) y para ello no podemos olvidarnos de nombrar al equipo, las dificultades vividas hasta el momento, los logros y otras curiosidades y anécdotas entrañables.

ELEGIR EL MEJOR MEDIO

Si contamos con un buen pitch y nuestra historia engancha, el periodista contactará con nosotros sin pensárselo dos veces. Es habitual que al ponernos en contacto con el periodista, este no conteste de inmediato. Lo ideal es esperar dos o tres días y si no responde, volver a intentarlo.

El medio más recomendable para contactar con el periodista será el correo electrónico. De esta forma, el periodista lo leerá cuando tenga un hueco, sin presión, y decidirá qué hacer cuando lo considere. De otra forma, con una llamada telefónica (aunque la respuesta es inmediata), le resulta más incómodo. Twitter es otra opción. Puedes captar su atención con un tuit y conseguir que te siga. De esta forma, podrías enviarle el pitch por mensaje privado.

Si no recibimos respuesta por ninguno de estos medios, querrá decir que debemos revisar el pitch y mejorarlo. Algunos, sin embargo, se toman la molestia de escribir para decir que el tema no les interesa. Podemos aprovechar esto para preguntarles por qué no les gustó y qué hacer para mejorarlo. Recibamos esa crítica de forma positiva y guardemos esa información para aplicarla con otros medios en un futuro. ☺

Edith Gómez (@edigomben) es editora en Gananci, apasionada del marketing digital, especializada en comunicación online. Se niega a irse a la cama cada noche sin haber aprendido algo nuevo. Le inquietan las ideas de negocio y, más aún, aportar una mirada creativa al pequeño mundo en el que vivimos.

www.dimtec.com

Para **todo problema** en la web tenemos **una solución...**

En **DIMTEC** nos especializamos en

- Seguridad
- Aceleración
- Distribución de Contenido
- Video

Aceleración y protección web para PYMES a un **COSTO PREFERENCIAL**

DIMTEC
DIGITAL MEDIA TECHNOLOGIES

¡Contáctanos ahora!
ventas@dimtec.com
01800 083 4683

Akamai **NETALLIANCE**
GOLD



LAS COSAS, EL INTERNET Y EL INTERNET DE LAS COSAS

En la evolución de la tecnología, y así como también sucede en la industria de la moda, todo en sí está marcado por tendencias que dictan el cómo, el por qué e incluso el dónde o el quién. Ante nuestros ojos hemos sido testigos de la llegada de los teléfonos celulares que hoy son smartphones, hemos visto cómo la generación de datos ha crecido hasta llegar al big data acompañado de su mundo bizarro: el dark data. Por supuesto sabemos que lo de hoy es el almacenamiento en la nube, que hay públicas y privadas; esas son solo algunas de las tendencias que nos han llevado de la mano hasta llegar a un punto de interconexión tal que ya no solo se trata de dispositivos diseñados y desarrollados para comunicarse entre sí, por ejemplo, sino de prácticamente cualquier cosa que tenga potencial para entregar y recibir datos valiosos de los usuarios, así es entonces que llegamos al internet de las cosas, IoT de cariño.

Google lo define como “un desarrollo de internet en el que los objetos cotidianos tienen una red que les permite conectarse para enviar y recibir datos”. Así pues, a continuación podrás adentrarte al fantástico mundo de las cosas conectadas, del refrigerador que te recuerda que hace falta comprar leche, del universo de beacons que sabrán todo de tus necesidades —o no necesariamente— para satisfacerlas casi al momento sin que tengas siquiera que preocuparte por ello.

Gartner estima que para finales de 2016 el gasto anual en servicios de IoT rondará los 235 mil millones de dólares. Prácticamente todas las industrias están explorando soluciones basadas en IoT para recabar tantos datos sean posibles, que les permitan entender mejor a sus clientes y obtener ventaja competitiva. En las siguientes páginas compartimos ideas y reflexiones de expertos sobre lo que implica el internet de las cosas para la tecnología empresarial.

PREPARÁNDONOS PARA LA DISRUPCIÓN DEL IOT

Por Curt Hall

● El internet de las cosas (IoT) promete generar disrupción en todas las industrias. Las empresas requieren examinar cómo pueden aprovechar el contar con productos y servicios conectados, y considerar los nuevos volúmenes de datos que se generarán como consecuencia de productos habilitados con sensores. Sin embargo, este auge en la innovación de productos también implica que las empresas deben considerar cuidadosamente cómo van a lidiar con la posibilidad de nuevos competidores con modelos de negocio basados en productos y servicios de IoT. A continuación comparto algunos puntos que recomiendo considerar.

NUEVAS PRÁCTICAS DE GESTIÓN DE DATOS

La disponibilidad de dispositivos conectados resultará en un aumento significativo en la cantidad de datos que generamos. Las organizaciones deben lidiar con la gestión de estos datos. Además de su gran volumen, los datos generados por aplicaciones IoT son rápidos (real-time, streaming) y variados. Esto impulsará la demanda para personal experimentado en el uso de herramientas y técnicas modernas de análisis de datos.

El IoT ya está teniendo un impacto en la industria de gestión de datos e inteligencia de negocios, impulsando el desarrollo y aplicación de nuevas tecnologías para la gestión y análisis avanzado de datos. Esto incluye tecnologías como Hadoop, Yarn, NoSQL, bases de grafos, bases multimodelo, y las denominadas "NewSQL" que combinan capacidades de RDBMS tradicionales con Hadoop y NoSQL. El IoT está avanzando el desarrollo y adopción de analítica en tiempo real como procesamiento de eventos, motores de reglas de negocio, streaming, machine learning y sistemas cognitivos (ej. IBM Watson).

INNOVACIÓN Y DISRUPCIÓN

Conforme la adopción de IoT tome inercia, se espera que generará grandes innovaciones en productos, que provocarán disrupción en industrias enteras. La innovación acelerada basada en IoT también puede resultar en un aumento en batallas por propiedad intelectual y confrontaciones entre competidores.

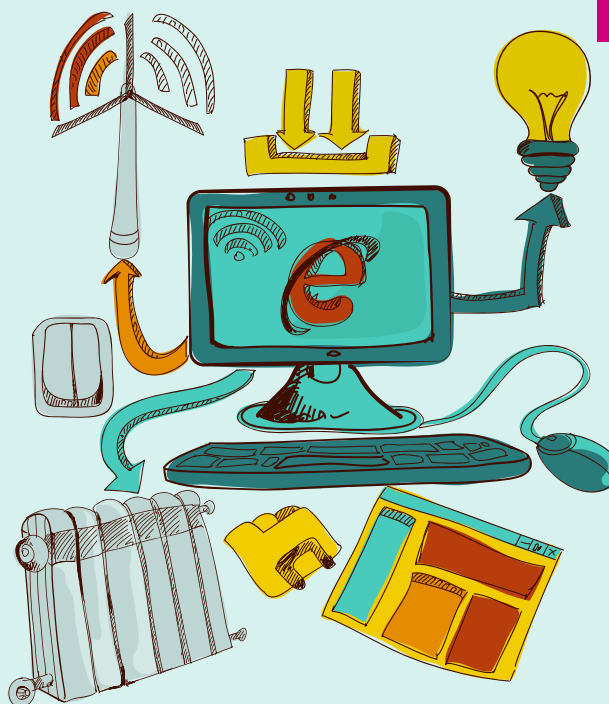
El internet de las cosas también afectará los modelos de servicio al cliente al obligar a las empresas a adoptar prácticas de servicio en tiempo real. Así que las empresas tendrán que implementar capacidades para el monitoreo remoto de productos en tiempo real. En algunas industrias como las de manufactura y transporte ya se están desarrollando productos que soportan mantenimiento predictivo en base a los datos obtenidos por sus sensores.

Algunas industrias experimentarán cambios dramáticos como la de taxis o la de servicios de hospedaje que se han visto impactadas con la llegada de empresas como Uber, Lyft, Airbnb, etc. Otras industrias como la automotriz, seguros y transporte de carga ya están comenzando a lidiar con el impacto de la disponibilidad de automóviles con sensores e incluso que se pueden manejar solos. Por ejemplo, consideremos el posible impacto que los vehículos autónomos podrían tener en la industria de aseguradoras, así como en las tendencias de venta de automóviles. Si

los automóviles se manejan solos, entonces ¿cuál será el incentivo para tener uno propio? Especialmente cuando puedes solicitar un automóvil desde tu teléfono para que llegue a recogerte a donde te encuentres, y no tienes que preocuparte por dónde estacionarte o por el mantenimiento del vehículo. Esto nos lleva a otro punto ... ¿cómo sería la publicidad de un automóvil deportivo autónomo? La publicidad actual tiende a mostrar a conductores experimentando felizmente la emoción de manejar rápido a través de algún camino con curvas y un lindo paisaje.

La industria automotriz no está cruzada de brazos en cuanto a enfrentar las oportunidades y retos del IoT. Uno de los ejemplos más dramáticos es la reciente inversión de General Motors en la empresa Lyft (similar a Uber) por 500 millones de dólares. Como parte de esta inversión, colaborarán para crear una red de vehículos autónomos disponibles bajo demanda. Ambas empresas también están planeando un autoservicio de renta de automóviles.

Otras industrias y empresas no están siendo tan proactivas en su adopción de IoT o al menos en su análisis de cómo puede traer disrupción a su industria. Estas empresas corren el riesgo de ser atacadas por sorpresa con innovaciones basadas en IoT. Incluso la juguetera Mattel se ha comenzado a reinventar y ya desarrolló una muñeca Barbie conectada. ☺



EL IMPACTO REAL DE IOT EN LAS EMPRESAS

Por Jorge Alvarado



● **En el Internet de las Cosas**, los objetos cotidianos son parte de una red que envía y recibe datos de otras “cosas”. Para los consumidores esto significa capacidades como poder ajustar el termostato de su hogar desde el otro lado del mundo. Pero para las empresas, el puede representar nuevas oportunidades para conectarse con sus clientes y socios, así como reunir, almacenar y analizar grandes volúmenes de datos. La gama de posibilidades que el IoT ofrece va en aumento y ahora las empresas de todo el mundo comienzan a andar los caminos destinados a aprovecharlo. Su impacto, sin duda, revolucionará la manera en que las empresas están haciendo negocio y elevará la productividad y eficiencia, similar a lo que sucedió con la llegada de las computadoras.

En su reporte TechRadar, Forrester revela que 23% de las empresas ya utilizan el IoT, mientras que 29% planean hacerlo en los próximos 12 meses. Aprovechan el IoT para transformar sus modelos de negocio, optimizar la utilización de activos físicos y financieros, y crear nuevas formas de relacionarse con sus clientes. De igual modo, Forrester prevé que el IoT creará grandes volúmenes de datos, por lo que la analítica de IoT se convertirá en una categoría y disciplina especializada. Por su parte, Gartner calcula que para cuando finalice 2016 el gasto en servicios de IoT alcanzará los \$235 mil millones de dólares, y que más de la mitad de los nuevos procesos de negocio y sistemas incorporarán algún elemento del IoT para el año 2020.

Mientras los teléfonos móviles, los smart TVs, electrodomésticos y autos inteligentes pueden seguir captando la imaginación de los consumidores, es posible ver que hay mucho que ganar con las implementaciones empresariales. A diferencia de los sensores integrados en los dispositivos de consumo, aquellos integrados en maquinaria, contenedores, trenes, plantas de producción o puntos de venta, por ejemplo, están generando la información para crear aplicaciones en logística, redes inteligentes, servicios públicos, transporte, manufactura y retail.

Pero más allá de los sensores y dispositivos, el valor real del IoT es determinado por los datos y las nuevas aplicaciones que los aprovechan. La inteligencia de

Jorge Alvarado es Jefe de Arquitectura en Red Hat México.

estos datos está dando paso a nuevos modelos de negocio y aplicaciones que nunca habían existido antes y que abren la puerta a la transformación digital.

LOS ALCANCES DEL IOT EMPRESARIAL

A continuación listo algunos ejemplos de aplicaciones empresariales:

Manufactura inteligente. El servicio o función de una máquina o una parte de ella pueden mejorar antes de que se presente una falla, eliminando así los costosos tiempos de inactividad y eventualidades no previstas.

Cadenas inteligentes de suministro. Proporcionando información en tiempo real de la oferta, demanda y envíos a los clientes. Las entregas pueden ser rastreadas y recuperadas si son extraviadas o robadas.

Infraestructuras inteligentes. Las oficinas inteligentes contribuyen a generar ahorros de energía, mejorar la autosustentabilidad y potenciar la colaboración entre los empleados.

DECISIONES EN FRACCIÓN DE SEGUNDOS

¿En qué radica la diferencia entre el IoT de consumo y el empresarial? Los casos de uso de IoT de consumo se crearon sobre una arquitectura en la que el dispositivo se conecta a internet aprovechando la red WiFi del hogar, o puenteando por medio de un teléfono móvil.

En este modelo, el dispositivo transmite todos los datos a un centro de datos en la nube, donde se realiza el análisis y, si se requiere una acción, se envía el mensaje correspondiente al dispositivo para que dispare la acción necesaria. Fundamentalmente, esta arquitectura funciona en este modelo por dos razones: la disponibilidad del ancho de banda y a que no hay que tomar decisiones rápidamente.

En una aplicación de consumo en la que el dispositivo se conecta a la nube por medio del WiFi del hogar, el uso de ancho de banda no tiene un costo adicional para el cliente, ya que típicamente son planes con transferencias de datos ilimitadas. Sin embargo, las conexiones en las empresas tienen más restricciones y cada byte cuenta. De hecho, reducir un solo byte de un mensaje puede ahorrarle a la empresa millones de dólares en costos de transmisión en los casos de uso del IoT industriales. Por lo tanto, se ha vuelto crucial para las arquitecturas de IoT considerar el ancho de banda en sus diseños.

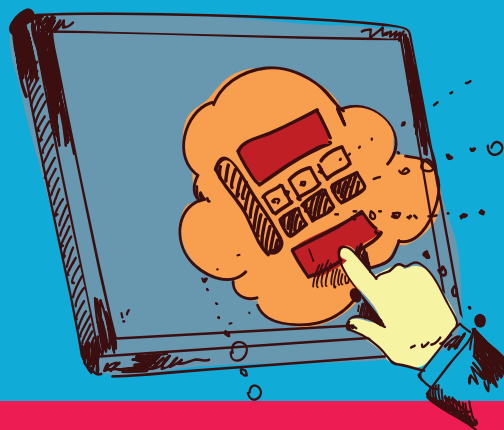
En segundo lugar, para los consumidores típicamente no es crítica la cantidad de tiempo que se necesita para tomar decisiones. En contraste, en un entorno empresarial de misión crítica, las decisiones se miden en fracciones de segundo. Por ejemplo, si el sensor de bajo voltaje de una red eléctrica esperara tres minutos para liberar capacidad adicional cuando el voltaje comienza

a disminuir, todas las redes podrían colapsar y perderse miles de millones de dólares en equipo.

De ahí que las empresas deban ser lo más ágiles posible y reaccionar de inmediato a la información que están recabando en tiempo real. Gartner habla del TI Bimodal, en donde las empresas por un lado tienen sistemas de cómputo tradicionales de alto rendimiento (modo 1), y por otro lado aplicaciones escalables y datos en la nube que requieren de agilidad para adaptarse a un mundo cambiante (modo 2). Típicamente el código abierto contribuye con éste último modo, ayudando a las empresas a innovar, ser más productivas y adoptar nuevas tendencias como la nube, la virtualización y el Internet de las Cosas, al tiempo de ahorrar costos y mejorar la seguridad.

Estas consideraciones nos permiten ver que una arquitectura de dos niveles es muy lenta para los datos importantes, y demasiado costosa para los que no lo son. En el plano empresarial surge una arquitectura de tres niveles creada alrededor de una nueva capa media (o el controlador). Estos controladores son lo suficientemente inteligentes para actuar rápidamente, al tiempo de enviar de vuelta únicamente los datos importantes al centro de datos. Este concepto, conocido como de Near Field Processing, permite que las decisiones se tomen más rápidamente y requiere que una menor cantidad de datos viaje hasta el centro de datos. Actuar ágilmente al final reduce los costos de transmisión y el tiempo para tomar decisiones, lo que le permite hacer del Internet de las Cosas una realidad para la empresa.

En el pasado, solo las compañías con más presupuesto podían beneficiarse de reunir datos de los dispositivos distribuidos para tomar mejores decisiones y obtener valor adicional. Hoy, las economías de la arquitectura de IoT: el hardware, la naturaleza ubicua de la conectividad, el big data y las soluciones de software empresarial en todas las capas, acompañan a la experiencia del usuario para ampliar el IoT, y lo hacen posible para que las empresas, no solo los consumidores, se beneficien. ☞



EL SALVAVIDAS DEL IOT EN LA EVOLUCIÓN DE LA TECNOLOGÍA

Por Esteban Montelongo

● **Actualmente, cuando hablamos del internet de las cosas** ya no se trata de hablar de comunicación entre computadoras, eso solo es parte de redes internas y externas. El internet de las cosas tiene una palabra clave: *things*, que básicamente describe las “trillones de cosas” que no tienen relación con una computadora como tal, pero que emiten datos e información relevante.

Bajo esta perspectiva, resulta indispensable tomar en cuenta la evolución de la información que ahora es posible obtener vía sensores, provenientes por ejemplo de equipos electrónicos.

En la actualidad, una empresa de retail puede calcular la reposición de sus inventarios, así como distribuir sus productos de acuerdo a un detector ubicado dentro de los refrigeradores de los consumidores. Basta con solo abrir la puerta del refrigerador y sacar un determinado producto, para que esto pueda ser identificado por los sensores como un “faltante” para que así esos datos se envíen de manera inmediata al proveedor, garantizando la venta de dichos productos en tiempo y forma, de acuerdo a los patrones de compra específicos de los consumidores y asegurando, incluso, el envío a la puerta del hogar... ¡Fascinante! ¿No?

Dentro de estas “things”, podemos mencionar a los equipos, máquinas y dispositivos que se utilizan dentro de las diferentes industrias tales como los sistemas de medición de temperatura, irrigación, emisión y/o consumo de combustible, señalización o bien, equipos de salud, tanto de uso personal, como equipos más sofisticados, entre muchos otros.

Sin embargo, para no ahogarse con la gran ola de información, el salvavidas vendrá marcado por los datos que resulten realmente valiosos, filtrados y maniobrados vía herramientas de analítica avanzada así como con arquitecturas que la soporten.

Esta oportunidad de facilitar la información en tiempo y forma así como la flexibilidad para crear modelos de predicción, asociación y clasificación, sin duda alguna está contribuyendo al nuevo éxito de las compañías.

Por ejemplo, hoy en día, el monitoreo de fletes vía sensores ya no solo otorga información sobre el seguimiento del paquete, sino que también puede monitorear las condiciones del transporte a través del trayecto, ¿cuánta humedad había en el contenedor?, ¿el paquete fue abierto o inspeccionado? Con las respuestas a la variedad de preguntas, se tendrá un control preciso de los productos durante el trayecto, con lo que las compañías aseguradoras podrán crear nuevos modelos de aseguramiento y también contar con el detalle de las características del siniestro ocurrido, evaluando el pago de una póliza. Por su parte, el área de logística utilizará esta

información para mejorar sus operaciones. Es simple, ahora podemos saber cuándo, dónde y cómo se realizó un determinado evento.

De acuerdo con datos de Gartner, para el año 2020 se calcula que el valor de la economía dentro del IoT sea de 1.9 trillones de dólares, derivada básicamente por las nuevas ofertas dirigidas a un usuario final tales como smart watches, wireless lighting entre otros, así como las orientadas a la operación y automatización de sistemas dentro de una manufacturera, en donde el mayor impacto se verá hasta dentro de algunos años.

La oportunidad para generar nuevas fuentes de ingresos, ventas y reducción de costos, entre otros problemas de negocio, resulta ser sumamente sencilla y simple, incorporando el salvavidas idóneo que sepa sacar provecho de las “cosas” que la evolución de la tecnología nos brinda. SG



Esteban Montelongo es Arquitecto de soluciones para analítica avanzada en Teradata.



EL MAYOR VALOR DE NEGOCIO EN INTERNET DE LAS COSAS SON LOS DATOS

—
Por Samuel dos Reis

● **Mucho del enfoque al hablar del Internet de las Cosas** se centra en “las cosas” mismas, como los wearables, sensores, beacons, y otros equipos conectados a la red. Sin embargo, el mayor valor para las organizaciones proviene de la combinación de datos generados por estos dispositivos con otros datos del cliente u operacionales para descubrir otras ideas y establecer modelos predictivos. Esta es la increíble promesa del IoT, pero sin la capacidad de vincular los datos de las cosas “inteligentes” conectadas con otros datos de negocio, su valor es limitado.

En lugar de proyectos de IoT aislados que no están conectados a la infraestructura corporativa central de datos, hay una oportunidad de usar los datos generados por el IoT para crear valor real de negocio. Las organizaciones pueden combinar datos de los clientes obtenidos de distintos dispositivos con otros datos de los clientes para obtener nuevas perspectivas e identificar nuevas señales sobre el potencial de compra o cambios de comportamiento. Este conocimiento puede llevar a las empresas a un entendimiento más profundo y una mayor capacidad de respuesta a los clientes. Por ejemplo, las tiendas pueden combinar datos obtenidos por beacons dentro de la tienda con el historial de transacciones y almacenar modelos de comportamientos de los clientes para determinar mejores promociones o diversas acciones a tomar, tales como notificar al personal de la tienda que un cliente VIP ha llegado.

También hay enormes oportunidades de poner los datos de IoT a trabajar en modelos predictivos para mejorar horarios de mantenimiento o proporcionar un servicio “justo

a tiempo”, antes de que el producto falle. Por ejemplo, muchos dueños de vehículos eléctricos reciben reportes de diagnósticos remotos con información sobre fallas o próximos servicios a necesitar, basados en interpretación de los datos generados por el vehículo.

El reto que muchas organizaciones enfrentan es cómo entender de forma sistemática los datos derivados de “las cosas” y combinarlos con otros datos relevantes de la empresa para crear valor. Los datos del IoT generalmente consisten en archivos de bitácora, a veces mal relacionados, y al parecer no estructurados; de hecho, los datos del IoT tienen estructura, aunque no están en un formato tradicional relacional o estándar. Los archivos de bitácora se estructuran e incluyen puntos de datos de medición que varían de un fabricante a otro, de un modelo a otro, de una versión de software a otra e, incluso de una compañía a otra. Por ejemplo, los terabytes de datos procedentes de un motor de reacción de un jet, difieren por aerolínea, así como por fabricante y modelo. Como otro ejemplo, productos para la agricultura automatizada y maquinaria industrial, todos con ellos con diferentes modelos y variantes, tienen cientos o miles de distintos formatos de archivos de registro procedentes de sus productos usados por clientes alrededor del mundo. Esto introduce una complejidad significativa en cualquier intento de interpretar los datos, darles formato, normalizarlos y combinarlos con otros datos relevantes para su análisis o para sistemas operacionales. Sin estas herramientas que ayuden a interpretar y analizar los datos del IoT, el tiempo y esfuerzo invertido será costoso, prolongado y propenso a errores manuales.

Con el fin de comprender plenamente el valor de combinar los datos del IoT con otros datos de la empresa de manera ágil, las organizaciones deberán aprovechar las modernas herramientas de gestión de datos para acelerar y automatizar los procesos. Utilizando herramientas para clasificar los datos, descubrir su estructura de forma inteligente, para luego extraerlos automáticamente y combinarlos con otros datos relevantes de la empresa de forma continua, hace que los datos del IoT sean más accesibles. Esto crea una infraestructura para una expansión y evolución continua en el uso de los datos del IoT para entender a los clientes y crear modelos predictivos para los sistemas operativos.

Las empresas pioneras en infraestructura para IoT están creando soluciones innovadoras que combinen modelos interactivos, visuales de los datos con aprendizaje automatizado, a su vez agilizando la capacidad de obtener valor del negocio a través del IoT. Una vez que el modelo de datos para un dispositivo en particular es entendido, mapeado y preparado, la transformación y la entrega de datos a los sistemas puede ser automatizada para un uso productivo.

En una era en la que vivimos en una mayor conectividad de las cosas, es tiempo de capitalizar toda la explosión de información que resulta de ellas, y poner orden a través de una correcta integración, calidad y limpieza de datos para que las empresas tomen mejores decisiones y generen productos y servicios específicos acordes a cada consumidor. ☺



CINCO MANERAS EN QUE EL IOT DARÁ LUGAR A UN MARKETING MÁS INTELIGENTE

Por Eduardo Lugo

● **El Internet de las Cosas** puede sonar como un “buzz word” más, pero en realidad se trata de una revolución tecnológica que tendrá un impacto en todo lo que hagamos.

El IoT es la interconexión entre las cosas que utilizan tecnología de comunicación inalámbrica (cada una con sus propios identificadores) para conectar objetos, lugares, animales o personas al Internet, lo que permite tanto la transmisión directa como el intercambio óptimo de datos.

En esencia, se refiere a los dispositivos de todos los días que son capaces de intercambiar información automáticamente a través de una red. El IoT también tiene un gran impacto en nuestras vidas cotidianas, ya que cambia la manera en que el tráfico, el clima, la contaminación y el ambiente se monitorean y la forma en que los datos se recopilan.

El IoT ya se encuentra entre nosotros con Nike Fuelband, Google Glass, Fitbit y los iWatch de Apple —todos ejemplos populares de nuestras vidas cada vez más conectadas—.

He aquí 5 maneras en que el IoT mejorará el ROI (retorno de la inversión) del marketing:

1. INTERCAMBIO SENCILLO DE DATOS DE VENTAS

Uno de los activos más valiosos de cualquier empresa son sus datos de ventas. La capacidad de acceder a información sobre cómo, dónde y por qué se están vendiendo sus productos le permitirá dirigir sus esfuerzos de marketing a clientes específicos.

Los dispositivos inteligentes capaces de recopilar esos datos y proporcionarlos en tiempo real, sin la necesidad de que

profesionales en TI manejen y monitoreen la interacción, permitirán a las empresas diseñar estrategias de marketing informadas y mejorar el ROI de las ventas futuras.

Más importante aún, sus clientes podrán ofrecer retroalimentación útil al instante. Por lo tanto, si un producto en específico no está cumpliendo las expectativas, podrá enterarse al momento, lo que significa que podrá reducir las pérdidas más pronto que tarde.

2. CRM MÁS INTELIGENTE: ANÁLISIS DEL CLIENTE AL INSTANTE

Cuando se utilice en conjunto con una buena herramienta de administración de relaciones con el cliente (CRM, por sus siglas en inglés), el IoT podrá hacer mucho más que sólo recopilar y organizar los datos del cliente; podrá también analizar de manera eficaz y precisa los datos para

Eduardo Lugo es vicepresidente de Ventas para Latinoamérica de Salesforce.com

brindar resultados accionables sobre su base de consumidores.

Para los comerciantes eso será invaluable, ya que la cadena de mando del comprador es generalmente larga y las decisiones tardan más tiempo en tomarse. Los dispositivos del IoT pueden acelerar ese proceso al ayudarle a conocer dónde se encuentra el cliente en el proceso de compra y, de esa manera, poder aprovechar cada segundo del día para resolver problemas y proporcionar la información correcta que finalmente ayudará a cerrar la venta.

3. LOS DISPOSITIVOS QUE ESTÁN MURIENDO

Uno de los aspectos más prometedores de los productos con tecnología inteligente es su capacidad de realizar su propio mantenimiento y diagnóstico periódicos.

Los automóviles se autodiagnostican desde hace algún tiempo, aunque a través de un método anticuado que emplea señales imprecisas. Gracias al poder del IoT, cada componente es “más inteligente”, de modo que la capacidad de identificar el problema, así como la solución, será mucho más rápida que antes.

En cuanto a los artículos y dispositivos convencionales, por lo general el primer indicio de que algo está mal es cuando el dispositivo repentinamente deja de funcionar. Cuando eso ocurre, no hay mucho que se pueda hacer, además de repararlo o solicitar uno nuevo y esperar a que llegue.

Los dispositivos del IoT podrían eliminar ese tiempo de espera, ya que constantemente monitorean sus propias funciones y, de ser necesario, contactan al soporte técnico. Y en caso de que detecte un problema grande e irreparable, el dispositivo del IoT puede solicitar fácilmente un reemplazo, de modo que cuando deje de funcionar por completo, el nuevo modelo ya habrá llegado y estará listo para utilizarse.

Lo mismo aplica para las actualizaciones. Muchos usuarios posponen la

actualización de sus dispositivos por temor a que ésta resulte defectuosa, tome mucho tiempo en instalarse o simplemente que algo salga mal. Por desgracia, dejar de actualizar el software deja los dispositivos vulnerables a ataques de seguridad o a cuestiones problemáticas conocidas. Los dispositivos del IoT omitirían la intervención del usuario y buscarían, descargarían e instalarían las nuevas actualizaciones por sí solos.

4. REDES SOCIALES PREDECIBLES

Cuando Facebook y Twitter aparecieron por primera vez hace ya varios años, la mayoría de los comerciantes no estaban del todo convencidos de que valiera la pena incursionar en esos nuevos sitios de “redes sociales”. Hoy sabemos cuán equivocados estaban. En la actualidad, el 74% de los comerciantes de marcas afirman que el tráfico en sus sitios ha aumentado considerablemente después de invertir tan sólo 6 horas a la semana en esfuerzos de marketing para redes sociales.

El IoT ya está optimizado para utilizarse en las redes sociales, ya que permite a los dispositivos publicar y compartir de manera automática y preparar el camino para que las nuevas comunidades en línea se centren alrededor de los usuarios de dispositivos específicos.

Los comerciantes que sean capaces de predecir el desarrollo de esas comunidades sociales y enfocar sus esfuerzos en ellas podrán llegar a los clientes potenciales que quizá antes no estaban disponibles. De la misma manera, los dispositivos del IoT, al combinarse con las redes sociales, permitirán a los comerciantes identificar y aprovechar las nuevas tendencias emergentes.

5. IMAGINE UN CTR (PROPORCIÓN DE CLICS) DE 100%

Al unificarse, esos factores apuntan hacia un objetivo final: una publicidad más inteligente y más relevante.


A medida que una cantidad cada vez mayor de nuestros dispositivos y objetos antes desconectados se acondicionen con sensores, y gracias al acceso constante al Internet, la cara de la publicidad cambiará tanto para el comerciante como para el consumidor.

Los comerciantes ya no tendrán que depender de carteles publicitarios ni anuncios emergentes que aparecieron en un sitio web que visitaste el martes, pues la mayoría de los dispositivos del IoT no podrán de ninguna manera procesar, mucho menos desplegar, tan primitivas tácticas.

Como resultado, la era del comercial interrumpido finalmente morirá en el lado del consumidor. En su lugar surgirá un nuevo mundo donde la publicidad será útil y relevante, donde nunca aparecerá un anuncio que no corresponda al 100% con los intereses, los hábitos y las compras anteriores del usuario.

¿Cómo será posible? Un ejemplo sería un foco que se funde en la casa “inteligente”. El hogar conectado no sólo registraría la necesidad de reemplazar el foco, sino que también proporcionaría al dueño de la casa un cupón digital para un foco nuevo directamente en su smartphone. Aún mejor, registraría y comunicaría el número exacto de horas que el foco se ha utilizado, lo que permitiría al dueño saber que el foco está por concluir su ciclo de vida.

Eso no sólo permitirá al cliente ahorrar tiempo al recibir únicamente anuncios relevantes, sino que los comerciantes ya no tendrán que gastar sumas estratosféricas en publicidad irrelevante.

Los comerciantes necesitarán entender por completo a sus consumidores para poder aprovechar las nuevas oportunidades que surjan, pero aquellos que logren realizar esa transición descubrirán que el IoT les brindará la oportunidad de dejar de ser comerciantes para finalmente convertirse en recursos empresariales valiosos. 

LIBERANDO EL POTENCIAL DEL INTERNET DE LAS COSAS

Por José M. Berruecos

● **Es posible que el internet de las cosas (IoT, por sus siglas en inglés)** todavía sea más un deseo que una realidad, rodeada por cierto nivel de escepticismo integrado, pero esto va a cambiar, y las empresas deben estar preparadas. Las principales empresas alemanas de telecomunicaciones, tecnología y fabricación ya formaron la Alianza del Internet de las Cosas (IoTA, por sus siglas en inglés) para adelantarse a la llegada inminente del IoT. El objetivo es asegurarse de que, cuando los dispositivos empiecen a comunicarse entre sí, lo hagan en alemán. Sin embargo, incluso con toda la planificación, se divisa un bloque molesto en el horizonte: la falta de infraestructura adecuada para administrar la enorme cantidad de datos diversificados, el tráfico, el almacenamiento y las demandas de procesamiento asociadas con el gigante de la IoT.

A medida que el IoT se expande, también aumentan la cantidad de cosas que se conectan a Internet, desde termostatos hasta autos, y la necesidad de tener una infraestructura sólida y poder de procesamiento requerido para respaldarla. Así es como se desencadenaron revoluciones en los centros de datos de todo el mundo.

HERVIDEROS DE INNOVACIÓN

El hecho de que los controladores de IoT sean cada vez más pequeños y accesibles hace que sea relativamente fácil habilitar la conectividad de la IoT en todo tipo de dispositivos. No obstante, no solo los refrigeradores y las lámparas pasan a estar en línea, sino que IoT se está usando a escala masiva, lo que posibilita una amplia variedad de innovaciones.

Por ejemplo, hay científicos que están colaborando con vinicultores innovadores en el Valle de Napa para modificar las anticuadas técnicas de irrigación por medio de sensores inteligentes incorporados en las vides [1]. Con la simple transmisión inalámbrica de las lecturas de los sensores, pueden analizar los resultados para determinar en nivel de hidratación de la planta, lo que da a los

viñedos rezagados un rendimiento óptimo usando menos agua en una época en la que California se encuentra en una sequía catastrófica ampliamente conocida. Los servicios de salud son un sector que aprovechó rápidamente los beneficios proporcionados por la IoT. Una de estas soluciones es el servicio de administración de medicamentos de Philips [2]. Diseñado para personas mayores que a menudo se olvidan de tomar su medicación, el dispositivo le indica al paciente cuando es la hora de tomar las píldoras. Cuando el paciente presiona el botón, se proporcionan vasos llenados previamente con el medicamento. El dispositivo está sincronizado con la línea de teléfono del usuario, por lo tanto, los mensajes se rastrean si la persona no toma una dosis, si es momento de reabastecer esta o si hay una interrupción en la alimentación, que impide el funcionamiento del dispositivo.

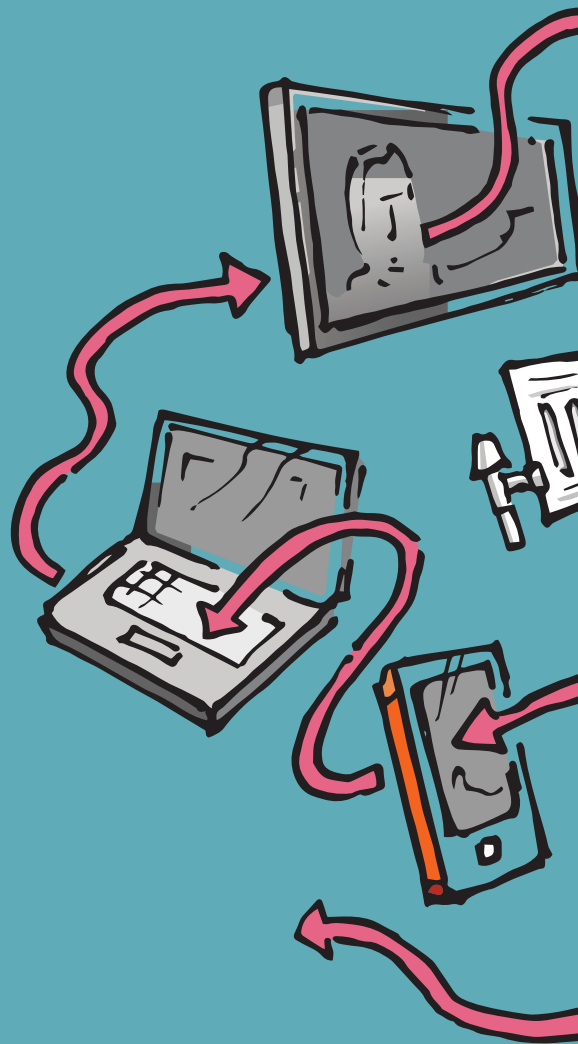
El pronóstico de Gartner [3] es que la cantidad de dispositivos IoT aumentará a aproximadamente 25,000 millones de unidades en 2020. En comparación, se espera que la cantidad total de computadoras, tabletas y teléfonos inteligentes en 2020 sea de solo 7,300 millones de unidades. BI Intelligence [4] espera que este año sea un punto de inflexión en el que la cantidad de dispositivos de la IoT supere la cantidad total de computadoras, tabletas y teléfonos inteligentes por primera vez. Para el final de la década, los dispositivos IoT producirán 40,000 exabytes (40 billones de gigabytes) de datos que deben almacenarse y transformarse en inteligencia significativa.

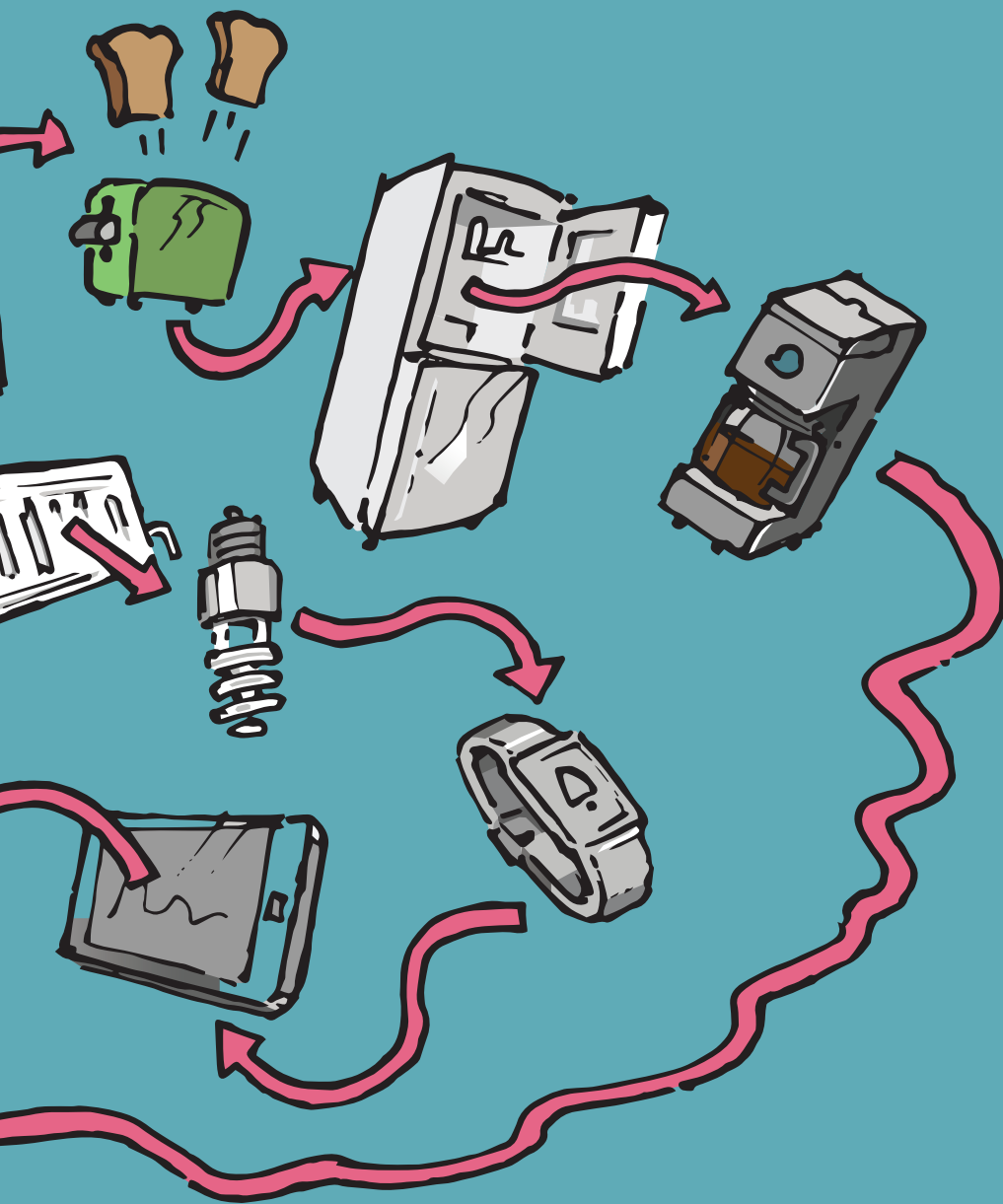
RELACIÓN SEÑAL/RUIDO

El índice de adopción de la IoT dependerá mucho de la facilidad de las empresas para evadir la ley alemana Bundesdatenschutzgesetz (BDSG) y desbloquear y usar la información recopilada, con el elemento clave, ya sea de dispositivos dirigidos a empresas o consumidores, con la aplicación y no el sensor. Con los miles de millones de dispositivos conectados

a la red proyectados para aparecer en los próximos años a medida que la tendencia de la IoT empiece a triunfar, los operadores de centros de datos deberán invertir para asegurarse de que sus instalaciones tengan la capacidad de datos suficiente para adaptarse a la afluencia de información que generarán las terminales de la IoT.

El problema es que, en general, el big data generado por la explosión de la IoT tiene una relación ruido/señal muy alta con una gran proporción de datos irrelevantes, que provocan una tasa alta de emisión de datos. El resultado será un impacto inmediato en el ancho de banda necesario desde el centro de datos y hacia este, y en el almacenamiento y el poder de procesamiento dentro de este.





Debido a esto, para que el IoT llegue al tipo de crecimiento proyectado por Gartner y otros, las empresas deberán tener una funcionalidad de centro de datos que pueda manejar esta posible afluencia inmensa de datos. Las inversiones en servicios de plataforma de IoT que residan en el centro de datos serán esenciales para cumplir con la promesa de la IoT de conectividad y contexto en cualquier momento, en cualquier lugar y de cualquier forma.

Es más, IDC advierte que es posible que la IoT no pueda proporcionar el potencial prometido sin que se realicen inversiones urgentes para actualizar los centros de datos en su investigación [5] sobre operaciones y exigencias del centro de datos. El grupo de analistas predice que la

capacidad del centro de datos consumida por las cargas de trabajo de la IoT aumentará aproximadamente un 750 % entre 2014 y 2019, lo que presionará a las instalaciones desde el punto de vista de redes, almacenamiento y analítica.

UN CAMBIO EN LOS REQUISITOS DE INFRAESTRUCTURA

Debido al número de dispositivos conectados y a la cantidad de datos generados, las empresas deben concentrarse en los requisitos de sus plataformas de servicio de IoT a nivel del centro de datos, no solo de los servidores individuales o dispositivos de almacenamiento. Y, con la proyección de IDC que dice que más del 90 % de los datos creados por dispositivos IoT estarán alojados en

la nube en los próximos cinco años, los centros de datos son los que sentirán la presión.

No obstante, las empresas que se esfuerzan por analizar y comprender los datos generados por la gran cantidad de dispositivos conectados a la red en tiempo real podrían llegar a la conclusión de que los antiguos centros de datos tradicionales no están preparados para esto. En lugar de tratar de cubrir las brechas con la compra de infraestructura según componente, veremos empresas conectadas y modernas que pasan a una infraestructura virtual convergente a fin de ofrecer flexibilidad para los grandes cambios futuros en los datos. Esto llevará al surgimiento de centros de datos de hiperscala construidos sobre infraestructuras convergentes e hiperconvergentes para poder adaptarse a las presiones específicas de la era de la IoT.

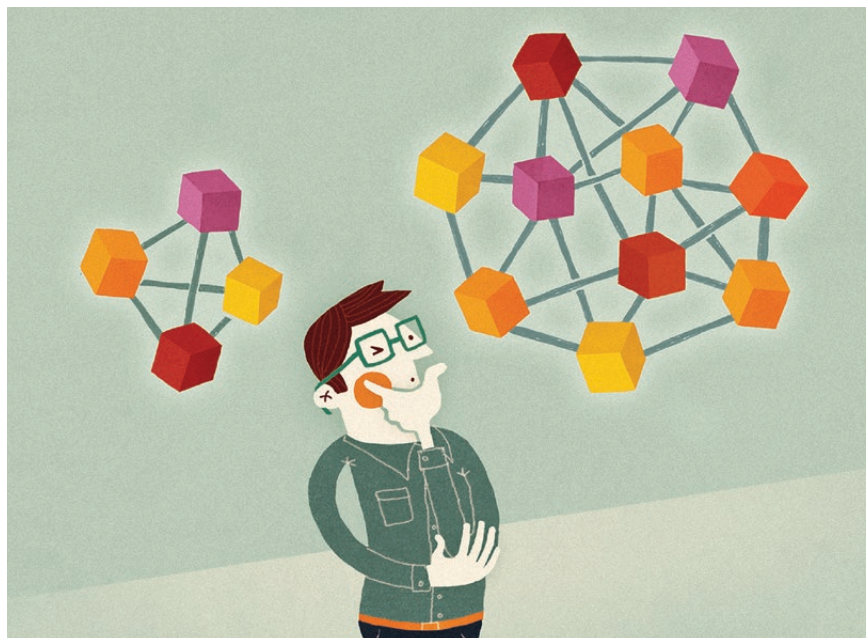
CAMBIO DE LAS PERCEPCIONES DE TI

Para permitir que esto suceda, TI debe cambiar su percepción de un centro de costo a un activador de ingresos. Solo con este cambio de mentalidad una organización podrá aprovechar las ventajas de la IoT. Una vez que la tecnología sea central, las empresas podrán lograr los objetivos de integración, velocidad, escalabilidad y resistencia de una organización transformada digitalmente.

Es posible que haya comenzado con un refrigerador conectado a Internet, pero el IoT sin duda se convertirá en el motivador principal de la expansión de TI a centros de datos más grandes en el futuro, lo que hará más rápida la transición a infraestructuras convergentes dirigidas a la nube que tienen la capacidad para escalar y a arquitecturas de plataforma de datos. ^{SG}

Referencias

- [1] <http://www.wired.com/2012/10/mf-fruition-sciences-winemakers>
- [2] <http://www.managemypills.com/content>
- [3] <http://www.gartner.com/newsroom/id/2905717>
- [4] <https://intelligence.businessinsider.com>
- [5] <http://www.idc.com/getdoc.jsp?containerId=255397>



UX También es Pensar en Productos

Por Misael León

● **En general cuando pensamos en UX** pensamos en hermosas interfaces fáciles de utilizar y amigables con los usuarios. Pero si ahondamos un poco más nos daremos cuenta que el diseño de interacción y la interfaz (UI) son solamente representaciones visibles de *features*; las cuales a su vez, conforman una pequeña parte de una estrategia de producto con objetivos bien específicos de negocio.

En los siguientes párrafos trataré de hacer un caso para demostrar que cuando hablamos de UX en realidad estamos hablando de productos.

Dado que por más subjetivo que sea su campo de acción, es decir “el diseño de la experiencia”, no significa que los esfuerzos de UX no puedan ser medidos en función de resultados y métricas financieras.

Esto es así porque la estrategia de UX vive dentro de la estrategia general de producto y mercadeo. No puede existir de manera aislada.

PIENSA EN PROBLEMAS NO EN FEATURES

Pensemos ahora en ese producto que estamos diseñando, su razón de existir debe ser siempre la resolución de un problema que los usuarios experimentan. Al hacer esto el producto provee valor a las personas y genera ganancias para el negocio. De ahí que UX sea una

parte fundamental para el éxito de los proyectos de software, porque facilita a través de la investigación de usuarios ese descubrimiento del producto ideal que debemos poner en el mercado.

A menudo se dice que los usuarios no saben lo que quieren hasta que lo ven, creo que eso es verdad. Pero ello no significa que un producto no exista en función de una necesidad percibida por el usuario.

Integrar actividades de UX dentro de la estrategia de producto minimiza el riesgo de construir algo que la gente no utilizará. Ya sea porque no lo entiende, porque no refleja su estilo de vida, o simplemente porque no resuelve ningún problema real. El tiempo se encarga de depurar el mercado de software de productos intrascendentes.

Un equipo de desarrollo que construye productos sin entender antes el problema siempre producirá soluciones prematuras. Producirá *features* robustas que no logran enganchar a las personas. La caída de esos productos será inminente. Serán esfuerzos que no rindieron frutos.

CONOCE A TU AUDIENCIA

Pensar en productos no es pensar en *features*, es pensar en el problema que estamos resolviendo. Para esto debemos definir

Misael León (@misaello) es UX Design Researcher en Nearsoft, Inc. una empresa de cultura democrática que desarrolla software y produce clientes felices. Le apasiona difundir las mejores prácticas de UX en comunidades de diseñadores y desarrolladores.

quién es nuestra audiencia. Entender profundamente para qué la gente está usando nuestro producto. Saber cómo viven, cómo piensan, comprender su contexto cultural y el entorno en el cual usan nuestro producto.

Definir *UX Personas* es siempre una buena idea para alinear el entendimiento del equipo. Hacer entrevistas con usuarios, salir a la calle a observar nuestra audiencia, y realizar talleres para generar este conocimiento son las metodologías que más me han funcionado.

La investigación de usuarios no brinda certeza completa pero es el mejor punto de partida. Es una herramienta efectiva para reducir la incertidumbre y elegir un camino a seguir. Nos permite descubrir problemas con potencial de generar negocio y la dimensión real de cómo estos se presentan. Lo que buscamos es conocer nuestra audiencia para no producir numerosas features sin sentido.

DEFINE LA ESTRATEGIA DE PRODUCTO

Existe una concepción equivocada y generalizada que nos hace creer que la responsabilidad de la definición del producto recae únicamente en el product manager; pero de ninguna manera es así. Lo ideal es que desarrolladores, diseñadores, mercadólogos también participen. Todos tienen un pedazo del rompecabezas que hay que resolver. El clan unido con un solo objetivo: construir las features ideales para la audiencia correcta.

Una estrategia de producto se construye contestando las siguientes preguntas [1]:

DIMENSIÓN ORIENTADA A USUARIOS:

¿Cuál es el problema que tiene que resolver?

¿Cuál es la audiencia para la cual se va a resolver el problema?

DIMENSIÓN ORIENTADA AL TRABAJO A REALIZAR:

¿Cuál es la visión detrás del trabajo que haremos?

¿Cómo lo construiremos?

DIMENSIÓN ORIENTADA A RESULTADOS:

¿Qué objetivos nos estamos fijando? ¿Cómo mediremos los resultados que lograremos?

¿Qué features incluye esto que construiremos? ¿Qué vamos a hacer para alcanzar nuestros objetivos?

Con este mapa inicial podemos descartar features innecesarias y comenzar a probar hipótesis sobre lo que creemos es lo idóneo para nuestro producto. Al final nos permitirá crear mejores productos guiando cada decisión que tomemos dentro del contexto de la resolución del problema que nuestra audiencia experimenta.

Recordemos que este problema que estamos resolviendo es la razón fundamental por la cual nuestro producto es usado y por el cual nuestro negocio subsiste.

EL EQUIPO DE PRODUCTO IDEAL

Un equipo de producto idealmente está compuesto por un product manager, un UX designer, y un equipo de entre seis y ocho desarrolladores, incluyendo un QA tester. Dependiendo del tamaño del proyecto también puede incluir un visual designer.

He aprendido que la mejor manera en la que un UX designer puede influir en las decisiones de negocio relativas a producto, es trabajar directamente con el product manager en cada paso del proceso. Uno posee las habilidades que el otro carece, juntos completan el cúmulo de información sobre lo que es deseable construir.

Sin embargo son los desarrolladores los únicos que tienen el conocimiento sobre lo que es posible construir en términos del tiempo disponible. La forma más efectiva de generar ideas de producto es exponer a los desarrolladores directamente al problema. Cara a cara con los usuarios. Esto con el fin de propiciar empatía hacia ellos.

Cuando juntamos “lo que se necesita” con lo que “es posible construir” es cuando surgen los productos realmente valiosos.

CUANDO HABLAMOS DE UX EN EL FONDO HABLAMOS DE PRODUCTOS

Adoptar un enfoque de producto es pensar en soluciones a problemas de gente real. Permite que todo el equipo tenga una visión clara sobre lo que se está construyendo. También facilita descartar peticiones de features nuevas cuando no están alineadas a dicha visión.

Todas las actividades de UX van encaminadas en facilitar la interacción del usuario con el producto de software [2]. Si la interacción es exitosa significa que el UX designer participó desde su conceptualización hasta su implementación, validación y subsecuentes iteraciones.

La responsabilidad de un UX designer no es únicamente la de generar ideas visuales. Es la de lograr que todo el equipo se sensibilice con el problema que se está resolviendo en función de los matices de comportamiento humano que implica.

Esa sensibilización al final del día tomará forma de un producto que deberá ser deseable, técnicamente posible y financieramente sustentable. Es por eso que cuando hablamos de UX en el fondo estamos hablando de productos.

Creo firmemente que identificar problemas sociales reales y brindar una solución centrada en las personas, es una forma honesta de dejar el mundo en mejores condiciones de como lo encontramos. Un producto de software a la vez. ☺

Referencias

- [1] N. Kellingley. “Product Thinking is Problem Solving.” Interaction Design Foundation. <https://www.interaction-design.org/literature/article/product-thinking-is-problem-solving>
- [2] M. León. “Cómo Iniciar tu Carrera en UX Design”. Software Guru. <http://sg.com.mx/revista/50/como-iniciar-tu-carrera-ux-design>

Los Niveles de Granularidad del Requerimiento Funcional

Por Guilherme Simões y Carlos Vázquez

● **Como la mayoría de los lectores de SG saben**, todo software posee dos dimensiones de requerimientos: funcional y no funcional. Los requerimientos funcionales (RF) describen el comportamiento del software para proporcionar tareas y servicios a sus usuarios; estos tipos de requerimientos están relacionados a lo que el software debe hacer. Los requerimientos no funcionales (RNF) están relacionados al cómo las funcionalidades del software serán entregadas a sus usuarios y pueden incluir aspectos de calidad, técnicos, ambientales y organizacionales.

Teniendo en cuenta un sistema de cajero automático de un banco se podría suponer los siguientes ejemplos para los dos tipos de requerimientos:

- RF: Consultar saldo de la cuenta corriente (el qué).
- RNF: Toda transacción debe ser completada en un máximo de 30 segundos (el cómo).

Aunque no sea una regla, el RF se acostumbra manifestar de manera explícita en el software, es un comportamiento que se puede determinar claramente en qué parte del software está. El RNF, por lo contrario, se acostumbra manifestar de manera general, se aplica a todo o casi todo el software.

Para la Ingeniería de Requerimientos los dos tipos son igualmente relevantes: ambos deben ser identificados, analizados, especificados, modelados y gestionados. Sin embargo, hay algunas diferencias entre estos dos tipos de requerimientos que valen la pena comentar por separado. Este artículo se enfocará en los RF, con el objetivo de presentar sus niveles de granularidad y su importancia en el trabajo de la Ingeniería de Requerimientos.

LOS NIVELES DE GRANULARIDAD

Continuando con el ejemplo del sistema de cajero automático, las siguientes descripciones podrían estar presentes en su especificación de requerimientos:

1. Realizar transacciones con la cuenta corriente.
2. Transferir una cifra de una cuenta corriente para otra cuenta corriente.

3. Validar la tarjeta y la contraseña del cliente.

4. Garantizar que la suma de todas las transacciones del cliente en el día no puede ser superior a \$5,000.

Aunque estos cuatro ejemplos sean casos válidos de RF, es posible percibir que el nivel de detalle (o granularidad) entre ellos es distinto. Lo más frecuente es que una especificación presente los requerimientos en distintos niveles de granularidad.

El nivel de granularidad es la mayor o menor amplitud de la descripción del comportamiento esperado para el software en una especificación funcional. Esto está relacionado al tipo de objetivo asociado al requerimiento. La Figura 1 ilustra la relación entre estos objetivos y el nivel de granularidad, utilizando una clasificación de tres niveles (agregado, usuario, subfunción) propuesta por Cockburn (2000) para casos de uso y generalizada por los autores para requerimientos funcionales. Esta clasificación no significa que el proceso de la ingeniería de requerimientos deba utilizar una estrategia de desarrollo por descomposición funcional.

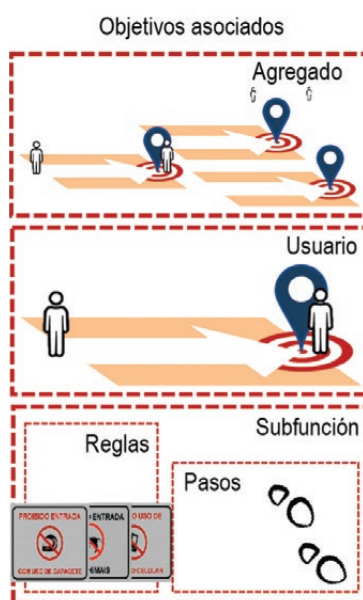


Figura 1.
Niveles de granularidad de los RF.

Guilherme Siqueira Simões (guilherme.simoese@fattocs.com) y Carlos Eduardo Vazquez (carlos.vazquez@fattocs.com) son autores de dos libros en portugués: uno sobre el Análisis de Puntos de Función y otro sobre la Ingeniería de Requerimientos. Ambos tienen más de 20 años de experiencia con desarrollo de software y son socios de Fatto Consultoria y Sistemas (www.fattocs.com), que oferta servicios de consultoría y entrenamiento en las disciplinas de medición, estimación y requerimientos de software.

NIVEL DE USUARIO

El requerimiento a nivel de objetivo de usuario es aquel que:

- Abarca una única tarea bajo la responsabilidad de un único individuo;
- Es realizado en el momento en que el usuario posee todo lo que es necesario para que la tarea sea completada hasta el límite de su responsabilidad en el flujo operativo donde está contenida.

Al final de la tarea el usuario alcanza su propósito, está satisfecho y no hay nada más que necesite hacer. Una vez que el requerimiento fue completado, todo lo que debería de realizarse para tratar un evento externo fue realizado. Esta tarea es casi siempre parte de un proceso de negocio que puede tener un flujo operativo más amplio y por esto puede aún no estar completa. Sin embargo, la perspectiva relevante en este caso no es del proceso de negocio, es de la tarea.

En general, si un trabajo involucra más de un individuo es porque hay más de una tarea presente en el contexto. Hay excepciones como un retiro en la cuenta corriente en la sucursal del banco donde dos individuos participan: la caja del banco que solicita e informa datos para el retiro y el cliente que informa la contraseña.

Esta es la granularidad del ejemplo 2 (“Transferir una cifra de una cuenta corriente para otra cuenta corriente.”). Es una única tarea (seguramente compuesta por varios pasos y reglas), bajo la responsabilidad de un único individuo que al final de todos los pasos está satisfecho con el objetivo alcanzado: una cifra fue transferida a otra cuenta corriente.

NIVEL AGREGADO

El requerimiento en este nivel agrega varios requerimientos a nivel de objetivo de usuario en una única especificación de más alto nivel. Tanto más alto sea su nivel, más generales son sus objetivos y para que un objetivo de nivel más alto sea alcanzado, otros objetivos de nivel más bajo deben ser alcanzados primero.

Este tipo de requerimiento está relacionado a objetivos más generales y su amplitud está asociada a objetivos colaborativos o asociados a procesos de negocio de alto nivel. No es relativo a una única tarea, por lo contrario, agrega un conjunto de tareas de uno o más usuarios. Esta es la granularidad del ejemplo 1 (“Realizar transacciones con la cuenta corriente.”).

¿Cuáles son las tareas asociadas a este tipo de requerimiento? Tal vez sean obvias para los lectores de la especificación (los interesados) o tal vez aún no sean conocidas. Sin embargo, es sabido que hay actividades a realizar para levantar (refinar) este requerimiento. Lo que importa es que ya está claro que este requerimiento está dentro del alcance del software a ser desarrollado.

Sin embargo, algunos RF de este tipo poseen patrones que eliminan la necesidad de proporcionar más detalles. Un ejemplo clásico son los formularios CRUD (Create, Read, Update, Delete) para que el usuario pueda mantener datos por medio del software. Es muy común que esto sea expresado como: “El sistema debe mantener productos.”. Hay un conocimiento tácito entre los interesados que el verbo “mantener” agrega las tareas del CRUD. Por lo tanto queda claro a todos los lectores que el software debe permitir al usuario realizar las siguientes tareas: agregar, modificar, eliminar y consultar datos de producto.

NIVEL DE SUBFUNCIÓN

Estos requerimientos son pedazos de requerimientos con objetivo de usuario; están relacionados a un conjunto de pasos o a reglas de una o más tareas de los usuarios.

El requerimiento en nivel de subfunción que representa un conjunto de pasos describe el cambio de datos en los dos sentidos entre el usuario y el software; y entre el software y los requerimientos de almacenamiento. Este es el caso del ejemplo 3 (“Validar la tarjeta y la contraseña del cliente.”). Cada tipo de transacción que utiliza la cuenta corriente (ej.: retiro, transferencia, pago, etc.) exige el mismo conjunto de pasos descrito por el ejemplo 3, que se podría suponer como:

- Verificar si la tarjeta es válida.
- Verificar si la transacción elegida es compatible con el tipo de tarjeta.
- Verificar si la contraseña informada es correcta.
- Incrementar el control de errores de contraseña en caso que la contraseña informada sea incorrecta.
- Cambiar a cero el control de errores de contraseña en caso que la contraseña informada sea correcta.

Validar la tarjeta y contraseña del cliente no es el objetivo del cliente al utilizar el sistema de cajero automático, sin embargo son pasos necesarios y intermediarios para alcanzar su objetivo: por ejemplo, hacer un retiro.

El requerimiento en el nivel de subfunción que está relacionado a reglas, en general se vincula a las leyes que gobiernan el negocio y describen de manera que complementan los procesos de negocio. También llamadas muchas veces como reglas de negocio. La regla puede describir políticas corporativas, reglamentos gubernamentales o estándares de la industria por los cuales el software debe estar subordinado.

Este es el caso del ejemplo 4 (“Garantizar que la suma de todas las transacciones del cliente en el día no puede ser superior a \$5,000.”). Las reglas de negocio muchas veces son compartidas entre varios RF, hasta entre distintos productos de software en la empresa.

IMPORTANCIA DE LA GRANULARIDAD

En una visión simplificada se puede decir que hay dos momentos clave donde una especificación de requerimientos es necesaria:

- A. Proporcionar una visión amplia del software (y tal vez aún no profunda).
- B. Proporcionar una visión profunda del software (de una parte o del todo).

El caso A es muy común en etapas tempranas de proyectos. El objetivo en este momento es primero obtener una visión amplia del alcance, por ejemplo: para planificar el proyecto, estimar un orden de magnitud de costo o plazo, o hacer un análisis de factibilidad. Un documento de visión es un ejemplo de documento que cumple este propósito. Especificar los RF en el nivel agregado es suficiente y más adecuado a este caso. El costo para obtener información y especificar en un nivel de granularidad más bajo puede significar desperdicio de esfuerzo para detallar lo que no es necesario.

Entonces el documento de requerimientos para el propósito A, tendrá la mayoría de los RF especificados en el nivel agregado. Es posible también que haya especificaciones en el nivel de usuario o subfunción, esto no es un error. A veces puede ser interesante especificar los RF más críticos (no todos o la mayoría) en un nivel más detallado, ya sea para facilitar la comprensión del alcance por el interesado o permitir una estimación con menos incertidumbre.

El principal propósito para especificar requerimientos en niveles agregados es establecer áreas de proceso objeto de informatización y/o automatización para que las necesidades de negocio queden resueltas.

Si el objetivo de la especificación es proporcionar una visión profunda de parte o de todo el software, por ejemplo para iniciar su desarrollo, la existencia de RF en el nivel agregado puede significar que varias decisiones sobre el alcance todavía siguen pendientes. Es decir, hay trabajo de levantamiento que debe ser hecho.

Para el propósito B, el nivel de granularidad más adecuado es del objetivo del usuario pues es el único que proporciona una definición de alcance del software de manera inequívoca; no hay dudas sobre lo que es abarcado. Esto es también el único nivel de descripción de procesos que puede ser estandarizado y consecuentemente es el nivel utilizado por todos los métodos de medición de tamaño funcional adherentes a la norma ISO/IEC 14143.

Consecuentemente, para el propósito B, lo más común es que un documento de requerimientos tenga los RF especificados en su mayoría con el objetivo de usuario. RF en el nivel agregado puede estar presente sólo cuando es el caso de que esté claro a todos los interesados que tareas son abarcadas por este RF.

La definición de un requerimiento en el nivel de subfunción es adecuada cuando el comportamiento especificado es común y compartido por varias otras tareas que el software entrega a los usuarios. Esto hace con que la especificación de requerimientos sea más fácil de modificar en respuesta a cambios, pues disminuye la redundancia de información al evitar describir el mismo conjunto de pasos o reglas más de una vez. También ayuda a alcanzar una especificación más consistente.

CONCLUSIÓN

Aunque el concepto de los niveles de granularidad del requerimiento funcional sea sencillo, en términos prácticos se percibe que muchos ingenieros de requerimientos no están atentos a esto cuando elaboran las especificaciones. Tener en cuenta los niveles de granularidad ayuda a definir el esfuerzo adecuado para ser dedicado a la elaboración de la especificación y también a alcanzar una especificación de requerimientos de mejor calidad.

El estándar 830 de la IEEE presenta los siguientes criterios para evaluar la calidad de una especificación de requerimientos: correcta, completa, clara (no ambigua), consistente, modificable, priorizada, verificable, trazable.

Siendo el propósito de la especificación proporcionar una visión profunda del alcance, estar atento al nivel de granularidad del RF permite que:

- Se descubra que la especificación puede no estar completa cuando hay RF agregados.
- Se proporcione la comprensión del alcance sin ninguna ambigüedad con respeto a las tareas que el software entregará cuando se enfoca el RF con objetivo de usuario.
- Se genere un documento de requerimientos más fácil de mantener y consecuentemente más consistente al especificar los RF adecuados con objetivo de subfunción. ☺

Referencias

- [1] A. Cockburn. *Writing Effective Use Cases*. Addison-Wesley, 2000.
- [2] COSMIC Measurement Manual: 4.0.1. *Common Software Measurement International Consortium*, 2015.
- [3] IEEE Std. 830-1998: *IEEE Recommended Practice for Software Requirements Specifications*. IEEE Computer Society, 1998.
- [4] *Function Point Counting Practices Manual*. Release 4.3.1. IFPUG, 2010.
- [5] C. Vazquez, G. Simões & R. Albert. *Análise de Pontos de Função, Medição, Estimativas e Gerenciamento de Projetos de Software*. Saraiva, 2013.
- [6] C. Vazquez, G. Simões. *Engenharia de Requisitos: Software Orientado ao Negócio*. Brasport, 2016.



Donde se encuentran los profesionistas que crean software grandioso



BENEFICIOS DE TENER TU PERFIL EN SG TALENTO:

- Muestra tu experiencia y habilidades en tu perfil público
- Contribuye a generar información estadística sobre el talento de TI en nuestra región.
- Si estas buscando oportunidades laborales, podrás contactar con reclutadores y aplicar a las vacantes disponibles.
- Además podrás participar en promociones especiales organizadas por SG: descuentos, rifas, concursos, entre otros.

www.sgtalento.com

JavaScript y Herencia Basada en Prototipos

Por Erik Grueter

● **Para los programadores** que vienen de lenguajes basados en clases, como Java o Ruby, JavaScript puede ser un poco confuso.

A diferencia de estos lenguajes, Javascript es dinámico y no provee una implementación de clase como tal. Para soportar herencia, JavaScript solo tiene un constructo: Object. Cada objeto tiene una referencia a otro objeto, llamado su prototipo.

Aunque esto pueda parecer una debilidad, en realidad este modelo de herencia basada en prototipos es muy poderoso. En este artículo explicaré como funciona la herencia en JavaScript y la cadena de prototipos.

HERENCIA DE PROPIEDADES

Los objetos en JavaScript no son más que estructuras de datos con varias propiedades. Cada objeto apunta tiene una referencia a su objeto padre, o prototipo.

Al intentar acceder una propiedad en un objeto, la propiedad se busca primero en el objeto, y si no se encuentra se procede a buscar la propiedad en el prototipo del objeto original, y si no se encuentra se busca en el prototipo del prototipo, y así sucesivamente hasta que se encuentre la propiedad o se llegue al final de la cadena.

El listado 1 muestra un ejemplo de este caso.

```
// Asumamos que tenemos un objeto o, con propiedades a y b:
// {a: 1, b: 2}
// o.[[Prototype]] tiene las propiedades b y c:
// {b: 3, c: 4}
// Finalmente, o.[[Prototype]].[[Prototype]] es null.
// Así que la cadena de prototipos queda como
// {a:1, b:2} → {b:3, c:4} → null

console.log(o.a); // 1
// Existe una propiedad 'a' en o? Sí, y su valor es 1.

console.log(o.b); // 2
// Existe una propiedad 'b' en o? Sí, y su valor es 2.
// El prototipo de o también tiene una propiedad 'b' pero no es visitada.
// Esto se conoce como "property shadowing"

console.log(o.c); // 4
// Existe una propiedad 'c' en o? No. Seguir la cadena de prototipo.
// Existe una propiedad 'c' en o.[[Prototype]]? Sí, y su valor es 4.

console.log(o.d); // undefined
// Existe una propiedad 'd' en o? No. Seguir la cadena de prototipo.
// Existe una propiedad 'd' en o.[[Prototype]]? No. Seguir la cadena de prototipo.
// o.[[Prototype]].[[Prototype]] es null, dejar de buscar.
// No se encontró la propiedad 'd' en la cadena, regresar undefined
```

Listado 1. Herencia de propiedades.

HERENCIA DE “MÉTODOS”

En realidad, Javascript no tiene métodos como los de los lenguajes de programación basados en clases. En JavaScript, cualquier función se puede agregar a un objeto como si fuera una propiedad. Estos objetos, y sus respectivas propiedades se asemejan a los pares llave-valor de un Hash.

El listado 2 muestra un ejemplo de una función como propiedad.

```
var o = {
  a: 2,
  m: function(b){
    return this.a + 1;
  }
};
```

Listado 2. Función como propiedad.

Como podemos ver, el objeto o tiene dos propiedades: a con el valor 2 y m con valor function(b), que regresa el valor de la propiedad a + 1, que en este caso equivaldría a 3.

Ahora vamos a extender el código anterior para mostrar cómo funciona la herencia basada en prototipos cuando la combinamos con funciones.

```
var o = {
  a: 2,
  m: function(b){
    return this.a + 1;
  }
};

console.log(o.m()); // 3
// 'this' se refiere a 'o' cuya propiedad 'a' vale 2,
// así que m() regresa 3.

var p = Object.create(o);
// 'p' es un objeto que hereda de 'o'

p.a = 12; // crea en 'p' la propiedad 'a' con valor 12
console.log(p.m()); // 13
// 'p' no define propiedad m() así que se aplica la del prototipo
// pero ahora 'this' se refiere a 'p' cuya propiedad 'a' vale 12,
// así que m() regresa 13.
```

Listado 3. Ejemplo extendido.

Como puedes ver, el valor de 'this' apunta al objeto heredado, a partir de ahí busca las propiedades. Como en el objeto p se encuentra que a = 12, no es necesario seguir recorriendo la cadena de herencia.

Ahora que entendemos cómo funciona la herencia basada en prototipos, consideremos cómo funcionan los arreglos en JavaScript. Consideremos el siguiente arreglo:

```
var a = ["hola", "qué onda", "?"]
```

Todos los arreglos en JavaScript heredan de Array.prototype. Es a este nivel donde están definidos métodos como indexOf, forEach, etcétera. La cadena de prototipo es a -> Array.prototype -> Object.prototype -> null.

CLASES EN ES6

Posiblemente estés enterado de que una de las novedades en EcmaScript 6 es que soporta clases y te preguntes cómo afecta a lo que hemos visto aquí. En realidad, lo único que se ha agregado en ES6 es que se soporta sintaxis similar a la de otros lenguajes basados en clases usando keywords como class, extends y super, pero internamente el comportamiento es el mismo que ya estudiamos en este artículo. El listado 4 muestra el ejemplo de una clase y una subclase usando sintaxis de ES6.

```
class Point {
  constructor(x, y) {
    this.x = x;
    this.y = y;
  }
  toString() {
    return '(' + this.x + ', ' + this.y + ')';
  }
}

class ColorPoint extends Point {
  constructor(x, y, color) {
    super(x, y); // (A)
    this.color = color;
  }
  toString() {
    return super.toString() + ' in ' + this.color; // (B)
  }
}
```

Listado 4. Definición de clases con sintaxis de ES6

CONCLUSIÓN

Entender cómo funciona la herencia basada en prototipos es clave para poder hacer programación avanzada en JavaScript. Esperamos que estos ejemplos te hayan sido de ayuda. ☺

Este artículo fue traducido y editado por SG con el permiso del autor a partir de la versión original en inglés disponible en <http://www.masterjavascript.io/blog/2016/05/29/introducing-javascript-classes>

Special Purpose Languages

PARTE 3. UN RESULTADO IMPACTANTE

Por Luis Vinicio León Carrillo



Luis Vinicio León Carrillo es Director General y co-fundador de e-Quallity. Fue profesor-investigador en la universidad ITESO. Realizó estudios de posgrado en Alemania, durante los cuales abordó temas relacionados con la prueba de software y los métodos y lenguajes formales.

● **En el número anterior** definimos lo que es un lenguaje formal de la siguiente manera:

1. partimos de la definición del concepto de alfabeto como un conjunto finito de caracteres $B = \{b_1, b_2, \dots, b_m\}$, y dijimos que B tiene una cardinalidad (cantidad de elementos) igual a m , lo cual escribimos como $|B| = m$;
2. definimos la operación de concatenación \cdot sobre caracteres (lo que nos permite construir $b_5 \cdot b_4 \cdot b_8 \cdot b_8$) y sobre conjuntos de caracteres (lo que nos permite construir $B \cdot A \cdot C$);
3. definimos la exponenciación de la concatenación B^x sobre un alfabeto B como la aplicación de la concatenación $x-1$ veces sobre el alfabeto ($B \cdot B \cdot \dots \cdot B$, donde B aparece x veces);
4. definimos la cerradura de Kleene B^* como la unión de todas las exponenciaciones sobre el alfabeto B ($B^* = \bigcup_{i=0}^{\infty} B^i$);
5. definimos que un lenguaje formal sobre un alfabeto B es cualquier subconjunto de B^* .

De lo anterior se desprende que, dado un código ASCII extendido que llamaremos AE de cualquier cantidad de caracteres, podemos ver al lenguaje Java como el subconjunto de cadenas de cualquier longitud escritas utilizando ese AE que son aceptadas como "correctas" por el compilador de Java. Lo mismo es cierto si sustituimos "Java" por el nombre de cualquier lenguaje de computación.

Ahora bien, en AE^* están incluidos no solo todos los lenguajes de programación existentes y por existir, sino también todos los compiladores (escritos y por escribirse) de todos esos lenguajes. De hecho, en AE^* se encuentran todas las cadenas de caracteres que podamos escribir en esa extensión del ASCII, incluyendo los programas "incorrectos" en todos los lenguajes.

Contestemos ahora las primeras 2 preguntas que planteamos al final del número anterior, en el cual dijimos que dado un conjunto A que tiene i elementos, la cantidad de subconjuntos que pueden formarse sobre A es igual a 2^i .

Dijimos también que B^* , la Cerradura de Kleene sobre el alfabeto B , es un conjunto infinito pero contable, porque tiene la misma cardinalidad que el conjunto de los números naturales N . Esto implica en particular que, así como ocurre en el conjunto de los números naturales N , dada una de las cadenas de B^* , siempre podemos definir cuál es la que le sigue y listar sus elementos en orden sin saltarnos alguno (en N , dado un número natural y , sabemos que el que le sigue es $y+1$).

Del penúltimo párrafo se sigue que la cantidad de lenguajes sobre cualquier alfabeto B es igual a 2^N ... El asunto es que 2^N es una cardinalidad estrictamente mayor que la de N (siguiendo el Teorema de Cantor), lo que lo vuelve un conjunto no-contable (aquellos conjuntos en los que, dado un elemento x , no es posible decir cuál elemento es el que sigue, como en el caso de los números reales R : si tenemos el número 0.0004 no podemos decir que el que le sigue es 0.0005; tampoco que es 0.00041; en realidad, entre cualesquiera 2 números reales hay una cantidad infinita de números reales).

Una consecuencia importantísima de lo anterior es que la cantidad de lenguajes (2^N) que pueden definirse sobre un alfabeto es estrictamente mayor que la cantidad de cadenas (N) que pueden construirse sobre ese alfabeto, así es que: independientemente del tamaño del alfabeto, siempre tendremos lenguajes sobre ese alfabeto para los que no podemos definir un compilador, esto es, siempre habrá lenguajes que no podremos procesar.

Para nuestro propósito de construir lenguajes propietarios, esto significa que para poder procesarlos, en principio debemos observar algunas restricciones para su diseño. El lado bueno de esto es que, dado que estamos acostumbrados a utilizar lenguajes de programación y sus estructuras obviamente son

Independientemente del tamaño del alfabeto, siempre tendremos lenguajes sobre ese alfabeto para los que no podemos definir un compilador.

procesables, podemos incluir estructuras de ese tipo en nuestro lenguaje. El lado malo es precisamente eso mismo: como estamos acostumbrados a ello, probablemente no incluyamos otras estructuras que serían más adecuadas para las actividades que nuestros programadores automatizarán.

El punto no es para subestimarse: las estructuras que tenemos a nuestra disposición en un lenguaje (sea artificial o natural) pueden imponer límites a lo que podemos expresar (sea programar o pensar); como ejemplo tenemos el comentario del reconocido científico computacional C.A.R. Hoare sobre su método de ordenamiento Quick-Sort, después de que tomó un curso de Algol-60 (primer lenguaje en ofrecer la posibilidad de escribir subrutinas recursivas):

“Fue entonces que aprendí por primera vez sobre procedimientos recursivos y vi cómo programar el método de ordenamiento que previamente había tenido tanta dificultad para explicar.”

Para contestar la tercera pregunta necesitaremos elementos que revisaremos más adelante; pero antes, veamos algunos ejemplos.

Ejemplos con “especificaciones algebraicas”

Usando las definiciones arriba mencionadas y tomando como convención que escribiremos:

- $b_1 b_2$ en lugar de $b_1 \cdot b_2$ (concatenación de caracteres);
- b_i^n en lugar de $b_i \cdot \dots \cdot b_i$ (concatenación en la que b_i aparece n veces);
- a en lugar de b_1 , b en lugar de b_2 , c en lugar de b_3 , y d en lugar de b_4 (haciendo concreto el alfabeto abstracto $B = \{b_1, b_2, \dots, b_4\}$);

especifiquemos algunos lenguajes formales:

- $a^m b^n c^i$, para m, n, i enteros mayores a cero e independientes entre sí. Este sencillo lenguaje, que llamaremos L_1 , especifica todas las cadenas que tienen una o más a 's seguidas por una o más b 's seguidas por una o más c 's.
- $a^m b^n c^n d^m$, para $m, n \geq 0$. Obviando detalles, este lenguaje que llamaremos L_2 (más complejo que L_1), es una abstracción

de los sublenguajes incluidos en prácticamente todos los lenguajes de programación, en los que por ejemplo, los paréntesis y las llaves deben venir por pares (uno de cierre por cada uno de apertura) y estar “emparejados adecuadamente” (por ejemplo, no se acepta “(... { ... (...) ... } ”);

- $a^m b^n c^m d^n$, para $m, n \geq 0$. Obviando detalles, este lenguaje que llamaremos L_3 (más complejo que L_2), es una abstracción de la especificación del mecanismo de paso de parámetros que utilizan muchos lenguajes de programación para exigir que la cantidad de parámetros establecidos en la definición de una subrutina sea la misma que cuando ésta se utiliza.

Gramáticas

Después de esa primera definición de lenguaje en términos de concatenaciones sobre alfabetos, que podríamos llamar algebraica y que es más bien declarativa, tomaremos ahora un enfoque un poco más imperativo utilizando reglas (que pueden verse como una abstracción del concepto de patrones).

Podemos definir lenguajes utilizando gramáticas, las cuales son cuádruplas de la forma

$\langle T, N, S_0, R \rangle$ donde

T es el conjunto de los símbolos terminales del lenguaje (en adelante “Terminales”), los cuales podemos ver como el alfabeto con cuyos elementos finalmente se construyen las cadenas de caracteres;

N es el conjunto de los símbolos no-terminales del lenguaje (en adelante “no-Terminales”); aquellos símbolos abstractos que deben definirse en términos de Terminales y no-Terminales; nota: debe cumplirse que $T \cap N = \{\}$;

R es el subconjunto del producto cartesiano $(T \cup N)^* \times (T \cup N)^*$ que define la relación matemática que denota las reglas de reescritura del lenguaje (en adelante “Reglas”);

S_0 es el elemento distintivo de los no-Terminales llamado “Símbolo-inicial”, que es con el que inicia el “procesamiento” de R .

Escribiremos las Reglas utilizando el "meta-lenguaje" (llamado así porque es un lenguaje con el cual definimos lenguajes) llamado BNF (Backus-Naur Form). En él, para cada regla se coloca el primer elemento del producto cartesiano, seguido del símbolo "→" seguido del segundo elemento del producto cartesiano.

Ahora bien, cuando tengamos un conjunto de reglas como

$\alpha \rightarrow \beta_1$
 $\alpha \rightarrow \beta_2$
 ...
 $\alpha \rightarrow \beta_n$, donde α y todas las β_i son elementos de $(T \cup N)^*$ (o sea, elementos de la cerradura de Kleene de la unión de los Terminales y no-Terminales; es decir, combinaciones de uno o Terminales y no-Terminales),

lo escribiremos como

$\alpha \rightarrow \beta_1 \mid \beta_2 \mid \dots \mid \beta_n$

Si alguna β_i contiene cero Terminales y cero no-Terminales, la escribiremos como λ (la letra griega lambda), que representará la cadena vacía.

Además, escribiremos los Terminales con *minúsculas*, los no-Terminales con *MAYÚSCULAS*, y S_0 será el no-Terminal que aparece a la izquierda de "→" en la primer Regla. Con ello R define la gramática por sí sola, porque con las convenciones queda explícito cuáles son los Terminales, los no-Terminales y S_0 . Entonces, en adelante podremos definir un lenguaje listando únicamente las reglas de R.

Una jerarquía de lenguajes

Tenemos la siguiente categorización de lenguajes, que se hace considerando la complejidad de la estructura de las reglas en la R particular que lo define (por cuestiones de espacio, obviaremos algunos detalles):

- Llamaremos gramáticas regulares a aquellas cuyas reglas en R sean de la forma $A \rightarrow B d \mid d$ (o menos compleja); y llamaremos lenguajes regulares (o de Tipo 3, o Lgs_3 de manera corta) a aquellos que puedan definirse utilizando gramáticas regulares. Nota: las reglas pueden ser de la forma $A \rightarrow d B \mid d$, pero no se deben mezclar ambas formas.)
- Llamaremos gramáticas libres de contexto a aquellas cuyas reglas en R sean de la forma $A \rightarrow \alpha$ (o menos compleja), y lenguajes libres de contexto (o de Tipo 2, o Lgs_2) a aquellos que puedan definirse utilizando gramáticas libres de contexto.

- Llamaremos gramáticas sensibles al contexto a aquellas cuyas reglas en R sean de la forma $\beta_1 A \beta_2 \rightarrow \beta_1 \alpha \beta_2$ (o menos compleja), donde $\alpha \neq \lambda$; y lenguajes sensibles al contexto (o de Tipo 1, o Lgs_1) a aquellos que puedan definirse utilizando gramáticas sensibles al contexto. Decimos que β_1 y β_2 representan el contexto en el que A se puede transformar en α . Ahora bien, como $\alpha \neq \lambda$, entonces la suma de la cantidad de Terminales y de no-Terminales en α es mayor o igual a uno; a esta cantidad la llamaremos la longitud de α y la denotaremos con $|\alpha|$. Ahora podemos generar otra definición menos restrictiva (aunque una que no deja ver el atributo de "sensible al contexto"): llamaremos gramáticas sensibles al contexto a aquellas cuyas reglas en R sean de la forma $\theta \rightarrow \alpha$ (o menos compleja), donde $\theta \in (T \cup N)^*$, $\theta \neq \lambda$ y $|\theta| \leq |\alpha|$; y llamaremos lenguajes sensibles al contexto a aquellos que puedan definirse utilizando este tipo de gramáticas

- Por último, llamaremos gramáticas estructuradas por frases a aquellas cuyas reglas en R sean de la forma $\theta \rightarrow \alpha$ (o menos compleja), donde $\theta \in (T \cup N)^*$ y $\theta \neq \lambda$; y llamaremos lenguajes estructurados por frases (o Tipo 0, o Lgs_0) a aquellos que puedan definirse utilizando gramáticas estructuradas por frases.

Lo anterior define lo que es conocido como la Jerarquía de Chomsky, en la cual ocurre, como se puede observar, que $Lgs_3 \subsetneq Lgs_2 \subsetneq Lgs_1 \subsetneq Lgs_0$ (esto es, los Lgs_i son subconjuntos propios de los Lgs_{i-1}). Para el caso particular de $Lgs_3 \subsetneq Lgs_2$, esto significa que todos los lenguajes regulares (todas las gramáticas regulares) son libres de contexto, pero que hay lenguajes libres de contexto que no son regulares (para los que no existen gramáticas regulares que los definan).

La Jerarquía de Chomsky muestra con claridad un crecimiento significativo en la complejidad de la estructura de las reglas, o dicho de otra manera, una disminución significativa en las restricciones sobre esa estructura. Esto obviamente impacta en la complejidad (incluso en la posibilidad) de procesar todos esos tipos de lenguajes. Retomaremos el tema con más profundidad más adelante. ☹

Referencias

- [1] L. León. "Los Special Purpose Languages, parte 1". Revista Software Guru #48. <http://sg.com.mx/revista/48/los-special-purpose-languages>
- [2] L. León. "Los Special Purpose Languages, parte 2". Revista Software Guru #50. <http://sg.com.mx/revista/50/los-special-purpose-languages-2>



¡Actualízate
con los gurús!

WEBINARS GRATUITOS

CURSOS PAGADOS

CURSOS GRATUITOS

Más de 5,000 miembros

Más de 12 mil horas de
capacitación online impartidas



 **SGCampus**

 **@SGCampus**

www.sgcampus.com.mx

ARM como Pivote del IoT

Por Gunnar Wolf



Gunnar Wolf es administrador de sistemas para el Instituto de Investigaciones Económicas de la UNAM y desarrollador del proyecto Debian GNU/Linux.

<http://gwolf.org>

● **A lo largo de los años**, fuimos testigos de cómo las arquitecturas rivales de cómputo pasaron de una gran gama de implementaciones luchando por la supremacía durante los 80 y 90 a una aparente monocultura. La primera década del milenio vio sucumbir a poderosas y prometedoras plataformas como Alpha, m680x0 y HPPA-RISC. Sparc y PowerPC redujeron su participación en el mercado a únicamente casos muy especializados. MIPS pasó de ser el último grito en lo relativo a hardware de alto rendimiento a ser una arquitectura para dispositivos de bajo consumo y bajo precio, como la mayor parte de los switches y puntos de acceso inalámbrico disponibles en el mercado.

CONSOLIDACIÓN DE UNA ARQUITECTURA REINANTE

Por varios años, parecía que la plataforma x86, definida por Intel como una serie de “parches” sucesivos al procesador 8086 de 1978, reinaría sin rivalidad en prácticamente todos los mercados. Computadoras de escritorio, portátiles, servidores. Hacia 2005, la plataforma x86 se erigía como clarísimo dominante. Y esto no por ser más potente o eficiente que su competencia, sino por tener una muchísima mayor base de usuarios ya instalada, por mantener una gran compatibilidad hacia atrás, y porque prácticamente todo el software de consumo masivo aparecía únicamente para éste.

Es muy sabido que esta compatibilidad hacia atrás no siempre es lo más conveniente técnicamente. En la década de 1980 nació RISC (*Reduced Instruction Set Computing*, cómputo con un conjunto reducido de instrucciones), primero como concepto teórico y poco más tarde implementado en varias arquitecturas de hardware (algunas de ellas mencionadas en el primer párrafo). Ya en esos años se veía que las principales arquitecturas disponibles tenían conjuntos de instrucciones demasiado complejos (recibiendo la denominación CISC, *Complex Instruction Set Computing*, cómputo con un conjunto de instrucciones complejo, en contraposición con la nueva tendencia).

Intel mismo intentó en dos ocasiones diseñar una nueva familia de procesadores que reemplazara a x86. La arquitectura RISC i860, lanzada en 1989 y abandonada unos cinco años más tarde, y la Itanium o IA-64, que se ha mantenido viva de manera formal

desde su debut en 2001, pero en la práctica y ante el abandono de prácticamente todos los fabricantes y proveedores de software, es una arquitectura cada vez más anecdótica.

ARM: JAQUE AL REY

Parecería que en este recuento de la gradual homogeneización de las plataformas de cómputo no hay mucho más que decir. Pero en los últimos años, se ha dejado sentir fuertemente un cambio drástico, y desde el segmento bajo del mercado.

La plataforma de hardware ARM no es bajo ningún concepto un nuevo jugador en el mundo del cómputo: su diseño original se produjo durante los años 1983 a 1985, al interior de la empresa Acorn, que buscaba desarrollar un sucesor para el popular CPU de 8 bits 6502 empleado en muchas computadoras de la época y, en particular, en la BBC Micro, plataforma de gran éxito en el Reino Unido. De hecho, ARM se manejó originalmente como acrónimo de “Acorn RISC Machine”, y una vez que la Acorn dejó de ser un hito reconocible en la mente de los usuarios británicos, para 1990 se independizó de la marca padre para denominarse “Advanced RISC Machines”.

Los procesadores ARM son un ejemplo para libro de texto de las diferencias y ventajas de RISC: si bien su conjunto reducido de instrucciones significa que, a una misma velocidad, pueden realizar mucho menos trabajo real que su competencia CISC, el número de transistores requerido para implementarlo resulta decenas de veces menor que el equivalente CISC de su época. Por ejemplo, cuando en 2007 se introdujo el ARM Cortex-A9, de 26 millones de transistores, los CPUs i7 de Intel excedían los 730 millones.

Menos transistores se traduce en muy distintas ventajas, todas ellas importantes para el cambio que hoy estamos observando:

- Fabricar CPUs ARM resulta más barato, dado que pueden usar procesos que requieren menor miniaturización y hay más fabricantes en condiciones de hacerlo (ARM no tiene ninguna fábrica, es dueño únicamente del diseño de los chips).
- La disipación de calor es menor. Un procesador

CISC actual requiere de enfriamiento activo, empleando masivos disipadores y ventiladores. La operación normal de un procesador ARM resulta mucho más aceptable para un equipo de perfil delgado, y más aún cuando está diseñado para llevarse en contacto con el cuerpo de su usuario.

- El consumo eléctrico es menor, permitiendo mucho mayor tiempo de operación a baterías. Naturalmente, el tamaño de los chips resultantes es menor. Mientras que un CPU Intel para equipos de escritorio mide unos 4cm. por lado, los procesadores ARM apenas y sobrepasan un centímetro cuadrado.

Los procesadores ARM siempre estuvieron presentes en nuestros dispositivos “menores”, pero dado que nuestra relación con estos nunca fue tan intensa y personal como en los últimos años, rara vez nos resultó claro. Al haber dado el gran salto cualitativo el mercado de los dispositivos móviles, la plataforma ha dado un salto tal que se volvió la primera respuesta al considerar cada vez más casos de uso, incluso ya en algunos casos entrando al segmento tradicionalmente fuerte de x86.

ARM es una arquitectura muy versátil; tienen tres perfiles de CPUs definidos. De más sencillo a más poderoso, “M” (para microcontroladores), “R” (para aplicaciones de tiempo real) y “A” (para aplicación).

COMPLICACIONES DE LA PLATAFORMA

La plataforma x86 tiene, sin embargo, una gran ventaja: la apertura y estandarización de las principales interfaces entre ésta y su sistema operativo. Para bien o para mal, desde la aparición de la IBM PC en 1981, todos los sistemas compatibles cuentan con un BIOS que se adhiere a una interfaz clara y pública; si bien esta ha sido extendida, mejorada y finalmente reemplazada — por UEFI, que a pesar de los puntos negativos que tenga, era una actualización largamente esperada — a lo largo de sus 35 años.

El BIOS funge de intermediario entre el hardware y el sistema operativo en sus primeras etapas (sobre todo en los primeros años de la PC; hoy en día la funcionalidad se limita casi en exclusiva a la inicialización del sistema). Al arrancar la computadora, se encarga de enumerar y realizar pruebas básicas de funcionalidad al hardware del sistema (el POST). Tras este paso, el sistema operativo puede lanzarse con conocimiento básico del hardware con el que operará.

En la arquitectura ARM, el BIOS sencillamente no existe. La razón para esto es que la compañía ARM funciona de forma distinta de Intel. La compañía no vende *chips*, no posee fábricas, sino que

licencia a terceros sus diseños como propiedad intelectual; los chip ARM son en realidad fabricados, entre muchas otras empresas, por AMD, Broadcom, Freescale, Texas Instruments, e incluso el mismo Intel.

En el mundo ARM, es muy común que los chips encapsulen no únicamente al CPU, controlador de memoria, controlador gráfico, y otros componentes esenciales, sino que prácticamente a todo el sistema de cómputo, inclusive la RAM y los controladores de dispositivos. A esta práctica se le conoce como “SoC” (System on a Chip); de hecho, si ven cualquiera de las computadoras basadas en ARM que han surgido en años recientes, uno de los primeros puntos que llama la atención es su relativa simplicidad. Y si acostumbran abrir, en claro ejercicio de un espíritu científico, sus teléfonos o tabletas, notarán precisamente lo mismo.

Las computadoras ARM, pues, no están normalmente concebidas para su expansión. Por sus características físicas, y por su bajo costo, es más bien un enfoque de un sistema para cada tipo de necesidad; desde el punto de vista en pro de la ecología suena nefasto, pero si requieren mayor poder de cómputo, la respuesta es cambiar de computadora. Además, la mayor parte de los fabricantes buscan ubicar sus equipos como “cajas cerradas”. ¡No es de su interés que yo pueda ejecutar mi software favorito en sus teléfonos en vez del “oficial”!

Ahora, dado que la computadora no requiere de la enumeración y prueba que realiza el BIOS, se ha optado por prescindir de este elemento. Es, pues, responsabilidad del sistema operativo saber qué dispositivos tiene y dónde se ubican. En el caso de Linux, el sistema operativo más común para entornos ARM, significa que debe construirse un núcleo ligeramente distinto para cada modelo de computadora. Este conocimiento del entorno se plasma en una estructura conocida como *Device Tree* (árbol de dispositivos).

Las plataformas Arduino y Raspberry Pi han resultado fundamentales, no por ser únicas o por ser las más económicas, sino por masificar el acceso a esta tecnología llevándolo al mercado educativo, y servir así como base para cientos de productos tangibles.

Adecuar Linux para su funcionamiento en una nueva sub-arquitectura no es una tarea trivial. Hay un fuerte empuje hoy en día para que los desarrolladores de hardware realicen este trabajo de adecuación con mayor colaboración y en consonancia con la comunidad desarrolladora primaria. Tengo plena esperanza en que, conforme se construye el ambiente de desarrollo para el afromado IoT, irá resultando claro a los fabricantes el valor de mantener la apertura ante los usuarios entusiastas. ☺

Hola Mundo IoT

Por Pedro Galván

● **En el mundo maker**, lo más cercano a un “hola mundo” es hacer que un microcontrolador prenda y apague un diodo emisor de luz (LED). Si llevamos esto a un contexto de Internet de las Cosas, el “hola mundo” sería lograr que dicho microcontrolador prenda y apague el LED en base a mensajes enviados por un dispositivo remoto.

Así que eso es justamente lo que haremos en este tutorial. Tendremos un microcontrolador que prenderá y apagará un LED en base a mensajes generados remotamente desde un Raspberry Pi.

ELIGIENDO EL PROTOCOLO DE COMUNICACIÓN

Lo primero que tenemos que decidir es cómo haremos para que nuestros dispositivos puedan enviar y recibir mensajes.

Quienes vienen del desarrollo web probablemente consideren que lo más sencillo sería implementar un servidor web que maneje peticiones REST sobre HTTP. Cuando se trata de dispositivos con alto poder de cómputo y un canal de comunicación con buen ancho de banda, esto tiene sentido. Pero en la mayoría de los casos de IoT, esto no sería ideal ya que HTTP es un protocolo relativamente pesado, además de que no es eficiente para escenarios de comunicación uno a muchos (se necesitaría hacer un POST a cada cliente).

Es aquí cuando nos encontramos con MQTT, un protocolo de comunicación M2M (machine-to-machine) donde dispositivos se comunican entre sí utilizando un patrón publicar/suscribir. Es muy ligero, por lo que es de especial utilidad en casos donde se requiere un muy bajo consumo de energía y ancho de banda. MQTT es un estándar ISO, así que tiene un gran respaldo. Por sus características, MQTT es un protocolo ideal para soluciones de Internet de las Cosas, así que es lo que utilizaremos para nuestro “hola IoT”.

FUNCIONAMIENTO DE MQTT

Conozcamos los componentes y acciones clave de MQTT.

- **Cliente:** un dispositivo que puede publicar mensajes, suscribirse para recibir mensajes, o ambos.

- **Broker:** el broker es el servidor que acepta mensajes publicados por clientes y los difunde entre los clientes suscritos.
- **Publicar:** cuando un cliente envía un mensaje al broker.
- **Tópico:** los mensajes deben estar etiquetados con algún tópico o tema. Los clientes se suscriben a tópicos específicos, de manera que solo reciben los mensajes publicados con dichos tópicos. Un tópico puede a su vez contener subtópicos.

MAPEANDO LOS COMPONENTES

Para este tutorial tendremos 3 elementos:

- Una computadora personal, que operará como servidor (broker).
- Un Raspberry Pi que por medio de un programa en Python enviará mensajes al broker pidiendo que se prenda o apague el LED.
- Un módulo ESP8266 que recibe los mensajes y realiza la acción deseada con el LED.

MOSQUITTO

Ya comentamos que MQTT es el protocolo que utilizaremos, pero ahora hay que escoger un servidor o implementación específica de broker MQTT. Existen diversas opciones instalables como Mosquitto, HiveMQ y VernMQ por mencionar algunas. También hay servicios de broker MQTT en la nube —por ejemplo AWS IoT e IBM Bluemix— listos para usarse.

En este caso utilizaremos Mosquitto, que es un servidor MQTT open source bastante popular. En <http://mosquitto.org> puedes descargar el instalador para Windows. En Mac puedes instalarlo con Homebrew, y en Linux encontrarás el paquete mosquitto o mosquitto-mqtt en los repositorios de tu sistema operativo.

Una vez que tengamos Mosquitto instalado, podemos arrancar el broker simplemente con el comando mosquitto, aunque es buena idea agregarle la bandera -v de “verbosidad” para que nos muestre el detalle de lo que esté sucediendo.

```
$ mosquitto -v
1467416210: mosquitto version 1.4.9 (build date 2016-06-18 16:16:54+0100) starting
1467416210: Using default config.
1467416210: Opening ipv4 listen socket on port 1883.
1467416210: Opening ipv6 listen socket on port 1883.
```

Figura 1. Arrancando el servidor de mosquitto.

La configuración default de mosquitto opera en el puerto 1883 prácticamente con ninguna seguridad. Esto es válido para propósitos de nuestro tutorial, pero una vez que nos dispongamos hacer un proyecto real, es altamente recomendable cambiar la configuración y restringir el acceso.

Una vez que está corriendo nuestro servidor, podemos suscribirnos para recibir mensajes sobre algún tópico. Antes de hacer la conexión desde dispositivos remotos, vamos a probar desde la misma computadora donde tenemos el servidor. Abrimos una nueva terminal y ejecutamos el comando `mosquitto_sub`:

```
$ mosquitto_sub -h 127.0.0.1 -t prueba -i testSub
```

En este caso estamos indicando que deseamos conectarnos al servidor en la ip 127.0.0.1 al tópico prueba y que nuestro identificador es testSub (este último parámetro es opcional, sino lo usamos, el servidor nos asigna algún identificador). Al ejecutar este comando, veremos en nuestra consola del servidor que ha recibido la petición de suscripción (ver figura 2).

```
1467417031: New connection from 127.0.0.1 on port 1883.
1467417031: New client connected from 127.0.0.1 as testSub (c1, k60).
1467417031: Sending CONNACK to testSub (0, 0)
1467417031: Received SUBSCRIBE from testSub
1467417031: prueba (QoS 0)
1467417031: testSub 0 prueba
1467417031: Sending SUBACK to testSub
```

Figura 2. Servidor mostrando suscripción.

Noten que el tópico “prueba” no se había definido previamente en el servidor, sino que se crea automáticamente cuando un cliente publica o se suscribe a él.

Ahora haremos la prueba de publicar un mensaje al tópico prueba. Así que desde otra terminal de comando ejecutamos el comando `mosquitto_pub`:

```
$ mosquitto_pub -h 127.0.0.1 -i testPub -t prueba -m 'Hola IoT'
```

Como podemos ver en la figura 3, el servidor recibe la conexión de testPub que publica un mensaje en el tópico prueba. Dado que testSub está suscrito a este tópico, el servidor le envía el mensaje.

Por último, en la figura 4 podemos apreciar que el cliente que está escuchando el tópico prueba recibe el mensaje publicado.

```
1467417556: New client connected from 127.0.0.1 as testPub (c1, k60).
1467417556: Sending CONNACK to testPub (0, 0)
1467417556: Received PUBLISH from testPub (d0, q0, r0, m0, 'prueba', ... (8 bytes))
1467417556: Sending PUBLISH to testSub (d0, q0, r0, m0, 'prueba', ... (8 bytes))
```

Figura 3. Servidor recibe conexión para publicar.

```
$ mosquitto_sub -h 127.0.0.1 -i testSub -t prueba
Hola IoT
```

Figura 3. Suscriptor recibiendo el mensaje publicado.

PUBLICAR DESDE LA RASPBERRY PI

Para publicar mensajes desde la Raspberry Pi podríamos simplemente instalar mosquitto y utilizar el comando `mosquitto_pub` tal y como lo hicimos en la prueba anterior. Sin embargo, para hacerlo de forma programática es mejor utilizar una librería para el lenguaje de programación de nuestra preferencia. En este caso mostraremos como hacerlo desde Python.

El proyecto Eclipse Paho provee clientes y librerías open source para MQTT en una gran variedad de lenguajes de programación. Podemos visitar <https://eclipse.org/paho/> y descargar el cliente para nuestro lenguaje, o en el caso de Python podemos hacerlo todavía más sencillo usando pip:

```
$ pip install paho-mqtt
```

Una vez que tengamos instalado Paho, podemos utilizar la librería en nuestro programa. El listado 1 muestra el código en python para publicar un mensaje.

```
import paho.mqtt.publish as publish
import time
print("Enviando 0...")
publish.single("ledStatus", "0", hostname="tuserver")
time.sleep(3)
print("Enviando 1...")
publish.single("ledStatus", "1", hostname="tuserver")
```

Listado 1. Programa para publicar mensaje con Python.

Como podemos ver, es muy sencillo. Simplemente importamos la librería publish de Paho y utilizamos la operación single para publicar un mensaje. Como parámetro enviamos el nombre del tópico (ledStatus), el mensaje (0 ó 1), y el servidor.

RECIBIR EL MENSAJE EN MICROCONTROLADOR

Una vez que hicimos el programa en nuestra Raspberry Pi para enviar mensajes, vamos a hacer el programa en nuestro microcontrolador para recibir los mensajes y prender o apagar el LED. En este caso usamos un ESP8266 ya que es un módulo muy económico con conectividad WiFi, pero podríamos usar cualquier Arduino con un shell Ethernet o WiFi.

Así como en la Raspberry Pi nos apoyamos en una librería, haremos lo propio del lado del microcontrolador. Utilizaremos PubSubClient, un cliente MQTT para Arduino. Puedes descargar PubSubClient desde <http://pubsubclient.knolleary.net> o instalarlo desde el administrador de paquetes del Arduino IDE.

El listado 2 muestra el código para el microcontrolador Arduino que se conecta al servidor de MQTT, se suscribe al tópico y en base al mensaje recibido, prende o apaga el LED. Vale la pena recalcar que hicimos la prueba con un ESP8266 Huzzah de Adafruit que tiene invertida la acción de los LED, cuando mandas HIGH se apaga y cuando mandas LOW se prende. Esto puede ser distinto en el microcontrolador específico que utilices.

CONCLUSIÓN

En este tutorial he mostrado un ejemplo muy sencillo de cómo dispositivos remotos pueden enviarse mensajes entre sí por medio de un servidor MQTT. Te recomiendo que conozcas más sobre MQTT ya que es un protocolo muy útil.

Si no deseas instalar tu propio servidor MQTT, te recuerdo que también hay servicios en la nube. De hecho, por el momento Eclipse ofrece un sandbox de MQTT abierto disponible en "iot.eclipse.org" con el puerto 1883 así que podrías intentar los ejercicios de este tutorial utilizando "iot.eclipse.org" como servidor MQTT y deberían funcionar. ☺

```
#include <EEPROM.h>
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include <Wire.h>
#include <Adafruit_INA219.h>

const char* ssid = "Tu Wifi";
const char* password = "tupassword";
const char* mqtt_server = "tuservidor";

WiFiClient espClient;
PubSubClient client(espClient);

const byte ledPin = 0; // pin con el LED

void callback(char* topic, byte* payload, unsigned int length) {
  Serial.print("Mensaje recibido [");
  Serial.print(topic);
  Serial.print("] ");
  for (int i=0;i<length;i++) {
    char receivedChar = (char)payload[i];
    Serial.print(receivedChar);
    if (receivedChar == '0')
      // En el ESP8266 Huzzah el led está invertido
      digitalWrite(ledPin, HIGH);
    if (receivedChar == '1')
      digitalWrite(ledPin, LOW);
  }
  Serial.println();
}

void reconnect() {
  while (!client.connected()) {
    Serial.print("Intentando conectarse...");
    if (client.connect("ESP8266 Client")) {
      Serial.println("conectado");
      client.subscribe("ledStatus");
    } else {
      Serial.print("falla, rc=");
      Serial.print(client.state());
      Serial.println(" reintentaremos en 5 segundos");
      delay(5000);
    }
  }
}

void setup() {
  Serial.begin(9600);
  client.setServer(mqtt_server, 1883);
  client.setCallback(callback);
  pinMode(ledPin, OUTPUT);
}

void loop() {
  if (!client.connected()) {
    reconnect();
  }
  client.loop();
}
```

Referencias

[1] J. Lewis. "MQTT Tutorial for Raspberry Pi, Arduino and ESP8266". <http://swgu.ru/rh>

Listado 2. Código para ESP8226



Administra la agenda del evento, conferencias, ponentes y documentos en tiempo real

Envía alertas y notificaciones a todos tus invitados

Conoce la opinión de tus invitados creando encuestas y votaciones

Comparte itinerarios de vuelos, hoteles y traslados dentro de Eventto App

www.eventto.com.mx

solicita una demostración



Personaliza **Eventto** con los colores y el logotipo de tu marca como algunos de nuestros clientes



contáctanos
info@eventto.org

Perdamos el Miedo al RTFM

Por Pedro Galván

● Recuerdo muy bien la primera vez que alguien me contestó "RTFM", y a la fecha le estoy agradecido.

Era yo un estudiante universitario, y en mis tiempos libres jugaba y programaba como voluntario en un MUD (*multi-user dungeon*). Este era un MUD 100% amateur, el servidor estaba hospedado en un ISP en Noruega, y los desarrolladores éramos voluntarios de todo el mundo. Había cuatro niveles de voluntarios: constructor, alquimista, lord y semi-dios. Los programadores eran los novatos, que hacían cuestiones básicas como cuartos y monstruos; los alquimistas hacían cosas más avanzadas como armas y hechizos; los lords eran como los gerentes, coordinaban a varios constructores para construir pueblos y regiones; los semidioses ocasionalmente mejoraban la plataforma base, pero más que nada fungían como mentores técnicos para los constructores y alquimistas.

Uno de los semidioses más activos y queridos se llamaba Taniwha (bueno, en el mundo real se llamaba Peter, pero en el MUD se hacía llamar Taniwha así que eso es lo que importa). Taniwha era un ingeniero de software que había trabajado en varias empresas de tecnología y luego fundó una empresa que fue comprada por IBM, así que ya solo trabajaba de medio tiempo como asesor, y el resto de su tiempo lo dedicaba a su familia y al MUD. Como se podrán imaginar, Taniwha era mi ídolo.

Después de algunas semanas como constructor haciendo cosas aburridas, decidí que era tiempo de hacer algo más divertido, así que opté por crear un látigo para sacerdotisas drow (elfa oscura). Los aficionados del tema recordarán que un látigo de estos tiene varias cabezas de serpiente (dependiendo del nivel de la sacerdotisa) y puede generar efectos secundarios (ej. envenenar, paralizar) a la víctima. Así que no era un arma sencilla de programar.

- Yo (chat público de desarrolladores): "¿Alguien sabe como puedo paralizar a una víctima?"
- Taniwha: "Revisa el hechizo de paralización, está en x directorio".
- Yo (mensaje directo a Taniwha): "¿Me puedes explicar como funciona?"
- Taniwha: "RTFM".
- Yo: "¿Qué es eso?"
- Taniwha: "Read the Fucking Manual".

Me quedé en shock. Mi ídolo me estaba mandando al carajo simplemente por preguntarle algo. O andaba muy enojado por otra cosa, o no era tan buena persona como yo creía. Después de unos momentos de respiración profunda, retomé la conversación.

- Yo: "Hey, ¿qué te pasa? No es mi culpa si tuviste un mal día".
- Taniwha: "¿De qué estás hablando?"
- Yo: "A tu RTFM".
- Taniwha: "Ah, ¿eso? No lo tomes personal, es una expresión común."
- Yo: "Pues no estoy de acuerdo. Se supone que me deberías enseñar, no mandarme al cuerno".
- Taniwha: "Te estoy enseñando. Cálmate y lee el manual."

Desde entonces, antes de preguntar, primero busco el manual. Es una práctica muy sana. De la misma forma, cuando alguien de mi equipo me pregunta algo, lo refiero al manual. Y si no hay manual, trato de crearlo.

HACIENDO MÁS RTFM EN LAS EMPRESAS

En ambientes de software libre o proyectos de voluntariado, es de lo más natural aplicarle a alguien un RTFM, especialmente en países sajones.

En Latinoamérica no he visto que se aplique tanto. Pensamos que enseñarle a alguien es sentarnos con la persona y resolver todas sus dudas, el número de veces que sea necesario. Siendo de sangre caliente, también acostumbramos tomarnos las cosas más personales ("es un hijo de \$%&@, le pregunté algo y me dijo que lea el manual, que se vaya al cuerno yo ya no quiero trabajar con él."). Y en ambientes corporativos las cosas se complican todavía más, ya que sin la cultura adecuada un RTFM resultará en desmotivación, renuncias e incluso peleas.

Esto es una lástima porque creo que es una práctica muy sana. En esencia, la práctica de RTFM se enfoca en dos cosas: por un lado que la gente senior busque que el conocimiento de la empresa sea documentado, y por otro que la gente novata aprenda a auto-capacitarse y entender cómo funcionan las cosas.

Así que si alguna vez te aplican un RTFM, no lo tomes a mal, lo están haciendo por tu bien. 🙏

Pese a la gran cantidad de datos públicos y privados existentes hay muy pocos capacitados para estructurarlos y generar información que facilite la toma de decisiones.

Es por esto que la



Sociedad de Científicos de Datos de México

te invita a sus próximos cursos:

BOOTCAMP DE CIENCIA DE DATOS (Espéralo en verano)

R Y PYTHON, niveles principiante y avanzado

SQL, niveles principiante y avanzado

VISUALIZACIÓN EFECTIVA Y SUS APLICACIONES

MÉTODOS DE APRENDIZAJE DE MÁQUINA

MINERÍA DE TEXTO

ANÁLISIS DE INFORMACIÓN GEOGRÁFICA

ARQUITECTURA E INFRAESTRUCTURA DE DATOS

Todos ellos son impartidos por científicos de datos con trayectoria tanto en la Academia como en el ámbito profesional.

Si deseas informes, puedes solicitarlos a contacto@sociedat.org



@sociedat



Sociedad Hispanoamericana de Ciencia de Datos

www.sociedat.org

1

KNOCKI

Imagina el siguiente escenario: te sientas en el sillón frente al televisor y no encuentras el control remoto, así que extiendes tu brazo y das un par de golpes en la mesa del sillón, y de pronto se prende. ¡Wow! Ese es un ejemplo sencillo del tipo de cosas que podrás hacer con Knocki. Es un pequeño dispositivo programable que se adhiere a superficies como paredes o mesas y detecta los golpes que das en éstas, y en base a ello dispara acciones en otros dispositivos a los cuales se conecta por medio de la red WiFi. Recientemente juntó más de 1 millón de dólares en Kickstarter y se espera que entregue sus primeras unidades en el primer trimestre de 2017.



2

PINBOX 3000

Pon en práctica eso que le has dicho a tus hijos de que “en mis tiempos hacíamos nuestros propios juegos” con este kit para construir tu propio juego de pinball de cartón. El kit incluye 12 hojas de cartón precortado, canicas, palancas, e instrucciones de armado. Para decorarlo puedes utilizar algunas de las ideas incluidas o hacer tu propio diseño y decoración. Si compras dos kits, puedes unirlos para que dos personas jueguen al mismo tiempo en modo de batalla. También puedes adquirir el kit de expansión Arduino, el cual incluye un microcontrolador Arduino Nano, LEDs, bocina, servomotor e instrucciones para agregar un mayor elemento de interacción a tu juego de pinball. El microcontrolador ya viene preprogramado con una rutina default, pero puedes modificarla al gusto.



3

SAMSUNG GEAR 360

Como podrás derivarlo de su nombre, la Gear 360 es una cámara para tomar fotografías y video de 360 grados. Cuenta con dos lentes de ojo de pez, uno en cada lado, equipados con sensores de 15 MP que se combinan para capturar video 3D a una resolución de 2840 x 1920 e imágenes estáticas a 7776 x 3888. Entre las ventajas de la Gear 360 está su diseño futurista, la calidad de sus imágenes y su facilidad de uso. El inconveniente principal es que para aprovechar al máximo sus capacidades, como por ejemplo ver en tiempo real el video que se está capturando, requieres un smartphone Samsung “moderno” (Galaxy S6 o posterior). La Gear 360 se lanzó en Asia en abril de este año, se espera que en Estados Unidos esté disponible antes de que termine el verano y llegue a Latinoamérica para la temporada navideña.

4

SEGWAY ROBOT @geek_mx



El Segway Robot es un pequeño asistente a control remoto que se mueve como un Segway regular, pero puede responder y obedecer comandos enviados desde un smartphone compatible. Utiliza un scooter Mini Pro para moverse y puede transformarse fácilmente entre robot autónomo y un vehículo que lleve a su dueño a diferentes lugares. Captura imágenes usando Intel RealSense y mapea una zona con todos sus obstáculos, navegando a través de complicados entornos. También puede hablar, seguir a las personas, reconocer el rostro de su dueño y mucho más. Para todos aquellos interesados en desarrollar para Segway Robot, se pueden unir al programa de SDK para conocer todo el potencial del proyecto, extendiendo la funcionalidad del robot a través de sus diferentes interfaces y extensiones de hardware. El programa de desarrollo dará inicio con la fase Alpha que se lanzará en octubre, seguido de la Beta en enero de 2017. Se espera un lanzamiento general en la primera mitad del próximo año; el precio se dará a conocer conforme se acerque esa fecha.

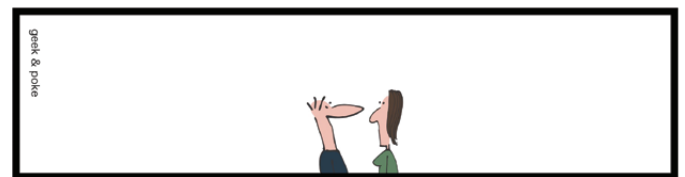


5

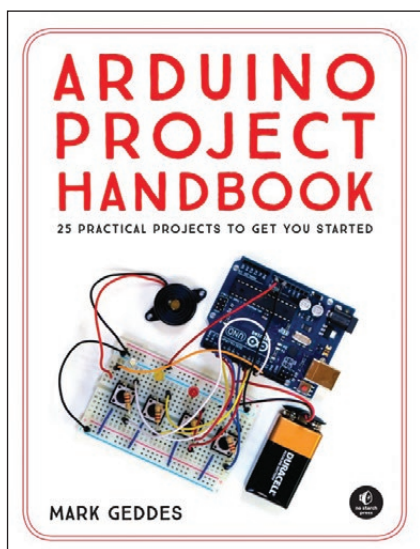
JOHNNY-FIVE INVENTOR'S KIT

El Johnny-Five Inventor's Kit (J5IK) es un kit para desarrollo de soluciones de robótica e internet de las cosas que combina dos plataformas: Johnny-Five, un framework de programación con JavaScript para IoT; y Tessel 2, una plataforma de hardware abierto para robótica e IoT basada en una computadora single-board. El J5IK está diseñado principalmente para principiantes que buscan hacer sus primeros proyectos. No requiere experiencia previa de programación ni soldadura. El kit incluye una gran variedad de componentes electrónicos de manera que tienes todo lo que necesitas para completar 14 proyectos distintos que te enseñarán a leer sensores, controlar motores, y aprender JavaScript. Después de realizar los proyectos tendrás el conocimiento necesario para diseñar y construir tus propios proyectos de IoT.

Humor



MUNDO CONECTADO



ARDUINO PROJECT HANDBOOK
Mark Geddes. No Starch Press, junio 2016

1

El Arduino Project Handbook es una colección de proyectos usando la tableta Arduino. Con una computadora, un microcontrolador Arduino y algunos componentes electrónicos, aprenderás a construir varios proyectos divertidos. El autor, Mark Geddes, originalmente escribió este libro para enseñar electrónica y microcontroladores a su hijo de 10 años sin tener que pasar primero por cientos de páginas de teoría.

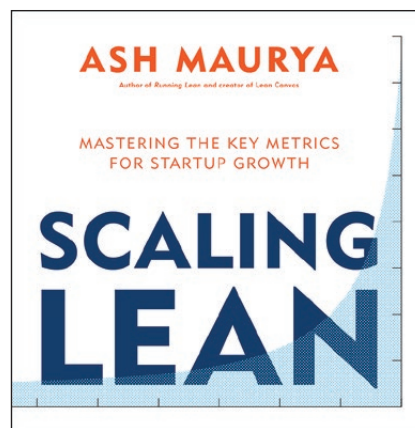
Son 25 proyectos y cada uno cuenta con instrucciones detalladas, fotografías, diagramas de circuitos y código completo, de tal manera que hasta los más novatos no tengan ningún problema para arrancar. El libro comienza desde lo más básico, una introducción a Arduino y consejos sobre herramientas y componentes electrónicos. A partir de eso ya puedes escoger si seguir los proyectos en orden o escoger aquellos que más te interesen.

Algunos de los proyectos incluidos son: una estación de monitoreo del clima, un juego de memoria basado en luces de colores, un sistema de alarma por detección de movimiento, un estroboscopio, un show de luces de colores, y una chapa que detecta claves para tocar la puerta, entre otros.

Arduino Project Handbook está disponible en versión impresa y digital. Adicionalmente, en <https://www.nostarch.com/arduinohandbook> puedes descargar un capítulo con el proyecto del juego de memoria.

SCALING LEAN
Ash Maurya. Lean Stack, junio 2016

2



Scaling Lean es el libro más reciente de Ash Maurya, uno de nuestros autores y consultores favoritos sobre emprendimiento tecnológico. Scaling Lean brinda una valiosa guía para modelar el camino de tu startup hacia el éxito. Aprenderás cómo obtener métricas esenciales que te permitirán medir la ejecución de tu modelo de negocio, obtener el pulso de tu startup, comunicarte con tus inversionistas y saber cuando es necesario intervenir para corregir el rumbo.

El autor, Ash Maurya, comenta que aunque su libro anterior, Running Lean, plantea un mapa de ruta táctico de cómo llevar una idea a la realidad, se dio cuenta que al aplicarlo en escenarios reales quedaban muchas dudas de implementación. Scaling Lean busca brindar una guía para contestar estas dudas. Entre los aspectos que se enfoca Scaling Lean están: cómo medir o cuantificar un modelo

de negocio, cómo balancear el aprendizaje interno del startup y la entrega de resultados a inversionistas, qué es lo que sucede después de lanzar un MVP y cómo mantenerse enfocado en los principales riesgos.

El libro también incluye guías para hacer cosas como:

- Evaluar la viabilidad de un modelo de negocio en unos cuantos minutos.
- Establecer metas progresivas orientadas a crecimientos exponenciales al largo plazo.
- Planear y ejecutar sprints lean de dos semanas para rápidamente obtener, priorizar y probar ideas de negocio.

Si planeas adquirir este libro, te recomendamos hacerlo desde <https://leanstack.com/scaling-lean-book> ya que obtendrás beneficios adicionales de acceso a contenido y herramientas de soporte.

DESCUENTO PARA GRUPOS
Asista con su equipo de trabajo
y aproveche el precio preferencial.



Gartner CIO & IT Executive Latin America Summit

14 - 17 noviembre 2016

Cancún, México

www.gartner.com/americas/cio

Tels: +52 55 9171 1714

Las sesiones de la conferencia serán
impartidas por Analistas de Gartner y
organizadas por los siguientes vertientes:

- **CIO:** Liderazgo, estrategias empresariales y transformación para su recorrido digital
- **Seguridad y Gestión de Riesgos:** Crear confianza y resiliencia a la velocidad de los negocios
- **Aplicaciones:** Aplicaciones desarrolladas y originadas en la economía digital
- **Datos y Analítica:** La Información, analítica y algoritmos impulsarán los negocios digitales

Si pides 5 alitas en el restaurante, ¿te comes una y tiras 4?



Entonces, ¿por qué desperdiciar el 80% de tu inversión con capacitación tradicional?

Evita el aprendizaje chatarra (*Scrap Learning*) y optimiza tu inversión en capacitación, con los **ambientes de aprendizaje** de Abiztar Learning Technologies.

VER VIDEO "El 80% de la capacitación se va a la basura"



Nuestro compromiso contigo no termina con un curso, termina con el éxito de tus proyectos.

+52 (55) 5594 6411

cursos@abiztar.com.mx / www.abiztar.com.mx

