

NEWSLETTER

El Problema del Huevo y la Gallina

La idea de la publicidad consiste en mentir sin ser descubierto. La mayoría de las empresas, cuando llevan a cabo una campaña de publicidad, simplemente cogen la idea más descabellada sobre su empresa, le dan la vuelta e inculcan esa mentira. Llamésmole «veracidad por aserciones repetitivas». Por ejemplo, los viajes en avión son incómodos, se dispone de poco espacio y los empleados de las líneas aéreas son maleducados y desagradables, sin contar que el aire *acondicionado* está diseñado como un elemento de tortura. Así, la mayoría de los anuncios de compañías aéreas nos van a mostrar cuán *cómodo y agradable* es volar y que *mimados* estaremos todo el viaje. Como cuando British Airways se anunció con un hombre de negocios en el asiento de un avión, durmiendo como si estuviese en la cuna de un niño; cualquier parecido con la realidad era pura coincidencia.

¿Hace falta poner otro ejemplo? Las compañías papeleras están devastando por completo nuestros bosques, y cortan árboles milenarios sin remordimientos. Así que cuando se anuncian, inevitablemente muestran algún precioso y viejo pino y hablan sobre como cuidan y se preocupan del medio ambiente. Fumar mata, así que los anuncios de cigarrillos promocionan vida. En todos esos anuncios se ve gente sonriente, sana y feliz haciendo ejercicio al aire libre.

Cuando Macintosh salió al mercado, no disponía de software. Así que, obviamente, Apple creó un catálogo gigante enumerando todo el maravilloso software «disponible». En la mitad de los artículos del catálogo decía, en letra muy bonita, «en desarrollo», y la otra mitad era o muy cara o inútil, y algunos de esos productos eran tan nefastos que nadie los compraría. Pero ni aún teniendo un voluminoso catálogo en papel satinado, con un producto de software por página lleno de elogios, se podía disfrazar el hecho de que no podías comprar un procesador de texto o una hoja de cálculo para ejecutar en un Macintosh de 128Kb. Había similares «guías de productos de software» para NeXT y BeOS. (Atención, fanáticos de NeXT y BeOS: No quiero ninguna crítica sobre vuestros puñeteros sistemas operativos, ¿de acuerdo? Escribid vuestro propio artículo). La única cosa que una guía de productos de software te dice es que no hay software disponible para ese sistema. Cuando veas una de estas bestias, escapa corriendo en la dirección opuesta.

La lista de «nuevas plataformas» que fracasaron suma y sigue: Amiga, Atari ST, Gem, IBM Top View, NeXT, BeOS, Windows CE, General Magic. La causa es que son *plataformas*: no son muy interesantes en sí mismas sin un apetitoso software que corra en ellas. Pero, con muy pocas excepciones (y estoy seguro que conseguiré llenar un *servidor* con emails de aburridos seguidores de plataformas misteriosas y que nadie quiere, como Amiga o RSTS-11), ningún desarrollador de software con el mínimo de sentido común escribiría intencionalmente para una plataforma con 100.000 usuarios en sus *mejores momentos*, como BeOS, cuando con el mismo esfuerzo podrían crear software para una plataforma con 100.000.000 de usuarios, como Windows. El hecho de que alguien escriba programas para esos descabellados sistemas prueba que el beneficio económico no lo es todo: el fervor religioso sigue vivo y coleando. Bien por ti. Has escrito un bonito clon de microEmacs para Timex Sinclair 1000. Bravo. Tómate algo de mi parte.

Si estás en el negocio de la creación de plataformas, probablemente sufres de lo que se denomina comúnmente como el *problema del huevo y la gallina*. Nadie va a comprar tu plataforma hasta que haya buenos programas, y nadie va a desarrollar programas para tu plataforma hasta que tengas una gran base de usuarios instalada. Oh Oh. Es una especie de nudo gordiano, aunque espiral gordiana de la muerte sería mas descriptivo.

El problema del huevo y la gallina y sus variantes es el elemento estratégico mas importante, y *hay* que que entenderlo. De acuerdo, probablemente se puede vivir sin entenderlo: Steve Jobs prácticamente desarrolló una carrera profesional sin entender el problema del huevo y la gallina. Y dos veces. Pero el resto de nosotros no tenemos el Campo Personal de Distorsión de la Realidad de Steve, así que tendremos que ponernos a trabajar en serio, y estudiar duro.

Lección uno. El dominio o escenario más clásico del problema del huevo y la gallina se encuentra en las plataformas de software. Pero se da en otros escenarios: cada mes, millones de empresas de tarjetas de crédito mandan por correo zillones de facturas a los clientes. La gente llena formularios en papel, los mete en trillones de sobres y los envía por correo. Los sobres se almacenan en grandes cajas y se llevan a los países donde la mano de obra es barata para que sean abiertos y procesados. Pero la operación completa es bastante costosa: la última noticia que tengo es que cuesta más de \$1 por factura.

Para nosotros, chicos sensatos de internet, parece un chiste. «Mándame por email la factura», dices. «La pagaré on-line», añades «Solo costará la 1/100000 parte de un centavo. Te ahorrarás millones». O algo parecido.

Y estás en lo cierto. Así que un montón de empresas han intentado introducirse en ese campo, que es conocido técnicamente como «Entrega de Facturas». Un ejemplo, es (adivina quien) Microsoft. Su solución, Transpoint, es algo parecido a esto: un web site. Vas ahí, te muestra las facturas, y las pagas.

Así que ahora, si te llegan las facturas a través del sistema de Microsoft, tienes que visitar la página cada pocos días y comprobar si las facturas han llegado. Si te llegan, por ejemplo, 10 facturas al mes, no es demasiado lío. En cambio subyace otro problema: hay solo un pequeño número de comerciantes que te facturarán mediante este sistema, así que para el resto de las facturas tendrás que ir a otro sitio.

¿Resultado final? No tiene mucho valor. Me sorprendería si 10.000 personas usasen este sistema. Ahora, Microsoft tiene que ir a los comerciantes y decirles: «¡factura a tus clientes mediante mi sistema!», y los comerciantes dirán «¡Vale! ¿Cuánto me costará?». Microsoft responderá, «¡50 centavos! Que es mucho mas barato que \$1», y los comerciantes dirán, «Muy bien, ¿algo mas?». A lo que Microsoft responderá, «¡Oh si! Costará sobre \$250,000 montar el software y conectar nuestros sistemas a los vuestros y poner todo en marcha.»

Y, dado que Microsoft tiene solo unos pocos usuarios, es difícil de imaginar por qué alguien pagaría \$250,000 para ahorrar 50 centavos con 37 usuarios. ¡Ajá! El problema del huevo y la gallina ha sacado a relucir su cara más fea. Los clientes no aparecerán hasta que haya comerciantes, y los comerciantes no aparecerán hasta que haya clientes. Eventualmente, lo que hará Microsoft es sencillamente salirse de este apuro; para las pequeñas empresas, esto no es una opción. ¿Qué se puede hacer, entonces?

Las plataformas de software nos dan algunas pistas sobre como asar el huevo y la gallina. Recordemos un poco de la historia de las plataformas de computadores personales, desde los años en que apareció el PC-IBM. Puede que saquemos algo en limpio.

La mayoría de la gente piensa que los IBM-PC requerían PC-DOS. Eso no es verdad. Cuando el IBM-PC salió al mercado había tres sistemas operativos a elegir: PC-DOS, XENIX (una versión pelele de UNIX de 8 bits desarrollado, y hablo en serio, por Microsoft), y una cosa llamada UCSD P-System, el cual era, si puedes creerlo, igual que Java: bonito, lento y de código portable, unos 20 años antes que Java.

Ahora, la mayoría de la gente nunca ha oído hablar de cosas raras como XENIX o UCSD. Vosotros, niños, probablemente penseis que esto es debido a que Microsoft se hizo dueño del mercado de sistemas operativos a fuerza de marketing o algo parecido. Absolutamente falso; Microsoft era una empresa débil en esos días. La compañía mas poderosa en el mercado era Digital Research, que tenía un sistema operativo diferente. Entonces, ¿por qué fue PC-DOS el ganador de esta carrera a tres?

Antes de la aparición del PC, el único sistema operativo real del que podías disponer era CP/M, aunque el mercado para los computadores basados en CP/M, que costaba sobre \$10.000, era demasiado pequeño. Eran raros, caros y poco amigables para el usuario. Pero aquellos que lo compraron lo podían usar como procesador de textos, porque podías conseguir un procesador de texto bastante bueno llamado WordStar para CP/M, mientras que el AppleII sencillamente no podía procesar texto (para empezar, no tenía minúsculas).

Ahora, surge un hecho poco conocido: DOS 1.0 fue diseñado con compatibilidad hacia atrás con CP/M *incorporada*. No solo tenía su propio interfaz de programación, conocido por los programadores como INT 21, pero además poseía completa compatibilidad con el viejo interfaz de programación de CP/M. Podía *casi* ejecutar software del CP/M. De hecho, WordStar fue portado a DOS cambiando un *único byte* del código fuente. (Los Verdaderos Programadores pueden decirte exactamente que byte exactamente era; yo lo he olvidado hace tiempo).

Esto es algo que merece la pena mencionar nuevamente. WordStar fue portado a DOS cambiando un *único byte* del código. Asumamos el hecho.

Eso es.

¿Lo lograron?

DOS fue popular porque *tenía software desde el primer día*. Y tenía software porque Tim Paterson había pensado en incluir la característica de compatibilidad hacia atrás en él, porque antiguamente en los años oscuros alguien razonó inteligentemente sobre la problemática del huevo y la gallina.

Pasemos hacia delante. En toda la *historia* de la plataforma PC, ha habido solo dos grandes cambios de dirección en los paradigmas que afectó a casi todos los usuarios de PC: todos migramos a Windows 3.x, y luego a Windows 95. Solo un número muy pequeño de gente cambió a algo más en el proceso. ¿Una conspiración de Microsoft para apoderarse del

mundo? Bien, puedes pensar eso si quieres. Yo creo que es por otra razón, y mas interesante aún, la cual simplemente proviene del problema del huevo y la gallina.

Todos migramos a Windows 3.x. La clave de la frase es el 3. ¿Por qué no todos migramos a Windows 1.0? O Windows 2.0? ¿O Windows 286 o Windows 386, que le siguieron? ¿Es porque a Microsoft le hicieron falta cinco versiones para acertar? **No.**

La razón actual es aún mas sutil que todo eso, y tiene que ver con ciertas características de hardware que Windows 3.0 requería, y que vieron la luz con el chip 80386 de Intel.

- Característica una: los viejos programas de DOS mostraban la salida en pantalla escribiendo directamente a direcciones de memoria que correspondían a celdas en la pantalla. Este era el único modo de hacer que la salida fuese lo suficientemente rápida para que el programa se mostrase correctamente. Pero Windows se ejecutaba en modo gráfico; en los chips de Intel mas antiguos, los ingenieros de Microsoft no tenían otra elección mas que saltar el modo de pantalla completa cuando estaban ejecutándose programas de DOS. Pero en los 80386, podían establecer bloques de memoria virtual y conjuntos de interrupciones, así que se notificaba al sistema operativo siempre que un programa intentaba escribir en memoria de pantalla. Entonces Windows era capaz de escribir el texto equivalente instantaneamente en una ventana gráfica en la pantalla.
- Característica dos: los viejos programas de DOS asumían que disponían del procesador íntegramente. Como resultado de esto, no se ejecutaban bien conjuntamente. Pero el Intel 80386 tenía la habilidad de crear un PC virtual, cada uno de los cuales se comportaba como un 8086 completo, así que los viejos programas de PC podrían aparentar que tenían el computador para ellos, aun mientras otros programas se estaban ejecutando, ellos mismos, emulaban que tenían el computador entero para ellos.

De esta manera, **Windows 3.x en los Intel 80386 fue la primera versión que podía ejecutar múltiples programas de DOS decentemente.** (Técnicamente, Windows 386 también podía, pero los 80386 eran raros y caros hasta que Windows 3.0 vió la luz). Windows 3.0 fue la primera versión que actualmente podía trabajar razonablemente con todo el software antiguo.

¿Windows 95? Sin problemas. Nueva y bonita API de 32 bits, pero todavía ejecutaba viejo software de 16 bits perfectamente. Microsoft, obsesionado sobre esto, gastó buena parte del presupuesto probando todos los viejos programas que se podían encontrar en Windows 95. Jon Ross, el cual escribió la versión original de Sim City para Windows 3.x, me comentó que accidentalmente dejó un fallo en Sim City donde leía memoria que ya había sido liberada. Funcionó perfectamente en Windows 3.x, porque la memoria no iba a ninguna parte. Ahora viene la parte interesante: en la versión beta de Windows 95, Sim City no funcionaba en las pruebas. Microsoft localizó el fallo y añadió un código específico para Windows 95 que comprueba si está presente Sim City. Si encuentra que Sim City se está ejecutando, se ejecuta el localizador de memoria en un modo especial que no libera memoria de ninguna manera. Esa es la clase de obsesión en la compatibilidad hacia atrás que hizo que la gente deseara actualizarse a Windows 95.

Ya deberías tener algunas ideas sobre como romper el problema del huevo y la gallina: proporciona un modo de compatibilidad hacia atrás que entregue o bien una carga de gallinas o bien una carga de huevos, según como lo mires, y espera sentado mientras te forras.

Volvamos a la entrega de facturas. ¿Recuerdas la entrega de facturas? El problema del huevo y la gallina aquí es que sólo puedes obtener las facturas de Con Ed, así que no usarás el servicio. ¿Cómo se puede resolver? Microsoft no podía entenderlo. PayMyBills.com (y media docena de «startups» de Silicon Valley) se dieron cuenta al mismo tiempo. Proporcionas un modo de compatibilidad hacia atrás: si el comerciante no soporta el sistema, simplemente consigue que mande su coñazo de facturas en papel a University Avenue, en Palo Alto, donde un montón de gente las abrirá y las escaneará. Ahora puedes tener todas tus facturas en su sitio web. Desde que todos los comerciantes sobre la faz de la tierra están disponibles en el sistema, los clientes están felices de usarlo, aunque se lleve a cabo este modo extraño de compatibilidad hacia atrás donde los estúpidos bancos miembros de Visa envían la factura electrónicamente a una impresora, la imprimen en papel, la meten en un sobre, la envían 1500 millas a California, donde el sobre es abierto, la publicidad que inevitablemente los bancos incluyen en estos sobres anunciando cosas inútiles es botada a la basura, la factura es escaneada en una computadora y finalmente colgada en la web, donde debería de haber sido enviada en un principio. Pero este estúpido modo de compatibilidad hacia atrás eventualmente siguió adelante, porque PayMyBills.com, a diferencia de Microsoft, puede actualmente conseguir clientes que usen su sistema. Así que en breve podrán ir a los estúpidos bancos miembros de Visa y decir «Oye, he conseguido 93.400 de tus clientes. ¿Por qué no os ahorrais \$93.400 cada mes conectando con nosotros?» Y de pronto PayMyBills.com tiene ganancias, mientras Microsoft está todavía tratando inútilmente de conseguir clientes. Quizás logren firmar un trato con una segunda compañía eléctrica. Una ubicada en Florida, para cambiar un poco el ritmo.

Se puede pensar en que las empresas que fallan en reconocer el problema del huevo y la gallina intentan hervir el agua del mar: sus negocios necesitan 93.000.000 de personas cooperando con su insano plan de negocios antes de que realmente funcione. Una de las ideas más extravagantes que me ha tocado encontrar se llamaba ActiveNames. Su idea consistía en que cada persona en el mundo instalaría un pequeño añadido en su cliente de correo, el cual buscaría los nombres de la gente en sus servidores centrales para conseguir su verdadera dirección de correo electrónico. Luego, en vez de decirle a la gente que tu dirección de correo es beto@plazasesamo.com, le dirías que tu ActiveName es «spolsky», y que si te quieren mandar un correo electrónico deben instalar este software especial. Respuesta incorrecta. Hoy en día, aún no puedo empezar a enumerar las razones por las cuales esta idea nunca funcionará.

Conclusión: si te encuentras en el mercado con el problema del huevo y la gallina, mejor proporciona una respuesta con compatibilidad hacia atrás que solucione el problema, o te va a llevar un largo tiempo el despegar (algo así como para siempre).

Hay un montón de empresas que se dieron cuenta del problema del huevo y la gallina y lo resolvieron inteligentemente. Cuando Transmeta sacó su nuevo procesador, fue la primera vez en mucho tiempo que una compañía que no fuese Intel admitió finalmente que si eres un procesador, y quieres que un zillón de personas te compren, tienes que ejecutar código x86. Esto después de que Hitachi, Motorola, IBM, MIPS, National Semiconductor, y quien

sabe cuantas otras compañías se engañasen a ellas mismas pensando que inventando un nuevo conjunto de instrucciones iban por el camino correcto. La arquitectura de Transmeta asume desde el primer día que cualquier plan de negocios que pase por construir un computador en el que no se pueda ejecutar Excel no va a ninguna parte. Por Joel Spolsky