

Especializar, u homogenizar. Cuestión de experiencia

David Gómez-Rosado, San Francisco. Estados Unidos

Tengo que reconocerlo: Uno de los aspectos más interesantes de nuestra práctica de experiencia de usuario es la capacidad que todavía tenemos de aceptar debate y de tomar pocas pautas como finales y decisivas. ¡Trata de hacer eso en arquitectura, por ejemplo!

Las bases tecnológicas de nuestro campo avanzan constantemente (hasta el punto de dejar nuestros conocimientos en la irrelevancia apenas han sido adquiridos y siquiera tenido tiempo de ser asimilados) ¿Quién se atreve todavía a promover la paleta *web-safe* para descargas más rápidas en las imágenes? (Si sabes lo que es “Debabelizer”, entiendes lo que significaba ese penoso ejercicio). ¿Quién apoya aún que todos los enlaces de hipertexto han de ser subrayados y en azul? (*No insistas, déjalo, de verdad, que nos están mirando*). La dura realidad es que las innovaciones son pronto convertidas en convenciones por un públi-

co que aprende a la misma velocidad que quienes las realizamos. Pretender que las verdades se mantienen estáticas y basar nuestro conocimiento enteramente en experiencia acumulada es hacerse objeto de risas piadosas por generaciones venideras... Sí, vosotros los de veintipoco (sabéis quiénes sois por que aún *celebráis* los cumpleaños). La apuesta de la experiencia de usuario es hacia adelante, no hacia atrás... Difícil de implementar cuando los datos de soporte en nuestra práctica se basan en estudios de anteaer (*¡Los pop-ups son malos, los framesets también!*). Cada vez que sale una Wii ó un iPhone al mercado, tenemos que salir corriendo a reajustar todas nuestras creencias profesionales.

Es duro... Así que empecemos con aquello de: “Sólo sabemos que no sabemos nada”.

Aceptado eso (*¿A que se siente uno mejor?*)... y aún así, voy a atreverme a dar una opinión: entre todos los debates que he encontrado a lo largo de mi carrera profesional, uno de los más interesantes (porque excede la práctica evolutiva de la experiencia de usuario) y que invariablemente surge en TODAS las compañías con las que he colaborado, es el compromiso de la optimización de una interfaz de usuario con la labor específica del programa... o la adopción en esta interfaz de convenciones homogéneas al ecosistema de programas interoperativos al que pertenece (*suite*, familia o sistema operativo).

Especializar... u homogenizar. He ahí el origen de mucho sudar.

Por ejemplo: ¿Debería Microsoft Word presentar un sistema de controles único para sus usuarios objetivos, optimizado exclusivamente para la manipulación de textos?... ¿O, por el contrario, debería reconocer y adoptar el mínimo común denominador para que la funcionalidad de Word se acople lo más posible a Excel, PowerPoint, Internet Explorer... o ya puestos, productos de Microsoft en la Web, como Live.com?

Lo cierto es que una decisión de ese calibre representa miles de horas de trabajo por un lado (innovar y construir desde cero) o por otro (atenerse a dependencias y obedecer métricas de estandarización).

Yo tengo una clara posición al respecto... pero hagamos un recuento de los beneficios y costes de adoptar cualquiera de las dos filosofías.

A. Ventajas de sistemas especializados

A1. Cuestión de prioridades

Está claro que la interfaz de un producto para una tarea especializada puede ahorrar muchos pasos en principio innecesarios al usuario. Por ejemplo: si un programa para la visualización y gestión de documentos y faxes tiene como fin último el imprimir, es razonable que tal función sea prominente y prioritaria

sobre otras. Por ejemplo, colocar en vez de un cursor, un ícono de impresora que sobrevuele cualquier documento y que con sólo hacer un clic lo envíe a la impresora. Quizás incluso subir el comando “Imprimir” al primer nivel de un menú principal, quizás sacarlo enteramente y posicionarlo en la esquina más ventajosa bajo la ley de Fitts. En el programa de correo de voz, por otro lado, el “escuchar” es lo más importante... y quizás el botón de “Reproducir” sea el que necesita de esos espacios privilegiados.

A2. Valor legible

Una interfaz diseñada para la especialización tiene otra gran ventaja: sus funciones principales son claramente legibles para el público objetivo. Aquellas capacidades que lo hacen especial, flotan en la superficie y el usuario novato es capaz de entender inmediatamente qué es lo que hace ese programa. Su “ergonomía” es aparente en su propia forma (el “mapping” de Donald A. Norman). Y principalmente: es fácil venderlo por los ojos. Un gran botón con un triángulo ladeado en el medio... y ¡bum! (para citar a Steve Jobs) ya sabes que puedes reproducir contenidos multimedia.

A3. Misión Crítica

En aquellos programas dedicados a una sola misión crítica, donde cualquier malentendido pueda costar vidas (software en hospitales, centrales nucleares, armamento, dosis de peta en los Peta-Zetas...) es de suponer que la eficiencia y la fiabilidad en la tarea prime por encima de todo... Aunque el coste de no asumir estándares puede que signifique mayor incidencia de error, el hecho de que haya que dar dos pasos en vez de cuatro puede ahorrar segundos críticos para una decisión rápida.

A4. ...

Hay más beneficios, por supuesto. Pero ya me cansé de hacer de abogado del diablo. Al grano. Esta es mi opinión:

El debate surge al analizar qué es más importante para el trabajo de cualquier “obrero de la información”... ¿Somos tan especializados que durante todo el día sólo operamos con un solo programa, en una sola función y en un solo entorno, o por el contrario, somos seres que prosperan y optimizan alternando múltiples tareas? Si esta última premisa es más frecuente, pensemos entonces en la ineficiencia de tener que aprender y alternar entre una docena de programas independientes, diferenciados, especializados, con botones y funciones diferentes.

B. Realidad de sistemas “especializados”

B1. ¿Optimización? ¿O lucha de poder?

En discusiones sobre este tema, normalmente el defensor de la optimización de una interfaz es invariablemente el dueño del proyecto que dará un producto interactivo como resultado. Para este responsable, las dependencias y compromisos por atenerse a normativas de UX y guías de estilo, no son más que complicaciones fuera de su control. Esa labor global, por propia naturaleza, es independiente de cualquier producto singular y normalmente externa a cualquier proyecto. Por lo tanto, en el mundo del Gestor de Proyecto, tan sólo significa retrasos, coste y futuros problemas ¡La consistencia no viene barata!

Por otro lado, la pasión por la misión enfocada, libre de obligaciones y compromisos, es sexy, muy sexy. Si a eso le sumamos la agradable creencia de tener la solución mejor y el ansia natural por aportar algo nuevo... Todo conduce a querer proponer nuevas metáforas constantemente.

A un Director o Gestor de Proyecto (que puede no permanecer más que unos pocos años en la misma compañía) no se le premia por los éxitos invisibles de larga duración, si no por el impacto tangible a corto plazo. ¿A quién le importa si el programa no escala bien en

complejidad dentro de dos años o no se integra en una creciente familia de servicios en muchos más?... Lo que cuenta es que en su lanzamiento su interfaz sea llamativa y despierte la curiosidad en sus consumidores potenciales. Compañías del calibre de Microsoft invierten en esa curiosidad por lo nuevo. ¿Sabéis el slogan de la campaña para anunciar los beneficios de Vista?: “Wow”... Que en España se traduciría como “Guau”, sonando casi igual de simplista.

B2. La política como determinante de GUI

Es pasmosamente fácil averiguar la política interna de una compañía con sólo ver sus productos. Cuanto más disgregada sea esta compañía en pequeños reinos de Taifas, más independientes y menos interconectados serán sus productos. Una compañía regida por una visión semi-dictatorial dará como resultado, por supuesto, una familia de productos más coherentes. Se pueden establecer relaciones claras con las compañías grandes de nuestro sector para probar la validez de esta teoría. Lo que puedo garantizar a pies juntillas, es que en las compañías más respetadas de la industria, el diseño de interfaz es influenciado por la política interna tanto como por las necesidades reales de los usuarios y el mercado... incluso más.

B3. Aplicaciones arrogantes

Políticas como las expuestas anteriormente dan lugar a “interfaces arrogantes”. Estos programas son fáciles de distinguir: Suelen saltarse a la torera las normas establecidas de interfaz y exigir del usuario que se adapte a su manera de entender las cosas (ya desde 2005 se utilizó mucho ese término para aquellos programas de prácticas abusivas: <http://blogs.pcworld.com/staffblog/archives/000975.html> ó <http://rabidpaladin.spaces.live.com/blog/cns!9EFB431594179295!292.entry> ó http://www.readwriteweb.com/archives/gmailgoogle_bac.php). Estos programas y sus autores hablan de “revolución” y no de “evolución”. Toman decisiones en nombre del usuario, empujan preferencias y en vez de

proponer, imponen... Lanzan interfaces de usuario a veces llamativas pero siempre ininteligibles, escudadas tras adjetivos como “único” e “innovador”; pero lo que logran principalmente es confundir al usuario, y en muchos casos incluso enfurecerle, pues parte de su actitud soberbia es saltarse los estándares y las convenciones en los que se apoya el consumidor.

Estos programas parecen bravuconear: “Yo y sólo yo sé cómo realmente se opera con la información de manera óptima, todos los demás se equivocan y mis procedimientos innovadores pronto se establecerán como la nueva norma”.

C. Ventajas de sistemas homogéneos

C1. Objetividad

Una ventaja de la consistencia es su claridad en las métricas de éxito. Mientras que “óptimo” es siempre cuestión de debate y gustos personales... “Consistente” es tan fácil como comparar el nivel de dos vasos de agua. Traducción: o el botón aparece en la misma esquina en cada pantalla de cada programa, o no. Punto. El que ésta sea la esquina adecuada sí se puede debatir largamente.

Muchos dirán que eso no es un beneficio, que “consistencia” en repetir un procedimiento equivocado es peor que atreverse a apostar por algo nuevo con mucho análisis por detrás.

El problema es que esa forma de pensar puede dar lugar a una amplia gama de opciones, todas ellas apostando a que son la “mejor solución” (y muchas de ellas sin garantía de estar en lo correcto)... Por otro lado, ser consistente en una percepción errónea no es mayor problema: la primera vez que un usuario percibe la ambigüedad, prueba, se equivoca, pero luego la corrige y subsecuentemente es capaz de traducir el mensaje o expectativa correcta para no tropezar dos veces con la misma piedra. La capacidad humana por adaptarse a un medio estable está bien documentada:

si hace demasiado frío, uno se abriga. Esto no funciona si el medio es inconsistente (como en San Francisco, donde uno no sabe si salir en playera o con bufanda).

C2. Conocimiento reciclado

¿Un programa con interfaz optimizada se beneficiará con la especialización de sus principios de experiencia de usuario?

No necesariamente. Si nos apoyamos en una analogía, lo mismo se podría decir de un camión o un taxi. Ambos vehículos son altamente especializados para tareas específicas. El primero transporta gente, principalmente por calles urbanas y trayectos cortos. El segundo transporta materiales por autopistas y largas distancias. Aun así, comparten una gran proporción de su interfaz, ambos se manejan dando vueltas a un aro y apretando pedales similares; ambos se rigen por reglas de tráfico parecidas, pues comparten una plataforma común (sistema de calles y carreteras)... Y lo más importante: ninguno requiere un exceso de aprendizaje adicional (lo cual minimiza la posibilidad de error), pues sus operarios pueden reciclar los mismos conocimientos cuando conducen la vagoneta familiar durante los fines de semana. Esta eficiencia se debe al reciclaje o traslado de aprendizaje. El usuario aprende cómo funciona un volante y su relación con las ruedas una vez y luego lo aplica a muchas situaciones diferentes.

Lo mismo puede ser aplicado a un programa de edición de vídeo u otro de arquitectura. Es tentador proclamar que modificar el vocabulario de sus funciones es beneficioso para cada tarea particular, pero quienes piensan así se olvidan de que sus operarios también tienen que utilizar media docena de otras aplicaciones para completar su día-a-día, y que el aprendizaje requerido para cada “optimización” requeriría una memorización exponencial de diferentes alternativas en el vocabulario de comportamientos interactivos y una habilidad envidiable para cambiar lenguajes visuales mientras se salta de una a otra... ¿O acaso el arquitecto no tiene que exportar el plano de un edificio a un gráfico EPS, para luego abrirlo en Illustrator y retocar ciertos colores, para luego reducirlo, guardarlo como

JPEG y añadirlo a un documento Word, luego escribir ciertas especificaciones al margen, para luego activar un programa de correo y enviarlo a sus socios y finalmente abrir un servicio online de blog y anunciar el proyecto al público? Si cada programa tuviese una manera diferente de “copiar y pegar” o una localidad diferente para abrir y exportar (“por no ser el escenario objetivo”) la experiencia de ese arquitecto sufriría a cada paso y las diferentes filosofías de interfaz entorpecerían el proceso. ¡Suficiente castigo divino tienen los arquitectos con luchar contra Autocad!

Moraleja: Cualquier ganancia en la optimización en una funcionalidad específica, es contrarrestada por la necesidad de aprenderla y contrastarla con otras similares.

C3. Eficiencia

“Convención” es igual a “Eficiencia” en la Web. Humana e infraestructural.

Ya que seguimos la analogía de la “interfaz de los vehículos”... sigamos con su siguiente ventaja: ¿No es mágico que uno pueda viajar a cualquier confín de la tierra, alquile un coche de cualquier marca y en cuestión de segundos esté utilizándolo sin mayores complicaciones? (variaciones en el UX de un coche, por otro lado ¡producen muertes!). ¿Por qué no puede ocurrir esto con el software? ¿Por qué un simple cambio de versión en Microsoft Office 2003 a la versión 2007 requiere semanas (si no meses) de re-aprendizaje de su interfaz? (no tan dramático como accidente de un coche, pero lo suficiente como para que ese cambio de UX se identifique como el mayor problema del programa por parte de la prensa más reconocida en la industria).

Por supuesto: mereció la pena cambiar menús a barras de herramientas porque claramente ésta es una versión mejor... Qué pena que Internet Explorer o incluso Outlook no sigan el mismo pensamiento (claro, ellos están optimizados para OTRAS tareas) y qué pena que el resto de la industria no pueda cambiar también... Para cuando lo hayan logrado, otra revolución ya estará en camino y millones de horas acumuladas malgastadas. ¿Os podéis imaginar si el Ford

Escort de 2007 decidiera que es mejor conducir desde el medio del asiento de atrás y con un joystick? ¿Y que luego cada coche tuviese su propia brillante idea de lo que es óptimo?

Este tipo de revoluciones sólo se pueden permitir cuando el producto representa la industria entera. Un buen caso es el iPhone. La experiencia táctil es lo suficientemente innovadora para el público general. Pero como el aparato controla todas las aplicaciones dentro de su entorno, lo que importa es que éstas sean consistentes para permitir la interoperabilidad. Es decir, los aviones pueden diferir en su conducción a la de un coche porque tienen otro entorno (incluyendo otra dimensión para empezar). La especialización se justifica. Eso sí, todos los aviones también deberían comportarse de manera similar.

Hasta ahora sólo hemos hablado de cómo la convención en procesos de UX ayuda en la eficiencia operativa del humano (trasladando aprendizaje de un programa a otro). Pero otro beneficio es claro: la consistencia sistemática ayuda enormemente en la producción de la infraestructura, tanto en el diseño de la señalética del interfaz (iconos, símbolos, menús, botones) como en la estructura de programación que lo sustenta.

Ya que éste es un artículo de opinión, he aquí la mía: las interfaces se benefician de un lenguaje visual consistente, tanto como una sociedad lo hace con un idioma común o una autopista lo hace con una señalética normalizada. Obviamente ningún sistema está congelado en el tiempo y nuevas tecnologías, costumbres e innovaciones van surgiendo. El sistema de navegación va adaptándose y proveyendo nuevas posibilidades. Pero raramente es éste un proceso revolucionario, si no más bien evolutivo... Las convenciones han de ser usadas a favor de la innovación, incluso como catalpa de aquellos nuevos comportamientos.

D. Realidad de sistemas “homogéneos”

La verdad es que la consistencia va más allá de los beneficios expuestos. Existen varios niveles de intero-

perabilidad y un buen programa habría de respetarlos todos. Por orden de exigencia en su similitud, podemos analizar tres grandes dimensiones:

Nivel Físico (Infraestructural)

1. Terminal o Plataforma
2. Sistema Operativo
3. Lenguaje de Programa

Nivel Comercial (Estratégico)

4. Marca
5. Suite o Familia

Nivel Social (Antropológico)

6. Cultura
7. Convenciones
8. Leyes y Normas

Consistencia a Nivel Físico (Infraestructural)

D1. Terminal o Plataforma

El primer nivel de consistencia se basa en el tipo de terminal. Es decir, la plataforma común en la que un conjunto de programas son operados. Esto se debe a que normalmente la tecnología de cada ámbito rige el formato de la experiencia y sus posibilidades (tamaño de resultados, señal de repuesta y herramientas de acceso). En un móvil Palm, por ejemplo, todos los programas aceptan manipulación de menús por selección táctil, porque la tecnología lo permite (en un portátil no)... En un iPhone también (al contrario que un Mac, a pesar de utilizar el mismo sistema operativo modificado). Es decir, aunque el sistema operativo estipula, la terminal rige físicamente la experiencia. Es natural que las diferentes tecnologías disponibles (para el input y el output) modulen por sí mismas una experiencia consistente.

“Cuando tienes un martillo, todos los problemas te parecen clavos”.



D2. Sistema Operativo

El segundo nivel de consistencia lo marca el sistema operativo. Independiente de la estrategia comercial, es casi obligado reparar en él, por la naturaleza compartida de los recursos que ese sistema otorga. Programar una función independiente de los recursos disponibles es mucho más costoso que adoptar el establecido por la arquitectura del sistema. Por ejemplo, la mayoría de los productos Mac utilizan la misma actividad para elegir colores o tipografía. Eso promueve la consistencia en la experiencia de usuario. En Windows XP no es así, pues su arquitectura no ha establecido un recurso general para esta actividad.

D3. Entorno de desarrollo

Aunque parecido al Terminal, el entorno de desarrollo seleccionado para crear programas, da también un “sabor específico” a la experiencia... Ya sea C#, Java o DHTML. Las librerías disponibles en cada lenguaje vienen ya predispuestas en cierta manera con las interacciones elegidas por el entorno. Estas prestaciones son (o no) provistas por una librería común. Por ejemplo, en el sistema Java, todos los aspectos del sistema operativo gráfico son llevados por las Java Foundation Classes (JFC), que incluyen los recursos complementarios Abstract Window Toolkit (AWT), Swing y Java 2D. Muchas veces (como el caso de Java) estas librerías funcionan por encima de las del sistema operativo y es fácil identificar con qué ingredientes fue elaborado un programa por las consistencias con otros programas hechos con los mismos elementos.

Consistencia a Nivel Comercial (Estratégico)

D4. Marca

Hay consistencias *impuestas* o explícitas por el medio físico (Sistema Operativo, Terminal, Lenguaje). Pero luego existen importantes consistencias que surgen de la elección implícita de una compañía de dejar su

“sello” característico en la manera especial de hacer las cosas. Apple, por ejemplo, ha logrado imponer un “estilo” de interacción homogéneo a lo largo de sus productos y fácil de identificar incluso a través de terminales (Mac, iPods, iPhones, Apple TVs, todos comparten por ejemplo el mismo sistema de “navegación jerárquica horizontal”). Esto es común también en compañías de productos físicos. Aquellos consumidores que hayan experimentado el dial de selección en productos Sony, la rueda infinita en los productos Apple o la de las cámaras Canon saben a lo que me refiero.

Las razones tras estas consistencias se deben a muchas estrategias básicas de marketing, pero principalmente se basan en que es fácil crear marca y dar presencia de solidez y confianza tras expectativas consistentes a lo largo de las ofertas de una corporación. La *experiencia* de una hamburguesa McDonalds tiene que ser la misma no importa dónde se adquiera... Un efecto en nada banal y aleatorio, pues todos los aspectos (desde el sabor, a la textura o la temperatura) son extremadamente prescritos y vigilados para lograr ese efecto de seguridad y familiaridad asociado a su marca.

D5. Suite (o Familia de Programas Integrados)

En una Suite de productos, la homogeneidad de su interfaz debería ser absoluta. De ello depende el éxito comercial de la estrategia de integrar los productos en un solo “bundle”. La razón de ser de una suite es precisamente la agregación de funcionalidad compartida y sus recursos integrados. Cualquier desviación de ese beneficio básico es evitada. ¡Y no existe motivo en contra de comprar productos por separado! (a no ser el precio reducido, pero muchas compañías miran más allá del coste inicial y saben que un sistema no integrado conlleva más coste de implementación y mantenimiento).

Ejemplos de Suite:

- **Office 2007 Ultimate:** Access 2007, Accounting Express 2007, Excel 2007, InfoPath 2007, Gro-

ove 2007, OneNote 2007, Outlook 2007 with Business Contact Manager, PowerPoint 2007, Publisher 2007

- **Adobe Creative Suite 3 Design Premium:** Adobe InDesign® CS3, Adobe Photoshop® CS3 Extended, Adobe Illustrator® CS3, Adobe Flash® CS3 Professional, Adobe Dreamweaver® CS3, Adobe Acrobat 8 Professional
- **iLife '06:** iPhoto, iMovie HD, iDVD, GarageBand, iWeb

Consistencia a Nivel Social (Antropológico)

D6. Cultura

Es obvio que cada cultura implica ciertas consistencias debido a sus costumbres, su alfabeto o su lenguaje. La metáfora de un buzón de correo americano es muy diferente a la de uno europeo; lo mismo ocurre con la señal de peligro, de información, etc., varía de país a país. El significado tras los colores también, el formato de escritura de derecha a izquierda o viceversa, incluso el diseño de sus caracteres establecen pautas de maquetado y coloración diferentes (por ejemplo, el Kanji japonés necesita colores menos saturados, más espacio en los márgenes y mayor interletrado para conseguir la misma legibilidad que un mensaje en alfabeto romano, debido a su densidad de detalle).

Es de esperar, por lo tanto, que gran parte de la consistencia sea conducida por la cultura de la audiencia del producto. Al igual que la Real Academia Española trata de mantener una consistencia en la lengua castellana para que una sociedad pueda prosperar y sofisticarse, hay consistencias en las experiencias de usuario que deben alimentar la estandarización.

D7. Convenciones

Aunque similares al aspecto cultural, las convenciones

tienen un espectro más transitorio y mutable. Son de hecho las “modas” adoptadas por un público como “normales” en un momento dado en los procesos de interacción. Las *convenciones* son constantemente alimentadas por *innovaciones* y por lo tanto resultan bastante cambiantes: hace unos años dar al botón derecho del ratón en una página Web no generaba efecto alguno, misma reacción fútil si se agarraba un objeto en ésta y se arrastraba a otro extremo. Gracias a avances en los estándares de diseño, estos comportamientos son ahora posibles y cada vez más comunes. La consistencia por razones de convención es inmediatamente visible con la perspectiva que nos otorga la distancia. Para un profesional del medio, es rápidamente identificable una interfaz de los años 90 o una actual: La falta de contenido y funcionalidad forzaba a estas primeras a “rellenar” la experiencia con mayor adorno en los controles (sombras, imitación 3D, etc.)... Curiosamente, tras un periodo de minimalismo en las interfaces gráficas, volvemos a recursos parecidos, pero esta vez por la posibilidad computacional de los ordenadores cada vez más capaces. Hay tanta potencia disponible que es tentador flexionar los músculos de la interfaz visual (los efectos parecidos de los 90 eran meros efectos “faux” al mejor estilo “Trompe l’oeil”)

D8. Leyes y Normativas

Hay leyes vigentes con tanta influencia que de cierta manera establecen nuevas convenciones de manera forzada y nuevas expectativas por parte de los usuarios. Por ejemplo, en España existe la ley 34/2002, de 11 de julio, de servicios de la Sociedad de la Información y de comercio electrónico (LSSICE), que exige accesibilidad para las personas con discapacidad y de edad avanzada a la información proporcionada por medios electrónicos. Esta ley en sí misma reprime ciertas exploraciones en artefactos de navegación y promueve otros más establecidos. La falta de consistencia puede acarrear graves consecuencias financieras (por ejemplo, perder el negocio lucrativo con entidades gubernamentales para empezar, o multas potenciales en ciertos países para terminar). Lo curioso es que estas normas, aunque muchas veces aplicables a un solo país, pueden afectar a la experiencia de usuario alre-

dedor del mundo por lo difícil que es segmentar el desarrollo de productos. Que le pregunten a Microsoft, forzada a cambiar sus prácticas incluso en Estados Unidos por causa de leyes europeas.

E. Conclusión

En resumen, y a riesgo de sonar a cascarrabias en contra de lo nuevo: La falta de consistencia en productos, aparte de hacer peligrar la paciencia del usuario, también puede impactar negativamente la eficiencia infraestructural, la estrategia de marca, la percepción de calidad, la comunicación de valor añadido, la legibilidad, y ya puestos, incluso la legalidad.

Si uno quiere escudarse en la “*innovación*” para justificar la falta de disciplina, merece la pena asegurarse que efectivamente cada atentado contra la consistencia sea ampliamente justificado con un beneficio claro. El resultado de este ejercicio exhaustivo resultará en un cambio hacia lo mejor.

Referencias

<http://www.nathnac.org/travel/factsheets/personal.htm>

Page SJ, Meyer D. Tourist accidents: an exploratory analysis. *Ann Tourism Res* 1996;23:666-690.

Sniezek JE, Smith SM. Injury mortality among non-US residents in the United States 1979-1984. *Int J Epidemiol* 1991;20:225-9.

http://reviews.cnet.com/office-suites/microsoft-office-standard-2007/4505-3524_7-32024133.html?tag=rev

<http://www.pcworld.com/article/id,123773-page,1/article.html?RSS=RSS>

<http://ptech.allthingsd.com/20070104/redesign-improves-office-2007/>

http://online.wsj.com/public/article/SB116786111022966326-mRGBGB2hIZH7SbdbE2Bg7YVHTDE_20070202.html?mod=tff_main_tff_top



David Gómez-Rosado es director de experiencia de usuario en Ask.com. Su carrera en el desarrollo de interfaces suma 13 años, con experiencia en diferentes industrias y distintos continentes. Ha trabajado como director de experiencia de usuario en Microsoft, y director creativo en Nike.com, HarrisDirect.com y Apogee Networks; jefe de desarrollo de Interfaz en Icon-Medialab; jefe creativo en Whitestone Networks y presidente de Lobe.



The Latest in Fashion - Elise Milley