

O que é Web 2.0

Padrões de design e modelos de negócios para a nova geração de software¹

30/09/2005

Tim O'Reilly²

O estouro da bolha das empresas ponto-com no outono de 2001 marcou uma virada na web. Muitos concluíram que a web tinha recebido uma publicidade exagerada quando, na realidade, bolhas e conseqüentes reorganizações parecem ser um traço comum a todas as revoluções tecnológicas. Tais crises tipicamente assinalam o momento em que uma tecnologia ascendente está pronta para assumir seu lugar no centro da ação. Impostores são eliminados, as histórias de verdadeiro sucesso mostram sua força e começa a haver uma compreensão sobre o que distingue um caso do outro.

O conceito de “Web 2.0” começou com uma conferência de *brainstorming* entre a O'Reilly e a MediaLive International. Dale Dougherty, pioneiro da web e vice-presidente da O'Reilly, notou que, ao contrário de haver explodido, a web estava mais importante do que nunca, apresentando instigantes aplicações novas e sites eclodindo com surpreendente regularidade. E, o que é melhor, parecia que as companhias que haviam sobrevivido ao colapso tinham algo em comum. Será que o colapso ponto-com marcou uma espécie de virada que deu sentido a uma convocação do tipo “Web 2.0”? Achamos que sim e, desse modo, nasceu a Conferência Web 2.0.

No ano e meio que se seguiu, o termo “Web 2.0” claramente se consagrou com mais de 9,5 milhões de citações no Google. Mas ainda existe um enorme desacordo sobre o que significa Web 2.0, com alguns menosprezando a expressão – como sendo um termo de marketing sem nenhum sentido e outros aceitando-a – como a nova forma convencional de conhecimento.

Esse artigo é uma tentativa de esclarecer o que queremos dizer com Web 2.0.

No *brainstorming* inicial, formulamos nossa idéia de Web 2.0 através de exemplo:

¹ Publicado em O'Reilly (<http://www.oreilly.com/>) Copyright 2006 O'Reilly Media, Inc.

Tradução: Miriam Medeiros. Revisão técnica: Julio Preuss. Novembro 2006

² Presidente e CEO. O'Reilly Media, Inc., Tim@oreilly.com

Web 1.0		Web 2.0
DoubleClick	-->	Google AdSense
Ofoto	-->	Flickr
Akamai	-->	Bit Torrent
mp3.com	-->	Napster
Britannica Online	-->	Wikipedia
Sites pessoais	-->	blogs
evite	-->	upcoming.org e EVDB
Especulação com nomes de domínio	-->	otimização para ferramenta de busca
page views	-->	custo por clique
"Screen scraping"	-->	serviços web
publicação	-->	participação
Sistemas de gerenciamento de conteúdo	-->	wikis
diretórios (taxonomia)	-->	tags ("folksonomia")
stickness	-->	syndication

A lista era interminável. Mas o que nos fazia identificar um aplicativo ou abordagem como "Web 1.0" e outro como "Web 2.0"? (A pergunta é especialmente premente porque a noção de Web 2.0 tornou-se tão disseminada que atualmente companhias estão usando o termo como uma palavra-chave de marketing sem realmente entender o que quer dizer. É particularmente difícil porque muitas dessas novas empresas viciadas na palavra-chave definitivamente *não* são Web 2.0 e alguns dos aplicativos que identificamos como Web 2.0, como o Napster e o BitTorrent nem mesmo são verdadeiros aplicativos web!) Começamos por tentar trazer à tona os princípios que, de alguma forma, são demonstrados por histórias de sucesso de web 1.0 e pelas novas aplicações mais interessantes.

1. A Web como plataforma

Como muitos conceitos importantes, o de Web 2.0 não tem fronteiras rígidas mas, pelo contrário, um centro gravitacional. Pode-se visualizar a Web 2.0 como um conjunto de princípios e práticas que interligam um verdadeiro sistema solar de sites que demonstram alguns ou todos esses princípios e que estão a distâncias variadas do centro.

MAPA de NOÇÕES WEB 2.0

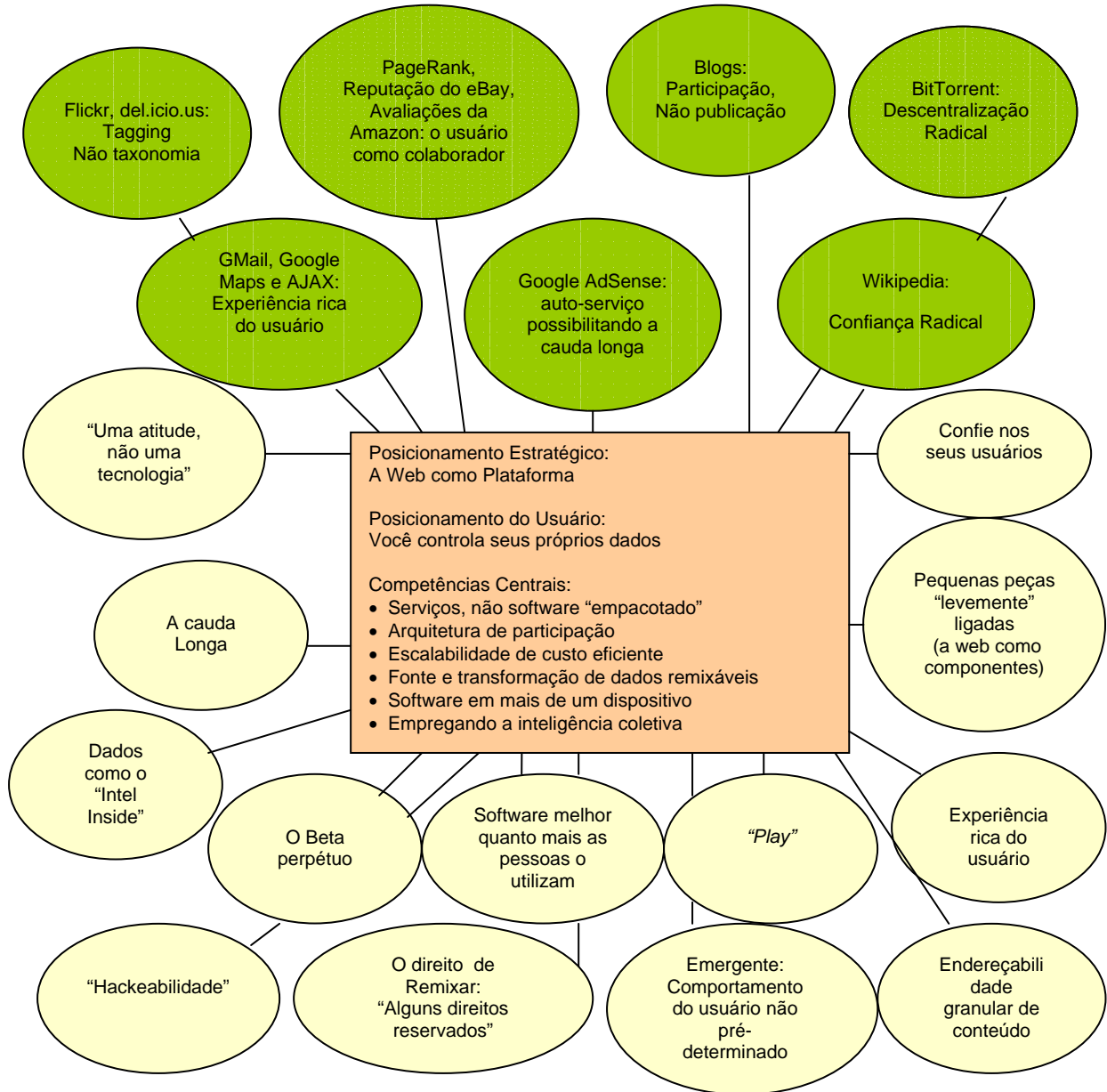


Figura 1

A figura 1 mostra um mapa de noções de Web 2.0 desenvolvido durante uma sessão de *brainstorming* durante a FOO Camp, uma conferência na O'Reilly Media. É um trabalho ainda em processo mas que mostra as várias idéias que irradiam do centro da Web.2.

Por exemplo, na primeira conferência de Web 2.0, em outubro de 2004, na palestra de abertura, John Battele e eu fizemos uma lista preliminar de princípios. O primeiro deles era “A web como uma plataforma”. Entretanto esse também era um lema da queridinha da Web 1.0, a Netscape, que pegou fogo depois de uma inflamada batalha com a Microsoft. Além disso, dois dos nossos exemplos de Web 1.0, a Double Click e a Akamai, eram ambas pioneiras em tratar a rede como plataforma. Não é freqüente as pessoas pensarem nelas como “serviços de rede” mas, de fato, o primeiro serviço web amplamente distribuído foi o veiculador de anúncios, assim como também ocorreu com os *mashups*³ (para usar um outro termo que vem ganhando terreno ultimamente). Cada anúncio em forma de banner é oferecido como uma cooperação sem costuras entre dois websites, ao apresentar uma página integrada ao leitor em um outro computador. A Akamai também trata a rede como plataforma e, em um nível mais profundo da pilha, construindo um sistema de entrega de cache transparente que desafoga o congestionamento da banda larga.

Entretanto, esses pioneiros forneceram contrastes úteis porque os que entraram depois levaram ainda mais longe suas soluções para o mesmo problema, compreendendo mais a fundo a natureza da nova plataforma. Tanto a DoubleClick como a Akamai foram pioneiras da Web 2.0, entretanto também podemos ver como é possível compreender mais sobre as possibilidades através da adoção de padrões adicionais de design Web 2.0.

Vamos nos aprofundar em cada um desses três casos, trazendo à tona alguns dos elementos essenciais de diferença.

Netscape versus Google

Se a Netscape era o exemplo padrão para a Web.1, a Google é, com toda certeza, o padrão para a Web 2.0, nem que seja porque seus respectivos lançamentos na bolsa de valores constituíram-se em acontecimentos determinantes para cada uma de suas eras. Vamos, portanto, começar com uma comparação entre essas duas companhias e seus posicionamentos.

3 N.T. Mashup: aplicativo web híbrido. Da Wikipedia: “Um *mashup* é um site web ou aplicativo web que combina, em uma experiência integrada (sem costura aparente), conteúdo de mais de uma fonte.

A Netscape definiu “a web como plataforma” em termos do velho paradigma de software: seu carro-chefe era o navegador - um aplicativo para desktop - e sua estratégia era usar seu domínio no mercado de navegadores para construir um mercado de altos preços para produtos para servidores. O controle sobre os padrões para a exibição de conteúdo e aplicativos no navegador, em tese, daria à Netscape o tipo de poder de mercado que a Microsoft desfrutava no mercado de PCs. Semelhante ao modo como a noção de “carruagem sem cavalos” apresentava o automóvel como uma extensão de algo familiar, a Netscape promoveu um *webtop* para substituir o desktop e planejou povoar este *webtop* com atualizações de informação e miniaplicativos empurrados para o *webtop* por provedores de informação que comprariam os servidores da Netscape.

No final, tanto os navegadores como os servidores web acabaram convertendo-se em commodities e o valor dirigiu-se para os serviços oferecidos via plataforma web.

A Google, ao contrário, começou sua vida como uma aplicação nativa da web, nunca foi vendida ou empacotada mas oferecida como um serviço pelo qual os usuários pagavam direta ou indiretamente. Nenhuma das armadilhas da velha indústria de software estava presente. Não havia prazos marcados para o lançamento de software, apenas contínuos aperfeiçoamentos. Nada de licenças ou vendas, apenas uso. Nada de conversões para diferentes plataformas para que os usuários pudessem rodar o software no seu próprio equipamento, apenas uma coleção altamente redimensionável de PCs commodity rodando sistemas operacionais de código aberto além de aplicativos e ferramentas caseiras que nunca serão vistos por ninguém de fora da companhia.

No fundo, a Google requer uma competência que a Netscape nunca precisou: gerenciamento de base de dados. A Google não se limita a uma coleção de ferramentas de software, é uma base de dados especializada. Sem os dados, as ferramentas são inúteis; sem o software, não se consegue gerenciar os dados. Licença de software e controle sobre os APIs⁴ – a alavanca de poder na era anterior – tornam-se irrelevantes porque o software não precisa mais ser distribuído mas apenas

⁴ N.T. Iniciais de *Application Program Interface*: Interface para Programação de Aplicativos.

executado e também porque sem a habilidade para coletar e gerenciar os dados, o software tem pouca utilidade. Na verdade, *o valor do software é proporcional à escala e dinamismo dos dados que ele ajuda a gerenciar.*

O serviço da Google não é um servidor – embora ele seja prestado através de uma maciça coleção de servidores de Internet –, nem um navegador – embora seja experimentado pelo usuário dentro do navegador. Nem o serviço de busca que é o seu carro-chefe sequer hospeda o conteúdo que ele permite o cliente encontrar. Muito parecido com um telefonema, que acontece não apenas nos aparelhos em cada extremo da ligação mas na rede entre eles, a Google acontece no espaço entre navegador e ferramenta de busca, e o servidor de conteúdo de destino, como um possibilitador ou intermediário entre o(a) usuário(a) e a sua experiência online.

Enquanto tanto a Netscape como a Google possam ser descritas como companhias de software, fica evidente que a Netscape pertencia ao mesmo mundo de software que Lotus, Microsoft, Oracle, SAP e outras companhias que tiveram seu início durante a revolução do software dos anos oitenta, enquanto os colegas do Google são outros aplicativos de Internet como eBay, Amazon, Napster e, sim, DoubleClick e Akamai.

DoubleClick versus Overture e AdSense

Como a Google, a DoubleClick é uma verdadeira cria da era da Internet. Ela funciona como um serviço, tem como competência principal o gerenciamento de dados e, como observado acima, foi uma pioneira em serviços web, muito antes que os serviços web sequer tivessem um nome. Entretanto, no final das contas, a DoubleClick foi limitada pelo seu modelo de negócios. Ela embarcou na noção dos anos noventa de que a rede tinha a ver com publicação e não com participação; os anunciantes, não os consumidores é que estariam no comando; que o tamanho importava e que a internet estava sendo cada vez mais dominada pelos principais websites, conforme os números da MediaMetrix e outras companhias de medidas de pontos de anúncios da rede.

Como resultado, a DoubleClick orgulhosamente menciona no seu site “mais de 2000 implementações bem sucedidas” do seu software. A Yahoo! Search Marketing (antiga

Overture) e a Google AdSense, ao contrário, cada uma delas já serve centenas de milhares de anunciantes.

O sucesso da Overture e da Google decorre da compreensão daquilo a que Chris Anderson se refere com “a cauda longa”, o poder coletivo de pequenos sites que constituem a maior parte do conteúdo da rede. As ofertas da DoubleClick requerem um contrato formal de venda, limitando seu mercado aos poucos milhares de maiores sites da rede. Overture e Google calcularam como tornar possível a colocação de anúncio em praticamente qualquer página da rede. E, mais ainda, evitaram os formatos amistosos de anúncio tais como os banners e os popups em favor de textos de anúncio minimamente invasivos, relacionados ao contexto e amistosos para com o consumidor.

A lição da Web 2.0: alavancar o auto-serviço do consumidor e algoritmos de gerenciamento de dados visando atingir a rede em toda sua extensão e não apenas o centro, até a cauda longa e não apenas a cabeça.

Não surpreende que outras histórias de sucesso da Web 2.0 demonstrem este mesmo tipo de comportamento. A eBay possibilita transações ocasionais de poucos dólares entre pessoas físicas, atuando como um intermediário mecânico. A Napster (embora fechada por razões de ordem legal) formou a sua rede não com a construção de um banco centralizado de músicas mas montando um sistema de tal forma, que qualquer *downloader* também se tornava um servidor e assim aumentava a rede.

Akamai versus BitTorrent

Como a DoubleClick, a Akamai está otimizada para fazer negócios com a cabeça, não com a cauda, com o centro, não com as bordas. Embora beneficie os indivíduos da periferia ao facilitar seu acesso aos sites de alta demanda do centro, ela retira sua receita dos sites centrais.

Uma plataforma sempre supera um aplicativo

Em cada um dos confrontos com rivais, a Microsoft, com sucesso, lançou a carta da plataforma como um trunfo até contra os mais poderosos aplicativos. O Windows permitiu que a Microsoft substituísse o Lotus 1-2-3 pelo Excel, o WordPerfect pelo Word e o Navegador Netscape pelo Internet Explorer.

A BitTorrent, como outras pioneiras do movimento P2P⁵, assume uma abordagem radical na descentralização da internet. Cada cliente também é um servidor; arquivos são desmembrados em fragmentos que podem ser fornecidos de múltiplas localizações, de forma transparente, empregando a rede de *downloaders* para prover tanto banda larga como dados para outros usuários. Na verdade, quanto mais popular for o arquivo mais rapidamente pode ser servido na medida em que há mais usuários fornecendo banda larga e fragmentos do arquivo completo.

A BitTorrent, portanto, demonstra um princípio chave da Web 2.0: *o serviço fica automaticamente melhor quanto mais forem os usuários que de se utilizam*. Enquanto a Akamai precisa adicionar servidores para melhorar o serviço, cada consumidor da BitTorrent traz os seus próprios recursos para o grupo. Existe uma “arquitetura de participação” implícita, uma ética de cooperação embutida pela qual o serviço atua, primariamente, como um intermediário inteligente que conecta as pontas entre si e combina o poder dos próprios usuários.

2. Tirando partido da inteligência coletiva

Desta vez, entretanto, o confronto não é entre uma plataforma e um aplicativo mas entre duas plataformas com modelos de negócios radicalmente diferentes. De um lado, um único fornecedor de software cuja imensa base instalada, sistema operacional firmemente integrado e APIs conferem controle sobre o paradigma da programação. Do outro lado, um sistema sem proprietário, reunido por um conjunto de protocolos, padrões abertos e acordos de cooperação.

O Windows representa o ápice do controle do proprietário através de APIs de software. A Netscape tentou, em vão, arrebatá-lo o controle da Microsoft usando as mesmas técnicas que a própria Microsoft havia usado contra outros rivais. Mas o Apache, que se ateve aos padrões abertos da web, prosperou. A batalha já não é desequilibrada – uma plataforma *versus* um único aplicativo – mas plataforma *versus* plataforma, com a questão sendo qual a plataforma e, mais profundamente, qual a arquitetura e qual o modelo de negócios está mais de acordo com as oportunidades que se apresentam.

O Windows foi uma solução brilhante para os problemas do início da era PC. Ele nivelou o campo para os desenvolvedores de aplicativos resolvendo uma infinidade de problemas que haviam, anteriormente, atormentado a indústria.

Mas uma única abordagem monolítica, controlada por um único fornecedor, já não é mais uma solução, é um problema.

⁵ N.T. *Peer-to-Peer*. Usuário a usuário.

O principal princípio por trás do sucesso dos gigantes nascidos na era Web 1.0 que sobreviveram para liderar a era Web 2.0 parece ser porque eles souberam aproveitar o poder que a rede tem de tirar partido da inteligência coletiva:

- Hiperlinks são o fundamento da rede. À medida que os usuários adicionam conteúdo e sites novos, esses passam a integrar a estrutura da rede à medida que outros usuários descobrem o conteúdo e se conectam a ele. Do mesmo modo que se formam sinapses no cérebro – com as associações fortalecendo-se em função da repetição ou da intensidade – a rede de conexões cresce organicamente, como resultado da atividade coletiva de todos os usuários da rede.
- A Yahoo!, a primeira grande história de sucesso na Internet, nasceu como um catálogo ou diretório de links, uma agregação do melhor trabalho de milhares e, depois, de milhões de usuários da rede. Apesar da Yahoo!, desde então, ter entrado no negócio de criação de vários tipos de conteúdo, seu papel como portal de entrada para o trabalho coletivo dos usuários da rede continua a ser o centro do seu valor.
- A entrada da Google na área de busca, que rapidamente a transformou em líder indiscutível nesse mercado, foi o PageRank, um método de fornecer melhores resultados de busca, usando a estrutura de *links* da rede, ao invés de só as características dos documentos.

Sistemas orientados para a comunicação – como certamente o é a “internet- como-plataforma” – requerem inter-operabilidade. A não ser que um fornecedor possa controlar as duas extremidades de qualquer interação, as possibilidades de manutenção forçada do usuário por APIs de software são limitadas.

Qualquer fornecedor Web 2.0 que procure assegurar seus ganhos com aplicativo através do controle da plataforma não estará, por definição, valendo-se da força da plataforma.

Isso não significa que não haja oportunidades para “lock-in” e vantagem competitiva, mas acreditamos que elas não serão obtidas através de controle de APIs de software e de protocolos. Há um novo jogo em ação. Darão certo na era Web 2.0 as companhias que compreenderem as regras desse jogo ao invés de tentar voltar às regras da era de software para PC.

- O produto da eBay é a atividade coletiva de todos os seus usuários; como a própria web, a eBay cresce organicamente em resposta à atividade do usuário e o papel da companhia é o de possibilitadora de um contexto em que essa atividade do usuário possa acontecer. Além disso, a vantagem competitiva da eBay deve-se, quase que totalmente, à massa crítica de compradores e vendedores tornando significativamente menos atraente qualquer novo concorrente que ofereça serviços semelhantes.
- a Amazon vende os mesmos produtos que concorrentes como Barnesandnoble.com e recebe dos seus fornecedores as mesmas descrições dos produtos, imagens de capa e conteúdo editorial. Mas a Amazon desenvolveu uma ciência sobre o engajamento do usuário. Ela tem infinitamente mais avaliações de usuários, convites para participar de várias formas em virtualmente todas as páginas – e, ainda mais importante – usa a atividade do usuário para produzir melhores resultados de busca. Enquanto uma busca da BarnesandNoble.com vai provavelmente levar a resultados encabeçados pelos produtos da própria companhia ou que sejam patrocinados, os da Amazon sempre apresentam em primeiro lugar os “mais populares”, uma computação em tempo real baseada não apenas em vendas mas em outros fatores que os conhecedores da Amazon chamam de “fluxo” em torno dos produtos. Contando com a participação do usuário em escala muitíssimo maior, não surpreende que também as vendas da Amazon suplantem as dos concorrentes.

Atualmente, as companhias inovadoras que captaram esse insight, levando-o talvez ainda mais longe, estão deixando sua marca na rede:

- Wikipedia, uma enciclopédia online baseada na noção duvidosa de que um verbete pode se adicionado por qualquer usuário da rede e editado por qualquer outro, é uma experiência de confiança radical, que aplica à criação de conteúdo o ditado de Eric Raymond (originalmente cunhado no contexto de software aberto de que “com um número suficiente de olhos, todos os bugs tornam-se visíveis”. A Wikipedia já está entre os cem *top sites* e muitos acreditam que logo estará entre os dez primeiros. Trata-se de uma profunda mudança na dinâmica de criação de conteúdo!

- Sites como o del.icio.us e o Flickr, duas companhias que recentemente vêm recebendo bastante atenção, inauguraram um conceito que alguns chamam de “folksonomia” (em oposição a taxonomia), um estilo de categorização colaborativa de sites que emprega palavras chaves livremente escolhidas, freqüentemente chamadas de tags⁶. O uso de tags permite associações múltiplas e superpostas como as que o próprio cérebro usa ao invés de categorias rígidas. No exemplo típico, uma fotografia de um filhote de cachorro no Flickr pode ser etiquetada tanto como “*filhote*” ou como “fofinho” – permitindo ser localizada através de eixos de atividade de usuário que foram naturalmente gerados.
- Produtos colaborativos de filtragem de spam como o Cloudmark agregam as decisões individuais de usuários de e-mail sobre o que é e o que não é spam, com melhor desempenho do que sistemas baseados na análise das próprias mensagens.
- É um truísmo que os maiores casos de sucesso na internet não anunciaram seus produtos. Sua utilização se deu por marketing viral, isto é, recomendações que partiram de usuário para usuário. Pode-se ter quase certeza de que, se um site ou produto depende de publicidade para se tornar conhecido, não é Web 2.0.
- Até mesmo muito da infraestrutura da rede – incluindo Linux, Apache, MySQL, e de código Perl, PHP ou Python em uso na maioria dos servidores web – apóiam-se nos métodos de produção comunitária de código aberto, eles próprios um exemplo de inteligência coletiva possibilitada pela rede. Há mais de cem mil projetos de software aberto listados no SourceForge.net. Qualquer pessoa pode adicionar um projeto, pode baixar e usar o código, e novos projetos migram da periferia para o centro como resultado de usuários que os fazem funcionar, em um processo orgânico de adoção de software baseado quase unicamente no marketing viral.

A lição: Efeitos na rede resultantes das contribuições dos usuários são a chave para a supremacia de mercado na era Web 2.0.

⁶ N.T. etiquetas

Blogs e a sabedoria das massas

Um dos aspectos mais intensamente comentados da era Web 2.0 é o crescimento dos blogs. Páginas pessoais sempre existiram desde os primórdios da rede, e o diário pessoal e a coluna de opinião diária existem desde antes disso, portanto por que tanto alvoroço?

Em seu aspecto mais básico, um blog é apenas uma página pessoal em formato de diário.

Mas, como Rich Skrenta assinala, a organização cronológica de um blog “parece uma diferença trivial, mas puxa uma cadeia completamente diferente de distribuição, de publicidade e de valor”.

Um dos fatores que fez diferença foi a tecnologia chamada RSS. O RSS é o avanço mais significativo na arquitetura básica da rede desde que os primeiros hackers perceberam que a CGI⁷ podia ser usada para criar sites baseados em base de dados. Permite que alguém não apenas acesse uma página mas faça uma assinatura sendo notificado cada vez que haja mudanças na página. Skrenta chama a isso de “rede incrementável”. Outros a chamam de “rede viva”.

Agora, é claro, “sites dinâmicos” (isto é, sites baseados em base de dados com conteúdo dinamicamente gerado) substituíram as

A arquitetura de participação

Alguns sistemas são projetados de forma a encorajar a participação. No seu trabalho, “A cornucópia dos Commons”, Dan Bricklin observou que existem três formas de se construir uma grande banco de dados. A primeira, demonstrada pela Yahoo!, é pagar pessoas para isso. A segunda, inspirada nas lições da comunidade de código aberto, é conseguir que voluntários executem a tarefa. O resultado é o “Open Directory Project”, um concorrente de código aberto da Yahoo! Mas a Napster mostrou uma terceira forma. Porque a Napster tem como padrão servir automaticamente qualquer música que tenha sido baixada, todo usuário automaticamente ajudou a construir o valor do banco de dados compartilhado. Esta mesma abordagem foi seguida por todos os outros serviços de compartilhamento de arquivo P2P.

Uma das lições chave da era Web 2.0 é essa: *Usuários adicionam valor*. Mas apenas uma pequena percentagem de usuários se darão ao trabalho de adicionar valor ao seu aplicativo através de meios explícitos. Portanto, as companhias Web 2.0 *programam padrões para agregar dados do usuário e gerar valor como um efeito paralelo ao uso comum do aplicativo*. Como assinalado acima, elas constroem sistemas que ficam melhores quanto mais as pessoas os utilizam.

⁷ N.T. Iniciais de *Common Gateway Interface*: Interface de Passagem Comum.

páginas estáticas há mais de dez anos. O que é dinâmico em termos da rede viva não são apenas as páginas mas os links. Espera-se que um link para um blog conduza a uma página em permanente mudança, com “permalinks”⁸ para qualquer entrada individual e notificação a cada mudança. Um link RSS é, portanto, muito mais forte do que, por exemplo, um favorito ou um link para uma única página.

O RSS também significa que o navegador de rede não é o único meio de se ver uma página. Enquanto alguns agregadores RSS, como o Bloglines são baseados na rede, outros são clientes desktop e ainda outros permitem que usuários de dispositivos portáteis assinem o conteúdo constantemente atualizado.

O RSS está sendo atualmente utilizado para enviar não apenas avisos de atualizações de blog mas também todos os tipos de atualização de dados, incluindo cotações da bolsa, previsão do tempo e disponibilidade de fotos. Esse uso é, na verdade, uma de suas raízes: o RSS nasceu em 1997, fruto da confluência da tecnologia “*Really Simple Syndication*” de Dave Winer – usada para enviar atualizações de blog – e o “*Rich Site Summary*” da Netscape que permitia aos usuários criar homepages do Netscape personalizadas, com fluxos de dados regularmente atualizados.

Mitch Kapor uma vez observou que “arquitetura é política.” A participação é intrínseca ao Napster, parte integral da sua arquitetura.

Esse insight de arquitetura também pode ser mais importante para o sucesso do software de código aberto do que o voluntariado que é mencionado com mais frequência. A arquitetura da Internet, e da World Wide Web, assim como de projetos de softwares de código aberto como Linux, Apache and Perl é tal que usuários que procuram “egoisticamente” satisfazer seus próprios interesses, acabam gerando valor coletivo como um sub-produto automático. Cada um desses projetos tem um pequeno centro – mecanismos de extensão bem definidos – e uma abordagem que permite que qualquer componente bem comportado seja adicionado por qualquer usuário, aumentando as camadas externas daquilo a que Larry Wall, o criador do Perl, se refere como “a cebola”. Em outras palavras, estas tecnologias demonstram efeitos de rede simplesmente pelo modo como foram projetadas.

Tais projetos podem ser vistos como tendo uma arquitetura natural de participação. Entretanto, como demonstrado pela Amazon, através de um esforço contínuo (assim como de incentivos econômicos com o programa de afiliados), é possível sobrepor tal arquitetura em um sistema que normalmente não pareceria possuí-la.

⁸ N.T. Aglutinação de *Permanent Link*.

A Netscape desinteressou-se, e a tecnologia foi levada adiante pelo pioneiro de blogs Userland, a companhia de Winer. Na atual safra de aplicações percebe-se, entretanto, a herança de ambos os pais.

Mas o RSS é apenas um dos aspectos que faz o blog ser diferente de uma página comum da rede. Tom Coates chama atenção para a importância do *permalink*:

O *permalink* pode parecer, agora, uma peça trivial de funcionalidade mas foi, na verdade, o dispositivo que transformou os blogs, de um fenômeno que facilitava a publicação de conteúdo em uma confusão envolvendo conversas entre comunidades superpostas. Pela primeira vez, tornou-se relativamente fácil apontar e discutir especificamente um post em algum site pessoal. Iniciavam-se discussões. O chat emergia. E como resultado, amizades formaram-se ou se estreitaram. O *permalink* foi a primeira – e mais bem sucedida – tentativa de se construir pontes entre blogs.

De muitas maneiras, a combinação de RSS e *permalinks* acrescenta ao http – o protocolo web – muitas das características do NNTP – o Network News Protocol Usenet. A “blogosfera” pode ser pensada como um novo meio de comunicação entre usuários, equivalente à Usenet e outros fóruns que foram os pontos de encontro do primeiros tempos da Internet. As pessoas podem não apenas assinar os sites umas das outras – obtendo fácil acesso aos comentários individuais de uma página – mas também – via um mecanismo conhecido como *trackbacks*⁹ – podem ver quando alguém cria links para a sua página e podem responder, quer criando links recíprocos, quer adicionando comentários.

Curiosamente, links de mão dupla eram o objetivo dos primeiros sistemas de hipertexto como o Xanadu. Os puristas do hipertexto comemoraram os *trackbacks* como um passo na direção dos links de mão dupla. Mas, observe-se que os *trackbacks* não são propriamente de mão dupla; na verdade, eles são links de mão única (potencialmente) simétricos que criam um efeito de links de mão dupla. A diferença pode parecer sutil mas, na prática, é enorme. Sistemas de redes sociais como Friendster, Orkut e LinkedIn – que requerem permissão do receptor a fim de que uma conexão possa ser criada – carecem da mesma “escalabilidade” que a web. Como assinalou Caterina

⁹ N.T. A tradução literal seria “seguir a pista”.

Fake, co-fundadora do serviço de compartilhamento de fotos Flickr, a atenção depende da coincidência para ser recíproca. (O Flickr, portanto, permite aos usuários estabelecer listas de acompanhamento – qualquer usuário pode assinar o fluxo de fotografias de qualquer outro, via RSS. O acesso é notificado mas o usuário não precisa autorizar a conexão).

Se uma parte essencial da Web 2.0 é tirar partido da inteligência coletiva, transformando a web em uma espécie de cérebro global, a blogosfera equivale a um constante bate-papo mental que tem lugar na parte frontal do cérebro, a voz que todos ouvimos em nossas cabeças. Pode não refletir a estrutura mais profunda do cérebro – freqüentemente inconsciente – mas equivale ao pensamento consciente. E, como reflexo do pensamento consciente e da atenção, a blogosfera começou a exercer um poderoso efeito.

Em primeiro lugar, pelo fato das ferramentas de busca usarem estrutura de links para predizer páginas importantes, os blogueiros – como os mais produtivos e atualizados usuários de links – exercem um papel desproporcional nos resultados dos mecanismos de busca. Em segundo lugar, o fato da comunidade de blogs ser tão auto-referida – com blogueiros focalizando a atenção em outros blogueiros – aumenta sua visibilidade e poder. A “câmara de eco” condenada pelos críticos funciona também como amplificador.

Se fossem meramente um amplificador, os blogs não despertariam nenhum interesse. Mas, como a Wikipedia, os blogs capitalizam a inteligência global como uma espécie de filtro. Entra em ação o que James Suriowecki chama de “a sabedoria das massas” e, do mesmo modo que o PageRank produz resultados melhores do que a análise de qualquer documento individual, a atenção coletiva da blogosfera seleciona o valor.

Enquanto a mídia tradicional¹⁰ pode encarar os blogs como concorrentes, o que realmente incomoda é que a competição é com a blogosfera como um todo. Não se trata apenas de uma competição entre sites, mas uma competição entre modelos de negócio. O mundo da Web 2.0 é também o mundo que Dan Gillmor chama de “nós, a

¹⁰ N.T. Em inglês: *mainstream media*

mídia”, um mundo em que “a antiga audiência” – não umas poucas pessoas nos bastidores – decide o que é importante.

3. Dados são o próximo Intel Inside

Até hoje, todo aplicativo importante da Internet tem se apoiado em um banco de dados especializado: robô de indexação do Google, diretório (e robô) do Yahoo!, banco de dados de produtos da Amazon, banco de dados de produtos e vendedores da eBay, banco de dados de mapas da MapQuest, banco de dados “espalhado” de músicas da Napster. No ano passado, em conversa pessoal, Hal Varian comentou que “o SQL¹¹ é o novo HTML”. O gerenciamento de bancos de dados é a nova competência central das companhias Web 2.0, de tal modo que temos, às vezes, nos referido a esses aplicativos como “*infoware*” ao invés de simplesmente software.

Este fato coloca uma questão central: a quem pertencem os dados?

Na era da Internet, já se conhecem vários casos em que o controle sobre um banco de dados levou ao controle do mercado e enormes retornos financeiros. O monopólio sobre o registro de nomes de domínios, inicialmente concedido pelo governo¹² à Network Solutions (mais tarde comprada pela Verisign), foi um dos primeiros grandes negócios lucrativos da Internet. Enquanto argumentamos que vantagens de negócios pelo controle de APIs¹³ é muito mais difícil nos tempos da internet, o mesmo não se dá em relação ao controle sobre fontes-chave de dados, especialmente quando for caro criar tais fontes ou quando elas permitirem retornos crescentes via efeitos de internet.

Leia os avisos de copyright abaixo de todo mapa gerado pela MapQuest, maps.yahoo.com, maps.msn.com, ou maps.google.com e encontrará a observação: “Maps copyright NavTeq, Tele Atlas” ou, a partir dos novos serviços de imagens de satélite, “Images copyright Digital Globe”. Essas companhias fizeram investimentos substanciais em seus bancos de dados (só a NavTeq diz ter investido 750 milhões de dólares para construir seu banco de dados de endereços e indicações de ruas. A

¹¹ N.T. Iniciais para *Structured Query Language*: Linguagem de Questão Estruturada. Usa banco de dados na configuração de uma busca.

¹² N.T. Governo dos Estados Unidos.

¹³ N.T. Iniciais de *Application Program Interface*: Interface para programação de aplicativos.

Digital Globe gastou 500 milhões de dólares para lançar seu próprio satélite visando aperfeiçoar as imagens fornecidas pelo governo). A NavTeq chegou ao ponto de imitar a conhecida logomarca “Intel Inside” da Intel: automóveis com sistemas de navegação recebem a marca “NavTeq Onboard”. Os dados são de fato o “Intel Inside” desses aplicativos, o único componente essencial em sistemas cuja infraestrutura de software é principalmente de código aberto ou de algum outro modo “commoditizado”

Atualmente a disputa acirrada que tem lugar na arena de mapas da web demonstra como a falta de percepção sobre a importância de possuir os dados centrais de um aplicativo poderá, eventualmente, enfraquecer o poder competitivo. A MapQuest foi pioneira na categoria de mapas, em 1995. Entretanto, quando a Yahoo!, a Microsoft e mais recentemente a Google decidiram entrar nesse mercado, elas conseguiram facilmente oferecer um aplicativo concorrente simplesmente obtendo licença sobre os mesmos dados.

Compare-se, entretanto, a posição da Amazon.com. Assim como concorrentes tais como a Barnesandnoble.com, seu banco de dados original veio da R.R. Bowker que fornece os registros ISBN¹⁴. Entretanto, de modo diferente da MapQuest, a Amazon foi implacável na melhoria dos dados, acrescentando informações fornecidas pelas editoras tais como imagens da capa, sumário, índice e acesso a trechos dos livros. E, o que foi mais importante, ela engajou seus usuários a fazerem comentários sobre os dados de tal forma que, depois de dez anos, a Amazon.com – e não mais a Bowker – é a principal fonte de dados bibliográficos sobre livros, referência para estudiosos e bibliotecários, tanto como para consumidores. Também introduziu seu próprio código de identificação, o ASIN¹⁵ – que corresponde ao ISBN quando este existe e/ou cria um código equivalente para produtos sem codificação. Efetivamente, a Amazon “abarcou e ampliou” seus fornecedores de dados.

Imagine se a MapQuest tivesse feito o mesmo: engajar seus usuários no comentário dos mapas e indicações, acrescentando camadas de valor. Teria sido muito mais difícil

¹⁴ N.T. Iniciais de *International Standard Book Number*. Número de código internacionalmente aceito de catalogação de livros.

¹⁵ N.T. Iniciais de *Amazon Standard Identification Number*. Segundo a Amazon informa no seu site: “Praticamente todo produto no nosso site tem o seu próprio ASIN – um código único que usamos para identificá-lo. Para livros o ASIN é o mesmo que o número do ISBN mas, para todos os outros produtos, cria-se um novo ASIN quando eles passam a integrar o nosso catálogo.

para os concorrentes entrarem no mercado apenas com a licença sobre o banco de dados.

A chegada recente do Google Maps servirá de laboratório vivo para a competição entre vendedores de aplicativos e seus fornecedores de dados. O modelo leve de programação do Google levou à criação de numerosos serviços de valor agregado na forma de mashups que ligam o Google Maps a outras fontes de dados acessíveis via internet. O exemplo por excelência de um desses mashups é a housingmaps.com, de Paul Rademacher, que reúne a Google Maps com a Craigslist – dados sobre aluguel de apartamentos e venda de casas – criando uma ferramenta interativa de busca de imóvel.

No momento, esses mashups são ainda, sobretudo, experiências inovadoras feitas por hackers, mas que logo serão seguidas por iniciativas empresariais. Na verdade, já se pode ver que, pelo menos para uma classe de desenvolvedores, a Google retirou da Navteq o papel de fonte de dados e se auto inseriu como um intermediário privilegiado. Nos próximos anos, esperamos assistir a disputas entre fornecedores de dados e vendedores de aplicativo à medida em que vão percebendo a importância de certas classes de dados como blocos de construção para os aplicativos Web 2.0.

Já começou a corrida pela posse de certa classe de dados centrais: localização, identidade, calendário de eventos públicos, identificadores de produtos e códigos. Em muitos casos – quando o custo de criação dos dados for significativo – poderá haver uma oportunidade para um jogo estilo Intel Inside, com uma única fonte de dados. Em outros, o vencedor será a companhia que primeiro atingir uma massa crítica de dados através da participação de usuários, e transformar esses dados agregados em um serviço.

Por exemplo, na área de identidade, PayPal, Amazon's 1-click e os milhões de usuários de sistemas de comunicação podem, todos eles, disputar legitimamente a construção de um banco de dados de identidade que alcance toda a rede. (Sob esse ponto de vista, a recente tentativa da Google de usar os números de telefones celulares como identificadores para as contas de Gmail pode ser um passo na direção de abranger e expandir o sistema telefônico.) Enquanto isso, novas empresas como a Sxip estão

explorando o potencial de identidade distribuída na busca de um tipo de “1-click distribuído” que fornecerá um subsistema de identidade Web 2.0 sem costura. Na área de calendário, o EVDB¹⁶ é uma tentativa de construir o maior calendário compartilhado do mundo através de uma arquitetura de participação de estilo wiki. Enquanto não sai o veredicto sobre o sucesso de nenhuma empresa ou abordagem em particular, é claro que padrões e soluções nessas áreas – que efetivamente transformem certas classes de dados em sistemas confiáveis do “sistema operacional de internet” – tornarão possível a próxima geração de aplicativos.

Ainda um outro ponto deve ser considerado em relação a dados: a preocupação do usuário com privacidade e direitos sobre seus dados próprios. Em muitos dos aplicativos do início da era web, o copyright é muito superficialmente garantido. Por exemplo, a Amazon pleiteia os direitos sobre qualquer avaliação encaminhada ao site mas, na ausência de fiscalização, as pessoas podem colocar a mesma avaliação em outro lugar. Entretanto, à medida que as companhias começam a perceber que o controle sobre os dados pode ser sua principal fonte de vantagem competitiva, poderemos testemunhar um aumento nas tentativas de controlá-los.

De forma muito semelhante a como o aumento do software proprietário levou ao movimento do “Free Software”¹⁷, há a expectativa de que, na próxima década, o crescimento dos bancos de dados proprietários resulte em um movimento “Free Data”. Os primeiros sinais dessa tendência inconformista já podem ser notados nos projetos de dados abertos como Wikipedia, Creative Commons e em projetos de software – como o Greasemonkey – que permitem aos usuários assumir o controle de como os dados aparecem no seu computador.

4. O fim do ciclo de lançamentos de software

Como observado acima quando da discussão Google *versus* Netscape, uma das características que define a era software da internet é que o software é apresentado como um serviço e não um produto. Esse fato acarreta mudanças fundamentais no modelo de negócios da companhia:

¹⁶ N.T. Iniciais de *Events and Venues Database*.

¹⁷ N.T. “Software livre”.

1. *As operações precisam se tornar uma competência central.* A competência em desenvolvimento de produto da Google ou da Yahoo! precisa estar no mesmo nível da competência das operações do dia-a-dia. A mudança do software como produto para software como serviço é tão fundamental que *o software deixará de funcionar a não ser que receba manutenção diária.* A Google tem que percorrer a rede continuamente e atualizar seus índices, filtrar o spam de links e outras tentativas de afetar seus resultados, responder contínua e dinamicamente a centenas de milhões de questões assíncronas dos usuários e, simultaneamente, estabelecer a correspondência entre elas e a publicidade relativa ao contexto.

Não é por acaso que o sistema de administração da Google, suas técnicas de rede e de balanceamento de carga sejam, talvez, segredos mais bem guardados do que seus algoritmos de busca. O sucesso da Google em automatizar tais processos é um fator chave da sua vantagem, em termos de custo, sobre os concorrentes.

Também não é por acaso que linguagens de *script* tais como Perl, Python, PHP e, agora, a Ruby sejam tão importantes para as companhias web 2.0. A Perl foi notavelmente descrita por Hassan Schroeder, o primeiro webmaster da Sun, como a “Silvertape da internet”. Linguagens dinâmicas (freqüentemente chamadas de linguagens de script e olhadas com desdém pelos engenheiros de software da era dos artefatos de software) são a ferramenta preferida dos administradores de sistema e de rede, assim como dos desenvolvedores de aplicativos para a construção de sistemas dinâmicos que requerem alteração constante.

2. *Usuários devem ser tratados como co-desenvolvedores*, em referência às práticas de desenvolvimento do código aberto (mesmo se for pouco provável que o software em questão seja lançado sob uma licença de código aberto). O lema do código aberto “lançar logo e lançar sempre” transformou-se em uma posição ainda mais radical, “o beta perpétuo”, em que o produto é desenvolvido em aberto, com novos recursos surgindo a cada mês, semana ou mesmo dia. Não é por acaso que se pode esperar

que serviços tais como Gmail, GoogleMaps, Flickr, del.icio.us e outros do mesmo tipo carreguem o logo “Beta” por anos a fio.

O monitoramento em tempo real do comportamento do usuário, para ver exatamente quais e como os novos recursos estão sendo usados, torna-se, portanto, uma outra importante competência a ser exigida. Um desenvolvedor web de um grande serviço online comentou: “Todo dia nós disponibilizamos dois ou três recursos em algum lugar do site e se os usuários não os utilizam, nós os removemos. Se eles agradam, nós os implementamos no site todo.”

Carl Henderson, o principal desenvolvedor do Flickr, recentemente revelou que eles implementam novos recursos a cada meia hora. Este é, claramente, um modelo de desenvolvimento radicalmente diferente! Embora nem todos os aplicativos web sejam desenvolvidos em estilo tão extremo como o do Flickr, quase todos têm um ciclo de desenvolvimento radicalmente diferente de qualquer coisa da era PC ou cliente-servidor. É por esta razão que um editorial recente da ZDnet concluiu que a Microsoft não será capaz de vencer a Google: “O modelo de negócios da Microsoft depende de todos fazerem um upgrade em seu ambiente de computador a cada dois ou três anos. O da Google depende de todos explorarem o que há de novo em seu ambiente de computador diariamente.”

Enquanto a Microsoft tem demonstrado uma enorme capacidade em aprender com a competição e até superá-la, não há dúvida de que, desta vez, a competição vai exigir que ela (e, por extensão, todas as demais companhias de software existentes) torne-se um tipo de companhia profundamente diferente. Companhias nativas da Web 2.0 desfrutam de uma vantagem natural porque não precisam se desfazer de antigos padrões (e seus correspondentes modelos de negócios e fontes de receita).

5. Modelos leves de programação

Assim que a idéia de serviços web se popularizou, grandes companhias entraram na briga com um conjunto complexo de serviços web, concebidos para criar ambientes de programação altamente confiáveis para aplicativos distribuídos.

Entretanto, da mesma forma que a web teve êxito exatamente porque subverteu grande parte da teoria de hipertexto, ao substituir um pragmatismo simples por um desenho ideal, o RSS talvez tenha se tornado o serviço web mais amplamente distribuído, provavelmente devido à sua simplicidade, enquanto os complexos serviços web corporativos ainda precisam alcançar ampla distribuição.

Do mesmo modo, os serviços web da Amazon.com são fornecidos de duas formas: uma que segue os formalismos do conjunto de serviços web SOAP (Simple Object Access Protocol), e a outra simplesmente fornecendo dados XML via HTTP, através de uma abordagem leve às vezes chamada de ReST (Representational State Transfer). Enquanto conexões B2B de alto valor (como as que ligam a Amazon a parceiros de varejo como a ToysRUs) usam SOAP, a Amazon informa que 95% do seu uso é de serviço leve do tipo ReST.

Essa mesma procura por simplicidade pode ser encontrada em outros serviços web “orgânicos”. O recente lançamento do Google Maps é um exemplo. Sua interface simples do tipo AJAX (Javascript e XML) foi rapidamente decifrada por hackers que passaram, então, a remixar os dados formando novos serviços.

Uma tese de investimento Web 2.0

O investidor de risco Paul Kedrosky escreve: “A chave é encontrar investimentos onde sua posição seja contrária à do consenso”. É interessante ver como cada faceta de Web 2.0 envolve discordar do consenso: todos defendiam a importância da privacidade dos dados; Flickr/Napster e outros os tornam públicos. Não é discordar para ser desagradável (petfood! online!), é discordar quando você puder construir algo em cima disso. O Flickr constrói comunidades, a Napster construiu uma vasta coleção.

Uma outra forma de se olhar é que todas as companhias bem sucedidas abrem mão de alguma coisa cara, mas considerada estratégica para, em troca, conseguir, de graça, algo valioso que antes era caro. Por exemplo, a Wikipedia abre mão do controle editorial central em troca de abrangência e rapidez. A Napster desistiu da idéia de “catálogo” (todas as músicas que estavam à venda) e ganhou abrangência. A Amazon desistiu da idéia de ter uma loja de vitrine mas passou a atender o mundo inteiro. A Google abriu mão dos grandes clientes (inicialmente) e conseguiu os 80% cujas necessidades não estavam sendo satisfeitas. Existe alguma coisa bem “aikido” (usar a força do seu oponente contra ele próprio) em dizer: “você sabe, você tem razão – absolutamente qualquer pessoa no mundo inteiro PODE atualizar esse artigo. E sabe de uma coisa? Isso é uma má notícia para você”.

Serviços web relacionados a mapas já estavam disponíveis havia algum tempo em fornecedores de GIS¹⁸ tais como ESRI, MapQuest e Microsoft MapPoint. Mas o Google Maps causou uma revolução devido à sua simplicidade. Enquanto a experiência com qualquer serviço web formal mantido por um fornecedor exigia um contrato formal entre as partes, o modo como o Google Maps foi implementado deixava os dados disponíveis e logo os hackers encontraram formas de reutilizá-los criativamente.

Existem aqui, várias lições significativas:

1. *Apóie modelos leves de programação que permitam a criação de sistemas levemente acoplados.* A complexidade do conjunto de serviços financiados por corporações é planejada para permitir acoplamento firme. Enquanto em muitos casos isso se faz necessário, muitos dos aplicativos mais interessantes podem, de fato, permanecer levemente acoplados e até mesmo frágeis. A mentalidade Web 2.0 é muito diferente da tradicional mentalidade IT¹⁹!
2. *Pense em sindicalização, não em coordenação.* Serviços web simples, como o RSS e os baseados no sistema ReST têm a ver com sindicalizar os dados para o exterior e não em controlar o que acontece com eles quando chegam ao outro lado da conexão. Esta idéia é fundamental para a própria internet, um reflexo do que se conhece por princípio ponta-a-ponta.
3. *Projete para “hackeabilidade” e “remixabilidade”.* Sistemas como a web original, RSS e AJAX têm em comum barreiras à reutilização bem fracas. Grande parte dos softwares úteis são de código aberto e, mesmo quando não o são, há pouco esforço no sentido de proteção à propriedade intelectual. A opção do navegador web “View Source” possibilitou que qualquer usuário copiasse a página web de qualquer outro; o RSS foi projetado para dar poder ao usuário de ver o conteúdo que ele quer e quando quer, sem precisar da autorização do fornecedor da informação; os serviços web de maior sucesso são os que foram mais fáceis de serem levados em direções nunca imaginadas por seus criadores. A expressão “alguns direitos reservados”, popularizada pelo Creative

¹⁸ N.T. Iniciais de *Geographic Information System* (Sistema de informação geográfica).

¹⁹ N.T. Iniciais de *Information Technology* (Tecnologia da informação).

Commons para contrastar com o mais típico “todos os direitos reservados”, é um indicativo útil.

Inovação na montagem

Modelos de negócios leves são um concomitante natural de programação leve e conexões leves. A mentalidade Web 2.0 é boa para a reutilização. Um serviço novo como o housingmaps.com foi construído simplesmente pela junção de dois outros já existentes. O housingmaps.com não tem (ainda) um modelo de negócios mas, para muitos serviços de pequena escala, a Google AdSense (ou talvez as comissões dos afiliados da Amazon, ou ambos) forneça o equivalente a um modelo encaixável de remuneração.

Estes exemplos propiciam um *insight* em relação a outro princípio-chave de web 2.0 que chamamos de “inovação na montagem”. Quando os componentes commodity são abundantes, pode-se criar valor simplesmente pela sua montagem de formas novas ou eficientes. Do mesmo modo como a revolução PC criou muitas oportunidades de inovação na montagem do hardware commodity, com companhias como a Dell criando uma ciência em cima dessa montagem e, portanto, derrotando companhias cujos modelos de negócios precisavam de inovação em desenvolvimento de produto, acreditamos que a Web 2.0 irá criar oportunidades para que companhias superem a concorrência ao serem melhores no aproveitamento e integração de serviços oferecidos por outras.

6. Software em mais de um dispositivo

Uma outra característica Web 2.0 que merece menção é o fato de que ela não se limita mais à plataforma PC. Ao deixar a Microsoft, o desenvolvedor de longa data Dave Stutz deixou o conselho que “Software útil que for escrito acima do nível do dispositivo único proporcionará altas margens por um bom tempo.”

Naturalmente, qualquer aplicativo web pode ser visto como software com mais de um dispositivo. Afinal de contas, até o aplicativo mais simples envolve pelo menos dois computadores: o que hospeda o servidor web e o que hospeda o navegador. E, como

já dissemos, o desenvolvimento da web como plataforma estende essa idéia a aplicativos sintéticos compostos de serviços fornecidos por múltiplos computadores.

Entretando, do mesmo modo como acontece com muitas áreas Web 2.0 – onde o “2.0” não é novidade e sim uma realização mais plena do verdadeiro potencial da plataforma web – esta expressão nos fornece um insight-chave em como projetar aplicativos e serviços para a nova plataforma.

Até hoje, o iTunes é o melhor exemplo desse princípio. Ele vai diretamente do dispositivo portátil até uma maciça infraestrutura web, com o PC atuando como *cache* e estação de controle local. Houve várias tentativas prévias de levar conteúdo web até dispositivos portáteis, mas a combinação iPod/iTunes é um dos primeiros aplicativos projetados do zero para atingir múltiplos dispositivos. O TiVo é um outro bom exemplo.

O iTunes e o TiVo também servem para demonstrar muitos dos outros princípios centrais de Web 2.0. Não são aplicativos web em si mas nivelam o poder da plataforma web, tornando-a uma parte integrante, quase invisível, da sua infraestrutura. O gerenciamento de dados é mais claramente o ponto central da oferta. São serviços e não aplicativos empacotados (embora, no caso do iTunes, possa ser usado como tal, gerenciando apenas os dados locais do usuário). Além disso, tanto TiVo como iTunes apresentam um uso crescente de inteligência coletiva, ainda que, nos dois casos, suas experiências estejam em guerra com o lobby da propriedade intelectual. Existe apenas uma limitada arquitetura de participação no iTunes embora o recente aparecimento dos podcasts traga uma mudança substancial para a equação.

Esta é uma das áreas de Web 2.0 onde esperamos ver algumas das maiores mudanças, à medida que for aumentando o número de dispositivos conectados à nova plataforma. Quais os aplicativos possíveis quando nossos telefones e nossos automóveis não estão consumindo mas, enviando dados? O monitoramento de tráfego em tempo real, *flash mobs* e jornalismo comunitário são apenas alguns dos sinais que chamam a atenção para as capacidades da nova plataforma.

7. Experiência rica do usuário

Desde o navegador Viola, de Pei Wei, em 1992, a web vinha sendo usada para enviar miniaplicativos e outros tipos de conteúdo ativo para dentro do navegador. A introdução do Java, em 1995, foi projetada em torno da entrega desses miniaplicativos. O JavaScript e, depois, o DHTML²⁰ foram introduzidos como formas leves de prover “programabilidade” e experiência rica do usuário para o lado cliente. Há alguns anos, a Macromedia cunhou a expressão “*Rich Internet Applications*”²¹ (que também foi aproveitada pela Laszlo Systems – concorrente de código aberto do Flash) para enfatizar a capacidade do Flash para entregar não apenas conteúdo multimídia mas também experiências com aplicativos de estilo GUI²².

Entretanto, o potencial da web para enviar aplicativos de grande porte não se concretizou até que a Google introduziu o Gmail, rapidamente seguido pelo GoogleMaps – aplicativos web com experiência rica do usuário e interatividade equivalente à do PC. A coleção de tecnologias usada pela Google foi batizada de AJAX²³, em um artigo seminal de Jesse James Garret, da firma de web design Adaptive Path. Dizia ele:

“AJAX não é uma tecnologia. São, na verdade, várias tecnologias, cada uma florescendo a seu modo, e reunindo-se em poderosas formas novas. AJAX incorpora:

- Apresentações baseadas em padrões usando XHTML e CSS;
- Exibição e interação dinâmicos usando o “*Document Object Model*”;
- Troca e manipulação de dados usando XML e XSLT;
- Recuperação de dados assíncronos usando XMLHttpRequest;
- e JavaScript para juntar tudo”.

O AJAX também é um componente-chave de aplicativos Web 2.0 como o Flickr – agora parte da Yahoo! –, os aplicativos basecamp e backpack do 37signals, assim como outros aplicativos Google como Gmail e Orkut. Estamos entrando em um período sem

²⁰ N.T. Iniciais de *Dynamic Hyper Text Markup Language*.

²¹ N.T. Em português: “aplicativos de internet ricos”.

²² N.T. Iniciais de *Graphical User Interface* (Interface de uso gráfico).

²³ N.T. Iniciais de *Asynchronous JavaScript + XML*.

precedentes de inovação na interface de usuário, à medida em que os desenvolvedores web estão finalmente em condições de construir aplicativos web tão ricos como os aplicativos locais dos PC.

Curiosamente, muitas das capacidades que estão sendo atualmente exploradas já estavam disponíveis há vários anos. No fim dos anos 90, tanto a Microsoft como a Netscape tinham idéia do tipo de capacidades que estão sendo, agora, finalmente concretizadas, mas a disputa sobre os padrões a serem empregados dificultou os aplicativos para múltiplos navegadores. Foi só quando a Microsoft definitivamente ganhou a guerra e passou a haver, de fato, um único padrão de escrita para navegador, que esse tipo de aplicativo se tornou possível. E, apesar do Firefox ter reintroduzido a competição no mercado de navegadores, pelo menos até agora, não temos assistido à mesma disputa destrutiva sobre padrões web, que deteve o progresso nos anos 90.

Esperamos ver muitos aplicativos web novos nos próximos anos, tanto os verdadeiramente novos, como ricas re-implementações para web de aplicativos PC. Até hoje, toda mudança de plataforma também criou oportunidades para uma mudança de liderança dos aplicativos dominantes da plataforma anterior.

Padrões de design Web 2.0

No seu livro, “A Pattern Language”, Christopher Alexander propõe um formato de descrição concisa para a solução dos problemas de arquitetura. Ele escreve: “Cada padrão descreve um problema que acontece repetidamente no nosso ambiente e, então, descreve o centro da solução para aquele problema, de tal forma que se pode usar essa mesma solução um milhão de vezes sem nunca repetí-la, do mesmo jeito, duas vezes sequer.”

1. A cauda longa

Pequenos sites compreendem a maior parte do conteúdo da internet; pequenos nichos constituem a massa de aplicações possíveis. *Portanto*: Incentive o auto-serviço do cliente e o gerenciamento algorítmico de dados para alcançar toda a web, até as bordas e não apenas o centro, até a cauda longa e não apenas a cabeça.

2. Dados são o próximo “Intel Inside”

Os aplicativos são cada vez mais baseados em dados. *Portanto*: Para ter vantagem competitiva, procure possuir uma única fonte de dados, difícil de ser recriada.

O Gmail já produziu algumas inovações interessantes na área de e-mail, combinando os pontos fortes da web (acesso de qualquer parte, imensas capacidades de bancos de dados, possibilidades de busca) com interfaces de usuário que se aproximam das interfaces PC em termos de usabilidade. Enquanto isso, outros clientes de e-mail na plataforma PC estão beliscando o problema pelo outro lado, adicionando IM²⁴ e capacidades presenciais. Quão distantes estamos de um cliente de comunicações integrado que combine o melhor do e-mail, IM e celular usando VoIP²⁵ para adicionar capacidades de voz aos aplicativos web? A corrida já começou.

É fácil visualizar como a Web 2.0 também vai refazer o livro de endereços. Um livro de endereços no estilo Web 2.0 iria tratar o livro de endereços local, do PC ou telefone, simplesmente como um cache dos contatos que você tivesse explicitamente pedido ao sistema para armazenar. Enquanto isso, um agente web de sincronização, do estilo Gmail, lembraria qualquer mensagem enviada ou recebida, todos os endereços de e-mail e números de telefone utilizados e iria construir heurísticas de redes sociais para decidir quais alternativas apresentar quando uma resposta não fosse localizada no cache local. Não encontrando ali a resposta, o sistema pesquisaria a rede social mais ampla.

3. **Usuários agregam valor**

A chave para a vantagem competitiva em aplicativos de internet é até onde os usuários acrescentam seus próprios dados àqueles que você fornece.

Portanto: Não restrinja sua “arquitetura de participação” no desenvolvimento de software. Envolve seus usuários, tanto implícita como explicitamente, na adição de valor ao seu aplicativo.

4. **Efeitos de rede como padrão**

Apenas uma pequena porcentagem de usuários dar-se-ão ao trabalho de adicionar valor ao seu aplicativo.

Portanto: Crie padrões para agregar dados de usuário como efeito colateral ao uso do aplicativo.

5. **Alguns direitos reservados**

A proteção à propriedade intelectual limita a reutilização e impede a experimentação.

Portanto: Quando há benefícios decorrentes da apropriação coletiva e não de restrições particulares, certifique-se de que as barreiras são baixas. Siga os padrões existentes e use licenças com o menor número de restrições possível. Projete para “hackeabilidade” e “remixabilidade”.

6. **O beta perpétuo**

Quando dispositivos e programas estão conectados à internet, os aplicativos não são mais artefatos de software, são serviços em andamento.

²⁴ N.T. Iniciais de *Instant Message*.

²⁵ N.T. Iniciais de *Voice Over Internet Protocol*.

Um editor de texto Web 2.0 suportaria a edição colaborativa no estilo wiki²⁶ e não apenas de documentos independentes. Mas também iria suportar a formatação rica que passamos a esperar de editores de texto PC. O Writely é um bom exemplo de um desses aplicativos, embora ainda não tenha muita aceitação.

A revolução Web 2.0 também não se limitará a aplicativos PC. A Salesforce.com mostra como a web pode ser usada para enviar software como um serviço, em aplicativos em escala empresarial como o CRM²⁷.

A oportunidade de competição para novos participantes está em abraçar inteiramente o potencial da Web 2.0. As empresas que conseguirem irão criar aplicativos que aprenderão com os seus usuários, usando uma arquitetura de participação para obter vantagem poderosa não apenas na interface de software, mas na riqueza dos dados compartilhados.

Competências das companhias Web 2.0

Ao explorar os sete princípios acima, enfatizamos alguns dos principais recursos Web 2.0. Cada um dos exemplos discutidos demonstra um ou mais desses sete princípios-chave, mas pode não ter outros. Vamos encerrar, portanto, resumindo o que acreditamos ser as competências centrais das companhias Web 2.0:

Portanto: Não “empacote” novos recursos em lançamentos monolíticos mas, ao invés disso, adicione-os regularmente como parte da experiência diária do usuário. Engage os seus usuários como avaliadores em tempo real e equipe o serviço para que você saiba como as pessoas estão usando os novos recursos.

7. **Coopere, não controle**

Aplicativos Web 2.0 são construídos a partir de uma rede cooperativa de serviços de dados.

Portanto: Ofereça interfaces para serviços web e sindicalização de conteúdo e reutilize os serviços de dados de outros. Encoraje modelos leves de programação que admitam sistemas levemente acoplados.

8. **Software com mais de um único dispositivo**

O PC não é mais o único dispositivo de acesso aos aplicativos de internet e os aplicativos limitados a um único dispositivo têm menos valor do que os que são conectados. *Portanto:* Projete seus aplicativos desde o início para integrar serviços entre dispositivos portáteis, PCs e servidores de internet.

²⁶ N.T. wiki: Web site colaborativo semelhante a um blog de grupo muito mais “adaptável”.

²⁷ N.T. Iniciais de *Customer relationship management*: gerenciamento de relação com o cliente.

- Serviços e não software empacotado, com “escalabilidade” de custo eficiente
- Controle sobre fontes de dados únicas e difíceis de serem criadas e que ficam mais ricas quanto mais as pessoas as utilizarem
- Confiança nos usuários como co-desenvolvedores
- Agregação de inteligência coletiva
- Estimular a cauda longa através de auto-serviço para o cliente
- Software para mais de um dispositivo
- Interfaces de usuário, modelos de desenvolvimento e modelos de negócios leves

Da próxima vez que uma companhia reivindicar ser “Web 2.0”, compare suas características com a lista acima. Quanto mais pontos fizer, mais fará jus ao nome. Lembre-se, no entanto, que excelência em uma área pode contar mais do que alguns pequenos passos em todas as sete.