A PLATAFORMIZAÇÃO DA WEB

ANNE HELMOND

Em 15 de agosto de 2006, o Facebook introduziu sua plataforma para desenvolvedores, ao conceder, a terceiros, acesso aos perfis de usuários e amigos, às fotos e aos eventos, a fim de expandir a "experiência Facebook" para aplicativos externos (Fatterman, 2006) — desse modo, tornou o Facebook um ambiente para desenvolvedores. Um ano depois, na primeira Conferência de Desenvolvimento F8, o Facebook lançou a plataforma Facebook, de modo a marcar oficialmente o avanço do Facebook como plataforma. A plataforma Facebook fornece aos desenvolvedores um conjunto de ferramentas para envio e recuperação de dados do Facebook e de fontes externas para o próprio Facebook, bem como uma profunda integração com o grafo social do Facebook (*Social Graph*) — um mapeamento das conexões entre pessoas e objetos, para a construção de aplicativos (Geminder, 2007; Hicks, 2010).

Neste capítulo eu investigo o desenvolvimento do Facebook como uma plataforma ao situá-lo no contexto de transformação dos sites de redes sociais em *plataformas* de mídias sociais. Situamos essa "plataformização", ou o advento da plataforma, como o modelo dominante de infraestrutura e economia da web, bem como as consequências disso, no contexto histórico desse processo. A plataformização implica a extensão das plataformas de mídias sociais ao restante da web, bem como o movimento de tais plataformas para tornarem os dados da web, que lhes são externos, prontos para configurarem plataformas. A arquitetura tecnológica específica e a distinção ontológica das plataformas serão examinadas ao levarmos em conta um aspecto da "especificidade do meio" (*medium-specificity*) que elas apresentam (Rogers, 2013) e sua programabilidade. Ao fazer isso, seguimos à convocatória de Langois e outros (2009) por uma "perspectiva baseada em plataforma", a qual, segundo Fenwick Mckelvey (2011), deveria criticamente

investigar a programabilidade das plataformas. A análise da descentralização das características da plataforma na web e a recentralização dos dados "prontos para plataforma" é uma maneira de examinarmos as consequências da programabilidade das plataformas de mídias sociais na web.

O novo modelo arquitetônico da plataforma explicitamente torna os sites na web acessíveis ao possibilitar a sua programabilidade com as interfaces técnicas — ou Interface de Programação de Aplicativos (a sigla API, em inglês). Para compreender esse acesso programático, recorro à noção de "fluxos de dados" (*data pours*), proposta por Alan Liu (2004), para conceituarmos as plataformas como sistemas de fluxos de dados, que configuram canais de dados, os quais possibilitam fluxos de dados com terceiros. Esses fluxos de dados não apenas configuram canais para que os dados fluam entre plataformas de mídias sociais e terceiros, mas também funcionam como canais de dados que preparam os dados externos para a configuração de plataformas.

PERSPECTIVA TÉCNICO-MATERIAL SOBRE PLATAFORMAS DE MÍDIAS SOCIAIS

O termo "plataforma" se tornou o conceito dominante tanto para o posicionamento de empresas de mídias sociais no mercado quanto para o modo como elas se dirigem aos usuários. Essa noção tem sido amplamente aceita pelos consumidores e pela imprensa (Gillespie, 2010). Em novos estudos sobre mídias, o conceito de plataforma ganhou destaque ao chamar a atenção para o "trabalho discursivo" que tais estudos empreendem (Gillespie, 2010, p. 348) e para o papel do software, que fortalece as mídias sociais na formação da participação e da socialidade (Bucher, 2012a; Hands, 2013; Langlois et al., 2009; Van Dijck, 2013).

Em uma das principais discussões sobre plataformas, Tartelon Gillespie (2010) apresenta uma variedade de plataformas com foco nas diferentes conotações do termo. No sentido computacional, o autor define plataforma como uma infraestrutura para a construção de aplicativos. Todavia, Gillespie (2010) afirma que as empresas de web 2.0 introduziram um sentido mais amplo da noção de "plataforma", que ultrapassa o seu sentido computacional:

> Esse uso mais conceitual de "plataforma" se baseia em todas as conotações do termo: computacional — algo no qual se pode construir e a partir do qual se pode inovar; política — um lugar a partir do qual se pode falar e ser

> ouvido; figurativa — a oportunidade é tanto uma promessa abstrata quanto prática; e arquitetônica — em que o YouTube é elaborado como um meio aberto e igualitário que facilita a expressão, e não um *gatekeeper* elitista com restrições técnicas e normativas (Gillespie, 2010, p. 352).

Gillespie (2010) argumenta que este uso conceitual possibilita que as plataformas tragam, junto delas, vários atores. O sentido computacional de plataforma se dirige para os desenvolvedores, enquanto as outras conotações se endereçam aos atores, tais como usuários, anunciantes e clientes (Gillespie, 2010). O autor descreve o que em termos econômicos é chamado "modelo de negócios multilateral" (*multi-sided market*), no qual uma plataforma possibilita interações entre duas ou mais partes distintas (Gillespie, 2010, p. 990). Um exemplo de plataforma multilateral é o Facebook, que conecta usuários, anunciantes, desenvolvedores, e experimenta efeitos de rede em que o valor aumenta para todas as partes à medida que mais pessoas o utilizam (Hagiu, 2014).

Na literatura sobre economia e administração, Annabelle Gawer argumenta que as plataformas têm sido teorizadas a partir de duas perspectivas distintas: "a teoria econômica conceitua as plataformas como mercados" (Rochet; Tirole, 2003), ao passo que "o design de engenharia as teoriza como 'arquiteturas tecnológicas modulares' (Baldwin; Woodard, 2009)" (Gawer, 2014, p. 1240). Bernhard Rieder e Guillaume Sire (2014, p. 197) fazem uma convocatória importante para juntarmos essas duas perspectivas: estudar as plataformas como multilaterais, eles argumentam, "pode ampliar as análises de configurações concretas de poder e identificar pontos de controle, dinâmicas estruturais e recursos importantes de argumentação" (Rieder; Sire, 2013, p. 208). Ao seguir essas perspectivas tecno-econômicas sobre plataformas, examinamos, neste capítulo, como a arquitetura técnica e modular de plataformas de mídias sociais se conecta com o modelo de negócios delas.

Em seu trabalho sobre as plataformas, Gillespie (2010) enfatiza os aspectos econômicos e de participação das plataformas em detrimento da dimensão computacional delas ao dizer que "plataformas são 'plataformas' não necessariamente porque possibilitam que códigos sejam escritos ou rodados, mas porque elas proveem uma oportunidade para comunicar, interagir ou vender" (Gillespie, 2010, p. 351). Outros autores, tais como Ian Bogost e Nick Montfort (2009), sugerem um foco mais estreito em relação às plataformas ao trazerem o aspecto computacional para primeiro plano. A seguir, o interesse deste trabalho está em desenvolver um pouco mais essa dimensão computacional das plataformas para analisar o "trabalho que as plataformas fazem", não em um sentido retórico

(cf. Gillespie, 2010), mas a partir de uma perspectiva técnico-material. Bogost e Montford (2009) refutam a ideia de que "nos dias de hoje, tudo é uma plataforma", e nos alertam para efetivamente tratarmos as plataformas como infraestruturas computacionais. Estes dois autores enxergam as plataformas, em seu sentido computacional, como uma camada das novas mídias que é pouco estudada (Bogos & Montfort, 2009). Para solucionar esse ponto cego, ambos introduzem o termo "estudo de plataforma" (*platform studies*), ou seja, uma chamada para "o rigor técnico e a investigação aprofundada de como as tecnologias computacionais funcionam" para analisar "a conexão entre especificidades técnicas e a cultura" (Bogost & Montfort, 2009, p. vii).

Essas conexões foram exploradas por vários autores engajados em uma perspectiva política sobre as plataformas para analisarem "as *affordances* tecnológicas das plataformas em relação aos seus interesses políticos, econômicos e sociais", considerando-as como locais importantes em que as "políticas de plataforma" se desenrolam (Hands, 2013; Langlois & Elmer, 2013).[1] As abordagens a respeito de políticas de plataformas incluem:

a) o questionamento crítico sobre o conceito de plataforma (Gillespie, 2010; Mckelvey, 2011),
b) a análise das "lógicas tecnoculturais" das plataformas (Gerlitz; Helmond, 2013; Langlois; Elmer; Mckelvey; Devereaux, 2009; Langlois et al., 2009),
c) a análise do papel da arquitetura de plataforma na configuração da socialidade em rede (Bucher, 2012a; Van Dijck, 2013),
d) e a análise das políticas de APIs (Bucher, 2013) e dos dados de plataforma (Puschmann & Burgess, 2013) (cf. Renzi, 2011).

Neste trabalho, tenho interesse pelo modo como as plataformas reformatam a web conforme a lógica das mídias sociais. Minha abordagem se baseia naquilo que Langlois e outros (2009) nomeiam "desagregação", e examina criticamente as plataformas de mídias sociais ao desmontá-las e investigar seus componentes específicos (Langlois & Mckelvey et al., 2009). Essa contribuição para os estudos de plataforma e de mídias sociais se encontra numa detalhada perspectiva técnico-material sobre o desenvolvimento e a emergência do que compreendemos

1 "Políticas de Plataforma" é o título de uma conferência realizada na *Anglia Ruskin University*, em Cambridge, no Reino Unido, nos dias 12 e 13 de maio de 2011. A conferência foi organizada por Josh Hands e Jussi Parikka, e reuniu vários estudiosos de políticas de plataforma. Após a conferência, a *Culture Machine* publicou uma edição especial com o título "Políticas de Plataformas" (2013).

hoje como plataformas de mídias sociais. Este argumento será desenvolvido adiante, ao focarmos no Facebook, uma das maiores e mais acessada plataforma de mídias sociais.[2]

FACEBOOK: SITE DE REDE SOCIAL OU PLATAFORMA?

Antes de o conceito de plataforma ganhar relevância, plataformas de mídias sociais, como o Facebook, eram muitas vezes conceituadas como sites de redes sociais, definidos por boyd e Ellison (2008, p. 211) como serviços web nos quais usuários podem criar um perfil, construir e disponibilizar uma lista de conexões com outros usuários na rede. Contudo, o Facebook sempre, cuidadosamente, absteve-se de chamar a si mesmo de rede social (Arrington, 2008; Locke, 2007). Em vez disso, ao longo do tempo, um dos fundadores do Facebook, Mark Zuckerberg, enquadrou a plataforma como um "diretório social" (Facebook Newsroom, 2006), um "utilitário social" (Facebook Newsroom, 2006), e uma "plataforma" (Facebook Newsroom, 2007). David Kirkpatrick (2010, pp. 215-217) descreve, em seu livro "O Efeito Facebook" (*The Facebook Effect*), como Zuckerberg sempre imaginou o Facebook como uma plataforma computacional para que outros aplicativos pudessem ser processados, desde o seu início, em 2004, como Thefacebook;

> Ele [Zuckerberg] quis fazer com a web o que Gates fez com o computador pessoal: criar uma infraestrutura de software padrão que tornasse fácil a construção de aplicativos — daquela vez, aplicativos que tivessem um componente social. "Queremos transformar o Facebook em algo como um sistema operacional, para que você possa executar aplicativos", ele [Zuckerberg] explicou. (Kirkpatrick, 2010, p. 217)

No outono de 2004, Zuckerberg trabalhava em outro projeto de software junto com o Thefacebook, o Wirehog, "um serviço de compartilhamento de conteúdo P2P (*peer-to-peer*)" (Kirkpatrick, 2010, p. 44). O aplicativo Wirehog foi integrado ao Thefacebook para utilizar as conexões de amizade dele a fim de compartilhar conteúdos no Thefacebook. Zuckerberg viu o Wirehog como "o primeiro exemplo que considerava o Facebook como uma plataforma para outros tipos de aplica-

2 Conforme o Facebook, a plataforma tinha mais de 936 milhões de usuários ativos, em média, em março de 2015 ("Company Info," s.d.). O Facebook.com é o segundo colocado na classificação dos "Top 500 sites da web", feita pelo site Alexa, cuja ordem é "calcula usando uma combinação da média de visitantes diários e visualizações de páginas nos último mês" ("The top 500 sites on the web", 2015).

tivos" (Kirkpatrick, 2010, pp. 99-100). Então, em vez de uma rede social, Mark Zuckerberg viu e elaborou o Facebook como uma plataforma, desde o seu início. O desenvolvimento do Facebook como uma plataforma deve ser percebido no vasto contexto da web 2.0 — "a web como plataforma" (O'Reilly, 2005) —, no qual a web foi situada como plataforma de desenvolvimento.

WEB 2.0: A WEB COMO PLATAFORMA

Os sites de redes sociais são tipicamente classificados como um tipo específico de aplicativo da web 2.0 (Beer & Burrows, 2007) ou um tipo dentre as mídias sociais (Van Dijck, 2013, p. 8). O termo "sites de redes sociais" se tornou popular na primeira conferência sobre web 2.0, em 2004, quando Tim O'Reilly definiu a web 2.0 ao considerar "a web como plataforma" — frase utilizada para situar a web como uma "robusta plataforma de desenvolvimento" na qual "websites se tornam componentes de softwares" (O'Reilly & Battelle, 2004). A web 2.0 ou "a web participativa" é agora compreendida como um grande conjunto de serviços que promovem a colaboração e a participação (Madden & Fox, 2006).[3]

O'Reilly colocou o sentido computacional do termo "plataforma" no centro do conceito de "web como plataforma". Com o advento da web 2.0, O'Reilly (2005) não mais via a web apenas como um meio para a publicação de informações — que ele retrospectivamente rotulava de web 1.0 —, mas como uma infraestrutura para a construção de aplicativos, um sistema operacional de distribuição, que poderia fornecer serviços de software. Por esse motivo, Matthew Allen (2013) argumenta que deveríamos ver a web 2.0 como uma "tecnologia retórica" na qual "a indústria de computação tentou mudar a maneira como pensamos a internet" (Allen, 2013, p. 264): de um canal de publicação para uma plataforma de desenvolvimento de software.

Todavia e depois da conferência mencionada, essa definição mais computacional da web 2.0, a "web como plataforma", não teve continuidade, tal como argumenta Robert Gehl (2010). Em vez disso, esse autor afirma que a web 2.0 era vista como um renascimento da indústria após o abalo do modelo "ponto com", e ainda mais no âmbito de debates públicos e acadêmicos, como uma revolução que reformularia o panorama midiático (Gehl, 2010, pp. 26-37). As tecnologias da web 2.0 eram vistas como aquilo que borrava as fronteiras entre produção e

3 Conferir a crítica feita por Michael Stevenson (2014) a respeito da suposta "virada participativa" da web.

consumo (Bruns, 2008), e faziam emergir novas formas de participação de usuários como parte de uma "cultura participativa" (Jenkins, 2006) online. Desse modo, embora a definição original de web 2.0 implicasse considerar a web como plataforma computacional, o termo seria incorporado em um sentido mais metafórico (cf. Gillespie, 2010), como plataforma de participação, associada à retórica de "empoderamento" e "democratização" (Beer, 2009, p. 986).

DE SITES DE REDES SOCIAIS PARA PLATAFORMAS DE MÍDIAS SOCIAIS

Para alternar o foco dessa definição conceitual mais ampla de plataformas de volta para uma compreensão computacional mais específica, e a fim de desenvolver uma crítica a respeito de considerar o Facebook como plataforma, procuramos explorar o desenvolvimento tecnológico de plataformas de software na web e, em particular, plataformas de mídias sociais. Fazemos isso ao recorrermos a outra definição computacional de plataforma, que foi proposta por Marc Andreessen (2007a), fundador da Netscape, em uma publicação que discutia o lançamento da plataforma Facebook:

> Definitivamente, uma "plataforma" é um sistema que pode ser reprogramado, portanto, customizado por outros desenvolvedores externos — usuários — e, dessa forma, adaptado a incontáveis necessidades e nichos, até então não pensados ou sequer ajustados pelos desenvolvedores originais.

Para Andreessen (2007b), o termo principal nessa definição de plataforma é "programável", que elimina os usos mais conceituais do termo: "Se você pode programá-la, então ela é uma plataforma. Se não pode, então ela não é uma plataforma."

A programabilidade de plataformas web 2.0, argumenta McKelvey (2011), oferece uma nova linha de crítica nos estudos de plataforma que iniciam com o questionamento a respeito de como uma plataforma performa a sua programabilidade. A noção de programabilidade tem sido fundamental para a compreensão da lógica das novas mídias (Chun, 2011; Manovich, 2001)[4] e, por extensão, figura central na análise da lógica subjacente das plataformas de mídias sociais.

4 Em seu livro seminal, intitulado "A Linguagem das Novas Mídias", Lev Manovich (2001) argumenta que todos os objetos das novas mídias são representações numéricas, e é isso o que torna as mídias

A fim de investigar as pré-condições da programabilidade de plataformas de mídias sociais, baseamo-nos na definição de plataforma de software proposta por Evans, Hagiu e Schmalensee (2006, p. vii): "um programa de software que torna serviços disponíveis para outros programas de software por meio de Interfaces de Programação de Aplicativos (APIs)". Decorre dessa definição que, para se tornar uma plataforma, um programa de software — ou um website — precisa oferecer uma interface que possibilita a sua reprogramação: uma API.

> Uma API é uma interface proporcionada por um aplicativo, que possibilita aos usuários interagirem com — ou em resposta a — dados ou solicitações de outros programas, outros aplicativos ou websites. As APIs facilitam a troca de dados entre os aplicativos, possibilitam a criação de novos aplicativos, e constroem a base para o conceito de "web como plataforma". (Murugesan, 2007, p. 36)

De volta à concepção de O'Reilly, que considera a web como plataforma de desenvolvimento para novos serviços, apreendemos que não apenas a web como um todo, mas também os websites são transformados em plataformas ao tornarem disponível uma API.[5] Por exemplo, o Facebook é uma plataforma porque ele disponibiliza uma API, que pode ser usada por desenvolvedores e proprietários de sites para construírem novos serviços no Facebook e integrarem novos websites e aplicativos aos dados e às funcionalidades do Facebook.[6] O aplicativo de encontros *Tinder* é um exemplo de um aplicativo que foi feito sobre a plataforma do Facebook: ele requer que os usuários façam o *login* com o Facebook, e se utiliza de dados do próprio Facebook, tais como curtidas e amigos em comum, para encontrar potenciais parceiros. Outra maneira de integração com o Facebook é demonstrada pelo ato de implementar funcionalidades específicas dessa plataforma, como o botão curtir (*like*), em outras páginas/sites.

Na "web como plataforma", os websites podem ter duas interfaces diferentes: uma interface de usuário, para o consumo humano (ex.: Facebook.com), e uma interface de software, para o consumo de máquinas (ex.: Facebook Graph API). Essa interface de software, a API, torna programável um website ao oferecer acesso estruturado aos seus dados e funcionalidades, e também o torna

programáveis. Essa programabilidade é fundamental para os princípios subjacentes às novas mídias: representação numérica, modularidade, automação, variabilidade e transcodificação cultural.

5 Isso chama nossa atenção para o uso da plataforma como um conceito recursivo, tal como apresentado pelo desenvolvedor de software Dave Winer (1995), quem desde cedo viu a internet como uma meta-plataforma ou "plataforma máquina" que pode ser usada para construir novas plataformas.

6 Mais de 30 milhões de aplicativos e websites foram integrados com a plataforma do Facebook (Liu, 2015).

uma plataforma na qual outros podem construir. Para ampliar essa ideia, as APIs são colocadas no centro da mudança de *sites* de redes sociais para *plataformas* de mídias sociais. No momento que os sites de redes sociais disponibilizam APIs, eles se tornam plataformas de mídias sociais ao disponibilizarem sua programabilidade. A API, então, torna-se um lócus fundamental para a investigação das consequências dessa programabilidade.

A ASCENSÃO DAS APIS DE MÍDIAS SOCIAIS

No campo dos estudos de mídias, as APIs de mídias sociais têm sido compreendidas como a "cola" tecnológica da web social ao conectarem serviços e possibilitarem o compartilhamento de conteúdos (Bodle, 2011; Bucher, 2013; Langlois; Mckelvey et al., 2009), como objetos protocolares (Bucher, 2013), como instrumentos regulatórios que governam relações entre a plataforma e terceiros (Puschmann & Burgess, 2013), como o modelo de negócios da web social[7] (Bodle, 2011; Bucher, 2013), e como ferramentas que constroem dados para o mercado de dados (Vis, 2013). Predominantemente, as APIs têm sido usadas e discutidas como "um método para a coleta de dados em plataformas de mídias sociais" (Lomborg & Bechmann, 2014). Contudo, menos atenção tem sido dedicada à história das APIs de mídias sociais,[8] isto é, a emergência delas na web como parte da infraestrutura material das plataformas de mídias sociais e as consequências delas para a adaptação de um modelo de plataforma. Um dos relatos mais completos até agora tem sido o do blogueiro de tecnologia Kin Lane (s. d.), que se nomeia "evangelista da API" e que estuda "os negócios e as políticas das APIs" desde 2010.

Lane (2012) traça o aparecimento histórico das APIs da web, voltadas para desenvolvedores externos, ao início dos anos 2000, quando a *Salesforce*, em 1999, que virou *eBay* em 2001, e a *Amazon*, em 2002, começaram a oferecer as APIs como soluções para o comércio eletrônico entre empresas. Essa primeira geração de APIs da web, sobretudo disponibilizada por empresas de comércio eletrônico, focou na troca de dados entre diferentes aplicativos de negócios para possibilitar transações e gerenciamento de vendas (Lane, 2012). A plataforma de serviços web

7 A indústria se refere a esse modo como "Economia de API" ou "os efeitos econômicos emergentes possibilitados por empresas, governos, organizações sem fins lucrativos e indivíduos que usam APIs para fornecerem acesso programável direto aos seus sistemas e processos" (Willmott & Balas, 2013).

8 Uma exceção dentro dos estudos de mídias é o trabalho de Taina Bucher (2013) acerca de "Objetos de sentimento intenso: o caso da API do Twitter" no qual a autora apresenta um contexto histórico do papel das APIs na engenharia de software e discute brevemente as primeiras APIs públicas da web.

da *Amazon*, por exemplo, possibilitou que websites terceirizados pesquisassem em seu catálogo, disponibilizassem produtos da própria *Amazon*, e ganhassem comissões por recomendações de seus próprios sites. Ao fazer isso, a API da *Amazon* ampliou seus serviços de comércio eletrônico a outros websites.

Em meados dos anos 2000, uma nova geração de APIs da web, ofertadas por sites de redes sociais, alterou o foco das transações de vendas para o acesso de conteúdos gerados por usuários, informações de usuários e suas conexões (Lane, 2012).

Em 2003, o site de marcações sociais (*social bookmarking*) del.icio.us começou a oferecer acesso programático ao seu site, seguido pelo Flickr, em 2004, pelo YouTube, em 2005, pelo Last.fm, Facebook e Twitter, em 2006, e depois por vários outros sites de redes sociais, que passaram a anunciar as suas APIs (Du Vander, 2012; Lane, 2012). Robert Bodle (2011, p. 325) descreve como esses sites disponibilizaram seus conteúdos e funcionalidades como parte de uma estratégia de negócios na qual terceiros podem adicionar valor a uma plataforma, por meio da construção de novos serviços tendo por base tal plataforma. O autor explica como Tim O'Reilly apoiou diferentes negócios a buscarem uma estratégia de plataforma, por meio da abertura de seus valiosos dados, a fim de tocarem no bloqueio de plataforma (Bodle, 2011, p. 325). Em seu manifesto pela web 2.0, O'Reilly (2005) encorajou ainda mais a reutilização de dados ao recomendar que eles fossem "projetados para a 'hackeabibilidade' e remixabilidade", por meio do acesso a tais dados e serviços por parte de terceiros. O'Reilly (2005) considerou os dados como os "blocos de construção" da web 2.0. Esse acesso fez emergir a típica prática da web 2.0, a criação de *mashups*, isto é, a construção de novos aplicativos por meio da remixagem de dados e funcionalidade proveniente de fontes existentes, que se utilizam de APIs (Benslimane, Dustdar & Sheth, 2008). Por isso, a web 2.0 também se tornou conhecida como "a web programável" (Anderson, 2012; O'Reilly, 2005). A seguir, analisamos os diferentes tipos de programabilidade que as plataformas de mídias sociais oferecem por meio de suas APIs, com a finalidade de formularmos uma crítica que considere o Facebook como plataforma ao enfatizar suas condições distintas de programação e suas consequências.

NÍVEIS E CONDIÇÕES DE PROGRAMABILIDADE

Em uma postagem sobre a nova plataforma do Facebook, Marc Andreessen (2007b) explicou como a programabilidade das plataformas da internet pode ser facilitada em diferentes níveis, de modo a produzir o que ele considera como três

tipos de plataformas de internet. Esses níveis também podem servir como uma maneira de investigação crítica do papel da arquitetura de plataforma.

De acordo com Andreessen (2007b), a maioria das plataformas de mídias sociais fornecem o tão chamado Nível 1 ou "API de acesso" (*Access* API). Neste nível, desenvolvedores externos podem acessar dados de uma plataforma e sua funcionalidade por meio de solicitações, feitas via API, que representam operações específicas para a realização de uma tarefa, por exemplo, ler, escrever ou apagar dados (Andreessen, 2007b). A API é acessada "do lado de fora do sistema principal", ou seja, "o código dos aplicativos dos desenvolvedores permanece fora da plataforma" (Andreessen, 2007b). O serviço de compartilhamento de fotos, Flickr, é um exemplo de uma API de acesso, em que um desenvolvedor pode construir um aplicativo terceirizado, tal como um visualizador de slides para mostrar fotos marcadas como "pôr do sol", ao usar a API do Flickr para requisitar esses dados. Nesse cenário, o código do aplicativo está localizado em um servidor externo, e o aplicativo está hospedado fora do Flickr. A programabilidade de uma plataforma de Nível 1 é caracterizada pelo simples acesso aos dados e à funcionalidade. Os desenvolvedores podem construir novos aplicativos com base na plataforma e integrarem dados e funcionalidades a seus sites e aplicativos externos, mas não podem reprogramar a própria plataforma. Em outras palavras, a programabilidade de plataformas de Nível 1 é um modo de as plataformas se expandirem para além de seus limites.

A "API Plug-In", de Nível 2, possibilita aos desenvolvedores "construírem novas funções que podem ser introduzidas ou plugadas ao sistema principal e à sua interface para usuários" (Andreessen, 2007b). Andreessen utiliza o Facebook como exemplo de API Plug-In, visto que essa plataforma não apenas possibilita aos desenvolvedores acessarem dados e funcionalidades do Facebook para construir novos aplicativos (API de acesso Nível 1), mas também possibilita aos desenvolvedores carregarem e usarem seus aplicativos no ambiente[9] do Facebook, por meio do *Canvas Frame*. O *Canvas* é "um painel gráfico no qual se possa colocar seu aplicativo ou jogo diretamente no Facebook.com" a fim de "integrá-los completamente à experiência principal do Facebook" (Facebook Developers, s. d.-a).[10]

9 Em 24 de março de 2015, o Facebook lançou a plataforma Messenger, que "possibilita aos desenvolvedores facilmente construírem aplicativos que se integram com o Messenger", sendo este o aplicativo de troca de mensagem do Facebook. Por meio da plataforma Messenger, os desenvolvedores podem plugar seus aplicativos ao Messenger, a nova plataforma de nível 2 do Facebook.

10 Em sua documentação para desenvolvedores o Facebook explica como isso funciona: "Aplicativos no Facebook são páginas carregadas no *Canvas frame*, que é um *canvas* em branco no Facebook no qual se pode rodar seu aplicativo. Você preenche o *Canvas frame* ao fornecer uma *Canvas* URL e uma *Secure Canvas* URL que contém o HTML, o JavaScript e o CSS que compõem o seu aplicativo. Estes serão

Enquanto o aplicativo roda no Facebook, o código desse aplicativo está localizado fora da plataforma do Facebook (Andreessen, 2007b).[11] Aplicativos de tela (*canvas apps*) possibilitam que usuários customizem suas experiências no Facebook, aspecto que chama atenção à reconsideração feita por McKelvey (2011) a respeito da ideia de "programar como um ato de composição", elaborada por John van Neumann.

No Nível 3, referente à API "Ambiente de tempo de execução" (*Runtime Environment* API), aplicativos terceirizados rodam com o ambiente de tempo de execução da própria plataforma (Andreessen, 2007b). Andreessen explica que essa abordagem é mais similar às "tradicionais" plataformas de computação, como o sistema operacional Windows, no qual desenvolvedores constroem aplicativos para serem executados nesse próprio sistema (Andreessen, 2007b). A plataforma como "ambiente de tempo de execução" é a abordagem menos comum na web, uma vez que ela requer um sistema técnico mais complexo tanto para desenvolvedores quanto para bancos de dados e gerenciamento de armazenamento (Andreessen, 2007b).[12] Como consequência, a programabilidade de plataformas de mídias sociais é tipicamente habilitada por meio da API de Acesso e da API Plug-In ou por ambas. De modo mais específico, nos termos de Andreessen (2007b), o tipo mais comum de plataforma de mídia social é a de Nível 1, a API de Acesso (Twitter, Facebook, YouTube, Tumblr e Instagram), seguido pela API Plug-In, de Nível 2 (Facebook).[13]

Ao diferenciar dois tipos de plataformas, Andreessen (2007b) oferece um sistema com o qual podemos avaliar plataformas individuais baseadas em suas condições de programabilidade. Similarmente, ao recorrer à tipologia de interfaces proposta por Florian Cramer e Matthew Fuller (2008), McKelvey (2011)

utilizados pelos usuários que navegam no Facebook em HTTP e HTTPS, respectivamente. Quando um usuário carrega seu aplicativo *Canvas* no Facebook, nós carregamos o URL da tela dentro de um *iframe* nessa página. Isso resulta na exibição de seu aplicativo dentro do cromo padrão do Facebook." (Facebook Developers, s. d.-a)

11 A Canvas URL aponta para o *host* externo em que o aplicativo é localizado, que então é carregado dentro de um *iframe* no Facebook.

12 Exemplos de plataformas de ambiente de tempo de execução de nível 3 mencionados por Andreessen incluem o Salesforce, que possibilita aos usuários inserirem o próprio código, e a plataforma Ning, do próprio Andreessen (2007b), que serve "para a criação e a execução de aplicativos de redes sociais". Apesar de Andreessen alegar que todas as "plataformas são boas, ponto final", ele afirma que chama "esses modelos de plataformas de internet de 'níveis' porque, como você vai do nível 1 para o nível 2, e então, para o nível 3, como será explicado, cada tipo de plataforma é mais difícil de construir, mas muito melhor para quem a desenvolve". Nesse sentido, ele promove plataformas de nível 3, incluindo a sua própria, a *Ning*, como sendo esse tipo de plataforma a melhor para desenvolvedores.

13 As plataformas "ambiente de tempo de execução" de nível 3 estão localizadas principalmente no domínio *business-to-business*, como a Salesforce ou a Amazon.

argumenta que "posto que plataformas têm interfaces diferentes, a linha crítica possibilita comparar como as plataformas facilitam a programabilidade". De modo mais claro, podemos comparar APIs de mídias sociais para analisarmos que tipo de programabilidade essas plataformas vislumbram, o que elas possibilitam e restringem, que tipo de funcionalidades e dados são disponibilizados aos usuários, e por quem eles são disponibilizados.

PLATAFORMIZAÇÃO DA WEB

O termo "plataformização" é utilizado para se referir à emergência da plataforma como modelo econômico e infraestrutural dominante da web social, bem como às consequências da expansão das plataformas de mídias sociais em outros espaços online. É central a esse processo a oferta de APIs, que tornam os sites de redes sociais em plataformas de mídias sociais. Para compreender esses efeitos, analisamos como as distintas condições de programabilidade de plataformas de mídias sociais possibilitam que tais plataformas se estendam pela web e formatem dados da web que lhe são externos. De modo mais claro, as plataformas promovem sua programabilidade para descentralizar a produção e recentralizar a coleta de dados (Gerlitz & Helmond, 2013).

Historicamente, os websites tornaram viável a sua programabilidade por meio da troca de dados, conteúdos e funcionalidades com terceiros de três maneiras, de modo a oferecerem as condições prévias de plataformização da web: a) separação entre conteúdo e apresentação, b) modularização de conteúdo e funcionalidades, e c) interface com base de dados.

Separação entre conteúdo e apresentação

A maioria dos websites são criados com o uso da Linguagem de Marcação de Hipertexto (*HyperText Markup Language* — HTML), que descreve o conteúdo e a apresentação de um documento da web. Uma vez que a HTML é uma tecnologia de apresentação, projetada para o consumo humano, e muitos sites HTML são mal formatados, é difícil para uma máquina extrair e processar informação estruturada proveniente de um website (MYLLYMAKI, 2002, p. 635). A Linguagem de Marcação Estendida (*Extensible Markup Language* — XML) aborda essas questões ao separar conteúdo, estrutura e apresentação em um formato baseado

em texto, para consumo de máquina (W3C, s. d.).[14] Esses formatos, legíveis para máquinas e pessoas, possibilitam o compartilhamento de informação estruturada entre sistemas, até então, incompatíveis (Myllymaki, 2002, p. 635; W3C, s. d.). A XML tem sido uma evolução extremamente importante para a web, pois possibilita que os dados de websites sejam lidos por máquinas e intercambiáveis entre sistemas diferentes. Essa linguagem possibilita a formatação estruturada de informação para fins de transmissão, de modo a formar a base para vários mecanismos de troca de dados, que possibilitam o fluxo de dados dentro e fora de outros websites[15]. Richardson e Ruby (2008) argumentam que a linguagem XML é fundamental para tecnologias como RSS, XML-RPC e SOAP, as quais "formaram uma web programável, que estendeu a web humana para a conveniência dos programas de software" (Richardson & Ruby, 2008, p. xviii).[16]

De acordo com Liu (2004), a separação entre conteúdo e apresentação, por meio da linguagem XML, explicita o conhecimento tecnológico subjacente da "transmissão de informação, pós-industrial", a qual requer que o conteúdo seja "transformável", "autonomamente móvel" e "automatizado" (LIU, 2004, pp. 57-58). Essa separação, conforme Liu (2004, p. 59), torna o conteúdo "transcendental", a fim de que ele possa ser despejado de um recipiente para outro, movendo-se do banco de dados para o "banco de dados na web". Liu (2004) também descreve como a linguagem XML sinaliza uma mudança da primeira geração de sites HTML independentes para um novo tipo de website, que é preenchido com dados provenientes de bancos de dados externos (LIU, 2004, p. 57). Essas novas páginas web

14 Estrutura de um documento XML:
<book category="Fiction">
<title lang="en">Emma</title>
<author>Jane Austen</author>
<year>1916</year>
</book>

15 A linguagem XML está no centro de vários mecanismos importantes de troca de dados na web, incluindo XML-RPC e SOAP. O protocolo XML-RPC é baseado na ideia de procedimento remoto, chamada (RPC), para "fornecer transferência de controle e dados uma rede de comunicação" (Birrell & Nelson, 1984, p. 39). Foi desenvolvido, em 1998, por Dave Winer da Userland e Microsoft, para fazer pedidos para um computador remoto e trocar dados na web (Laurent; Johnston; Dumbill & Winer, 2001, p. x). O SOAP (*Simple Object Access Protocol*) é um "protocolo leve, usado para troca de informação codificada em XML" (Laurent et al., 2001, p. 172). Os serviços da web baseados em XML-RPC e SOAP possibilitam a troca de dados estruturados entre diferentes máquinas na web, que se comunicam por meio do protocolo de transmissão HTTP. Recentemente, o JSON se tornou o formato preferido para transmitir dados, em detrimento do XML, considerado um formato mais leve. Além disso, o estilo arquitetônico REST (*Representational State Transfer*), ganhou destaque para a construção de serviços da web. Por exemplo, a plataforma de mídias socias Twitter oferece uma API REST que retorna dados em JSON.

16 Conferir a nota anterior. O RSS é um formato de distribuição da web para sites e blogs, para publicar um *feed* de seu conteúdo mais recente. Ele se baseia em XML, conferir:<http://cyber.law.harvard.edu/rss/rss.html>.

empregam, para extraírem e exibirem conteúdo dinâmico de terceiros, o que Liu (2004) nomeia "fluxo de dados" (*data pours*). O fluxo de dados é um código incorporado em uma página web, que demarca um espaço nessa página, o qual transfere dados a partir de bancos de dados externos e para tais bancos (Liu, 2004, p. 59).

Publicada nos primeiros dias da web 2.0, a ideia de "fluxo de dados" de Liu (2008) pode ser lida como uma reflexão antecipada a respeito da modularidade crescente da web, que foi, depois, atualizada por ele:

> Minhas presentes observações a respeito dos "fluxos de dados" se aplicam, com ainda mais força, à web 2.0, onde conteúdos produzidos por usuários fluem tanto para dentro quanto para fora de bancos de dados (*back-end databases*), por meio de páginas web "modelo" (*template*), que muitas vezes são elegantes e projetadas de modo minimalista, isto é, construídas em volta de uma abertura cega de código parametrizado — como um buraco negro reversível — que soca todo o conteúdo para dentro e o arremessa para fora novamente (Liu, 2008, p. 320).

Os "fluxo de dados", agora comuns à web 2.0, estabelecem canais de dados, para os fluxos de dados, entre websites e banco de dados externos.

Modularização de conteúdos e recursos

Ao separar conteúdo de apresentação, a linguagem XML compartimentaliza o conteúdo web, por meio da descrição estruturada de cada elemento em uma página web, e da transformação desses pequenos módulos de dados, que podem ser reutilizados. A compartimentalização de conteúdo torna os conteúdos existentes na web disponíveis para o consumo de máquina, e possibilita a circulação de conteúdo por meio de elementos modulares. A modularização é um aspecto fundamental no moderno design de software, que possibilita o gerenciamento de sistemas complexos ao dividi-los em módulos menores e encorajar a reutilização desses módulos (Baldwin & Clark, 2000; Gehl, 2012; Mckelvey, 2011). Ulrich e outros (2008) argumentam que, na web 2.0, "serviços muitas vezes disseminam sua funcionalidade ao acoplarem componentes modulares, nomeados *widgets*", quer dizer, "uma arquitetura de plataforma exibe um tipo especial de modularidade, no qual um produto ou sistema é dividido em um conjunto de componentes" (Baldwin & Woodard, 2009, p. 25). Tais *widgets* possibilitam a integração de

um conteúdo e funcionalidade de determinado serviço com outro website, o qual conta com algumas linhas de código, que criam um fluxo de dados. Os *widgets* se tornaram centrais, pois são objetos específicos para plataformas de mídias sociais distribuírem seus conteúdos em diferentes espaços da web e se estenderem pela web.

Um importante desenvolvimento referente a essa extensão veio do site de compartilhamento de vídeos YouTube. Em 7 de julho de 2005, o YouTube (2005) anunciou um novo recurso, que possibilita aos usuários inserirem uma lista de seus vídeos no YouTube em seus próprios websites, ao copiarem e colarem o código HTML gerado. Este código incorporou um *widget* do YouTube e mostra uma lista de vídeos e miniaturas de imagens (*thumbnails*) que se vinculam aos vídeos no YouTube. Um mês depois, o YouTube anunciou um novo *widget*, que incorporou o *player* de vídeos, de modo que os vídeos do YouTube podem ser agora diretamente iniciados a partir de qualquer website (YouTube, 2005). O *widget* possibilitou a distribuição e a visualização de vídeos do YouTube fora desse mesmo site. Esse recurso de incorporação de vídeos é comumente visto como um fator importante do sucesso do YouTube, pois possibilitou ao YouTube fazer circular vídeos em redes sociais online, blogs e outras partes da web, por meio da modularização e descentralização dos recursos de sua plataforma (Cheng, Dale & Liu, 2008).

Enquanto o YouTube criou seus próprios *widgets* para distribuir conteúdos *fora* de seu próprio website, a rede social online MySpace teve um importante papel na popularização de *widgets* terceirizados, que compartilham conteúdos *dentro* dessa rede. Em contraste com outras redes sociais online populares em 2005 e 2006 — como o Friendster —, o MySpace possibilitou aos usuários inserirem códigos em suas páginas de perfil e adicionar botões de inicialização de músicas, álbuns de fotos e vídeos. Esta foi a primeira rede social online que apresentou tal arquitetura aberta e, com isso, fez surgir uma cultura de customização e assessoria de perfis (Boyd, 2007).

Com a habilidade de inserir códigos que podem ser incorporados em páginas de perfis, desenvolvedores terceirizados começaram a criar *widgets* a aparência e a funcionalidade do MySpace. Em novembro de 2005, a RockYou lançou o seu primeiro *widget* para o MySpace, em *flash*, para a criação e disponibilização de *slideshows* de fotos. Um aspecto importante desses *widgets* iniciais é que, diferentemente dos *widgets* de compartilhamento do YouTube, eles não interagiam diretamente com o banco de dados do MySpace. Os usuários não podiam carregar suas fotos diretamente do MySpace para o *widget* porque esse site não oferecia acesso estruturado (por meio de uma API ou outro recurso) a essas fotos. Em vez disso, os usuários tinham que, primeiramente, carregar

suas fotos para o site externo de hospedagem, ImageShack, dentro do *widget* RockYou (Tokuda, 2009).

Esta falta de uma interface direta com o banco de dados do MySpace diz respeito àquilo que Gehl (2012) se refere como a "falha de abstratação" do MySpace para extrair e monetizar conteúdo a partir de sua própria rede (Gehl, 2012, pp. 111--112). Ao passo que os *widgets* do MySpace estavam mais orientados a integrarem e distribuírem conteúdo em sua própria rede, os *widgets* do YouTube estavam orientados para a distribuição de conteúdo e funcionalidade fora de sua própria rede. Como muitos sites 2.0 começaram a oferecer códigos de incorporação e *widgets* para distribuírem seus conteúdos na web, a descentralização de recursos de plataformas se tornou central. Uma segunda distinção importante, distinta dos *widgets* do MySpace, diz respeito à interação direta dos *widgets* do YouTube com o banco de dados desse site. Todavia, os *widgets* voltados para o banco de dados do YouTube eram baseados em *fluxo de dados unidirecional (one-way data streams)*, nas dinâmicas de descentralização, em que o conteúdo era recuperado a partir do banco de dados e disposto em um website externo. A nova geração de *widgets* seria baseada na interação direta com bancos de dados, a fim de possibilitar o *fluxo de dados bidirecional (two-way data streams)*, nas dinâmicas de descentralização e recentralização, não apenas para ler os dados de bancos de dados, mas também para escrever novos dados neles.

Interação com bancos de dados

Os *plug-ins* sociais do Facebook são um conjunto de ferramentas, ou *widgets*, incluindo o ubíquo botão "curtir" (*like*), "que te possibilita compartilhar a sua experiência com o Facebook com seus amigos e outros no Facebook" (Facebook Help, s. d.). Os *plug-ins* funcionam como módulos que ampliam as funcionalidades das plataformas para websites externos (cf. Bodle, 2011). Ao mesmo tempo, Tania Bucher (2012b) argumenta: eles funcionam como "dispositivos criadores de conexões", coletando dados criados pelas conexões ou "arestas" fora do Facebook.com, e os enviando de volta ao banco de dados da plataforma (Bucher, 2012b, p. 6). Os *plug-ins* sociais são uma parte importante da arquitetura de plataforma do Facebook, habilitando a descentralização da funcionalidade da plataforma e de dados nela produzidos, e possibilitando a recentralização de dados produzidos fora dela (Gerlitz & Helmond, 2013). Ao incorporarem um *plug-in* em seu website, donos e criadores de sites configuram canais bidirecionais de dados, fluxos de dados, nos quais os dados fluem entre o website e o banco de dados do Facebook.

Tecnicamente, um *plug-in* social funciona como uma chamada ou ligação à API (Helmond, 2013) e envia requisições específicas para a plataforma do Facebook. Por exemplo, obtém o número total de pessoas que gostaram de tal publicação ou publicaram um novo "curtir" (*like*) depois de clicar no botão "curtir".

PREPARAÇÃO DE DADOS PARA AS PLATAFORMAS

Antes que esses *plug-ins* possam interagir com o banco de dados do Facebook a partir de um website externo, donos e criadores de sites precisam tornar seus websites compatíveis com a infraestrutura da plataforma do Facebook. Para fazer isso, os donos e criadores de sites precisam incorporar uma parte do código *JavaScript* em seus websites, que estabelece um canal de comunicação de dados com a plataforma do Facebook. Esse código inicializa o Facebook *Software Development Kit* (SDK) para a utilização de *plug-ins* sociais, Facebook *login*, e fazer chamadas à API, para os bancos de dados (Facebook Developers, s. d.-e). Ao fazerem isso, os donos e criadores de sites estão preparando suas páginas, como plataformas, para a comunicação de dados com o Facebook. Essa noção de preparar websites externos e dados web de plataformas amplia a ideia proposta por Gillespie (2014) referente a como os dados são "preparados como algoritmos" (Gillespie, 2014, p. 168) para realçarem o papel da infraestrutura de plataforma em reconfigurarem dados externos para se adequarem à agenda da plataforma.

Uma outra parte importante da infraestrutura do Facebook é o Facebook *OpenGraph*, que é explicitamente voltado para tornar os dados externos, prontos para platforma. O *OpenGraph* "te possibilita integrar aplicativos diretamente à experiência do Facebook, o que aumenta o engajamento, a distribuição e o crescimento" (Facebook Developers, s. d.-d). Para integrar um aplicativo, os desenvolvedores precisam utilizar o Facebook SDK e o Facebook *Login* para configurarem relações entre o aplicativo, o Facebook e o usuário (Facebook Developers, s. d.-d). Essa integração possibilita aos aplicativos contarem "histórias" no Facebook tal como "Maria correu seis milhas com o *MyRunningApp* (Facebook Developers, s. d.-d). Os aplicativos submetem essas histórias ao *OpenGraph* de uma maneira muito estruturada, organizada em volta de quatro elementos, por exemplo: João (ator) está lendo *A Odisseia* (o objeto) no *Goodreads* (aplicativo). Há um número de ações pré-definidas tais como: "curtir", "visualizar" e "ler", mas os desenvolvedores também podem criar suas próprias ações. Bucher (2012b, p. 5) descreve esses esforços do Facebook "como um caminho para a construção de um mapa semântico da internet". As integrações de aplicativos possibilitam que o Facebook colete

dados externos de aplicativos e atividades de uma maneira bastante estruturada, envie-os de volta ao banco de dados e os conecte a um usuário ou a outro dado. Isso amplia ainda mais as técnicas de coleta de dados no Facebook para aplicativos externos, e formata tais dados de acordo com a lógica da plataforma, para que, então, os dados sejam colocados em novas relações com a plataforma.

Donos e criadores de sites também podem preparar suas plataformas de websites ao marcarem esses sites com as *tags OpenGraph* (Facebook Developers, n.d.-b). Essas meta *tags* alimentam o *crawler* do Facebook com "informações estruturadas a respeito de uma página, tais como: título, descrição, imagem prévia etc.", e controlam como o conteúdo aparece no Facebook a fim de "aprimorar a distribuição e o engajamento" (Facebook Developers, s. d.-c). De modo similar à prática de Otimização de Mecanismo de Busca (*search engine optimization* — SEO), realizada por donos e criadores de sites, preparar websites para o Facebook pode ser vista como uma forma de otimização de mídias sociais.

O *OpenGraph* mostra como o Facebook rigorosamente estrutura o fluxo de dados de aplicativos e websites externos para a plataforma a fim de prepará-lo. Enquanto as plataformas se posicionam como intermediárias neutras ou utilidades (Gillespie, 2010; Van Dijck, 2013), elas (pré)formatam a passagem de dados, por meio de sua infraestrutura, de acordo com a lógica de suas infraestruturas subjacentes.

DUPLA LÓGICA DE PLATAFORMIZAÇÃO

Os exemplos anteriores mostraram como o Facebook emprega a sua plataforma como um modelo de infraestrutura para ampliar a si mesmo para espaços online externos, e como ele emprega essas extensões para formatar dados para que a sua plataforma se encaixe em seus interesses econômicos, por meio da mercantilização das atividades de usuários e conteúdos da web e de aplicativos. Essa plataformização, eu argumento, está baseada na lógica dual de expansão das plataformas de mídias sociais para o resto da web e, simultaneamente, no movimento delas para prepararem dados externos da web e de aplicativos para fins de plataforma.

Como modelos de infraestrutura, as plataformas de mídias sociais ofertam um sistema tecnológico no qual outros possam construir, adaptam-se à conexão e alcançam resultados em outros websites, aplicativos, bem como em relação aos dados deles. Ao mesmo tempo, a preparação de dados externos para os seus próprios bancos de dados é fundamental para o modelo econômico das

plataformas de mídias sociais. Esse duplo processo — descentralização de recursos de plataforma e recentralização de dados prontos para plataforma — caracteriza o que chamamos dupla lógica de plataformização. Essa dupla lógica é operacionalizada por meio de objetos oriundos da própria plataforma, tais como: APIs, *plug-ins* sociais e o *OpenGraph* — que conecta o modelo de infraestrutura da plataforma aos seus objetivos econômicos. Esses elementos servem como dispositivos primordiais para que plataformas de mídias sociais se estendam pela web e para a criação de canais de dados — fluxos de dados — a fim de coletarem e formatarem dados da web externos às plataformas, para que possam se encaixar na lógica subjacente da plataforma.

Ao propormos uma perspectiva técnico-material a respeito das plataformas, mostramos o "trabalho realizado pelas plataformas" não em um sentido retórico (cf. Gillespie, 2010), mas em um sentido computacional. A noção de plataformização foi introduzida como meio de criticar as consequências da programabilidade das plataformas. Essa foi uma primeira exploração nessa área, ao mostrar como as plataformas de redes sociais colocam em ação a sua programabilidade para redesenhar a web para redes sociais.

REFERÊNCIAS

Allen, M. (2013). What was Web 2.0? Versions as the dominant mode of internet history. *New Media & Society*, 15, 260-275. doi:10.1177/1461444812451567

Amazon. (2002, July 16). *Amazon.com launches web services* [Press Release]. Recuperado de http://phx.corporate-ir.net/phoenix.zhtml?c=176060&p=irol-newsArticle&ID=503034

Anderson, P. (2012). *Web 2.0 and beyond: Principles and technologies*. Boca Raton, FL: CRC Press.

Andreessen, M. (2007a, June 12). *Analyzing the Facebook Platform, three weeks in* [Blog post]. Recuperado de https://web.archive. org/web/20071002070223/http://blog.pmarca.com/2007/06/analyzing_the_f.html

Andreessen, M. (2007b, September 16). *The three kinds of platforms you meet on the Internet* [Blog post]. Recuperado de https://web.archive.org/web/20071002031605/http://blog.pmarca.com/2007/09/the-three-kinds.html

Arrington, M. (2008, September 15). *Facebook isn't a social network: And stop trying to make new friends there* [Blog post]. Recuperado de http://techcrunch.com/2008/09/15/facebookisnt-a-social-network-and-dont-try-to-make-new-friends-there/

Baldwin, C. Y., & Clark, K. B. (2000). *Design rules: The power of modularity*. Cambridge, MA: MIT Press.

Baldwin, C. Y., & Woodard, C. J. (2009). The architecture of platforms: A unified view. In A. Gawer (Ed.), *Platforms, markets and innovation* (pp. 19-44). Cheltenham, UK: Edward Elgar Publishing Limited.

Beer, D. (2009). Power through the algorithm? Participatory web cultures and the technological unconscious. *New Media & Society*, 11, 985-1002. doi:10.1177/1461444809336551

Beer, D., & Burrows, R. (2007). Sociology and, of and in Web 2.0: Some initial considerations. Sociological Research Online, 12(5). Recuperado de http://dx.doi.org/10.5153/sro.1560

Benslimane, D., Dustdar, S., & Sheth, A. (2008). Services mashups: The new generation of web applications. *IEEE Internet Computing*, 12(5), 13-15.

Birrell, A. D., & Nelson, B. J. (1984). Implementing remote procedure calls. ACM Transactions on Computer Systems, 2, 39-59. doi:10.1145/2080.357392

Bodle, R. (2011). Regimes of sharing. *Information, Communication & Society*, 14, 320-337. doi :10.1080/1369118X.2010.542825

Bogost, I., & Montfort, N. (2009). Platform studies: Frequently questioned answers. *Digital Arts and Culture*. Recuperado de http://escholarship.org/uc/item/01r0k9br.pdf

boyd, d. (2007, June 24). *Viewing American class divisions through Facebook and MySpace* [Blog post]. Recuperado de http://www.danah.org/papers/essays/ClassDivisions.html

boyd, d., & Ellison, N. (2008). Social network sites: Definition, history, and scholarship. *Journal of Computer-Mediated Communication*, 13, 210-230. doi:10.1111/j.1083-6101.2007.00393.x

Bruns, A. (2008). *Blogs, Wikipedia, second life, and beyond: From production to produsage*. New York, NY: Peter Lang.

Bucher, T. (2012a). *Programmed sociality: A software studies perspective on social networking sites* (Doctoral dissertation). University of Oslo, Oslo, Norway.

Bucher, T. (2012b). A technicity of attention: How software "makes sense." *Culture Machine*, 13, 1-13. Recuperado de http://www.culturemachine.net/index.php/cm/article/viewArticle/470

Bucher, T. (2013). Objects of intense feeling: The case of the Twitter API. *Computational Culture*, 3. Recuperado de http://computationalculture.net/article/objects-of-intense-feelingthe-case-of-the-twitter-api

Cheng, X., Dale, C., & Liu, J. (2008). Statistics and social network of YouTube videos. In *16th International Workshop on Quality of Service* (IWQoS) (pp. 229-238). New York, NY: IEEE. Recuperado de http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4539688

Chun, W. H. K. (2011). *Programmed visions: Software and memory*. Cambridge, MA: MIT Press. Company Info. (n.d.). Recuperado de http://newsroom.fb.com/company-info/

Cramer, F., & Fuller, M. (2008). Interface. In M. Fuller (Ed.), *Software studies: A lexicon* (pp. 149-152). Cambridge, MA: MIT Press.

DuVander, A. (2012, January 4). *Over 2,000 APIs added in 2011: Social, telephony, open government [Blog post]*. Recuperado de http://www.programmableweb.com/news/over-2000-apis-added-2011-social-telephony-open-government/2012/01/04

Evans, D. S., Hagiu, A., & Schmalensee, R. (2006). *Invisible engines: How software platforms drive innovation and transform industries*. Cambridge, MA: MIT Press.

Facebook Developers. (n.d.-a). *Canvas*. Recuperado de https://developers.facebook.com/docs/games/canvas/

Facebook Developers. (n.d.-b). *Facebook content sharing best practices*. Recuperado de https://developers.facebook.com/docs/sharing/best-practices

Facebook Developers. (n.d.-c). *A guide to sharing for webmasters*. Recuperado de https://developers.facebook.com/docs/sharing/Webmasters

Facebook Developers. (n.d.-d). *Open Graph overview*. Recuperado de https://developers.facebook.com/docs/opengraph/overview/

Facebook Developers. (n.d.-e). *Quickstart: Facebook SDK for JavaScript*. Recuperado de https://developers.facebook.com/docs/javascript/quickstart/v2.2

Facebook Help. (n.d.). *What are social plugins?* Recuperado de https://www.facebook.com/help/103828869708800

Facebook Newsroom. (2006, May 3). *Facebook expands to include work networks*. Recuperado de https://newsroom.fb.com/news/2006/05/facebook-expands-to-include-work-networks-2/

Facebook Newsroom. (2007, May 24). *Facebook unveils platform for developers of social applications*. Recuperado de https://newsroom.fb.com/news/2007/05/facebook-unveils-platformfor-developers-of-social-applications/

Fetterman, D. (2006, August 15). *Facebook development platform launches*. Recuperado de https://www.facebook.com/notes/2207512130

Gawer, A. (2014). Bridging differing perspectives on technological platforms: Toward an integrative framework. *Research Policy*, 43, 1239-1249. doi:10.1016/j.respol.2014.03.006

Gehl, R. (2010). *A cultural and political economy of Web 2.0* (Doctoral dissertation). George Mason University, Fairfax, VA.

Gehl, R. (2012). Real (software) abstractions on the rise of Facebook and the fall of MySpace. *Social Text*, 30, 99-119.

Geminder, K. (2007, June 2). Platform is here. Recuperado de https://www.facebook.com/notes/facebook/platform-ishere/2437282130

Gerlitz, C., & Helmond, A. (2013). The Like economy: Social buttons and the data-intensive web. *New Media & Society*, 15, 1348-1365. doi:10.1177/1461444812472322

Gillespie, T. (2010). The politics of "platforms." *New Media & Society*, 12, 347-364. doi:10.1177/1461444809342738

Gillespie, T. (2014). The relevance of algorithms. In P. Boczkowski, K. Foot, & T. Gillespie (Eds.), *Media technologies* (pp. 167-194). Cambridge, MA: MIT Press.

Hagiu, A. (2014). Strategic decisions for multisided platforms. MIT Sloan Management Review, 55, 71-80.

Hands, J. (2013). Introduction: Politics, power and "platformativity." *Culture Machine*, 14. Retrieved from http://www.culturemachine.net/index.php/cm/article/viewArticle/504

Helmond, A. (2013). The algorithmization of the hyperlink. *Computational Culture*, issue 3. Recuperado de http://computationalculture.net/article/the-algorithmization-of-the-hyperlink

Hicks, M. (2010, April 21). *Building the social web together*. Recuperado de https://www.facebook.com/notes/facebook/building-the-social-web-together/383404517130

Jenkins, H. (2006). *Convergence culture: Where old and new media collide*. New York: New York University press.

Kirkpatrick, D. (2010). *The Facebook effect: The real inside story of Mark Zuckerberg and the world's fastest growing company*. New York, NY: Random House.

Lane, K. (2012, December 20). History of APIs. Recuperado de http://history.apievangelist.com/

Lane, K. (n.d.). *About Kin Lane*. Recuperado de http://kinlane.com/about/

Langlois, G., & Elmer, G. (2013). The research politics of social media platforms. *Culture Machine*, 14, 1-17. Recuperado de http://www.culturemachine.net/index.php/cm/article/view-Article/505

Langlois, G., Elmer, G., McKelvey, F., & Devereaux, Z. (2009). Networked publics: The double articulation of code and politics on Facebook. *Canadian Journal of Communication*, 34(3). Recuperado de http://www.cjc-online.ca/index.php/journal/article/viewArticle/2114

Langlois, G., McKelvey, F., Elmer, G., & Werbin, K. (2009). Mapping commercial Web 2.0 worlds: Towards a new critical ontogenesis. *Fibreculture*, issue 14. Recuperado de http://fourteen.fibreculturejournal.org/fcj-095-mapping-commercialweb-2-0-worlds-towards-a-new-critical-ontogenesis/

Laurent, S. S., Johnston, J., Dumbill, E., & Winer, D. (2001). *Programming web services with XML-RPC*. Sebastopol, CA: O'Reilly Media, Inc.

Liu, A. (2004). Transcendental data: Toward a cultural history and aesthetics of the new encoded discourse. *Critical Inquiry*, 31, 49-84. doi:10.1086/427302/

Liu, A. (2008). *Local transcendence: Essays on postmodern historicism and the database*. Chicago, IL: University of Chicago Press.

Liu, D. (2015, March 25). *F8 2015: New ways to connect with the Facebook family of apps*. Recuperado de https://newsroom. fb.com/news/2015/03/f8-day-one-2015/

Locke, L. (2007). *The future of Facebook*. Recuperado de http://content.time.com/time/business/article/0,8599,1644040,00.Html

Lomborg, S., & Bechmann, A. (2014). Using APIs for data collection on social media. *The Information Society*, 30, 256-265. doi: 10.1080/01972243.2014.915276

Madden, M., & Fox, S. (2006). Riding the waves of "Web 2.0." *Pew Internet and American Life Project*, 5. Recuperado de http://www.pewinternet.org/files/old-media/Files/Reports/2006/PIP_Web_2.0.pdf.pdf

Manovich, L. (2001). *The language of new media*. Cambridge, MA: MIT Press.

McKelvey, F. (2011). A programmable platform? Drupal, modularity, and the future of the web. *Fibreculture*, 18. Recuperado de http://eighteen.fibreculturejournal.org/2011/10/09/fcj-128-programmable-platform-drupal-modularity-and-thefuture-of-the-web/

Montfort, N., & Bogost, I. (2009). *Racing the beam: The Atari video computer system*. *Cambridge*, MA: MIT Press. Understanding Web 2.0. *IT Professional*, 9(4), 34-41. doi:10.1109/MITP.2007.78. Murugesan, S. (2007).

Myllymaki, J. (2002). Effective web data extraction with standard XML technologies. *Computer Networks*, 39, 635-644.

O'Reilly, T. (2005). *What is Web 2.0: Design patterns and business models for the next generation of software*. Recuperado de http://oreilly.com/web2/archive/what-is-web-20.html

O'Reilly, T., & Battelle, J. (2004). *Opening welcome: The state of the internet industry*. Presented at the Web 2.0 Conference, Hotel Nikko, San Francisco, CA. Retrieved from http://web-2con.com/presentations/web2con/intro_tim_john.ppt

Puschmann, C., & Burgess, J. (2013). *The politics of Twitter data* (HIIG Discussion Paper Series No. 2013-01). Recuperado de http://papers.ssrn.com/abstract=2206225

Renzi, A. (2011). What is the politics of platform politics? *Television & New Media*, 12, 483-485.

Richardson, L., & Ruby, S. (2008). *RESTful web services*. Sebastopol, CA: O'Reilly Media, Inc.

Rieder, B., & Sire, G. (2014). Conflicts of interest and incentives to bias: A microeconomic critique of Google's tangled position on the Web. *New Media & Society*, 16, 195-211. doi:10.1177/1461444813481195

Rochet, J.-C., & Tirole, J. (2003). Platform competition in twosided markets. *Journal of the European Economic Association*, 1, 990-1029. doi:10.1162/154247603322493212

Rogers, R. (2013). Digital methods. Cambridge, MA: MIT Press.

Stevenson, M. (2014). Rethinking the participatory web: A history of HotWired's "new publishing paradigm," 1994-1997. *New Media & Society*. Advance online publication. doi:10.1177/1461444814555950

Tokuda, L. (2009, June). *Social applications: Growth, use, and monetization*. Presented at the Global ICT Summit 2009, Tokyo, Japan. Recuperado de http://www.ict-summit.jp/2009/pdf/report6_tokuda.pdf

The top 500 sites on the web. (2015, June 14). Recuperado de http://www.alexa.com/topsites

Ullrich, C., Borau, K., Luo, H., Tan, X., Shen, L., & Shen, R. (2008). Why web 2.0 is good for learning and for research: Principles and prototypes. In *Proceedings of the 17th International Conference on World Wide Web* (pp. 705-714). New York, NY: ACM. Recuperado de http://dx.doi.org/10.1145/1367497.1367593

Van Dijck, J. (2013). *The culture of connectivity: A critical history of social media*. New York, NY: Oxford University Press.

Vis, F. (2013). A critical reflection on Big Data: Considering APIs, researchers and tools as data makers. *First Monday*, 18(10). Recuperado de http://firstmonday.org/ojs/index.php/fm/article/view/4878

W3C. (n.d.). *XML Essentials*. Recuperado de http://www.w3.org/standards/xml/core

Willmott, S., & Balas, G. (2013) Winning in the API economy. Recuperado de http://www.3scale.net/wp-content/uploads/2013/10/Winning-in-the-API-Economy-eBook--3scale.pdf

Winer, D. (1995, August 22). *What is a platform?* Recuperado de http://scripting.com/davenet/1995/08/22/whatisaplatform.html

YouTube. (2005, August 21). *August 2005*. Recuperado de http://youtube-global.blogspot.nl/2005_08_01_archive.html

*Tradução por Tiago Salgado.