

## Standards Tecnológicos, *Open Source* e Espaço Público

António Machuco Rosa\*

Universidade Lusófona de Humanidades e Tecnologias

### Sumário:

Nesta comunicação procuraremos identificar a presença dos mecanismos de ordem e desordem (tal como foram tematizados pela teoria da informação) no desenvolvimento histórico do *software*. Analisaremos o movimento de privatização de Unix, e de seguida mostraremos as consequências económicas que decorrem da produção de *software em regime de open source*.

### Os primórdios do *software*

O desenvolvimento histórico, e progressiva comercialização, de *software* ilustra com particular nitidez a forma como os sistemas em geral oscilam em torno de duas grandes fases, as fases de ordem e desordem. Ele permite igualmente reapreciar a velha oposição entre espaço público e espaço privado. Para o compreendermos, recorde-se rapidamente como se processava a produção e distribuição de *software* nas décadas iniciais da computação, as décadas de cinquenta e sessenta (cf. Ceruzzi, 2003). Não existia na época algo que realmente se pudesse chamar um mercado comercial de *software*. O negócio da indústria informática concentrava-se então no fabrico de *hardware*, e os fabricantes produziam o (reduzido) *software* necessário ao seu funcionamento sem que este possuísse um estatuto comercial específico e autónomo. Na realidade, e literalmente, apenas existia *hardware*, visto o *software* ser pensado como uma sua parte integrante. Um programa estava completamente dedicado a uma certa máquina e não era executado nos computadores de outros fabricantes. Mesmo que isso não fosse um objectivo perseguido explicitamente pelos diversos fabricantes (IBM, Honeywell, DEC, etc.), as diversas máquinas eram incompatíveis entre si pelo que nem surgia a ideia de

---

\* Este artigo foi elaborado no quadro do projecto de investigação *Trends on Portuguese Networks Culture*, projecto financiado pela FCT/POCTI/34436.

comercializar um programa à partida confinado à máquina para que tinha sido concebido.

Durante o início da década de setenta existiram acontecimentos decisivos para a evolução da futura da informática. Em particular, foi desenvolvida uma linguagem de programação, a linguagem C, linguagem de alto nível aspirando à universalidade, isto é, independente das especificidades do *hardware*. Foi nela que foi programado o sistema operativo UNIX, o qual se tornou por esse facto um sistema operativo passível de ser executado em *hardware* com arquitecturas e especificações técnicas completamente distintas. Podemos afirmar ser esse o momento em que o *software* se tornou verdadeiramente *autónomo*.

Essa autonomia não significou, contudo, privatização e comercialização. O sistema UNIX era redistribuído acompanhado do respectivo código-fonte, e também acompanhado de uma licença de utilização que permitia modificar e redistribuir o código-fonte acabado de modificar. O UNIX não era comercializado. Essa espécie de espaço público de troca existente na sombra do espaço mercantil da venda e produção de mais-valia vai progressivamente desaparecer. A comercialização vai ocorrer a partir dos finais da década de setenta e, independentemente dos motivos específicos que levaram diversas empresas a inaugurar esse movimento de privatização, devemos procurar identificar um grande tema que o enquadra teoricamente e que, em última análise, revela a sua natureza inteiramente *sui generis*.

### **A trajectória de UNIX**

O tema é ilustrado pela trajectória evolutiva de UNIX (cf. Ritchie, 1979, McKusick, 1999), a qual, no entanto, mais não é que um exemplo de uma dinâmica perfeitamente geral (cf. Machuco Rosa, 2004, para outros exemplos). Ela encontra-se no âmago de uma característica distintiva das tecnologias da informação e da comunicação: o papel fundamental que nelas desempenham os *standards*-plataformas. Antes de passarmos ao caso específico de Unix, devemos referir as linhas essenciais da dinâmica genérica.

A adopção espontânea de *standards* consiste num percurso evolutivo que parte de uma situação de fragmentação ou competição (fase de desordem) até que um conjunto de

factores acaba por induzir a imposição de um certo *standard* dominante e monopolista (fase final de ordem). Na fase de desordem existe uma intensa competição na qual cada candidato a *standard* é independente dos outros (fase de incompatibilidade dos *standards*). Nessa fase, cada *standard* é o *standard*; cada *standard* proposto auto-designa-se enquanto tal, invocando-se para isso as mais diversas razões, em particular razões ligadas à qualidade intrínseca das suas especificações técnicas. Contudo, na realidade, é *standard* aquilo que os outros actores designam enquanto tal, isto é, que recebe a aprovação – a adopção – dos outros. Um *standard* não passa do particular fragmentado à aprovação universal devido a supostas ou reais qualidades técnicas intrínsecas objectivas (cf. Arthur, 1994), mas sim através de um mecanismo de retroacção positiva que se auto-reforça a si mesmo: quando mais antigo, e quanto maior a base instalada, maior a probabilidade de essa base crescer e assim receber uma adesão cada vez mais universal. Cada actor imita os outros, e assim tende a reforçar o poder da base instalada já existente, indicando por sua vez a ulteriores actores qual é efectivamente o *standard*, até que se dá uma transição crítica que determina que todos convergem no mesmo sentido.

No caso de UNIX, já referimos que ele começou a ser desenvolvido no início da década de 70, na Universidade de Berkeley graças aos trabalhos de Ken Thompson e Dennis Ritchie. e num espírito de partilha comunitária de código sem qualquer motivação de natureza comercial (cf. Ritchie, 1979). Por volta de 1977, o lendário programador Bill Joy desenvolveu uma versão de UNIX distribuída nos termos da *Berkeley Software Distribution* (BSD), que é uma das primeiras licenças elaboradas no espírito daquilo que mais tarde viria a ser designado por *open source*: essa licença permite a qualquer utilizador de um programa fazer praticamente tudo o que quiser com ele – nomeadamente, copiá-lo, redistribuí-lo e modificá-lo, apenas se exigindo uma referência à Universidade que o originou. É também por volta dessa altura que a agência que geria a Arpanet, rede precursora da actual Internet, começa a mostrar interesse no UNIX. As razões eram os custos decorrentes da necessidade de substituir um grande número de computadores tornados obsoletos e os igualmente elevados custos de instalação de *software* em máquinas de diversos fabricantes com especificações de *hardware* incompatíveis entre si. Perante esta situação de desordem, a Arpa teve a ideia de unificar ao nível do sistema operativo (McKusick, 1999), criando assim uma

plataforma comum externa e objectiva que permite a comunicação. E a escolha de UNIX como força unificadora das operações de computação em rede foi em larga medida determinada pelo facto de o código de Unix se encontrar livremente acessível (cf. Leonard, 2000).

Aparentemente, encontravam-se assim reunidas as condições para que Unix se tornasse um espécie de *standard* ao nível dos sistemas operativos, isto é, que, cumprindo os objectivos que levaram à sua criação, gerasse a convergência para a ordem. No entanto, esse esforço e pressão unificadora não foi suficiente. A história de UNIX reflecte perfeitamente a emergência da fragmentação enquanto força oposta, mas desenvolvendo-se através de um mecanismo similar, àquela que conduz à convergência unânime. De facto, a partir dos anos oitenta, começaram a surgir *múltiplas* versões de UNIX com código fechado

Aquilo que era inicialmente um espaço público, mesmo que ainda não tenha acedido à universalidade completa da standardização, começa a desaparecer segundo uma dinâmica capitalista de privatização. É uma dinâmica típica em tecnologias da informação e da comunicação, na qual a quase totalidade dos diversos actores competindo pela plataforma são todos eles imediatamente animados pela orientação geral da própria dinâmica.

Assim, em 1983, já existiam seis sistemas operativos UNIX (com código-fonte fechado) destinados a seis tipos diferentes de computadores, sistemas que tendiam a diferir cada vez mais entre si. Inicia-se de seguida uma intensa competição designada pelas ‘primeiras guerras do UNIX’ (cf. Raymond, 2003; Ceruzzi, 2003). Ela formou-se em torno da rivalidade gémea intensa entre o UNIX da Sun Microsystems e a versão da AT&T. Um movimento inverso surge pouco depois com tentativas de conciliar essas duas versões em torno de *standards* comuns e criação de API’s (*Application Programming Interface*). Esse esforço levou à formação de uma aliança entre a AT&T e a Sun (*Unix International*), que acordaram numa mesma plataforma. Mas as forças da fragmentação em sistemas que funcionam como plataformas comuns aspirando à universalidade são demasiado fortes, pelo que se formou uma outra coligação oposta à *Unix International*, constituída pela IBM, DEC, Hewlett-Packard, entre outros, e chamada *Open Software Foundation* (OSF). Mesmo entre os membros de cada uma das duas alianças começaram a ser produzidas versões incompatíveis de UNIX, e quando

mais rivais entre si eram cada par de empresas maior era a sua propensão para alinhar nos dois campos gémeos rivais. Em resumo, a fragmentação não deixou de se acelerar até às actualmente mais de 30 versões existentes de UNIX.

Avaliaremos adiante a possibilidade de superar a fragmentação dentro do espírito inicial de UNIX. De momento, realce-se que inicialmente pequenas diferenças nas versões do sistema operativo se foram progressivamente amplificando e acabaram por originar versões inteiramente incompatíveis, incompatibilidades dificilmente superáveis devido ao código-fonte se encontrar fechado. Está-lhe subjacente a retroacção positiva, que aqui se traduz numa *força imitativa rival*. Essa força pode levar à adesão a um único *standard*. Mas é a mesma força que leva a cabo um tipo de unanimidade oposto, e que se traduz por cada um se procurar afastar de cada outro. No caso de UNIX, cada fabricante não tinha qualquer interesse em abrir o código, determinando que o *software* apenas fosse compatível com o seu tipo de *hardware*, e assim cada um visava passar do particular ao universal da estandardização procurando capturar a totalidade do mercado. Todos tinham como objectivo dominar a plataforma *standard* dos sistemas operativos. A sua rivalidade pela posse do objecto desejado vai obrigar a que eles rivalizem directamente entre si: todos rivalizam na produção de incompatibilidades. Imitando-se, gerando as incompatibilidades que cada rival gémeo também gerava, os fabricantes fugiram todos eles em direcções independentes e dessa violência competitiva antagónica resultou a inevitável fragmentação.

## Open Source

Seria possível mostrar em detalhe (cf. Machuco Rosa, 2004) que a mesma força que é responsável pela fragmentação ou desordem pode igualmente ser uma força de convergência ou ordem no caso do *software* proprietário. O exemplo paradigmático dessa força inversa seria fornecidos pelo sistema operativo monopolista da Microsoft, alicerçado a partir da base instalada do Ms-Dos.

No entanto, podemos também ver a emergência da ordem num movimento que surge como contraponto à privatização do *software*, seja através da disponibilização pública de *software* anteriormente existente em regime proprietário (veja-se o caso do *browser* Firefox, descendente do Netscape), seja impedindo a privatização de *software* não-

proprietário. Em nossa opinião, os aspectos essenciais a destacar a propósito do chamado movimento *open source* incidem sobre dinâmicas económicas e sobre mecanismos de regulação social e política, sendo essencial constatar que esses dois aspectos se encontram estreitamente interligados. Note-se que essa interligação também ocorre com *software* proprietário; de facto, é um traço distintivo das novas tecnologias que a ligação entre aspectos económicos e mecanismos de regulação política seja potencialmente muito mais marcada que nos sectores económicos tradicionais.

Recordando-se, uma vez mais, que a prática de *software* livre era usual nas primeiras décadas da computação, o facto é que o chamado movimento *open source* apenas se viu verdadeiramente institucionalizado com a criação, por Richard Stallman, em meados dos anos oitenta da *Free Software Foundation* e, sobretudo, com a criação de um tipo particular de licenciamento de *software*, a *General Public Licence* (GPL). Seguindo as indicações de Stallman, ao falarmos de *software* livre devemos ter sobretudo em vista o *software* licenciado sob GPL, e não *software open source* em sentido mais geral. Nesse sentido mais geral, *software open source* estipula que o *software* pode ser distribuído e modificado livremente, sem que no entanto se impeça que o código assim disponibilizado possa vir a ser privatizado ao serem fechadas as modificações realizadas em linhas de código-fonte anteriormente abertas. O carácter distintivo de GPL consiste precisamente em essa licença ter como efeito obrigar que o código modificado seja novamente redistribuído de forma aberta. A garantia última dessa liberdade de acesso, partilha e distribuição é uma exterioridade, a própria lei do *copyright*, pois se alguém não aceitar os termos de GPL, então a lei geral do *copyright* aplica-se por *default*. Trata-se de um engenhoso mecanismo (*copyleft*) que utiliza a lei do *copyright* para garantir que o *software* permaneça *livre*, permaneça *neutral* e ao abrigo das rivalidades concorrenciais directas. Pelo contrário, o regime proprietário é ‘um sistema baseado na divisão do público e em manter os utilizadores sem ajuda’

Como Stallman constantemente refere, o *software* proprietário ergue ‘barreiras’, ‘sebas’, ‘paredes’, e conduz a um combate directo entre a totalidade dos actores - utilizadores e empresas - , gravitando em torno da indústria das novas tecnologias (Stallman, 2001). As suas posições têm sido (por vezes intencionalmente) constantemente mal-interpretados, fazendo de Stallman uma espécie de adepto da socialização integral dos mercados enquanto, por contraposição, as grandes empresas, como a Microsoft, seriam

o representante típico do mercado livre. Isto é o mundo completamente de pernas para o ar! Vejamos um pouco mais em detalhe.

Ao se falar em ‘paredes’ dividindo o público, tem-se em vista o facto de ser do interesse estratégico das empresas proprietárias a produção de incompatibilidades entre programas, sobretudo daqueles que de algum modo funcionam como plataformas (os programas mais divulgados entram praticamente todos nesta categoria). Se esse é o interesse de certas empresas, seguramente que essa não é a esfera do interesse público, podendo alguns argumentar ser *moralmente inaceitável* que se utilizem estratégias assentes no secretismo dos formatos proprietários e nas leis do *copyright* e das patentes com o fim de se impedir que indivíduos e programas interajam entre si, para obstar que alguém possa modificar um programa de forma a satisfazer uma sua necessidade, ou ainda que seja forçados a reescrever por completo um programa já existente para realizar uma certa função, tendo assim de se ‘reinventar a roda’ (e é sabido que uma das razões que levavam a que o código dos programas fosse inicialmente livremente partilhado se prendia precisamente com este último aspecto, cf. Ceruzzi, 2003). Não é necessário proceder a grandes análises para intuitivamente se compreender quão nocivas do ponto de vista do aumento do bem-estar económico da sociedade são tais práticas (cf. Arrow, 1962, para as consequências socio-económicas dos regimes de propriedade intelectual). Tão pouco são precisos excessivos conhecimentos de economia para se compreender como os monopólios distorcem completamente os preços que se deveriam verificar num mercado onde existisse concorrência aberta. A enésima unidade do Windows tem um custo marginal nulo; esse deveria tendencialmente o seu preço, o que obviamente não ocorre devido à sua posição de monopólio *de facto*. A produção de incompatibilidade, o uso constante da ameaça judicial com base no *portfolio* de propriedade intelectual seguramente não é o traço característico de empresas operando num mercado concorrencial previamente constituído. Seguramente que empresas como a Microsoft não operam no ambiente descrito pela teoria neoclássica da economia: mercados livremente concorrenciais em que a informação é completa e em que os preços tendem a ser iguais ao custo marginal da enésima unidade produzida.

Pelo contrário, e ironicamente, é o *software* em regime GPL que mais estritamente funciona segundo as dinâmicas económicas do que se convencionou designar por mercados livres. A generalidade dos defensores de *software* livre (ou *open source*)

sublinha que, em primeiro lugar, as plataformas-núcleo devem permanecer livres e neutrais. Essa neutralidade possui um duplo aspecto. Em primeiro lugar, demonstra-se formalmente (Economides, 2004) que um *standard* como um sistema operativo em regime livre conduz a um aumento do bem-estar social bastante superior ao que é produzido por um sistema operativo proprietário. Se é indiscutível que a existência de um *standard* (superação da fragmentação) é economicamente benéfica em termos sociais, já começa a ser clara a conclusão para que avançamos, e que consiste em afirmar que os *standards* de *software* não têm que ser necessariamente privados. Em segundo lugar, a neutralidade das plataformas abertas torna-as uma instância geral do princípio geral de *design end-to-end* dos sistemas. Quer isso dizer que a plataforma é neutral, não discrimina, não constrange as aplicações que sobre ela correm. Insistimos em múltiplas ocasiões sobre a importância que um tal princípio de *design* teve no desenvolvimento da Internet (cf. Machuco Rosa, 2003): ao não constranger a natureza das aplicações que se viram a desenvolver nas suas pontas, um *standard end-to-end*, ‘estúpido’, indiferente ao conteúdo que sobre ele corre, como o TCP/IP garante a enorme inovação que a Internet tem vindo a exhibir. Do mesmo modo, um sistema operativo aberto como Linux (cf. mais abaixo) é *end-to-end*, pois ele em nada constrange a inovação que se pode desenvolver nas suas ‘pontas’ (Weber, 2004). Evidentemente que nada garante que essa seja a situação quando uma empresa monopoliza a plataforma.

No fundo, essa dupla face (o que não significa que outras faces não existam) da neutralidade dos *standards*-plataformas revela qual é a real dinâmica económica do *software* livre. O combate, a rivalidade mimética antagónica, não se deve exercer *pelo* (controlo) do *standard* ou plataforma; a concorrência (aberta) deve processar-se em *cima* da plataforma. Os programas que correm em cima da plataforma-*standard* poderão concorrer entre si desencadeando a competição livre no mercado que a teoria neoclássica da economia prevê e que irá levar a uma convergência do preço de venda para o custo marginal do produto. Mais, são igualmente claras as oportunidades que programas com código aberto proporcionam ao se incentivar que as empresas se especializem na parametrização do programa de acordo com as necessidades de cada cliente. Em suma, com *software* livre, ao invés do que ocorre com *software* privado, os programas passariam a ser uma matéria-prima do tipo tradicional (O’Reilly, 1999), a



competição dar-se-ia *no* mercado e não *pelo* mercado (Shapiro, 2001), e portanto teríamos as dinâmicas económicas dos bens tradicionais em livre concorrência.

Como já se observou, nas tecnologias da informação, os aspectos económicos estão indissolúvelmente ligados à regulação social do espaço público. O *software* livre forma um espaço público (não confundir com domínio público). Por outro lado, sublinhámos que a imposição de um *standard* como o sistema operativo Windows foi um poderoso factor de convergência que permitiu em larga medida aplacar as poderosas forças fragmentadoras existentes nas tecnologias da informação. No entanto, não se deve perder de vista que o S.O. Windows não é um *espaço público*, no sentido pleno desta última expressão. Um espaço público de *software* é, *simultaneamente*, uma plataforma monopolista e uma plataforma cujo código-fonte está aberto. Se bem que as tecnologias da informação induzam necessariamente monopólios, não se segue que estes tenham de ser *privados*. Não é necessário ser um adepto da socialização integral dos mercados para se ter essa posição. Os *standards* podem, e tendencialmente devem, ser públicos.

As forças que conduzem à ordem são as mesmas que podem reconduzir à desordem, pelo que é esta é um perigo bem real. No decorrer do processo que lhe foi recentemente movido pelo governo dos Estados-Unidos, a Microsoft argumentou contra a sua cisão em diversas empresas precisamente devido a esses perigos. Mais, a empresa de Seattle tem vindo ultimamente a conduzir a sua ofensiva contra o *software* livre sustentando que ele encerra em si mesmo poderosas tendências fragmentadoras devido à possibilidade de versões distintas de um mesmo programa de *software* livre poderem acabar por divergir radicalmente gerando incompatibilidades (fenómeno dito de *forking*). Mas poderá argumentar-se que tais posições estão teoricamente e factualmente erradas por os grandes projectos GPL, como o sistema operativo Linux, já terem uma massa crítica de aderentes suficientemente grande para o que o *forking* seja desincentivado (devido exactamente à mesma força que levou um *standard* privado a impor-se), e por existirem indivíduos que autenticam a versão oficial (caso de Linus Torvalds no caso de Linux). Pode mesmo sustentar-se que não só a fragmentação pode não ocorrer nesses sistemas como, inversamente, as forças da ordem são mais fortes em programas *open source* que em programas proprietários (Young, 1999, pp. 123-24).

De facto, no desenvolvimento de programas semelhantes, programas, pequenas variações iniciais tendem a amplificar-se com o decorrer do tempo. Essa amplificação

reforça-se num regime proprietário, no qual existe a necessidade em satisfazer em exclusivo certos clientes, donde a variação ser mantida fechada e secreta. Emerge então a desordem ou fragmentação, marcada por uma intensa concorrência que conduz os diferentes vendedores a produzir versões incompatíveis dos seus produtos em vista a monopolizar o mercado. Todos eles se imitam e, eventualmente, a unanimidade da ordem cristaliza-se em torno de um único *standard*. Essa mesma força imitativa está presente na possibilidade da emergência da ordem fundada em *standards* públicos. Neste caso, a emergência da ordem está associada a ‘um ponto de referência comum’, o código-fonte aberto, o qual permite que todos introduzam uma mesma inovação que se torne popular no mercado, tendendo-se assim a harmonizar as eventuais incompatibilidades existentes. Portanto, conclusão chave, os perigos de fragmentação podem mesmo ser *menores* em código aberto que em código fechado, isto é, o espaço comum e público de GPL é uma poderosa força de ordem ou convergência. Esse tipo de licença não apenas garante certas liberdades fundamentais dos indivíduos mas revela ser a única verdadeiramente compatível com o ideal económico subjacente ao mercados livre que tem caracterizado a modernidade.

### Referências bibliográficas

**Arrow**, K., ‘Economic Welfare and the Allocation of Resources for Invention’, in *The Rate and Direction of Economic Activity: Economic and Social Factors* (E. Nelson, ed.), Princeton University Press, Princeton, 1962.

**Arthur**, W. B., *Increasing returns and Path dependence in the Economy*, University of Michigan Press, Ann Arbour, 1994.

**Ceruzzi**, P., *A History of Modern Computing*, The Mit Press, Cambridge, 2003.

**Economides**, N., ‘Two-sided competition of proprietary vs. open source technology platforms and the implications for the software industry’, 2004, disponível em: <http://www.stern.nyu.edu/networks>

**Leonard**, A., *BSD Unix: Power to the People, from the Code*, 2000, disponível em : [http://dir.salon.com/tech/fsp/2000/05/16/chapter\\_2\\_part\\_one/index.html](http://dir.salon.com/tech/fsp/2000/05/16/chapter_2_part_one/index.html).

**Machuco Rosa**, A., ‘Emergence and Fragmentation of *Standards* in Information Technologies’, in *Interactive Television : contents, applications and challenges*, M.J. Damásio (ed.) ULHT, Lisboa, 2004, pp. 181-197.

**Machuco Rosa**, A., *Internet – Uma História*, 2ª edição, revista, corrigida e ampliada, Edições Universitárias Lusófonas, Lisboa, 2003.

**McKusick**, M., ‘Twenty Years of Berkely Unix: from AT&T-Owned to Freely Redistribuable’, in DiBona, C., Ockman, S., Stone, M., (eds.), *Open Sources. Voices from the Open Source Revolution*, O’Reilly & Associates, 1999, pp. 31-46.

**O’Reilly**, T., ‘Hardware, Software and Infoware’, in, *Open Sources. Voices from the Open Source Revolution*, DiBona, C., Ockman, S., Stone, M., (eds.), O’Reilly & Associates, 1999, pp. 189-196.

**Raymond**, E., *The Art of Programming Unix*, Perseus, Boston, 2003.

**Ritchie**, D., *The Evolution of the Unix Time-Sharing System*, 1979, disponível em :<http://cm.bell-labs.com/cm/cs/who/dmr/hist.html>, 2

**Shapiro**, C., ‘Setting Compatibility Standards Cooperation or Collusion?’ in Expanding the Bounds of Intellectual Property, Rochelle Dreyfuss, Diane Zimmerman e Harry First, (eds.), Oxford University Press, Oxford, 2001.

**Stallman**, R., *Free Software, Free Society: Selected essays of Richard M. Stallman*, GNU Press, Boston, 2001.

**Weber**, S., *The Success of Open Source*, Harvard University Press, Cambridge, 2004

**Young**, R., ‘Giving it Away: How Red Hat Software Across a New Economic Model and Helped Improve an Industry’ in, *Open Sources. Voices from the Open Source Revolution*, DiBona, C., Ockman, S., Stone, M., (eds.), O’Reilly & Associates, 1999, pp. 133-126.