

Danilo Braga *

Design hacker e as customizações de interface gráfica



Egresso Mestrado | 2018

Danilo Braga é Mestre em Design pela Universidade Anhembi Morumbi (UAM), pesquisando padrões de design hacker, visualização de dados, linguagens e interfaces móveis. Atualmente trabalhando como designer sênior na Amadeus IT Group (Berlim, Alemanha).

<danilo@danilobraga.com>

ORCID: 0000-0001-6151-9584

Resumo Compilado e revisita de descobertas da pesquisa de mesmo nome, este artigo se divide em três pilares: no primeiro, é realizado um exercício livremente inspirado na arqueologia de interface, com foco na customização na história da computação. Investigamos como as opções de customização se consolidam em sistemas operacionais *desktop* para Macintosh. O segundo pilar é dedicado à definição e exploração da figura do *hacker*, seu surgimento, atuação, grupos, ética e classes. Por fim, exploramos as técnicas do *jailbreak*, na qual *hackers* aplicam técnicas e ferramentas que buscam modificar a estrutura do sistema operacional iOS e as customizações de interface desbloqueadas por essa prática.

Palavras chave Design de interface, customização de interfaces, *hacker*, *jailbreak*, iOS.

Hacker design and graphic user interface customizations

Abstract *Compiling and reviewing the findings of the research of the same name, this article is founded from three pillars: in the first, a freely inspired by interface archeology exercise, with a focus on customization in the history of personal computing. We investigate how customization options are consolidated in desktop operating systems for Macintosh. The second pillar is dedicated to the definition and exploration of the figure of the hacker, his appearance, performance, groups, ethics and classes. Finally, we explore the techniques of jailbreak, where hackers apply techniques and tools that seek to modify the structure of the iOS operating system and the interface customizations unlocked by the practice*

Keywords *Interface design, interface customization, hacker, jailbreak, iOS.*

Design hacker y personalizaciones de interfaz gráfica

Resumen *Recopilado y revisita los hallazgos de la investigación del mismo nombre, se divide en tres pilares: en el primero, un ejercicio está inspirado libremente en la arqueología de la interfaz, con un enfoque en la personalización en la historia de la informática. Investigamos cómo se consolidan las opciones de personalización en los sistemas operativos de escritorio para Macintosh. El segundo pilar está dedicado a la definición y exploración de la figura del hacker, su apariencia, desempeño, grupos, ética y clases. Finalmente, exploramos las técnicas de jailbreak, donde los hackers aplican técnicas y herramientas que buscan modificar la estructura del sistema operativo iOS y las personalizaciones de la interfaz desbloqueadas por la práctica.*

Palabras clave *Diseño de interfaz, personalización de interfaz, hacker, jailbreak, iOS.*

Introdução

A popularização dos dispositivos móveis, em especiais os smartphones, vem afetando de maneira direta a relação da tecnologia com as pessoas que tem acesso aos equipamentos e serviços atrelados à conectividade portátil. Desde o início da década de 2010, podemos notar a proliferação pelos centros urbanos e periferias desses pequenos aparelhos cujo design físico se caracteriza por um ou dois botões e uma tela frontal ocupando quase a totalidade do produto. Seus usuários são fiéis: em média, checam o dispositivo 74 vezes por dia. E entre a parcela mais jovem, de 18 a 24 anos, a frequência aumenta: são 82 vezes, segundo dados globais da agência estadunidense ComScore. O smartphone passou a ser, para boa parte dos usuários, o primeiro e último contato com qualquer artefato durante um dia: 87% dos entrevistados em um levantamento realizado pela consultoria Deloitte checa o aparelho como primeira atitude ao acordar, e 81% tem a interação com o dispositivo como última ação ao ir dormir. Esses dados indicam uma tendência de crescimento da proximidade entre o smartphone e seus usuários: Desde 2013, o uso global de mídia móvel cresceu mais de 111%. Por aqui, a curva é ainda mais acentuada: o número aumenta para 265% em países latino-americanos e no México. Com um número previsto de 3.8 bilhões de usuários ao redor do mundo em 2021, o modelo físico – *form factor*, no jargão da indústria – do modelo vigente de smartphone privilegia a interação pelo toque das mãos na tela do dispositivo. Com isso, a interface gráfica se torna a protagonista de uma verdadeira revolução na usabilidade e no acesso à informação e comunicação.

Dada a identificação de uma tendência de aumento quantitativo (em número de dispositivos) e qualitativo (em proximidade na relação homem-máquina) na penetração dessa tecnologia no cotidiano de seus usuários, o estudo do entendimento e da evolução das interfaces gráficas se tornou uma prioridade como pesquisa de design gráfico e digital, refletido na indústria que vem cada mais valorizando os processos e metodologias do design para uma boa experiência em serviços e produtos, fazendo contratações e reorganizando profissionais de diversas disciplinas nos setores de *user experience design*, ou UX.

Esse movimento de mercado, entretanto, traz uma característica de definir e limitar a jornada de uso dos usuários em padrões cognitivos e de representação que carregam uma bagagem histórica de popularização das interfaces gráficas e das metáforas visuais legados dos primeiros sistemas operacionais computacionais baseados em desktop. Com isso, surge um fenômeno contrário à pasteurização da interação com esses dispositivos por designers que, por meio de técnicas *hacker* de quebra de códigos e travas de segurança nos softwares, customizam as interfaces gráficas e disponibilizam para a comunidade seus kits de customização para dar ao usuário a liberdade de escolher alterar a maneira como interagir com seus smartphones e a aparência visual dos elementos que compõem a interface gráfica.

O presente texto, recorte da pesquisa homônima de mestrado, é estruturado a partir de três questões principais: na primeira delas, fazemos um levantamento arqueológico das possibilidades de customização presentes nos sistemas operacionais dos últimos 25 anos, discutindo os casos e movimentos mais importantes desse aspecto. Note que, para esse artigo, focaremos nos sistemas operacionais da Apple, historicamente mais proprietários e fechados para customizações e hacks. A partir daí, discutimos a figura do *designer/hacker/designer*, que altera, recombina e se apropria do controle sobre os aspectos técnicos, de usabilidade e visuais da interface gráfica desses sistemas. O terceiro vértice se solidifica na investigação da atuação *hacker* – em sua faceta de quebra de código para a alteração do estado natural do sistema – sobre a interface gráfica em dispositivos móveis. Na pesquisa originalmente publicada, tomamos como base o fenômeno e as comunidades em volta do jailbreak, um conjunto de técnicas que permite a alteração do sistema operacional utilizado nos smartphones da Apple – o iOS.

A customização na interface gráfica: um levantamento arqueológico

Um dispositivo é considerado um computador moderno quando, usando circuitos eletrônicos integrados pode armazenar, processar e acessar dados. Dados esses que são processados como uma sequência de impulsos elétricos ininteligíveis ao homem. Para que possamos programar, inserir dados e entender os dados processados, medimos esses impulsos elétricos em dois estados, comumente associados aos numerais 1 e 0, nos quais damos o nome de código binário. Por si só, essa já é uma interface de comunicação entre a linguagem humana e os impulsos elétricos em um circuito de um computador.

Portanto o aparecimento de metáforas visuais para a composição de elementos da interface gráfica data antes mesmo da capacidade das máquinas de gerarem gráficos em um monitor imagético por si só: um forte exemplo é o Bombe (1936) de Alan Turing. Um dos primeiros projetos de máquina no modelo computacional contemporâneo, o Bombe foi projetado para decodificar mensagens nazistas de guerra, encriptadas em um código que dificilmente teria sua engenharia reversa usando técnicas de cálculo manual. Enquanto trabalhava, o Bombe girava seus tambores de metal até que uma combinação fosse válida. Quando isso acontecia, os cilindros paravam de rodar, mostrando, como em um ponteiro de relógio, a equivalência de determinado caractere codificado com seu par decodificado. Apesar de se tratar de um sistema mecânico, o conjunto de tambores compunham um painel, que alfabeticamente, mostrava a resposta do enigma nazista. Essas interfaces eram customizadas e construídas para a aplicação de um único uso da máquina, modelo informático que durou até o fim da década de 1950.

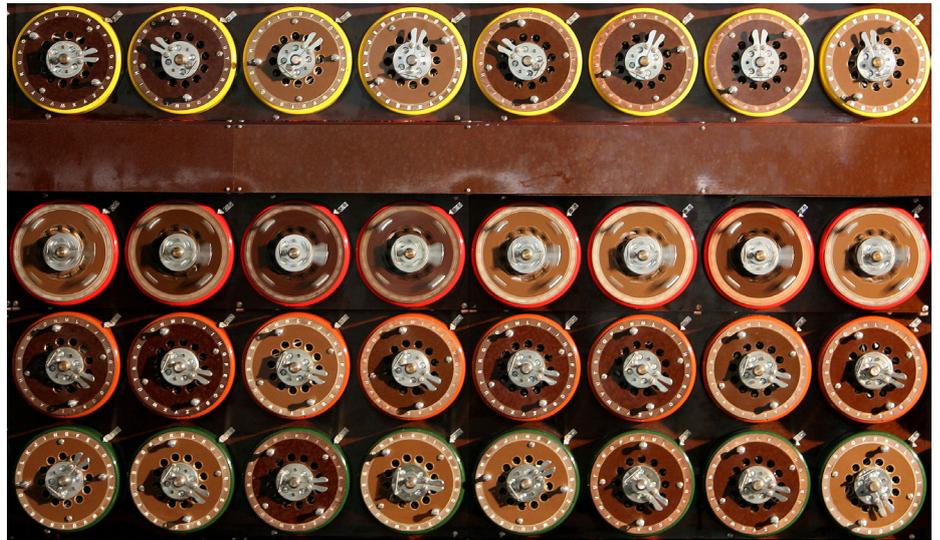


Fig 1. Painel do Bombe, de Alan Turing (1936) | Fonte: Reuters Media

Esse paradigma, entretanto, sofre uma manobra em seu modelo a partir da década de 1960, com pesquisadores buscando a construção de um modelo de computador “universal”, que pudesse ser usado por um grupo crescente de operadores – e depois usuários – para executar diferentes tipos de tarefas e trabalhos. Um dos nomes mais importantes desse movimento foi Doug Engelbart, da Universidade de Stanford. Com o NLS, ele e seu time juntam diversos dispositivos de entrada e saída de dados já existentes (como monitor, mouse e teclado) para formar o modelo de computador pessoal que é vigente até hoje.

O programa de pesquisa que mostrarei a vocês é rapidamente caracterizável por dizer, que se em seu escritório, você como um trabalhador intelectual, lhe fosse fornecido uma tela de computador, alimentado por um computador disponível para você o dia todo e instantaneamente responsivo para cada ação que você tomasse, qual o valor que você daria para isso? Bem, isso basicamente caracteriza o que nós estamos buscando contemplar por muitos anos no que chamamos de Centro de Pesquisa do Intelecto Humano Aumentado, no Instituto de Pesquisa de Stanford. (ENGELBART, 1968)¹

A partir do modelo projetado por Engelbart, as pesquisas e protótipos do que seria a computação das décadas seguintes não cessavam de aparecer nos departamentos de computação e ciência nas universidades americanas e europeias. E um dos projetos mais importantes para a formação do conceito de interface gráfica que aplicamos até hoje surge em Palo Alto, na Califórnia, hoje parte da região conhecida como Vale do Silício.

A Xerox, até então predominante na maioria das empresas estadunidenses na década de 1970, estava preocupada com a tendência e perspectiva de um escritório do futuro onde as pessoas não dependessem de papéis físicos para o fluxo de trabalho em uma companhia. Como seu principal negócio girava em torno de fotocopiadoras, investiu no PARC – Palo Alto Research Center, laboratório de pesquisa e desenvolvimento encarregado de investigar e desenvolver a tecnologia que substituiria o papel – qualquer que ela fosse. A aposta dos pesquisadores foi um sistema de computação em rede onde os usuários não precisassem saber programar uma máquina para utilizar seus recursos. O utilizado em um dos protótipos, o Xerox Star, foi chamado de Conceptual User Model e era baseado na metáfora de objetos e ações encontradas nos escritórios, como calendários, papéis, pastas, impressoras, gaveteiros e, principalmente, a mesa. Representados por ícones de baixa resolução na tela de um monitor, a intenção era que o usuário realizasse ações usando o teclado e mouse, como se estivesse fisicamente interagindo com esses objetos no mundo real. Depois adotado – e popularizado – pelo Macintosh, da Apple, e o Windows, da Microsoft, esse modelo se tornou solução eficaz para a popularização do computador pessoal em escritórios e depois em ambientes domésticos em um cenário onde predominava o ineditismo computacional, e com ele, o alfabetismo digital. A contrapartida desse modelo, entretanto, culminou em um longo legado de falta de opções de customização e personalização do operacional e visual das interfaces gráficas, que carregamos até os dias de hoje. O problema, entretanto, já era reconhecido pelos criadores do Conceptual User Model, que reconheciam a necessidade de que um sistema fosse customizável. No início da década de 1980, época da publicação da pesquisa, acreditavam-se em extensões do usuário em detrimento à modificação do sistema em si, outro sistema vigente até hoje.

“Não importa quão genérico ou poderoso seja o sistema, ele nunca vai satisfazer todos os seus potenciais usuários. As pessoas sempre buscam formas de agilizar ações repetitivas, entretanto, as pessoas são diferentes entre si. A única solução é projetar um sistema com suporte nativo a extensões do usuário.” (SMITH et al., 1982).²

O que os pesquisadores da Xerox definiam como customizável, muito se baseava em na navegação e visualização de arquivos no sistema, baseando-se em opções de visualização temporária de dados, ordenação e filtragem. Mas desde a década de 80 em diante, o computador (e depois, o smartphone) se torna um objeto muito pessoal e, ainda assim, seu aspecto pressupõe de um pensamento baseado em ser fácil de usar por diferentes usuários e compatível com sistemas e outros dispositivos que, no fim, o torna um objeto impessoal. Apesar do Conceptual User Model ser o principal modelo seguido nas interfaces gráficas desde então, o aspecto de customização tem seus protagonismos e apagamentos durante a história da interface gráfica do chamado computador pessoal. Voltaremos nesse assunto mais a frente.

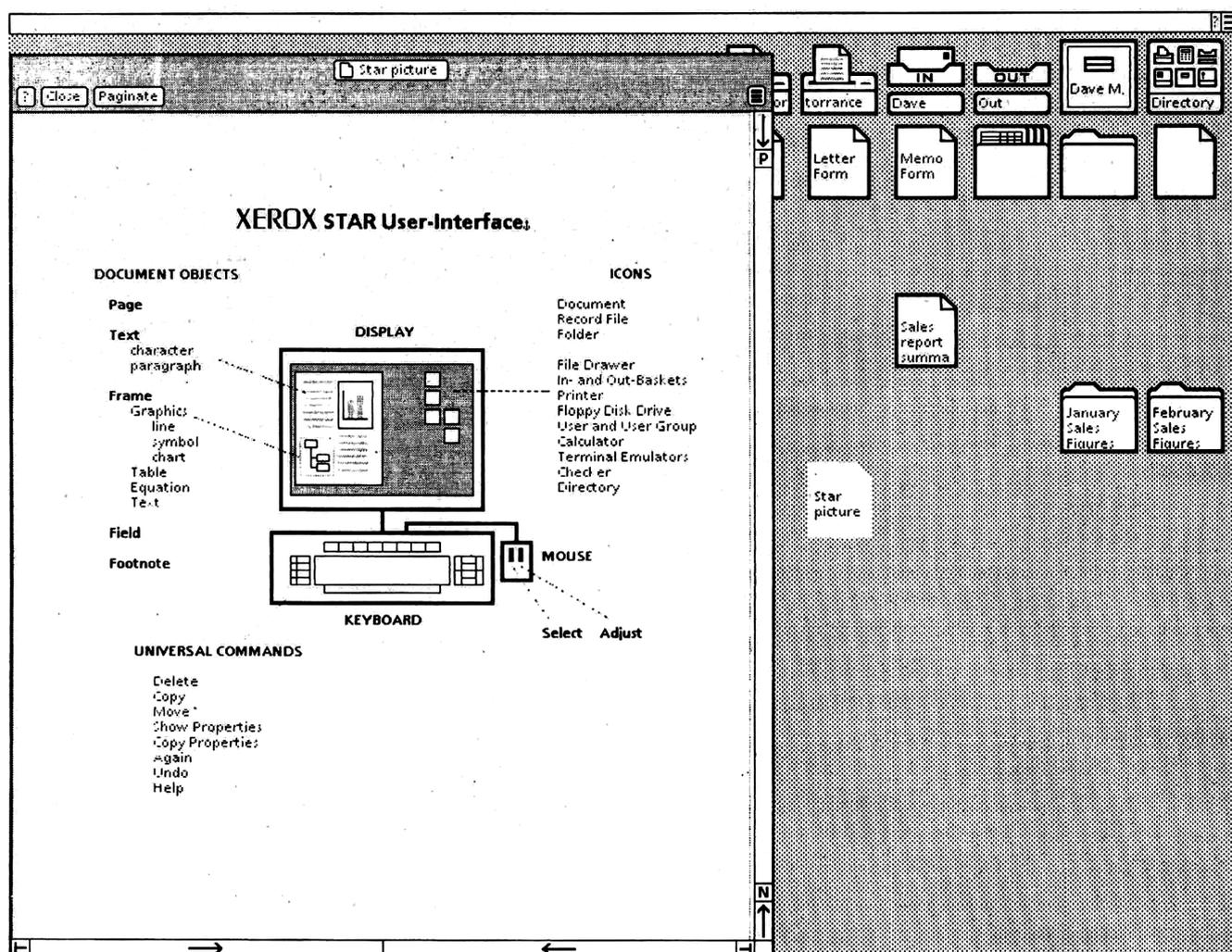


Fig 2. Interface gráfica do Xerox Star

Fonte: Byte Magazine

Customização no Desktop doméstico

O sistema operacional utilizado no Macintosh, modelo de computador pessoal voltado para uso doméstico – e que depois evoluiu para uso corporativo – segue um versionamento que começa em 1978 com o lançamento do primeiro computador do modelo Macintosh. Até a versão 7, ainda com Steve Jobs na liderança da empresa, as opções de customização eram bem limitadas: apenas com a chegada dos monitores coloridos, era possível trocar a cor de destaque, o peso e tamanho da tipografia padrão e densidade do grafismo no fundo da área de trabalho.

Em 1997, o Mac vinha perdendo cada vez mais usuários para o Windows 95, que introduz uma nova interface gráfica como um dos principais atrativos do novo sistema operacional da Microsoft. O hardware e a computação pessoal evoluiu muito rapidamente entre 1984 e 1994, e os sistemas

operacionais de ambas companhias tiveram que serem reescrito para dar suporte aos hardwares que surgiriam nos anos seguintes, incluindo arquivos maiores, telas com maior resolução e a predominância dos processadores de 32 bits. A Microsoft introduz essa nova iteração com o lançamento do Windows 95 em agosto de 1995, mas é só em 1997 que a Apple lança para o mercado o Mac OS 8, quase sete anos depois do lançamento da versão 7 do sistema. Com o código reescrito para dar suporte a hardwares futuros – semelhante ao que a Microsoft fez – a Apple opta por manter a renderização da interface gráfica em um módulo separado, o que permite que usuários avançados e *hackers* alterem a composição dos elementos da interface.

Nas versões anteriores do sistema operacional Macintosh, o aspecto dos itens da interface do usuário foi incorporado nas funções de definição para os itens individuais. Estes foram armazenados em recursos isolados e os padrões ficavam junto aos arquivos do sistema. Mas os programadores podiam escrever os seus próprios aspectos visuais para personalizar a interface do usuário. As aparências da janela eram definidas pelos recursos WDEF, menus por MDEFs e controles por CDEFs. Dar à área de trabalho um novo visual requeria reescrever todos esses pedaços separados de código. (BASALGANGSTER, 2011)³

A customização da interface, entretanto, parecia estar nos planos da empresa para o sistema operacional. Em versões beta, a oitava versão do sistema operacional trazia quatro opções de identidade visual para que o usuário escolhesse qual lhe fosse mais interessante. Entretanto, na versão oficial de lançamento, apenas uma prevaleceu – chamada de Platinum, como o nome oficial da interface gráfica do novo sistema.

Em paralelo, fazendo uso dessa realocação da interface gráfica em um módulo isolado no sistema operacional, surgiram ferramentas que permitiam que o usuário comum, sem nenhum conhecimento de programação, pudesse customizar a interface gráfica. O principal protagonista desse tipo de ferramenta foi o Kaleidoscope, que junto com o software oferece ao público o Theme Archive, um website repositório onde usuários podiam fazer upload e download de visuais – também chamados temas – criados pela comunidade de usuários.

Essa comunidade, por sua vez, passa a chamar a atenção da Apple que incluí um menu nativo de seleção de temas e customização, incluindo a opção de importar temas externos, na versão beta do Mac OS 8.5. Pouco antes da atualização do sistema, a Apple adquire a NeXT, companhia de Steve Jobs, para ter o direito ao código fonte do sistema operacional que seria lançado futuramente como Mac OS X. Com isso, as opções de customização desapareceram na versão oficial de lançamento do Mac OS 8.5.

Não temos informação em primeira mão sobre o porquê da Apple ter desistido dos planos originais nessa área, mas inúmeras fontes dizem que Steve Jobs cortou os temas alternativos assim que o Mac OS 8.5 chegou no status de candidato a versão final. Algumas pessoas afirmam que ele temia que a Apple fosse culpada por problemas de compatibilidade e estabilidade que poderiam acontecer quando determinados aplicativos rodassem em interfaces que não fossem a Platinum. A interpretação mais comum, entretanto, é a de que ele simplesmente não gostou das interfaces High-Tech e Gizmo. (NORR, 2000)⁴

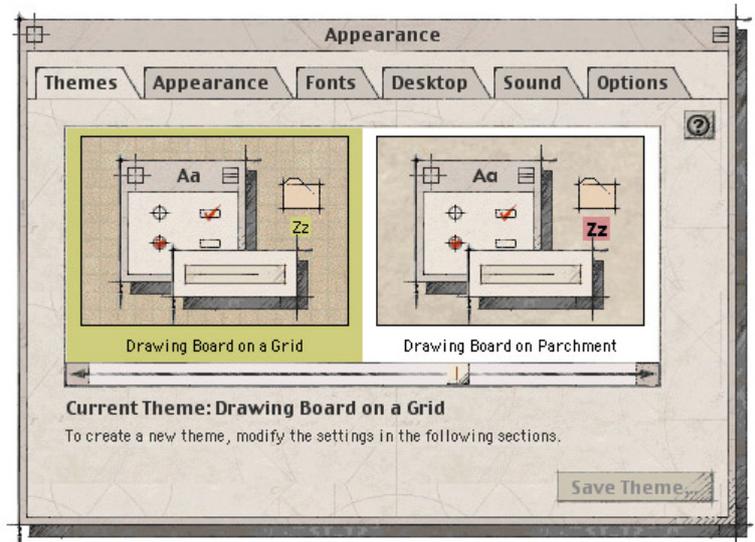
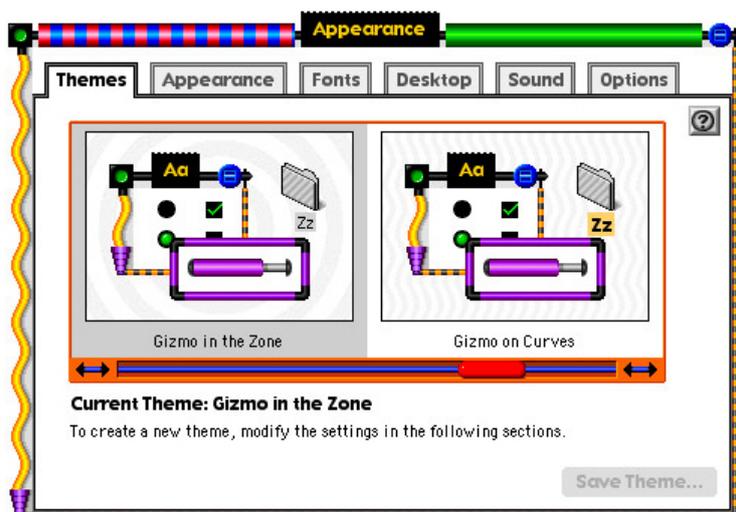
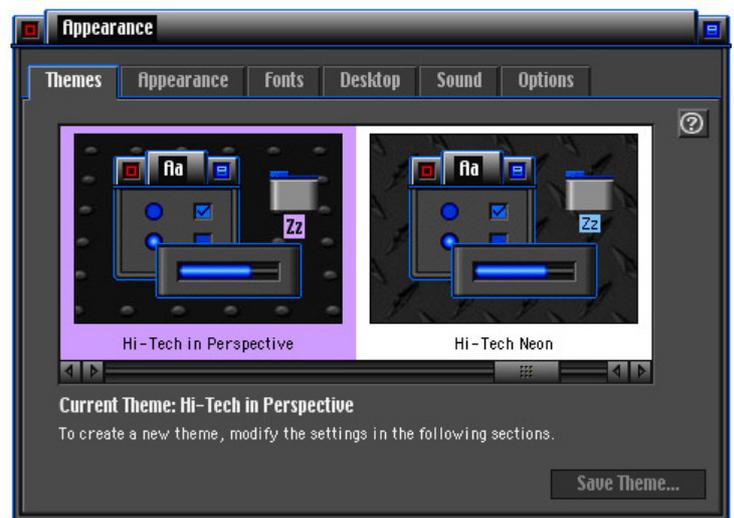
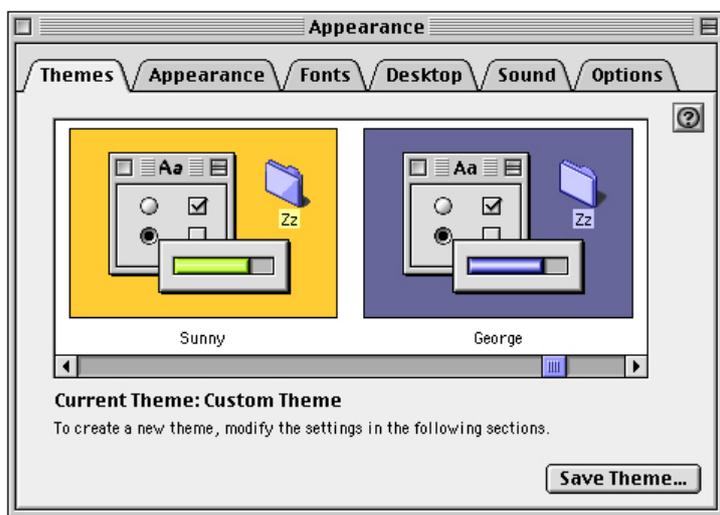


Fig 3. Opções de customização do Mac OS 8 e os temas beta

Fonte: Guidebook

Com o lançamento do Mac OS X em 2001 e sua nova interface gráfica – batizada de Aqua – O Kaleidoscope deixa de funcionar, mas ainda recebendo inúmeras notificações da Apple contra o Theme Archive, onde usuários faziam uploads de temas muito similares ao Aqua do Mac OS X. Nativamente, o OS X não suportava qualquer opção de customização pelo usuário além da cor de destaque e imagem na área de trabalho – semelhante às opções oferecidas pelos primeiros sistemas operacionais do Macintosh quando Jobs também comandava a empresa. Mas a comunidade de customização não interrompe a pesquisa e desenvolvimento de soluções – mesmo que temporárias – para a edição dos aspectos visuais da interface gráfica, como uma repetitiva caça de gato e rato.

Em 2006 um grupo de hackers e desenvolvedores decodifica a maneira como o OS X controla a aparência da interface gráfica e lança o software UNO, que permite a aplicação de temas na interface gráfica no OS X 10 até o 10.4. Com o lançamento do OS X 10.5 em 2008, a Apple muda mais uma vez a maneira como a interface gráfica é controlada no sistema, mas o grupo responsável continua a pesquisa e desenvolvimento da próxima ferramenta. Em 2013 o grupo disponibiliza a evolução do UNO, rebatizado como Flavours, dando suporte para customização das interfaces do OS X até a versão 10.10, quebrando as proteções do sistema a cada nova versão e atualização disponibilizada pela Apple.

O cenário muda em 2015, com o lançamento do OS X 10.11, que inclui uma revisão avançada na segurança da plataforma e – portanto – também dos elementos da interface gráfica como efeito colateral.

Durante anos, colocamos nosso amor e paixão nos temas de interface do Mac. (...) durante todo esse período, investimos muito tempo, recursos, amor e dinheiro no desenvolvimento do Flavours; nosso retorno foi terrivelmente fraco, mas pelo menos nós fomos pagos com amor e apoio da comunidade. Agora, com o OS El Capitan (10.11) anunciado para o outono de 2015, decidimos suspender todo o desenvolvimento do Flavours. (...) obrigado por todo o seu apoio e compreensão. Esperamos que você goste do Flavours 2 Lite, que é, provavelmente, o último suspiro de customização no OS X. (FLAVOURS, 2015)⁵

Em combinação com a implementação do Apple T1, um chip dedicado a criptografia e segurança da informação nos Macintosh lançados a partir de 2016 e depois o Apple T2, ainda mais complexo e seguro. Com isso, a alteração da interface gráfica se torna uma tarefa ainda mais difícil para os hackers, que seguem tentando explorar brechas no sistema. Mas antes de aprofundarmos nessa questão e entrarmos no universo das interfaces gráficas de smartphones, vamos voltar nossa atenção para a comunidade que pesquisa e quebra esses códigos de segurança – os hackers.

Hackers

Popular arquétipo na ficção científica, o hacker é normalmente representado como um super humano, com habilidades virtualmente infinitas de interação com redes e computadores – desde Case, um pirata virtual que banido do ciberespaço luta por sua reintegração em *Neuromancer*, publicado originalmente em 1984, até a ficção contemporânea, como Elliot, anti-herói protagonista da série *Mr. Robot*, televisionada desde 2015, cujo malware afeta uma das mais importantes corporações, levando a sociedade em direção a um colapso econômico e social sem precedentes. No entanto, a realidade não é tão binária como retratado no universo da ficção. Uma das definições mais presentes na literatura do tema é cunhada por Eric Raymond, desenvolvedor, hacker e ativista.

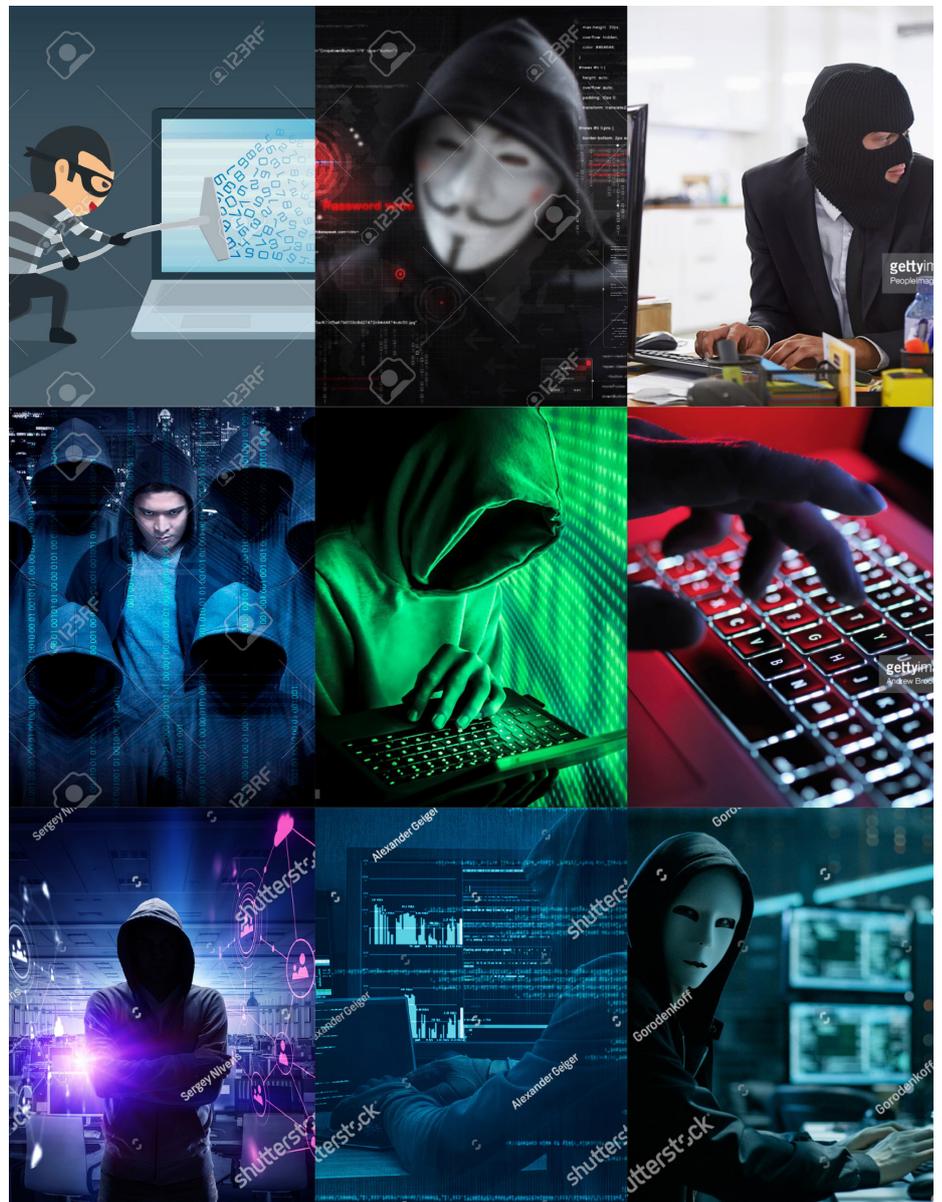


Fig 4. Uma rápida busca nos bancos de imagem mais populares mostra o estereótipo da figura do *hacker* criada no universo da ficção: um personagem suspeito, anônimo e, principalmente, mal-intencionado.

Fonte: Shutterstock, iStock, Getty Images, 123RF, Dreamstime

“Uma pessoa que gosta de explorar os detalhes dos sistemas programáveis e como expandir suas capacidades, ao contrário da maioria dos usuários, que preferem aprender apenas o mínimo necessário. O RFC1392⁶, o Glossário dos usuários da Internet, amplifica isso de forma útil: Uma pessoa que se delicia em ter uma compreensão íntima do funcionamento interno de um sistema, computadores e redes de computadores em particular. [...] Aquele que programa com entusiasmo (mesmo obsessivamente) ou que gosta de programação em vez de apenas teorizar sobre programação. [...] Um especialista ou entusiasta de qualquer tipo. Pode ser um hacker de astronomia, por exemplo. [...] Quem gosta do desafio intelectual de superar ou contornar limitações de forma criativa”. (RAYMOND, 2004)⁷

O termo surge com o início rudimentar e popularização da computação pessoal, no início da década de 1980. Raymond, um dos principais retratistas do surgimento das comunidades de pessoas interessadas em manipular – e expandir – o potencial do maquinário presente nos laboratórios de computação nos câmpus das universidades. Um importante evento na aproximação dessas pessoas foi a ARPAnet, rede de conexão informática antecessora da internet, disponível nos institutos de pesquisa que desenvolviam o sistema. Entre esses pontos de conexão, estava o Massachusetts Institute of Technology (MIT), um dos mais importantes polos de atividade no surgimento da *cultura hacker*.

A "*cultura hacker*" é, na verdade, um aglomerado de subculturas em rede que, no entanto, é consciente de algumas importantes experiências, raízes e valores compartilhados. Tem seus próprios mitos, heróis, vilões, épocas folclóricas, piadas, tabus e sonhos. Os hackers, como um grupo, são pessoas particularmente criativas que se definem em parte pela rejeição de valores "normais" e hábitos de trabalho, tem tradições inusitadamente ricas e conscientes para uma cultura intencional com menos de 50 anos de existência. (RAYMOND, 2004)⁸

O jornalista e historiador Steven Levy confirma esse fenômeno, destacando a importância dos clubes de trens em miniatura, com origem ainda na década de 1950. Geralmente, esses modelos eram complexos, e assim como um computador, dependiam de esquemas eletrônicos elaborados, com circuitos magnéticos, transistores, cabos, componentes e lógicas próprias. Os alunos, em busca de reputação entre os membros dos próprios clubes e em demonstrações para outros grupos, alteravam e aprimoravam os componentes e mecanismos buscando mais eficiência nos problemas logísticos das miniaturas, implementando mecanismos mais simples ou remo-

vendo camadas e camadas de complexidade do sistema, ao mesmo tempo em que experimentavam com a adição de características e funcionalidades nos modelos. É uma das primeiras vezes em que o termo *hack* é utilizado.

"Embora alguém possa chamar uma conexão inteligente entre os [componentes] de um 'mero hack', entende-se que, para se qualificar como um hack, a façanha deve ser imbuída de inovação, estilo e virtuosismo técnico". (LEVY, 2010 p.190)⁹

Especialmente no MIT da década de 1980, a comunidade hacker começa a desenvolver interesse por levar o movimento de expandir, alterar e melhorar para os computadores disponíveis nos laboratórios. Com um sistema de "olheiros", que alertavam aos grupos quando um pesquisador não comparecia a um agendamento para o uso de um dos gigantes e custosos mainframes – reservado para poucos discentes com um objetivo de pesquisa muito específico – alertava os membros dessa comunidade que, sorrateiramente ocupava aquela vaga na agenda dos laboratórios para testar códigos, programas e lógicas desenvolvidas em cadernos e calculadoras nas madrugadas adentro dos dormitórios da universidade. O resultado era então compilado e compartilhado com os membros, que buscavam alternativas quando o código não tinha o resultado esperado ou para comemorar quando uma nova melhoria – um *hack* – teve um resultado positivo.

Surpreendentemente, alguns desses programadores, estudantes de pós-graduação, [...] tinham escrito um programa que utilizava uma das linhas de pequenas luzes no equipamento: as luzes ficam acesas de tal modo que parecia uma pequena bola, passando da direita para a esquerda: se um operador acionasse um interruptor no momento certo, o movimento das luzes podia ser revertido. Ping-pong no computador! (LEVY, 2010 p.194)¹⁰

Um outro importante autor na conceptualização da *cultura hacker* é Richard Thieme, ex-eclesiástico que encontrou na tecnologia uma devoção mais apaixonada. Uma das figuras mais respeitadas em fóruns, imprensa e conferências hackers, o pensador e crítico de tecnologia associa esse surgimento em sua literatura e identifica três gerações de hackers. A primeira delas compartilha muitas das características descritas por Levy (2010) e Raymond (2004), como seres curiosos e empenhados em extrair o máximo possível de cada sistema, equipamento ou software. Mas tem uma observação importante: essa primeira geração de hackers começa a surgir na década de 1980, portanto, antes da internet. As atuações eram em sua maioria locais e com a troca de técnicas e descobertas entre outras comunidades (e universidades) de forma mais lenta. É essa geração que começa a explorar

as redes pré-web que surgem e em fóruns e bulletin boards começa a dar escala para a comunidade que antes era hiperlocalizada nas universidades e centros de pesquisa e desenvolvimento em tecnologia. Apesar do surgimento espontâneo, descentralizado e multidisciplinar, Levy (2010) observa similaridades de pensamento entre essas comunidades iniciais, que o pesquisador denominou de ética hacker. Os principais pontos apontados por Levy (2010), de forma simplificada são:

- **O acesso a computadores, e a qualquer coisa que possa ser útil no entendimento de como o mundo funciona, deve ser total e ilimitada.**

Hackers acreditam que o caminho para tornar o mundo um lugar melhor é através do entendimento de como as coisas funcionam. E para isso, são necessárias desmontagens, testes e investigações, para que projetos possam ser melhorados, aprimorados e até mesmo novas coisas sejam criadas. Um sistema imperfeito incomoda o hacker, que com sua curiosidade e habilidade técnica, irá fazer de tudo para tornar melhor – ou pelo menos questionar esse status – um sistema, software ou equipamento.

- **Toda informação deve ser livre.**

Muitas das descobertas e experiências realizadas pelos hackers provêm de partes de descobertas e resultados obtidos por outros hackers. Com isso, a ética hacker envolve a defesa de um conhecimento livre, aberto e sem barreiras. O conhecimento que é filtrado para evitar o entendimento de um determinado equipamento, software, sistema ou proprietário para fins de monetização são vistos de forma negativa pela comunidade.

- **Questione a autoridade, promova a descentralização.**

Segundo a ética hacker, uma das formas mais pujantes de impedir o acesso às informações ou equipamentos são as autoridades. Nesse contexto, estão incluídas corporações, governos e universidades. Na visão hacker, essas instituições são sistemas falhos, impedem os ímpetos exploradores da comunidade. Em um aforismo, hackers enxergam as autoridades como seguidores de regras arbitrárias, ao contrário dos computadores, que seguem lógicas e algoritmos. Um dos ícones mais fortes de autoridade se traduz na figura da IBM, uma das empresas mais importantes para a computação até a década de 1980. Suas metodologias e processos iam na contramão do pensamento hacker.

- **Hackers devem ser considerados pelos seus hacks, não por sua escolaridade, idade, raça ou posição social.**

A natureza colaborativa e descentralizada da comunidade hacker se opõe às classificações herdadas de outros segmentos da sociedade e instituições autoritárias. Portanto, cada hacker deve ser julgado e provar seu valor por meio dos hacks que executa, e não sua titulação acadêmica, idade ou patrimônio.

- **É possível criar arte e beleza em um computador.**

Os hackers acreditavam ser possível expandir não só a capacidade dos computadores, mas também seu uso. Até os anos 1980, isso exigia muita criatividade e destreza técnica. Um bom exemplo apontado pelo pesquisador é o software desenvolvido por Samson para fazer com que o computador reproduzisse um trecho de uma peça de Bach em seu alto-falante monofônico desenhado apenas para indicar erros de operação em um programa. A beleza do código era validada a partir dos parâmetros de se conseguir executar mais coisas com menos instruções. Na fase descrita por Levy (2010), as máquinas tinham limitações de memória e capacidade de entrada de informações, o que fazia com que houvesse um clima de competição entre os hackers, que passavam horas tentando deixar um programa cada vez mais simples.

- **Os computadores podem transformar sua vida para melhor.**

Esse pensamento era embasado pela experiência dos hackers na comunidade. As máquinas transformaram a vida daqueles que nela descobriram uma paixão. Para além do endocentrismo da comunidade, eles acreditavam que era possível que qualquer pessoa pudesse se beneficiar da existência desse tipo de equipamento. Desde que a comunidade hacker descobrisse novos usos para salas completas de fios e válvulas.

- **Um computador pode realizar seus desejos, assim como uma lâmpada mágica.**

Os computadores eram encarados pelos hackers como potenciais infinitos de transformação, e dependiam apenas das instruções que lhes eram dadas para alcançar seus objetivos.

Entretanto, para alguns professores e corporações, o tipo de experimentação realizada pelos hackers era um desperdício de tempo, e principalmente dinheiro. Boa parte dos coordenadores dos laboratórios nas universidades pregavam que o tempo de computação nas máquinas deveriam ser usados apenas para operações que preenchessem a capacidade total da máquina, e não tentar fazer com que o computador tivesse um caráter interativo, como para fazer contas ou jogar uma partida de xadrez. Com isso, a comunidade hacker tinha não só o papel de aumentar o potencial das máquinas, como também o de elevar seu potencial de uso, evangelizando a comunidade que ia na contramão de seus princípios.

A segunda geração no agrupamento proposto por Thieme (2003) é a que surge já com a rede solidificada, e a internet comercial dando seus primeiros passos. A geração de 1990, que lucra monetizando o tráfego de informações, empreende e leva a pauta do universo “pontocom” para o dia-a-dia dos noticiários econômicos – até a bolha estourar. A terceira geração, nasce e explora o caráter de anonimato trazido pela internet, como um numeroso universo composto de homens e mulheres sem rosto (THIEME, 2003). Com uma abrangência de escala global, é essa geração que vai quebrar códigos para roubar informações sigilosas, prejudicar pessoas e companhias e piratear software, mas é também a geração que fortalece o hacktivism, o desenvolvimento dos grupos de software livre e as comunidades de customização.

Se o hacking sempre foi uma atividade carregada de traços políticos, também é possível afirmar que o hacktivism dá um passo além. Enquanto as primeiras gerações de hackers estavam centradas nas políticas relacionadas a softwares e hardwares, os hacktivistas transpuseram mais claramente esse caráter político ao plano social, realizando ações diretas de desobediência civil. Assim, o ativismo hacker pode ser definido como o uso de ferramentas digitais tendo em vista fins exclusivamente políticos, que não raro são logrados de maneiras transgressivas e/ou disruptivas. De forma mais ampla, trata-se da junção das ferramentas e conhecimentos técnicos encontrados no hacking e de uma forma especial de ativismo político – realizado por meio das redes digitais. (MACHADO, 2015 p.153)

O enfrentamento às autoridades e a personalidade combativa não surge com o anonimato, tampouco com o hacktivism. Uma das mais famosas manifestações dessa atitude pode ser encontrada no manifesto *The Conscience of a Hacker*, uma brincadeira provocativa a *The Conscience of a Conservative*, de Barry Goldwater. O texto foi escrito por Loyd Blakenship, o hacker por trás do codinome *The Mentor*, que em 1986 foi preso por hackear um servidor público. O texto foi publicado no e-zine hacker *Phrack* e ficou conhecido como o *Manifesto Hacker*. Nele, *The Mentor* questiona as acusações e o clima de ilegalidade que gira em torno da comunidade hacker.

Este é o nosso mundo agora. O mundo do elétron e o interruptor, a beleza de um baud¹¹. Fazemos uso de um serviço já existente sem pagar o que poderia ser barato se não fosse administrado por glutões lucrativos, e vocês nos chamam de criminosos. Nós exploramos... e vocês nos chamam de criminosos. Nós existimos sem cor de pele, sem nacionalidade, sem viés religioso... e vocês nos chamam de criminosos. Vocês constroem bombas atômicas, vocês fazem guerras, vocês assassinam, vocês enganam, mentem e tentam nos fazer acreditar que é para o nosso bem, e ainda assim nós somos os criminosos. Sim, eu sou um criminoso. O meu crime é a curiosidade. Meu crime é julgar as pessoas pelo que elas dizem e pensam, não pelo que elas parecem. Meu crime é o de superar você, algo que você nunca vai me perdoar. Eu sou um hacker, e este é o meu manifesto. Você pode parar esse indivíduo, mas você não pode parar todos nós. (BLANKENSHIP, 1986)¹²

Em um contexto contemporâneo e muito utilizado por membros da comunidade, existem em atuação basicamente três tipos de hackers que ativamente pesquisam e manipulam códigos desenvolvidos como camadas de proteção e segurança e buscam brechas que podem ser utilizadas para manipulação. O primeiro deles, é o *White Hat*, hacker com traços semelhantes ao descrito originalmente. Quando descobre uma vulnerabilidade, o *White Hat* comunica aos desenvolvedores responsáveis pelo software, para que esses façam uma atualização resolvendo o problema. Bancos, softwares de segurança e aplicações de infraestrutura costumam pagar bem para hackers que descubram as vulnerabilidades muitas vezes não encontradas por seus times internos de desenvolvimento. Seu oposto é o *Black Hat*. Esse, quando descobre ou cria uma vulnerabilidade em um software ou sistema a usa exclusivamente para benefício próprio: roubo de banco de dados, informações bancárias, propriedade intelectual ou interceptação de e-mail. Muitas vezes o ato de roubar ou interceptar não é realizado pelo hacker em si, mas o *exploit* é vendido ou disponibilizado para criminosos, concorrentes ou qualquer um que não deveria estar em posse das informações que só puderam ser obtidas através do hack. E, como essa divisão não é necessariamente binária, os hackers que se encontram entre esses dois polos são chamados *Gray Hats*. Assim como sua denominação, a moral e ética desses hackers entra em uma área cinzenta, com legalidade discutível. Os hacks que se enquadram nesse ponto da escala podem ser obtidos de forma ilegal, sem consentimento ou autorização das partes envolvidas, que podem oferecer para as companhias uma correção por meio do pagamento de uma taxa ou explorar essa vulnerabilidade sem intenção de lucro, mas que de alguma forma contribua para a comunidade.

Na arte, na ciência, na filosofia e na cultura ou em qualquer forma de produção de conhecimento onde dados podem ser compilados, onde a informação pode ser extraída e onde dessa informação são produzidas novas possibilidades para o mundo, há hackers que pirateiam o novo a partir do antigo. Apesar dos hackers criarem esses novos mundos, nós não os possuímos. O que criamos é financiado para os outros, e para os interesses dos outros, para estados e corporações que controlam os meios para fazer mundos que nós descobrimos sozinhos. Nós não possuímos o que produzimos - a produção é quem nos possui. [...] Hackers fazem parte de uma classe, mas uma classe abstrata, uma classe que precisa se hackear em existência, se manifesta como ela própria. (WARK, 2004 p.10)¹³

Jailbreak

iOS, o sistema operacional dos smartphones e tablets fabricados pela Apple é proprietário. Diferente do seu principal concorrente, o Google Android, o sistema foi desenhado para que não sofra modificações por seus usuários, incluindo customizações na interface gráfica, como vimos anteriormente na versão equivalente para desktop, o macOS. Dois fatores são essenciais na comparação com o sistema Android e sistemas operacionais de computadores, incluindo o macOS e o Windows: O *App Gating* e *Sandboxing*.

O primeiro deles, *App Gating*, não tem como função primária a segurança, essa passa a ser quase uma consequência desse modelo. Em sistemas com *App Gating* (portão de aplicativos, em tradução livre) só é possível instalar software distribuído pela fabricante, nesse caso, a App Store da Apple. Ao desenvolver um aplicativo ou game, o desenvolvedor deve submeter o código à companhia, que vai moderar e passar por um rigoroso critério de aprovação antes que o software seja disponibilizado para os usuários. Diferentemente dos sistemas Android, Windows, MacOS e Linux, não há outra maneira de instalar um software no aparelho (como baixar de um website ou transferir de outro aparelho, por exemplo). De acordo com a companhia, fazer a mediação entre os desenvolvedores e os usuários permite que nenhum aplicativo sobrecarregue os dispositivos, mintam sobre suas características, desempenhem abaixo do prometido ou que possam de alguma maneira, proporcionar uma experiência negativa aos seus usuários. Mas isso também impede que desenvolvedores cobrem pelo aplicativo ou suas funcionalidades fora do sistema de pagamento da App Store, que retém 30% do valor de cada transação ou assinatura.

O *sandboxing*, por sua vez, delimita um espaço virtual onde o aplicativo possa executar suas funções, sem acesso aos arquivos armazenados no dispositivo ou a outros aplicativos. Acesso à galeria de imagens, câmera ou microfone, deve ser autorizado pelo usuário no primeiro uso des-

ses recursos. Quando a transmissão de dados entre aplicativos (um jogo e uma rede social, por exemplo) a mediação é feita pelo sistema operacional. Com isso, nenhum aplicativo tem acesso ou pode executar rotinas de código que façam qualquer leitura ou escrita no sistema operacional, inviabilizando os kits de customização como os encontrados no macOS e discutidos anteriormente.

Desde as primeiras versões do iOS, grupos de hackers ao redor do mundo se reúnem para reverter a engenharia de segurança do sistema operacional e retirar as restrições para que possam alterar o funcionamento do sistema operacional de forma livre. Nos fóruns e comunidades virtuais, um *hack* que tem sucesso em desempenhar esse papel é chamado de *jailbreak*, ou quebra da prisão em tradução livre. É considerado um *jailbreak* uma solução composta de diversos hacks explorando a vulnerabilidades do iOS que consigam reverter o controle sobre aplicativos instalados (*App Gating*), acessar os arquivos do sistema e outros aplicativos (*Sandboxing*) e, portanto, realizar alterações no sistema operacional.

Algumas pessoas, particularmente hackers e modders¹⁴, insistem que o *jailbreaking* é um direito e o fazem como uma proclamação. Eles geralmente comparam o *jailbreak* com um carro: imagine comprar um carro com o capô soldado para que você não possa modificá-lo. Alguns dizem que isso não está certo. (KELLER, 2012)¹⁵

O *jailbreak* foi rapidamente adotado pela comunidade cracker, que usam hacks para a quebra de sistemas de proteção a direito autoral e realizam a distribuição de softwares que quebram a proteção à pirataria de software. No *jailbreak*, essas comunidades construíram redes e repositórios de distribuição de aplicativos e jogos pirateados. Um dispositivo que passou pelo processo de *jailbreak* irá executar qualquer código compatível com o sistema independentemente de sua origem. essa prática se torna uma porta de entrada para aqueles que querem instalar os aplicativos sem pagar o valor cobrado na App Store. Lojas alternativas que distribuem gratuitamente ou por uma fração do preço na App Store ameaçam o modelo de negócios explorado pela Apple com o faturamento provido pela parcela cobrada em cada venda de aplicativo, recurso ou assinatura, um mercado que em 2019 alcançou a figura de meio trilhão de dólares desde o lançamento oficial da App Store, em 2008.

Na pesquisa que compartilha o mesmo título do presente texto, criamos um protótipo/laboratório para explorar as possibilidades de customização na interface gráfica. O dispositivo laboratório teve como objetivo ter sua interface customizada com elementos visuais que fizessem remissão ao universo fantasioso da ficção baseada na figura do hacker, com elementos de código, glitches, bugs e hacks. Uma interface gráfica que fosse não fosse anti-usuário, mas sim, anti-controle.

O aparelho¹⁶, um iPhone 4, não sofreu nenhuma alteração física de hardware com exceção a uma necessária troca de bateria, irrelevante como elemento facilitador do *jailbreak*. Para garantir que nenhuma interferência externa atrapalhasse os tweaks, mantivemos as condições originais do dispositivo, como se o iPhone tivesse acabado de sair da fábrica com a última versão do sistema operacional disponível para esse modelo, o iOS 7.2.1. Também optamos por manter um cartão SIM com uma linha funcional, habilitada em uma operadora. O objetivo dessa análise é alterar a interface gráfica do iPhone no seu máximo possível mantendo-o ainda utilizável, analisando os tweaks individualmente e como eles contribuem para a transformação do protótipo em uma interface gráfica fundamentalmente diferente daquela encontrada originalmente no iOS. (BRAGA, 2018).

Para a composição da interface, que foi envelopada e distribuída para a comunidade *jailbreaker* com o nome UI_dsgHckr, utilizamos 11 hacks que alteraram a tela inicial, barra de status do sistema, habilitaram a troca rápida multitarefas (que só vai aparecer oficialmente no iOS três anos depois), invertem a exibição padrão de fundo branco para fundo negativo (só aparece cinco anos depois de forma oficial), ícones, animações e tipografia de todo o sistema, além de hacks intermediários necessários para a composição visual em um sistema operacional tão modificado. A descrição completa e detalhada dos procedimentos necessários e componentes utilizados, encontra-se no texto pesquisa original.

Para Freeman, o pai do Cydia¹⁷, um homem que viu inúmeros *jailbreaks*, basicamente acabou. Nos velhos tempos, ele diz, os *jailbreaks* funcionavam por meses. Agora, quando um *jailbreak* é publicado, eles são imediatamente desativados. "A Apple tem aumentado a prioridade no bloqueio de *jailbreaks*, a um ponto que estamos não próximos ao topo [da lista de prioridades] que ficou realmente perigoso", diz ele.

Antes um desbravador do *jailbreak*, Freeman chegou no ponto em que não recomenda mais que as pessoas deixem seus telefones livres. É perigoso, devido ao maior risco de ser hackeado, e não vale a pena mais, diz ele num recente telefonema. "O que ganha no final das contas?" ele pergunta. "Costumava ser sobre ter recursos matadores, que quase eram basicamente o motivo pelo qual você possuía o telefone. E agora você recebe uma pequena customização menor". (FRANCESCHI-BICCHIERAI; MERCHANT, 2017)¹⁸

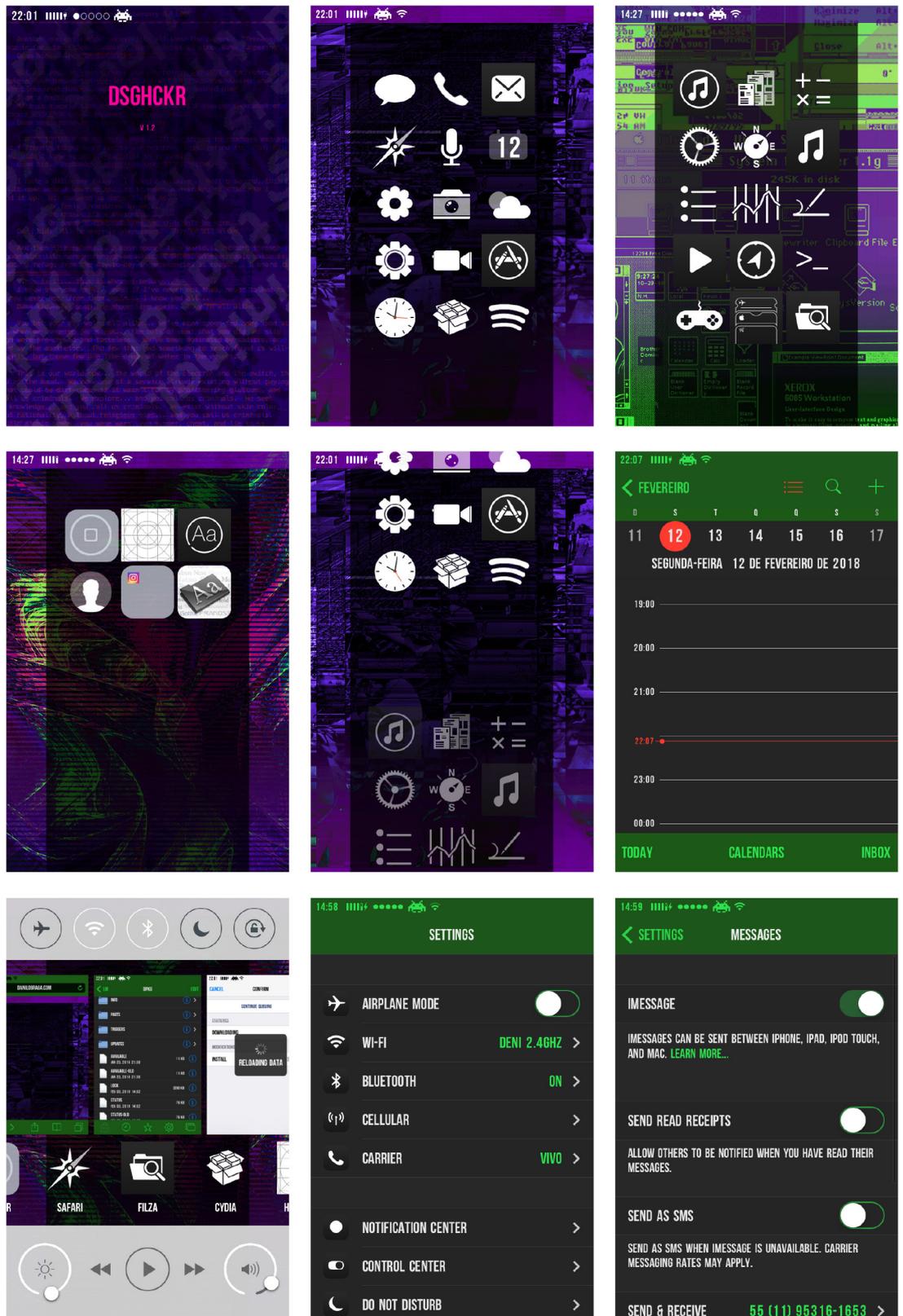


Fig 5. Capturas de tela do protótipo dshgckr, desenvolvido durante a pesquisa.

Fonte: Autor

Conclusão

A partir da geração de dispositivos protagonizado pelo iPhone 7 e todos os modelos seguintes, o time de desenvolvimento da companhia passou a adotar uma nova arquitetura de hardware nos aparelhos que em prol da segurança de dados e informações, insere restrições e criptografias severas impedindo o acesso que permitia que hackers utilizassem a quebra de código para alterar o sistema operacional. No momento da publicação da pesquisa original (em 2018), o sistema operacional iOS estava em sua 11ª versão e o jailbreak parecia estar longe de ter uma volta significativa: os hacks se perdiam ao reiniciar o dispositivo, a atuação era limitada ou o resultado era instável. Mas a comunidade hacker não encerrou a pesquisa e distribuição de kits não encerrou as atividades e as restrições criadas pela arquitetura de segurança atual já foram quebradas. Em sua versão mais recente, o iOS 13, é possível realizar o jailbreak estável e de forma permanente, assim como os primeiros kits disponibilizados para o iOS 7 (utilizado no dispositivo laboratório da pesquisa). A cada kit publicado, a Apple lança em seguida uma atualização corrigindo as vulnerabilidades exploradas pelos hackers. O usuário que tem interesse em explorar o potencial dos dispositivos deve acompanhar de perto os fóruns e comunidades jailbreakers e desativar as atualizações automáticas, fazendo uso de uma vulnerabilidade explorada por um dos publicadores de kit.

A customização no sistema, por outro lado, vem ganhando mais espaço a cada nova atualização na plataforma dos dispositivos. Modificações populares entre a comunidade jailbreaker – como a instalação de tipografias, a inversão da interface para fundo negativo, rearranjo livre de ícones na tela principal e automações de tarefas – se tornaram recursos oficiais da plataforma, com destaque para a próxima versão, iOS 14, que traz esse e mais recursos como opcionais sem a necessidade de quebrar qualquer camada de proteção do sistema operacional.

Cinquenta anos atrás, nos clubes de modelagem de trens, os hackers percursos da expansão de sistemas computacionais pré-determinados criavam mecanismos, jogos, soluções e atalhos de acesso que se tornaram solidificadas indústrias no cotidiano urbano: games, startups, aplicativos e ferramentas que movimentam boa parte da economia global. Dos pesquisadores que acreditavam que a metáfora do escritório facilitaria a operação dos complexos computadores que seriam comuns em escritórios do mundo todo abriram o caminho para a consolidação de um dispositivo que carregamos conosco o tempo todo, expandem a conectividade para ambientes sem fio e revolucionam como a sociedade consome, produz e se comunica. O papel do smartphone tem cada vez ganhado mais espaço no cotidiano – acelerado ainda mais pela pandemia causada pelo vírus causador da COVID-19. E a customização, a passos lentos, ajuda a tornar esse aparelho tão pessoal em um objeto cada vez mais íntimo e singular para cada um de seus usuários. A comunidade hacker é um catalizador do processo de inovação e independência – contratual e de customização – na relação homem máquina e

um exercício de observação no que é distribuído nessas comunidades pode ser um exercício de futurologia e inovação na experiência do usuário, onde é possível ter uma visão nítida do que pode ser a interação no futuro não muito distante. O designer, como observador do comportamento humano e da relação com seus artefatos, tem o privilégio ímpar de poder acessar esse repertório vasto de hacks dando conta do que o mercado ainda não dá, expandindo, alterando e modificando seu devir projetual em um universo livre, colaborativo e como sempre, curioso.

1 Citação original: "The research program that I'm going to describe to you is quickly characterizable by saying, if in your office, you as an intellectual worker were supplied with a computer display, backed up by a computer that was alive for you all day and was instantly responsible, responsive... instantly responsive to every action you had, how much value could you derive from that? Well, this basically characterizes what we've been pursuing for many years in what we call the Augmented Human Intellect Research Center at Stanford Research Institute."

2 Citação original: "No matter how general or powerful a system is, it will never satisfy all its potential users. People always want ways to speed up often-performed operations. Yet, everyone is different. The only solution is to design the system with provisions for user extensibility built in."

3 Citação original: "In previous versions of the Macintosh OS, the look of user interface items was built into the definition functions for the individual items. These were stored in resources, and the standard ones were in the System file. But programmers could write their own to customize the user interface. Window appearances were defined by WDEF resources, menus by MDEFs and controls by CDEFs. Giving the desktop a new look required rewriting all of those separate pieces of code".

4 Citação original: "I have no first-hand information about why Apple backed away from its original plans in this area, but numerous sources say Steve Jobs axed the alternative themes after Mac OS 8.5 had already reached "final candidate." Some people contend that he was afraid Apple would be blamed for compatibility and stability problems that may occur when certain applications are run under the non-Platinum appearances. The most common interpretation, however, is that he simply didn't like the High-Tech and Gizmo looks".

5 Citação original: "For years, we have put our love and passion on Mac theming. (...) During all this time, we have invested lots of time, resources, love and money on Flavors development; our return was terribly poor, but at least we were paid with love and engagement from the community. Now, with OS El Capitan (10.11) announced for Fall 2015, we decided to halt all further development on Flavors. (...) Thank you for all your support and understanding. We hope you enjoy Flavors 2 Lite, which is, probably, the last breath of theming on OS X".

6 Glossário de termos da internet, publicado em janeiro de 1993 por Tracy LaQuey Parker e Gert Scott Malkin (PARKER; MALKIN, 1993).

7 Citação original: "A person who enjoys exploring the details of programmable systems and how to stretch their capabilities, as opposed to most users, who prefer to learn only the minimum necessary. RFC1392, the Internet Users' Glossary, usefully amplifies this as: A person who delights in having an intimate understanding of the internal workings of a system, computers and computer networks in particular. [...] One who programs enthusiastically (even obsessively) or who enjoys programming rather than just theorizing about programming. [...] An expert or enthusiast of any kind. One might be an astronomy *hacker*, for example. [...] One who enjoys the intellectual challenge of creatively overcoming or circumventing limitations".

8 Citação original: "The '*hacker* culture' is actually a loosely networked collection of subcultures that is nevertheless conscious of some important shared experiences, shared roots, and shared values. It has its own myths, heroes, villains, folk epics, in-jokes, taboos, and dreams. Because *hackers* as a group are particularly creative people who define themselves partly by rejection of 'normal' values and working habits, it has unusually rich and conscious traditions for an intentional culture less than 50 years old".

9 Citação original: "While someone might call a clever connection between relays a "mere hack", it would be understood that, to qualify as a hack, the feat must be imbued with innovation, style, and technical virtuosity".

10 Citação original: "Amazingly, a few of these programmers, grad students working with McCarthy, had even written a program that utilized one of the rows of tiny lights: the lights would be lit in such an order that it looked like a little ball was being passed from right to left: if an operator hit a switch at just the right time, the motion of the lights could be reversed Computer Ping-Pong!".

11 Unidade de medida que determina a velocidade de sinalização usada em uma linha de transmissão de dados.

12 Citação original: "This is our world now... the world of the electron and the switch, the beauty of the baud. We make use of a service already existing without paying for what could be dirt-cheap if it wasn't run by profiteering gluttons, and you call us criminals. We explore... and you call us criminals. We seek after knowledge... and you call us criminals. We exist without skin color, without nationality, without religious bias... and you call us criminals. You build atomic bombs, you wage wars, you murder, cheat, and lie to us and try to make us believe it's for our own good, yet we're the criminals. Yes, I am a criminal. My crime is that of curiosity. My crime is that of judging people by what they say and think, not what they look like. My crime is that of outsmarting you, something that you will never forgive me for".

13 Citação original: "In art, in science, in philosophy and culture, in any production of knowledge where data can be gathered, where information can be extracted from it, and where in that information new possibilities for the world are produced, there are *hackers* hacking the new out of the old. While *hackers* create these new worlds, we do not possess them. That which we create is mortgaged to others, and to the interests of others, to states and corporations who control the means for making worlds we alone discover. We do not own what we produce - it owns us. [...] *Hackers* are a class, but an abstract class, a class as yet to *hack* itself into manifest existence as itself".

14 Comunidade que produz as modificações, mas não explora as vulnerabilidades. No texto do autor, há essa divisão entre quem quebra as camadas de segurança e quem produz as modificações. Para efeitos gerais, tratamos nessa pesquisa da comunidade *hacker* como abrangendo também os modders.

15 Citação original: "Some folks, particularly *hackers* and modders, insist that jailbreaking is a right and do it as something of a statement. They'll often compare it soaping up a car: Imagine buying a car with the hood welding shut so you can't modify it. Some say that's just not right."

16 Inicialmente um iPhone 5C foi o aparelho designado para ser a plataforma para os processos de jailbreak da pesquisa. Entretanto, incompatibilidades com a versão do iOS instalada no smartphone, iOS 10.3.2, apareceram com a metodologia de jailbreak escolhida, um jailbreak semipersistente nomeado como o *hacker* que o programou, o h3lix.

17 Principal loja alternativa de hacks para dispositivos que passaram pelo processo de jailbreak.

18 Citação original: "For Freeman, the father of Cydia, a man who's seen countless jailbreaks, it's basically all over. Back in the good old days, he says, jailbreaks would work for months. Now, when there are public jailbreaks, they get killed immediately.

"Apple has both upped priority on fixing jailbreaks but also we have moved so far up the stack that we're actually dangerous," he says.

Once a jailbreak crusader, it's gotten to the point where Freeman no longer recommends that people jailbreak their phones. It's dangerous, due to the higher risk of getting hacked, and it's not even worth it anymore, he says in a recent phone call. "What do you get in the end?" he asks. "It used to be that you got killer features that almost were the reason you owned the phone. And now you get a small minor modification".

Referências

- BASALGANGSTER, Copland. **The Long View**. Disponível em: <http://basalgangster.macgui.com/RetroMacComputing/The_Long_View/Entries/2011/2/26_Copland.html>. Acesso em: 30 ago. 2017.
- BLANKENSHIP, Loyd. **Gravação do painel “The Conscience of a Hacker”**. Convenção H2K2, Nova York: [s.n.], 2002. Disponível em: <<https://www.youtube.com/watch?v=0tEnnvZbYek>>. Acesso em: 26 dez. 2017.
- BLANKENSHIP, Loyd. **The Conscience of a Hacker**. Phrack, v. 1, n. 7, 1986. Disponível em: <<http://www.phrack.org/issues/7/3.html#article>>. Acesso em: 28 dez. 2017.
- BRAGA, Danilo. **Design Hacker: Customização de interfaces gráficas móveis por alteração de código**. Dissertação (Mestrado em Design) - Programa de Pós-Graduação em Design, Universidade Anhembi Morumbi. São Paulo, 240 f., 2018.
- DOUG ENGELBART INSTITUTE. **Doug’s 1968 Demo**. Disponível em: <<http://dougengelbart.org/firsts/dougs-1968-demo.html>>. Acesso em: 30 ago. 2017.
- DVORAK, John. **A drastic change in direction**. Disponível em: <<http://www.guidebookgallery.org/articles/adrastricchangeindirection>>. Acesso em: 30 ago. 2017.
- ENGELBART, Douglas. **1968 Demo - FJCC Conference Presentation Reel #1**. [s.l.: s.n.], 1968. Disponível em: <https://archive.org/details/XD300-23_68HighlightsAResearchCntAugHumanIntellect>.
- FLAVOURS. **Flavours - Tasteful Themes for Mac. Flavours**. Disponível em: <<http://flavours-classic.interacto.net>>.
- FLAVOURS. **Flavours 2 - Themes for Mac. Flavours 2**. Disponível em: <<http://flavours.interacto.net>>.
- FRANCESCHI-BICCHIERAI, Lorenzo Lorenzo; MERCHANT, Brian. **The Life, Death, and Legacy of iPhone Jailbreaking**. Motherboard, 2017. Disponível em: <https://motherboard.vice.com/en_us/article/8xa4ka/iphone-jailbreak-life-death-legacy>. Acesso em: 3 fev. 2018.
- FULGONI, Gian. **The State of Digital**. In: [s.l.: s.n.], 2017. Disponível em: <<https://www.comscore.com/Insights/Presentations-and-Whitepapers/2017/The-State-of-Digital>>. Acesso em: 16 nov. 2017.
- HAYES, Frank; BARAN, Nick. **A Guide to GUIs**. Disponível em: <<http://www.guidebookgallery.org/articles/aguidetoguis>>. Acesso em: 30 ago. 2017.
- INTERACTO. **interacto.net - Flavours - Tasteful Themes for Mac**. Disponível em: <<http://flavours-classic.interacto.net>>. Acesso em: 30 ago. 2017.
- KELLER, Mike. **Geek 101: What Is Jailbreaking?** PCWorld, 2012. Disponível em: <https://www.pcworld.com/article/249091/geek_101_what_is_jailbreaking_.html>. Acesso em: 3 jan. 2018.
- LEVY, Steven. **Hackers: Heroes of the Computer Revolution**. Sebastopol, CA: O’Reilly Media, 2010.
- MACHADO, Murilo Bansi. **Entre o controle e o ativismo hacker: a ação política dos Anonymous Brasil**. História, Ciências, Saúde-Manguinhos, v. 22, n. suppl, p. 1531–1549, 2015.
- MACHRONE, Bill. **Drawing back the curtain on Windows shows Microsoft has a clear edge**. Disponível em: <<http://www.guidebookgallery.org/articles/drawingbackthecurtain>>. Acesso em: 30 ago. 2017.

- MACHRONE, Bill. **Time for a new interface**. Disponível em: <<http://www.guidebookgallery.org/articles/timeforanewinterface>>. Acesso em: 30 ago. 2017.
- NORR, Henry. **Mac OS 8.5 Special Report: Themes**. Disponível em: <https://www.macintouch.com/m85_themes.html>. Acesso em: 30 ago. 2017.
- NORR, Henry. **Mac OS 8.5 Special Report: User Interface**. Disponível em: <https://www.macintouch.com/m85_face.html>. Acesso em: 30 ago. 2017.
- RAYMOND, Eric. **A Brief History of Hackerdom**. Disponível em: <<http://www.catb.org/~esr/writings/cathedral-bazaar/hacker-history/index.html>>. Acesso em: 19 dez. 2017.
- RAYMOND, Eric. **Crackers, Phreaks, and Lamers**. Disponível em: <<http://www.catb.org/jargon/html/crackers.html>>. Acesso em: 30 ago. 2017.
- RAYMOND, Eric. **Hacker Speech Style**. Disponível em: <<http://www.catb.org/jargon/html/speech-style.html>>. Acesso em: 30 ago. 2017.
- RAYMOND, Eric. **Hacker Writing Style**. Disponível em: <<http://www.catb.org/jargon/html/writing-style.html>>. Acesso em: 30 ago. 2017.
- RAYMOND, Eric. **hacker**. Disponível em: <<http://www.catb.org/jargon/html/H/hacker.html>>. Acesso em: 30 ago. 2017.
- RAYMOND, Eric. **Overgeneralization**. Disponível em: <<http://www.catb.org/jargon/html/overgeneralization.html>>. Acesso em: 30 ago. 2017.
- RAYMOND, Eric. **The Jargon Lexicon**. Disponível em: <<http://www.catb.org/jargon/html/lexicon.html>>. Acesso em: 30 ago. 2017.
- RAYMOND, Eric. **The Meaning of 'Hack'**. Disponível em: <<http://www.catb.org/jargon/html/meaning-of-hack.html>>. Acesso em: 30 ago. 2017.
- SEYMOUR, Jim. **Making good GUI sense**. Disponível em: <<http://www.guidebookgallery.org/articles/makinggoodguisense>>. Acesso em: 30 ago. 2017.
- SEYMOUR, Jim. **New interface dilemmas**. Disponível em: <<http://www.guidebookgallery.org/articles/newinterfacedilemmas>>. Acesso em: 30 ago. 2017.
- SMITH, David; IRBY, Charles; KIMBALL, Ralph; et al. **Designing the Star User Interface**. Byte Magazine, v. 7, n. 4, p. 544, 1982.
- THIEME, Richard. **Hacker Generations**. CTheory, v. e127. Event Scenes, 2003. Disponível em: <http://ctheory.net/ctheory_wp/hacker-generations/>. Acesso em: 26 dez. 2017.
- WARK, McKenzie. **A Hacker Manifesto**. Harvard University Press, 2004.

Recebido: 14 de setembro de 2020.

Aprovado: 13 de outubro de 2020.