

**ESCOLA SUPERIOR ABERTA DO BRASIL - ESAB
CURSO DE PÓS-GRADUAÇÃO LATO SENSU EM ENGENHARIA DE
SISTEMAS**

JACK SUSLIK POGORELSKY JUNIOR

O NOVO SISTEMA FIREFOX OS PARA CELULARES

**VILA VELHA - ES
2013**

JACK SUSLIK POGORELSKY JUNIOR

O NOVO SISTEMA FIREFOX OS PARA CELULARES

Monografia apresentada ao Curso de Pós-Graduação em Engenharia de Sistemas da Escola Superior Aberta do Brasil, como pré-requisito para obtenção do título de especialista em Engenharia de Sistemas, sob orientação da Professora Janaína Costa Binda.

**VILA VELHA - ES
2013**

JACK SUSLIK POGORELSKY JUNIOR

O NOVO SISTEMA FIREFOX OS PARA CELULARES

Monografia aprovada em 19 de fevereiro de 2014

Banca Examinadora

Almir da Cruz Souza

Luciana Genelhú Zonta

Janaina Costa Binda e Marcony Brandão Uliana (suplentes)

**VILA VELHA - ES
2013**

RESUMO

Este trabalho tem como objetivo analisar, por meio de pesquisa e levantamento de informações em livros, artigos, manuais e sites dos fabricantes de sistemas operacionais e smartphones, as funcionalidades do Firefox OS em relação aos principais concorrentes (Symbian, Android, Windows Phone, iOS, etc), comparando suas principais características, vantagens e desvantagens em relação aos sistemas operacionais pesquisados. Os smartphones são aparelhos que unem nos mesmos dispositivos recursos como telefone, SMS, GPS, câmera, bluetooth, e-mail, internet e wi-fi. Para que os smartphones funcionem é necessário que os aplicativos acessem recursos do hardware. Para que um aplicativo acesse um recurso de hardware é preciso que exista um sistema operacional (como Unix, Windows e Linux) e no caso dos smartphones um sistema operacional móvel (como Android, iOS e Firefox OS). Os sistemas operacionais podem ser de código fonte fechado, desenvolvido apenas pelas empresas proprietárias ou de código fonte aberto, desenvolvido pela empresa ou fundação mantenedora com apoio de uma comunidade de voluntários. O Firefox OS é o sistema operacional móvel de código aberto desenvolvido pela Mozilla Foundation para smartphones com arquitetura ARM com apoio de uma comunidade de desenvolvedores voluntários espalhados pelo mundo. O Firefox OS é baseado no núcleo Gonk, no engine runtime Gecko e na interface gráfica Gaia. Os aplicativos para o sistema podem ser criados em HTML5, CSS3 e JavaScript como uma aplicação web.

Palavras-Chave: Smartphones, Firefox OS, Mozilla Foundation, Sistemas Operacionais

LISTA DE FIGURAS

Figura 01: Palmtop.....	3
Figura 02: Processador ARM.....	4
Figura 03: Logotipo da Mozilla Foundation.....	18
Figura 04: Arquitetura resumida do Firefox OS.....	33
Figura 05: Arquitetura completa do Firefox OS.....	34
Figura 06: Interface Gráfica do Firefox OS.....	37
Figura 07: Notícia de lançamento no portal G1.....	39
Figura 08: Smartphone Keon com Firefox OS.....	40
Figura 09: Smartphone Peak com Firefox OS.....	41
Figura 10: Parceiros da Mozilla Foundation.....	42

LISTA DE TABELAS

Tabela 01: Comparação entre os sistemas operacionais móveis.....	43
--	----

SUMÁRIO

1 INTRODUÇÃO.....	1
2 SMARTPHONES.....	2
2.1 HISTÓRIA DOS SMARTPHONES.....	3
2.2 ARQUITETURA DOS SMARTPHONES.....	3
2.3. SOFTWARE LIVRE.....	6
2.3.1 Histórico.....	7
2.3.2 Organizações Importantes.....	17
2.4. SISTEMAS OPERACIONAIS.....	19
2.4.1 Os Mais Importantes Sistemas Operacionais.....	20
2.4.2 Sistemas Operacionais Móveis.....	23
2.5. HTML5, CSS3 E JAVASCRIPT.....	26
2.5.1 Html5.....	26
2.5.2 CSS3.....	27
2.5.3 JavaScript.....	29
3 FIREFOX OS.....	31
3.1 ARQUITETURA.....	33
3.1.1 Gonk.....	35
3.1.2 Gecko.....	36
3.1.3 Gaia.....	36
3.2 DESENVOLVIMENTO DE APLICATIVOS.....	38
3.3 LANÇAMENTO.....	38
3.4 PRIMEIROS APARELHOS.....	39
3.5 PARCEIROS NO DESENVOLVIMENTO.....	41
4 COMPARAÇÃO ENTRE OS SISTEMAS OPERACIONAIS MÓVEIS.....	43
5 CONCLUSÕES.....	44
6 REFERÊNCIAS.....	45
APÊNDICES.....	48

1 INTRODUÇÃO

O objetivo deste trabalho é analisar as funcionalidades do Firefox OS em relação aos principais concorrentes (Symbian, Android, Windows Phone, iOS, etc), comparando as principais características, as vantagens e as desvantagens entre os sistemas operacionais pesquisados. A metodologia utilizada foi a pesquisa bibliográfica, fazendo levantamento de informações em livros, artigos, manuais e sites dos fabricantes de sistemas operacionais e smartphones,

Este trabalho aborda os temas necessários para o entendimento do Firefox OS, que são os smartphones, o Software Livre, os Sistemas Operacionais, o HTML5, o CSS3, o JavaScript e a arquitetura do próprio Firefox OS.

Com previsão de lançamento em 2013 o Firefox OS é um sistema operacional para dispositivos móveis desenvolvido pela Mozilla Foundation. O projeto é de código aberto e por conta disso o seu desenvolvimento conta com apoio de uma comunidade de desenvolvedores voluntários.

O desenvolvimento e as características do Firefox OS precisam ser compreendidas no contexto das necessidades de mercado, dos concorrentes, das tecnologias disponíveis, da sua arquitetura, das bases de sua programação e das características de desenvolvimento de um sistema operacional livre para dispositivos móveis.

2 SMARTPHONES

Um smartphone reúne num aparelho as principais funcionalidades de comunicação necessárias atualmente, como telefone, SMS, GPS, câmera, bluetooth, e-mail, internet e wi-fi . Conforme Morimoto (2009) podemos definir os smartphones como:

Os smartphones são a combinação de duas classes de dispositivos: os celulares e os assistentes pessoais (como os Palms e os PDAs). Diferente dos antecessores, os smartphones podem se conectar à web através de conexões 3G ou Wi-Fi, o que permite que eles ofereçam uma enorme variedade de recursos.

Para que os smartphones funcionem é necessário que os aplicativos acessem recursos do hardware. Para que um aplicativo acesse um recurso de hardware é preciso que exista um sistema operacional (como Unix, Windows e Linux) e no caso dos smartphones um sistema operacional móvel (como Android, iOS e Firefox OS). Ao optar pela compra de um smartphones de determinada marca e modelo está se determinando qual sistema operacional será utilizado e com isso os aplicativos que já virão instalados por padrão e os aplicativos que estarão a disposição para instalação.

Ao contrário do que ocorre nos computadores pessoais onde com certa facilidade se pode alterar o sistema operacional (ou até mesmo instalar mais de um simultaneamente) num smartphone essa troca é muito difícil e na maioria das vezes impossível. Morimoto (2009) afirma que:

Diferente de um PC, onde você pode escolher qual sistema operacional utilizar, substituindo o Windows por Linux (ou vice-versa), ou mesmo instalando vários sistemas no mesmo micro, nos smartphones a escolhas são mais restritas, já que, com poucas exceções, você não tem como substituir o sistema pré-instalado por outros.

2.1 HISTÓRIA DOS SMARTPHONES

A história dos smartphones é muito recente sendo a Agenda Eletrônica, que possuía apenas 8 bits e pouca memória SRAM compartilhada para trabalho e armazenamento, o seu antecessor, reunindo as principais funcionalidades (ainda de forma bastante primitiva) que não necessitassem de conexão ao telefone e internet.

A partir da década de 90 surgiram os handhelds e palmtops (Figura 01) que deram origem aos smartphones da atualidade.



Figura 01: Palmtop

2.2 ARQUITETURA DOS SMARTPHONES

Criada pelo Acom Computers Limited em 1983, ARM é a arquitetura mais utilizada para smartphones na atualidade devido a baixa demanda de energia, o seu tamanho reduzido de núcleo e por ser suportado por todos sistemas operacionais móveis.

ARM é propriedade da Advanced RISC Machines que não produz processadores, mas licencia empresas como Intel, Apple, Motorola, Qualcomm e Samsung.

A arquitetura dos smartphones segundo Morimoto (2009) se divide em Processadores, Displays, Acelerômetro, Bluetooth, Wi-Fi, GPS, Câmera, Método de Escrita, Memória e Baterias, podendo ser detalhada conforme abaixo:

- Processadores: atualmente os processadores ARM (Figura 02), licenciados pela ARM (Advanced RISC Machine), são os mais utilizados em smartphones. RISC (Conjunto Reduzido de Instruções de Computador) é um conjunto de instruções utilizadas durante o processamento. Processadores não são encontrados como ARM, pois a Advanced Risc Machine não fabrica, apenas licencia e projeta os núcleos de processamento. Os complementos como memória RAM, Redes Sem Fio e Controlador USB são incluídos pela empresa fabricante do SoC (System on Chip). Por esse motivo temos a Apple com SoC próprio, Samsung com o Exynos, a Qualcomm com o Snapdragon, a Nvidia com o Tegra.



Figura 02: Processador ARM

- Displays: atualmente os displays são dominados pelas tecnologias LCD, OLED e AMOLED. LCD (Liquid Crystal Display): é o display que domina as telas dos smartphones há muito tempo, pois apresenta uma boa qualidade de cores e um consumo relativamente baixo de energia. A tecnologia OLED (Organic Light-Emitting Diode) surge com telas mais finas, menor consumo de energia, maior resistência e menor tempo de resposta em relação ao LCD. A tecnologia AMOLED (Active-Matrix Organic Light-Emitting Diode) é o display OLED de matriz ativa, apresentando cores mais realistas e tela mais flexível.
- Acelerômetro: é o componente que permite orientar a tela conforme o Smartphone é girado permitindo o efeito de alternar entre retrato e paisagem conforme a necessidade.
- Bluetooth: um recurso de baixo consumo de energia para redes PAN de curta

distância. Possui velocidade baixa para utilização em redes mas adequado para pequenas transferências de arquivos e para ligar o smartphones a periféricos como uma impressora.

- Wi-Fi: recurso que já vem instalado (necessitando apenas configuração) utilizado para conectar o smartphone a redes locais e à internet. É um recurso de maior velocidade e maior consumo de energia que o Bluetooth.
- GPS: é um sistema que informa a posição geográfica a partir de sinais recebidos de satélites que pode vir integrado ao smartphone. Apesar dos smartphones apresentarem menos recursos e menor sensibilidade que um aparelho de GPS convencional eles tem a vantagem de unir os dois aparelhos sem aumento muito significativo no valor de custo do aparelho.
- Câmera: embora não tenha a mesma qualidade e os mesmos recursos que uma câmera digital os smartphones atualmente apresentam uma foto de boa qualidade para visualização de fotos em tela de computadores e tem a vantagem de reunir vários recursos no mesmo aparelho..
- Método de Escrita: os métodos utilizados para escrever nos smartphones são a escrita através dos teclados físicos (numérico ou QWERTY) e touchscreen.
- Memória: nos smartphones temos a memória RAM que armazena os programas que estão sendo executados no momento e define a quantidade de programas que poderão ser utilizados simultaneamente. A memória RAM é complementada pela memória Flash que armazena o sistema operacional, configurações, programas instalados e arquivos.
- Baterias: atualmente os smartphones utilizam bateria de Li-Ion (Figura 09) que possui maior capacidade de armazenamento de energia, menor peso e são livres do “efeito memória”, podendo ser recarregadas a qualquer momento sem esperar a descarga completa como acontecia com as baterias de Níquel-Cádmio.

2.3. SOFTWARE LIVRE

O Software Livre ou Software de código aberto é promovido principalmente por duas grandes fundações a FSF (Free Software Foundation) e a OSI (Open Source Initiative).

Segundo a Free Software Foundation (FSF) os usuários devem ter o direito de executar, copiar, distribuir, estudar, mudar e melhorar o software. A Free Software Foundation (2013) define o conceito de software Livre::

Por Software Livre devemos entender aquele software que respeita a liberdade e senso de comunidade dos usuários. Grosso modo, os usuários possuem a liberdade de executar, copiar, distribuir, estudar, mudar e melhorar o software. Com essas liberdades, os usuários (tanto individualmente quanto coletivamente) controlam o programa e o que ele faz por eles. Quando os usuários não controlam o programa, o programa controla os usuários. O desenvolvedor controla o programa e, por meio dele, controla os usuários. Esse programa não-livre e 'proprietário' é, portanto, um instrumento de poder injusto.

A Open Source Initiative (OSI) define software livre com base na “Open Source Definition” do projeto Debian. O site do projeto Debian (2013) disponibiliza a definição conforme:

1. Redistribuição livre: A licença de um componente Debian não pode restringir nenhuma parte interessada em vendê-lo, ou distribuir o software como parte de uma distribuição agregada de software contendo programas de diversas fontes diferentes. A licença não pode exigir um royalty ou outra taxa por esta venda.
2. Código Fonte: O programa deve incluir código fonte e deve permitir a distribuição em código fonte, bem como em formato compilado.
3. Trabalhos Derivados: A licença deve permitir modificações e trabalhos derivados, e deve permitir que estes sejam distribuídos sob a mesma licença que o trabalho original.
4. Integridade do Código Fonte do Autor: A licença pode restringir o código fonte de ser distribuído de forma modificada somente se a licença permitir a distribuição de "patch files" com o código fonte, com o propósito de modificar o programa em tempo de compilação. A licença deve permitir explicitamente a distribuição de software compilado a partir do código fonte modificado. A licença pode exigir que trabalhos derivados tenham um nome ou número de versão diferente do software original (este é um meio-termo; o grupo Debian encoraja todos os autores a não restringir nenhum arquivo, fonte ou binário, de ser modificado).
5. Não à discriminação contra pessoas ou grupos: A licença não pode

discriminar nenhuma pessoa ou grupo de pessoas.

6. Não à discriminação contra Fins de Utilização: A licença não pode restringir ninguém de fazer uso do programa para um fim específico. Por exemplo, ela não pode restringir o programa de ser usado no comércio, ou de ser usado para pesquisa genética.

7. Distribuição de Licença: Os direitos atribuídos ao programa devem aplicar-se a todos aqueles para quem o programa é redistribuído, sem a necessidade de execução de uma licença adicional por aquelas pessoas.

8. A Licença não pode ser específica para o Debian: Os direitos atribuídos ao programa não podem depender do programa ser parte de um sistema Debian. Se o programa for extraído do Debian e usado ou distribuído sem o Debian, dentro dos termos da licença do programa, os mesmos direitos garantidos em conjunto ao sistema Debian deverão ser garantidos àqueles que o utilizam.

9. A Licença não deve contaminar outros softwares: A licença não poderá colocar restrições em outro software que é distribuído juntamente com o software licenciado. Por exemplo, a licença não pode insistir que todos os outros programas distribuídos na mesma mídia sejam software livre.

10. Licenças Exemplo: As licenças "GPL", "BSD" e "Artistic" são exemplos de licenças que consideramos "livres".

Segundo o site da Free Software Foundation (2013) para um programa poder ser considerado Software Livre precisa obedecer as quatro liberdades essenciais:

Liberdade 0: A liberdade de executar o programa, para qualquer propósito;

Liberdade 1: A liberdade de estudar como o programa funciona, e adaptá-lo às suas necessidades. Para tanto, acesso ao código-fonte é um pré-requisito;

Liberdade 2: A liberdade de redistribuir cópias de modo que você possa ajudar ao próximo;

Liberdade 3: A liberdade de distribuir cópias de suas versões modificadas a outros. Desta forma, você pode dar a toda comunidade a chance de beneficiar de suas mudanças. Para tanto, acesso ao código-fonte é um pré-requisito.

2.3.1 Histórico

Cronologia dos fatos mais importantes na história do Software Livre conforme disponibilizado no site Wikipédia (2013) inicia nos Estados Unidos em 1956.

1956: O governo dos EUA proíbe a AT&T de entrar no comércio de software (isso levou, posteriormente, o Bell Labs a distribuir livremente o seu Unix).

1960: O software é distribuído com seu código-fonte e sem nenhuma restrição em grupos de software como SHARE (IBM) e DECUS (DEC).

1969: O Request for Comments (RFC), que descreve a primeira Internet (depois chamada de ARPANET), é publicado; Ken Thompson e Dennis Ritchie desenvolvem uma versão simplificada do MULTICS para um PDP-7. Esse sistema foi batizado por Brian Kernighan de UNICS e depois como UNIX (um sistema operacional multitarefa).

Surgimento do Software Restrito e Berço do Software Livre: Durante a década de 60 uma mudança começou a ocorrer: conforme os sistemas operacionais e os compiladores das linguagens de programação evoluíram, os custos manufatureiros de software aumentavam dramaticamente. Uma pequena, embora crescente indústria de software se formava, e competia com os softwares empacotados pelas manufaturas de hardware (o custo do software empacotado estava incluído nos custos do hardware), máquinas arrendadas exigiam suporte de software, mas não forneciam nenhum retorno pelo software, e alguns clientes capazes de satisfazer melhor suas próprias necessidades não queriam estes custos de software embutidos nos custos dos produtos de hardware. No processo antitruste Estados Unidos contra IBM, iniciado em 17 de Janeiro de 1969, o governo americano cobrou que software empacotado era anticompetitivo .

A IBM, líder do mercado de computadores de grande porte, anunciou que a partir de 1970 iria vender parte de seus programas separada do hardware. Assim, a indústria de software mudou a sua cultura, tornando cada vez mais comum as restrições de acesso e as possibilidades de compartilhamento de código entre desenvolvedores. O software começou a ser comercializado como os produtos de prateleira de um supermercado. Bill Gates, aos 21 anos, através da “carta aberta aos hobistas” levanta o potencial comercial do software como produto no mercado de microcomputadores e questiona a viabilidade do desenvolvimento fora do “modelo de prateleira”. Nesta carta, Bill Gates alega que a prática de compartilhamento de

software não é justa e afirma que tal prática evita que software bem feito seja escrito. Neste mesmo período houve iniciativas que levaram à produção de sistemas de software livre que estão em uso até os dias de hoje, como o SPICE, TeX e Unix. Desde 1972, o Unix está em contínuo desenvolvimento e deu origem a diversas variantes, comercializadas por dezenas de empresas. O Unix tornou-se popular entre os desenvolvedores por conta da sua filosofia de compartilhamento e a relação com as universidades e centros de pesquisas. Mesmo assim, em 1991, a AT&T Bell Labs demonstrou uma mudança de mentalidade ao tentar processar a Universidade de Berkeley por conta da publicação do código Unix BSD que o grupo de pesquisa de Berkeley (CSRG – Computer Systems Research Group) tinha criado. Desde 1973, o CSRG havia sido um dos principais centros de desenvolvimento do Unix e de aplicações a ele relacionadas.

1970: Surge a ARPANET, precursora da Internet; A IBM começa a vender seu software separadamente, estabelecendo assim o início da indústria do software restrito.

1972: Portagem do Unix para outros PDP. Para simplificar esse processo, a equipe de Ken Thompson tenta reescrever o Unix em outra linguagem (B ou BCPL). A equipe melhora a linguagem B e cria a linguagem C para reescrever o Unix; Unix começa a ser distribuído em universidades e centros de pesquisa.

1973: Inicia-se a história do Unix BSD com a chegada do Unix à Universidade de Berkeley, na Califórnia; SPICE é colocado por Donald Pederson em domínio público.

1975: Lançada a primeira versão do Ingres, banco de dados livre (o ancestral do PostgreSQL).

1976: Bill Gates escreve a “carta aberta aos hobistas”.

1977: A Arpanet atinge mais de 100 computadores.

1978: Donald Knuth, da Universidade de Stanford, começou a trabalhar no TeX, distribuído como software livre.

1979: AT&T começa a forçar licenças restritivas ao decidir que poderia lucrar através do sistema Unix.

O Nascimento do Movimento do Software Livre: Em 1980 foram criados os primeiros projetos organizados de forma consciente para serem software livre, assim como foram estabelecidos os fundamentos éticos, legais e financeiros desse movimento. Em 1984, Richard Stallman, então funcionário do laboratório de inteligência artificial do (AI Lab) MIT, deixou seu emprego e começou a trabalhar no Projeto GNU. Stallman gostava de compartilhar seus interesses tecnológicos, conhecimentos e código, algo incompatível com seu ambiente de trabalho no MIT. Podemos exemplificar alguns fatos ocorridos: um caso onde uma irritante falha no software de uma impressora Xerox não podia ser corrigida pois o código-fonte estava inacessível para seus usuários; um desacordo entre Stallman e a Symbolics sobre o acesso do MIT às atualizações feitas em sua máquina Lisp. Estes fatos motivaram Richard Stallman a criar um mecanismo legal a fim de garantir que, além daqueles que receberiam os programas diretamente do Projeto GNU, todos os demais pudessem desfrutar os direitos de copiar, redistribuir e modificar o software. Inclusive, todos aqueles que recebessem algum software após qualquer número de redistribuições e, eventualmente, modificações, também deveriam poder gozar dos mesmos direitos associados ao software original distribuído pelo Projeto GNU. Esta foi a razão pela qual ele elaborou a Licença GPL. Além disso, para institucionalizar o Projeto GNU, bem como obter fundos para desenvolver e proteger o software livre, de acordo com os princípios éticos que ele publicou no Manifesto GNU, Stallman fundou a Free Software Foundation (FSF). Dessa forma, definitivamente nomeado, nasce o Movimento do Software Livre.

1981: A IBM fecha acordo com Microsoft para que ela forneça o DOS para o PC, mas desconsidera a relevância do software e abre mão do copyright do DOS, abrindo espaço para a expansão da Microsoft, graças ao mercado de clones do PC surgido posteriormente.

1983: Richard Stallman posta mensagem no grupo net.unix-wizards com o assunto “new Unix implementation”; Anuncia a criação do GNU e explica seus princípios para

a necessidade de criação de um novo Unix; Menciona que serão necessários um núcleo, um editor e um compilador, entre outras ferramentas. No final ele pede contribuições na forma de máquinas, dinheiro e ajuda para escrever o software.

1984: Stallman pede demissão do AI Lab do MIT para se dedicar ao Projeto GNU, e usa o termo software livre no Manifesto GNU; O primeiro software do Projeto GNU é liberado, o GNU Emacs, escrito por Richard Stallman e Guy L. Steele.

1985: O consórcio X distribui o sistema de janela X como software livre; Richard Stallman funda a Free Software Foundation (FSF). A FSF define software livre baseado em 4 liberdades fundamentais; GCC, escrito por Richard Stallman e Len Towe, tem sua primeira versão finalizada.

1987: Andrew Tanenbaum escreve uma versão didática simplificada do Unix, denominada MINIX; Para arrecadar dinheiro, a FSF vende cópias do software GNU em fita magnética por 150 dólares.

1989: A FSF cria o conceito de copyleft e a Licença GPL para garantir as 4 liberdades do software; Michael Stonebreaker, criador do Ingres, lança o PostgreSQL como software livre; A Cygnus, primeira empresa que essencialmente começou a prover serviços para software livre é fundada por Michael Tiemann, David Henkel-Wallace e John Gilmore; Começa a ser desenvolvido o Network Simulator (NS), um simulador de rede de telecomunicações livre, que passaria a ser o mais usado por pesquisadores de todo o mundo. O advento da Usenet no começo de 1980 interligou ainda mais a comunidade de programadores e forneceu uma maneira mais simples para programadores compartilharem software e contribuírem para softwares escritos por outros 8 .

Linux: Em julho de 1991, Linus Torvalds, um estudante finlandês da Universidade de Helsinki, de 21 anos de idade, divulgou sua primeira mensagem mencionando o seu projeto de construir um sistema livre similar ao Minix, e obteve ajuda de diversos desenvolvedores, de diversas partes do planeta. Linus já estava na pós-graduação e resolveu fazer experiências com o novo computador 386 que recebera na época. Ele conseguiu fazer com que um primeiro esboço do que seria o núcleo de seu sistema

operacional executasse dois programas concorrentemente. Assim, anunciou na Internet que tinha um protótipo de sistema operacional. Em setembro do mesmo ano, Linus lançou uma versão oficial. Em março de 1994, a versão 1.0, a primeira a ser chamada de estável, foi liberada. Durante esse período, centenas de desenvolvedores se juntaram ao projeto para integrar todo o sistema GNU em torno do núcleo do Linux. Ao contrário dos BSDs, o núcleo do Linux e um grande número de componentes integrados em torno dele foram distribuídos sob licença GPL. A falta de um núcleo informático para o Projeto GNU significa que não havia nenhum sistema operacional completamente livre. O desenvolvimento do núcleo de Torvalds fechou este último vácuo. A combinação do quase completo sistema operacional GNU e o núcleo Linux criou o primeiro sistema operacional totalmente livre. Dessa forma, nasceu o sistema operacional GNU/Linux. Linus relata que não queria dinheiro para esta empreitada por uma série de razões. Quando ele postou o Linux originalmente, ele sentiu que estava seguindo os passos de centenas de cientistas e outros acadêmicos: Pessoas que construíram seu trabalho apoiando-se em outros, ou seja, “apoiando-se nos ombros de gigantes”. Em 1992, surgiram as primeiras distribuições GNU/Linux, entre elas a SLS, que mais tarde deu origem ao Slackware (ainda distribuída atualmente). Isso levou à criação de uma competição no mundo dos sistemas empacotados em torno do GNU/Linux. Desse momento em diante, o modelo original de desenvolvimento de software e compartilhamento de conhecimento foi resgatado, porém coexistindo com o modelo de software restrito que ganhara força nas duas décadas anteriores. O aspecto mais revolucionário do movimento de software livre não está na questão do código ser aberto, mas no fato de ter sido a primeira comunidade a explorar as novas possibilidades de desenvolvimento e colaboração, geograficamente distribuídos, que a Internet possibilita. Dentre as distribuições GNU/Linux, a Debian GNU/Linux, iniciada por Ian Murdock em 1993, é notável por ser explicitamente comprometida com os princípios do GNU e da FSF para com softwares livres. Os princípios dos desenvolvedores do Debian estão expressos no Contrato social do Debian. Desde a sua inepção, o projeto Debian tem sido ligado de perto com a FSF, e foi de fato patrocinado pela FSF durante o ano de 1994-5. Em 1997, o ex-líder do projeto Debian Bruce Perens também ajudou a encontrar Software in the Public Interest, uma organização sem fins lucrativos de apoio e financiamento para vários projetos de software livre 9 . O GNU/Linux permanece software livre sob os termos da GNU GPL, mas muitas

empresas oferecem produtos personalizados baseados nele, ou suporte técnico profissional, em contratos comerciais. A nomenclatura permanece controversa dentre as comunidades, com grupos argumentando para ambos "Linux" (o nome genérico para código apenas aberto) e "GNU/Linux" (para código totalmente livre) para o sistema operacional completo.

1990: A FSF anuncia que pretende desenvolver um núcleo para o sistema GNU, chamado de GNU Hurd, com o objetivo de completar o sistema operacional.

1991: William e Lynne Jolitz escrevem uma série sobre “como adaptar o BSD Unix para PC i386”; Linus Torvalds anuncia a criação do Free Minix, usando as ferramentas do Projeto GNU, como o GCC, e poucos meses depois lança a primeira versão do Linux.

1992: A força aérea dos Estados Unidos faz um contrato com a universidade de Nova York para desenvolver uma versão livre do compilador ADA. Eles escolhem o GNU GCC como base e criam o GNAT (GNU NYU Ada 95 Translator); Lançado 386BSD 0.1, que depois deu origem ao NetBSD, FreeBSD e OpenBSD.

1993: A empresa SuSE é fundada com negócios são baseados em Slackware traduzido para o Alemão; FreeBSD 1.0 é disponibilizado na Internet.

1994: A Ada Core Technologies é fundada pelos desenvolvedores do GNAT e se torna líder do mercado de compiladores ADA; Debian GNU/Linux 0.91 é lançado, como resultado do desenvolvimento voluntário de 20 pessoas; Marc Ewing lança a primeira versão do Red Hat Linux.

1995: Bob Young funda a Red Hat Software ao comprar a distribuição Red Hat Linux; O Red Hat Linux 2.0 é lançado, a primeira com o formato de empacotamento RPM; Apache 0.6.2, primeira versão oficial, é lançado; Criado o MySQL.

1996: O projeto KDE é anunciado com o objetivo de tratar os problemas de usabilidade para o usuário final de ambientes similares ao Unix, que utilizam o sistema de janelas X.

Em 1997, Eric Raymond, apresenta o artigo e palestra A cathedral e o bazar, onde discute as vantagens técnicas do software livre e aborda os mecanismos de funcionamento do desenvolvimento descentralizado. Em 1998, Raymond foi um dos protagonistas, junto com Linus Torvalds, da criação da Open Source Initiative (OSI), defendendo a adoção do software livre por razões técnicas e sugerindo o uso da expressão open source ao invés de free software. A principal motivação para a adoção da expressão open source foi introduzir o software livre no mundo dos negócios de uma forma mais palatável para empresas mais conservadoras, evitando a ambiguidade do termo free (que pode significar tanto livre quanto gratuito, na língua inglesa). A Open Source Initiative foi fundada em Fevereiro de 1998 para encorajar o uso do novo termo e evangelizar os princípios do código aberto para todos .

Para divulgar o lançamento desta nova forma de apresentar o movimento do software livre, Raymond publica, em fevereiro de 1998, o texto Goodbye, free software; hello, open source . Alguns membros da comunidade, em especial Richard Stallman e parte da comunidade em torno da FSF, não concordaram com o uso do termo pois, para eles, o termo software livre é mais apropriado. Assim, Stallman publica, em resposta, o texto: Why Free Software is better than Open Source. Como a receptividade do termo Open Source nas empresas foi bem significativa, ele foi adotado por muitos como a melhor maneira de se referir ao software livre, particularmente na língua inglesa. A pluralidade de ideias e concorrência natural entre os sistemas e aplicações dentro do movimento software livre fazem parte de seu mecanismo de evolução, bem como influencia positivamente em sua qualidade. A concorrência entre os navegadores, ferramentas de escritório, gerenciador de janelas e banco de dados são os exemplos mais conhecidos. Do restante da história do software livre até os dias atuais, podemos encontrar uma grande quantidade de soluções de alta qualidade que foram e estão sendo liberadas sob licença livre, em geral apoiadas tanto pela OSI) quanto pela FSF.

1997: Eric Raymond apresenta a palestra A cathedral e o bazar. Miguel de Icaza anuncia o projeto GNOME, um concorrente do KDE que nasce como resposta da FSF a problemas de licença com o KDE.

1998: Netscape Communications Corporation libera o navegador Netscape Communicator como software livre. Este código se transformou dentre outros, no popular Mozilla Firefox e no Thunderbird; Eric Raymond cria o movimento Open Source e cria a OSI; Corel anuncia o NetWinder, uma rede de computador baseada em Linux; Sun Microsystems e Adaptec são as primeiras grandes empresas a fazerem parte da Linux International; IBM anuncia que irá comercializar e dar suporte à Apache; Debian GNU/Linux 2.0, como trabalho de mais de 300 voluntários; KDE 1.0 é lançado e incorporado a várias distribuições GNU/Linux; Linus é capa da revista Forbes, representando o reconhecimento do mundo corporativo ao Linux e ao software livre; Microsoft reconhece o GNU/Linux e o software livre como importantes concorrentes e descreve como atacá-los, de acordo com os Halloween documents; O arcabouço Qt foi licenciado sob a licença Q Public License QPL.

1999: Dell, HP e SGI anunciam que irão dar suporte a GNU/Linux em seus computadores; GNOME 1.0 é lançado; Red Hat Software compra a Cygnus e se torna a maior empresa do mundo na área de software livre; Criado o SourceForge.net, maior repositório de projetos de software livre.

2000: Mozilla M13, o primeiro considerado estável, é lançado; Criada a fundação GNOME; A Sun libera o código-fonte do seu StartOffice sob licença LGPL e é criado o projeto OpenOffice.org.

2001: Linux 2.4 é lançado; IBM anuncia investimento de US\$ 1 bilhão no Linux; Surge a Wikipedia; Neste mesmo ano, Marcelo Tosatti (brasileiro!) tornou-se o primeiro latino americano a manter um código-fonte do kernel, indicado por Alan Cox e aprovado por Linus.

2002: Consórcio ObjectWeb é fundado pela Bull, France Telecom e INRIA, na França; KDE 3.0 e GNOME 2.0 são lançados e os gerenciadores de janelas chegam ao nível de concorrer com os desktops comerciais; Mozilla 1.0, o primeiro oficial estável, é disponibilizado; OpenOffice.org 1.0 é lançado; As primeiras licenças Creative Commons para compartilhamento de outros tipos de obras intelectuais (que não software) são publicadas.

2003: Motorola inicia vendas do A760, o primeiro telefone celular com um Linux; Mozilla Foundation é criada; Lançado o Fedora Core, versão comunitária do Red Hat Linux; SCO processa IBM por suposto código de sua propriedade inserido no Linux, levantando dúvidas no mercado sobre a legitimidade do software livre e da licença GPL, mas perde.

2004: Novell compra a SuSE por 210 milhões de dólares; Primeira versão do Ubuntu e do Firefox são lançadas;

2005: MandrakeSoft compra a empresa brasileira Conectiva e a americana Lycoris, resultando na Mandriva; Open Document Format (ODF), usado pelo OpenOffice 2.0, é reconhecido como padrão pela OASIS; Sun Microsystems disponibiliza o Open Solaris, versão livre do sistema operacional Solaris; Nicholas Negroponte anuncia o projeto OLPC (One Laptop Per Child), projeto do MIT tendo como base o Linux.

2006: Primeiro protótipo do XO (One Laptop per Child) é disponibilizado; Sun libera a máquina virtual Java sob licença GPL (alterada para permitir que se distribua em aplicações comerciais); ODF é padronizado pela OASIS ODF TC e se torna um padrão ISO; Firefox atinge 200 milhões de downloads (12% do mercado mundial e 20% do Europeu); Neo1973, usando a plataforma OpenMoko para telefones móveis, é apresentado; Marcelo Tosatti desliga-se na tarefa de manutenção do núcleo e passa a trabalhar na RedHat.

2007: A Sun libera também a JDK sob GPLv2; Após longas controvérsias, FSF lança a versão definitiva da GPL versão 3; O estudo FLOSSImpact 11 sobre o efeito (especialmente econômico) do software livre, financiado pela Comissão Europeia, é publicado (o primeiro estudo de larga escala sobre o assunto); A Red Hat lança o IcedTea.

2008: Nokia compra TrollTech, dona da biblioteca multiplataforma e livre Qt, e anuncia transformação do Symbian em software livre.

2009: Oracle compra Sun Microsystems por US\$ 7,4 bilhões e entra definitivamente

no mercado de software livre, inclusive adquirindo o MySQL que havia sido comprado pela Sun.

2010: Oracle fecha escritórios da Sun na América Latina e não mantém a mesma relação com a comunidade software livre. A comunidade do Java ligada ao movimento do software livre volta a discutir a preferência e investimentos no OpenJDK; A Trolltech tornou a versão compatível com Unix das bibliotecas Qt disponíveis sob a GPL, além da QPL; A Global Graphics completa uma pesquisa de opinião com 400 diretores de informática (em inglês: CIO) de organizações acima de 1000 empregados nos EUA e no Reino Unido que mostra que aproximadamente três quartos, 76 por cento, das grandes organizações utiliza software livre em seus negócios, e mais da metade, 51 por cento, planeja aumentar esta utilização em 2010 12 .

2011: Linus Torvalds disponibiliza a versão 2.6.38 do kernel do Linux. Entre as novidades, uma modificação no gerenciamento de processos (o wonder patch), que permite um melhor desempenho aos ambientes Desktop; Rompendo a tradição, neste mesmo ano foi lançada a versão 3.0 sem que houvesse alterações de grande impacto. A razão para isto é uma homenagem ao software livre, que faz 20 anos; OSI define uma nova política de estruturação interna que prevê a participação direta da comunidade em sua governança. Anuncia-se que os membros do corpo diretivo da OSI serão agora eleitos pelas instituições associadas a OSI e pelos grupos de trabalho da comunidade.

2.3.2 Organizações Importantes

O Software Livre é incentivado e mantido por comunidades, fundações e empresas através do trabalho de seus funcionários, apoio de comunidades de voluntários e de patrocinadores dos projetos. As mais reconhecidas organizações são:

- Mozilla Foundation (Figura 03) que é uma fundação sem fins lucrativos que mantém Firefox, Thunderbird, Bugzilla, SeaMonkey, Firefox OS, entre outros projetos.



Figura 03: Logotipo da Mozilla Foundation

- Apache Software Foundation que é uma comunidade que desenvolve o servidor HTTP e o OpenOffice;
- Linux Foundation que é uma fundação que promove, protege e padroniza a marca Linux.
- KDE que é uma comunidade que produz diversos aplicativos, mas reconhecida principalmente pelo Desktop Plasma;
- Projeto Gnome que é um projeto que produz diversos aplicativos, mas reconhecida principalmente pelo Desktop Gnome;
- XFCE que é uma comunidade que produz diversos aplicativos, mas reconhecida principalmente pelo Desktop XFCE;
- Red Hat que é uma empresa que patrocina o projeto Fedora e desenvolve o Red Hat Enterprise Linux.
- Novell que é uma empresa que patrocina o projeto OpenSuse e desenvolve o Suse Linux.
- Canonical que é uma empresa que patrocina e desenvolve Ubuntu/Linux e o Ubuntu Phone.
- Google que é uma empresa que patrocina e desenvolve o Android e o Navegador Chromium.

2.4. SISTEMAS OPERACIONAIS

O site do Projeto Debian (2013) define que “Sistema operacional é um conjunto de programas básicos e utilitários que fazem o seu computador funcional”. Por analogia “Sistema operacional móvel é um conjunto de programas básicos e utilitários que fazem o seu dispositivo móvel funcional”.

Conforme Ferreira (2009, p. 21) é possível definir como as funções principais de um sistema operacional:

- Tratamento de Interrupções: o sistema operacional deve ter condições de tratar eventos inesperados;
- Tratamento de Exceções: o sistema operacional deve ter condições de tratar interrupções originadas em algum programa em execução gerando aviso apenas no programa que causou a exceção.
- Gerenciamento de Programas: o sistema operacional deve gerenciar todos os programas executados no computador.
- Gerenciamento de Memória RAM: o sistema operacional deve gerenciar a disponibilidade de memória RAM a todo momento.
- Gerenciamento de Arquivos: o sistema operacional deve gerenciar toda criação, edição e exclusão de arquivos feitas no computador.
- Gerenciamento de Dispositivos: o sistema operacional deve gerenciar todos os dispositivos conectados ao computador.
- Suporte a Redes: o sistema operacional deve gerenciar o acesso do computador as redes de computadores.
- Sistema de Segurança: o sistema operacional deve gerenciar as autorizações de acesso de usuários e programas autorizados.

2.4.1 Os Mais Importantes Sistemas Operacionais

Dentre os Sistemas Operacionais mais importantes e mais utilizados atualmente destacam-se o Unix, o Windows, o Linux e o MAC OS, no entanto existem outros sistemas operacionais utilizados, mesmo que em baixa escala e mercados restritos.

O Unix é um sistema operacional multitarefa, multiutilizador, de código fechado, criado na AT&T em 1969, composto basicamente de um kernel e programas de sistema (shell, bibliotecas, utilitários e em alguns casos de ambiente gráfico). Do Unix derivaram o Linux, o BSD, o Solaris e até o MAC OS X. Um consórcio de várias empresas de informática, chamado The Open Group é proprietário da marca Unix.

A estrutura dos diretórios do Unix, muito semelhante a de seus derivados, é constituída conforme abaixo.

- / - Diretório raiz
- /bin - programas do sistema
- /boot - inicialização do sistema.
- /dev - arquivos para acesso aos dispositivos.
- /etc - arquivos de configuração
- /home - arquivos
- /lib - bibliotecas usadas pelo sistema e módulos do núcleo.
- /mnt - montagem de dispositivos.
- /proc - arquivos do núcleo
- /root - usuário root.
- /sbin - programas usados pelo superusuário (root)
- /tmp - arquivos temporários
- /usr - programas
- /var - arquivos gravados com frequência pelos programas do sistema.

O Windows é um Sistema Operacional com interface gráfica, multitarefa, multiusuário, de código fechado e de propriedade da Microsoft, surgiu com a necessidade de criar um ambiente gráfico para o Sistema Operacional MS-DOS, que não dispunha de interface gráfica amigável. Atualmente o Windows é o sistema operacional mais utilizado no mundo, embora grande parte da sua utilização seja decorrência de cópias ilegais.

O Linux é o núcleo de um sistema operacional de código aberto, licenciado por GPL, criado por Linus Torvalds e que pode ser executado em diversas plataformas, de relógios a grandes computadores. Muito utilizado em servidores de grande porte está ganhando espaço em computadores pessoais com interfaces que facilitam a vida do usuários como KDE, XFCE, LXDE e Gnome.

Muitos aplicativos, interfaces gráficas, utilitários que são necessários para um Sistema Operacional ter utilidade são desenvolvidos pelo projeto GNU, por isso a forma mais adequada é nomear o Sistema Operacional de “GNU/Linux” quando há uma referência ao núcleo do sistema operacional acrescido de um conjunto de funcionalidades oriundas do projeto GNU.

A combinação de ferramentas GNU com núcleo Linux é detalhada no site do Projeto Debian (2013):

A combinação da filosofia e metodologia Debian e as ferramentas GNU, o kernel Linux, e outro importante software livre, formam uma distribuição de software única chamada Debian GNU/Linux. Esta distribuição é feita de um grande número de pacotes de software. Cada pacote da distribuição contém executáveis, scripts, documentação, e informação de configuração, e tem um maintainer que é o primeiro responsável por manter o pacote atualizado, seguir relatórios de bugs, e comunicar com o(s) autor(es) do software original do pacote. A nossa extremamente grande base de utilizadores, combinada com o nosso sistema de seguimento de bugs assegura que os problemas são encontrados e resolvidos rapidamente.

Os sistemas operacionais com núcleo Linux são apoiados por empresas como Red Hat, Canonical, Novell, Google, HP e IBM.

As distribuições GNU/Linux são compostas pelo núcleo e um conjunto de softwares escolhidos (e alguns desenvolvidos) para prover funcionalidades que se adaptem as necessidades dos usuários de suas comunidades. O usuário, porém, tem a condição de remover ou acrescentar programas conforme sua necessidade. As principais distribuições atualmente, conforme o site Distrowatch (2013) são Mint, Debian, Ubuntu, Mageia, Fedora, PCLinuxOS e OpenSuse.

O FreeBSD é um sistema operacional livre originado do BSD (Berkeley Software Distribution), que por sua vez tem origem no Unix nos anos 1970. Os sistemas BSD

utilizam licença BSD, diferente da licença GPL do Linux. A partir da licença BSD, ao contrário da GPL) é possível utilizar o código fonte para criação de produtos fechados. O site do FreeBSD (2013) define o sistema operacional como:

Em síntese, FreeBSD é um sistema operacional UNIX-like para plataformas i386 e Alpha/AXP, baseado no "4.4BSD-Lite" da Universidade da Califórnia em Berkeley, com alguns aprimoramentos adotados do "4.4BSD-Lite2". O FreeBSD também é baseado, indiretamente, na conversão de William Jolitz conhecida como "386BSD" para a plataforma i386 do "Net/2" da Universidade da Califórnia, em Berkeley; apesar que pouquíssimo código originado do 386BSD ainda exista no FreeBSD. Uma descrição mais abrangente do que é FreeBSD e como o sistema funciona, pode ser encontrada na página principal do FreeBSD.

O FreeBSD é amplamente utilizado por empresas, Provedores de Serviço Internet, pesquisadores, profissionais de informática, estudantes e usuários domésticos no mundo todo, para trabalho, educação e recreação. Alguns destes exemplos podem ser observados na Galeria FreeBSD,

Muitos aplicativos do Linux como Skype podem ser utilizados no FreeBSD com bom desempenho. O Linux por sua vez não consegue executar binários do FreeBSD.

Dos sistemas originários BSD destacam-se o FreeBSD, OpenBSD e o NetBSD.

O Solaris é um sistema operacional proprietário derivado do Unix e baseado no BSD. Criado pela Sun, hoje é propriedade da Oracle. Utilizado principalmente por grandes empresas ele é utilizado principalmente em SPARC e x86. Boa parte do seu código fonte era disponibilizada como OpenSolaris enquanto era propriedade da empresa Sun. Após a compra da Sun pela Oracle o OpenSolaris foi descontinuado. A comunidade que tinha interesse em manter o código aberto e dar continuidade, ao que deveria ter sido o OpenSolaris, criou o Projeto OpenIndiana.

Segundo o site da Free Software Foundation (2013) há o desenvolvimento do seu próprio núcleo de sistema operacional, chamado Hurd. Esse projeto visa substituir o Unix. Ainda não possui uma versão estável mas, por ser um projeto de Sistema Operacional livre da Free Software Foundation, tem grande importância no cenário do Software Livre.

O MAC OS X é um sistema operacional de propriedade da Apple, originado do Unix (possui núcleo certificado Unix) e baseado no BSD. É considerado um sistema

operacional estável, seguro e sua interface gráfica Aqua considerada uma das mais bonitas. A união de Núcleo poderoso, interface bonita, estabilidade e segurança tornaram o MAC OS X o sistema originário de Unix mais vendido até hoje. O site da proprietária Apple (2013) destaca a segurança da origem no Unix do sistema operacional:

Todo Mac vem com tecnologias avançadas e poderosas que funcionam juntas organizando, codificando, atualizando e, em última instância, deixando o seu Mac mais seguro. Para os novatos, o OS X foi desenvolvido sob um comprovado e seguro alicerce UNIX. Vem com vários recursos destinados a proteger o seu Mac e as informações contida nele. Com o Gatekeeper, é mais seguro baixar e instalar apps. Já o FileVault protege os dados com criptografia poderosa. Buscar meu Mac ajuda a localizar o seu Mac perdido em um mapa, definir uma senha de maneira remota e muito mais. Configurações de privacidade que podem ser personalizadas ajudam a manter as suas informações pessoais sob controle.

2.4.2 Sistemas Operacionais Móveis

Ao optar por determinado smartphone é preciso se ter mais cuidado com a escolha do sistema operacional do que na escolha de um computador uma vez que a troca de sistema operacional é muito difícil e na maioria das vezes impossível. Conforme Morimoto (2009):

Diferente de um PC, onde você pode escolher qual sistema operacional utilizar, substituindo o Windows por Linux (ou vice-versa), ou mesmo instalando vários sistemas no mesmo micro, nos smartphones a escolhas são mais restritas, já que, com poucas exceções, você não tem como substituir o sistema pré-instalado por outros.

Os Sistemas Operacionais móveis mais utilizados, ou em desenvolvimento atualmente são Android, iOS, Symbian e Windows Phone, no entanto existem outros sistemas importantes.

O Android é um Sistema Operacional móvel com parte do código fonte aberto e alguns recursos com código fonte fechado (sob a Licença Apache) baseado no kernel Linux para dispositivos móveis. Desenvolvido pela Open Handset Alliance

(liderada pelo Google). De acordo com o site da Android (2013) o sistema através da loja de aplicativos Google Play oferece diversas opções ao usuário:

Com o Google Play, você pode descobrir milhões de músicas, assistir os melhores e mais recentes filmes e programas de TV, navegar na maior eBookstore do mundo, e até mesmo folhear suas revistas favoritas. Use o seu telefone ou tablet Android, ou o seu navegador web.

O Symbian é um Sistema operacional móvel de propriedade da Nokia e mantido pela Accenture. Inicialmente um sistema operacional proprietário de código fechado abriu o código fonte em 2008 quando a Nokia comprou os direitos totais sobre o sistema, criando uma fundação neutra: a Symbian Foundation. A versão atual do sistema é o Symbian^3 com atualização Nokia Belle Feature Pack 2 ou Nokia Belle Refresh. Suporta desenvolvimento em C e Java e a gestão de pacotes é .sis, .sisx, .jar e disponível para arquiteturas ARM. A Nokia decidiu não desenvolver mais dispositivos com Symbian em janeiro de 2013, mas dará suporte até pelo menos 2016. A forma de realizar as atualizações gratuitas estão disponíveis no site da Nokia (2013).

O Windows Phone é um Sistema operacional móvel de código fechado de propriedade da Microsoft lançado em outubro de 2010, sucessor do Windows Mobile, porém focado no mercado consumidor ao contrário do antecessor, focado no mercado empresarial (essa característica aliada a interface gráfica "Metro" é uma ruptura total com as versões anteriores do sistema operacional). A interface gráfica Metro não permite personalizações profundas e controla rigorosamente todo o hardware.

Lançado em 2010, o Windows Phone, tinha a promessa de parceria com a HP, o que não se confirmou e em fevereiro de 2011 foi anunciada parceria com a Nokia, que priorizou o Windows Phone em seus smartphones. Em maio de 2011 foi anunciada parceria com Acer, Fujitsu e ZTE.

O iOS é o Sistema operacional de código fechado de propriedade da Apple, desenvolvido para arquitetura ARM e utilizados nos iPhone, Ipod, Apple TV iPad. A empresa proprietária proíbe que o sistema operacional iOS seja utilizado em

hardware de terceiros.

O Ubuntu Phone é um sistema operacional móvel de código aberto patrocinado pela Canonical desenvolvido com apoio da comunidade. O sistema é baseado no kernel Linux e tem o objetivo de criar smartphones com versão de sistema operacional muito semelhante a dos desktops. Atualmente está em fase de testes. Segundo o site do Ubuntu (2013) o sistema operacional deve funcionar satisfatoriamente inclusive em smartphones com hardwares modestos.

O Tizen é um sistema operacional de código aberto, baseado no Linux, desenvolvido pela Linux Foundation com objetivo de ser executado em smartphones, tablets e netbooks. O sistema está em desenvolvimento e conta com apoio de empresas como Intel e Samsung. De acordo com o site da Tizen (2013) há intenção de lançar o sistema comercialmente ainda em 2013.

O BlackBerry OS é um sistema operacional móvel de código fechado de propriedade da empresa canadense BlackBerry (antiga RIM) desenvolvido exclusivamente para os modelos de celulares da própria empresa. A primeira versão foi lançada em janeiro de 1999.

O Bada OS é um Sistema Operacional Móvel de propriedade da Samsung, criado em 2010 marcando o ingresso da companhia no mercado de smartphones como uma resposta as concorrentes Nokia (OS Symbian), RIM e Apple (iPhone OS) que possuíam seus Sistemas Operacionais próprios. Até o ano de 2011 aplicações VOIP eram proibidas no Bada e aplicações como Skype não eram suportadas. Atualmente, de acordo com o site do sistema operacional Bada (2013) a intenção da Samsung (que prioriza a inclusão do sistema Android em seus smartphones) no seu desenvolvimento é ter uma alternativa de baixo custo e não concorrer com os demais sistemas operacionais.

2.5. HTML5, CSS3 E JAVASCRIPT

O Firefox OS é um sistema operacional móvel de código aberto, com núcleo baseado no Linux. Seus aplicativos usam a web como plataforma de desenvolvimento, ou seja, os recursos utilizados são basicamente o Html5, CSS3 e JavaScript.

De acordo com Serra (2011, p. 32) é preciso mais do que o Html5 puro para implementar com boa qualidade uma aplicação web moderna.

O HTML5 por si só, não consegue implementar aplicações web, pois necessita do JavaScript e do CSS, como pode ser visto na analogia presente na Figura 2.11. O HTML é uma linguagem que define os conteúdos, precisando destas tecnologias para adornar e completar as aplicações quanto à apresentação e interação.



Figura 2.11: Analogia funcional do HTML, CSS e JavaScript

2.5.1 Html5

A linguagem Html5, ou a quinta versão da linguagem Html, começou a ser desenvolvida em 2004 pela W3C sendo definida conforme:

HTML é uma abreviação de Hypertext Markup Language - Linguagem de Marcação de Hipertexto. Resumindo em uma frase: o HTML é uma linguagem para publicação de conteúdo (texto, imagem, vídeo, áudio e etc) na Web. O HTML é baseado no conceito de Hipertexto. Hipertexto são conjuntos de elementos – ou nós – ligados por conexões. Estes elementos podem ser palavras, imagens, vídeos, áudio, documentos etc. Estes elementos conectados formam uma grande rede de informação. Eles não

estão conectados linearmente como se fossem textos de um livro, onde um assunto é ligado ao outro seguidamente. A conexão feita em um hipertexto é algo imprevisto que permite a comunicação de dados, organizando conhecimentos e guardando informações relacionadas. Para distribuir informação de uma maneira global, é necessário haver uma linguagem que seja entendida universalmente por diversos meios de acesso. O HTML se propõe a ser esta linguagem (W3C Brasil, 2010).

A partir da implementação do Html5, que traz novas funcionalidades (canvas, web storage, web workers, offline web applications, audio, video, entre outros), os navegadores web estarão livres da necessidade de implantação de plug-ins (como Adobe Flash e Microsoft Silverlight) para exibição de conteúdo multimídia.

Html5 é a mais recente evolução do padrão de linguagem Html de acordo com o site da Mozilla Foundation (2010):

HTML5 é a mais recente evolução do padrão que define o HTML. O termo representa dois diferentes conceitos: E uma nova versão da linguagem HTML, com novos elementos, atributos, e comportamentos e um conjunto maior de tecnologias que permite o desenvolvimento de aplicações e web sites mais diversos e poderosos. Este conjunto é chamado HTML5 & friends e muitas vezes abreviado apenas como HTML5. Embora as especificações do HTML5 ainda não tenham sido finalizadas e estejam sujeitas as mudanças, navegadores como o Mozilla e de outros desenvolvedores começaram a implementar partes deste padrão.

2.5.2 CSS3

A “Cascading Style Sheets” ou CSS é uma linguagem utilizada para aprimorar a formatação e apresentação dos documentos criados em Html. A terceira versão da linguagem (CSS3) apresenta algumas novidades apresentadas por Serra (2011):

O Cascading Style Sheets ou CSS é uma linguagem usada para a apresentação e formatação de um documento HTML. O CSS permite a separação entre os conteúdos dos documentos, definidos por exemplo usando o HTML, e a apresentação deste. Atualmente está a ser desenvolvida a versão 3 pela W3C, que começou a ser implementada em Dezembro de 2005. Algumas das novas funcionalidades do CSS3:

- Suporte de novos módulos de cor como o HSL, CMYK, HSLA e RGBA;
- Bordas gradientes;
- Bordas com imagens;
- Caixas com cantos redondos;
- Caixas com sombreados;

- Múltiplas colunas de texto sem necessitar de tabelas;
- Textos com sombreados.

Tal como o Html5, o CSS é uma plataforma aberta que não necessita de qualquer software ser executado, sendo somente necessário o navegador. Juntamente com o CSS e JavaScript, o Html5 consegue disponibilizar recursos para desenvolver aplicações mais dinâmicas e interativas.

A terceira versão da “Cascading Style Sheets” (CSS3) começou a ser implementada pela W3C em 2005 e é definida como:

O CSS formata a informação entregue pelo HTML. Essa informação pode ser qualquer coisa: imagem, texto, vídeo, áudio ou qualquer outro elemento criado. Grave isso: CSS formata a informação. Essa formatação na maioria das vezes é visual, mas não necessariamente. No CSS Aural, nós manipulamos o áudio entregue ao visitante pelo sistema de leitura de tela. Nós controlamos volume, profundidade, tipo da voz ou em qual das caixas de som a voz sairá. De certa forma você está formatando a informação que está em formato de áudio e que o visitante está consumindo ao entrar no site utilizando um dispositivo ou um sistema de leitura de tela. O CSS prepara essa informação para que ela seja consumida da melhor maneira possível (W3C Brasil, 2013).

O Firefox OS já suporta a versão CSS3 que traz mudanças em relação as versões anteriores conforme a W3C Brasil (2013):

Com as atualizações do CSS3 e com os browsers atualizando o suporte do CSS2.1, nós entramos em patamar onde sem dúvida o CSS é a arma mais poderosa para o designer web. Segue uma pequena lista dos principais pontos que podemos controlar nesse novo patamar:

- selecionar primeiro e último elemento;
- selecionar elementos pares ou ímpares;
- selecionar elementos específicos de um determinado grupo de elementos;
- gradiente em textos e elementos;
- bordas arredondadas;
- sombras em texto e elementos;
- manipulação de opacidade;
- controle de rotação;
- controle de perspectiva;
- animação;
- estruturação independente da posição no código HTML”

2.5.3 JavaScript.

A linguagem JavaScript é uma linguagem multiplataforma há muito tempo utilizada na Web (uma das mais antigas) com objetivo de proporcionar interatividade e dinamismo, estando presente em quase todas as páginas da internet.

A linguagem JavaScript foi criada em 1995 pela Netscape de acordo com Seibt (2013):

JavaScript é uma linguagem de script criada pela Netscape em 1995, tendo sido lançada junto com o navegador Netscape Navigator 2.0. Com seu lançamento, as páginas na internet começaram a ganhar vida, implementando um mínimo de dinamicidade devido ao modo como a linguagem acessa e manipula os componentes do seu ambiente hospedeiro (nesse caso, o Netscape Navigator).

A Mozilla Foundation (2013) define JavaScript como uma linguagem orientada a objetos que segue o padrão ECMAScript:

JavaScript (às vezes chamado apenas de JS) é uma linguagem de programação interpretada e orientada a objetos baseada em protótipos e em first-class functions, mais conhecida como a linguagem de script da Web, mas utilizada em vários outros ambientes como node.js e Couchbase. O padrão JavaScript é o ECMAScript. Desde 2012, todos os navegadores modernos possuem suporte total ao ECMAScript 5.1. Navegadores mais antigos suportam pelo menos ECMAScript 3. Existe uma sexta versão do padrão sendo trabalhada atualmente.

O padrão ECMAScript define a linguagem como multiplataforma, portanto pode ser habilitada em diversos navegadores como Safari, Firefox, Chrome, Chromium, Iceweasel, Opera, Iceape, Seamonkey, entre outros. O padrão ECMAScript é explicado por Seibt (2013, p. 1):

A ECMA (European Computer Manufacturers Association), em sua especificação de número 262, descreve a linguagem ECMAScript como “uma linguagem de programação multiplataforma de propósitos gerais”. Ela foi desenvolvida a partir do JavaScript 1.1 e hoje define o núcleo das linguagens JavaScript e Java, o que, atualmente, é a única semelhança entre ambas.

O JavaScript é uma linguagem de programação geralmente executada no navegador do usuário tornando as páginas funcionais para propósitos específicos sem necessidade de acessar servidores remotos (diferente de programas escritos

em PHP, por exemplo, que precisam ser hospedados em um servidor remoto). Se o recurso JavaScript estiver desabilitado ou ausente, o conteúdo ou funcionalidade da página pode se tornar limitado ou indisponível.

Ser uma linguagem funcional, dinâmica, imperativa e estruturada são as mais importantes características da linguagem JavaScript de acordo com Serra (2011, p. 32):

A utilização do JavaScript na Internet é normalmente efetuada pelo cliente, por exemplo, através da validação dos valores de input nos formulários. Pode também ser usado pelo servidor, para comunicar com a base de dados relacional ou manipular ficheiros no servidor. As características mais importantes do JavaScript são:

- Imperativa e estruturada – suporta a sintaxe da programação estruturada C, o que facilita a implementação;
- Dinâmica – Os tipos de objetos estão associados aos valores e não com as variáveis;
- Funcional – As funções são elas próprias objetos, e podem ser definidas para variáveis, passadas como argumento, e retornadas por outras funções.

Apesar do nome semelhante a linguagem JavaScript não faz parte da plataforma Java. Há diferenças entre JavaScript e Java, conforme disponibilizado no site da Oracle (2013):

a linguagem de programação JavaScript, desenvolvida pela Netscape, Inc., não faz parte da plataforma Java. O JavaScript não cria applets nem aplicações independentes. Em sua forma mais comum hoje, JavaScript reside dentro de documentos HTML e pode fornecer níveis de interatividade com páginas da Web que não podem ser obtidos com HTML simples. As diferenças principais entre Java e JavaScript estão listadas.

- Java é uma linguagem de programação OOP, ao passo que JavaScript é uma linguagem de scripts OOP.
- Java cria aplicações executadas em uma máquina virtual ou em um browser, ao passo que o código JavaScript é executado apenas em um browser.
- O código Java precisa ser compilado, ao passo que os códigos JavaScript estão totalmente em texto.
- Eles requerem plug-ins diferentes.

3 FIREFOX OS

A Mozilla Foundation é a responsável pelo desenvolvimento do sistema operacional Firefox OS. O Manifesto Mozilla, documento com as diretrizes e compromissos da Fundação está disponibilizado na íntegra no Apêndice A.

A história da Mozilla Foundation, que não possui fins lucrativos, iniciada em 1994, é resumida conforme:

A história da Mozilla, fundação criadora do navegador Firefox, começa ainda no ano de 1994, quando o nome "Mozilla" foi utilizado pela primeira vez no Netscape, pois seu apelido era "Mosaic killer" (Mosaic era um navegador da época, que perdeu usuários para o Netscape). A Mozilla com a estrutura que conhecemos hoje começou em 1998, quando a Netscape decidiu liberar publicamente para a comunidade o código-fonte do seu navegador que estava em declínio. O navegador que se tornou o Firefox foi criado em 2002, mas o nome Firefox 1.0 foi usado somente em 2004. (MOZILLA FOUNDATION, 2013).

O Firefox OS é um sistema operacional móvel de código aberto, desenvolvido pela Mozilla Foundation com apoio da comunidade mundial de desenvolvedores. Baseado nos padrões web de Html5, CSS3 e JavaScript é definido pela Mozilla Foundation (2013) no site do Manual do Usuário Firefox OS:

O Firefox OS é um sistema operacional móvel de código aberto, baseado em padrões web, desenvolvido pela Mozilla.

A Mozilla criou uma parceria com inúmeras operadoras e fabricantes de aparelhos no desenvolvimento do Firefox OS para atender a necessidade dos mercados emergentes em 2013.

No Firefox OS todos os aplicativos usados são aplicativos web com a implantação completa do Html5 o que proporciona baixo custo e bom desempenho. Com essa facilidade de utilizar basicamente linguagens já conhecidas dos programadores da web a Mozilla proporciona facilidade para criação de novos aplicativos por membros da comunidade uma vez que não exige o aprendizado de novas linguagens ou tecnologias.

Assim como os melhores Sistemas Operacionais Móveis existentes atualmente o Firefox OS oferece soluções para as principais necessidades dos usuários como e-mail, mensagens de texto (SMS), buscas na internet, mapas, GPS, tocador de

música, rádio FM, câmera, navegador web, Facebook e Twitter.

Novos aplicativos podem ser procurados e instalados no Firefox OS através do Firefox Marketplace, uma ferramenta similar ao Google Play do Android, ao Marketplace do Windows Phone e ao App Store do iOS.

Uma funcionalidade do Firefox OS, não presente nos sistemas concorrentes, é a “Descoberta Dinâmica de Aplicações” que consiste na possibilidade de buscar aplicativos na internet e testá-los previamente sem instalar conforme Manual do Usuário Firefox OS (2013):

Descoberta dinâmica de aplicações - Firefox OS tem um mecanismo dinâmico de busca de aplicações, apenas digite uma palavra como sushi, por exemplo, para descobrir todos os aplicativos relacionados em sua região, organizada por relevância e experiência de uso e ajustado para trazer o que você quer sem digitar.

1. Deslize para direita para encontrar a tela de busca dinâmica de aplicações.
2. Toque na categoria ou digite uma palavra-chave na caixa de busca no topo da tela. aplicativos com conteúdos móveis relevantes aparecem na tela.
3. Toque em um aplicativo para experimentá-lo, sem efetuar download ou instalá-lo.
4. Toque e segure o ícone do aplicativo para instalá-lo na sua tela principal.

A privacidade que é uma das preocupações da Mozilla Foundation presente no navegador Firefox, no Thunderbird e no SeaMonkey, também está presente no Firefox OS. Os dados do usuário são tratados pelo mecanismo “Mozilla Persona” e mantidos em sigilo total conforme divulgado no Manual do Usuário Firefox OS (2013):

O Mozilla Persona (<https://www.persona.org/>) é o mecanismo utilizado para tratar suas informações pessoais. Seus dados ficam em sigilo total, pois o Persona utiliza somente um usuário e senha, e não rastreia sua navegação.

O Firefox Marketplace respeita suas informações. Todos os detalhes podem ser encontrados na Política de Privacidade da Mozilla (<https://marketplace.firefox.com/privacy-policy>).

A privacidade também é garantida através da funcionalidade “Não Me Rastreie”, impedindo que seu comportamento na web seja rastreado pelos sites. O conceito de “rastreamento” e a funcionalidade “Não Me rastreie” são definidas no Manual do Usuário Firefox OS (2013):

Rastreamento é um termo que inclui vários métodos que sites, anunciantes e outros usam para aprender sobre seu comportamento na web. Isso inclui informação sobre quais sites você visita, coisas de que você gosta, não gosta, e compra. Eles normalmente usam esta informação para exibir propagandas, produtos e serviços que correspondem a seu comportamento.

Quando você ativa a função Não me rastreie, o Firefox diz para todos os sites que você visita (assim como aos anunciantes e outros provedores de conteúdo) que você não quer que seu comportamento na web seja rastreado.

3.1 ARQUITETURA

Arquitetura do Firefox OS é dividida em 3 componentes Principais (Gonk, Gecko e Gaia) conforme a Figura 04 e a Figura 05.

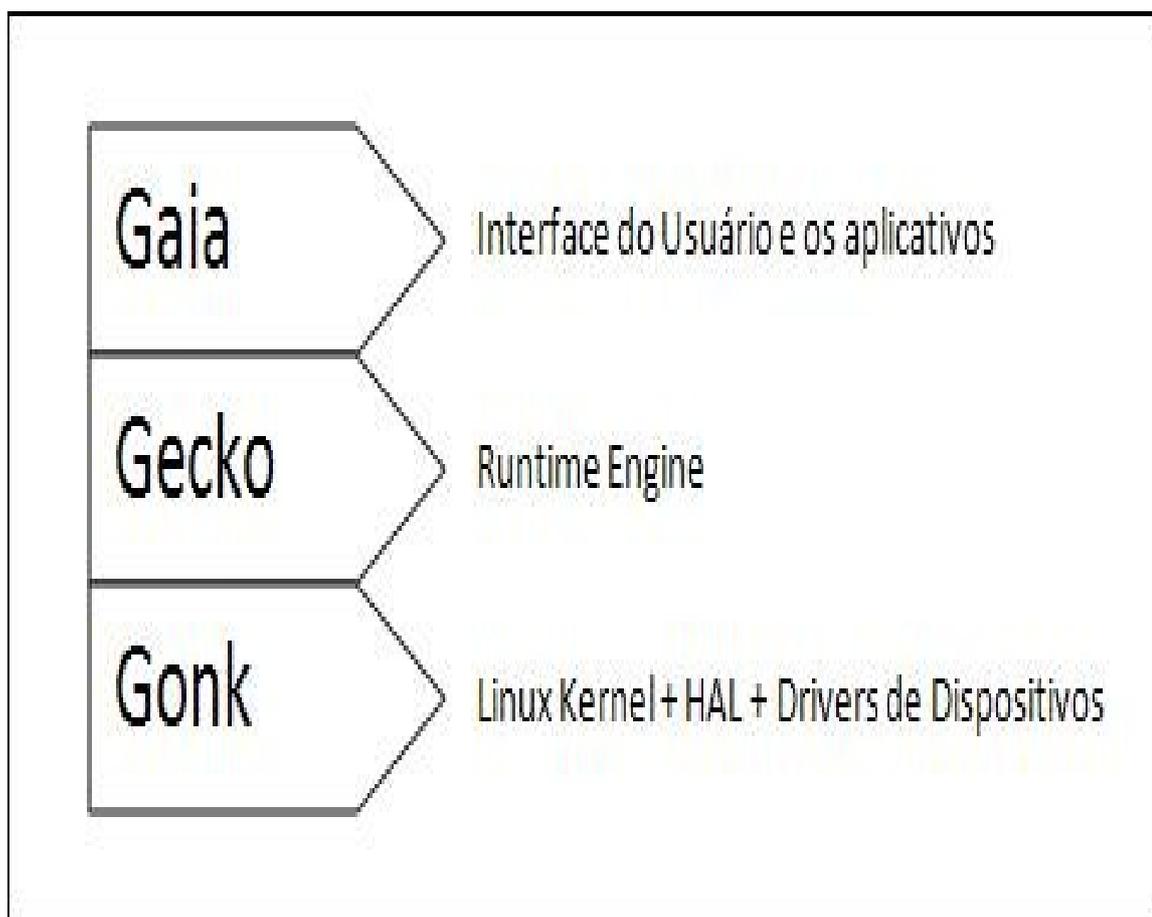


Figura 04: Arquitetura resumida do Firefox OS

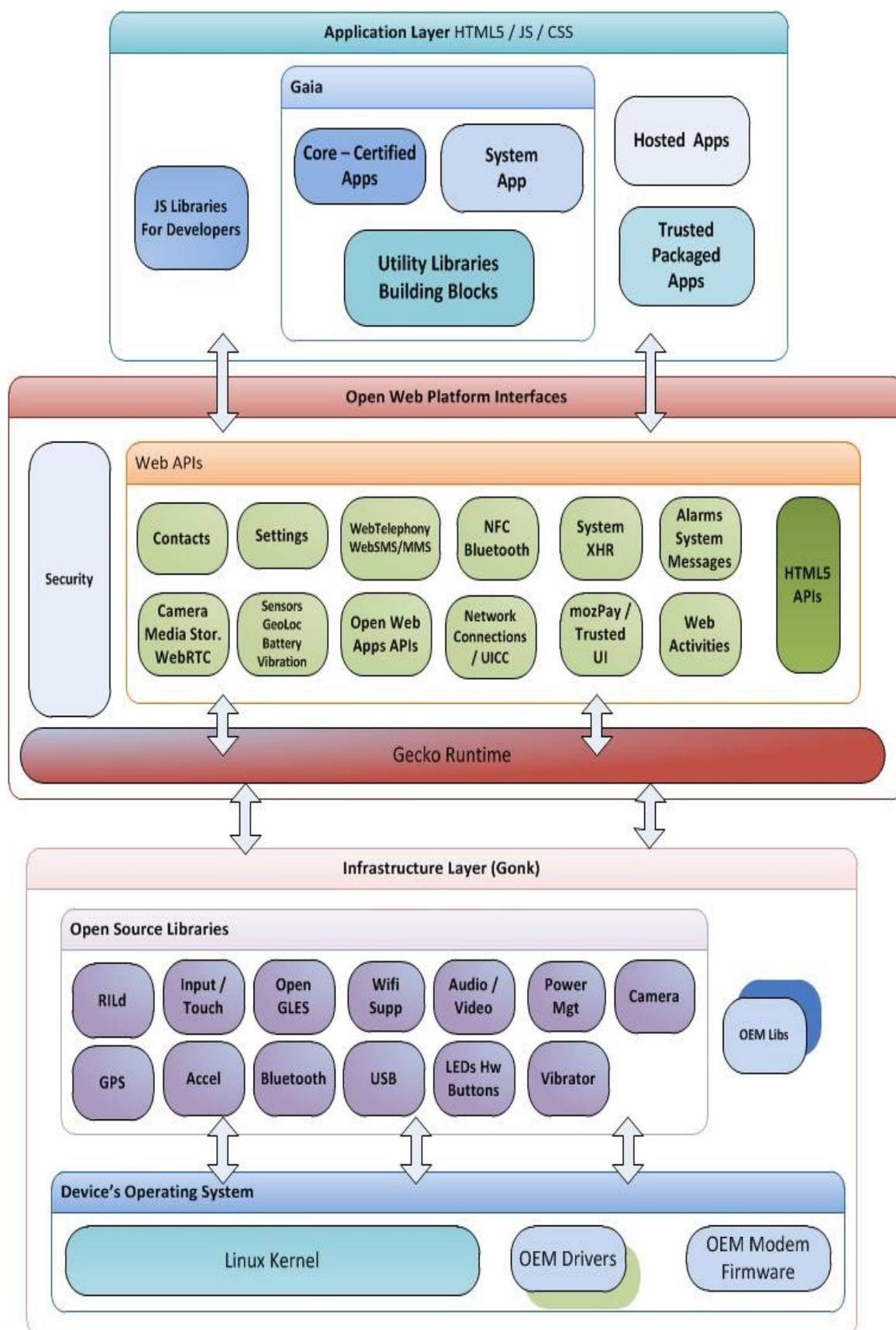


Figura 05: Arquitetura completa do Firefox OS

3.1.1 Gonk

Gonk é a mais baixa camada do Sistema Operacional Firefox OS, consistindo numa distribuição Linux muito simples com alvo no engine runtime, ou motor de layout, Gecko.

O Gonk é composto pelo kernel Linux, sendo basicamente o Sistema Operacional. É definido de acordo com o site do Projeto Mozilla (2013) como:

O Gonk é a camada mais baixa do sistema operacional e consiste em um kernel Linux e o HAL (hardware abstraction layer). O kernel e várias das bibliotecas são ligados a projetos de código aberto conhecidos: Linux, libusb, bluez, dentre outros. Algumas outras partes do HAL são compartilhadas com o projeto Android: GPS, camera, dentre outros. Pode-se dizer que o Gonk é uma distribuição Linux simplificada. O Gonk é uma plataforma alvo (porting target) do Gecko, ou seja, existe o target Gonk assim como existem plataformas de saída para Mac OS X, Windows, e Android. Uma vez que o projeto do Firefox OS tem controle sobre o Gonk, é possível que interfaces baixo nível sejam expostas ao Gecko, por exemplo, o acesso direto para a pilha de serviços de telefonia e também o acesso direto ao frame buffer gráfico. Estas características são novas e não ocorrem no caso de outros sistemas operacionais.

O Gonk é o Sistema Operacional de baixo nível, composto basicamente pelo kernel Linux e pela camada de abstração de hardware HAL (Hardware Abstraction Layer) e os drivers dos dispositivos, com a função de expor o hardware para o engine runtime Gecko.

Camada de abstração de hardware é a camada entre o hardware e o software, tendo a função de eliminar as diferenças de hardware e oferecer uma plataforma uniforme para a execução das aplicações, o que é importante para que o Sistema Operacional possa ser executado em hardwares diferentes com maior facilidade. O HAL conecta os softwares de aplicação ou as linguagens de programação de alto nível aos componentes físicos do smartphone. O HAL descobre, identifica, numera e permite acesso aos dispositivos de hardware. O HAL do Firefox OS tem algumas partes em comum com o Android como a câmera e o GPS

No Firefox OS o Gecko tem controle total do Gonk, diferentemente do acesso apenas parcial que tem aos núcleos dos demais sistemas operacionais. Esse

controle total sobre o núcleo permite o desenvolvimento de recursos para todo o hardware.

3.1.2 Gecko

Gecko é basicamente o engine runtime, ou motor de layout, das aplicações, a plataforma onde são executadas todas as diferentes aplicações web em Html5, CSS3, JavaScript e mostra o conteúdo na tela.

O engine runtime, ou motor de layout, Gecko possui código aberto e é desenvolvido pela Mozilla, sendo utilizado no navegador Firefox e no sistema operacional móvel Firefox OS. É no Gecko que estão implementados os padrões HTML, CSS e JavaScript que permite que as interfaces funcionem satisfatoriamente em diferentes sistemas operacionais.

O engine runtime, ou motor de layout, Gecko é definido conforme o Projeto Mozilla (2013) como:

O runtime web específico para as aplicações do Firefox OS. O Gecko oferece uma infra estrutura de padrões web como HTML5, CSS, SVG, WebGL, JavaScript, dentre outros. Ele também oferece recursos do sistema como device API assim permitindo que aplicações web possam acessar serviços do dispositivo móvel, como por exemplo o GPS e câmera. O Gecko também carrega várias funcionalidades básicas que são importantes para que aplicações web possam rodar corretamente, como a camada de rede e segurança, a camada de gráficos, o motor máquina virtual JavaScript, o engine de layout, dentre outros.

3.1.3 Gaia

Gaia é o nome da interface do usuário, escrito em Html5, CSS3 e JavaScript. Todo

conjunto de aplicativos web são parte do Gaia. No Firefox OS os aplicativos web serão capazes de gerenciar o hardware e todos demais recursos, funcionalidades e aplicativos como telefone, câmera, GPS, calendário, internet, rádio, alarme, etc.

Gaia é a interface gráfica do Sistema Operacional Firefox OS e bastante semelhante ao padrão das interfaces dos sistemas operacionais iOS e Android, que diferem do conceito usado pelo Windows Phone. A interface do Firefox OS é apresentada na Figura 06:



Figura 06: Interface Gráfica do Firefox OS

A interface gráfica Gaia, implementada inteiramente em HTML, CSS e JavaScript é definida pelo Projeto Mozilla (2013) como:

A interface visual do Firefox OS, que apresenta a experiência do sistema ao usuário. A camada Gaia contém várias aplicações padrão como lock screen, home screen, e várias aplicações que são esperadas de um celular tipo smartphone. O Gaia é completamente implementado com padrões web como HTML, CSS, JavaScript, dentre outros. As interfaces entre a camada web e os recursos do sistema operacional são feitas via APIs web — algumas abertas e outras que estão em desenvolvimento na camada Gecko. O Gaia oferece maneiras para a instalação de aplicações de terceiros.

3.2 DESENVOLVIMENTO DE APLICATIVOS

Os aplicativos podem ser desenvolvidos em Html5, CSS3 e JavaScript. O desenvolvedor não precisa aprender a desenvolver em uma API específica.

Conforme Assis (2012, p. 2) o desenvolvimento de um aplicativo é baseado em camadas de informação (Html5), formatação (CSS3) e comportamento (JavaScript) o que permite fazer alterações em camadas independentemente:

O desenvolvimento de páginas e aplicativos web é baseado em um esquema de três camadas principais: informação, formatação e conhecimento. As camadas nos permitem desenvolver independentemente cada área da página, mudando apenas uma das camadas presentes sem que as demais precisem ser alteradas.

A primeira camada, a de informação (Html5), é a mais importante pois marca e dá sentido à informação, que precisa estar disponível independente de local, hora e dispositivo que for realizar a leitura dos dados.

A segunda camada, a de formatação (CSS3) tem a função de controlar o visual, realizando a formatação da informação (áudio, vídeo, imagem, texto, etc) para que o conteúdo seja apresentado de forma agradável independentemente do dispositivo utilizado para o acesso (smartphones, tablets, notebooks, desktops, etc).

Na terceira camada, a de comportamento (JavaScript) é definido o modo de funcionamento do aplicativo. Como a linguagem JavaScript é geralmente utilizada para funcionar no navegador do usuário é necessário que haja um interpretador de JavaScript presente e ativado

3.3 LANÇAMENTO

Os primeiros aparelhos com o Sistema Operacional Firefox OS foram

comercializados no dia 23 de abril de 2013 através do site da Mozilla Foundation e divulgado conforme a notícia publicada abaixo pelo site G1 na Figura 07.



Figura 07: Notícia de lançamento no portal G1

3.4 PRIMEIROS APARELHOS

Os primeiros aparelhos com o Sistema Operacional Firefox OS comercializados para os desenvolvedores foram os Geeksphone Keon e Geeksphone Peak.

O Geeksphone Keon (Figura 08) foi um dos dois primeiros smartphones colocados a venda para testes do Firefox OS. Segundo o site do fabricante Geeksphone (2013) o Geeksphone Keon é definido como:

Um dispositivo leve e versátil para desenvolvedores, mas também muito poderoso, possui todos os recursos de hardware que você precisa para começar a desenvolver aplicativos para o Firefox OS com um ambiente de alto desempenho, sendo também um companheiro ideal para suas tarefas diárias.

Especificações do fabricante:

- CPU Qualcomm Snapdragon S1 7225AB 1Ghz.

- UMTS 2100/1900/900 (3G HSPA).
- GSM 850/900/1800/1900 (2G EDGE).
- Tela 3.5" HVGA Multitouch.
- Câmera 3 MP.
- 4 GB (ROM) e 512 MB (RAM).
- MicroSD, Wifi N, Bluetooth 2.1 EDR, Radio FM, Light & Prox. Sensor, G-Sensor, GPS, MicroUSB.
- Bateria 1580 mAh.



Figura 08: Smartphone Keon com Firefox OS

1. O Geeksphone Peak (Figura 09) foi um dos dois primeiros smartphones colocados a venda para testes do Firefox OS. Segundo o site do fabricante Geeksphone (2013) o Geeksphone Keon é definido como:

Um dispositivo poderoso para desenvolvedores, que introduz características de ponta, com uma tela de 4,3" QHD, a um preço que você jamais imaginou. Para aqueles que gostam de estar um passo a frente, Peak é o seu dispositivo para desenvolvimento.

Especificações do fabricante:

- CPU Qualcomm Snapdragon S4 8225 1.2Ghz x2.
- UMTS 2100/1900/900 (3G HSPA).
- GSM 850/900/1800/1900 (2G EDGE).
- Tela 4.3" qHD IPS Multitouch.

- Câmera 8 MP (back) + 2 MP (front).
- 4 GB (ROM) e 512 MB (RAM).
- MicroSD, Wi-fi N, Bluetooth 2.1 EDR, Radio FM, Light & Prox. Sensor, G-Sensor, GPS, MicroUSB, Flash (câmera).
- Bateria 1800 mAh.



Figura 09: Smartphone Peak com Firefox OS

3.5 PARCEIROS NO DESENVOLVIMENTO

Os principais parceiros da Mozilla Foundation no desenvolvimento do Firefox OS são a operadora de telefonia móvel Telefonica, as fabricantes de smartphones ZTE e Alcatel e a fabricante de processadores Qualcomm. Outras empresas e operadoras também apoiam o Firefox OS conforme lista de parceiros da Figura 10 disponível no site da Mozilla Foundation (2013).



Figura 10: Parceiros da Mozilla Foundation

4 COMPARAÇÃO ENTRE OS SISTEMAS OPERACIONAIS MÓVEIS

Os sistemas operacionais móveis apresentam semelhanças e diferenças. Algumas delas estão relacionadas na Tabela 1:

	Firefox OS	Android	Symbian	Windows Phone	iOS	Ubuntu Phone	Tizen	Blackberry OS	Bada OS
Código Fonte	Aberto	Aberto com partes fechadas	Inicialmente fechado e posteriormente aberto	Fechado	Fechado	Aberto	Aberto	Fechado	Fechado
Desenvolvedor	Mozilla	Open Handset Alliance e Google	Nokia e Accenture	Microsoft	Apple	Canonical	Linux Foundation, Samsung e Intel	Blackberry	Samsung
Arquiteturas	ARM	ARM, MIPS, x86	ARM	ARM	ARM	ARM, x86	ARM, x86	ARM	ARM
Engine Runtime	Gecko	WebKit	WebKit	Trident	WebKit	QtWebKit	WebKit	WebKit	WebKit
Família do SO	Linux	Linux	Mobile OS	Windows CE	Darwin	Linux	Linux	Mobile OS	Linux
Loja Oficial de Apps	Marketplace	Google Play	Symbian Horizon, Nokia store	Windows Phone Store	App Store	Ubuntu Software Center	Tizen Store	App World	Samsung Apps
Programado em	Html5, C++, CSS3, JavaScript	C, C++, Java	C++	C++	C, C++, Objective-C	C, C++, QML, HTML5	C, C++, HTML5	Java	C++
Aplicativos padrão	Navegador internet, e-mail, contatos, SMS, calendário, GPS, câmera, telefone, bluetooth, wifi	Navegador internet, e-mail, contatos, SMS, calendário, GPS, câmera, telefone, bluetooth, wifi	Navegador internet, e-mail, contatos, SMS, calendário, GPS, câmera, telefone, bluetooth, wifi	Navegador internet, e-mail, contatos, SMS, calendário, GPS, câmera, telefone, bluetooth, wifi	Navegador internet, e-mail, contatos, SMS, calendário, GPS, câmera, telefone, bluetooth, wifi	Navegador internet, e-mail, contatos, SMS, calendário, GPS, câmera, telefone, bluetooth, wifi	Navegador internet, e-mail, contatos, SMS, calendário, GPS, câmera, telefone, bluetooth, wifi	Navegador internet, e-mail, contatos, SMS, calendário, GPS, câmera, telefone, bluetooth, wifi	Navegador internet, e-mail, contatos, SMS, calendário, GPS, câmera, telefone, bluetooth, wifi

Tabela 01: Comparação entre os sistemas operacionais móveis

5 CONCLUSÕES

O trabalho teve o objetivo de analisar as funcionalidades do Firefox OS em relação aos principais concorrentes, comparando as principais características, as vantagens e as desvantagens entre os sistemas operacionais pesquisados

Quanto a análise das características dos sistemas operacionais móveis a que mais diferencia o Firefox OS dos concorrentes é a de possuir código totalmente aberto, por conta disso o seu desenvolvimento conta com apoio de desenvolvedores voluntários, permitindo maior participação da comunidade no desenvolvimento de todos os seus recursos, diferente de sistemas operacionais proprietários como Blackberry OS, Windows Phone e iOS nos quais o código fonte é restrito à empresa mantenedora. O Android apenas disponibiliza parte do código fonte.

Firefox OS tem como desafio (para poder enfrentar os líderes de mercado) oferecer, junto com seus parceiros, hardwares similares com melhor desempenho e maior facilidade de criação e obtenção de aplicativos. Como concorrentes fortes na linha de Sistema Operacional de código aberto o Firefox OS encontrará o Tizen e o Ubuntu Phone que também incentivam a participação de voluntários do mundo inteiro. Os maiores concorrentes do Firefox no entanto serão o iOS da Apple de código fechado e o Android da Google com seu código aberto, porém com partes fechadas e com muitas restrições por parte da empresa.

Quanto a análise das vantagens e desvantagens destaca-se que a principal vantagem do Firefox OS em relação aos seus concorrentes está na grande comunidade de desenvolvedores e no fato de todos os seus aplicativos serem baseados em HTML5, CSS3 e JavaScript, o que torna desnecessário novos conhecimentos para quem já desenvolve aplicativos para web. Já a principal desvantagem está em entrar num mercado já dominado por sistemas operacionais fortes como Android, iOS e Windows Phone.

6 REFERÊNCIAS

ANDROID. **Android**. Disponível em: <<http://www.android.com>>, acessado em 05 de outubro de 2013.

APPLE. **Tudo que você precisa saber sobre OS X**. Disponível em: <<http://www.apple.com/br/osx/what-is>>, acessado em 05 de outubro de 2013.

ASSIS, André Fernando de. **Desenvolvimento e Validação de Uma Rede Social para Universitários Atena**. Joaçaba: Unoesc. 2012. 20p.

BADA. **What is Bada**. Disponível em: <<http://www.bada.com/whatisbada/index.html>>. acessado em 24 de setembro de 2013.

DEBIAN. **A Definição Debian de Software Livre (DFSG)**. Disponível em: <http://www.debian.org/social_contract#guidelines>. acessado em 15 de maio de 2013.

DEBIAN. **Debian**. Disponível em: <<http://www.debian.org>>. acessado em 15 de maio de 2013.

DISTROWATCH. **Distrowatch.com**. Disponível em: <<http://www.distrowatch.com>>, acessado em 05 de outubro de 2013.

FERREIRA, Felipe de Casto. **Módulo de: Sistemas Operacionais**. Vila Velha: ESAB, 2008.

FREEBSD. **FreeBSD**. Disponível em: <<http://www.freebsd.org>>, acessado em 05 de outubro de 2013.

FREE SOFTWARE FOUNDATION. **O que é o software livre?**. Disponível em: <<http://www.gnu.org/philosophy/free-sw.pt-br.html>>. acessado em 15 de maio de 2013.

GEEKSPHONE, **Geekspphone**. Disponível em: <http://www.geekspphone.com>. acessado em 15 de maio de 2013.

G1. Smartphone com sistema Firefox começa a ser vendido pela internet. Disponível em: <<http://g1.globo.com/tecnologia/noticia/2013/04/smartphone-com/sistema-firefox-com-eca-ser-vendido-pela-internet.html>>. acessado em 15 de maio de 2013.

MORIMOTO, C E. **Smartphones: Guia Prático**. Porto Alegre: Sulina, 2009.

MOZILLA FOUNDATION. **Firefox OS**. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Mozilla/Firefox_OS>, acessado em 19 de janeiro de 2013.

MOZILLA FOUNDATION. **Manifesto Mozilla**. Disponível em: <<http://www.mozilla.org/pt-BR/about/manifesto/>>, acessado em 24 de janeiro de 2013.

MOZILLA FOUNDATION. **Manual do Usuário Firefox OS**. Disponível em: <<https://support.mozilla.org/pt-BR/kb/manual-do-usuario-firefox-os>>. acessado em 23 de setembro de 2013.

NOKIA. **Atualização de Software para Nokia Belle e Symbian**. Disponível em: <<http://www.nokia.com/br-pt/suporte/nokia-belle-symbian-software-update>>. acessado em 24 de setembro de 2013.

ORACLE. **Qual é a diferença entre o JavaScript e o Java?** Disponível em: <www.java.com/pt_BR/download/faq/java_javascript.xml>, acessado em 19 de janeiro de 2013.

SEIBT, Diego Leonardo. **Artigo: JavaScript**. Disponível em: <<https://fit.faccat.br/~leonardoseibt/ArtigoJavaScript.pdf>>, Taquara: Faculdades de Informática de Taquara. acessado em 24 de setembro de 2013.

SERRA, Ricardo Jorge. **Mestrado Integrado em Engenharia Informática e Computação : Interfaces tácteis baseadas em HTML5/CSS3/JavaScript**. Porto: Faculdade de Engenharia da Universidade do Porto. 2011. 89p.

TIZEN. **Tizen**. Disponível em <<http://www.tizen.org>>. acessado em 24 de setembro de 2013.

UBUNTU. **Ubuntu Phone**. Disponível em <<http://www.ubuntu.com/phone>>. acessado em 24 de setembro de 2013.

W3C BRASIL. **Curso W3C Escritório Brasil: CSS3**, São Paulo, 2013, 62p

W3C BRASIL. **Curso W3C Escritório Brasil: HTML5** , São Paulo, 2010, 106p.

WIKIPEDIA. **História do Software Livre.**
<http://pt.wikipedia.org/wiki/História_do_software_livre>, acessado em 19 de janeiro de 2013.

APÊNDICES

APÊNDICE A – Manifesto Mozilla

MANIFESTO MOZILLA

1. INTRODUÇÃO

A internet está se tornando uma parte cada vez mais importante de nossas vidas.

O projeto Mozilla traduz-se numa comunidade global de pessoas que acreditam que a abertura da rede, inovação e oportunidade são elementos chave para a continuidade de uma Internet saudável. Nós temos trabalhado conjuntamente desde 1998 para assegurar que a Internet seja desenvolvida de forma a beneficiar a todos. Somos melhor conhecidos por ter criado o navegador Mozilla Firefox.

O projeto Mozilla baseia-se numa perspectiva que tem na comunidade a fonte de criação de software de código aberto e de desenvolvimento de novas formas de atividades colaborativas. Criamos comunidades de pessoas envolvidas em fazer a experiência de uso da Internet melhor para todos nós.

Como resultado destes esforços, destilamos uma série de princípios que acreditamos sejam críticos para que a Internet continue beneficiando tanto o bem público e comum, como os aspectos comerciais da vida. Estes princípios estão listados abaixo.

Os objetivos do Manifesto são:

1. Articular uma visão sobre a Internet que reflita o que os participantes da Mozilla desejam que a Mozilla Foundation siga;
2. Falar com as pessoas independentemente da profundidade de seu

- conhecimento técnico;
- 3. Fazer com que os colaboradores do Mozilla sintam-se orgulhosos do que estamos fazendo e nos motivem a continuar; e
- 4. Fornecer um arcabouço para que outras pessoas propaguem esta visão sobre a Internet.

Estes princípios não se tornarão realidade por si próprios. Pessoas são necessárias para fazer com que a Internet mantenha-se aberta e participativa – pessoas atuando como indivíduos, trabalhando em grupos e liderando outras pessoas. A Mozilla Foundation tem um compromisso com o avanço dos princípios estabelecidos neste Manifesto Mozilla. Convidamos a todos para nos acompanhar na tarefa de tornar a Internet um lugar cada vez melhor para todo mundo.

2. PRINCÍPIOS

1. A Internet é parte integral da vida moderna – sendo um componente chave para a educação, comunicação, colaboração, negócios, entretenimento e para a sociedade como um todo;
2. A Internet é um recurso público global que deve permanecer aberto e acessível;
3. A Internet deve enriquecer a vida dos seres humanos como indivíduos;
4. A segurança dos indivíduos na Internet é fundamental e não pode ser tratada como opcional;
5. Os indivíduos devem ter a habilidade de moldar suas próprias experiências na Internet.
6. A efetividade da Internet como um recurso público depende de interoperabilidade (protocolos, formato de dados e conteúdo), inovação e participação descentralizada mundialmente.
7. Software livre e aberto promovem o desenvolvimento da Internet como um bem público.

8. Processos transparentes e baseados em comunidades promovem participação, responsabilidade e confiança.
9. Envolvimento comercial com o desenvolvimento da Internet traz vários benefícios; um equilíbrio entre os objetivos comerciais e os benefícios ao público é essencial.
10. Ampliar o benefício público promovido pela Internet é um importante objetivo ao qual vale dedicar tempo, atenção e compromisso.

3. IMPLEMENTANDO O MANIFESTO MOZILLA

Existem diversas formas de implementar os princípios do Manifesto Mozilla. Consideramos bem-vinda uma ampla gama de atividades e prevemos o mesmo tipo de criatividade que os participantes do Mozilla apresentaram em outras áreas do projeto. Para indivíduos não profundamente envolvidos com o projeto Mozilla, uma forma básica e muito efetiva de apoiar o Manifesto é usar o Mozilla Firefox e outros produtos que incorporam os princípios do Manifesto.

4. COMPROMISSO DA MOZILLA FOUNDATION

A Mozilla Foundation compromete-se a manter e empregar o Manifesto Mozilla em suas atividades. Especificamente, comprometemo-nos a:

- construir e permitir o desenvolvimento de tecnologias abertas e comunidades que apoiam os princípios do Manifesto;
- construir e distribuir aos consumidores bons produtos que apoiem os princípios do Manifesto;

- utilizar os bens gerados pelo Mozilla (propriedade intelectual como por exemplo direitos autorais e marcas; infra-estrutura, recursos financeiros e reputação) para manter a Internet como uma plataforma livre;
- promover modelos que criem valor econômico para o benefício público, e
- promover os princípios do Manifesto Mozilla em nosso discurso público e com a indústria da Internet.

Algumas atividades da Fundação – atualmente vinculadas à criação, distribuição e promoção de produtos ao consumidor – são conduzidas, primordialmente, por meio da Corporação Mozilla, subsidiária controlada pela Mozilla Foundation.

5. CONVITE

A Mozilla Foundation convida a todos que concordam e suportam os princípios do Manifesto Mozilla a juntarem-se a nós e encontrar novas formas para tornar esta visão sobre a Internet uma realidade.

(Disponível no site <http://www.mozilla.org/pt-BR/about/manifesto>, conforme visita em 24 de setembro de 2013)