

UNIDADE 4 – IMPLEMENTAÇÕES DE SISTEMAS OPERACIONAIS

MÓDULO 1 – SISTEMA OPERACIONAL WINDOWS

01

1 - HISTÓRICO DO MICROSOFT WINDOWS

O Windows é um Sistema Operacional proprietário lançado pela empresa Microsoft. Ao longo das mais de duas décadas de sua existência, foram lançadas mais de uma dezena de versões do Sistema Operacional e incorporadas uma série de funcionalidades que acabaram por se tornar padrão de mercado. Dentre estas contribuições, uma que se destaca é a popularização do uso do dispositivo apontador, ou mouse, e as constantes pesquisas na melhoria da interface gráfica e da usabilidade do Sistema.

Inicialmente proposto para equipar estações de trabalho, durante boa parte do seu ciclo de vida o Windows foi sendo segmentado para atingir diferentes nichos de mercado, como o setor corporativo e a computação móvel. Curiosamente, nos últimos anos esta política tem-se modificado diametralmente, tendo a empresa mantenedora iniciado uma diretriz de integrar cada vez mais as diferentes plataformas.

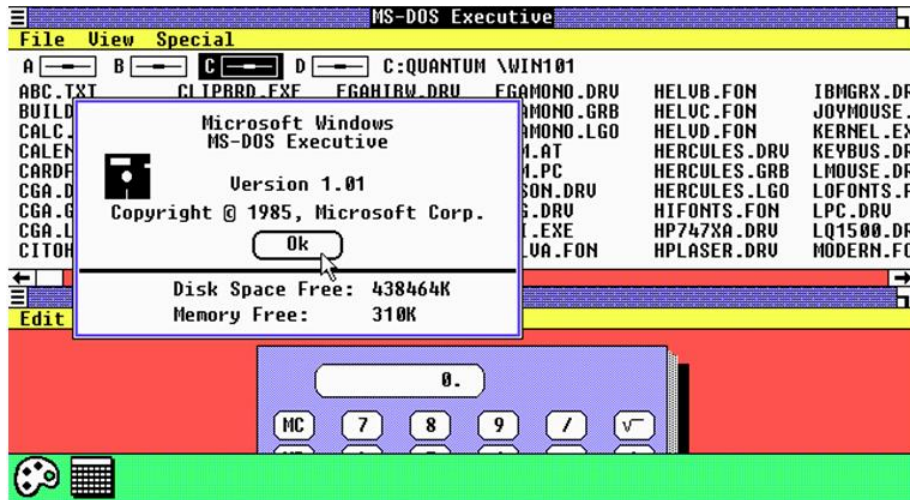
A constante pesquisa tecnológica, a busca pela inovação e o foco na usabilidade fez do Microsoft Windows o Sistema Operacional mais utilizado em estações de trabalho em todo o mundo, com uma larga vantagem em relação aos seus principais concorrentes – as distribuições Linux e o sistema MAC OS.

No decorrer do nosso estudo será apresentado um pouco do histórico do **Sistema Operacional Windows**, com destaque para as principais funcionalidades e inovações propostas no lançamento de cada nova versão. Além disso, serão abordados detalhes do funcionamento de sua arquitetura interna e dos seus principais componentes.

**02**

- **Década de 1980**

A primeira versão do Windows foi lançada em 1985, como a primeira tentativa real da Microsoft em permitir ao usuário utilizar uma interface gráfica sobre o sistema operacional MS DOS. Uma das principais novidades foi a forte necessidade de utilização do mouse para operação da interface em um momento em que este dispositivo de E/S ainda era uma inovação no mercado de TI. Abaixo é possível visualizar a interface gráfica do Windows 1.01.



A versão 2 do Windows não apresentou grandes mudanças em relação à anterior, com exceção da inclusão das funcionalidades de minimizar e maximizar, que permitiam a sobreposição entre as janelas dos aplicativos, e da criação de um painel de controle para centralizar as configurações do sistema, elementos ainda presentes nas versões mais recentes do SO. A criação desta versão do Windows foi um dos fatores responsáveis por tornar a Microsoft a maior empresa vendedora de *software* no ano de 1988.

Até esta versão, era possível executar o Windows apenas com a utilização de disquetes, ou seja, sem a necessidade de se ter um disco rígido. O SO foi desenhado desta forma em função de uma realidade de momento, em que as tecnologias de armazenamento de disco ainda eram recentes e o custo associado a aquisição deste componente de hardware era alto.

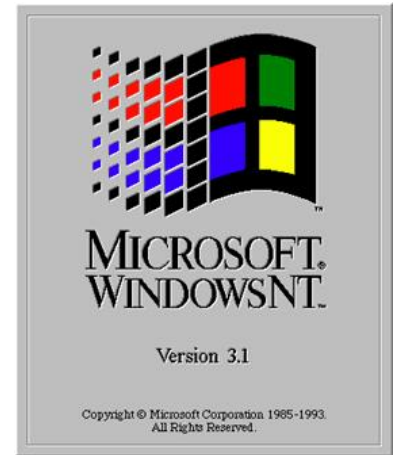
03

• Década de 1990

A primeira distribuição do Windows a requerer obrigatoriamente a presença de um HD foi a 3.0, lançada no ano de 1990. A instalação completa do *software* ocupava entre 10 MB e 14 MB de armazenamento em disco, além de requerer pelo menos 1 MB de espaço em Memória principal para a correta execução. Além destes requisitos, o Windows 3 trouxe uma série de outras inovações, como a habilidade de executar programas originalmente escritos para o MS DOS, permitindo, desta forma, a aplicação do conceito de multitarefa para estes aplicativos legados.

A partir da sua versão 3.1, o Windows foi dividido em duas linhas distintas de desenvolvimento, uma voltada para o usuário doméstico, que manteve o nome original, e outra com foco no meio corporativo, denominada **Windows NT 3.1**. Apesar de aparentemente ambas as versões manterem a mesma interface gráfica, o código interno do Windows NT foi baseado em um conceito completamente novo que tinha como principais objetivos a portabilidade para múltiplas plataformas, o aumento da segurança e o incremento da escalabilidade em relação às versões que eram executadas sobre o DOS. Saiba+

O Windows NT 3.1 foi a primeira versão do Windows desenvolvida em 32 bits. O seu lançamento ocorreu trazendo duas diferentes configurações, o Windows NT propriamente dito, voltado aos usuários corporativos que trabalhavam em rede e que necessitavam consumir serviços corporativos, e o Windows NT Advanced Server, com foco na instalação em servidores e voltado ao provimento dos serviços corporativos.



Saiba+

Esta estratégia utilizada pela Microsoft auxiliou no aumento da liderança da companhia no mercado de *software*. As vendas somadas das versões 3 e 3.1 do Windows alcançaram marca superior a dez milhões de cópias em pouco mais de dois anos, isto em uma época onde a quantidade de computadores instalados em empresas e residências era incomparavelmente inferior aos dias atuais.

04

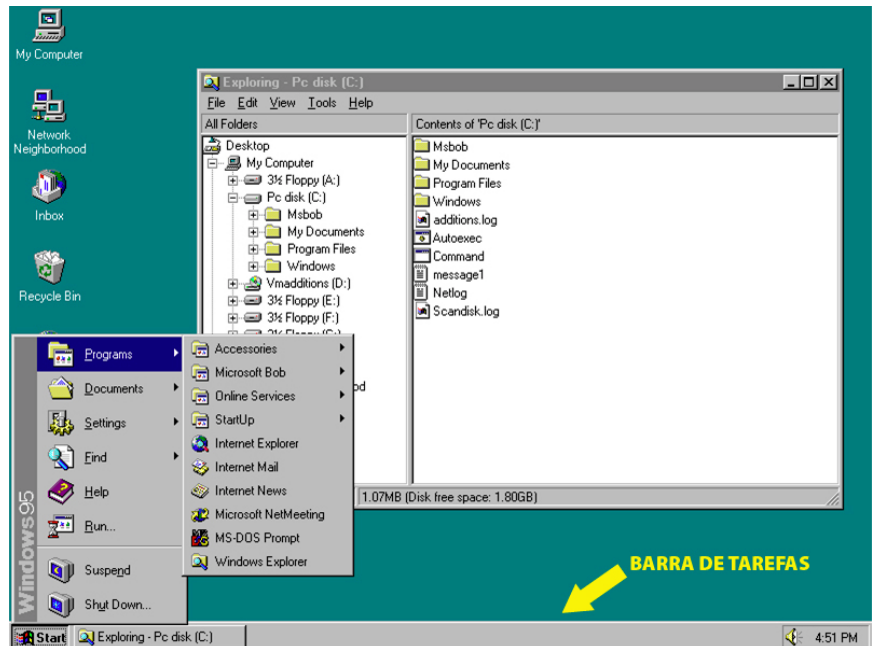
Um ano depois do seu último lançamento, em 1993, a Microsoft pôs no mercado a versão 3.5 do Windows NT, tendo como principal meta melhorar o desempenho geral do Sistema Operacional. Assim como na versão anterior, o *software* foi dividido em duas diferentes versões:

- o Windows NT Workstation, em substituição ao Windows NT,
- o Windows NT Server, que ocupou o lugar do Windows NT Advanced Server.

Esta nova versão trouxe uma série de melhorias em elementos de rede e na estruturação do sistema de arquivos. Já versão subsequente do Windows NT, numerada como 3.51, foi publicada com duas grandes alterações em relação à anterior.

A partir de 1995, a lógica de numeração das versões do SO Windows voltadas aos usuários domésticos foi modificada, foi deixado de lado o modelo sequencial numérico e foi adotada a utilização do ano de lançamento do SO, como parte de uma estratégia de *marketing*.

A primeira versão posta no mercado sob este novo método foi o **Windows 95**, lançado em agosto do mesmo ano e responsável por trazer alguns dos conceitos que fizeram com que a marca se tornasse tão forte entre os usuários de computadores. Além de introduzir uma interface gráfica completamente nova e o conceito de 32 bits para usuários domésticos, o SO foi desenvolvido com foco na realização de múltiplas tarefas pelo usuário, elemento materializado através da barra de tarefas e na possibilidade de alternância entre as janelas. Além disto, o Windows 95 também introduziu no mundo Microsoft o conceito de *Plug-and-play*. Na imagem é possível visualizar a interface gráfica proposta pelo Microsoft Windows 95.



Alterações

As alterações foram a possibilidade de executar o SO na arquitetura PowerPC, e o provimento de interoperabilidade cliente/servidor com a versão Windows 95, que estava próxima de ser lançada.

Plug-and-play

Significa “ligar e usar”. Essa tecnologia foi criada com o objetivo de fazer com que o computador reconheça e configure automaticamente qualquer dispositivo que seja instalado, facilitando a expansão segura dos computadores e eliminando a configuração manual.

05

A versão seguinte do Sistema Operacional, voltado para o usuário doméstico, lançada pela Microsoft, foi o **Windows 98**.

Esta versão teve como uma das principais características a integração do visualizador de arquivos Windows Explorer com o navegador Internet Explorer. Outra importante mudança foi a introdução do “Windows Drive Model”, modelo que previa a construção de um único driver de componente que desse suporte a todas as versões futuras do Windows.



No mundo corporativo, o Windows NT 4.0, sucessor do NT 3.51, foi publicado em 1996 e introduziu a nova interface do Windows 95 na linha NT. Em comparação com a versão anterior, pode-se destacar a inclusão das bibliotecas de criptografia CryptoAPI e a introdução do conceito de Políticas do Sistema.

06

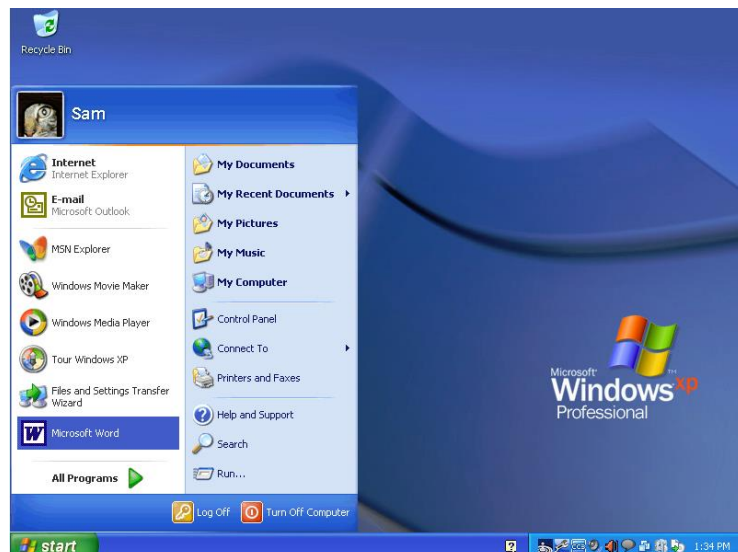
• Década de 2000

O ano 2000 foi marcado por dois lançamentos praticamente simultâneos, o Windows Millenium Edition (ME), voltado para usuários domésticos, e o Windows 2000, com foco corporativo. O Windows ME, última versão do Windows baseada no MS-DOS, incluiu uma série de novos conceitos para os usuários, com ênfase na experiência multimídia e na possibilidade de restauração do sistema. Já o Windows 2000 acrescentou grandes melhorias em itens como confiabilidade, facilidade de uso, suporte para computação móvel e compatibilidade com a Internet.

O Windows ME ficou marcado na história como um dos piores sistemas operacionais da linha Windows, sobretudo em função da grande quantidade de falhas encontradas no sistema. O *software* chegou a ser, inclusive, listado como um dos piores produtos tecnológicos de todos os tempos pela conceituada revista PC World. Esta realidade fez com que muitos usuários mantivessem a versão anterior do Windows em seus computadores, evitando realizar a atualização do SO.

A situação viria a mudar diametralmente em 2001, com o lançamento do **Windows XP**, uma das mais rápidas e estáveis versões do Sistema Operacional da Microsoft.

Diferentemente do seu antecessor, o Windows XP teve a sua arquitetura baseada na versão do Windows 2000 sem, no entanto, deixar de trazer os elementos de interface e a facilidade de uso presente no Windows ME. Saiba+



PC World

Para saber mais, acesse http://www.pcworld.com/article/125772/worst_products_ever.html?page=2

Saiba+

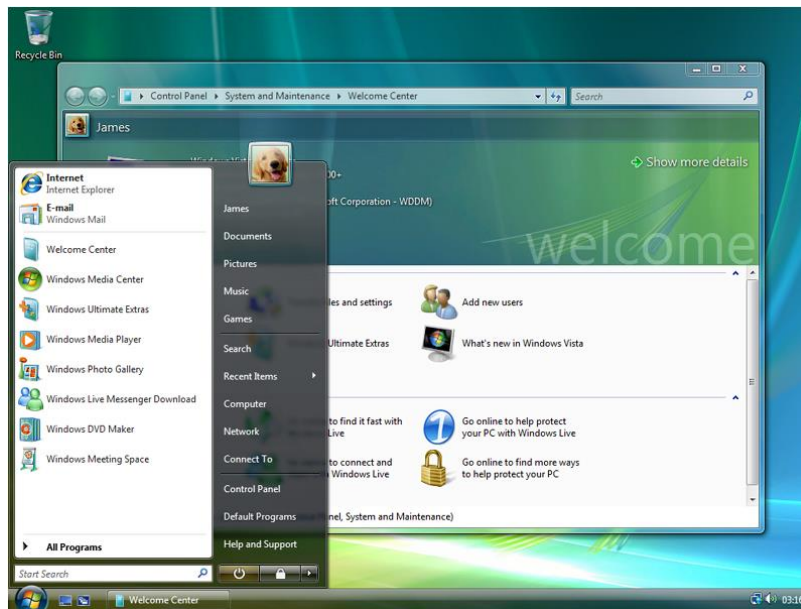
Durante o seu período de operação, o Sistema recebeu uma série de elogios de críticos, sobretudo por sua estabilidade, herdada da arquitetura NT, e pelo acerto no redesenho da interface gráfica, que a tornou a mais fácil de operar e mais intuitiva ao usuário. Estima-se que quase um bilhão de cópias foram vendidas ao longo dos 13 anos do ciclo de vida deste SO.

07

Com a proliferação do Windows XP, percebeu-se um considerável aumento dos ataques ao Sistema Operacional da Microsoft. Foi neste contexto, de foco em segurança, que foi lançado o **Windows Vista**, no ano de 2006.

Além de incrementar os itens de segurança, esta versão também se caracterizou pela completa mudança no visual do Sistema Operacional, com utilização de barras e bordas de janelas translúcidas, renderização do desktop em 3D e a utilização de gadgets na área de trabalho.

O SO trouxe, ainda, um redesenho de funções do sistema, como os serviços de rede, áudio e impressão.

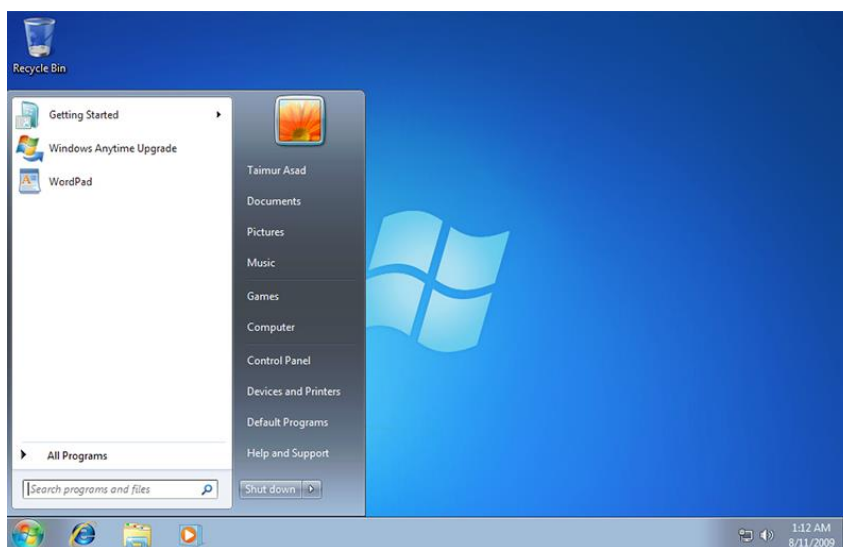
**Gadget**

No contexto dos Sistemas Operacionais, denomina-se gadget algum pequeno *software*, módulo ou ferramenta que é agregado ao ambiente da área de trabalho, como, por exemplo, a apresentação de uma janela com um elemento agregador de e-mails visível diretamente na área de trabalho.

A primeira década dos anos 2000 ainda apresentou mais um lançamento do Sistema Operacional Windows para usuários finais, a versão **Windows 7**, desenvolvida para o mundo sem fios dos dias atuais. Além disso, o lançamento deste SO trouxe de volta o antigo método para versionamento dos Sistemas Operacionais Windows, abandonando a utilização do ano de lançamento como identificador da versão.

Muitos acreditam que a publicação desta versão serviu para corrigir uma série de problemas apresentados pelo Vista que, assim como o Windows ME, não conseguiu se estabelecer no mercado. O Windows 7 se apresentou como um sistema operacional ágil, estável e de fácil utilização, se mostrando como o verdadeiro sucessor do Windows XP.

Dentre as inovações da versão, pode-se destacar a estreia do Windows Touch, elemento que permite aos usuários navegar na web, navegar por fotos ou arquivos através de uma *telatouchscreen*, além de três novas formas de se manipular as janelas, chamadas de Snap, Peek e Shake, que auxiliaram sobremaneira na organização da área de trabalho.



Para descobrir algumas das principais inovações do Windows 7, como as funções Snap e Peek, acesse o vídeo disponível em: <https://www.youtube.com/watch?v=Z2ncZH378n4>.

Em relação à linha voltada aos servidores corporativos, foram lançadas duas versões do Windows, o Server 2003 e o Server 2008.

Server 2003

O Windows Server 2003 foi basicamente uma versão melhorada do Windows XP, só que voltada para o ambiente dos Centros de Processamento de Dados. As principais vantagens em comparação ao seu antecessor se relacionaram a uma melhor performance e uma maior escalabilidade. A versão trouxe, ainda, melhoria nos serviços de provimento de páginas web (IIS), no gerenciamento do diretório de Usuários (Active Directory), na administração das políticas de grupo e no sistema de backup.

Server 2008

O Windows 2008 teve como base o código do Windows Vista sem, no entanto, carregar os inúmeros problemas apresentados pela versão de usuário final. Como principais novidades, destacou-se a reescrita da pilha dos protocolos de rede, melhoria do processo de instalação e recuperação e implementação de novas diretrizes de segurança, como o aprimoramento do Firewall do Windows.

09

• Década de 2010

A versão 8 do Windows foi a primeira produzida na década de 2010, tendo seu lançamento ocorrido no ano de 2012. Esta versão representou mais uma quebra de paradigma em relação ao desenho da interface de usuário do Sistema Operacional, já que removeu o botão iniciar da área de trabalho, o substituindo por uma tela de início mais adequada ao modelo *touch screen*.

Esta iniciativa teve como principal objetivo iniciar a integração das plataformas desenvolvidas pela Microsoft, tornando mais transparente e intuitivo para os usuários navegar entre as versões do Sistema Operacional instaladas em dispositivos de arquitetura diferente, como computadores pessoais, smartphones e tablets.

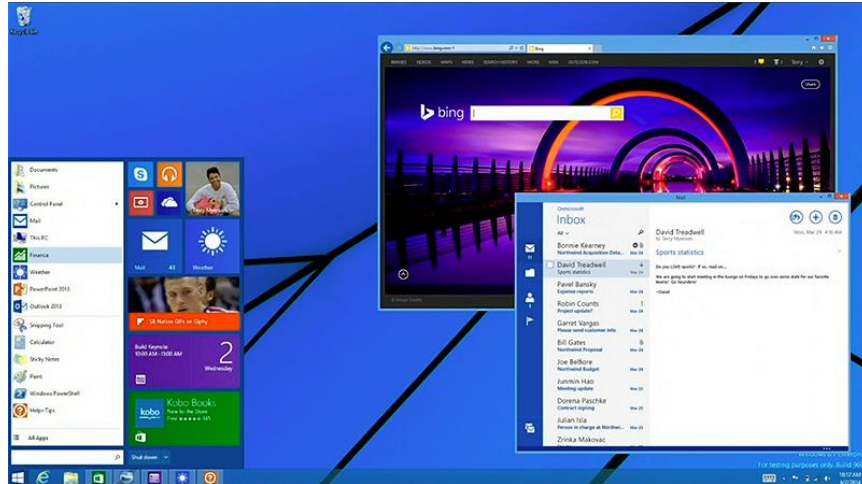
A imagem abaixo apresenta a nova área de trabalho proposta pelo Windows 8, com a utilização de caixas de conteúdo e possibilidade de navegar facilmente pelas diferentes telas através da utilização do *touch screen*, recurso presente não apenas em dispositivos móveis, mas também em alguns modelos de notebook. A imagem a seguir apresenta a tela inicial do Windows 8.



10

Ao contrário do que previa a Microsoft, a mudança da forma de operação da interface não agradou vários dos usuários do Windows, já acostumados com a utilização da barra de tarefas e do botão iniciar na tela inicial da área de trabalho.

Esta realidade forçou a empresa a disponibilizar a versão subsequente do Windows, a 8.1, com a reinserção do botão e do menu Iniciar.

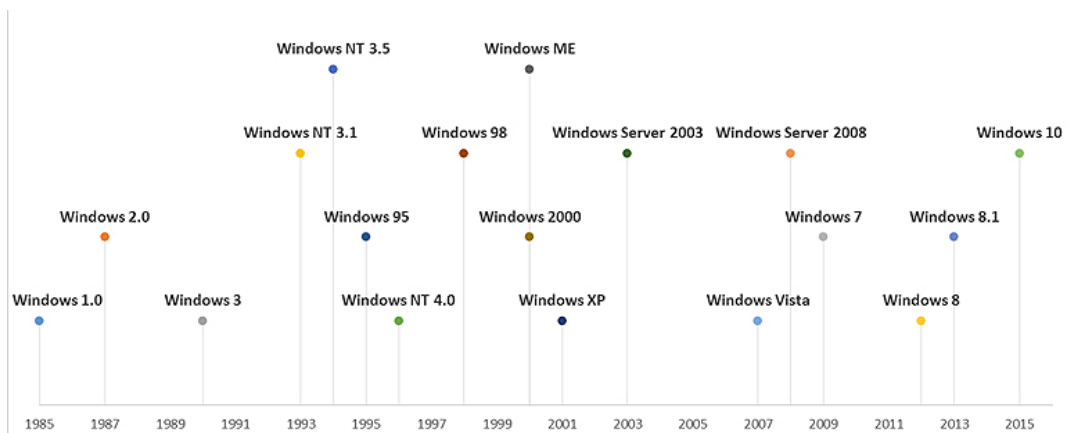


11

O ano de 2015 será o de lançamento da versão mais recente do Sistema Operacional da Microsoft, o Windows 10, que foi concebido para ser a versão definitiva de unificação do Windows através de todos os dispositivos, incluindo tablets e smartphones.

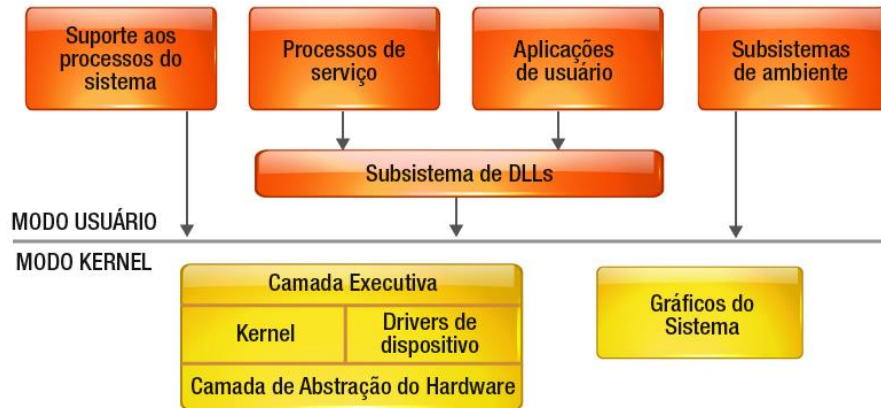
Uma das características mais interessantes do SO, e relacionada justamente à interoperabilidade entre dispositivos, é a **capacidade de alternância entre o modo de operação tablet e o modo de operação com o mouse e teclado.**

A linha do tempo com a evolução das versões do MS Windows ao longo dos seus mais de 20 anos de existência é exibida na imagem a seguir.



2 - ESTRUTURA DO WINDOWS

Assim como grande parte dos sistemas operacionais modernos, o Microsoft Windows é um SO monolítico, mas que possui as funcionalidades segmentadas em blocos e que funcionam parte em modo usuário e parte em **modo kernel**, conforme pode ser visualizado na figura abaixo, que exibe o esquema de funcionamento interno do Windows.



No modo usuário, se localizam os serviços relacionados ao suporte aos processos do sistema, os processos de serviço, o subsistema de ambiente e as aplicações de usuário. Os processos de serviço e as aplicações de usuário utilizam as interfaces de serviços providas pelo kernel do SO através da chamada de funções localizadas no subsistema de DLLs.

Já no modo kernel, se localiza o bloco central do sistema, que contempla o subsistema executivo, os drivers de dispositivo, o kernel e a camada de abstração de hardware. É neste modo que se localiza, ainda, o módulo responsável pela gerência dos gráficos.

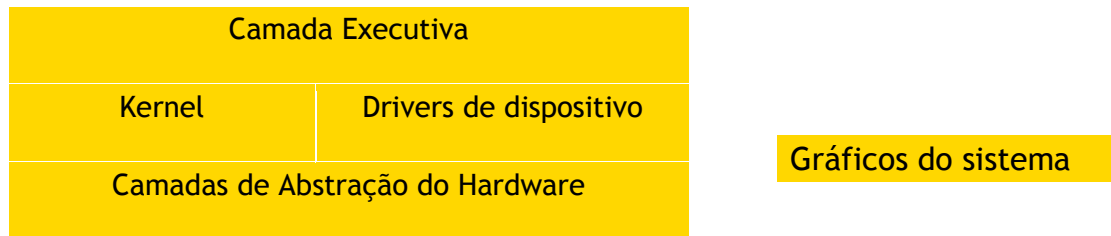
Monolítico

A arquitetura monolítica é semelhante a uma aplicação formada por vários módulos que são compilados separadamente e depois linkados formando um grande e único programa executável.

a) Estrutura do Modo Kernel

Como já comentado, a porção de código do núcleo Windows que é executada em modo kernel é dividida em quatro partes principais: a camada de abstração de hardware (HAL), os subsistemas executivos, o kernel propriamente dito e os drivers de dispositivo, e um módulo auxiliar para apoio ao gerenciamento dos gráficos do sistema.

Modo Kernel



De uma forma geral, todas as camadas localizadas no modo kernel têm permissão para:

- Acessar diretamente o hardware sem a necessidade de utilizar obrigatoriamente uma interface de software.
- Executar processos em um grau maior de prioridade do que os executados no modo usuário.
- Acessar diretamente o espaço de memória física do computador.

Camada Executiva

A camada **executiva** do Windows é a primeira da pilha de camadas executadas em modo kernel, sendo responsável pelo provimento dos serviços base do sistema operacional, como o gerenciamento de memória, o gerenciamento de processos, a gestão da rede e a comunicação entre os processos.

Kernel

A camada de **kernel**, localizada imediatamente abaixo, tem a função de prover os serviços de mais baixo nível do SO, como o de agendamento das threads, controle de interrupções e sincronização de múltiplos processadores. É responsável, ainda, por prover uma série de interfaces que são utilizadas pela camada executiva para prover os seus serviços de mais alto nível.

Drivers de dispositivo

A camada de **drivers de dispositivo** agrupa tanto os drivers de dispositivo ligados ao hardware, que, como já apresentado, atua na intermediação do acesso ao hardware abstraindo a sua complexidade, quanto drivers de dispositivo não ligados ao hardware, como, por exemplo, os que tratam do acesso ao sistema de arquivos.

Camada de Abstração do hardware

A camada de **abstração de hardware** é a porção de código que atua no isolamento das camadas de kernel, de drivers de dispositivo e de todo o resto da plataforma executiva do Windows da estrutura de hardware do equipamento. Esta iniciativa tem como principal objetivo permitir a portabilidade dos

sistemas Windows entre diferentes equipamentos.

Gráficos do sistema

A última camada executada no modo kernel é a de **gráficos do sistema**, que tem por atribuição implementar as funções relacionadas a interface gráfica do usuário.

14

b) Estrutura do Modo Usuário

Todas as tarefas em modo usuário se caracterizam por serem executadas em um espaço de endereçamento de processos protegido, sendo fornecido um endereço próprio e privado para cada elemento em execução.

São quatro os módulos básicos executados em modo usuário no Microsoft Windows, dentre os quais:

- Suporte aos processos do sistema

Agrupa os serviços que não são serviços do Windows propriamente ditos, ou seja, não são iniciados pelo gerenciador de controle dos serviços, a exemplo do processo de autenticação no sistema e o gerenciamento de sessões.

- Processos de serviço

Agrupa os serviços que são iniciados pelo gerenciador de controle de serviços e, por este motivo, são tratados como serviços do Windows, a exemplo do gerenciador de tarefas e do spooler de impressão.

- Aplicações de usuário

Ambiente reservado para execução das aplicações utilizadas pelo usuário final.

- Subsistemas de ambiente

Implementam o suporte a execução de aplicações que não foram nativamente desenvolvidas para o Windows.

Dentre os grupos de serviços do modo usuário, o subsistema de ambiente se apresenta como um elemento que merece destaque. A arquitetura do Windows foi desenhada para dar suporte a múltiplos e independentes subsistemas de ambiente. Saiba+

Saiba+

Na prática, é como se o sistema operacional provesse um conjunto de interfaces que permitisse acesso a funcionalidades dos serviços executivos de sistema, que originalmente são executados em modo kernel, para aplicações que não são obrigatoriamente desenvolvidas para serem executadas no Windows.

Imagine, por exemplo, que determinadas versões do Windows fornecem um subsistema para aplicações UNIX, o *Subsystem for Unix-based Applications (SUA)*. A existência deste subsistema permite que aplicações desenvolvidas para UNIX possam ser compiladas e executadas em um computador que está executando o MS Windows.

RESUMO

Ao longo das mais de duas décadas de existência do Windows, foram lançadas mais de uma dezena de versões, e incorporadas uma série de funcionalidades que acabaram por se tornar padrão de mercado para qualquer sistema operacional voltado ao usuário final. Apesar de não lançar o uso do dispositivo apontador, ou mouse, o MS Windows foi o responsável pela popularização, além disso, foi um dos principais incentivadores da melhoria da usabilidade das interfaces gráficas dos Sistemas Operacionais, de modo a facilitar o uso e reduzir o tempo de aprendizado dos operadores.

A primeira versão do Windows foi lançada em 1985, como a primeira tentativa real da Microsoft em permitir ao usuário utilizar uma interface gráfica sobre o sistema operacional MS DOS. Deste período até os dias atuais, o desenvolvimento do *software* foi marcado por alguns fracassos, como o Windows ME e Windows Vista, e por grandes sucessos, como o Windows XP e o Windows 7.

Um importante fato a ser observado é que as últimas versões do MS Windows têm se voltado cada vez mais em direção a computação móvel. A versão 10 do Windows, que está prestes a ser lançada, foi concebida para ser definitivamente a versão de unificação da plataforma do Windows através de todos os dispositivos, incluindo tablets e smartphones.

Internamente, assim como grande parte dos sistemas operacionais modernos, o Microsoft Windows é um SO monolítico. A sua estrutura se segmenta em camadas separadas em dois modos de operação, modo usuário e modo kernel. No modo usuário, se localizam os serviços relacionados ao suporte aos processos do sistema, os processos de serviço, o subsistema de ambiente e as aplicações de usuário. Os processos de serviço e as aplicações de usuário utilizam as interfaces de serviços providas pelo kernel do SO através da chamada de funções localizadas no subsistema de DLLs. Já no modo kernel, se localiza o bloco central do sistema, que contempla o subsistema executivo, os drivers de dispositivo, o kernel e a camada de abstração de hardware. É neste modo que se localiza, ainda, o módulo responsável pela gerência dos gráficos.

UNIDADE 4 – IMPLEMENTAÇÕES DE SISTEMAS OPERACIONAIS

MÓDULO 2 – SISTEMA OPERACIONAL LINUX

01

1 - INTRODUÇÃO AO LINUX

As distribuições Linux surgiram em um contexto onde o mercado de sistemas operacionais era dominado por uma opção proprietária, não existindo uma alternativa gratuita em condições de competir em igualdade.

Antes de continuarmos, é importante ressaltar a diferença existente entre os termos “Linux” e “Distribuição Linux”.

O nome **Linux** é dado ao kernel do Sistema Operacional, elemento responsável por executar as operações básicas do SO.

Quando falamos de uma **distribuição Linux**, estamos fazendo referência a um pacote completo, composto pelo kernel Linux e uma série de aplicações e bibliotecas associadas para prover a usabilidade necessária para utilização do Sistema Operacional.

A história do Linux se inicia décadas antes do seu lançamento, quando ainda na década de sessenta pesquisadores vinculados a diversas empresas e instituições de ensino, como a AT&T's Bell Laboratories e o MIT, iniciaram um projeto para construir um sistema operacional voltado para Mainframes e que fosse multiusuário e multitarefa. Apesar do fracasso deste projeto, denominado MULTICS, uma série de ideias extraídas do trabalho desenvolvido motivaram os pesquisadores da Bell no desenvolvimento da primeira versão de outro Sistema Operacional, que foi lançada no ano de 1971 sob a alcunha de **UNIX**.

A reescrita completa em linguagem C do código fonte do UNIX o tornou **portável**, ou seja, fez que o sistema operacional pudesse ser desenvolvido como um ambiente de *software* comum, sem a necessidade de ser reescrito a cada novo computador que fosse lançado.

A portabilidade do UNIX foi um dos grandes marcos na história dos sistemas operacionais modernos e fez que, rapidamente, versões do UNIX se espalhassem por universidades e centros de pesquisa, o que ajudou no desenvolvimento e aprimoramento das funcionalidades do Sistema Operacional.

02

A grande reviravolta do UNIX ocorreu quando a AT&T passou a vender versões comerciais do Sistema Operacional, não mais disponibilizando o código fonte do SO para estudo ou desenvolvimento por terceiros. A principal consequência desta mudança foi uma batalha legal entre AT&T e outros desenvolvedores do UNIX, o que resultou na criação de duas diferentes linhas de desenvolvimento, o SYSV, que representou a continuidade da ideia original do UNIX e permaneceu sendo tocado pela AT&T e parceiros comerciais, e o BSD (Berkeley Software Distribution), lançado pela *University of California at Berkeley* e que teve o seu desenvolvimento continuado por alunos da Universidade e de instituições de pesquisa parceiras.

Além da criação do BSD, o fim da divulgação do código fonte das versões do UNIX produziu ainda uma série de lançamentos de outros projetos independentes nos anos 80 como, por exemplo, os sistemas operacionais AIX, criado pela IBM e HP-UX, de propriedade da HP.



Esta grande quantidade de linhas de desenvolvimento do UNIX, patrocinadas por diferentes fornecedores, motivou a criação de dezenas de diferentes dialetos de operação do SO, cada um funcionando apenas em uma determinada distribuição, mesmo que propostos para realizar tarefas

similares. Isto fez com que o conceito original do SO, onde diversas instituições trabalhavam no aprimoramento de um conceito comum, fosse perdido, e se criasse uma competição entre cada fornecedor, que passaram a não compartilhar as melhorias implementadas como forma de manter vantagem competitiva uns sobre os outros.

03

Foi neste contexto, e competição e isolamento, que Richard Stallman lançou o projeto GNU (GNU is not UNIX).

O projeto GNU tinha como principal objetivo criar um sistema operacional que fosse aberto para todos e onde todos pudessem trabalhar juntos, resgatando um pouco do que já acontecia nos anos setenta.

O projeto GNU começou a ser desenvolvido no ano de 1984 e almejava criar um SO que fosse compatível com o UNIX. Até o final de 1991, Stallman e sua equipe de colaboradores já haviam desenvolvido quase todos os componentes principais de um sistema operacional, como, por exemplo, os compiladores, interfaces gráficas e bibliotecas de sistema; entretanto, ainda faltava o elemento mais importante: o kernel do sistema.

Para sorte da equipe GNU, justamente nesta época, Linux Torvalds, então estudante da Universidade de Helsinki, publicou como código aberto o fonte de um kernel tipo UNIX que havia desenvolvido durante seu período na universidade, componente que havia nomeado como **kernel Linux**. A junção do pacote de componentes GNU com o kernel Linux resultou na criação de um dos mais famosos sistemas operacionais da atualidade.

O núcleo do GNU/Linux é inteiramente original, ou seja, não utiliza nenhum código proveniente de distribuições UNIX. Entretanto, a manutenção da compatibilidade com o UNIX faz com que o Sistema Operacional GNU/Linux consiga executar boa parte dos códigos de aplicativos desenvolvidos para este sistema operacional.

04

2 - DISTRIBUIÇÕES DO LINUX

À medida que o Linux foi se tornando popular, uma série de versões começou a ser desenvolvida por diferentes atores, que mantinham o núcleo padrão, mas proviam melhorias em elementos acessórios do SO, como utilitários do sistema ou interfaces de usuário.

Estas diferentes linhas de desenvolvimento, nas quais cada fornecedor montava o seu sistema operacional com um determinado conjunto de componentes, originaram o conceito que ficou conhecido como **distribuições Linux**.

Alguns pesquisadores determinam o MCC Interim Linux como a primeira distribuição Linux capaz de ser instalada em uma máquina sem que nenhum outro sistema operacional estivesse previamente instalado. Ao passar dos anos uma série de outras distribuições foram lançadas, a mais antiga e que ainda se encontra atualmente é o Slackware. Atualmente, são contabilizadas mais de quatrocentas

distribuições Linux ativas, algumas das distribuições mais utilizadas serão analisadas nas seções subsequentes.



05

a) Debian

O projeto de criação do Debian GNU/Linux foi iniciado no ano de 1993 por Ian Murdoc, quando o conceito de montagem de diferentes distribuições do Linux ainda era algo recente. Em seu início, a distribuição Debian era a única dentre as grandes distribuições que mantinha o código completamente aberto para qualquer desenvolvedor ou usuário que desejasse contribuir com o projeto. Saiba+

Em seus estágios iniciais, o Debian era distribuído através do código fonte, que deveria ser descompactado e compilado pelos usuários finais. Entretanto, não demorou muito para que fosse criado o gerenciador de pacotes binários próprio do Debian e os arquivos “.deb”, uma das marcas registradas do sistema operacional. Estes elementos, além de prover a instalação automática dos pacotes desejados, também permitiam o controle de dependências e de conflitos entre os pacotes. O sistema se tornaria completo com a inclusão do *Advanced Package Tool* (APT), no Debian 2.1, e a criação dos repositórios online de pacotes, fato que tornou intuitivo e trivial instalar novos aplicativos no sistema e que é considerado um dos principais elementos das versões atuais do Debian.



Uma das principais características do Debian é a sua estabilidade, que em muito está relacionada ao conservadorismo da equipe responsável pela manutenção do sistema. Os mantenedores do Debian têm por diretriz protelar a inclusão de novas tecnologias no SO até que estas estejam largamente estabelecidas e testadas. Além disso, o Debian é um SO de ciclo de publicação lento, ou seja, normalmente levam um a três anos para que cada nova versão estável seja lançada.

Saiba+

É inquestionável o fato de que o Debian é uma das principais distribuições Linux do mercado. O sucesso do Debian pode ser mensurado pelo atual número de colaboradores voluntários, que ultrapassa o primeiro milhar, e pelas mais de cento e vinte distribuições desenvolvidas com base neste sistema operacional, dentre elas algumas extremamente relevantes, como o Linux Ubuntu.

06

b) Ubuntu

A criação da distribuição Ubuntu, no ano de 2004, aconteceu em um momento em que a utilização do Linux já estava estabelecida na plataforma corporativa, mas ainda apresentava dificuldades em se estabelecer nas estações de trabalho dos usuários domésticos. Neste contexto, um pequeno grupo de usuários se reuniu e, tomando como base uma das mais estáveis distribuições da época (Debian), montou um ambiente desktop como maior facilidade de uso por parte dos usuários, com a ideia de competir em condições de igualdade e ganhar mercado frente ao principal SO neste segmento, o Microsoft Windows.

Apesar de não conseguir alcançar o objetivo de desbancar o SO Microsoft Windows, o Ubuntu logrou algum êxito em sua empreitada.

Nos últimos anos, a equipe de desenvolvimento do Ubuntu tem se arriscado em um novo nicho de mercado. Após passar quase todo o seu período de desenvolvimento focado no mercado de computadores pessoais, o Ubuntu Linux começou um movimento em direção a plataformas móveis, como *smartphones* e *tablets*.

A estimativa de lançamento do primeiro smartphone executando o Sistema Operacional Ubuntu Touch, versão específica para dispositivos móveis, é o ano de 2015. Os principais elementos enfatizados no desenvolvimento deste SO são a criação de um novo paradigma de interface de usuário, que promove a entrega de conteúdo e serviços diretamente em telas personalizadas, e no desenvolvimento de um método fácil para criação e integração de novas aplicações com o Sistema Operacional.

Protótipo do Ubuntu executado em dispositivo móvel



Descubra algumas das principais funcionalidades do Ubuntu Touch, versão para aplicativos móveis do Linux Ubuntu, acessando o **vídeo**.

Êxito

Estimativas da Canonical, empresa responsável por dar suporte comercial ao Ubuntu, apontam que existem cerca de vinte e cinco milhões de usuários do Sistema Operacional ao redor do mundo, o que torna o SO o terceiro mais popular no geral e o primeiro dentre as distribuições Linux, a frente de fortes distribuições concorrentes, como o próprio Debian.

07

C) Linux Mint

Dentre as distribuições Linux que tem tido mais sucesso nos últimos anos, o Linux Mint é uma das que teve o início do desenvolvimento mais recentemente, já que foi lançada em sua primeira versão apenas no ano de 2006. Apesar de ser baseada na distribuição Ubuntu, o Linux Mint conseguiu se tornar ainda mais intuitivo e próximo do usuário do que a distribuição que o originou, já que atuou na correção de diversos dos problemas de usabilidade herdados do Ubuntu.

Dentre as razões que fizeram com que o Linux Mint se tornasse uma das distribuições Linux mais utilizadas em todo o mundo, pode-se citar:

- A instalação do sistema é extremamente simples e intuitiva, com usabilidade ainda melhor do que a do Ubuntu.
- Possui uma versão inteligente do desktop, que permite personalizá-lo de acordo com o desejo do usuário.
- A distribuição já vem com uma série de aplicativos que são automaticamente instalados em conjunto com o SO, como o pacote de escritório LibreOffice e o Gerenciador de Arquivos Nautilus.
- Os temas disponíveis para utilização no desktop são originais e seu uso se encaixa em diversos ambientes, pessoais ou profissionais.

- O processo de compartilhamento de pastas em ambientes de rede com estações de trabalho Windows é simplificado, se comparado com outras distribuições Linux.
- Vem com um simples e intuitivo sistema de controle de acesso, permitindo nativamente que os pais bloqueiem o acesso dos filhos a conteúdo não autorizado.

Imagem de área de trabalho personalizada no Linux Mint



08

d) Red Hat, Red Hat Enterprise e CentOS

Apesar do espaço que perdeu na última década, a distribuição Red Hat Linux, lançada em 1994, certamente ainda é uma das mais conhecidas em todo o mundo. Uma das grandes inovações iniciada com o lançamento do Red Hat, e hoje presente na maioria das grandes distribuições, é a utilização de pacotes binários para instalação de aplicativos e atualização do sistema. É importante ressaltar que, à época do lançamento da funcionalidade, as distribuições eram instaladas a partir da realização da cópia de um disco com o Linux em operação ou através de disquetes, onde o sistema se mantinha fragmentado em diferentes partes.

A lógica de utilização de pacotes binários auxiliou na resolução de um dos principais problemas das distribuições Linux de uma forma geral, a atualização completa do sistema. Saiba+

A diminuição da base de usuários do Linux Red Hat se iniciou em 2003, quando a versão gratuita foi descontinuada em detrimento da versão Red Hat Enterprise Linux (RHEL), voltada a ambientes corporativos e que passou a ser comercializada em conjunto com a prestação do serviço de suporte. Como toda a base do código fonte é baseada em código livre, apesar de não fornecer os binários compilados do RHEL gratuitamente, a Red Hat foi obrigada a disponibilizar o código completo da sua versão Enterprise em um repositório público.

Tela de login no RHEL 5



Com base no código fonte do RHEL, diversos grupos de usuário se formaram para fornecer gratuitamente sistemas compilados clones do Red Hat. Dentre estes, o CentOS foi a iniciativa que obteve o maior sucesso, atingindo a marca de sistema operacional mais utilizado em servidores corporativos no ano de 2010.

Saiba+

Antes do lançamento dos pacotes, um dos procedimentos utilizados dentre as distribuições era o de deletar o Sistema antigo do disco rígido e começar uma nova instalação. Como cada pacote provê um pedaço diferente de *software*, tornou-se possível identificar apenas os pacotes que se encontravam desatualizados promovendo a atualização personalizada, sem a necessidade de reinstalar todo o sistema.

09

3 - ESTRUTURA INTERNA DO LINUX

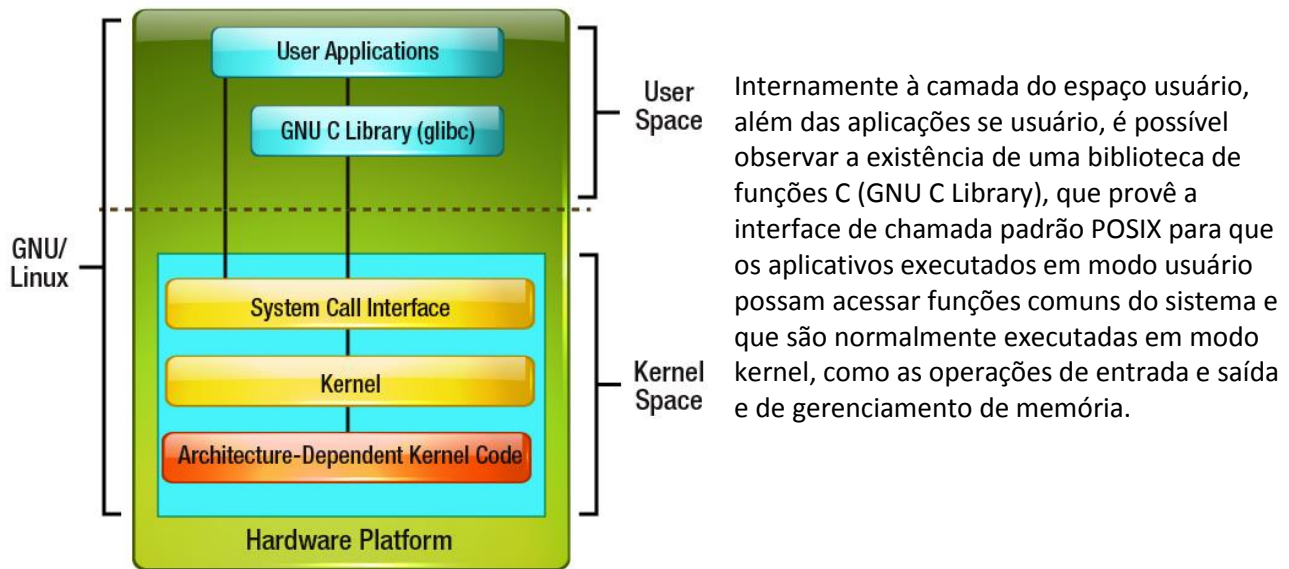
A grande maioria das distribuições Linux é executada sobre um kernel monolítico e capaz de carregar e descarregar dinamicamente algumas porções de código ligado ao núcleo, usualmente chamados de **módulos**, que tem como um dos principais exemplos os drivers de dispositivo.

Além disso, o kernel Linux também tem a característica de ser preemptivo que, conforme explanado anteriormente, é aquele que permite a execução de uma interrupção no meio da execução de um determinado código, como resposta a uma chamada do sistema, criando a ilusão ao usuário de que diversos processos são executados simultaneamente em cada processador.

Quando a arquitetura do sistema Linux é pensada em alto nível, pode-se segmentá-la em função de duas camadas distintas:

- o **espaço de usuário**, que é o local onde as aplicações são executadas,
- e o **espaço do kernel**, que é onde efetivamente são executadas as operações de gestão do Sistema Operacional.

A figura abaixo traz uma visão esquemática do funcionamento do Linux.



Já no espaço kernel, pode-se observar três grandes blocos: um que agrupa as **interfaces de chamada do sistema**, que implementa as funções mais básicas e se localiza no topo da pilha; um segundo que agrupa o **kernel** do sistema, que é independente da arquitetura do computador e que contém as funções como as de gerenciamento de processos e de memória, o sistema de arquivos virtual, a pilha de rede e drivers de dispositivo; e um terceiro, mais abaixo, que agrupa todo o código que é **dependente da plataforma**.

10

4 - PROCESSOS DE GERENCIAMENTO

O gerenciamento de processos no Linux é focado na atividade de execução. Geralmente, o kernel provê acesso às funções de criação, parada e sincronização dos processos diretamente através da interface de chamada do sistema ou através das chamadas padrão POSIX, utilizando a biblioteca C (*glibc*) como intermediária.

Já em relação ao gerenciamento de memória, costuma utilizar páginas de 4 kb, como forma de prover maior eficiência ao sistema por meio da criação de uma área de swap entre memória principal e disco.

Em relação ao gerenciamento do sistema de arquivos, nativamente, os principais sistemas utilizados nas diversas distribuições Linux são variações do Extended File System, como o ext3 ou o ext4, e o ReiserFS. Além disso, o Linux utiliza o conceito de sistema de **arquivos virtual** (Virtual File System - VFS) para manter a compatibilidade do Linux com os sistemas de arquivos utilizados em outros Sistemas Operacionais.

Através do VFS, o kernel do Linux permite que sejam montados diretamente no SO os sistemas de arquivos nativos de todas as versões do MS-DOS e MS Windows, como o File Allocation Table (FAT) e o New Technology File System (NTFS).

11

RESUMO

As distribuições Linux surgiram em um contexto onde o mercado de sistemas operacionais era dominado por uma opção proprietária, não existindo uma alternativa gratuita em condições de competir em igualdade. O nome Linux é dado ao kernel do Sistema Operacional, elemento responsável por executar as operações básicas do SO, já quando falamos de uma distribuição Linux, estamos fazendo referência a um pacote completo, composto pelo kernel Linux e uma série de aplicações e bibliotecas associadas para prover a usabilidade necessária para utilização do Sistema Operacional.

Ao longo dos seus anos de desenvolvimento, foram criadas diversas distribuições Linux. Alguns pesquisadores determinam o MCC Interim Linux como a primeira distribuição Linux capaz de ser instalada em uma máquina sem que nenhum outro sistema operacional estivesse previamente instalado. Atualmente, entretanto, são outras as principais distribuições, tendo destaque o Debian GNU/Linux, o Linux Ubuntu, o Linux Mint, as distribuições Red Hat e o CentOS.

A grande maioria das distribuições Linux é executada sobre um kernel monolítico e, em sua estrutura interna, pode ser segmentado em função de duas camadas distintas, o espaço de usuário, que é o local onde as aplicações são executadas, e o espaço do kernel, que é onde efetivamente são executadas as operações de gestão do Sistema Operacional.

O gerenciamento de processos no Linux é focado na atividade de execução, sendo que as interfaces para a execução de funções sobre os processos são providas normalmente através da biblioteca C (*glibc*). Já em relação ao gerenciamento de memória, as distribuições costumam utilizar páginas de 4 kb, como forma de prover maior eficiência ao sistema através da criação de uma área de swap entre memória principal e disco. Em relação ao gerenciamento do sistema de arquivos, nativamente, os principais sistemas utilizados nas diversas distribuições Linux são variações do Extended File System, como o ext3 ou o ext4, e o ReiserFS. Além disso, o Linux utiliza o conceito de sistema de arquivos virtual (VFS) para manter a compatibilidade do Linux com os sistemas de arquivos utilizados em outros Sistemas Operacionais.

UNIDADE 4 – IMPLEMENTAÇÕES DE SISTEMAS OPERACIONAIS

MÓDULO 3 – SISTEMA OPERACIONAL ANDROID

01

1 - HISTÓRIA DO SISTEMA OPERACIONAL ANDROID

Por muitos anos, os sistemas operacionais foram pensados prioritariamente para as estações de trabalho dos usuários ou para operacionalizar serviços corporativos. Entretanto, com a miniaturização

dos componentes e o aumento do poder de processamento dos dispositivos portáteis, percebeu-se um crescimento exponencial dos estudos relacionados com a computação móvel.

Um dos principais resultados destes estudos foi a transformação dos telefones celulares em dispositivos inteligentes e com poder de processamento próxima a determinadas categorias de computadores de mesa. O incremento do poder computacional destes dispositivos levou à necessidade de desenvolvimento de um sistema operacional robusto e que comportasse a execução de aplicações complexas. Foi nesta linha, de prover uma maior gama de funcionalidades aos celulares inteligentes, que surgiram diversas iniciativas de desenvolvimento dos Sistemas Operacionais, a exemplo do BlackBerry OS, do IOS e do Android.

O Android nunca se apresentou como um Sistema Operacional adaptado para o mundo móvel já que, desde a sua origem, sempre foi pensado como uma ferramenta voltada à plataforma móvel.

Uma das principais características do SO Android é o fato de que mantém o seu código fonte aberto, permitindo a sua personalização e utilização em uma vasta gama de dispositivos, dos mais variados fabricantes, fator fundamental para que se tornasse o Sistema Operacional mais utilizado entre os *smartphones*.

Nas seções seguintes, serão apresentados um breve histórico deste tão conhecido Sistema Operacional e alguns detalhes da sua estrutura interna de funcionamento.



02

Antes de ser adquirido pelo Google em 2005, o Sistema Operacional Android era desenvolvido por uma empresa *startup* chamada Android Inc.

A primeira versão comercial do Android, denominada **1.0 G1**, foi lançada em fevereiro de 2008, embarcada no T-Mobile G1, *smartphone* desenvolvido pela HTC. Dentre as principais novidades do novo SO, destacava-se o suporte ao Bluetooth e GPS, a integração com os principais aplicativos do Google, como o Maps, o Calendar e o Gmail, e a conexão com a Android Market, loja de aplicativos que contava com apenas 35 itens em sua estreia.





A próxima versão relevante do SO do Google, a **1.5 Cupcake**, foi disponibilizada para o mercado em abril de 2009 e incorporou ao Android uma caixa de busca universal. Incluiu, ainda, a possibilidade de se alterar entre modo de fotografia e filmagem de vídeo e expandiu o suporte a gestos para operação do SO.

Ainda em 2009 foi lançada a versão **1.6 Donut**, momento no qual o sistema operacional foi provido com bluetooth estéreo, teclado virtual na tela e a possibilidade de upload direto de arquivos para youtube e Picasa, aplicativos também de propriedade do Google.



03

Antes de ser adquirido pelo Google em 2005, o Sistema Operacional Android era desenvolvido por uma empresa startup chamada Android Inc.

Apenas um mês após o lançamento do Android Donut, o Google anunciou a disponibilização da versão **2.0 Eclair**, que trouxe como principais funcionalidades a possibilidade de criação de múltiplas contas de usuário, o provimento de suporte ao MS Exchange e a utilização de uma caixa de correio eletrônico universal, além de melhorias no teclado, através da incorporação de dicionários de sugestão de nomes durante a digitação. Já a **versão 2.1** apresentou poucas melhorias em relação à antecessora, como, por exemplo, a utilização de múltiplas telas iniciais e a adição da funcionalidade que convertia automaticamente voz para texto escrito.



O lançamento seguinte do Android foi a versão **3.0 HoneyComb**. Dentre as novas características incorporadas ao sistema, destaca-se a inclusão de suporte a gráficos 3D, a possibilidade de utilização de chats de vídeo e o compartilhamento da conexão internet com outros dispositivos através do uso do Bluetooth (*Bluetooth tethering*).

Apesar das melhorias incluídas no Android HoneyComb, o grande divisor de águas entre as versões do Sistema Operacional aconteceu com o lançamento da versão 4.0, o **Android Ice Cream Sandwich**. [Clique aqui para ver alguns avanços trazidos por essa release.](#)



Clique aqui

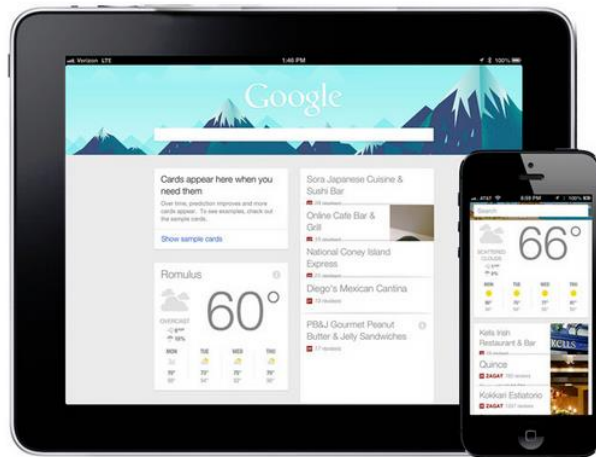
Algumas das principais modificações ocorridas com o lançamento do Android Ice Cream Sandwich são exibidas abaixo.

- Adição do Android Beam, funcionalidade que dá suporte a troca de informações entre dois aparelhos que estejam executando o SO Android apenas encostando um equipamento no outro.
- Possibilidade do desbloqueio do celular através do uso de reconhecimento facial
- Integração da agenda de contatos com as redes sociais, permitindo a recuperação automática de contatos online em complemento àqueles adicionados manualmente.
- Melhor gerenciamento dos dados, com o controle e monitoramento do uso da atividade da rede e visualização do fluxo de dados de cada aplicativo.
- Nova interface de usuário, com design futurista e disponibilização de ícones multitarefa que permitiam o acesso às aplicações recentemente utilizadas.
- Screen Shots – a possibilidade de registrar uma fotografia momentânea da tela do Android só apareceu na versão 4.0 do SO.
- Diversas características de acessibilidade, como o modo explore-by-touch, que permite a um deficiente visual navegar pelo SO tocando na tela do aparelho e ouvindo uma descrição de voz do conteúdo acessado.
- Possibilidade de operar o aparelho executando as funções através da utilização de comandos de voz, com a possibilidade de, por exemplo, criar um novo e-mail ou até mesmo realizar uma ligação sem precisar utilizar a tela sensível ao toque.
- Sincronização automática de sites favoritos e páginas salvas entre diversos equipamentos.

Suporte a Wi-fi direto

04

Ainda dentro dos releases da versão 4, algumas atualizações importantes foram lançadas. O Android **Jelly Bean** 4.1, disponibilizado em Julho de 2012, trouxe o Chrome como navegador internet padrão do SO e a possibilidade de busca no sistema por meio de comandos de voz. Incorporou, ainda, o aplicativo Google Now, modelo de assistente pessoal criado pelo Google com o objetivo de organizar a rotina do usuário, provendo informações sobre trânsito, notícias de interesse e previsão do tempo baseando-se em perfis individuais.



As versões **Android 4.2 e 4.3**, por sua vez, se destacaram pela melhoria nas funções de acessibilidade, por permitir a localização do dispositivo através da utilização de wi-fi ou triangulação de antenas e por prover suporte a dispositivos equipados com telas em tecnologia 4k. Já a versão 4.4, o **Android Kitkat**, incorporou ao sistema o serviço de impressão sem fio através do “Google Cloud Print”, e o modo imersivo, onde as aplicações passaram a ter a possibilidade de ocultar barras de navegação de status.

05

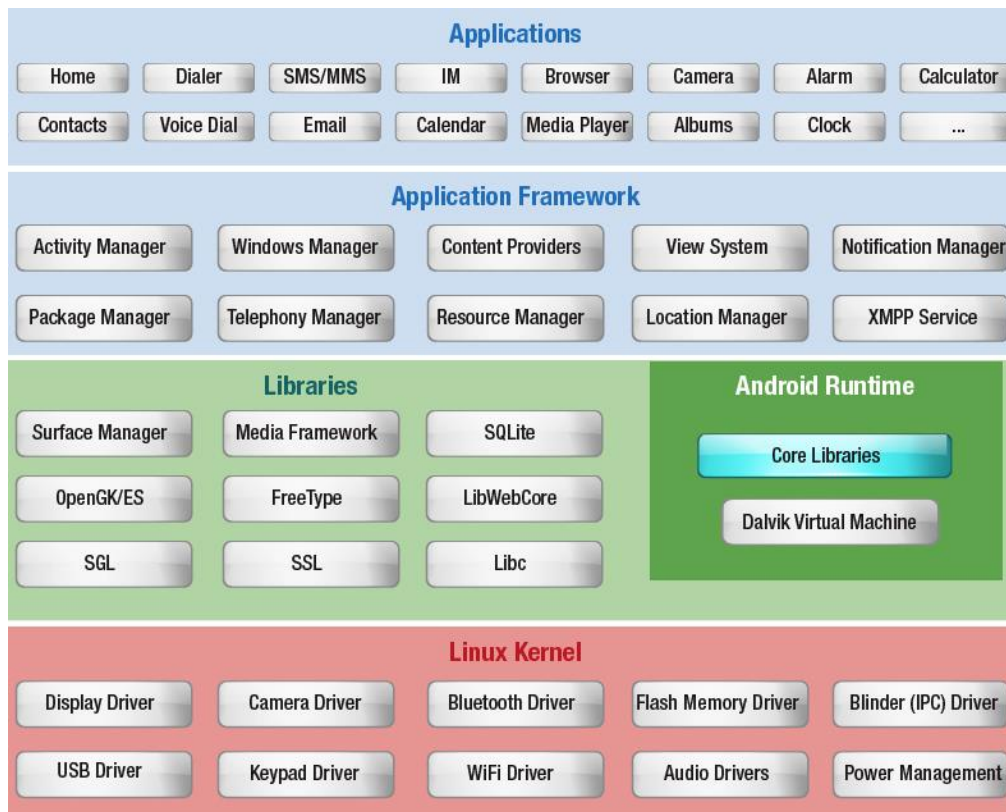
A versão mais recente do Android, a **Lollipop**, foi revelada em outubro de 2014 com o objetivo de ser o maior lançamento desde a origem do Sistema Operacional do Google. Uma das principais inovações da versão foi a inclusão de uma interface de usuário completamente remodelada, denominada “material”, que trouxe consigo a filosofia de atuar na integração de diferentes plataformas.



Além da interface material, o Android Lollipop tornou o gerenciamento de notificações mais acessível e fácil de operar, aproveitando todo o espaço disponível na tela para apresentar informações relevantes, como mensagens de e-mail ou textos SMS. Outro ponto relevante da versão foi a inclusão, por padrão, do ambiente de execução ART, o que melhorou o desempenho da execução das aplicações e permitiu uma redução no consumo de bateria.

2 - ESTRUTURA DO ANDROID

A estrutura interna do Android foi desenvolvida em função de camadas que se organizam em formato de uma pilha de *software*, onde cada componente assume uma função própria. Uma visão de alto nível das camadas do sistema operacional é exibida na figura abaixo.



Como pôde ser visualizado, o Android é composto basicamente de quatro camadas de *software*:

- a de aplicações, que fica no topo da pilha;
- uma segunda camada onde estão localizadas as funções de frameworks de aplicação;
- uma camada de bibliotecas e ambiente de Execução Android;
- e uma última camada que representa o kernel Linux, componente fundamental do SO e que fica na parte de baixo da pilha de *software*.

Cada uma destas camadas será detalhada nas seções posteriores.

a) Kernel Linux

A utilização do kernel do Linux como base central do desenvolvimento do Android é uma herança da época em que o SO ainda era um produto Android inc, e visava reduzir o custo e o tempo de desenvolvimento do produto através da reutilização do máximo possível de código existente em outros *softwares* abertos.

O Kernel Linux desempenha as funções de mais baixo nível do Sistema Android, sendo responsável, por exemplo, pelo gerenciamento dos processos, da memória, da comunicação e o gerenciamento do sistema de arquivos. Esta camada é responsável também por prover a interface de abstração do *hardware*, agrupando o principal conjunto de drivers de dispositivo, como os de vídeo, os drivers USB e Bluetooth, dentre outros.

É importante ressaltar que, apesar de ser construído sobre o kernel do Linux, o Android não se configura como uma distribuição Linux. Prova disto é que não é possível executar aplicativos nativamente compilados para Linux em um ambiente Android.

08

b) Bibliotecas e Ambiente de Execução Android

A camada imediatamente acima do kernel Linux é a que agrupa as bibliotecas nativas do sistema e o ambiente de execução do Android (*Android Runtime*).

As bibliotecas nativas são escritas nas Linguagens C e C++ e habilitam os dispositivos a lidar com diferentes tipos de dados. Dentre as principais bibliotecas, pode-se citar:

- SQLite.
- OpenGL;
- Surface Manager;
- Media Framework;
- LibWebCore.

Como já mencionado, além das bibliotecas nativas, esta camada também congrega o ambiente de execução do SO. Neste sentido, é importante ressaltar que o Android tem um requisito especial e que o diferencia em muito do seu principal concorrente, o IOS. Esta característica se relaciona ao fato de que o Android teve que ser pensado para dar suporte a uma infinidade de diferentes plataformas de *smartphones*, conquanto o concorrente foi desenvolvido para ser executado apenas em poucos *hardwares* da Apple, como o iPhone e o iPad.

SQLite

Banco de dados utilizado pelo Android com o propósito de armazenamento de dados do sistema.

OpenGL

Utilizada para prover a renderização na tela de conteúdos gráficos em 2D e 3D.

Surface Manager

É utilizado para compor o gerenciador de janelas. Neste modelo, o desenho feito diretamente na tela vai para um espaço de armazenamento temporário e só é plotado na tela final do usuário após uma combinação com outros modelos previamente definidos pelo sistema operacional.

Media Framework

Provê os principais codecs de áudio e vídeo, fornecendo suporte a execução e gravação de mídias em diferentes formatos.

LibWebCore

Biblioteca para renderização rápida de código em HTML.

09

Por conta da necessidade de dar suporte a uma diversidade de *hardwares*, o Google projetou o Android baseado na utilização de uma **máquina virtual** para executar os aplicativos de usuário, dispositivo este que foi denominado **Dalvik**.

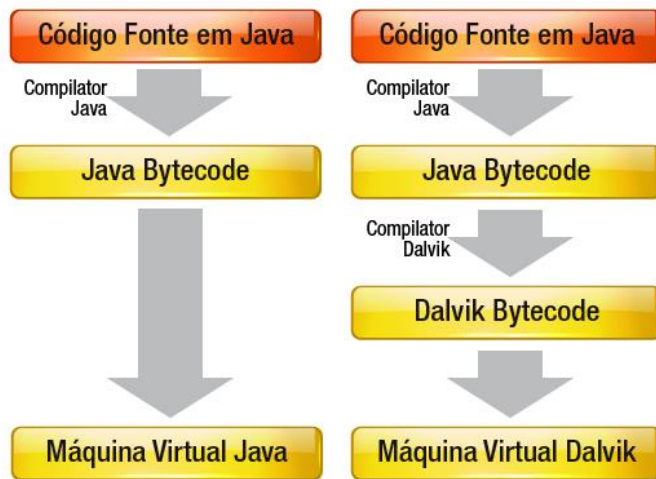


Na prática, qualquer aplicativo executado no Android é interpretado pela Máquina virtual Dalvik, o que permite, ao menos em tese, a execução do código independentemente do dispositivo de *hardware*.



O lado negativo desta abordagem fica por conta da perda de performance na execução das aplicações, já que a necessidade de realizar uma etapa adicional no processamento ocasiona um maior consumo de recursos do sistema.

A figura abaixo demonstra o esquema de funcionamento do modelo proposto e apresenta o *overhead* causado pela necessidade de converter os aplicativos para a máquina virtual Dalvik antes de realizar a sua execução.



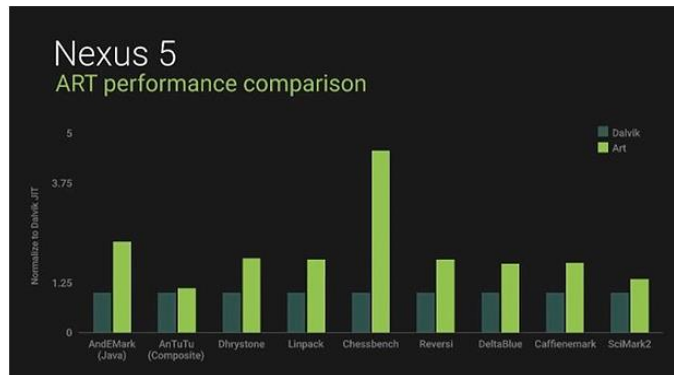
A obrigatoriedade de se realizar uma conversão extra do código dos aplicativos resolveu a questão da compatibilidade, mas por outro lado, apresentou-se como um grande problema, sobretudo quando da execução de aplicativos mais pesados, já que o consumo exagerado de recursos causado pelo procedimento chegava a causar diversos travamentos no Sistema Operacional, o que fez com que o Google começasse a pensar em uma alternativa para substituição do modelo.

10

Logo após o lançamento da versão 4.4. do Android, o Google anunciou que havia desenvolvido uma nova versão de compilador em tempo de execução chamado **ART (Android Runtime)**, e que este novo modelo poderia eventualmente vir a substituir o Dalvik em próximas versões do SO. Os resultados dos testes foram tão promissores que, com o lançamento do Android Lollipop 5, o Android Runtime se tornou o ambiente padrão de execução, substituindo por completo o Dalvik.

Diferentemente do Dalvik, que fazia a compilação do código da aplicação apenas pouco antes desta ser iniciada, conceito definido como JIT (*Just in Time*), o ART compila os bytecodes no momento da instalação da aplicação. Esta função, chamada de Ahead-Of-Time (AOT), se mostrou muito mais eficiente do que o modelo anterior, promovendo uma melhoria do desempenho do sistema e uma redução no consumo da bateria do equipamento. Além disso, o ART compilava o código nativo java diretamente para linguagem de máquina, eliminando a necessidade de carregar uma Máquina Virtual para cada nova aplicação iniciada do dispositivo, modificação esta que reduzia consideravelmente o uso de recursos do SO.

Como já comentado, os testes realizados apontaram uma considerável melhoria de performance na execução de aplicações de usuário com a mudança do Dalvik para o ART. Alguns dados relacionados a estes testes podem ser visualizados na figura abaixo.



Apesar da melhoria de performance, a adição do ART também trouxe consigo algumas características negativas.

Características negativas

A principal foi o fato de que, em média, a conversão prévia do código da aplicação para o código nativo da máquina realizado pelo ART fez com que o aplicativo passasse a ocupar de 10% a 20% mais espaço de armazenamento do que ocupava anteriormente. Outro ponto negativo observado foi o de que como a compilação do ART é realizada no momento da instalação da aplicação, percebeu-se um incremento no tempo necessário para inclusão de novos aplicativos no SO, causando um impacto negativo na usabilidade do Sistema Operacional.

11

c) *Framework* da Aplicação

As aplicações de usuário executadas no sistema Android não interagem diretamente com as bibliotecas nativas do sistema localizadas na camada de Bibliotecas e Ambiente de Execução, e sim com serviços localizados na camada de *framework* da aplicação, já que é nesta camada que estão agrupados programas que gerenciam em alto nível funções básicas do dispositivo, como o gerenciamento de recursos e das chamadas de voz.

Dentre os principais **blocos de funções** presentes na camada de *framework* de aplicação, pode-se destacar:

<u>Activity Manager</u>	Controla o ciclo de vida de todas as aplicações executadas no dispositivo.
<u>Content Provider</u>	Encapsula os dados que são compartilhados, a exemplo da agenda de contatos.
<u>Notification Manager</u>	Atua no gerenciamento de eventos, como a chegada de mensagens de texto ou de alertas da agenda de compromissos.
<u>Telephony Manager</u>	Gerencia todas as chamadas de voz.
<u>Resource Manager</u>	Gerencia os diversos tipos de recursos utilizados pelas aplicações de usuário.
<u>Location Manager</u>	É responsável pelo gerenciamento da localização do dispositivo, para isso utiliza itens como o GPS, as torres de celular e o WI-FI.

d) Aplicações

Esta é a camada *software* que está localizada no topo da pilha da estrutura interna do Sistema Operacional Android. Este é o espaço responsável por prover o ambiente de execução das aplicações de usuário, a exemplo de:

• SMS/MMS	Aplicativo responsável por gerenciar o envio e recebimento de mensagens de texto por parte do usuário.
• IM	Também chamado de Instant Messenger, é utilizado para troca de mensagens instantâneas entre usuários.
• Browser (Navegador)	Aplicativo que provê a interface necessária para navegação por páginas da internet.
• Câmera	Permite a utilização do dispositivo para captura de imagens ou gravação de vídeos.
• Contacts (Contatos)	Efetua a gerência de todos os contatos do usuário armazenados no dispositivo.
• E-mail	Aplicativo que prove a interface para utilização do correio eletrônico.
• Media Player	Permite a execução de áudio e vídeo no dispositivo do usuário.
• Álbuns	Gerencia os álbuns e imagens armazenadas no dispositivo do usuário.

RESUMO

Por muitos anos, os sistemas operacionais foram pensados prioritariamente para as estações de trabalho dos usuários ou para operacionalizar serviços corporativos. Entretanto, com a miniaturização dos componentes e o aumento do poder de processamento dos dispositivos portáteis, percebeu-se um crescimento exponencial dos estudos relacionados com a computação móvel. Foi neste contexto que o Android foi proposto como um Sistema Operacional voltado ao uso em celulares inteligentes, tablets e outros dispositivos móveis.

A primeira versão comercial do Android foi lançada em fevereiro de 2008, embarcada no T-Mobile G1, *smartphone* desenvolvido pela HTC. Nestes sete anos de desenvolvimento, foram lançadas diversas novas versões do SO, culminando com o Android 5, também chamado de Lollipop, que foi revelada em Outubro de 2014 com o objetivo de ser o maior lançamento desde a origem do Sistema Operacional do Google. Uma das principais inovações da versão foi a inclusão de uma interface de usuário completamente remodelada, denominada “material”, que trouxe consigo a filosofia de atuar na integração de diferentes plataformas.

Internamente, o Android se organiza em função de quatro camadas que se agrupam em formato de uma pilha de *software*. A camada mais abaixo é a do Kernel Linux, e desempenha as funções de mais baixo nível do Sistema Android, como o gerenciamento dos processos, da memória, da comunicação e o gerenciamento do sistema de arquivos. Acima desta camada se encontra a de Bibliotecas nativas do Sistema e o ambiente de execução do Android (*Android Runtime*). As bibliotecas nativas são escritas nas Linguagens C e C++ e habilitam os dispositivos a lidar com diferentes tipos de dados, já o ambiente de execução tem por finalidade prover os requisitos necessários para promover a execução de aplicativos e usuário. Atualmente, para realizar esta tarefa, o Android utiliza o ART (*Android Runtime*).

As aplicações de usuário, que estão no topo das camadas de *software*, executadas no sistema Android não interagem diretamente com as bibliotecas nativas do sistema localizadas na camada de Bibliotecas e Ambiente de Execução, e sim com serviços localizados em uma camada intermediária chamada de framework da aplicação. É nesta camada que estão agrupados programas que gerenciam em alto nível funções básicas do dispositivo, como o gerenciamento de recursos e das chamadas de voz.

UNIDADE 4 – IMPLEMENTAÇÕES DE SISTEMAS OPERACIONAIS

MÓDULO 4 – SISTEMA OPERACIONAL IOS

01

1 - HISTÓRICO DO IOS

O iOS é o sistema operacional desenvolvido pela Apple para equipar os seus dispositivos móveis, a exemplo do *smartphone* iPhone, do tablet iPad e do reproduutor de mídia iPod. O SO provê a interface necessária para o desenvolvimento de novas aplicações e gerencia todo o *hardware* dos dispositivos da Apple. Além disso, atua como um integrador de diversos outros componentes fornecidos pela fabricante, como a interface do telefone, do e-mail e de navegação na web.

Uma das principais características que diferencia o iOS dos outros sistemas operacionais apresentados é que ele não é licenciável, ou seja, a fabricante não permite que o Sistema Operacional seja adquirido e instalado em equipamentos de terceiros. Por conta desta vinculação obrigatória ao *hardware* da Apple, muitos pesquisadores classificam o iOS como um Sistema Operacional embarcado.

A vinculação do sistema operacional a poucas plataformas de *hardware* auxilia no aumento da sua estabilidade e na maximização da capacidade dos dispositivos, já que a interface de desenvolvimento do iOS é construída levando em consideração características específicas dos componentes de *hardware* dos equipamentos da Apple, ao contrário do seu principal concorrente, o Android, que é construído para ser executado em uma infinidade de plataformas.



02

A primeira versão deste sistema operacional, lançada em 2007, ainda não tinha um nome específico, sendo chamado de **iPhone OS 1** por equipar a primeira versão do *smartphone* da Apple. Dentre características chave do iPhone OS 1, pode-se destacar a capacidade *multitouch* e um processamento ininterrupto dos requisitos multitarefa (*seamless multitasking*).



A versão 2 do iPhone OS, publicada em 2008, surgiu com o objetivo de equipar os smartphones com tecnologia 3G, diferentemente da versão anterior, que operava em equipamentos 2G. A principal novidade advinda com a versão 2 do iPhone OS foi o lançamento da loja virtual da Apple – a “*Apple App Store*”, que iniciou um modelo de se aferir lucro com os dispositivos de *hardware* através da intermediação da venda de aplicativos de terceiros.

O lançamento da versão 3 do iPhone OS finalmente tornou possível a utilização dos tão conhecidos comandos de copiar, cortar e colar em dispositivos móveis. Além disso, passou a permitir que aplicativos de terceiros utilizassem o painel de notificações do Sistema Operacional, que era utilizado anteriormente apenas para tarefas do próprio SO. Outra inovação advinda com esta versão foi a possibilidade de se comprar vídeos, músicas e livros diretamente dos dispositivos móveis, como o iPhone ou iPod Touch.

03

A próxima versão marcante do Sistema Operacional iOS foi a 3.2, caracterizada sobretudo por equipar a linha de tablets da Apple, os iPads. Algumas mudanças foram realizadas com o objetivo de prover uma melhor experiência para o usuário em um dispositivo com uma tela maior, o que acabou resultando na criação de novos painéis e barras laterais. É importante ressaltar que a versão 3.2 equipou apenas os iPads, representando a primeira segmentação da plataforma do iOS.



O iOS 4, publicado em 2010, iniciou a sequência atual de nomes utilizados nos sistemas operacionais dos *smartphones* da Apple, substituindo o então iPhone OS. A versão 4 do SO foi lançada com a promessa de aumentar a performance do uso do iPhone através da utilização do conceito de multitarefa no dispositivo de *hardware*.

Já o iOS 5, que foi lançado em conjunto com o iPhone 4S, trouxe como principal inovação o aplicativo Siri, assistente pessoal da Apple que utiliza recursos de linguagem natural para melhorar a experiência do usuário, respondendo perguntas e recomendando ações. O iOS 6, por sua vez, trouxe a possibilidade de utilização do FaceTime pela rede celular, promoveu uma integração mais íntima do SO com o Facebook e incrementou os recursos de acessibilidade, provendo mais recursos para pessoas com deficiências visual, auditiva ou de mobilidade.

04

A versão 7 do iOS foi marcada como uma mudança radical no paradigma estético até então utilização no sistema operacional da Apple. O visual mais limpo, com a remoção de gráficos desnecessários e modificação das fontes utilizadas na tela de trabalho, fez parecer que o iOS 7 era um sistema completamente novo. O lançamento mais recente do sistema iOS foi a sua oitava versão, disponibilizada no ano de 2014 em conjunto com o iPhone 6. Clique aqui para conhecer as principais funcionalidades do iOS 8.

Mesmo com as novas funcionalidades, o iOS 8 foi considerado uma das versões com maior quantidade de erros na história do SO. Dados da pesquisa realizada pela Wired apontam que o percentual de travamentos de aplicativos no iOS 8 é 60% maior do que no seu antecessor, o iOS 7.

Acredita-se que o aumento da quantidade de erros é decorrente do aumento do portfólio de dispositivos comercializados pela Apple, o que teria aumentado a complexidade necessária para o desenvolvimento do Sistema Operacional. Ao invés de apenas uma versão do iPhone, como aconteceu na primeira versão do iOS, atualmente a Apple vende quatro diferentes modelos de seu telefone inteligente, fora o fato de que o iOS ainda tem que dar suporte a outros seis modelos legados. Em relação ao iPad a situação não é diferente, são cinco modelos atualmente comercializados e oito suportados.



Wired

Saiba mais sobre a pesquisa em <http://www.wired.com/2014/11/ios-8-buggiest>

Clique aqui

Dentre as principais funcionalidades do iOS 8, pode-se citar:

- Busca inteligente de fotos e álbuns e melhoria nas ferramentas de edição de imagem.
- Melhoria no sistema de mensagens do SO, com a possibilidade de adicionar fotos ou vídeos, a exemplo do que já é possível nas principais ferramentas de mensagens instantâneas do mercado.
- Um novo e aprimorado teclado inteligente, que facilita a digitação com sugestão de palavras relacionadas ao contexto levando em consideração, por exemplo, se o usuário está escrevendo um e-mail ou uma mensagem de texto.
- Uma série de aplicativos para monitoramento da saúde e boa forma do usuário que aproveitam os novos sensores do iPhone, a exemplo do sensor de batimentos cardíacos.
- Novo sistema de notificações, melhorando a experiência do usuário na utilização do SO.
- Possibilidade de compartilhamento familiar das compras feitas no iTunes, iBooks e AppStore.

O serviço *Handoof*, que permite a utilização conjunta dos diferentes dispositivos da Apple.

05

2 - ESTRUTURA DO IOS

A arquitetura do IOS é segmentada em quatro camadas principais, que atuam como intermediárias na comunicação dos aplicativos de usuário com o *hardware* do equipamento. A implementação destas camadas pode ser visualizada na figura abaixo.



Cada uma das camadas tem funções específicas e se comunicam umas com as outras através da utilização de interfaces próprias. Uma análise detalhada de cada uma destas camadas será realizada nas próximas seções.

06

a) Camada Cocoa Touch

A camada Cocoa Touch agrupa uma série de *frameworks* que têm como principal funcionalidade definir a aparência das aplicações desenvolvidas e executadas em modo usuário. Isto acontece porque, diferentemente de sistemas operacionais como Windows e Linux, que permitem o desenvolvimento de aplicações em uma série de outros modelos e linguagens, a criação de aplicações para iOS devem seguir um padrão de modelagem e escrita definidos pela Apple.

Dentre as funcionalidades providas às aplicações de usuário por esta camada, destacam-se o suporte a diversas tecnologias, como a utilização de computação multitarefa, controle de entradas baseadas no toque da tela, o uso de notificações do sistema e de uma série de outras funções próprias do iOS, como o *Handoff*, o *AirDrop*, os *Storyboards* e o reconhecimento de gestos, por exemplo.

Os **Storyboards** permitem ao desenvolvedor criar toda a interface da aplicação em um só lugar, de modo que possa visualizar todas as telas e controles possíveis e entender como eles funcionam juntos.

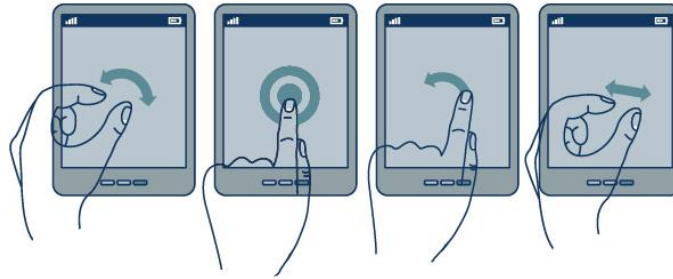
Já o **Handoff** é uma funcionalidade que permite ao usuário da aplicação iniciar uma atividade em um dispositivo, o iPhone por exemplo, mudar para um outro, como o iPad, e continuar a atividade neste outro dispositivo a partir de onde parou no primeiro.



07

O **AirDrop** é um serviço de compartilhamento baseado em wi-fi que permite aos usuários compartilhar fotos, vídeos, sites e uma série de outros recursos com pessoas próximas que também possuam um dispositivo executando iOS.

O reconhecimento de gestos, por sua vez, é uma função provida pelo iOS que permite o reconhecimento e detecção de gestos comuns executados pelos usuários na tela do dispositivo, como deslizar o dedo ou pressionar demoradamente a tela.



08

b) Camada de Mídia

Nesta camada estão localizadas as tecnologias relacionadas a vídeo, gráficos e áudio, providas pelo Sistema Operacional para viabilizar a implementação de aplicativos de usuário. As tecnologias de gráficos do iOS provêm suporte a uma vasta gama de ferramentas que permitem, dentre outras funções, a exibição de imagens personalizadas na tela do dispositivo.

A **tecnologia de áudio** permite aos desenvolvedores incluírem a habilidade de reproduzir e gravar áudio de alta qualidade ou gerenciar conteúdo MIDI em aplicações em desenvolvimento.

Já as **tecnologias de vídeo** provêm suporte ao gerenciamento de conteúdo estático de vídeo ou execução de streamings de vídeo diretamente da internet.

Dentre os *frameworks* típicos disponibilizados por esta camada, pode-se destacar:

- Core Graphics Framework;
- Core Animation Framework;
- Open GL;
- Media Player Framework;
- AV Foundation Framework.

Core Graphics Framework

É a Engine de desenho utilizada pelas aplicações iOS, provendo suporte a criação de vetores 2D e renderização de imagens.

Core Animation Framework

Provê a tecnologia base para o provimento de suporte ao uso de animação em aplicações desenvolvidas para o iOS.

Open GL

Gerencia a renderização de imagens 2D e 3D utilizando interfaces de aceleração de *hardware*.

Media Player Framework

Provê ao usuário acesso a biblioteca do iTunes e suporte a execução de playlists e trilhas de áudio.

AV Foundation Framework

É uma interface para o gerenciamento da execução e gravação de áudio e vídeo em dispositivos que executam o iOS.

09**c) Camada de Serviços Centrais**

Esta camada agrupa uma série de serviços fundamentais de sistema que precisam ser consumidos pelos aplicativos de usuário. Uma das principais funcionalidades provida por esta camada é o suporte ao *Peer-to-peer*, que permite a comunicação ponto a ponto entre dispositivos.

A camada de serviços centrais provê, ainda, o serviço de armazenamento na nuvem privada da Apple, a *iCloud*.

Na prática, o uso desta funcionalidade permite aos aplicativos de usuário escrever documentos ou outro tipo de dado em uma localização central e remota, passando este o conteúdo a estar automaticamente acessível para uma diversidade de dispositivos, como computadores pessoais ou outros dispositivos executando o iOS.

Dois outros serviços providos por esta camada são o de proteção de dados e o suporte ao padrão XML. O primeiro se caracteriza por permitir que as aplicações trabalhem com dados sensíveis do usuário utilizando elementos de criptografia presentes em alguns dispositivos da Apple. Já o segundo provê um *framework* para recuperação e processamento de elementos em documentos XML.

10**d) Camada de Núcleo ou Camada Central do SO**

Esta camada é responsável por gerenciar todas as características de baixo nível do sistema iOS. Em função de sua complexidade e importância, os serviços disponibilizados nesta camada são segmentados em seis outras seções:

Esta camada é responsável por gerenciar todas as características de baixo nível do sistema iOS. Em função de sua complexidade e importância, os serviços disponibilizados nesta camada são segmentados em seis outras seções:

- Accelerate Framework,
- Núcleo Bluetooth,

- External Accessory framework,
- Framework Genérico de Segurança,
- Framework de Segurança e
- Seção do Sistema (System).

O **Accelerate Framework** contém as interfaces utilizadas para realizar a execução de aplicativos que demandam grande uso de recursos computacionais, como o processamento de imagens ou determinados cálculos de álgebra linear. O **Núcleo Bluetooth**, por sua vez, contém o framework utilizado para realizar a conexão e desconexão do sistema com outros dispositivos Bluetooth. Já o **External Accessory Framework** é utilizado para prover a comunicação do dispositivo com qualquer acessório conectado ao mesmo.

Tanto o **Framework Genérico de Segurança** dos serviços quanto o **Framework de Segurança** oferecem elementos de segurança padrão do SO que são utilizados pelos aplicativos que são executados sobre o iOS. Dentre estas funcionalidades, estão o gerenciamento de certificados, de chaves públicas e privadas, uso de políticas de confiança, além do suporte a criptografia simétrica.

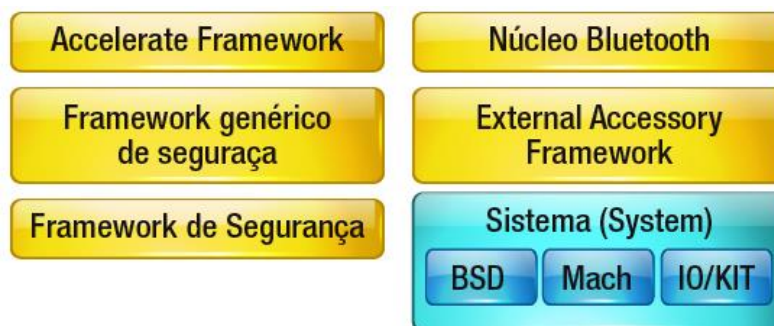
Como já comentado, a seção de **Sistema** é vista como a parte mais importante da camada central do SO, desta forma, a mesma será melhor detalhada em uma seção a parte.

11

e) Seção do Sistema (System)

A seção de sistema agrupa as funções de mais baixo nível do sistema iOS, sendo vista, por vezes, como o item central do sistema operacional. Este elemento segue um padrão próprio de arquitetura, que é muito similar à encontrada no sistema MacOS, e que se baseia em três componentes principais:

- a camada Mach, derivada do núcleo do sistema operacional de mesmo nome desenvolvido pela Universidade Carnegie Mellon,
- a camada BSD, que provê uma interface de programação aos programadores de aplicações e que permite o acesso a elementos do sistema,
- e a camada IO/Kit, que provê um *framework* orientado a objeto aos drivers de dispositivo. A representação esquemática deste modelo é apresentada na figura abaixo.



Camada Mach

A **camada Mach** representa o centro da estrutura do kernel do iOS. Na prática esta camada atua provendo serviços de baixo nível para camadas de mais alto nível, como a BSD e a I/O Kit, abstraindo toda a complexidade do *hardware*. Além desta funcionalidade, a camada Mach também é a responsável pelo agendamento das Threads, dando suporte ao multiprocessamento simétrico (SMP) no Sistema Operacional IOS. Toda a atividade de gerenciamento de memória no IOS também é executada pela camada Mach, que coordena a alocação dos espaços de endereçamento físico e provê uma interface de acesso para as camadas de mais alto nível.

Camada BSD

A **camada BSD**, por sua vez, está localizada acima da camada Mach e atua prioritariamente na implementação de uma série de funções base para funcionamento do Sistema Operacional, utilizando, para isso, chamadas a interfaces providas pela camada Mach. Como o próprio nome induz a pensar, parte do código utilizado na camada BSD é oriundo do kernel do sistema operacional FreeBSD.

Camada IO/KIT

A camada I/O Kit se configura como um framework orientado a objetos para o desenvolvimento de drivers de dispositivo e outras extensões do kernel do sistema. Na prática, I/O Kit utiliza chamadas a funções de baixo nível providas pela camada Mach para implementar a abstração de *hardware* que será utilizada pelos desenvolvedores para criação dos novos drivers de dispositivo.

12

No sistema IOS, todo o código da camada BSD é executado em modo privilegiado, ou modo kernel, e não como uma tarefa de ambiente de usuário. Isto faz com que a camada BSD não atue em um espaço de memória próprio e protegido, e sim compartilhando o mesmo espaço de endereçamento utilizado pelas camadas Mach e I/O Kit. Dentre as principais funções da camada BSD, pode-se citar:

- **Chamadas de Sistema** – todas as chamadas de sistema do IOS são implementadas na camada BSD, desta forma, sempre que algum aplicativo de usuário, ou qualquer outro *software* que estiver sendo executado em modo usuário, necessitar de acesso a serviços básicos ou de baixo nível providos pelo kernel do sistema, ele irá efetuar uma chamada a API provida pela camada BSD.
- **Rede** – É o maior subsistema existente na camada BSD, tendo como principal atribuição gerenciar a maior parte dos elementos de rede do sistema operacional, como os detalhes de comunicação dos sockets e a implementação dos principais protocolos, como o TCP/IP, o DHCP, o UDP e o ICMP. É importante ressaltar que os drivers de rede de baixo nível não estão localizados nesta camada, e sim na I/O Kit.
- **Gerenciamento de Processos e de usuário** – provê a interface para o gerenciamento de usuários, com a criação dos identificadores de usuário (UID) e grupo (GID), e de processos, com a manutenção dos identificadores de processo (PID).
- **Gerenciamento de Memória** – utiliza chamadas a funções da camada Mach para prover

facilidades para alocação de memória, como a utilização do *Unified Buffer Cache (UBC)*.

13

RESUMO

O iOS é o sistema operacional desenvolvido pela Apple para equipar os seus dispositivos móveis, a exemplo do smartphone iPhone, do tablet iPad e do reprodutor de mídia iPod. A primeira versão deste sistema operacional, lançada em 2007, ainda não tinha um nome específico, sendo chamado de iPhone OS 1, por equipar a primeira versão do smartphone da Apple. Já a versão mais recente, o iOS 8, foi lançado em 2014 em conjunto com o iPhone 6.

A arquitetura do IOS é segmentada em quatro camadas principais, que atuam como intermediárias na comunicação dos aplicativos de usuário com o *hardware* do equipamento. São elas:

- A camada *Cocoa Touch* agrupa uma série de frameworks que têm como principal funcionalidade definir a aparência das aplicações desenvolvidas e executadas em modo usuário.
- A camada Mídia agrupa as tecnologias relacionadas a vídeo, gráficos e áudio, providas pelo Sistema Operacional para viabilizar a implementação de aplicativos de usuário.
- A camada de serviços centrais congrega uma série de serviços fundamentais de sistema que precisam ser consumidos pelos aplicativos de usuário, a exemplo do peer-to-peer.
- A Camada de Núcleo ou Camada Central do SO é responsável por gerenciar todas as características de baixo nível do sistema iOS.

Em função da complexidade e importância inerente a Camada Central do SO, os serviços disponibilizados nesta camada foram segmentados em seis outras seções.

- O Accelerate Framework, que contém as interfaces utilizadas para realizar a execução de aplicativos que demandam grande uso de recursos computacionais.
- O Núcleo Bluetooth, que contém o framework utilizado para realizar a conexão e desconexão do sistema com outros dispositivos Bluetooth.
- O External Accessory framework, que é utilizado para prover a comunicação do dispositivo com qualquer acessório conectado ao mesmo.
- O Framework Genérico de Segurança dos serviços e o Framework de Segurança, que oferecem elementos de segurança padrão do SO.
- A seção de Sistema, que agrupa as funções de mais baixo nível do sistema iOS, sendo vista, por vezes, como o item central do sistema operacional.