

3 Ambientes Celulares

O avanço da tecnologia é imenso a cada ano que passa, de modo que, nas últimas décadas, novos conceitos foram introduzidos no cotidiano das pessoas.

Dentre as principais evoluções estão o aumento da capacidade dos processadores, juntamente com a diminuição do consumo de energia, o aumento na capacidade de armazenamento e o avanço de várias tecnologias baseadas na comunicação por rádio-freqüência, além da diminuição do tamanho físico dos dispositivos.

Galgada nesses avanços, há hoje uma grande difusão de aparelhos portáteis com os mais variados fins, como *notebooks*, *tablet pcs*, tocadores de mp3, câmeras digitais e telefones celulares.

Dentre esses dispositivos, o que notoriamente se difundiu na grande população foi o telefone celular, chegando a aproximadamente 90 milhões de aparelhos no Brasil, principalmente devido a sua natureza essencialmente de comunicação pessoal e também móvel.

Aliadas a essas características inerentes aos telefones celulares, e ao contínuo avanço da tecnologia, outras funcionalidades foram incluídas neste tipo de dispositivo, iniciando com aplicações de PIM (*Personal Information Management*), entre outras, até a sua utilização como plataformas de jogos e aplicativos diversos, e a sua integração com outros dispositivos como tocadores de MP3 e câmeras de foto e vídeo.

Com essa nova realidade e o foco do desenvolvimento em aparelhos móveis cada vez maior, vemos também o surgimento de aparelhos com grande capacidade gráfica, como os jogos portáteis PSP (*Player Station Portable*) da Sony, NintendoDS da Nintendo e o NGage da Nokia.

Muitos dos processadores empregados nestes equipamentos já se igualam à velocidade de processamento de antigos PCs, alguns chegando a centenas de MHz, e já há o surgimento de processadores gráficos desenhados exclusivamente para o uso em dispositivos que tenham restrições de tamanho e consumo de energia, como por exemplo, os produzidos pela Nvidia e ATI.

Todavia, a maioria dos celulares ainda não apresenta uma grande capacidade de processamento, e nem processadores especializados em de imagem.

Dentre os telefones celulares, os que apresentam grande capacidade de processamento foram integrados aos chamados PDA (*Personal Digital Assistant*) e são conhecidos como *Smart Phones*. Estes não atingem uma grande fatia do público devido ao seu elevado custo em relação aos celulares de desempenho inferior.

A seguir serão apresentados os principais ambientes de desenvolvimento para as plataformas celulares presentes no mercado.

3.1. Plataformas de desenvolvimento para celulares

Atualmente, com a separação entre celulares considerados *Smart Phones* e os celulares com menor capacidade de processamento e armazenamento, temos no mercado duas grandes plataformas mais difundidas em cada uma dessas classes.

Para celulares considerados *Smart Phones* duas plataformas que possibilitam o desenvolvimento de aplicações se destacam: o Symbian e o Windows Mobile.

Estas plataformas são, na verdade, sistemas operacionais mais completos que os presentes nos celulares com menor capacidade de processamento, mas também provêm *frameworks* e APIs para o desenvolvimento de aplicações. Suas características serão apresentadas mais a frente.

Existem celulares que utilizam uma distribuição de Linux como sistema operacional, apesar de não serem difundidos no mercado, tais como os Motorolas A760 e ROKR E2. Porém, eles apresentam como possibilidade de plataforma de desenvolvimento o J2ME (*Java 2 Platform Micro Edition*).

Para os celulares com uma menor capacidade de processamento que os *Smart Phones*, há uma variedade de sistemas operacionais, normalmente desenvolvidos e exclusivos de cada fabricante. Alguns desses sistemas disponibilizam seus próprios *frameworks* e APIs, todavia, o foco deste tópico são as duas plataformas que possibilitam uma maior convergência: BREW (*Binary Runtime Environment for Wireless*), criado e mantido pela Qualcomm Incorporated e o já citado J2ME, amparado pela empresa Sun Microsystems.

3.1.1. Symbian OS

Symbian é um avançado sistema operacional de padrão aberto para telefones celulares. Possui um núcleo multi-tarefas e *multithreaded*, um *framework* para interface gráfica, componentes para serviços de transferência e processamento de dados, engines para aplicações e gerenciamento de dados pessoais e comunicação sem fio.

Esse sistema conta com bibliotecas associadas e implementações de referência de ferramentas padrão, produzidas pela Symbian Ltd.

O Symbian tem uma arquitetura modularizada, e os fabricantes que desejarem utilizá-lo em algum produto podem adicionar, remover ou até mesmo modificar alguns componentes. Isso significa que o um software escrito para um telefone específico pode ou não ser compatível com outro modelo. Além da possível incompatibilidade entre dispositivos, ainda pode haver incompatibilidade entre diferentes versões desse sistema.

Muitos dos celulares que possuem o Symbian como sistema operacional, também disponibilizam o J2ME como plataforma de desenvolvimento, como por exemplo a chamada série 60 da Nokia, licenciada também para outros fabricantes como Samsung e Siemens.

Além disso, como retro mencionado, o Symbian possui uma coleção de bibliotecas e APIs, específicas para o desenvolvimento de aplicações que executem diretamente acima do sistema operacional.

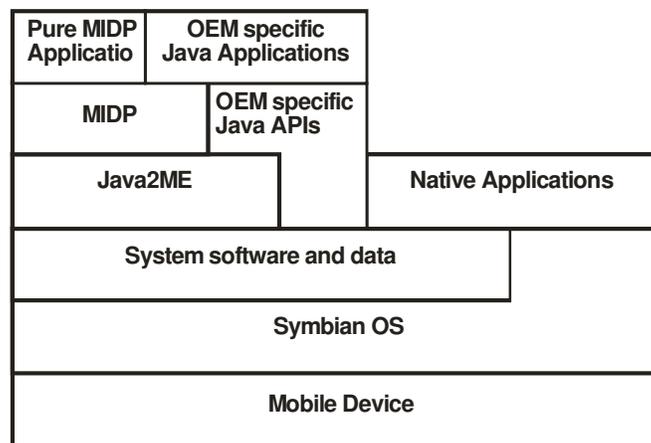


Figura 3 - Uma arquitetura típica de um telefone celular utilizando o Symbian OS – Retirado de Symbian Application Delevopment White Paper, Sonera Lab - 2003

O sistema em questão foi desenvolvido para funcionar em aparelhos com as mais diversas arquiteturas e interfaces com o usuário. Para se adaptar a

essas diferenças de interfaces, até a sexta versão (Symbian OS v 6), ele contava com categorias chamadas de *Device Family References Desings* (DFRD).

Essas categorias eram divididas da seguinte forma: *Quartz*, para aparelhos que possuem interface operada por canetas e *touch screen* como PDAs; *Crystal*, para aparelhos que possuem teclados *QWERTY*; e *Pearl*, para aparelhos que possuem apenas o teclado numérico, presente nos telefones celulares.

A partir da versão 7, essa classificação foi descartada e agora é utilizada uma abordagem por plataforma de interface com o usuário, onde cada uma provê uma interface para o sistema operacional e um conjunto de aplicações para a execução de tarefas como envio de mensagem, navegação na Internet, telefonia, multimídia, comunicações sem fio e PIM. Essas aplicações normalmente utilizam outras aplicações genéricas, providas pelo grupo que desenvolve o Symbian OS.

Cada plataforma dessa deve dar suporte para o desenvolvimento e instalação de aplicações feitas por terceiros, escritas em código nativo C++. Atualmente existem quatro plataformas e, assim, quatro conjuntos de ferramentas de desenvolvimento: Nokia Series 60, Nokia Series 80 (antiga “9200 communicator series”), Nokia Series 90 e UIQ.

Para ajudar no desenvolvimento de aplicações, várias IDEs estão disponíveis, inclusive com integração com os SDK (conjunto de ferramenta de desenvolvimento) existentes. Pode-se, virtualmente, utilizar qualquer ambiente que possibilite o desenvolvimento de aplicações em código C++. Apenas para citar alguns, é possível o uso do Eclipse, em conjunto com alguns plugins como o Carbide.c++, além do Visual Studio e do Borland C++ Mobile Edition.

Além dessas ferramentas, ainda foram criados e adaptados alguns ambientes de execução, possibilitando a execução de aplicações criadas com outras plataformas e linguagens como Python, VB.NET e Flash (Adobe Systems).

3.1.2. Microsoft Windows Mobile

O chamado Windows Mobile é uma configuração do Windows CE para os originalmente chamados de *Smart Phones* e para PDAs.

Esse sistema operacional começou a sua história como o sistema de um simples organizador pessoal, e continuou evoluindo, permitindo, assim como o Symbian OS, que ele pudesse ser adaptado para diferentes perfis de hardware.

O desenvolvimento de aplicações em primeira instância só poderia ser feito em C e C++, através de um subconjunto da *Win 32 API*, seguindo o mesmo estilo de programação das aplicações nativas para os sistemas operacionais para computadores pessoais desenvolvidos pela Microsoft.

Com a evolução do Windows CE, este foi ganhando poder de processamento e estabilidade, e novos componentes e funcionalidades foram incluídas, como a possibilidade de desenvolvimento de aplicações baseadas no modelo de componentes COM. Atualmente conta também com o *Compact .NET Framework 2.0*.

A atual versão, Windows Mobile 5.0, baseada no Windows CE 5.0, conta com alguns componentes inovadores, como a inclusão do *Microsoft Direct3D Mobile (D3DMobile)* dando suporte à aceleração de gráficos DirectX e o *Microsoft SQL Mobile*, que dá acesso a dados de uma base de forma transparente, permitindo que aqueles transmitidos para o dispositivo sejam modificados e depois sincronizados.

As ferramentas de desenvolvimento disponíveis são o Microsoft Embedded Visual C++, para aplicações nativas em C++, e as IDEs Visual Studio 2003 e 2005, para aplicações utilizando o *.NET framework*, podendo ser utilizadas quaisquer linguagens compatíveis com este *framework* da Microsoft.

Devido a sua natureza, a maioria das aplicações desenvolvidas comercialmente para a plataforma Windows Mobile tem um caráter corporativo.

3.1.3. BREW

O BREW (*Binary Runtime Environment for Wireless*), diversamente dos ambientes descritos anteriormente, não se traduz em um sistema operacional para pequenos dispositivos, ele é uma plataforma para o desenvolvimento de aplicações, independente do sistema operacional utilizado.

Essa plataforma foi criada pela empresa Qualcomm Incorporated, conhecida mundialmente por soluções para tecnologias sem fio, e mais notoriamente por deter o maior número de patentes para o protocolo de transmissão CDMA, comandando o mercado de telefonia celular baseado neste protocolo.

O BREW faz parte de um chipset, tendo sua execução diretamente acima da camada de hardware do telefone celular.

Diferentemente das outras plataformas apresentadas neste capítulo, o BREW conta com um sistema comercial de *download* de aplicações totalmente integrado com o sistema de cobrança da operadora de telefonia celular, além da necessidade de que estas sejam assinadas digitalmente para que possam ser executadas no ambiente. Pelo modelo atual, para que sejam disponibilizadas para o público, é necessário que passem por uma bateria de testes, garantindo o seu correto funcionamento. Estes testes podem ser efetuados pela própria operadora ou por terceiros, podendo ser cobrados ou não.

A plataforma, como as descritas anteriormente, também possui um grande conjunto de APIs disponíveis para a interação da aplicação como acesso às funções de telefonia, agenda, criação e reprodução de conteúdo multimídia e acesso a interfaces de comunicação sem fio, apenas para citar algumas.

Apesar de possuir uma grande quantidade de bibliotecas e funções que permitem a interação da aplicação com o hardware do dispositivo, essas dependem de quais componentes do hardware foram expostos na integração do ambiente de execução BREW.

Seu modelo de execução é baseado em eventos, existindo eventos específicos para cada uma das interações que possam ocorrer, tanto entre a aplicação e o usuário, quanto à aplicação e o hardware do telefone, porém, todos os eventos possuem apenas um único ponto de entrada na aplicação, o que pode tornar a implementação, se esta não for realizada cuidadosamente, muito confusa.

A plataforma BREW não possui um sistema de *multithread* que permita que duas aplicações executem, ou simulem, uma execução concorrente. Assim, ciclos de execução que sejam demoradas são terminantemente proibidos, e, se ocorrem, ocasionam o término da execução da aplicação. Para evitar que tal evento ocorra com chamadas interfaces do sistema, normalmente estas disponibilizam callbacks, que, aliados ao sistema de eventos, tornam a execução de algumas tarefas totalmente assíncronas.

Apesar da inexistência de concorrência na execução de aplicações, devido à sua natureza assíncrona baseada em eventos, a partir da versão 2.1 da plataforma BREW é possível que mais de uma aplicação seja executada ao mesmo tempo, podendo se comunicar através de mensagens, mas apenas uma delas tem acesso às interfaces com o usuário.

A plataforma, além de aplicações, possibilita o desenvolvimento de extensões, que podem ser integradas como parte de uma aplicação. O sistema que controla o comércio das aplicações também controla, de forma inteligente,

as extensões necessárias, instalando-as quando preciso. Estas extensões podem ser compartilhadas por mais de uma aplicação. Dentre as extensões disponíveis encontram-se uma para processamento de arquivos XML, uma para a execução de aplicações escritas em Java e outra para a execução de arquivos Flash (Adobe Systems), apenas para citar algumas.

As aplicações BREW podem ser desenvolvidas tanto utilizando a linguagem C, quanto a linguagem C++, mas, apenas o conjunto de funções que fazem parte da biblioteca padrão C estão presentes no celular. Assim, apesar de poder ter uma aplicação escrita em C++, nem todas as funcionalidades da linguagem estão disponíveis, como o sistema de exceções; e os operadores *new* e *delete* necessitam de serem sobrescritos para terem o correto funcionamento.

O sistema de execução também tem uma natureza totalmente dinâmica, e veta terminantemente o uso de variáveis globais ou estáticas em qualquer parte do código.

Outra característica importante do *runtime* é a limitação no tamanho da pilha de execução (variável entre os diferentes modelos), que pode ocasionar erros se a aplicação apresentar muitas chamadas de funções encadeadas ou grande recursividade.

Como IDEs, o desenvolvedor pode utilizar quaisquer ferramentas que sejam capazes de gerar uma biblioteca dinâmica, que pode ser executada como uma aplicação BREW no emulador disponibilizado pela Qualcomm em seu portal de desenvolvimento. Neste portal estão disponíveis plugins e ferramentas de desenvolvimento para serem integradas com o Visual Studio 2003 e 2005, desenvolvidos pela Microsoft.

Para a criação da versão executável, o desenvolvedor precisa compilá-lo utilizando ferramentas, oferecidas sem taxas pela Qualcomm, para o *GNU Cross Compiler* ou o *RealView Compilation Tools for BREW*, o qual necessita de licenciamento, apesar de que a segunda ferramenta, atualmente, gera um executável menor, que economiza espaço no dispositivo do usuário final.

Mesmo sendo possível a criação de aplicações corporativas e de m-commerce, pois dispõe de interface de comunicação de dados e criptografia, a maioria dos produtos comercializados são jogos. Por causa da força que este tipo de aplicação exerceu, assim como no J2ME, a partir de sua versão 2.0, a plataforma conta com um conjunto de funções especializadas para a manipulação gráfica, com implementação de sprites e possibilidade de criação e manipulação de bitmaps em memória, para posteriormente desenhá-los na tela.

Atualmente, o BREW possui interface OpenGL ES, que é um sub conjunto da Open GL, desenhada para ambientes com restrições de memória e processamento, que será comentada mais adiante neste capítulo.

3.1.4. J2ME

Uma outra plataforma para o desenvolvimento de aplicações em dispositivos com grandes restrições de processamento e memória é a *Java Micro Edition* (J2ME).

O J2ME é, na verdade, um conjunto de tecnologias e especificações que definem um ambiente de execução Java para uma classe específica de aparelhos como telefones celulares, PDAs, *Smart Phones*, *set-top boxes* de tv digital, entre outros. Ele é formado por componentes denominados *configurações*, *perfis* e *pacotes adicionais*. As chamadas *configurações* encontram-se na base deste conjunto e detalham a máquina virtual Java e definem as bibliotecas básicas necessárias. A *configuração* mais comumente utilizada em celulares de baixa capacidade de processamento é a *Connected Limited Device Configuration* (CLDC). Esta foi projetada para dispositivos que contam com:

- Um processador de 16 ou 32 bits com pelo menos 6MHz de clock.
- Pelo menos 160 KB de memória não volátil alocada para as bibliotecas e máquina virtual
- Pelo menos 192 KB de memória disponível para a plataforma Java
- Baixo consumo de energia.
- Algum tipo de conectividade, normalmente intermitente e de banda limitada, como as conexões sem fio.

Os perfis complementam as configuration APIs mais específicas, definindo o ciclo de vida da aplicação, interface com o usuário, persistência de dados entre outras propriedades.

A distribuição que une o *J2ME Mobile Information Devices Profile* (MIDP) e o *Connected Limited Device Configuration* tem se tornado ubíqua em telefones celulares, estando presente em milhões de aparelhos.

Por tratar-se de um ambiente bem restrito, como citado anteriormente, muitas características presentes no *Java Standard Edition* (J2SE) tiveram de ser modificadas, principalmente no que tange a segurança. Como os requisitos de

memória, somente das classes responsáveis pela segurança no J2SE, excedem o total de memória disponível para o J2ME MIDP/CLDC, a especificação define um modelo de “*sandbox*”, muito mais simples.

Uma aplicação MIDP, também chamada de *MIDlet*, é executada em um ambiente fechado e só pode acessar as classes e bibliotecas suportadas pelo dispositivo que são expostas pela CLDC, MIDP. Para reforçar essa restrição, a máquina virtual da CLDC tem um *class loader* que só carrega classes desse conjunto predefinido ou classes presentes no pacote da aplicação (JAR). Para evitar qualquer tentativa de burlar essa restrição, o *class loader* não pode ser substituído ou reconfigurado como em uma máquina virtual J2SE, e não há suporte para carregar bibliotecas nativas, nem suporte a *Java Native Interface* (JNI).

Como a CLDC/MIDP foi desenvolvida como um “mínimo denominador comum”, focada nas capacidades dos telefones na época em que a especificação foi definida, é reconhecido que muitos dispositivos têm muito mais recursos de hardware e poderiam suportar uma API mais rica.

Assim, os pacotes opcionais que fazem parte da arquitetura do J2ME adicionam funcionalidade à API básica, provendo interface para que as aplicações possam acessar funções mais específicas dos dispositivos, como acesso a base de dados remotas, acesso a informações de PIM contidas no celular, envio de mensagens instantâneas, renderização de gráficos em três dimensões ou funcionalidades específicas de alguns fabricantes, como a interface gráfica da NOKIA (Nokia UI).

Para o desenvolvimento das aplicações, a plataforma J2ME necessita de um conjunto de ferramentas conhecido como *Wireless Toolkit* e disponibilizado pela Sun Microsystems.

Os fabricantes de telefones celulares normalmente provêem conjuntos de ferramentas que possibilitam a utilização de bibliotecas específicas de cada uma e o uso de emuladores para cada modelo de aparelho.

O J2ME é uma plataforma derivada do *Java Standard Edition*, inclusive compartilhando o mesmo compilador, praticamente todas as IDEs disponíveis para a criação de aplicações Java podem ser utilizadas. Além disso, a maioria possui extensões exclusivas para facilitar o desenvolvimento das *Midlets*.

3.2. Processamento Gráfico

Uma parte importante na qualidade das aplicações desenvolvidas para os telefones celulares é a sua apresentação gráfica. Abaixo é apresentada uma breve discussão sobre o processamento gráfico nas plataformas citadas anteriormente.

As bibliotecas e interfaces de programação disponíveis em Symbian e Windows Mobile normalmente já possibilitam grande poder de processamento de imagens, inclusive podendo-se acessar diretamente a memória de vídeo do dispositivo, via aplicação, além de operações de rasterização.

As plataformas Brew e Java, em suas versões mais atuais, apresentam classes e funções específicas para o processamento de bitmaps, com o objetivo de melhorar os efeitos gráficos destinadas a jogos, principalmente manipulação de sprites (escalonamento e rotação e translação).

BREW também possibilita o acesso, indireto, à memória de vídeo, criando uma cópia do conteúdo para uma representação de bitmap, que pode ser modificado pela aplicação, além de possuir algumas operações de rasterização básicas.

Em sua segunda versão (MIDP 2.0) o J2ME também apresenta uma maior manipulação direta de bitmaps, com métodos para a captura dos valores de pixels de uma imagem presente na memória, assim como a criação de imagens a partir de um conjunto de inteiros, que representem os valores ARGB (alpha, red, green, blue) dos pixels dessa imagem.

Mais recentemente, com alguns celulares ganhando poder de processamento e impulsionado por aplicações de visualização de mapas, utilizadas para navegação e localização, estas duas plataformas ganharam suporte a gráficos vetoriais, ambos compatíveis com o formato *Scalable Vector Graphics* (SVG) definido pelo *World Wide Web Consortium* (W3C).

Apesar dessas facilidades, o processamento de imagens nessas plataformas pode tornar-se um grande desafio, devido a grande variedade de configurações de hardware e de tamanhos de tela, além da variedade de memória disponível para a execução das aplicações.

Mesmo que alguns telefones não possuam uma grande capacidade de processamento, já existem interfaces para o desenvolvimento de aplicações e jogos utilizando gráficos em terceira dimensão.

Para as plataformas disponíveis nos celulares de menor desempenho, BREW e J2ME, estão atualmente disponíveis duas interfaces: *OpenGL for Embedded Systems* (OpenGL ES) e *Mobile 3D Graphics* (M3G – JSR 184), respectivamente.

3.2.1. OpenGL ES

O OpenGL ES é definido a partir de um subconjunto de funções do OpenGL, projetado para os computadores pessoais, de forma que tenha um pequeno *footprint* e um baixo consumo de energia, contendo menos instruções e tráfego de dados. Como muitos processadores não possuem suporte para operações com ponto flutuante, foram incluídos funções e um novo tipo básico, para utilizar operações com ponto fixo.

Outra característica dessa especificação é permitir que todas as funções do pipeline possam ser executadas, tanto por software, quanto por um hardware especializado.

Assim, engines podem ser desenvolvidas e utilizadas em celulares que não tem nenhum tipo de aceleração, e mais tarde podem ser utilizadas em celulares que possuam processadores especializados.

O OpenGL ES conta com dois perfis distintos, o comum, projetado para jogos e para ter total funcionalidade de renderização 3D e mapeamento de texturas, e o Safety Critical, que é projetado para ser utilizado em aplicações industriais, como utilização em displays presentes em automóveis e aeronaves, tendo como principais características estabilidade e confiabilidade.

O OpenGL ES também conta com duas vertentes diferentes, a primeira, chamada de 1.X, já conta com as versões 1.0 e 1.1, e é projetada para hardware de funções fixas; a segunda, chamada de 2.X, é projetada para dispositivos mais poderosos, que contam com hardware programável, possibilitando a inclusão de *shaders* utilizando GLSL.

Sua versão 1.0 é baseada na especificação de versão 1.3 para computadores pessoais, assim como o *Mobile 3D Graphics* que será apresentado a seguir. A versão 1.1, tem como base a o OpenGL 1.5.

A plataforma BREW, assim como o *Mobile 3D Graphics*, apresenta interfaces de abstração e bibliotecas complementares as funções expostas pelo OpenGL ES, possibilitando a utilização de modelos gerados externamente através de ferramentas de autoria, e sua manipulação através das aplicações.

3.2.2. Mobile 3D Graphics

O *Mobile 3D Graphics* (M3G) provê um conjunto de métodos para a criação de gráficos em três dimensões na plataforma J2ME, mais especificamente a configuração CLDC/MIDP.

O M3G, também conhecido como JSR 184, assim como o OpenGL ES, também é projetado para se adequar tanto a celulares que não possuam grande capacidade de processamento, operações de ponto flutuante, ou hardware com aceleração gráfica, quanto os que possuam.

Essa API pode ser implementada eficientemente acima do OpenGL ES, facilitando a sua integração com hardware especializado no futuro. Outra especificação foi a não criação de um novo tipo para lidar com as operações que necessitem de ponto flutuante, ou ponto fixo nos hardwares que não possuam suporte a operações deste tipo. Assim, o M3G sempre só pode estar disponível em plataformas com CLDC 1.1, que contem suporte ao tipo de ponto flutuante do J2ME.

Ela conta com dois modos de operação. O chamado *immediate mode* possui funcionalidades parecidas com as providas pelo OpenGL. O chamado *retained mode* possui um nível maior de abstração, podendo-se construir um grafo de cena completo a ser renderizado e, de acordo com a especificação, este deve ser implementado utilizando-se os métodos definidos para o *immediate mode*. Também há a possibilidade de se carregar cenas inteiras que tenham sido modeladas fora da aplicação, através de ferramentas. Os dois modos foram criados com o intuito de possibilitar flexibilidade na criação dos gráficos, assim como a criação de jogos e aplicações de forma mais rápida e gerando menos código, o que resulta num menor tamanho final do pacote da aplicação, salvando espaço no celular do usuário final.

3.3. Hardware

Os telefones celulares atualmente são produtos com foco em vários públicos diferentes. Deste modo, há uma grande variedade de dispositivos integrados, desenhos de interface e configurações de hardware.

Com tantas configurações, há uma diferença muito grande de desempenho entre os celulares.

Para ilustrar esta diferença a figura 4 mostra um comparativo do tempo de processamento entre alguns modelos presentes no mercado. Para tanto foi utilizado um algoritmo simples executando um milhão de multiplicações entre inteiros.

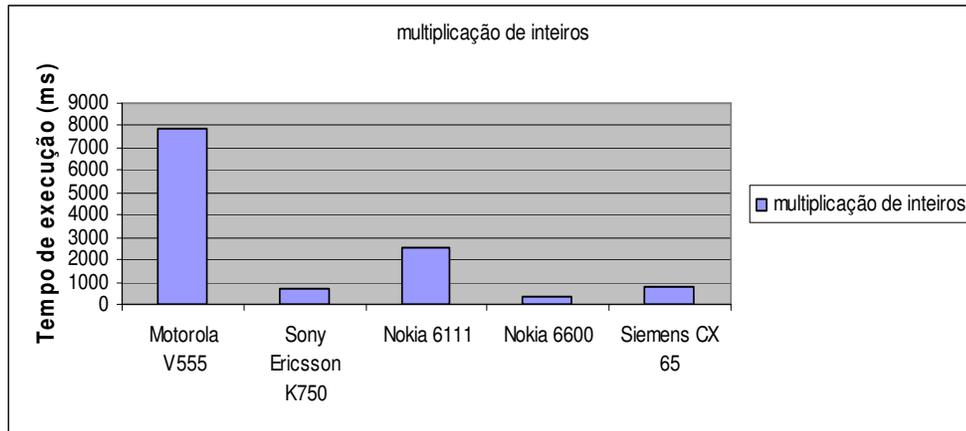


Figura 4 - Tabela comparativa de velocidade de processamento entre alguns aparelhos de celular disponíveis comercialmente

O modelo Nokia 6600 possui o sistema operacional Symbian OS, normalmente presente em celulares com uma maior capacidade de processamento. Já o celular Motorola V555 é o que apresenta menor quantidade de memória de execução, apenas 800KB, e menor capacidade de processamento, não tendo suporte operações de ponto flutuante, já que a plataforma J2ME presente possui CLDC 1.0 .

Já os outros três modelos possuem capacidade de processamento mediana, memória de execução de aproximadamente 2MB e API *Mobile 3D Graphics*, juntamente com CLDC 1.1, o que lhes confere operações de ponto flutuante disponíveis na plataforma J2ME.

Como muitas das informações sobre as diversas configurações de hardware utilizadas estão protegidas por questões de sigilo comercial, é difícil fazer uma análise mais profunda sobre a real capacidade de processamento da maioria dispositivos disponíveis no mercado.

Em relação aos cálculos com ponto flutuante, que são essenciais para certas as rotinas gráficas e para esta dissertação, existem algumas características interessantes nesta variedade de configurações.

Analisando a família de processadores fabricados pela ARM e Texas Instruments, para citar algumas fabricantes, pode-se notar que muitos não apresentam suporte a operações com ponto flutuante ou ponto fixo.

Para executar cálculos com o uso de números fracionais, necessários em rotinas de processamento gráfico, algumas abordagens são possíveis.

Os processadores ARM, por exemplo, podem adotar três modos diferentes. O primeiro é a adoção de biblioteca de software que traduz as operações em ponto flutuante para outras com números inteiros. Tal abordagem ocasiona um aumento no número de instruções executadas pelo processador.

O segundo é a adoção de um co-processador especializado nesse tipo de operação. Essa abordagem, apesar de ser a mais eficiente, apresenta um aumento de custo no hardware final, tanto financeiramente, quanto de espaço e energia.

Por fim, existe a possibilidade de utilização de um co-processador emulado por software.

É razoável supor que, mesmo com diferenças de arquitetura entre fabricantes, soluções parecidas sejam adotadas.

3.3.1. Aceleração Gráfica

Apesar de não estar presente em todos os modelos de celulares disponíveis no mercado, já é possível encontrar dispositivos equipados com hardware especializado no processamento de imagens e multimídia.

Os telefones comercializados hoje, em sua maioria, apresentam funcionalidades multimídia e estão integrados com câmeras fotográficas e de vídeo, sendo capazes de capturar e de reproduzir conteúdo digital, enviar mensagens multimídia (MMS) e visualização gráfica de qualidade.

Muitos desses dispositivos atualmente possuem co-processadores gráficos desenvolvidos por empresas como a Nvidia e ATI. Outras propostas como a apresentada pela Qualcomm, prevê a integração de aceleração gráfica no próprio core do processador do celular.

Como exemplo de co-processadores com capacidade de captura de imagens de resolução elevada, além de proporcionar uma melhor performance no processamento de imagens, estão a família Imageon 2200 da ATI e os processadores GoForce 2150 e 400 da Nvidia. Eles contam normalmente com capacidade de codificação e decodificação MPEG e JPEG, suporte a câmera de

até 3 mega pixels e aceleração para gráficos 2D, além de serem desenhados para ter baixo consumo de energia.

A adoção de aceleração de gráficos 3D em dispositivos móveis começou com o lançamento do processador GoForce 4500 da NVidia, fabricado com tecnologia de 0.15 microns. Ele já contava com um processador geométrico e a possibilidade de programação de *shaders*.

Atualmente a Nvidia oferece os processadores GoForce 4800 e 5500 que contam com pipeline de cores de 40 bits, suporte a multi-texturas, e com suporte as especificações OpenGL ES.

Apesar de não ser divulgado detalhes da arquitetura, os processadores GoForce 5500 contam com a capacidade de geração de 2.67 milhões de triângulos por segundo, e 200 milhões de pixels por segundo de taxa de preenchimento 3D. Um exemplo de telefone celular que utiliza um processador desta classe é o W900 da Sony Ericsson, que utiliza o processador GoForce 4800.

A ATI também conta com processadores com características parecidas como os da família Imageon 23XX, porém, suas características particulares não são amplamente divulgadas.