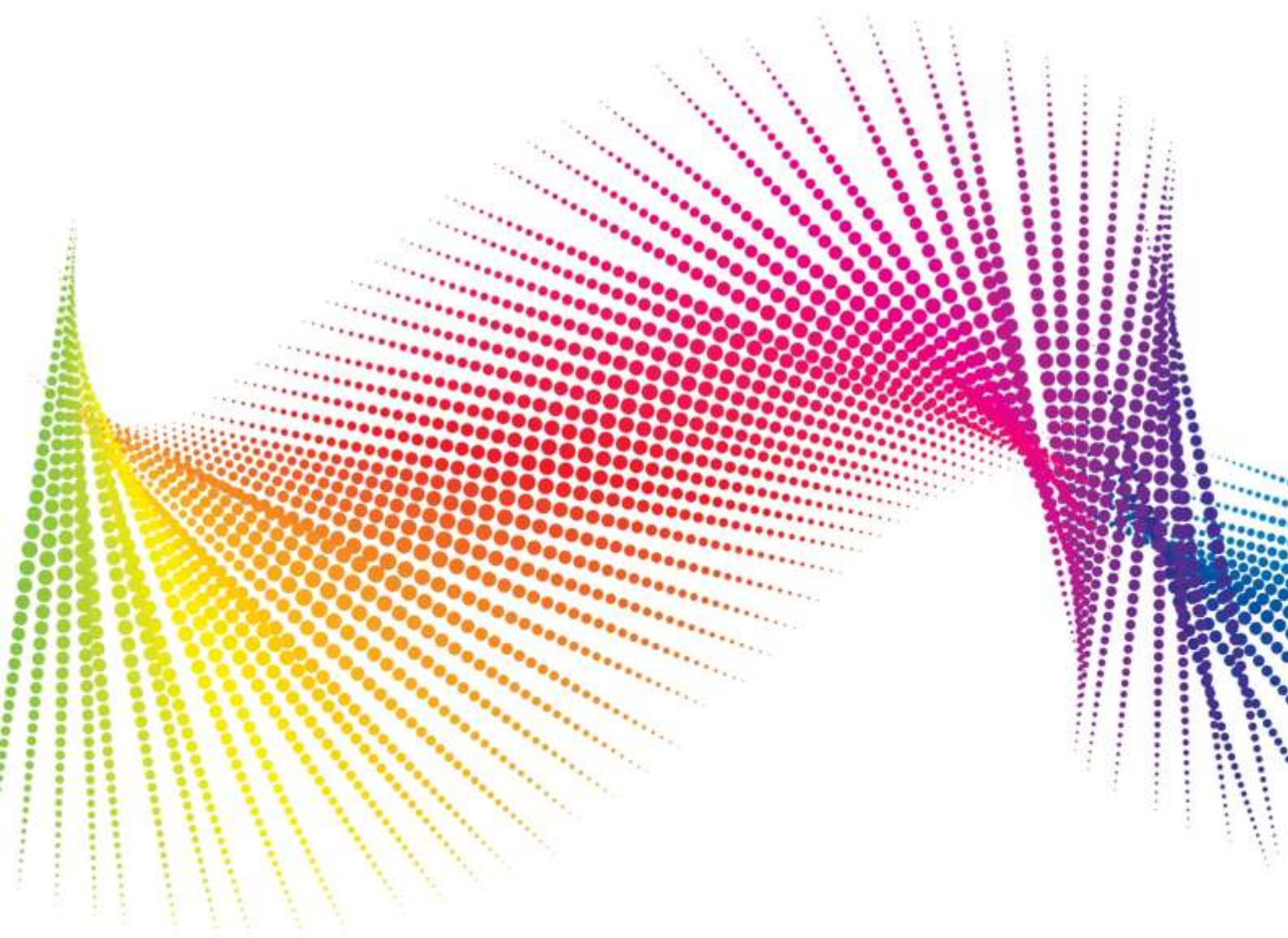


# Computação Móvel

Aula 02



Este material é parte integrante da disciplina oferecida pela UNINOVE.

O acesso às atividades, conteúdos multimídia e interativo, encontros virtuais, fóruns de discussão e a comunicação com o professor devem ser feitos diretamente no ambiente virtual de aprendizagem UNINOVE.

Uso consciente do papel.

Cause boa impressão, imprima menos.

## **Aula 02: Plataformas de desenvolvimento para aplicações móveis**

**Objetivo:** Demonstrar as principais ferramentas de desenvolvimento de aplicações móveis existentes no mercado, ressaltando as facilidades, os objetivos e as limitações presentes em cada uma delas em função do sistema para o qual foi desenvolvida.

### **J2ME**

Java 2 Micro Edition (J2ME) é a versão da linguagem Java idealizada para programação em dispositivos móveis, que possuem como características: mobilidade, baixa capacidade de processamento e pouca memória, alimentação por baterias, áreas de vídeo pequenas e limitadas e vários métodos de entrada e saída, ou seja, dispositivos como os vistos na aula passada (telefones celulares, PDAs – Assistentes Digitais Pessoais –, smartphones, entre outros).

Devemos entender, também, que o J2ME não é novo ou outro tipo de Java, mas, sim, de uma adaptação da plataforma existente para que se possam desenvolver, de maneira mais eficiente, programas para dispositivos móveis com as características citadas nesta linguagem. Deste modo, todos os programas desenvolvidos utilizando-se a plataforma J2ME também poderão ser executados sem nenhum problema nas edições Standard (J2SE) e Enterprise (J2EE), isso, logicamente, se APIs usadas no desenvolvimento estejam presentes na plataforma utilizada.

O que podemos fazer com J2ME?

Com a plataforma J2ME, procurou-se estender os conceitos utilizados na padronização de desenvolvimento de aplicações para pequenos dispositivos, idealizando pacotes que fazem transparecer ao desenvolvedor todos os detalhes proprietários do fabricante, como arquitetura de hardware utilizado e sistema operacional do dispositivo.

Para que se possa entender a evolução que isso representa, anteriormente, todo dispositivo possuía uma estrutura constante, oferecido com um conjunto fixo de funcionalidades em que a programação era realizada exclusivamente pelo

fabricante, utilizando uma tecnologia produzida, exclusivamente, pelos proprietários. Com o J2ME, isso sofreu uma grande alteração, tornando possível desenvolver, atualizar e instalar outros aplicativos que variam com as necessidades de cada usuário. É possível, depois de feito o download do programa para o dispositivo, usá-lo online ou offline. Até mesmo, caso se esteja trabalhando offline, faz-se a sincronização dos dados e das informações utilizadas quando a rede estiver presente, se este for o caso. Na plataforma J2ME, podem-se desenvolver jogos, aplicações utilitárias domésticas, aplicações que acessam banco de dados, etc.

Tendo o J2ME como ferramenta de programação para aparelhos móveis, ganham-se todas as vantagens desta tecnologia, como:

- **Dinamismo:** pode-se fazer download de novas aplicações da rede e instalá-las em um dispositivo móvel a qualquer instante;
- **Segurança:** garante a proteção das informações presentes. Os dados utilizados em uma aplicação não são acessíveis por outras. Havendo uma verificação de classes, forte tipagem, garbage collection, etc.
- **Portabilidade:** as aplicações podem ser executadas em uma grande variedade de dispositivos, independente do fabricante e de seus tipos.
- **Orientação a objetos:** alto nível de abstração do código, modularização e reusabilidade.

## **Android**

Android é um conjunto de programas para dispositivos móveis que estão inseridos em um sistema operacional, um middleware e um conjunto de aplicações-chaves. Desenvolvedores podem construir programas para esta plataforma fazendo uso do Android SDK. Os programas feitos para serem executados no Android são comumente escritos em linguagem de programação Java e executam sobre o Dalvik (uma máquina virtual idealizada especialmente para dispositivos que possuam principalmente poucos recursos, como baixa capacidade computacional, pouca área para armazenamento de dados).

## **Arquitetura**

### **Kernel do Linux**

A arquitetura do Android foi feita com base no kernel do GNU/Linux, versão 2.6. O kernel do sistema funciona como uma camada de abstração entre o hardware e o restante da pilha de softwares da plataforma.

O kernel GNU/Linux possui vários recursos necessários para a execução de aplicações, como gerenciamento de memória e de processos, pilha de protocolos de rede, módulo de segurança e vários módulos do núcleo de infraestrutura.

Desta maneira, quando existe a necessidade de desenvolver-se, por exemplo, um novo Driver ou fazer-se melhorias num existente, isto será muito facilitado, uma vez que o sistema operacional é conhecido.

### **Bibliotecas**

O Android possui um novo grupo de bibliotecas C/C++ que são utilizados pelos diversos componentes do sistema. Sendo que as funcionalidades são utilizadas no framework do Android. Algumas destas bibliotecas que fazem parte núcleo da arquitetura estão listadas abaixo:

- Surface manager: gerencia o acesso ao subsistema de display. Gera as camadas gráficas 2D e 3D de múltiplas aplicações.
- 3D libraries: representa uma implementação baseada na estrutura utilizada no OpenGL 1.0.
- SGL: biblioteca utilizada na criação de gráficos 2D.
- Media libraries: bibliotecas que suportam playback e gravação de formatos variados de áudio e de vídeo, bem como imagens estáticas, os formatos disponíveis são MPEG4, H.264, MP3, AAC, AMR, JPG e PNG.
- FreeType: biblioteca utilizada para desenhar fontes.
- SSL: fornece encriptação de dados enviados pela Internet.
- SQLite: biblioteca na linguagem C que dá suporte à estrutura do banco de dados SQL presente no Android.

Precisamos ter em mente que programas que utilizam a SQLite possuem as interações com o banco de dados SQL sem executar um processo RDBMS em separado, pois o SQLite não é apenas uma biblioteca de cliente utilizada para prover a conexão com um servidor de banco de dados, é um servidor de banco de dados e pode ler e escrever diretamente no arquivo do banco de dados do Android.

## **Android Runtime**

Quando executamos uma aplicação Android, ela cria seu próprio processo, ou seja, cria sua própria instância da máquina virtual Dalvik, para que os dispositivos possam executar múltiplas máquinas virtuais simultaneamente da maneira mais eficiente possível.

A máquina virtual Dalvik executa uma série de classes compiladas da linguagem Java, sendo que os arquivos .class gerados são transformados no formato .dex pela ferramenta dx, inclusa no SDK (Software Development Kit) do Android, e são esses arquivos que são executados pela máquina virtual Dalvik.

Lembrando que a máquina virtual Dalvik utiliza do kernel do GNU/Linux, o que lhe gera múltiplas funcionalidades, como threads e gerenciamento de memória de baixo nível.

No entanto, existem diferenças entre a máquina virtual Dalvik e a máquina virtual Java (JVM). A primeira é baseada em registradores, enquanto a segunda é baseada em pilhas. Isso foi feito devido ao fato de que uma estrutura baseada em registradores traz benefícios em ambientes restritos, como telefones celulares, além de que, numa análise mais detalhada, percebemos que máquinas virtuais baseadas em registradores executam mais rapidamente os programas do que as baseadas em pilhas.

Outra diferença importante que está presente na Dalvik é a estrutura, que permite que múltiplas instâncias da máquina virtual possam ser executadas ao mesmo tempo, com pequenas quantidades de memória. Sendo que cada aplicação é executada como um processo Linux independente.

## **XCode**

Trata-se de um ambiente integrado de desenvolvimento de softwares que gerencia projetos baseados na estrutura do sistema operacional de Mac OS X, criado como sendo um software livre da Apple Inc. O XCode concede um grupo de ferramentas para o usuário criar e aprimorar seus aplicativos. Sendo, também, um software poderoso, porém simples de utilizar, principalmente no desenvolvimento de aplicativos grandes. O Xcode pode ser obtido no DVD de instalação do Mac OS X nas versões "Leopard" e "Snow Leopard" e no website de desenvolvedores da Apple, porém não será distribuído com o Mac OS X "Lion", pois ele não possui DVD de instalação.

### **Desenvolvimento para Mac OS X**

O Xcode possui todas as ferramentas necessárias ao desenvolvimento de aplicações para o Mac OS X e suporta, por padrão, Objective-C e Apple-Script, que são suas linguagens de programação.

### **Desenvolvimento para iOS**

O Xcode também possui um conjunto de ferramentas "extras" de desenvolvimento, que são o SDK, fornecidos pela Apple Inc. no website de desenvolvimento de iOS.

## **Resumo**

Nesta aula, apresentamos as principais ferramentas para desenvolvimentos de aplicações móveis, especificando algumas de suas principais características e sua influencia no desenvolvimento das aplicações móveis.



## Próxima aula

Conhecendo as plataformas de desenvolvimento e os conceitos associados a elas, estamos preparados para dar o próximo passo – Aula 3 –, no qual ampliaremos nossos horizontes, aprendendo sobre as necessidades e recursos presentes na programação Java, que facilitaram o desenvolvimento das aplicações móveis.



### EXERCÍCIOS

Agora, veja os exercícios disponíveis acessando o AVA, ou via QR Code\*. Não deixe de visualizar esses exercícios, pois eles fazem parte da sequência desta aula e, portanto, são essenciais para a aprendizagem.



\* O QR Code é um código de barras que armazena links às páginas da web. Utilize o leitor de QR Code de sua preferência para acessar esses links de um celular, tablet ou outro dispositivo com o plugin Flash instalado.

## REFERÊNCIAS

LECHETA, Ricard R. *Android – aprenda a criar aplicações para dispositivos móveis com o android SDK*. 2. ed. São Paulo: Editora Novatec, 2010.

ROGERS, Rick *et al.* *Desenvolvimento de aplicações android*. São Paulo: Editora Novatec, 2009.