

Computação II – Orientação a Objetos

Fabio Mascarenhas - 2014.1

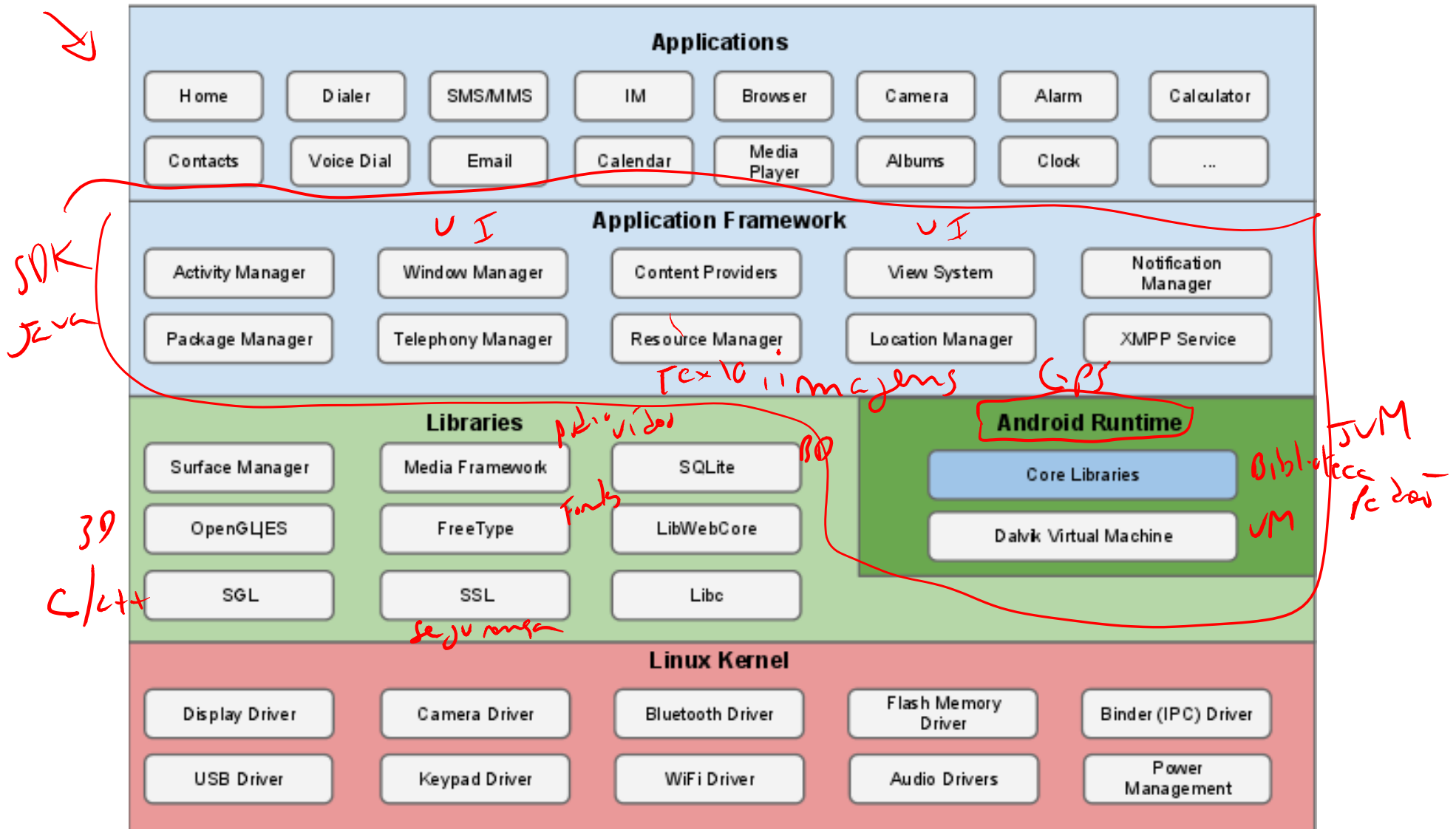
<http://www.dcc.ufrj.br/~fabiom/java>

Android

- Android é um sistema operacional para dispositivos móveis
 - *Kernel* Linux, drivers e bibliotecas do sistema, *frameworks* de aplicação (Android Software Development Kit, Android Native Development Kit) e aplicações embutidas
- Aqui estaremos interessados no Android SDK, um framework para desenvolvimento de aplicações para Android na linguagem Java
- O sistema Android possui [farta documentação](#)

↑
Java

Arquitetura do SO Android



Java no Android

- O sistema Android não usa a implementação oficial de Java, mas a sua própria, equivalente à versão 1.6 da implementação oficial
- A maior parte das classes nos pacotes java.* e javax.* estão presentes
Exceção: gráficos
- Classes específicas do sistema Android estão nos pacotes android.*, e classes úteis para aplicações Android estão nos pacotes org.*
- Outras bibliotecas Java geralmente funcionam sem modificações, mas às vezes dependem de partes que não estão no Android, mesmo que possuam equivalentes

Android Development Tools

- O ADT é um pacote que junta todo o necessário para desenvolver em Android:
 - O Android SDK
 - O *emulador* Android e uma *imagem* para o emulador
 - Uma versão do Eclipse específica para programar para Android
- Baixe o .zip do ADT [nesse link](#) (cerca de 500Mb!)
- Deixe ele descompactado em um pen drive, para podermos usar no laboratório

Hello, Android

- Vamos criar e rodar uma aplicação
- Uma aplicação Android já vem com muita estrutura, pois ela está inserida em um framework bastante complexo
- Ela já está separada em um *controlador* (a classe principal) e uma *visão* (descrita em arquivos XML), e mesmo algumas partes que formam um *modelo* bem simples (no arquivo `strings.xml`)
- Para rodar essa aplicação, primeiro precisamos criar um *dispositivo virtual* no emulador (ou conectar um dispositivo real via USB)

O Emulador

- Se usarmos a imagem de sistema padrão o emulador pode ficar lento demais
- Então vamos baixar uma imagem de sistema Intel x86 no Android SDK Manager, e criar um dispositivo virtual que use essa imagem
- Ainda assim é possível que o emulador fique lento, especialmente se a resolução do dispositivo virtual for alta
- O melhor mesmo é plugar um dispositivo via USB!

Componentes de uma App Android

código Java

- Atividades (Activities) → C do MVC
- Serviços (Services) → segundo plano (multitarefa)
- Comunicadores (Broadcastreceivers) → mensagens entre apps
- Provedores de Conteúdo (Content Providers) → dados para outros apps
- Recursos (Resources) → texto, imagens, layout das telas
- O sistema Android instancia os componentes sob demanda, e cada um deles tem seu propósito e API → sistema é o mediador

Recursos

- Recursos são metadados usados pela aplicação
 - Strings, layouts, imagens, menus, animações...
- Poderiam ser todos instanciados por código Java, mas usar arquivos de recursos deixa as aplicações mais flexíveis e configuráveis
- Normalmente recursos são descritos em arquivos XML
- Exemplo: internacionalização

XML

- Um formato para descrever dados
- Um arquivo XML é um conjunto de blocos, onde cada bloco é delimitado por uma *tag de abertura* e uma *tag de fechamento*
- Uma *tag de abertura* é composta por <, o *nome da tag*, os *atributos*, e >, enquanto uma *tag de fechamento* é </, o nome da tag, e >
- Cada *atributo* é composto do *nome* do atributo, o operador =, e o *valor* do atributo

Strings

- Strings são tipicamente usadas para configurar o idioma da aplicação
- Descritas em um arquivo strings.xml, com blocos do tipo:

```
<string name="hello_world">Olá, Android!</string>
```

- Podem ter *tags* ****, *mejrito* **<i>** e *sublinhado* **<u>** para formatação
itálico
- Referenciadas por outros recursos com `@string/hello_world` ou em código Java por `R.string.hello_world`

Layouts

- Layouts são descrições dos componentes visuais que formam as telas da aplicação
- Eles estão arquivos XML em `res/layout`, e contêm um bloco XML com uma das tags de layout possíveis; dentro desse bloco colocamos outros blocos com tags dos componentes que queremos
- No HelloAndroid, usamos um `RelativeLayout` em `fragment_hello`, e podemos referenciar esse layout com `@layout/fragment_hello` em um recurso, ou `R.layout.fragment_hello` em código Java

AndroidManifest

- O *manifesto da aplicação* descreve os atributos gerais da aplicação:
 - Nome, versão e ícone
 - Atividade principal da aplicação, e seus outros componentes
 - Versão mínima do Android que ela precisa
 - Permissões (câmera, internet, contatos, etc.)

acesso direto

Comunicação é sempre possível
abrir o browser não precisa de
permissão

Atividades

- Uma atividade é uma subclasse da classe abstrata `Activity`, e representa uma ação do usuário, e sua interface gráfica
- O método principal que uma atividade deve implementar é o `onCreate`, onde ela:
 - Restaura o seu *estado*
 - Diz qual a *visão* que ela vai exibir
 - Inicializa os componentes da interface, e liga seus eventos a ações