



Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis - IMESA

DANILO RODRIGUES CATAPAN

DESENVOLVIMENTO DE APLICATIVO PARA ANDROID SDK

ASSIS
2009



Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis - IMESA

DESENVOLVIMENTO DE APLICATIVO PARA ANDROID SDK

DANILO RODRIGUES CATAPAN

Trabalho de Conclusão de Curso apresentado ao
Instituto Municipal de Ensino Superior de Assis,
como requisito do Curso de Graduação, analisado
pela seguinte comissão examinadora:

Orientador: Profa. Dra. Marisa Atsuko Nitto

Analisador (1): Alexandre Charles Cassiano

ASSIS
2009



Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis - IMESA

DANILO RODRIGUES CATAPAN

DESENVOLVIMENTO DE APLICATIVO PARA ANDROID SDK

Trabalho de conclusão de Curso apresentado ao
Instituto Municipal de Ensino Superior de Assis,
como requisito do Curso de Graduação, analisado
pela seguinte comissão examinadora:

Orientadora: Profa. Dra. Marisa Atsuko Nitto

Área de Concentração: Informática

ASSIS
2009

Dedico este trabalho, primeiramente, a Deus por acreditar que nossa existência pressupõe uma outra infinitamente superior, à minha família e as pessoas que acreditaram em mim.

AGRADECIMENTOS

À Profa. Dra. Marisa Atsuko Nitto, pela orientação clara e objetiva permitindo a realização deste trabalho;

Aos meus pais, os quais me incentivaram e apoiaram durante todo o tempo, além da ajuda financeira disponibilizada no decorrer do curso;

Aos meus familiares que se fizeram presentes em todos os momentos em que necessitei;

Aos amigos que sempre me apoiaram e a todos aqueles que direta ou indiretamente contribuíram com o meu crescimento pessoal e profissional;

A minha namorada, por acrescentar razão e beleza aos meus dias, os quais fizeram grande diferença na minha vida e que nunca serão esquecidos.

“O único lugar onde o sucesso vem antes do trabalho é no dicionário.”
Albert Einstein

RESUMO

O mercado de celulares está crescendo cada vez mais e estudos mostram que hoje em dia mais de três bilhões de pessoas possuem um aparelho celular, e isso corresponde a mais ou menos metade da população mundial. Os usuários comuns estão procurando cada vez mais celulares com diversos recursos. Com isso a busca por uma plataforma moderna e ágil para o desenvolvimento de aplicações se torna de extrema importância. A nova sensação do momento é o Android e promete revolucionar o desenvolvimento de aplicações.

O Android é a plataforma open-source criada pelo Google para o desenvolvimento de aplicações para dispositivos móveis. Inclui um sistema operacional baseado no Linux e diversas aplicações, com uma rica interface gráfica, um browser para navegar na internet, integração com o Google Maps, suporte a multimídia, GPS, banco de dados integrado, jogos em 3D e muito mais.

A proposta deste projeto de pesquisa é adquirir conhecimento de toda a tecnologia envolvendo esta plataforma e explorar os recursos do Android para desenvolver uma aplicação de cadastro de alunos de uma faculdade através de seu emulador, utilizando um banco de dados.

Palavras-chaves: Google Android. Banco de dados.

ABSTRACT

The market of cellular is growing more and more and studies show that nowadays more than three billion people possess cellular apparel, and that corresponds more or less half of the world population. The common users are seeking more and more cellular with several resources. With that the search for a modern and agile platform for the development of applications becomes of extreme importance. The new sensation of the moment is Android and it promises to revolutionize the development of applications.

Android is the platform open-source created by Google for the development of applications for movable devices. It includes an operating system based on Linux and several applications, with a rich graphic interface, a browser to navigate in the internet, integration with Google Maps, support the multimedia, GPS, integrated database, games in 3D and much more.

The proposal of this research project is to acquire knowledge of the whole technology involving this platform and to explore the resources of Android to develop an application of students' of an university cadaster through your emulator, using a database.

Keywords: Google Android. Data base

LISTA DE ILUSTRAÇÕES

Figura 1 - Camadas do Software Android.....	17
Figura 2 - Demonstração de Aplicativos.....	19
Figura 3 - Demonstração de Aplicativos de Framework.....	20
Figura 4 - Exemplos de Views.....	21
Figura 5 - Exemplos de Views.....	21
Figura 6 - Exemplos de Views.....	21
Figura 7 - Exemplos de Views.....	22
Figura 8 - Exemplos de Views.....	22
Figura 9 - Exemplos de Views.....	22
Figura 10 - Exemplos de Views.....	23
Figura 11 - Demonstração de Bibliotecas.....	23
Figura 12 - Demonstração do Android Runtime.....	24
Figura 13 - Demonstração do Kernel do Linux.....	25
Figura 14 - Instalação no Android sdk.....	27
Figura 15 - Instalação no Android sdk.....	28
Figura 16 - Instalação no Android sdk.....	29
Figura 17 - Instalação no Android sdk.....	29
Figura 18 - Instalação no Android sdk.....	30
Figura 19 - Instalação no Android sdk.....	31
Figura 20 - Instalação no Android sdk.....	32
Figura 21 - Instalação no Android sdk.....	32
Figura 22 - Instalação no Android sdk.....	33
Figura 23 - Instalação no Android sdk.....	34
Figura 24 - Modelagem do problema.....	35
Figura 25 - Criação de um banco de dados.....	38
Figura 26 - Criação da tabela e seus valores a serem inseridos.....	38
Figura 27 - Tabela já criada e seus valores.....	39
Figura 28 - Envio do banco de dados para o emulador.....	42
Figura 29 - Tela com a lista de alunos.....	46
Figura 30 - Tela para inserir ou editar um aluno.....	47
Figura 31 - Tela para buscar um aluno.....	48

SUMÁRIO

1. INTRODUÇÃO	11
1.2 OBJETIVOS.....	12
1.3 JUSTIFICATIVAS E MOTIVAÇÕES.....	12
1.4 ESTRUTURA DO TRABALHO.....	13
2. FUNDAMENTAÇÃO SOBRE O ANDROID SDK	14
2.1 UMA BREVE HISTÓRIA DO ANDROID.....	15
2.2 A PLATAFORMA ANDROID.....	15
2.2.1 Aplicações.....	19
2.2.2 <i>Framework</i>	20
2.2.3 Bibliotecas.....	23
2.2.4 <i>Android Runtime</i>	24
2.2.5 Linux Kernel.....	25
3. INSTALAÇÃO E CONFIGURAÇÃO DO ANDROID SDK	26
3.1 INSTALAÇÃO DOS <i>SOFTWARES</i> E <i>PLUGINS</i>	27
3.2 INSTALAÇÃO DO BANCO DE DADOS <i>SQLITE</i>	34
4. DESENVOLVIMENTO DO APLICATIVO	35
4.1 DESCRIÇÃO DO PROBLEMA.....	35
4.2 MODELAGEM DO PROBLEMA.....	35
4.3 IMPLEMENTAÇÃO.....	36
4.3.1 CRIAÇÃO DO BANCO DE DADOS.....	37
4.3.2 ESTRUTURA DE UM BANCO DE DADOS.....	39
4.3.2.1 <i>Android Manifest</i>	40
4.3.2.2 Tipos de Elementos.....	40
4.3.2.3 Aplicação para inserir, alterar e excluir alunos.....	41
4.3.2.4 Inserção de registros no banco de dados.....	42
4.3.2.5 Atualização de registros no banco de dados.....	43
4.3.2.6 Exclusão de registros no banco de dados.....	44
4.3.2.7 Busca de registros no banco de dados.....	45
4.3.2.8 Execução da aplicação de cadastro.....	46
5. CONCLUSÃO	49
REFERÊNCIAS BIBLIOGRÁFICAS	50

CAPÍTULO 1

INTRODUÇÃO

O desenvolvimento de aplicações para dispositivos móveis tem evoluído exponencialmente com o tempo e já se tornou um padrão de desenvolvimento quase onipresente no sentido de que muitas empresas estão escalonando ou recrutando novos desenvolvedores para formarem seus grupos de desenvolvimento para sistemas móveis a fim de criar novas ou adaptar soluções de serviços existentes para suprir a demanda do mercado (empresas bancárias disponibilizando serviços de operações financeiras pelo celular, controle de estoque de materiais e outros) [1].

Hoje são várias as plataformas para desenvolvimento de aplicações para dispositivos móveis. Algumas das tecnologias de desenvolvimento são: Symbian, JavaME, .NET Compact Framework e Flash Lite.

E depois de muitos rumores informando que a Google entraria também no ramo de desenvolvimento de dispositivos móveis por meio do lançamento do “GPhone”, o mesmo foi muito mais além. Seguindo essa evolução tecnológica, a Google lançou em 2007 o *Android* que é uma plataforma de código aberto para dispositivos portáteis criado em parceria com gigantes do mercado de mobilidade e que formam o grupo Open Handset Alliance (OHA). Esta parceria visa consolidar uma plataforma única com diversos recursos que os usuários procuram em um celular moderno, e ainda revolucionar a maneira de construir aplicações para dispositivos móveis e acirrar a disputa no mercado corporativo. A OHA pretende repetir a estratégia da IBM com a fabricação do PC, composto por partes facilmente adquiridas no mercado [2].

Essa plataforma funciona como um sistema operacional como os já existentes *Symbian* e *Windows® Mobile*, com a diferença de ser baseado em código aberto. Com isso, qualquer desenvolvedor pode criar aplicativos para a plataforma Android. O kit de desenvolvimento do Android provê ferramentas e chamadas via APIs para o desenvolvimento de aplicativos baseados na linguagem Java. Em outras palavras,

trata-se de um pacote com programas para celulares, já com um sistema operacional, *middleware*, aplicativos e interface de usuário. Tais recursos permitirão que o Google e outras empresas ofereçam recursos mais ricos aos usuários que se encontram distantes de um computador de mesa. Os principais recursos dessa plataforma móvel são a máquina virtual otimizada, o navegador integrado, a biblioteca 2D e 3D, o banco *SQLite* e o *plugin* para o *Eclipse*, chamado *ADT*.

Neste projeto de pesquisa serão focados dois aspectos, o primeiro é adquirir um vasto conhecimento dessas tecnologias e o outro de desenvolver uma aplicação utilizando os recursos que a plataforma Android oferece.

1.2- OBJETIVOS

O objetivo deste trabalho é desenvolver um aplicativo para a plataforma *Android SDK*, com a vantagem que é um sistema operacional *open-source*, e que pode vir a ser a plataforma dominante entre os *smartphones* ao longo dos próximos anos. Além disso, adquirir conhecimento sobre o *Android SDK*, tendo em vista que empresas e os desenvolvedores buscam plataforma moderna e ágil para o desenvolvimento de aplicações.

1.3- JUSTIFICATIVAS E MOTIVAÇÕES

A área de entretenimento é uma das que mais cresceu nestes últimos anos. E outro motivo é que aplicativos em plataformas para dispositivos móveis ainda é muito pequena, tanto que algumas empresas têm oferecido premiações para os desenvolvedores.

1.4- ESTRUTURA DO TRABALHO

No primeiro capítulo está sendo apresentada a introdução sobre o tema, para que tenha uma noção sobre o que será abordada na pesquisa e dentre objetivos e justificativas para que a mesma fosse desenvolvida.

No capítulo dois a fundamentação sobre o Android SDK, em que explica a sua história de desenvolvimento junto da sua arquitetura em detalhes.

O capítulo três demonstra como adquirir o Android SDK e configurar o mesmo no aplicativo de desenvolvimento o Eclipse.

No capítulo quatro serão definidos o problema e a modelagem do problema. O aplicativo que será desenvolvido para o Android SDK utilizando um banco de dados.

E no capítulo final, o quinto, será a conclusão após todo o estudo apresentado.

CAPÍTULO 2

FUNDAMENTAÇÃO SOBRE O ANDROID SDK

O Android é um sistema operacional completo com base no kernel Linux®. Inicialmente, o objetivo de implementação para o Android era para o telefone móvel, incluindo telefones inteligentes e celulares mais baratos. Entretanto, a variedade completa do Android de tarefas de computação e o suporte funcional têm potencial para ir além do mercado de telefones móveis. O Android pode ser útil para outras plataformas e aplicativos.

O BlackBerry e o iPhone, cujas plataformas móvel são atraentes e de alto volume, estão abordando extremidades opostas de um espectro. O BlackBerry é completamente seguro para o usuário de negócios corporativo. Para um dispositivo de usuário final, é difícil competir com o iPhone em termos de facilidade de uso e "estar na moda." Android, uma plataforma jovem e ainda não comprovada, tem o potencial de jogar dos dois lados do espectro de telefonia móvel e, talvez, de criar uma ponte sobre o abismo entre trabalho e diversão. O Android poderia ser, provavelmente, um "Linux melhor" [3].

2.1.- UMA BREVE HISTÓRIA DO ANDROID

A plataforma Android é o produto do Open Handset Alliance (OHA), um grupo de organizações colaborando para a construção de um telefone móvel melhor. O grupo, liderado pelo Google, inclui operadores de telefonia móvel, fabricantes de aparelhos portáteis, fabricantes de componentes, provedores de plataformas e soluções de software e empresas de marketing. A partir de um ponto de vista de desenvolvimento de software, o Android fica bem ao centro do mundo do software livre. O primeiro telefone portátil com capacidade para Android no mercado foi o dispositivo G1 fabricado pela HTC e fornecido pela T-Mobile. O dispositivo se tornou disponível após quase um ano de especulações, quando as únicas ferramentas de desenvolvimento de software disponíveis eram alguns releases do SDK em constante aprimoramento. Conforme a data de release do G1 se aproximava, a equipe do Android liberou o SDK V1.0 e os aplicativos começaram a aparecer para a nova plataforma. Para estimular a inovação, o Google patrocinou duas séries do "Android Developer Challenges," onde milhões de dólares foram envolvidos nas melhores participações. Alguns meses depois do G1, o Android Market foi lançado, permitindo que os usuários navegassem e fizessem o download de aplicativos diretamente em seus telefones. Após cerca de 18 meses, uma nova plataforma móvel entrava na arena pública[4].

2.2- A PLATAFORMA ANDROID

O Android foi lançado inicialmente pelo Google. A plataforma está sendo mantida pelo Open Handset Alliance, que é um grupo formado por mais de 30 empresas as quais se uniram para inovar e acelerar o desenvolvimento de aplicações, serviços, trazendo aos consumidores uma experiência com mais recursos, com menos despesas em termos financeiros para o mercado móvel [1].

Pode-se dizer que a plataforma Android é a primeira plataforma móvel completa, aberta e livre.

A plataforma foi desenvolvida sobre o sistema operacional Linux. Assim todas as características deste sistema foram incorporadas, bem como sistema de arquivos, o kernel, os servidores de terminais, etc. Aqui estão algumas características da plataforma:

- **Framework de Aplicação** permitindo reuso de componentes;
- **Máquina Virtual Dalvik** otimizada para dispositivos móveis;
- **Navegador Web Integrado** baseado na *engine* open source WebKit;
- **Gráficos Otimizados** por meio de uma biblioteca de gráficos 2D; e gráficos 3D baseados na especificação OpenGL ES 1.0;
- **SQLite** para armazenamento de dados em formato de estruturas relacionais;
- **Suporte para Mídias** de áudio (formatos MP3, AAC, AMR), vídeo (MPEG4 e H.264) e imagens (formatos JPG, PNG, GIF);
- **Telefonia GSM** (dependente de hardware);
- **Bluetooth, EDGE, 3G, e WiFi** (dependente de hardware)
- **Câmera, GPS, Bússola, e Acelerômetro** (dependente de hardware)
- **Poderoso Ambiente de Desenvolvimento**, incluindo um emulador de dispositivo, ferramentas para depuração, analisador de memória e performance; e um *plugin* para a IDE Eclipse[1].

Com tantas variedades de recursos do Android, é fácil confundi-lo com um sistema operacional desktop. O Android é um ambiente em camadas baseado em kernel Linux e que inclui funções ricas [4].

- Janelas;
- Visualizações;
- Widgets para a exibição de elementos comuns como caixas de edição, listas e listas suspensas

O Android tem o navegador incorporável baseado em WebKit, o mesmo mecanismo navegador de software equipando o navegador Mobile Safari do iPhone. O Android ostenta varias opções de conectividade, WiFi, Bluetooth e dados wireless através de uma conexão celular. Uma técnica popular nos aplicativos do Android é estabelecer uma conexão com o Google Maps para exibir um endereço diretamente em um aplicativo. O suporte para serviços baseados em locais (como GPS) e acelerômetros também está disponível no software Android, embora nem todos os dispositivos Android sejam equipados com o hardware necessário. Existe também suporte para câmera.

Duas áreas onde aplicações móveis lutaram para acompanhar suas contrapartes de desktop são gráfico/mídia e métodos de armazenamento de dados. O Android aborda o desafio dos gráficos com suporte integrado para gráficos em 2-D e 3-D, incluindo a biblioteca OpenGL. O peso do armazenamento de dados é amenizado porque a plataforma Android inclui o banco de dados SQLite de software livre popular [4]. A Figura 1 mostra uma visualização simplificada das camadas do software Android.



Figura 1: Camadas do Software Android. Fonte: [6].

A arquitetura do Android é dividida em cinco camadas, organizadas como mostra a Figura 1: *kernel Linux*, bibliotecas, ambiente de execução, *framework* e aplicativo. A camada do *kernel Linux* é composta pelo kernel 2.6 e se responsabiliza pelos serviços, segurança, gerenciamento de memória e processos, rede e *drivers*. Essa camada também é responsável pela abstração do hardware do dispositivo. A segunda camada, a de bibliotecas, carrega consigo um conjunto de bibliotecas C/C++ utilizadas pelo sistema. Estão incluídas nesse conjunto a biblioteca C padrão (*Libc*) e também aquelas das áreas de multimídia, visualização de camadas 2D e 3D, funções para navegadores web, funções para gráficos, funções de aceleração de *hardware*, renderização 3D, fontes bitmap e vetorizadas e funções de acesso ao banco SQLite. Todos esses recursos estão disponíveis no *framework* para o desenvolvimento de aplicativos. A pequena camada do ambiente de execução (*Android Runtime*) é uma instância da máquina virtual *Dalvik* criada para cada aplicação executada no *Android*. A *Dalvik* é uma máquina virtual com melhor desempenho, maior integração com a nova geração de hardware e projetada para executar várias máquinas virtuais paralelamente [6].

Além disso, é otimizada para consumo mínimo de memória, bateria e CPU. Sobre essas camadas, localiza-se a camada do *framework (Application Framework)*. Nelas estão todas as APIs e os recursos utilizados pelos aplicativos, com classes visuais como botões e *views*, provedor de conteúdo (troca de recursos entre aplicativos) e gerenciadores de recursos, de notificação e de pacotes.

Acima de todas as outras camadas está a de aplicativos, na qual se encontram todos os aplicativos (escritos em Java) do Android, como cliente de email, navegador web, contatos e outros. Isso significa que, para desenvolver programas para a plataforma *Android*, será necessário criar os aplicativos em Java na máquina virtual *Dalvik*.

A plataforma *Android* agrada também aos fabricantes de chips, pois adiciona novos recursos aos aparelhos, que aumentará a busca por processadores mais velozes, impulsionando o desenvolvimento e a comercialização de novos produtos. Como o sistema é com o código aberto, existe possibilidade de portá-lo para

diferentes plataformas conforme necessário, assim como no caso do Linux, que roda tanto em micros PC, quanto em clusters de servidores e em *smartphones*.

Para melhor entendimento será explicada cada uma das partes da estrutura do Android sdk e sua composição.

2.2.1- Aplicações



Figura 2: Demonstração de Aplicativos.

Calendário

Mapas

Navegador

Contatos

Etc

2.2.2- Framework



Figura 3: Demonstração de Aplicativos de Framework.

View: Um conjunto rico e extensível de Visões que podem ser usadas para construir uma aplicação, incluindo listas, *grids*, caixas de texto, botões, etc.

Content Providers: Que permitem que as aplicações acessem os dados de outras aplicações (como contatos) ou compartilhem seus próprios dados.

Resource Manager: Que fornece acesso aos recursos *non-code* como *strings* localizadas, gráficos e arquivos de *layout*.

Notification Manager: Que permite que todas as aplicações exibam alertas na barra de status.

Activity Manager: Que gerencia o ciclo de vida das aplicações e fornece um *common navigation backstack* .

As Figuras 4, 5, 6, 7, 8, 9 e 10 são exemplos de vários tipos de *views*.



Figura 4: Exemplos de Views.

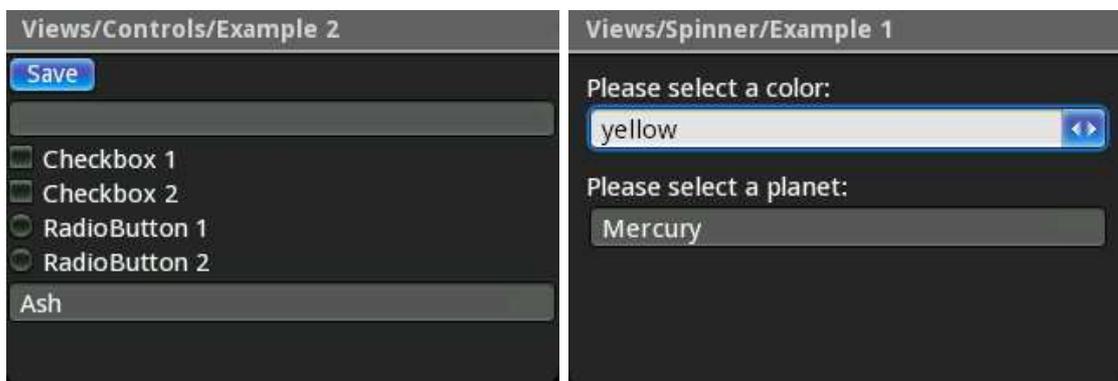


Figura 5: Exemplos de Views.



Figura 6: Exemplos de Views.

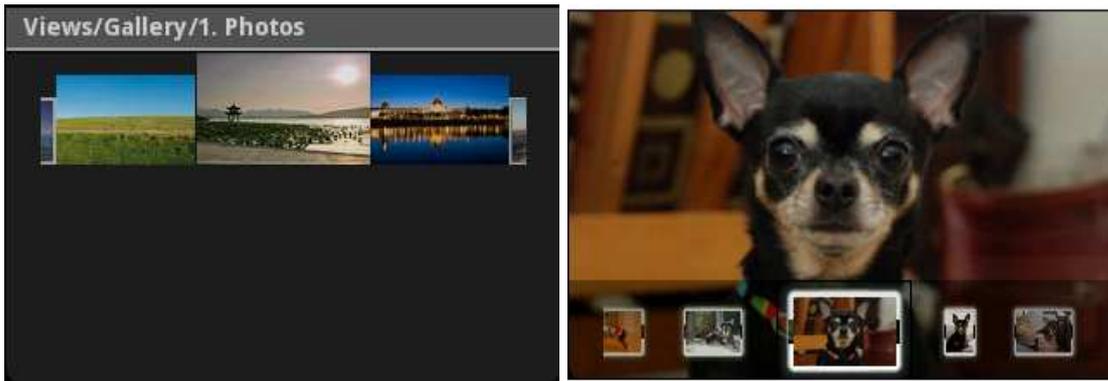


Figura 7: Exemplos de Views.

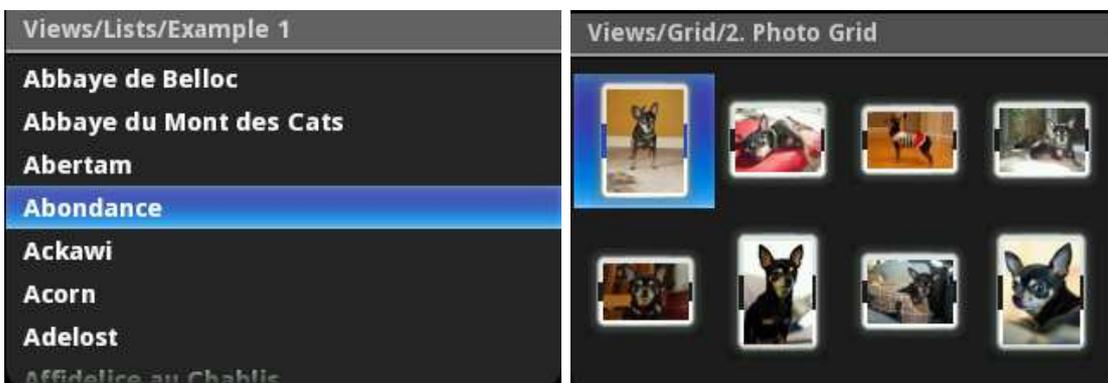


Figura 8: Exemplos de Views.

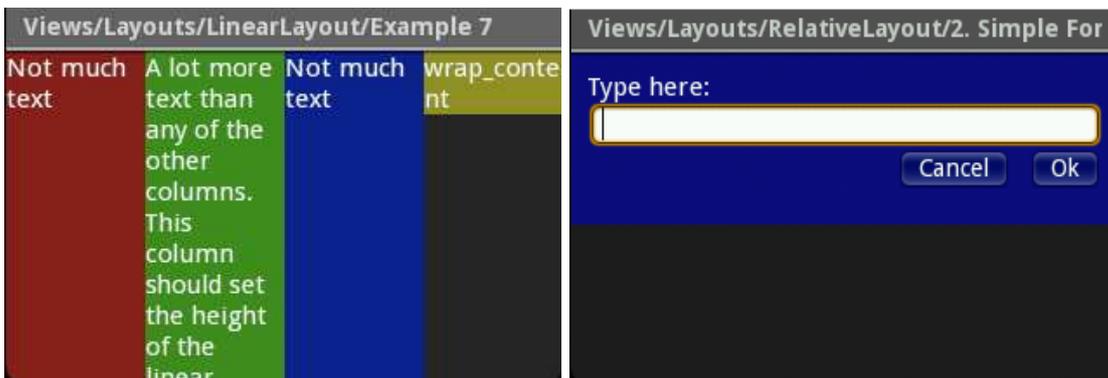


Figura 9: Exemplos de Views.

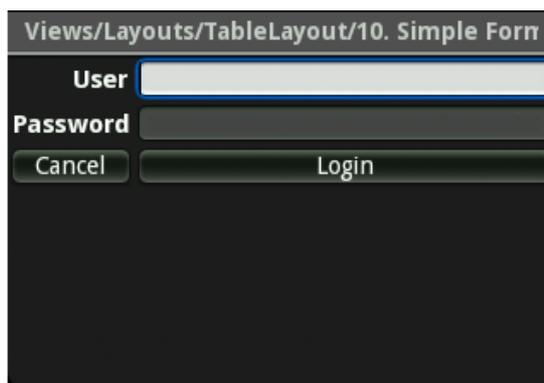


Figura 10: Exemplos de Views.

2.2.3- Bibliotecas



Figura 11: Demonstração de Bibliotecas.

System C library – uma implementação derivada da biblioteca C padrão sistema (libc) do BSD adaptada para dispositivos rodando Linux.

Media Libraries – baseado no PacketVideo's OpenCORE; as bibliotecas suportam os mais populares formatos de áudio e vídeo, bem como imagens estáticas.

Surface Manager – acesso ao subsistema de exibição bem como as múltiplas camadas de aplicações 2D e 3D;

LibWebCore – um *web browser engine* utilizado no Android Browser

SGL – o *engine* de gráficos 2D

3D libraries – uma implementação baseada no OpenGL 1.0; as bibliotecas utilizam aceleração 3D via hardware (quando disponível) ou o software de renderização 3D altamente otimizado incluído no Android.

FreeType – renderização de fontes bitmap e vetor

SQLite – um poderoso e leve *engine* de banco de dados relacional disponível para todas as aplicações

2.2.4- Android Runtime

A Figura 12 mostra

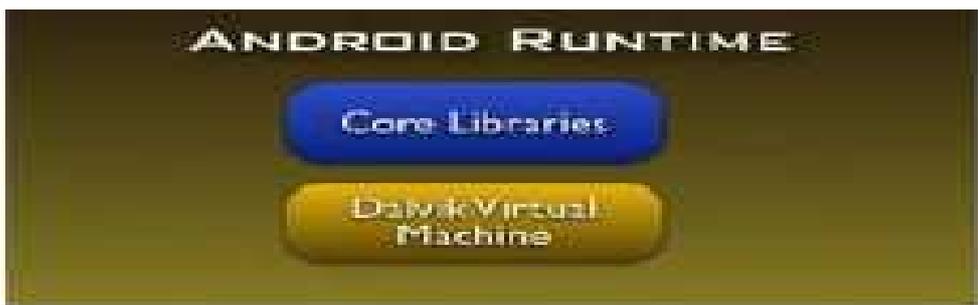


Figura 12: Demonstração do Android Runtime.

É um processo Linux

Própria instância da máquina virtual Dalvik

Executa arquivos.dex (ferramenta dx)

Toda aplicação Android executa em seu próprio processo, com sua própria instância da máquina virtual Dalvik. Dalvik foi escrito de forma que um dispositivo possa executar múltiplas VMs eficientemente. A máquina virtual Dalvik executa arquivos executáveis no formato executável .dex que é otimizado para um mínimo de memória *footprint*. A máquina virtual é baseada em registro e executa classes compiladas por um compilador da linguagem Java que tem sido transformada no formato .dex pela ferramenta incluída dx.

A máquina virtual Dalvik conta com o kernel do Linux para a funcionalidade base como *threading* e gerenciamento de memória de baixo nível.

2.2.5- Linux Kernel



Figura 13: Demonstração do Kernel do Linux.

Versão do Kernel: 2.6

O Kernel atua como uma camada de abstração entre o *hardware* e o resto da pilha de *software*

Serviços:

Segurança

Gerenciamento de memória

Gerenciamento de processos

Pilha de protocolos de rede

Modelo de drives

CAPÍTULO 3

INSTALAÇÃO E CONFIGURAÇÃO DO ANDROID SDK

Serão explicadas como são feitas as instalações dos componentes necessários para o desenvolvimento do aplicativo que são o Eclipse, o Android e o SQLite Expert Personal, desde o download dos *softwares* até a finalização de suas instalações. Para a elaboração do projeto, foi utilizado Eclipse Ganymede e o SDK do Android `sdk-windows-1.0_r2` e o *plugin* do Android para o Eclipse ADT-0.4.0. Qualquer versão dos programas citados serve. Para que toda essa aplicação funcione é necessário que tenha instalado antes de tudo, a Máquina Virtual Java (de preferência a versão 5 ou posterior) [7]. Os links para download dos softwares utilizados são:

- 1- <http://code.google.com/android/download.html> para SDK do Android e o *plugin* do Android para as versões específicas.
- 2- <http://www.eclipse.org/downloads/> para Eclipse Europa (ou superior).

Para saber se possui uma Máquina virtual Java instalada é necessário entrar no prompt de comando e digitar a seguinte linha:

```
java -version
```

Após o comando deve mostrar algo parecido com o código abaixo:

```
java version "1.6.0_07"  
Java(TM) SE Runtime Environment (build 1.6.0_07-b06)  
Java HotSpot(TM) Client VM (build 10.0-b23, mixed mode, sharing)
```

Caso o código seja o mesmo apresentado, isto significa que possui uma máquina virtual Java instalada no seu computador. No caso de não ter a máquina

virtual Java instalada é necessário entrar no site da Sun e fazer o seu download. O link é: <http://java.sun.com/javase/downloads/index.jsp>

3.1- Instalação dos softwares e plugins

Para instalar os programas e *plugins* é necessário seguir as seguintes etapas:

- 1ª etapa:** instalar primeiro o Eclipse. Para instalar o eclipse, descompacte-o num local desejado;
- 2ª etapa:** instalar o SDK Android. Para instalar o SDK Android, descompacte-o em um local apropriado;
- 3ª etapa:** instalar o *plugin* do Android. Para instalar o *plugin* do Android descompacte-o num local desejado.

Executar o eclipse, caso esteja executando pela primeira vez, ele solicita a definição de uma pasta padrão (um *Workspace*). Defina a pasta e continue o carregamento do eclipse. Depois de carregado, se for a primeira vez, feche a Janela do “*Welcome*” (Bem-vindo). Para instalar o *plugin* do Android, vá ao menu “Help / Softwares Updates / Find and Install”, como mostra a figura 14:

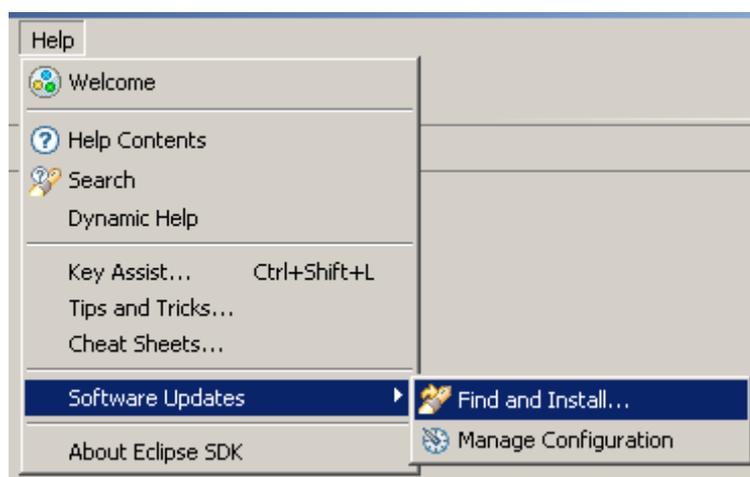


Figura 14: Instalação no Android sdk.

Selecione a opção “*Search for new features to install*” e clique no botão “*Next*”. Será exibida uma tela conforme a Figura 15.

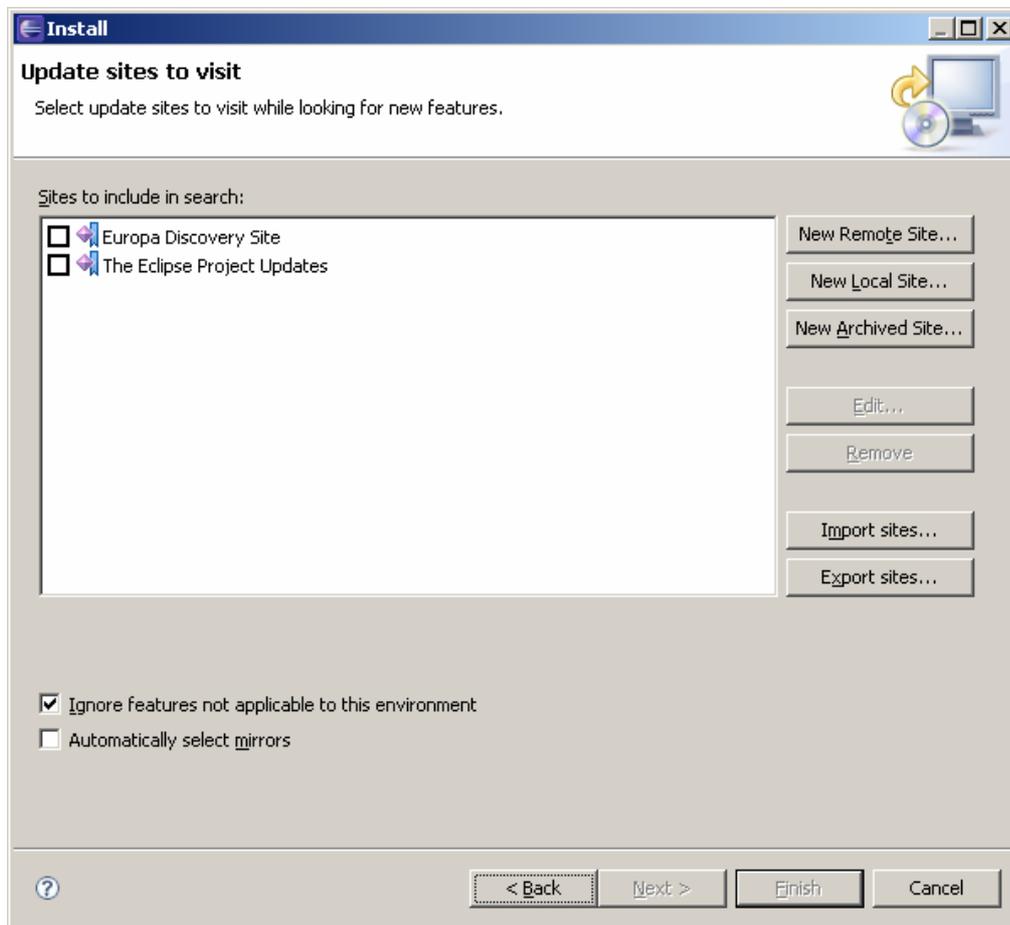


Figura 15: Instalação no Android sdk.

Feito isso, clique no botão “*New Local Site*” e indique o diretório do ADT já descompactado no local onde extraiu. Para ficar mais claro, dentro do diretório ADT existem dois diretórios `__MACOSX` e um outro chamado `ADTx.x.x`, selecione este. Feito isso, clique em “*OK*” e será exibida uma caixa de diálogo conforme Figura 16.

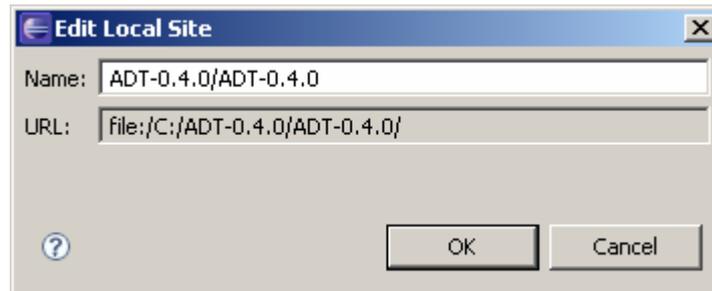


Figura 16: Instalação no Android SDK.

Na caixa de diálogo da Figura 16, basta clicar no botão “OK”. Pronto! O item foi inserido, conforme a Figura 17. Agora é só clicar em “*Finish*”.

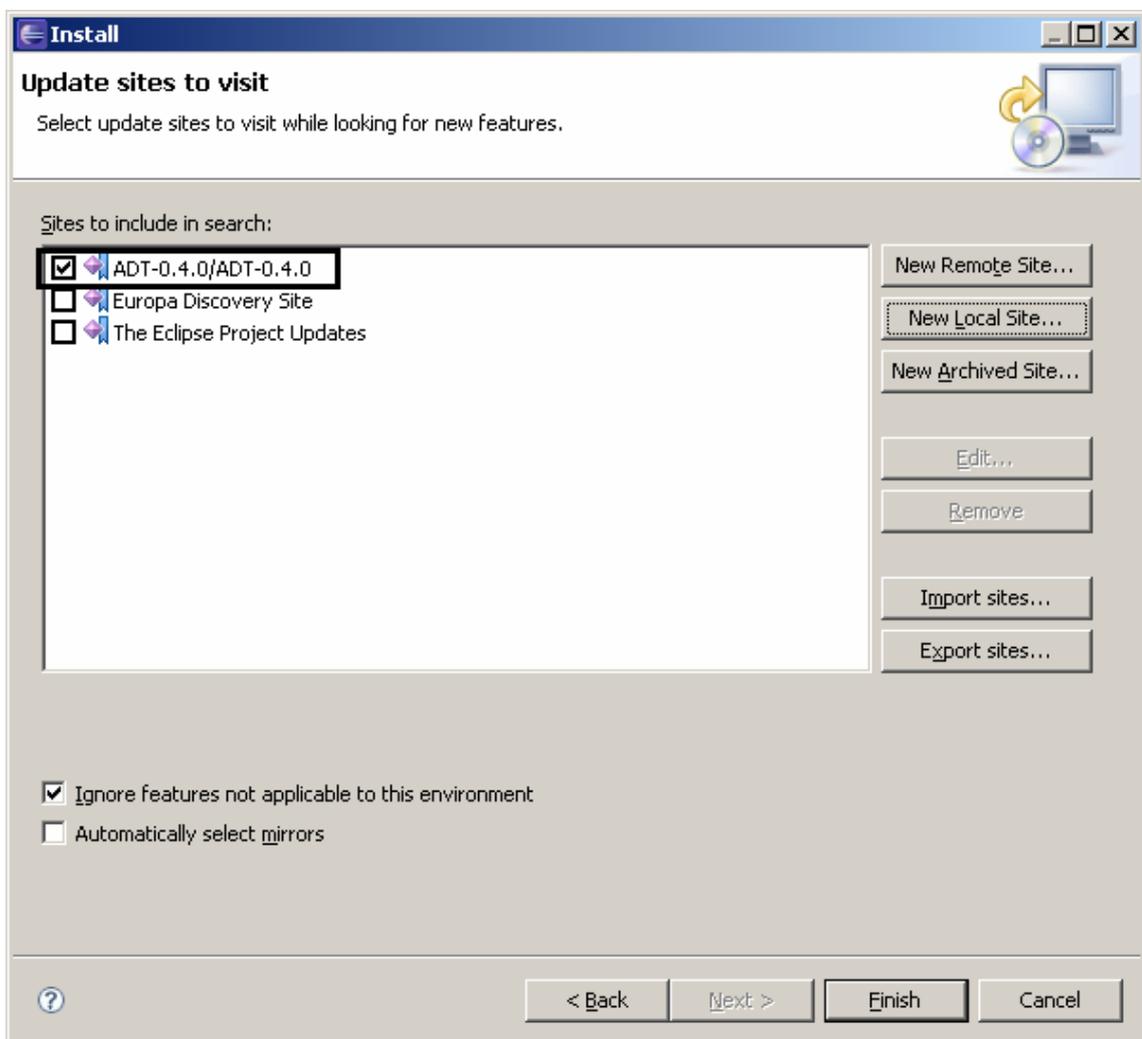


Figura 17: Instalação no Android SDK.

Depois de clicar em “*Finish*”, será mostrado uma nova tela conforme Figura 18.

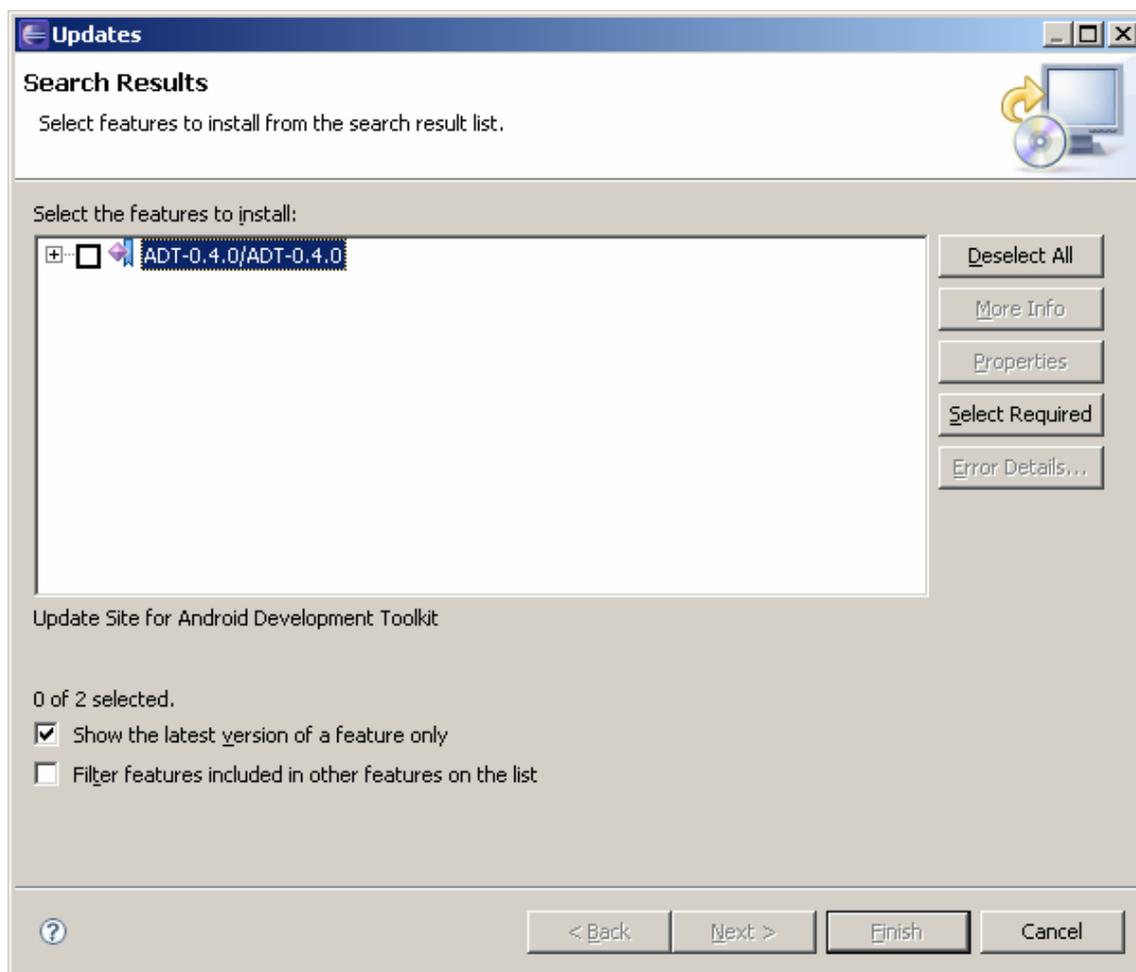


Figura 18: Instalação no Android SDK.

Clique no sinal “+” e faça as seleções dos *plugins* de acordo com a Figura 19. Caso esteja utilizando outra versão do *plugin* é necessário selecionar todas as opções desde que nenhuma delas resulte em alguma mensagem de erro. No caso da versão do *plugin* utilizada aqui, o último item (Android Editors) não pode ser instalado pois requer a existência de um outro *plugin*.

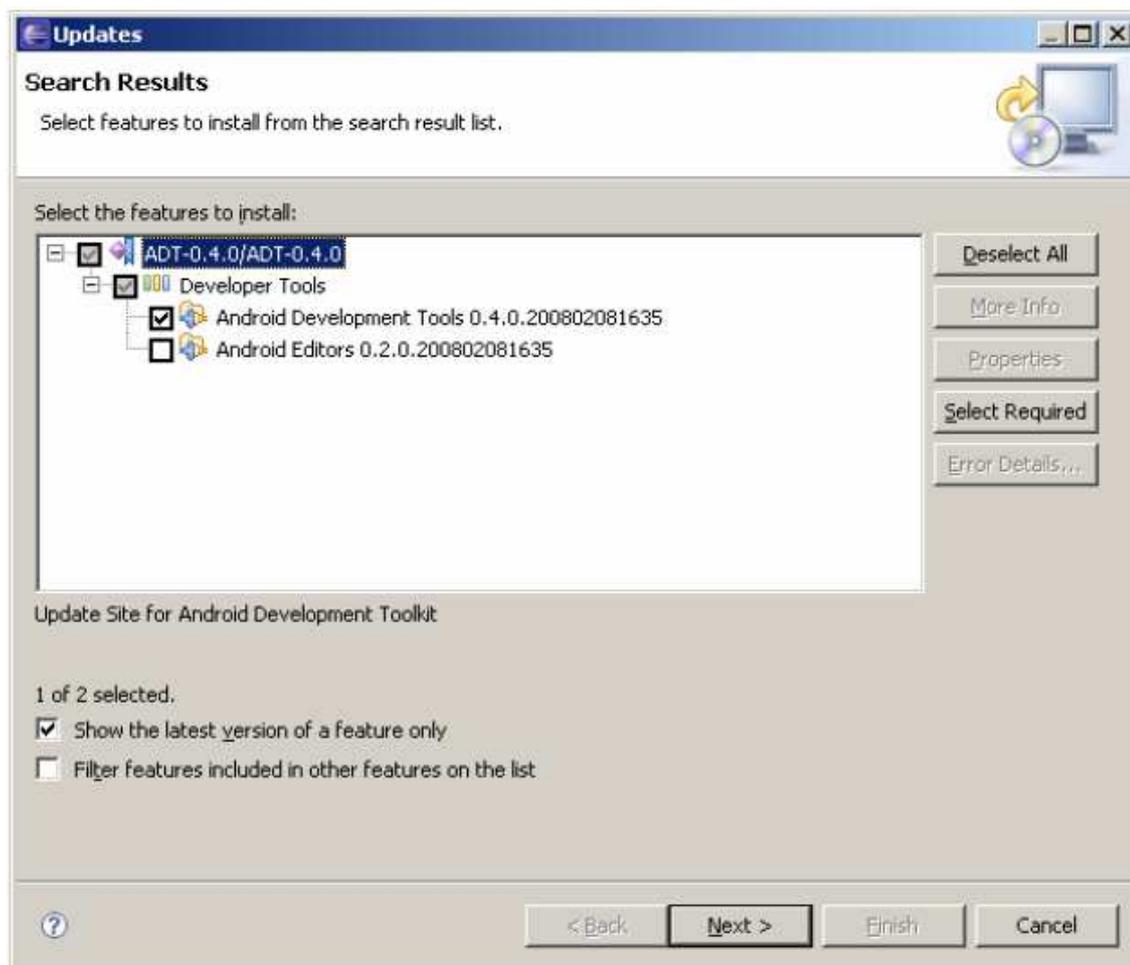


Figura 19: Instalação no Android SDK.

Feito isso, clique no botão “Next”, e será exibida uma tela de *license* de *software*. Nesta tela é só selecionar a opção “*I accept the terms in the license agreement*” e clique no botão “Next”. Na próxima tela, simplesmente clique no botão “Finish”. Na Figura 20, clique no botão “Install All”.

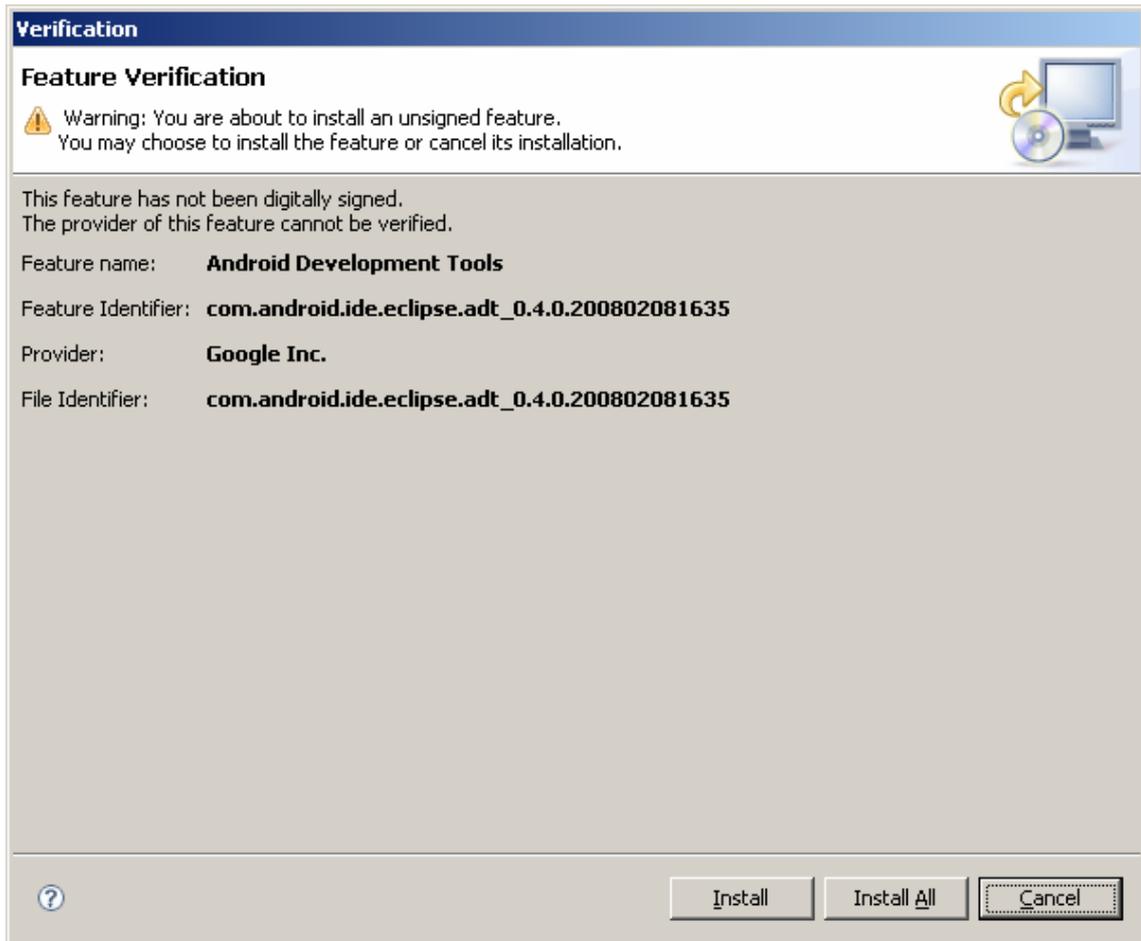


Figura 20: Instalação no Android SDK.

Após instalar os *plugins*, será exibida uma mensagem, conforme a Figura 21, recomendando ao usuário que ele reinicie o eclipse. Simplesmente clique no botão “Yes”.

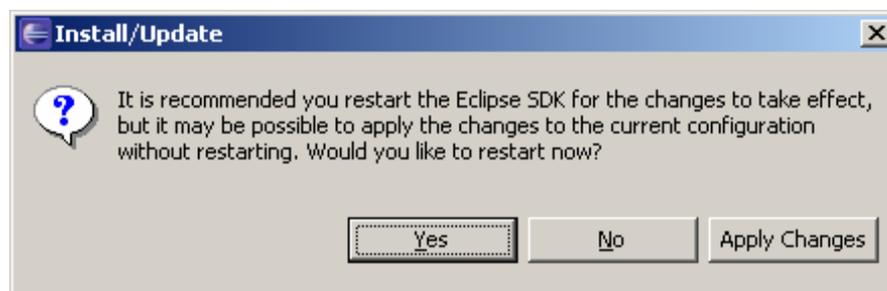


Figura 21: Instalação no Android SDK.

Após o eclipse ter reiniciado, serão feitas as configurações para fazer conexão com o emulador do Android. No menu “Window” / “Preferences”. Aberta a caixa de diálogo, selecione o item “Android”, serão mostradas duas mensagens de erro, ignore-as. Indicar o local onde se encontra o emulador do Android, simplesmente clicando no botão “Browse...” e indicando o local onde foi instalado. Feito isso, a tela que será mostrada é a da Figura 22. Se estiver (mostrando lá no topo o título Android) o diretório está correto e é só clicar em “OK”.

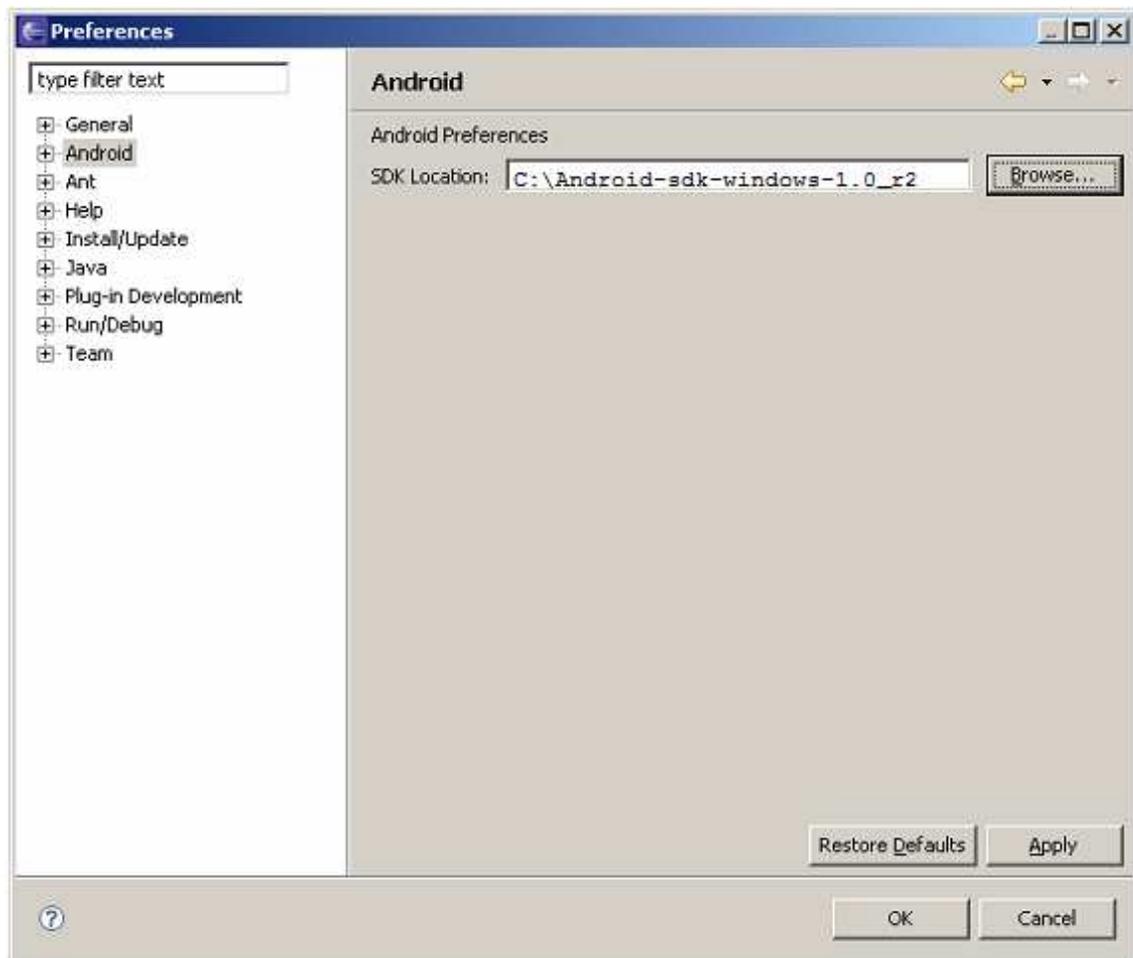


Figura 22: Instalação no Android SDK.

3.2- Instalação do Banco de Dados SQLite

O Android tem suporte ao SQLite [8], sendo um leve e poderoso banco de dados. Cada aplicação pode criar um ou mais banco de dados, que ficam localizados na seguinte pasta, relativa ao nome do pacote do projeto:

```
/data/data/nome_pacote/database/
```

Neste projeto, o banco de dados será criado com a ferramenta SQLite Expert Personal, que é uma ferramenta bem simples e intuitiva, similar a outros aplicativos existentes para gerenciar banco de dados. O link para o download é: http://www.softsland.com/sqlite_expert_personal.html

A instalação é simples, sem complicação alguma, sempre dando “Next” como mostra a Figura 23 e será exibida uma tela de licença de software. Nesta tela é só selecionar a opção “I accept the terms in the license agreement” e clique no botão “Next”, “Next”, “Next” e “Install”.



Figura 23: Instalação no Android SDK.

CAPÍTULO 4

DESENVOLVIMENTO DO APLICATIVO

Neste capítulo, serão definidos o problema e a modelagem do problema. O aplicativo será desenvolvido para o Android SDK utilizando o banco de dados SQLite Expert Personal.

4.1- Descrição do problema

O problema que será abordado neste projeto de pesquisa, consiste em desenvolver uma aplicação na plataforma Android através de seu emulador, utilizando um banco de dados para demonstrar uma simples movimentação de cadastro de alunos de uma faculdade, com os campos nome, cidade e o curso. Esta aplicação será desenvolvida através da linguagem de programação JAVA e um ambiente de desenvolvimento de alto nível e produtividade, o Eclipse.

4.2- Modelagem do problema

A Figura 24 mostra a modelagem do problema. Ela foi dividida em módulos para facilitar o desenvolvimento de cada parte que constitui a aplicação.

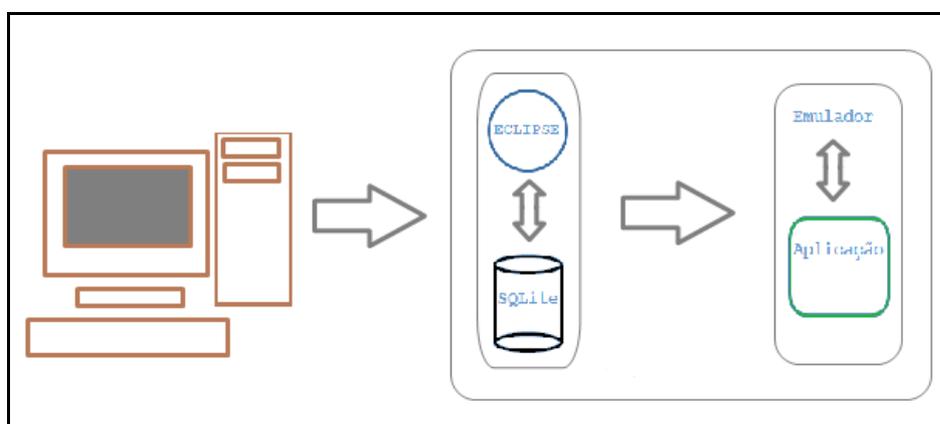


Figura 24: Modelagem do problema.

Os módulos que serão desenvolvidos são: criação do banco de dados, implementação e execução.

- **Modulo 1: Criação do Banco de Dados.**

Antes de começar a desenvolver a aplicação no Eclipse é feita a criação do banco de dados que será usado no projeto, sendo assim necessita-se da instalação do SQLite Expert Personal e a criação do banco junto com as tabelas que serão utilizadas.

- **Modulo 2: Implementação.**

Após a criação do banco que será utilizado no projeto, tem início a implementação onde é programado pela linguagem JAVA utilizando o ambiente de desenvolvimento Eclipse.

- **Modulo 3: Execução.**

Com a implementação concluída o próximo passo é a execução, após a execução inicia-se o emulador do Android, onde o projeto deixa de necessitar do Eclipse e do SQLite Expert Personal e fica executando independente. Utilizando seus próprios recursos.

4.3- Implementação

Será apresentado a implementação Java do problema descrito junto ao banco de dados SQLite no Android SDK. Primeiro passo para o desenvolvimento do aplicativo é a criação do banco de dados.

4.3.1- Criação do banco de dados

O Android tem integração com o SQLite, um leve e poderoso banco de dados, permitindo que você utilize banco de dados normalmente em sua aplicação [7]. Embora o armazenamento em banco de dados seja a forma mais comum de persistência, o Android também permite que arquivos sejam salvos facilmente, além de ter um sistema simples de persistência de chave e valor [9] e [10].

Uma vez que o Android tem suporte ao SQLite é mais prático utilizar uma aplicação gráfica capaz de gerenciar esse banco de dados. Partindo desta ideia, será utilizado o SQLite Expert Personal, uma ferramenta bem simples e intuitiva, similar a outros aplicativos existentes para gerenciar banco de dados.

Depois de instalar e abrir a ferramenta, será criado um banco de dados, para isso entre no menu **File > Create Database** e preencha o assistente como demonstrado a seguir.

- **Database File:** arquivo salvo em disco que corresponde ao banco de dados criado. Nessa aplicação será salvo o arquivo em `c:\Projeto\Qualificacao`. Observe que o arquivo não tem extensão porque ao ser importado para o emulador do Android ela não será utilizada.
- **Database Alias:** apelido para identificar o banco de dados. Será o nome utilizado no código-fonte para referenciar o banco.

A Figura 25 mostra a criação de um banco de dados. A interface é bem simples, mas contém todas as informações necessárias.

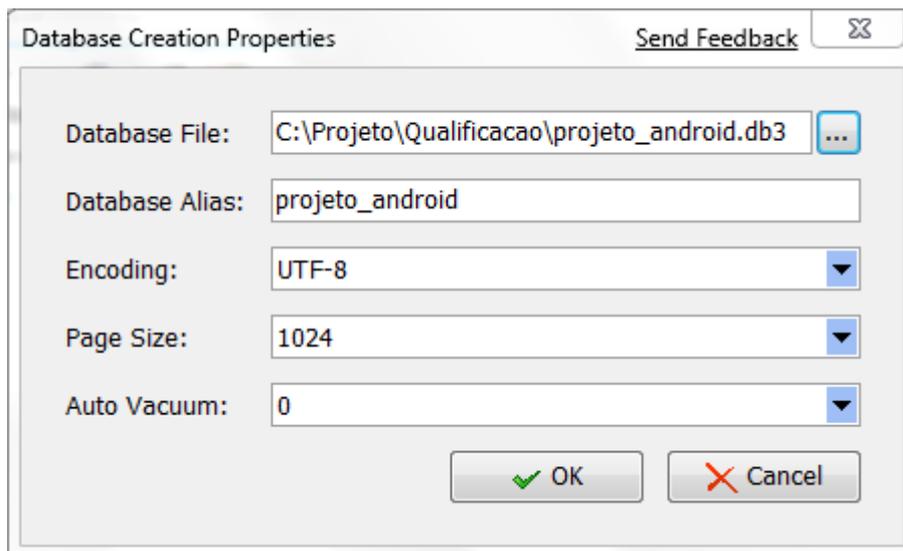


Figura 25: Criação de um banco de dados.

Depois de criar o banco de dados, é possível visualizar a árvore do banco à esquerda da ferramenta. Na aba **SQL** foi digitado o seguinte *script* para criar uma tabela chamada “alunos” e já inserido três alunos, conforme a Figura 26.

```
1 create table aluno (_id integer primary key autoincrement, nome text not null,
2 cidade text not null, curso text not null);
3 insert into aluno(nome, cidade, curso) values ('Joao', 'Paraguaçu Paulista', 'BCC');
4 insert into aluno(nome, cidade, curso) values ('Pedro', 'Paraguaçu Paulista', 'BCC');
5 insert into aluno(nome, cidade, curso) values ('Maria', 'Paraguaçu Paulista', 'BCC');
```

Figura 26: Criação da tabela e seus valores a serem inseridos.

Name	Declared Type	Type	Size	Precision	Not Null	Not Null On Conflict	Default Value	Collate
▶ _id	integer	integer	0	0	<input type="checkbox"/>			
nome	text	text	0	0	<input checked="" type="checkbox"/>			
cidade	text	text	0	0	<input checked="" type="checkbox"/>			
curso	text	text	0	0	<input checked="" type="checkbox"/>			

Figura 27: Tabela já criada e seus valores.

Após a criação do banco de dados, o próximo passo é importar o banco de dados criado para o emulador.

4.3.2- Estrutura de uma aplicação Android

A estrutura de uma aplicação Android possui quatro (o cinco) tipos básicos de elementos:

- Activities
- Intents e IntentFilters (e BroadcastReceivers)
- Services
- Content Providers

Nem toda aplicação necessita ter todos os tipos, mas haverá alguma combinação destes. Os componentes que compõe a aplicação são indicados em um arquivo chamado AndroidManifest.xml.

4.3.2.1- Android Manifest

- AndroidManifest.xml é necessário para cada aplicação.
- Descreve os elementos da aplicação: (Activities, Content Providers, Services, and Intent Receivers) globalmente visíveis, as classes de implementação de cada componente, e qual tipo de dado ele pode tratar, quando pode ser ativado, etc.

Para a ativação, contém os intentFilters que descrevem quando uma atividade deve ser iniciada.

- Pode-se também descrever as permissões e parâmetros de instrumentação.

4.3.2.2- Tipos de elementos

ACTIVITY: Uma *activity* é uma tela em uma aplicação Android. Cada *activity* é implementada como uma única classe que estende da classe base *Activity*. Essa classe irá mostrar uma interface composta por *Views* e que responde a eventos. A maioria das aplicações consiste em várias telas. Por exemplo, uma aplicação de mensagem pode ter uma tela que mostra uma lista de recados, uma segunda tela para escrever um recado para um contato e outra tela para rever os recados antigos ou mudar configurações. Cada tela deve ser implementada como uma *activity*. A mudança de uma tela para outra é feita no início de uma nova *activity*. Em alguns casos, uma *activity* tem que retornar um valor para a *activity* prévia. Por exemplo, uma *activity* que permite que o usuário escolha uma musica, retornaria a musica escolhida para o chamador. Quando abre uma tela nova, a tela chamadora é pausada e posta numa pilha. O usuário pode navegar de volta através das telas abertas anteriormente na pilha.

INTENT RECEPTOR: Usado quando o código de uma aplicação deve executar em reação a um evento externo, por exemplo, quando o celular toca ou quando os dados da rede estão disponíveis. Receptores de intenção não mostram uma

interface de usuário, embora eles possam usar o gerenciador de notificação para alertar o usuário quando algum evento de interesse ocorre. Os receptores de intenção são registrados no `AndroidManifest.xml`, mas também é possível registrá-los no código usando `Context.registerReceiver()`. Uma aplicação não necessita precisamente estar executando para que seus receptores de intenção sejam chamados. O sistema iniciará a aplicação quando um receptor de intenção é iniciado. Aplicações podem também enviar suas próprias intenções por broadcast para outras aplicações com `Context.broadcastIntent()`.

SERVICE: Serviço é o código que está executando em *background* sem uma interface gráfica. Um exemplo é um tocador de música. No tocador, há uma ou mais *activities* que permitem ao escolher músicas e iniciá-las. O tocador não deveria ser tratado por uma atividade porque o usuário espera que a música continue tocando em *background*. O sistema então irá manter o serviço do tocador de música executando até que ele acabe. É possível que uma aplicação conecte-se a um serviço para poder iniciá-lo (se ele não estiver executando) com `context.bindservice()`. Uma aplicação pode comunicar-se com um serviço através de uma interface do serviço. Para o serviço de música, ele pode ser usado para pausar uma música ou avançá-la, por exemplo.

CONTENT PROVIDER: As aplicações podem armazenar seus registros em arquivos, como o banco de dados SQLite. Um provedor de conteúdo, todavia, é útil se os dados da aplicação podem ser compartilhados com outras aplicações. Provedor de conteúdo é uma classe que implementa um conjunto padrão de métodos para permitir que outras aplicações armazenem e recuperem tipos de dados que são tratados pelo provedor de conteúdo.

4.3.2.3- Aplicação para inserir, alterar, e excluir alunos

No banco de dados que foi criado, o próximo passo é enviá-lo para o

emulador. Antes de enviá-lo é preciso criar o projeto no Eclipse, porque cada banco de dados é salvo no pacote de sua aplicação, de modo que o pacote precisa existir antes de o arquivo ser importado.

Após o projeto estar criado, crie uma classe **CadastrarAlunos**, como a *activity* principal. Feito isso, o projeto é executado pelo menu **Run as > Android Application**. Isso fará com que o projeto seja instalado no emulador e o pacote seja criado na estrutura de diretórios do sistema operacional.

Agora já é possível abrir a janela **File Explorer** pelo Eclipse e navegar até a seguinte pasta: `/data/data/br.aplicacao.android.qualificacao/databases/`. Após a execução dos passos, é enviado o banco de dados que foi salvo no caminho `c:\Projeto\Qualificacao` para essa pasta no emulador, conforme a Figura 27.

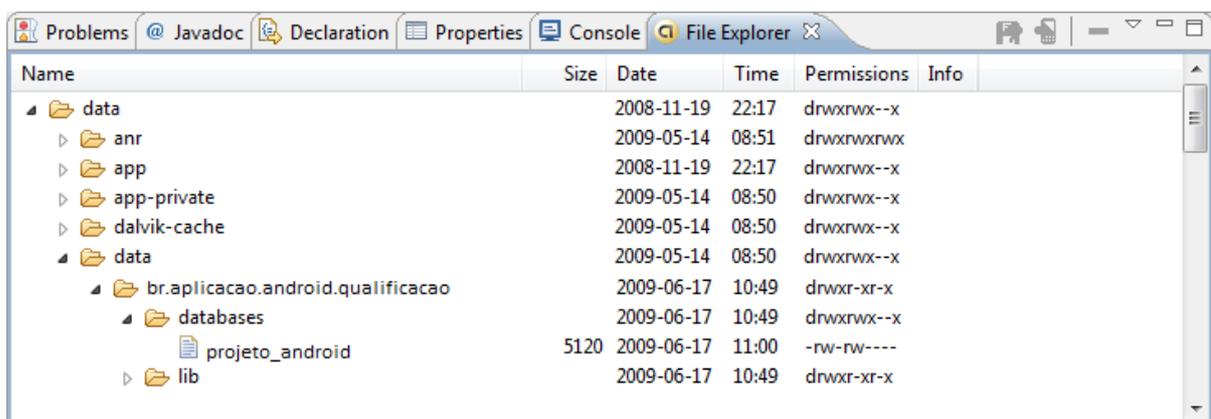


Figura 28: Envio do banco de dados para o emulador.

4.3.2.4- Inserção de registros no banco de dados

Para inserir registros no banco de dados é bem simples. Basta criar um objeto do tipo **android.content.ContentValues** com as informações necessárias, que funciona de forma similar a uma *HashTable*, com chave e valor. A implementação da inserção de alunos em uma tabela que contém as colunas: nomes, cidade e curso. O trecho do código fonte desta inserção é mostrado para se ter uma idéia de como fica a estrutura.

```

//Insere um novo aluno
ContentValues values = new ContentValues();
values.put("nome" , "Aluno x");
values.put("cidade", "Cidade y");
values.put("curso", "Curso z");

//insert into aluno ...
db.insert("aluno", null, valores);

```

Para os programadores habituados com SQL, o código fonte anterior é equivalente ao seguinte SQL.

```

insert into table aluno(nome, cidade, curso)
values ("Aluno x", "Cidade y", "Curso z");

```

4.3.2.5- Atualização de registros no banco de dados

Para atualizar um registro, utilizar o método **SQLiteDatabase.update(tabela, valores, where, whereArgs)** e passar uma string para o argumento *where* do método, onde o valor do id do registro pode ser utilizado para identificar o registro que deve ser atualizado. O trecho do código fonte para atualizar um aluno com o id=1 é mostrado para se ter uma idéia de como fica a estrutura.

```

//Atualiza o aluno 1
long id = 1;
String _id = String.valueOf(id);
ContentValues values = new ContentValues();
values.put("nome", "novo nome");
values.put("cidade", "nova cidade");
values.put("curso", "novo curso");

//Atualiza o carro com id=1
db.update("aluno", valores, "_id=" + id, null);

//Ou informando os parâmetros no terceiro parâmetro (array)
db.update("aluno", valores, "_id=?", new String[]{id});

```

Para os programadores habituados com SQL, o código fonte anterior é equivalente ao seguinte SQL.

```
Update aluno set nome="novo nome", cidade="nova cidade",  
curso= novo curso where _id=1;
```

4.3.2.6- Exclusão de registros no banco de dados

Para remover um registro, utilizar o método **SQLiteDatabase.delete(tabela, where, whereArgs[])** e passar uma string para o argumento *where* do método para identificar o registro, da mesma forma que foi explicada para realizar a atualização. O trecho do código fonte para atualizar um aluno com o id=1 é mostrado para se ter uma idéia de como fica a estrutura.

```
// Deleta o aluno 1  
long id = 1;  
String _id = String.valueOf(id);  
  
// Deleta o aluno com id=1  
db.delete("aluno", "_id="+id, null);  
  
//Ou informando o terceiro parâmetro  
b.delete("aluno", "_id=?", new String[]{_id});
```

Para os programadores habituados com SQL, o código fonte anterior é equivalente ao seguinte SQL.

```
Delete from aluno where _id=1;
```

4.3.2.7- Busca de registros no banco de dados

Para buscar informações no banco de dados, de forma semelhante a um SQL `select * from tabela...`, é utilizado o método `query(distinct ,tabela ,colunas ,selecao ,selecaoArgs ,groupBy ,orderBy)` informando as colunas desejadas e a cláusula `where`. O método `query(distinct ,tabela ,colunas ,selecao ,selecaoArgs ,groupBy ,orderBy)` retorna um objeto do tipo `android.database.Cursor` que tem um funcionamento similar ao `Java.sql.ResultSet` da API do JDBC. Com o objeto `android.database.Cursor`, é possível percorrer todos os registros retornados e recuperar os valores desejados.

Será mostrado como fazer para buscar as informações de um aluno utilizando seu nome. O SQL que representa essa ação seria algo assim:

SELECT _id, nome, cidade, curso from ALUNO where nome = ?

O trecho do código fonte necessário para fazer a mesma coisa que esse SQL é mostrado para se ter uma idéia de como fica a estrutura.

```
String nome = "nome do aluno procurado";

//Idem a: SELECT _id,nome,cidade,curso from ALUNO where nome = ?
Cursor c = db.query(TABELA_ALUNO, new String[] {"_id","nome","cidade","curso"}, "nome=?",
    new String[]{nome},null,null,null);

//Se encontrou...
if (c.getCount() > 0){
    Aluno aluno = new Aluno();

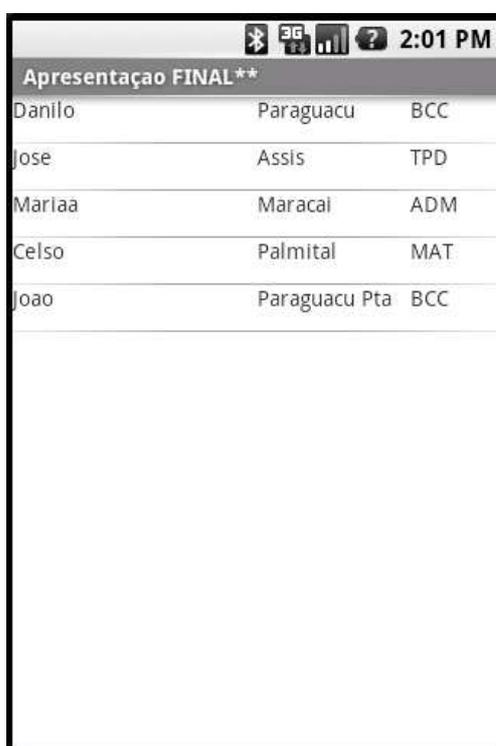
    //Posiciona no primeiro resultado
    c.moveToFirst();

    //Utiliza os métodos getLong(indice), getString(indice), getInt(indice) etc,
    // para recuperar os valores

    aluno.id = c.getLong(0);
    aluno.nome = c.getString(1);
    aluno.cidade = c.getString(2);
    aluno.curso = c.getInt(3);
```

4.3.2.8- Execução da aplicação de cadastro

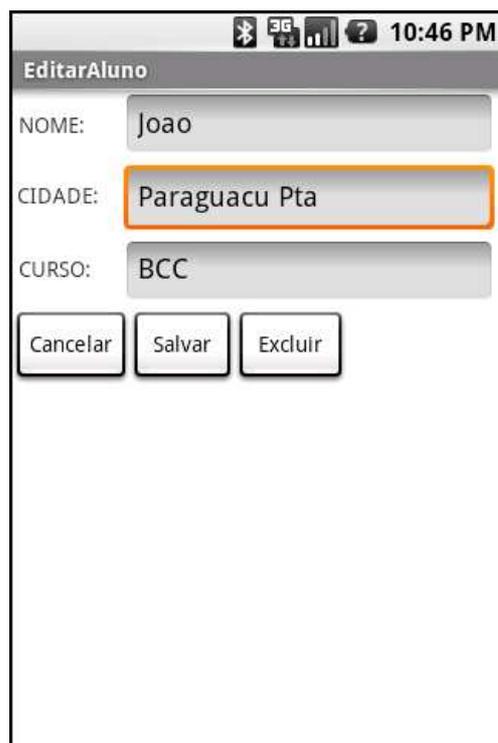
Depois de apresentar todos os códigos fonte do aplicativo para a plataforma Android através de seu emulador, utilizando um banco de dados para demonstrar uma simples movimentação de cadastro de alunos de uma faculdade, será mostrado algumas telas. A Figura 29, exibe a lista de alunos no emulador e o *menu* com as opções de inserir um novo aluno ou buscar pelo nome.



Apresentação FINAL**		
Danilo	Paraguacu	BCC
Jose	Assis	TPD
Mariaa	Maracai	ADM
Celso	Palmital	MAT
Joao	Paraguacu Pta	BCC

Figura 29: Tela com a lista de alunos.

Ao clicar no *menu* inserir, o formulário para inserir um aluno é aberto. Ao selecionar algum aluno na lista, o mesmo formulário é aberto, mas agora em modo de edição para o aluno selecionado, conforme a Figura 30.



EditarAluno

NOME: Joao

CIDADE: Paraguacu Pta

CURSO: BCC

Cancelar Salvar Excluir

Figura 30: Tela para inserir ou editar um aluno.

Nesse formulário é possível cancelar a tela, salvar ou excluir o aluno. Voltando à tela de lista de alunos, se a opção buscar for selecionada no *menu*, a tela de busca aparece e é possível buscar um aluno com o nome, conforme visualizado na Figura 31.

The image shows a mobile application interface titled "BuscarAluno". At the top, there is a status bar with icons for Bluetooth, 3G, signal strength, and battery, along with the time "10:46 PM". Below the title, there is a form with the following fields:

- Nome:** A text input field containing the name "Joad", which is highlighted with an orange border.
- Buscar:** A button with the text "Buscar".
- Cidade:** A text input field containing "Paraguacu Pta".
- Curso:** A text input field containing "BCC".

Figura 31: Tela para buscar um aluno.

CAPÍTULO 5

CONCLUSÃO

O interesse pelo tema é devido ao grande avanço tecnológico na área de *móviles*, e a necessidade de aplicativos para a plataforma Android. Utilizando um banco de dados em um celular pode se fazer n aplicações.

A implementação do modelo proposto é bem simples, porém mostra um caminho muito amplo nessa área. Através dele podem ser feitos aplicativos de modo geral que utilize o banco de dados. Fica claro, também, que programar para a plataforma Android é muito bom, dando sim uma liberdade enorme para seu desenvolvimento em varias áreas, tanto para o setor comercial como para o setor de entretenimentos. Com os conhecimentos adquiridos para o desenvolvimento deste projeto de pesquisa foi possível verificar um leque de opções profissionalmente. O objetivo, futuramente, é desenvolver aplicativos na área de entretenimento, principalmente em desenvolvimentos de jogos.

Algumas dificuldades surgiram em virtude desta plataforma ser nova e ainda pouco utilizada nos meios acadêmicos. Entretanto, foi possível desenvolver com sucesso uma pequena aplicação de cadastro.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Rabello, R.R.: *Android – Um Novo Paradigma de Desenvolvimento Móvel*, www.cesar.org.br/files/file/WM20_Android_LBS.pdf
- [2] Ableson, F.; Collins, C. and Sen R.: *Unlocking Android-A Developer's Guide*, Hanning, 2008.
- [3] Morimoto, C.E.: *Smarphones um Guia Prático*, Editora GDH Press e Sul Editores
- [4] Ableson, Frank. Introdução ao Desenvolvimento do Android. Disponível em < <http://www.ibm.com/developerworks/br/library/os-android-devel/index.html>>. Acesso em: Agosto 2009.
- [5] Morimoto, C. E.: Entendendo o Google Android, <http://www.guiadohardware.net/artigos/google-android/>. Acesso em: fevereiro 2009.
- [6] Faria, A.O.: *Programe seu Android*, Linux Magazine, N° 43, 2008.
- [7] Lecheta, R. R.: *Google Android-Aprenda a Criar Aplicações para Dispositivos Móveis com o Android SDK*, Novatec, SP, 2009.
- [8] <http://www.sqlite.org>. Acesso em março de 2009.
- [9] Endle, Markus. *A plataforma Android: Uma Introdução*. Departamento de Informática, 2009.
- [10] Rogers, Rick. *Android Application Development*, 1st Edition, 2009.