PROGRAMAÇÃO PARA DISPOSITIVOS MÓVEIS EM ANDROID - CCT0959

Semana Aula: 1

Aula 1 - Introdução à Computação móvel e ao Desenvolvimento de Aplicações Móveis

Tema

Introdução à Computação móvel e ao Desenvolvimento de Aplicações Móveis.

Palavras-chave

Computação móvel Desenvolvimento Móveis android

Objetivos

Apresentar o conceito de computação móvel, suas categorias bem como os fundamentos do desenvolvimento de aplicações móveis e suas abordagens.

Estrutura de Conteúdo

1. Computação móvel

A computação móvel se refere ao conjunto de tecnologias (hardware e software) para permitir a comunicação, processamento e armazenamento a qualquer hora e lugar. Dentre os dispositivos utilizados pode-se destacar os dispositivos móveis utilizados pelos usuários finais. Esse dispositivo tem como principais características: dimensões pequenas (em comparação aos PC's), portátil, capacidade de processamento e armazenamento local, conectividade sem fio e fonte de alimentação própria. Eles podem ser classificados conforme suas dimensões e capacidade de processamento: computadores móveis, computadores de mão e computadores de vestir.

A computação móvel está revolucionando o mundo quanto a interação e aplicações. O nº de dispositivos móveis vendidos no mundo supera o de PC's. Passa-se mais tempo acessando aplicações de um dispositivo móvel que assistindo TV. Acessa-se mais a internet via smartphone e tablet que via PC.

2. Desenvolvimento de Aplicações Móveis

A grande demanda por dispositivos móveis traz proporcionalmente a demanda por aplicações para atender aos seus usuários. O processo de desenvolvimento de aplicações para dispositivos móveis evolui rapidamente com as tecnologias destes. Atualmente destacam-se 3 grandes sistemas operacionais para estes dispositivos: IOS (Apple), Android(Google) e Windows Phone (Microsoft). Cada uma oferece uma plataforma de software para o desenvolvimento de aplicações para dispositivos móveis. O desenvolvimento de aplicações para dispositivos móveis podem ser categorizados em 4 tipos: Nativos, web mobile, híbrido e multiplataforma. A estratégia de desenvolvimento depende de requisitos, em especial: complexidade da aplicação, desempenho e segurança. Essas plataformas possuem lojas virtuais onde aplicações podem ser disponibilizadas para comercialização: Apple Store (Apple), Google Play (Google) e Windows Phone Store (Microsoft).Os tópicos abaixo relacionam os principais conceitos a serem apresentados nessa semana de aula:

1. Computação Móvel
a. Conceitos
    i. Computação móvel
    ii. Dispositivos móveis
b. Características essenciais
    ii. Mobilidade
c. Classificação
    i. Computadores móveis
    ii. Computadores de mão
    iii. Computadores vestíveis
d. Conceituação e caraterização de cada classe de dispositivo móvel
e. Tendências futuras

2. Desenvolvimento de Aplicações Móveis
a. Conceitos
b. Abordagens de desenvolvimento de app móveis
    i. Nativa - Uso de linguagem de programação suportada pela plataforma móvel e seu respectivo SDK - Software Development Kit. Plataforma específica.

    ii. Web mobile - Uso de linguagem web (html, css e javascript) e design responsivo. Multiplataforma.

    iii. Híbrida - Usa da solução nativa juntamente com solução web mobile. Plataforma específica.

    iv. Multiplataforma - Uso de ferramenta proprietária para geração de aplicações móveis para várias plataformas móveis.

c. Principais plataformas
d. Mercado

Estratégias de Aprendizagem

Indicação de Leitura Específica

Aplicação: articulação teoria e prática

Os sites abaixo apresentam videos e textos que abordam os principais conceitos a serem apresentados pelo professor.
http://m.olhardigital.uol.com.br/video/dispositivos-moveis/4682
Site:http://www.mobilebit.com.br/
Site:http://news.artics.com.br/desenvolvimento-mobile-qual-a-melhor-maneira-de-desenvolver-apps/
Site:http://www-03.ibm.com/software/products/pt/category/mobile-application-development

PROGRAMAÇÃO PARA DISPOSITIVOS MÓVEIS EM ANDROID - CCT0959

Semana Aula: 2

Aula 2 - Plataforma de Desenvolvimento Móvel

Tema

Plataforma de Desenvolvimento Móvel - Visão geral e introdução a plataforma Android

Palavras-chave

Plataforma Desenvolvimento Móvel

Objetivos

Apresentar os conceitos básicos de plataformas de desenvolvimento móvel, e aprofundar os conhecimentos na plataforma Android.

Estrutura de Conteúdo

As plataformas de desenvolvimento de aplicações móveis fornecem um conjunto de softwares para a utilização e desenvolvimento de aplicações. Cada uma das principais plataformas móveis do mercado (IOS, Android e Windows Phone) possuem características próprias, tais como: arquitetura do Sistema Operacional, SDK - kit de desenvolvimento padrão, ambiente de desenvolvimento, linguagens de desenvolvimento suportadas. Cada uma dessas plataformas têm requisitos específicos para instalação do ambiente de desenvolvimento, bem como procedimento e valor de assinatura para publicação de aplicações para comercialização nas suas respectivas lojas virtuais.

A plataforma Android vem se destacando como a principal plataforma para dispositivos móveis. É a mais utilizada no mundo, segundo pesquisas de mercado. Ocupando mais de 50% do mercado de dispositivos vendidos e oferece mais de 1 milhão de aplicações na Google Play. Várias versões foram disponibilizadas desde seu lançamento em 2008. Atualmente conta com a versão 5.0 - Lollipop para smartphones e tablets. O Android SDK, kit de desenvolvimento da Google composto pelo framework Android e ferramentas como o emulador e o gerenciador de SDK. Oambiente de desenvolvimento Android correspondende a uma IDE - Integrated Development Environment, composto por editores e plugins para o Android.O A linguagem de programação base para o Android é o Java. Toda a estrutura da linguagem é utilizada juntamente com o conjunto de classes base do Android para o desenvolvimento de aplicações nessa plataforma. Os tópicos abaixo representam os principais conceitos a serem apresentados nessa semana de aula:

1.1 Plataformas de Desenvolvimento Móvel
a.Características
b.Linguagens de programação base
c.Pre-requisitos do ambiente de desenvolvimento
d.Custo de produção
e.Procedimento de publicação
f.Visão de mercado

2.Introdução à plataforma de desenvolvimento Móvel Android
a.Conceito
b.Histórico
c.Versões
d.Características da última versão
e.Principais classes
f.Ambiente de desenvolvimento oficial Android
g.Instalação e configuração

Estratégias de Aprendizagem

Indicação de Leitura Específica

Aplicação: articulação teoria e prática

O aluno deverá realizar uma leitura dos textos abaixo e implementar os exemplos práticos apresentados nos textos abaixo:

Livro: Google Android: Crie aplicações para celulares e tablets (João Bosco Monteiro)
Capítulo 1 Construa sua primeira aplicação

Livro: Google Android: Aprenda a criar aplicações para dispositivos móveis com o Android SDK (Ricardo R. Lecheta)
Capítulo 1 - Introdução ao Android
Capítulo 2 - Configuração do ambiente

Site oficial da Google Android, seção do desenvolvedor:http://developer.android.com/training/index.html

Considerações Adicionais

PROGRAMAÇÃO PARA DISPOSITIVOS MÓVEIS EM ANDROID - CCT0959

Semana Aula: 3

Aula 3 - Fundamentos da Programação Móvel com Java

Tema

Fundamentos da Programação Móvel com Java

Palavras-chave

Programação Móvel Java

Objetivos

Apresentar os conceitos básicos da programação orientada a objetos com Java, e sua integração com o SDK do Android.

Estrutura de Conteúdo

O desenvolvimento de aplicações para Android requer o conhecimento da programação orientada a objetos na linguagem JAVA. Uma aplicação Android é composta por classes Java integradas com as bibliotecas do Android. Uma classe descreve características e funcionalidades de objetos do mundo real. Um objeto é a representação virtual de um item do mundo real. É uma instância (materialização) de uma classe. Criar componentes de software de uma aplicação Android requer o conhecimento da aplicação dos princípios da orientação a objetos na linguagem JAVA: abstração (classes), encapsulamento (Java Beans - POJO´s), herança (extensão de classes), composição (classes formadas por outras classes) e polimorfismo (implementação de interfaces por classes distintas e sobrescrição de comportamentos entre classe Pai e classe Filha). Abaixo são apresentados os principais tópicos abordados ao longo da semana de aula:

1. Visão geral da Programação OO com Java
a. Visão geral da Orientação a Objetos
b. Princípios da Orientação a Objetos e a Linguagem Java
i. Abstração - Representação da essência de um objeto dentro de um contexto.
ii. Encapsulamento - proteger a estrutura de dados e oferecer serviços para manipulá-los.
iii. Herança - Permitir definir novos objetos a partir de objetos existentes.
iv. Composição - Permitir criar objetos complexos através da associação com objetos existentes.
v. Polimorfismo - Permitir criar serviços padronizados porém com comportamento diferente.
c. Programação orientada a interfaces
d. Controle e tratamento de exceções

Estratégias de Aprendizagem

Indicação de Leitura Específica

Aplicação: articulação teoria e prática

O aluno deverá ler os textos e implementar os exemplos apresentados nos tutoriais abaixo:

Site:http://www.palmbrasil.com.br/android/faq/129-desenvolvimento-de-aplicacoes-para-android/1937-e-necessario-conhecer-java-para-programar-para-android

Site:http://www.thiengo.com.br/infografico-10-conhecimentos-para-um-bom-desenvolvimento-em-android

Site:http://code.tutsplus.com/tutorials/learn-java-for-android-development-introduction-to-java--mobile-2604

Site:http://code.tutsplus.com/tutorials/learn-java-for-android-development-introduction-to-java--mobile-2604

Considerações Adicionais

PROGRAMAÇÃO PARA DISPOSITIVOS MÓVEIS EM ANDROID - CCT0959

Semana Aula: 4

Aula 4 - Introdução à Programação Móvel Android

Tema

Introdução à programação Móvel com Java e Android.

Palavras-chave

Programação Móvel Android

Objetivos

Integrar a linguagem de programação Java com o SDK do Android para o desenvolvimento de aplicações Android simples e executá-los em um dispositivo virtual Android.

Estrutura de Conteúdo

Programar para Android requer conhecimentos básicos da linguagem Java, porém é necessário o conhecimento de classes básicas do Android para a construção de uma aplicação móvel. O SDK do Android fornece um conjunto de API's que permitem a construção de telas e a para a interação com o usuário, bem como para realizar a navegação entre telas. As principais classes do Android são: Activity - Classe base do Android que representa o controle de uma tela, View - Conjunto de componentes utilizados na construção da interface gráfica do usuário, Intent - Classe do Android que permite acesso a recursos do dispositivo, bem como a navegação entre telas de uma aplicação, Services - Classe que permite realizar ações como serviços, não possui interface com o usuário e Content Providers - Classe que permite registrar e acessar fontes de dados compartilháveis entre aplicações.

O desenvolvimento de aplicações Android faz uso da linguagem de programação Java, ou seja, programa-se em Java referenciando-se as classes do Android. Na construção de uma aplicação, a tela é um item fundamental. O Android emprega a arquitetura MVC como padrão na construção de aplicações Android, onde a interface é construida de forma declarativa (XML) e o controlador é uma classe Java que extende a classe do Android Activity. A interação com a tela se dá através do tratamento de eventos de toque, implementando-se o método onClick() da interface OnClickListener do Android. Isso pode ser realizado de 4 formas: através de classe anônima associada a um componente do tipo widget, através da implementação da interface OnClickListener pela classe, ou através da ligação entre a propriedade onClick de um componente widget ao método da classe Java que especializa a Activity. Mensagens de alerta podem ser criadas no Android através das classes: AlertDialog e Toast. O AVD manager é uma ferramenta provida pelo SDK para a criação de dispositivos móveis virtuais para execução e teste da aplicação. Permite criar vários dipositivos móveis virtuais que emulam o ambiente de um dispositivo móvel real.

Os tópicos abaixo apresentam os principais conceitos a serem apresentados ao longo da semana de aula.

1. Fundamentos da programação Android
    a. Principais classes do Android
        i. Activity - Classe base para criação de telas no Android.
        ii. View - Componentes gráficos utilizados na criação da interface do usuário.
        iii. Intent - Recurso para navegação entre telas e execução de aplicações externas no Android.
        iv. Services - Recurso para criação de serviços que são executados em segundo plano.
        v. Content Providers - Recurso para compartilhamento de fontes de dados entre aplicações Android.
    b. Integração Android e Java
    c. MVC e Android
    d. Implementação de uma tela Android.
        i. Arquitetura MVC e Android
        ii. Activity - Classe da API do Android para construção de telas.
        iii. Layout - Gerenciadores de layout para estruturação dos componentes visuais da interface do usuário.
        iv. Views -Componentes visuais da interface do usuário.
    e. Tratamento a eventos de click.
        i. A interface OnClickListener
        ii. Formas de implementação:
            - classe anônima
            - implementação da interfaceOnClickListener
            - propriedade "onclick" dos componentes visuais.
    f. Exibindo mensagens através de janelas de diálogo.
        i. AlertDialog - Janela de diálogo que permite a interação com o usuário.
        ii. Toast - Janela de diálogo temporária para exibição de mensagens sem interação com o usuário.
    g. O AVD Manager e a criação de um dispositivo móvel virtual.
    h. Criação e execução de uma aplicação simples.

Estratégias de Aprendizagem

Indicação de Leitura Específica

Aplicação: articulação teoria e prática

O aluno deverá ler e implementar os exemplos apresentados nos capítulos abaixo:

Livro: Google Android: Crie aplicações para celulares e tablets (João Bosco Monteiro)
Capítulo 2 Entenda o funcionamento do Android

Livro: Google Android: Aprenda a criar aplicações para dispositivos móveis com o Android SDK (Ricardo R. Lecheta)
Capítulo 3 - Conceitos básicos do Android

Site oficial da Google Android, seção do desenvolvedor: http://developer.android.com/training/index.html

Considerações Adicionais

PROGRAMAÇÃO PARA DISPOSITIVOS MÓVEIS EM ANDROID - CCT0959

Semana Aula: 5

Aula 5 - Entendendo a classe Activity

Tema

Activity - Aprofundamento e aplicações

Palavras-chave

Classe Activity

Objetivos

Apresentar de forma detalhada a classe Activity, a pilha de execução de uma aplicação Android, osestados de uma Activity e seu ciclo de vida.

Estrutura de Conteúdo

A classe Activity é a principal classe do Android no desenvolvimento de aplicações móveis para essa plataforma. A Activity representa uma tela do Android. Cada tela do Android é descrita em um arquivo de layout (XML) porém controlada pela classe Activity. A implementação de uma Activity se dá pela especialização da classe Activity do Android. Cada aplicação Android é composta por várias telas (Activity). As telas em execução são empilhadas na pilha de execução de Activity, gerenciadas pelo sistema operacional Android. Uma Activity possui como método principal o método "onCreate()". Esse método é responsável por inicializar a tela, definir qual layout exibir e aplicar as regras de negócio de sua utilização. A Activity possui um ciclo de vida dividido em 9 métodos callback: onCreate, onStart, onResume, onPause, onStop, onRestart e onDestroy. Cada um desses métodos são invocados durante a execução de uma aplicação.

Para acompanhar o funcionamento interno e o monitoramento de uma aplicação Android, o SDK fornece a classe Log para permitir realizar o Logging de mensagens em 5 categorias: INFO, VERBOSE, DEBUG, WARNING e ERROR. O SDK fornece também a ferramenta de visualização de log "LogCat", que através de um plugin (que já vém do ambiente de desenvolvimento) permite visualizar mensagens de log, bem como criar filtros para seleção de mensagens.

Os tópicos abaixo representam os principais conceitos a serem apresentados nessa semana de aula:

1. A vida de uma Activity
    a. A pilha de execução
    b. Os estados de uma Activity
    c. O ciclo de vida de uma Activity
    d. Subníveis do ciclo de vida
    e. Logging - procedimento deregistro de atividades realizadas durante a execução de uma aplicação que pode ser utilizado por administradores de sistemas para diagnosticar problemas.
    f. LogCat - Ferramenta do SDK Android para acompanhamento de logs do Android.

Estratégias de Aprendizagem

Indicação de Leitura Específica

Aplicação: articulação teoria e prática

O aluno deverá ler e implementar os exemplos apresentados nos capítulos abaixo:

Livro: Google Android: Crie aplicações para celulares e tablets (João Bosco Monteiro)
Capítulo 2 Entenda o funcionamento do Android

Livro: Google Android: Aprenda a criar aplicações para dispositivos móveis com o Android SDK (Ricardo R. Lecheta)
Capítulo 4 - Activity

Site oficial da Google Android, seção do desenvolvedor: http://developer.android.com/training/index.html

Considerações Adicionais

PROGRAMAÇÃO PARA DISPOSITIVOS MÓVEIS EM ANDROID - CCT0959

Semana Aula: 6

Aula 6 - Navegação entre Telas Android

Tema

Navegando entre telas Android através de Activity e Intents,

Palavras-chave

Navegação Telas Android

Objetivos

Implementar aplicações Android composta por várias telas através do uso da classe Intent do Android.

Estrutura de Conteúdo

A navegação entre telas e aplicações Android se dá através da instância da classe Intent do Android. Essa classe permite enviar uma mensagem ao sistema operacional Android informando o que ser deseja realizar (que recurso deseja ser acessado: outra tela dentro do projeto, a aplicação externa da câmera, ou de audio, etc). A Intent permite anexar dados utilizados na comunicação entre telas/aplicações. Esses dados podem ser desde primitivas Java até tipos complexos como classes. O envio de instância de classes entre telas (Activity) requer que sejam classes serializadas (java.io) ou que implementem a interface Parcelable do Android.

A navegação no Android através da classe Intent pode ser realizado de 2 formas básicas: navegação simples, onde uma Activity requer a execução de outra Activity, não criando vínculo entre elas. Ou através da navegação com retorno de dados, onde uma Activity requer a execução de outra Activity, porém aguardando o retorno de dados da Activity chamada para finalizar o processo.

Os tópicos abaixo apresentam os principais conceitos a serem apresentados ao longo da semana de aula:

1. Navegação entre Telas
    a. Métodos da Activity de navagação entre telas
    b. A classe Intent
    c. Navegação simples entre telas
    d. Navegação com retorno de resposta entre telas
    e. Passagem de parâmetros entre telas
    f. Passando classes como parâmetros entre telas
        i. Serializable - Interface Java que permite a representação binária de um objeto para transferência entre aplicações.
        ii. Parcelable - Interface Android que permite a transferência de um objeto entre aplicações.

Estratégias de Aprendizagem

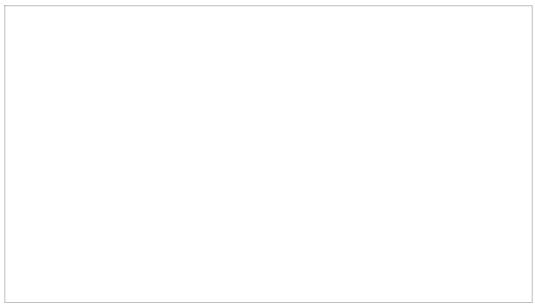
Prezado(a) aluno(a),

Para este tema, selecionamos as tarefas que deverão ser realizadas **DEPOIS** da aula presencial, a fim de que você participe dela mais efetivamente:

1-Execute o Android Studio

2-Copie e descompacte o projeto aula6.zip, dentro existe a pasta aula

3-abra pelo Android Studio



4- Crie a uma nova Activity (empty) com o nome de TelaInicial

e adicione o método:

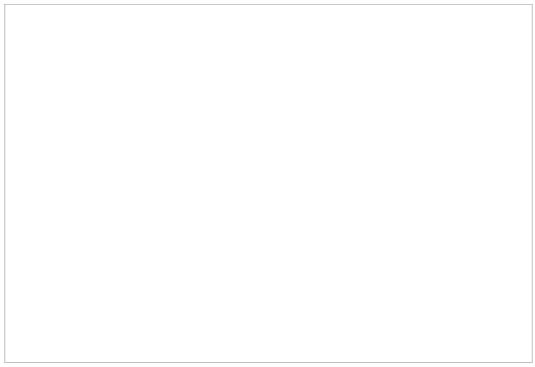
```
public void proximaTela(View v){

Intent i=new Intent(this, TelaInicial.class);

startActivity(i);
}
```

5-Adicione uma ImageView e altere a propriedade source da imageView para a Imagem

Agenda.jpg



6-volte na tela telaInicial e altere a propriedade OnClick da imageview apontando para a próximaTela.

7-Sincronize e execute o projeto, verifique se os botões estão respondendo e o clicar da figura inicial troca de tela

Com isso, esperamos que você seja capaz de:

· Gerenciar um projeto na IDE de trabalho

· Iniciar a preparação de uma interface

· Fazer a navegação entre as telas

Indicação de Leitura Específica

Aplicação: articulação teoria e prática

O aluno deverá ler e implementar os exemplos apresentados nos capítulos abaixo:

Livro: Google Android: Crie aplicações para celulares e tablets (João Bosco Monteiro)
Capítulo 2 Entenda o funcionamento do Android

Livro: Google Android: Aprenda a criar aplicações para dispositivos móveis com o Android SDK (Ricardo R. Lecheta)
Capítulo 5 - Intent

Site oficial da Google Android, seção do desenvolvedor: http://developer.android.com/training/index.html

Considerações Adicionais

PROGRAMAÇÃO PARA DISPOSITIVOS MÓVEIS EM ANDROID - CCT0959

Semana Aula: 7

Aula 7 - Construção de telas (GUI)

Tema

Construção da Interface Gráfica do Usuário

Palavras-chave

Construção telas GUI

Objetivos

Conhecer os componentes View (view, viewgroup e widget) e sua utilização na construção da interface gráfica do usuário em aplicações Android.

Estrutura de Conteúdo

Uma tela Android é composta por componentes View. View a classe base do Android que representam os componentes utilizados na construção da interface gráfica do usuário (GUI). A View possui dois tipos: widget e Viewgroup. Views do tipo widget representam os componentes de tela visualizados pelo usuário e utilizados para interação com a aplicação. Views do tipo Viewgroup são utilizados para organizar os componente widget na tela. São gerenciadores de layout.

O Android fornece diversos componentes widgets. Ele provê widgets básicos como: TextView, EditText e Button, mais especializados como RadioButton, Checkbox, DatePicker e TimePicker, e os componentes baseados em listas como: Spinner e ListView.

O Android fornece também diversos tipos de gerenciadores de layout, tais como: LinearLayout, RelativeLayout, TableLayout e GridLayout dentre outros.

A GUI pode ser criada de duas formas: Declarativa ou Programática. A forma declarativa se refere a criação de telas através de arquivos XML de layout. Esta forma emprega a arquitetura MVC, separando a tela em 3 partes: o layout (XML) que representa a porção VIEW, a Activity que representa o CONTROLLER e as classes POJO do Java que representam os MODEL.

Os tópicos abaixo apresentam os principais conceitos a serem apresentados ao longo da semana de aula:

1. Componentes da interface gráfica do usuário
    a. Classe View
        i. View - Classe Android que representa componentes gráficos de uma tela Android.
        ii. Widget x Viewgroup
        iii. Árvore de componentes de layout
    b. Tipos de Viewgroups básicos
        i. LinearLayout - Gerenciador de layout onde os componentes widget são dispostos linearmente.
        ii. RelativeLayout- Gerenciador de layout onde os componentes widget são dispostos com base na posição relativa de outro componente.

        iii. AbsoluteLayout- Gerenciador de layout onde os componentes widget são dispostos baseados nas posições absolutas X e Y do sistemas de coordenadas da tela.

        iv. TableLayout- Gerenciador de layout onde os componentes widget são dispostos em forma de tabela.
        v. FrameLayout- Gerenciador de layout onde os componentes widget são dispostos em forma de empilhamento (somente 1 widget será visualizado por vez).

        vi. GridLayout- Gerenciador de layout onde os componentes widget são dispostos em forma de grid.

        vii. ScrollView- Gerenciador de layout que acrescenta a rolagem vertical a tela.

    c. Componentes Widget
        i. TextView - componente visual que representa um texto fixo (rótulo) na tela.
        ii. EditText- componente visual que representa uma caixa de entrada de dados na tela.
        iii. Button- componente visual que representa um botão de ação na tela.
        iv. Checkbox- componente visual que representa um item selecionável na tela.
        v. Radiobutton- componente visual que representa um item de seleção única na tela.
        vi. DateTime Picker- componente visual que representa data e hora na tela.
        vii. Spinner- componente visual similar a um combobox.
        viii. ImageView- componente visual que representa uma imagem na tela.
    d. Formas de implementação de telas
        i. Declarativa - A tela é descrita através do uso de tags XML de componentes View do Android.
        ii. Programática -A tela é descrita através da instanciação de classes de componentes do tipo View do Android.

    e. Ferramentas de prototipação de telas

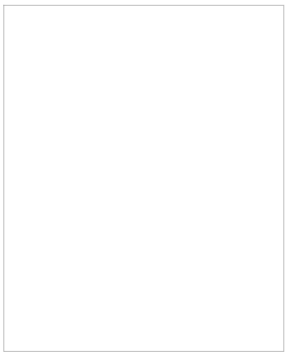
Estratégias de Aprendizagem

1-Execute o Android Studio

2-Copie e descompacte o projeto aula7.zip, dentro existe a pasta aula

3-Abra pelo Android Studio

4-Edite a interface gráfica app-res-layout-telacadastro.xml para a tela abaixo:



5- Crie o modelo Pessoa (classe simples) com os atributos nome e telefone

```
public class Pessoa {

private String nome;
private String telefone;

public String getNome() {
return nome;
}

public void setNome(String nome) {
this.nome = nome;
}

public String getTelefone() {
return telefone;
}

public void setTelefone(String telefone) {
this.telefone = telefone;
}
}
```

4-Na classe Activity adicione os métodos de controle para os botões:

```
public void cadastrar(View v)
{
String n=((EditText)findViewById(R.id.editText)).getText().toString();
String t=((EditText)findViewById(R.id.editText2)).getText().toString();
Pessoa p=new Pessoa();
p.setNome(n);
p.setTelefone(t);

Toast.makeText(this,"Cadastrado com sucesso",Toast.LENGTH_LONG).show();


}
```

6-Configure a propriedade OnClick do Botão na janela de propriedades para

7-Sincronize e execute o projeto, verifique se os botões estão respondendo

Com isso, esperamos que você seja capaz de:

· Realizar a implementação de funcionalidades em GUI

· Atualizar uma aplicação existente

Indicação de Leitura Específica

Aplicação: articulação teoria e prática

O aluno deverá ler e implementar os exemplos apresentados nos capítulos abaixo:

Livro: Google Android: Crie aplicações para celulares e tablets (João Bosco Monteiro)
Capítulo 2 Entenda o funcionamento do Android
Capítulo 3 Domine os principais elementos de Interface Gráfica

Livro: Google Android: Aprenda a criar aplicações para dispositivos móveis com o Android SDK (Ricardo R. Lecheta)
Capítulo 6 - Interface gráfica ? gerenciadores de layout
Capítulo 7 - Interface gráfica ? View

Site oficial da Google Android, seção do desenvolvedor: http://developer.android.com/training/index.html

Considerações Adicionais

PROGRAMAÇÃO PARA DISPOSITIVOS MÓVEIS EM ANDROID - CCT0959

Semana Aula: 8

Aula 8 - Construção de telas complexas

Tema

Construção de Interface Gráfica Avançada

Palavras-chave

telas complexas

Objetivos

Implementar telas gráficas complexas através do aninhamento de gerenciadores de layout do Android.

Estrutura de Conteúdo

O Android permite criar interfaces complexas compostas por diversos widgets. Telas complexas podem ser criadas através do aninhamento de Viewgroups. Essa forma permite que os componentes do tipo widget sejam organizados de diversas formas. O aninhamento de viewgroups se refere a incluir um viewgroup dentro de outro viewgroup. Assim é possível incluir um TableLayout dentro de um LinearLayout vertical para permitir organizar parte do layout da tela de forma horizontal.

A organização de componentes na tela permitiu criar certos padrões para dispositivos Android. Dentre esses padrões tem-se: Dashboard, Menulists, etc. Conhecer esses padrões se torna importante para criação de aplicações intuitivas e segundo um padrão de navegação conhecidos.

Abaixo seguem os tópicos a serem aborados ao longo da semana de aula:

1. Aninhando Viewgroups para interface gráficas complexas
    a. Aninhamento de Viewgroups
    b. Combinação de Viewgroups
    c. Padrões de Projeto de Interface Gráfica do Usuário

Estratégias de Aprendizagem

Indicação de Leitura Específica

Aplicação: articulação teoria e prática

O aluno deverá ler e implementar os exemplos apresentados nos sites e capítulos de livro abaixo:

Site:http://www.devmedia.com.br/artigo-webmobile-21-construindo-layouts-complexos-em-android/10761

Artigo:http://www.cesar.org.br/site/files/file/WM21_Android_Layouts.pdf

Livro: Google Android: Crie aplicações para celulares e tablets (João Bosco Monteiro)
Capítulo 2 Entenda o funcionamento do Android
Capítulo 3 Domine os principais elementos de Interface Gráfica

Livro: Google Android: Aprenda a criar aplicações para dispositivos móveis com o Android SDK (Ricardo R. Lecheta)
Capítulo 6 - Interface gráfica - gerenciadores de layout
Capítulo 7 - Interface gráfica - View

Site oficial da Google Android, seção do desenvolvedor: http://developer.android.com/training/index.html

Considerações Adicionais

PROGRAMAÇÃO PARA DISPOSITIVOS MÓVEIS EM ANDROID - CCT0959

Semana Aula: 9

Aula 9 - Construção de Telas baseadas em Lista de Itens

Tema

Construção de telas baseadas em listas de itens

Palavras-chave

Lista Itens

Objetivos

Implementar telas gráficas compostas por listas de itens simples e complexas.

Estrutura de Conteúdo

O Android provê um conjunto de componentes widget utilizado para construção da interface gráfica do usuário (GUI). Dentre esses componentes pode-se destacar os os componentes que apresentam uma lista de itens para que o usuário possa selecionar um item. A caixa de seleção, muito conhecida na internet como Combobox, que apresenta uma lista de itens para que o usuário possa selecionar um item. Muito comum para seleção da unidade da federação (UF), em alguns casos também utilizado para seleção de municípios. A lista de contatos, do smartphone é outro exemplo de widget baseado em lista. Diferente do primeiro exemplo, a lista de contatos apresenta a relação de contatos em que você pode selecionar um contato para detalhar os dados ou realizar uma ligação telefônica para o contato selecionado.

O Spinner é nome do componente widget do Android que representa uma caixa de seleção similar ao Combobox, onde apenas um item da lista pode ser selecionado. O ListView é o nome de outro componente widget do Android que representa a lista de dados descrito no exemplo da lista de contatos.

Esses componentes representam elementos gráficos de interação do usuário com a aplicação. A lista de dados exibidos devem ser repassados para esses widgets através de adaptadores de dados. Esses adaptadores definem quais dados deverão ser exibidos e qual o layout a ser exibido. O Android provê diversas classes que representam esses adaptadores, tais como: BaseAdapter, ArrayAdapter, CursorAdapter, etc. O Layout de exibição da lista de dados pode ser baseado em um layout predefinido do Android (android.R.layout.<nome do tipo do layout>), ou definido pelo desenvolvedor. Neste último caso é possível criar layouts complexos, onde cada item da lista apresenta diversos dados a serem exibidos. Por exemplo, uma lista de contatos onde cada item da lista apresenta a foto do contato, o nome e o número telefônico.

O Spinner e o ListView permite responder a eventos de toque sobre o item. Para isso é necessário habilitar um ouvinte(listener) a evento de toque ao componente. No caso do Spinner para permitir o tratamento ao evento de seleção de um item da lista deve-se implementar o método onItemSelected da interface OnItemSelectedListener. Já o ListView, deve-se implementar o método onItemClick da interface OnItemClickListener.

O Android provê um tipo de tela preconfigurado que representa uma tela composta por uma lista de itens (ListView). Essa tela é uma ListActivity. Essa classe já possui um componente ListView implementado que ocupa a tela toda. Para utilizá-la deve-se apenar especializar a ListActivity, em seu método onCreate( ) deve-se apenas criar o adaptador de dados e atribuí-la a lista através do método setListAdapter( ) da ListActivity. O tratamento de eventos se dá através a implementação do método onListItemClick( ), não sendo necessário implementar nenhuma interface.

Os tópicos abaixo representam os principais conceitos a serem apresentados nessa semana de aula:

1. Listas de Itens
    a. Conceitos - Componentes baseados em listas de itens.
    b. Tipos de Listas de itens
        i. ListView - Classe Android que representa um componente de tela em forma de lista de itens.
        ii.ListActivity - Classe Android que representa uma tela pré-configurada com um ListView.

    c. Tratamento ao evento de seleção de um item da lista
        i. ListView - implementação do evento ItemClickListener.
        ii. ListActivity- implementação do evento ListItemClickListener.

2. Adaptadores
    a. Conceito de Adaptadores
    b. Tipos de Adapters
    c. Formas de alimentação de listas de itens

3. Criando Listas de Itens
    a. Lista de Itens Simples - itens compostos por texto simples
    b. Lista de Itens Complexas - itens compostos por vários componentes widgets

Estratégias de Aprendizagem

Indicação de Leitura Específica

Aplicação: articulação teoria e prática

O aluno deverá ler e implementar os exemplos apresentados nos sites e capítulos de livros abaixo:

Site:http://developer.android.com/guide/topics/ui/layout/listview.html

Site:http://www.linhadecodigo.com.br/artigo/3331/introducao-listview-no-android.aspx

Livro: Google Android: Crie aplicações para celulares e tablets (João Bosco Monteiro)
Capítulo 3 Domine os principais elementos de Interface Gráfica

Livro: Google Android: Aprenda a criar aplicações para dispositivos móveis com o Android SDK (Ricardo R. Lecheta)
Capítulo 7 - Interface gráfica ? View

Site oficial da Google Android, seção do desenvolvedor: http://developer.android.com/training/index.html

Considerações Adicionais

PROGRAMAÇÃO PARA DISPOSITIVOS MÓVEIS EM ANDROID - CCT0959

Semana Aula: 10

Aula 10 - O Componente Menu e Fragmentos de Telas

Tema

Construção de Menus e Interface Gráfica baseados em Fragmentos

Palavras-chave

Menu Fragmento Tela

Objetivos

Implementar telas gráficas compostas por menus, bem como telas utilizando conceitos de Fragments.

Estrutura de Conteúdo

Dispositivos baseados em Android apresentam 3 botões de navegação: Back Key (voltar), Home (Inicio) e . Esses botões são reconfigurado pelo Android e auxiliam o usuário a navegar entre as aplicações instaladas no dispositivo. Esses dipositivos possuem limitação quanto ao tamanho do display, e como consequencia impacto no nº de funcionalidades que uma aplicação pode exibir para interação do usuário. Para contornar esse problema, o Android permite criar Menus através de componentes específicos para essa finalidade. Os componentes de menu permitem criar menus de opção ou menus popups (um menu de contexto). Os menus podem ser agrupados para permitir que compartilhem propriedades como estados ativados ou visibilidade.

O método onOptionsItemSelected() deve ser implementado para tratar o evento de seleção de um item do menu. Assim pode-se implementar a lógica da funcionalidade para o botão do menu selecionado. Esse método tem como parâmetro uma instância da classe MenuItem, alimentado pelo Android quando o item do menu for selecionado. Através dessa instância pode-se recuperar informações para identificar qual item do menu foi pressionado e executar a respectivia funcionalidade.

As primeiras versões do Android tinha enfoque voltados para Smartphones, com limitação quanto ao tamanho do display fisico. Com o surgimento dos tablets os display aumentaram de tamanho. A dupla: Activity + layout foram criados para atender a implementação de aplicações para smartphones. Para tablets eles se tornaram um problema. A versão 3.0 - Honeycomb introduziu o conceito de Fragments para permitir a construção de fragmentos de telas, facilitando a criação de aplicações para displays com diferentes dimensões.

Um Fragment (fragmento) é um componente autônomo de interface do usuário. Similar a Activity, ele possui um ciclo de vida e um gerenciamento de estado (FragmentManager). Um Fragment pode ser atrelado a uma Activity (Attached), podendo estar visivel ou não na tela.

O Fragment permite criar telas dinâmicas mais facilmente em relação a utilização da Activity + Layout. Pode-se usar o Fragment em versões anteriores a versão 11 (HoneyComb).

Os tópicos abaixo apresentam os principais conceitos a serem abordados ao longo da semana de aula:

1. Construção da Telas compostas por Menu
    a. Componentes de menu
    b. tipos de menus
        i. de opção - menu composto por opções de funcionalidades.
        ii. de contexto - menu exclusivo para um item de uma lista
        iii. popup - menu composto por lista de opções na vertical
        iv. grupos de menus - submenus
    c. Tratamento a eventos de menu

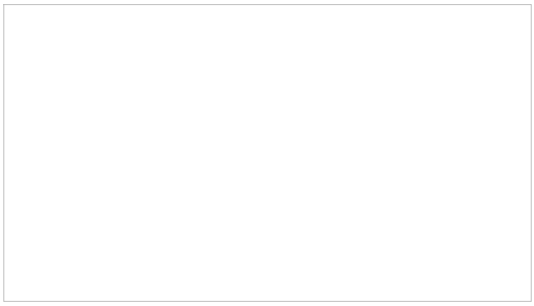
2. Construção de telas baseadas em Fragments
    a. Conceitos de Fragments
    b. Tipos de fragments
    c. Ciclo de vida de um Fragment
    d. Integrando Activity e Fragments
    e. Implementação comportamento de Fragments
    f. Reuso de Fragments.

Estratégias de Aprendizagem

1-Execute o Android Studio

2-Copie e descompacte o projeto aula10.zip, dentro existe a pasta aula

3-Abra pelo Android Studio



4- Crie adicione um menu, com o cursor em res, botão direito, new resource file, tipo menu

5-Volte na classe Activity TelaInicial e adicione o método:

```
public boolean onCreateOptionsMenu(Menu menu) {

  MenuInflater inflater = getMenuInflater();
  inflater.inflate(R.menu.menuprincipal, menu);


  return super.onCreateOptionsMenu(menu);
 }

 public boolean onOptionsItemSelected(MenuItem item) {
  switch (item.getItemId()) {
  case R.id.menuRefresh:
  // go home
  Intent intent = new Intent(this, TelaInicial.class);
  startActivity(intent);
  return true;

  }

  return super.onOptionsItemSelected(item);
 }
```

6-Edite o menu.xml criado e adicione um item de menu:

```
<?xml version="1.0" encoding="utf-8"?>
<menu
xmlns:android="http://schemas.android.com/apk/res/android">
<item android:id="@+id/menuRefresh"
android:icon="@android:drawable/ic_popup_sync"
android:title="agenda"
android:alphabeticShortcut="r"
android:showAsAction="always" />

</menu>
```

7-Sincronize e execute o projeto, verifique se os botões estão respondendo e o clicar da figura inicial troca de tela

Com isso, esperamos que você seja capaz de:

· Implementar a funcionalidade de Menu em um App

· Atualizar uma aplicação existente

Indicação de Leitura Específica

Aplicação: articulação teoria e prática

O aluno deverá ler e implementar os exemplos apresentados nos sites e capítulos de livros abaixo:

Site:http://developer.android.com/guide/topics/ui/menus.html
Site:http://escoladeandroid.blogspot.com.br/2012/02/android-criando-menu.html

Site:http://developer.android.com/guide/components/fragments.html
Site:https://androiddevbr.wordpress.com/2014/01/30/entendendo-fragment/
Site:http://blog.caelum.com.br/layouts-mais-simples-com-android-fragments/

Livro: Google Android: Crie aplicações para celulares e tablets (João Bosco Monteiro)
Capítulo 9 Suporte Tablets e outros dispositivos

Livro: Google Android: Aprenda a criar aplicações para dispositivos móveis com o Android SDK (Ricardo R. Lecheta)
Capítulo 7 - Interface gráfica ? View

Considerações Adicionais

PROGRAMAÇÃO PARA DISPOSITIVOS MÓVEIS EM ANDROID - CCT0959

Semana Aula: 11

Aula 11 - Mensagens, Estilos e Temas

Tema

Padronização de mensagens, estilos e temas

Palavras-chave

Mensagens Estilos Temas

Objetivos

Projetar e implementar padronização de telas de uma aplicação através de mensagens, estilos e/ou temas.

Estrutura de Conteúdo

O Android permite adicionar textos fixos a rótulos e mensagens em janelas de diálogo diretamente através da propriedade TEXT. Essa forma não segue as regras de boas praticas do Android, que sugere a declaração em formato <string name="chave"> valor </string>, no arquivo XML chamado strings.xml, constante na pasta res/values do projeto Android.

Cada string declarada nesse arquivo pode ser referenciado na propriedade TEXT utilizando-se a anotação @string/<chave>. Essa forma de trabalho possui muitas vantagens, em especial facilitar a manutenção a aplicação e a internacionalização dessas mensagens. Pode-se criar versões do arquivo strings.xml para diversas línguas, como o inglês, o espanhol, etc. O Android seleciona o arquivo de mensagens apropriado a partir das configurações do aparelho. Isso amplia o domínio de usuários conforme o nº de línguas suportadas pela sua aplicação.

Similar a aplicações web, o Android permite criar arquivos XML que predefinem a aparencia dos componentes de uma tela. Arquivo de estilos (Style) representa um conceito similiar ao CSS da web. Esse arquivo permite definir um padrão de aparência aos widgets do Android e são descritos estilos (style) padronizadas no arquivo ?styles.xml?, localizado na pasta "/res/values". Cada estilo (style) possui um nome que é utilizado como referência em um ou mais widgets. Isso promove a padronização da aparência dos componentes em diversas telas, o reuso da definição a vários componentes widgets, bem como a facilidade de manutenção.

Estilos podem ser aplicado a uma activity inteira, ou em toda a aplicação. Esse tipo de estilo é chamado de Theme (Tema). Ele é configurado no arquivo AndroidManifest.xml através das tags <activity ... android:theme=?valor?> ou <application ... android:theme=?valor?>.

Os tópicos abaixo apresentam os principais conceitos a serem abordados ao longo da semana de aula:

1. Padronização de Mensagens
    a. Definindo mensagens padronizadas (strings.xml)
    b. Referenciando mensagens padronizadas no Layout
    c. Obtendo mensagens padronizadas na Activity
    d. Internacionalização de mensagens

2. Estilos
    a. Definindo cores
    b. Referenciando cores no Layout
    c. Definindo estilos
    d. Referenciando estilo no Layout
    e. Aplicando estilo na aplicação

3. Temas
    a. Conceito de Tema (Theme)
    b. Tipos de temas
    c. Definindo temas
    d. Herança
    e. Referenciando tema no Layout
    f. Aplicando tema na aplicação

Estratégias de Aprendizagem

Indicação de Leitura Específica

Aplicação: articulação teoria e prática

O aluno deverá ler e implementar os exemplos apresentados nos sites e capítulos de livros abaixo:

Site:http://developer.android.com/guide/topics/resources/string-resource.html
Site:http://mobgeek.com.br/blog/apps-android-multi-idiomas
Site:http://www.devmedia.com.br/alterando-o-idioma-da-aplicacao-android-de-acordo-com-o-idioma-do-aparelho/27365
Site:https://developer.android.com/samples/BasicMediaRouter/res/values/colors.html
Site:http://celeiroandroid.blogspot.com.br/2011/03/interface-de-usuarios-estilos-e-temas.html

Livro: Google Android: Aprenda a criar aplicações para dispositivos móveis com o Android SDK (Ricardo R. Lecheta)
Capítulo 7 - Interface gráfica ? View

Considerações Adicionais

PROGRAMAÇÃO PARA DISPOSITIVOS MÓVEIS EM ANDROID - CCT0959

Semana Aula: 12

Aula 12 - Introdução a Persistência de Dados

Tema

Persistência de Dados no Android

Palavras-chave

Persistência Dados

Objetivos

Projetar e implementar persistência de dados em aplicações Android.

Estrutura de Conteúdo

É muito comum utilizarmos aplicações Android que permitem armazenar dados e recuperar esses dados. Aplicações como a lista de contatos e configurações fazem uso desse tipo de solução, persistindo dados na memória interna ou externa (cartão SD) do dispositivo. Em geral os dados persistidos na memória interna são persistidos e visíveis a que as criou. Caso seja necessário compartilhar esses dados deve-se fazer uso da classe do Android "ContentProvider".

O Android provê 3 formas básicas de persistência de dados: através da classe SharedPreferences, ou através de Streams (Java.io) ou através de um banco de dados relacional (SQLite). Os dados são mantidos dentro do pacote da aplicação, e somente a aplicação que criou o banco de dados ou o arquivo pode acessá-las. A classe do Android "ContentProvider" representa o conceito de provedor de conteúdo. Seu objetivo é permitir que determinados dados sejam públicos para qualquer aplicação instalada no dispositivo móvel.

Toda a comunicação entre aplicações e providers é feita através dos métodos da interface ContentProvider. A comunicação entre uma aplicação e um Content Providers é realizada através de um URI. Um ContentProveider provê métodos para enviar e recuparar dados a um provider, dentre eles tem-se : query(), insert( ), update( ), delete( ) e getType( ).

Os tópicos abaixo apresentam os principais conceitos a serem abordados ao longo da semana de aula:

1. Persistência de dados
    a. Conceito de persistência de dados em dispositivos móveis
    b. Formas de Persistência de dados no Android

2. Persistindo em Arquivos Texto/Binários
    a. Conceito de arquivos textos e binários
    b. Classes InputStream e OutputStream
    b. Forma de implementação
    c. Aplicação da persistência em arquivos textos e binários

3. Persistindo com SharedPreference
    a. Conceito de SharedPreference
    b. Forma de implementação
    c. Persistindo preferências do usuário.

4. Content Providers
    a. Conceito de provedor de conteúdo
    b. Manipulação de um provedor de conteúdo
    c. Criação de um provedor de conteúdo
    d. ContentProvider básicos

Estratégias de Aprendizagem

1-Execute o Android Studio

2-Copie e descompacte o projeto aula12.zip, dentro existe a pasta aula

3-Abra pelo Android Studio

4- Crie a classe simples DAOPessoa para manipular o banco agenda do SQLite

```
import android.content.Context;
 import android.database.Cursor;
 import android.database.sqlite.SQLiteDatabase;
 import android.database.sqlite.SQLiteOpenHelper;

 public class DaoPessoa extends SQLiteOpenHelper{
```

```
  public DaoPessoa(Context context) {
  super(context, "Agenda",null , 1);
  }

  public void onCreate(SQLiteDatabase sqLiteDatabase) {
  sqLiteDatabase.execSQL("create Table pessoa(nome varchar(20),tel varchar(20))");
  }
  public void cadastrar(Pessoa p){
  getWritableDatabase().execSQL("insert into pessoa values(?,?)",new String[]{p.getNome(),p.getTelefone()});

  }
  public Pessoa consultar(Pessoa p){
  Cursor c=getReadableDatabase().rawQuery("select * from pessoa where nome=?", new String[]{p.getNome()});
  if(c.moveToNext())
  p.setTelefone(c.getString(1));
  c.close();
  return p;
  }
  public void excluir(Pessoa p){
  getWritableDatabase().execSQL("delete from pessoa where nome=?",new String[]{p.getNome(),p.getTelefone()});

  }
  public void alterar(Pessoa p){
  getWritableDatabase().execSQL("update pessoa set tel=? where where nome=?",new String[]{p.getTelefone(),p.getNome()});
  }

  @Override
  public void onUpgrade(SQLiteDatabase sqLiteDatabase, int i, int i1) {

  }
 }

}
}
```

5-Volte na classe MainActivity e acerte os métodos:

```
public void cadastrar(View v)
 {
  String n=((EditText)findViewById(R.id.editText)).getText().toString();
  String t=((EditText)findViewById(R.id.editText2)).getText().toString();
  Pessoa p=new Pessoa();
  p.setNome(n);
  p.setTelefone(t);

  daop.cadastrar(p);
  Toast.makeText(this,"Cadastrado com sucesso",Toast.LENGTH_LONG).show();


 }
 public void consultar(View v)
 {
  String n=((EditText)findViewById(R.id.editText)).getText().toString();

  Pessoa p=new Pessoa();
  p.setNome(n);
  p=daop.consultar(p);

  Toast.makeText(this,"telefone;"+p.getTelefone(),Toast.LENGTH_LONG).show();


 }
 public void excluir(View v)
 {
  String n=((EditText)findViewById(R.id.editText)).getText().toString();
  String t=((EditText)findViewById(R.id.editText2)).getText().toString();
  Pessoa p=new Pessoa();
  p.setNome(n);
  p.setTelefone(t);
  daop.excluir(p);
  Toast.makeText(this,"Excluído com sucesso",Toast.LENGTH_LONG).show();


 }
 public void alterar(View v)
 {
  String n=((EditText)findViewById(R.id.editText)).getText().toString();
  String t=((EditText)findViewById(R.id.editText2)).getText().toString();
  Pessoa p=new Pessoa();
  p.setNome(n);
  p.setTelefone(t);
  daop.alterar(p);
  Toast.makeText(this,"Alterado com sucesso",Toast.LENGTH_LONG).show();


 }
```

6-Sincronize e execute o projeto, verifique se os botões estão respondendo

Com isso, esperamos que você seja capaz de:

· Persistir os dados de um APP em um SGBD

· Atualizar uma aplicação existente

Indicação de Leitura Específica

Aplicação: articulação teoria e prática

O aluno deverá ler e implementar os exemplos apresentados nos sites e capítulos de livros abaixo:

Site:http://developer.android.com/training/building-content-sharing.html
Site:http://www.mobiltec.com.br/blog/index.php/tutorial-android-parte-4-persistencia-de-dados/

Livro: Google Android: Crie aplicações para celulares e tablets (João Bosco Monteiro)
Capítulo 4 Persistência de dados no Android com SQLite
Capítulo 5 Compartilhe dados entre aplicações com os Content Providers

Livro: Google Android: Aprenda a criar aplicações para dispositivos móveis com o Android SDK (Ricardo R. Lecheta)
Capítulo 14 - Banco de dados
Capítulo 15 - Content Provider

Considerações Adicionais

PROGRAMAÇÃO PARA DISPOSITIVOS MÓVEIS EM ANDROID - CCT0959

Semana Aula: 13

Aula 13 - Persistência de Dados com SQlite

Tema

Persistência de Dados em Banco de dados SQlite no Android.

Palavras-chave

Persistência Dados SQlite

Objetivos

Projetar e implantar aplicações Android com persistência de dados em um banco de dado SQLite.

Estrutura de Conteúdo

Dados podem ser persistidos em um Banco de Dados no Android. O Android provê uma API para interação com banco de dados SQLite. O SQLite é um banco de dados pequeno e poderoso. É multiplataforma e não pertence ao Android. A API SQLite do Android permite executar instruções SQL DDL(Data Definition Language) e DML(Data Manipulation Language) do banco de dados SQLite.

A API SQLite provê 2 classes básicas para manipulação do BD: SQLiteOpenHelper e SQLiteDatabase. A classe SQLiteOpenHelper deve ser especializada em uma classe JAVA da aplicação onde define-se os métodos onCreate e onUpgrade para tratar as regras para criação do BD e atualização da versão do BD da aplicação. Essa classe permite também realizar a abertura do BD para somente Leitura ou Gravação, retornando uma instância da classe SQLiteDatabase. A classe SQLiteDatabase por sua vez permite a execução de instruções DML, provendo métodos tais como: execSQL, rawQuery, query, insert, update e delete.

Os tópicos abaixo apresentam os principais conceitos a serem abordados ao longo da semana de aula:

1. Persistência em BD em dispositivos Android
    a. Conceito de Banco de Dados
    b. Principais SGBD's para dispositivos móveis
    c. ORM para dispositivos móveis

2. Persistindo dados com SQLite
    a. Conceito SQLite
    b. Ferramentas externas
    c. Formas de implantação BD SQLite
    d. Criando um BD SQLite
    e. Instruções SQL do SQLite

Estratégias de Aprendizagem

Indicação de Leitura Específica

Aplicação: articulação teoria e prática

O aluno deverá ler e implementar os exemplos apresentados nos sites e capítulos de livros abaixo:

Site:http://developer.android.com/reference/android/database/sqlite/package-summary.html
Site: http://www.mobiltec.com.br/blog/index.php/android-persistencia-de-dados-usando-sqlite/

Livro: Google Android: Crie aplicações para celulares e tablets (João Bosco Monteiro)
Capítulo 4 Persistência de dados no Android com SQLite

Livro: Google Android: Aprenda a criar aplicações para dispositivos móveis com o Android SDK (Ricardo R. Lecheta)
Capítulo 14 - Banco de dados

Considerações Adicionais

PROGRAMAÇÃO PARA DISPOSITIVOS MÓVEIS EM ANDROID - CCT0959

Semana Aula: 14

Aula 14 - Implementando solução Android com API SQlite

Tema

Uso da API SQLite na implementação de classes de persistência de Dados no Android.

Palavras-chave

Android API SQlite

Objetivos

Implementar aplicação Android que permita realizar funcionalidades CRUD (Create, Read, Update, Delete) em um BD SQLite.

Estrutura de Conteúdo

A construção de uma aplicação Android que faz acesso a banco de dados pode ser dividida em camadas tradicionais: camada de apresentação, camada de negócio e camada de dados. A camada de dados contém as classes Java que implementam o padrão DAO - Data Access Object. Essa classe utiliza a uma classe Java do projeto que especializa a classe SQLiteOpenHelper para atuar como gerenciador da acesso ao BD. A classe DAO deve então prover métodos para persistência de dados como: insert, update, delete ou query.

Para ilustrar a utilização desse padrão e o acesso a um BD SQLite, será criada uma aplicação junto ao professor, ligando os conceitos vistos anteriormente com o acesso a base de dados SQLite.

Os tópicos abaixo apresentam os principais conceitos a serem abordados ao longo da semana de aula:

1. Fundamentos da API SQLite
    a. Características da API
    b. Principais classes e métodos

2. Usando o padrão DAO com API SQLite
    a. Conceito DAO
    b. Implementação da classe DAO
    c. Implementação do CRUD
    d. Carregando uma ListView com dados do BD
    e. Controle de transações no SQLite

Estratégias de Aprendizagem

Indicação de Leitura Específica

Aplicação: articulação teoria e prática

O aluno deverá ler e implementar os exemplos apresentados nos sites e capítulos de livros abaixo:

Site: http://www.mobiltec.com.br/blog/index.php/android-persistencia-de-dados-usando-sqlite/
Site:http://programmerguru.com/android-tutorial/android-sqlite-crud-example/
Site:http://www.devmedia.com.br/realizando-operacoes-de-crud-no-android-com-java/25855
Site:http://www.thecodebakers.org/2011/03/licao-3-criando-um-crud-no-sqlite.html

Livro: Google Android: Crie aplicações para celulares e tablets (João Bosco Monteiro)
Capítulo 4 Persistência de dados no Android com SQLite

Livro: Google Android: Aprenda a criar aplicações para dispositivos móveis com o Android SDK (Ricardo R. Lecheta)
Capítulo 14 - Banco de dados

Considerações Adicionais

PROGRAMAÇÃO PARA DISPOSITIVOS MÓVEIS EM ANDROID - CCT0959

Semana Aula: 15

Aula 15 - Consumindo Webservice no Android

Tema

Implementação cliente Android para consumir serviços web.

Palavras-chave

Webservice Android

Objetivos

Conhecer os fundamentos de Web Services e implementar aplicações Android que consumam web services.

Estrutura de Conteúdo

Para acessar um Web Service via Android deve-se utilizar a biblioteca KSOAP2.
Um Web Service é um padrão utilizado na integração de dados entre aplicações ou sistemas heterogêneos. Isso permite que aplicações desenvolvidas em tecnologias distintas possam realizar a troca de dados utilizando a internet como meio, transimitindo e recebendo dados por meio de requisições e respostas HTTP.

Aplicações Android podem consumir serviços web (web services) para expandir o acesso a dados, seja para obter dados da web, ou armazenar dados na web. Exemplos básicos como utilizar o serviço de consulta a CEP da empresa de correios e telégrafos ECT.
Isso permite tornar as aplicações mais leves, permite reutilizar funcionalidades existentes, acesso e manutenção de dados em um servidor remoto, etc.

O Android dá suporte a criação de classes clientes para consumir serviços web baseados em protocolo SOAP ou REST. Permite também obter e tratar dados em formato XML ou JSON.

Os tópicos abaixo apresentam os principais conceitos a serem abordados ao longo da semana de aula:

1.Fundamentos de Web Services
- a. Conceito de web service e seus componentes básicos
- b. Arquitetura e componentes
- c. Padrões:
  - i.SOA (Service-Oriented Architecture) -estilo de arquitetura de software onde as funcionalidades de aplicações devem ser disponibilizadas na forma de serviços. Padrão rígido.
  - ii.REST (Representational State Transfer) -estilo arquitetural que consiste de um conjunto coordenado de restrições arquiteturais aplicadas a componentes, conectores e elementos de dados dentro de um sistema de hipermídia distribuído.
- d. Integração de aplicações Movel com Web

2. Implementação cliente para consumir Webservice REST
- a. Conceito de web service do tipo REST
- b. Características de um web service REST
- c. Padrão JSON (JavaScript Object Notation) - formato leve para intercâmbio de dados computacionais.
- d. Implementando cliente webservice REST no Android

Estratégias de Aprendizagem

Indicação de Leitura Específica

Aplicação: articulação teoria e prática

O aluno deverá ler e implementar os exemplos apresentados nos sites e capítulos de livros abaixo:

Artigo:http://www.cesar.org.br/site/files/file/WM23_Android_WebServices.pdf
Site:http://www.portalandroid.org/comunidade/viewtopic.php?f=7&t=17465

Livro: Google Android: Crie aplicações para celulares e tablets (João Bosco Monteiro)
Capítulo 6 Integração de aplicações Android com serviços REST

Livro: Google Android: Aprenda a criar aplicações para dispositivos móveis com o Android SDK (Ricardo R. Lecheta)
Capítulo 17 - Http, sockets e web services

Considerações Adicionais

PROGRAMAÇÃO PARA DISPOSITIVOS MÓVEIS EM ANDROID - CCT0959

Semana Aula: 16

Aula 16 - Integrando Serviços Google a uma aplicação Android

Tema

Apresentação dos serviços do Google que podem ser integrados a aplicações Android.

Palavras-chave

Serviços Google Android

Objetivos

Conhecer os principais serviços oferecidos pelo Google para integração em aplicações móveis Android.

Estrutura de Conteúdo

O Google oferece uma variedade de serviços que permite gerir a distribuição aplicativo, rastrear o uso de aplicativo, e melhorar o seu aplicativo com recursos como mapas login ea messaging nuvem.

Estes serviços não estão incluídos na plataforma Android, porém são suportados pela maioria dos dispositivos com Android. Dentre os serviços pode-se destacar: Google Maps, Google+, Cloud Platform Google, Google Cloud Messaging, Google Cloud Salvar, Google Play In-App Billing, Google Wallet Instantânea Comprar, Google Analytics, Google Mobile Ads.

O Android provê um conjunto de classes que permite realizar a integração de sua aplicação a esses serviços oferecidos pelo Google. Dentre eles a API GPS integrado ao serviço do Google Maps para geolocalização.

Os tópicos abaixo apresentam os principais conceitos a serem abordados ao longo da semana de aula:

1. Serviços do Google para Android
    a. Introdução a serviços do Google
    b. Principais serviços do Google
    c. integração serviços Google a aplicação Android.

Estratégias de Aprendizagem

Indicação de Leitura Específica

Aplicação: articulação teoria e prática

O aluno deverá ler e implementar os exemplos apresentados nos sites e capítulos de livros abaixo:

Site:http://blog.caelum.com.br/usando-o-google-maps-e-gps-no-android/

Livro: Google Android: Crie aplicações para celulares e tablets (João Bosco Monteiro)
Capítulo 7 Utilize Google APIs e crie funcionalidades interessantes

Livro: Google Android: Aprenda a criar aplicações para dispositivos móveis com o Android SDK (Ricardo R. Lecheta)
Capítulo 16 - Mapas e GPS

Considerações Adicionais