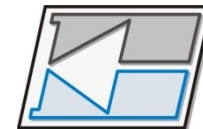


Introdução à Programação Android

Prof. Samir Bonho

Aula 5

Florianópolis, 26 de Março de 2014

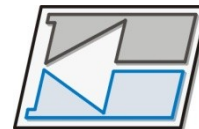


Sumário

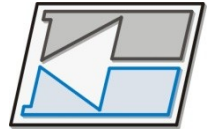
✓ Persistência de Dados

✓ Fontes:

✓ Android Developers. Disponível em <http://developer.android.com/>

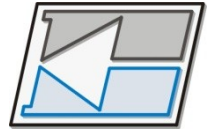


Armazenamento



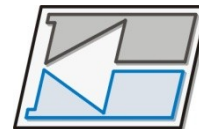
Armazenamento

- Armazenar o que?
 - Pontuação de um usuário em um jogo;
 - Configurações do aplicativo;
 - Dados de login, senha, preferências;
 - Informações sobre a rede, dados de navegação...

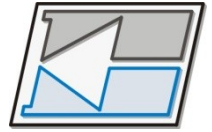


Armazenamento

- Como armazenar?
 - Múltiplas tabelas relacionadas entre si
 - Base de dados existente em um aparelho celular tradicional e um repositório remoto de dados existente na web
 - Arquivos na memória interna do dispositivo
 - Arquivos na memória externa do dispositivo

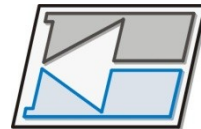


Opções de Armazenamento



Armazenamento: Android

- **Shared Preferences**
 - Permite armazenar dados de tipo primitivo utilizando o formato chave-valor.
- **Internal Storage**
 - Armazenamento na memória interna do dispositivo.
- **External Storage**
 - Armazenamento na memória externa do dispositivo, geralmente em SD Cards.

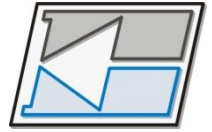


- **SQLite Databases**

- Tabelas estruturadas num banco de dados do dispositivo.

- **Network Connection**

- Armazenamento em algum servidor web acessado via conexão TCP/IP.



Armazenamento: Android

- **Shared Preferences**

- Armazena dados de tipo primitivo utilizando o formato chave-valor.
- Trabalha com todos os tipos primitivos incluindo também o tipo String.
- Útil no armazenamento de preferências e configurações de uma aplicação: poucos dados primitivos.

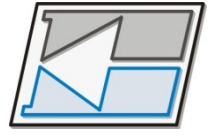
```
protected void onCreate(Bundle state){
    super.onCreate(state);
    ...

    // Restore preferences
    SharedPreferences settings = getSharedPreferences(PREFS_NAME, 0);
    boolean silent = settings.getBoolean("silentMode", false);
    setSilent(silent);
}

@Override
protected void onStop(){
    super.onStop();

    // We need an Editor object to make preference changes.
    // All objects are from android.context.Context
    SharedPreferences settings = getSharedPreferences(PREFS_NAME, 0);
    SharedPreferences.Editor editor = settings.edit();
    editor.putBoolean("silentMode", mSilentMode);

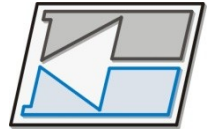
    // Commit the edits!
    editor.commit();
}
}
```



Armazenamento: Android

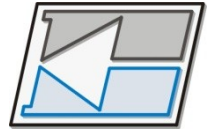
- **Internal Storage**

- Armazena dados na estrutura de arquivos interna da aplicação.
- Os arquivos salvos são exclusivos da aplicação e não podem ser acessados por outros aplicativos
- Quando o aplicativo é desinstalado o arquivo é deletado junto.
- Arquivos podem ser manipulados pelas classes na biblioteca java.io.
- Cada aplicação possui seu próprio espaço.
 - Normalmente data/data/nome_pacote



• Internal Storage

```
try {  
    // Cria o arquivo onde serão salvas as informações.  
    File file = new File(getFilesDir().getPath()+"/arquivo_internal.txt");  
    FileWriter = new FileWriter(file, true);  
    fileWriter.append(texto);  
    fileWriter.append("n");// Quebra de linha.  
    fileWriter.append(String.valueOf(numero));  
    // Escreve no arquivo.  
    fileWriter.flush();  
} catch (IOException e) {Log.e("Erros", "Erro ao salvar usando Internal Storage", e); } finally {  
    // Fecha os recursos.  
    if (fileWriter != null) {  
        try {fileWriter.close();}  
        catch (Exception e){}  
    }  
}
```

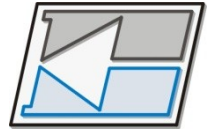


Armazenamento: Android

- **External Storage**

- Armazena dados na memória externa do dispositivo.
- Necessário adicionar permissão no manifest:
 - **READ_EXTERNAL_STORAGE** OU **WRITE_EXTERNAL_STORAGE**

```
<uses-permission  
android:name="android.permission.WRITE_EXTERNAL_STORAGE"></use  
s-permission>
```



Armazenamento: Android

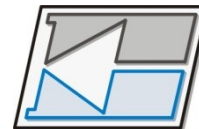
- **External Storage**

- Sempre necessário checar se a mídia de armazenamento externo está disponível.

- -
 -
 - String mediaState = Environment.getExternalStorageState();

- if (mediaState.equals(Environment.MEDIA_MOUNTED))

- -
 -



Exercícios