

Bancos de Dados Móveis



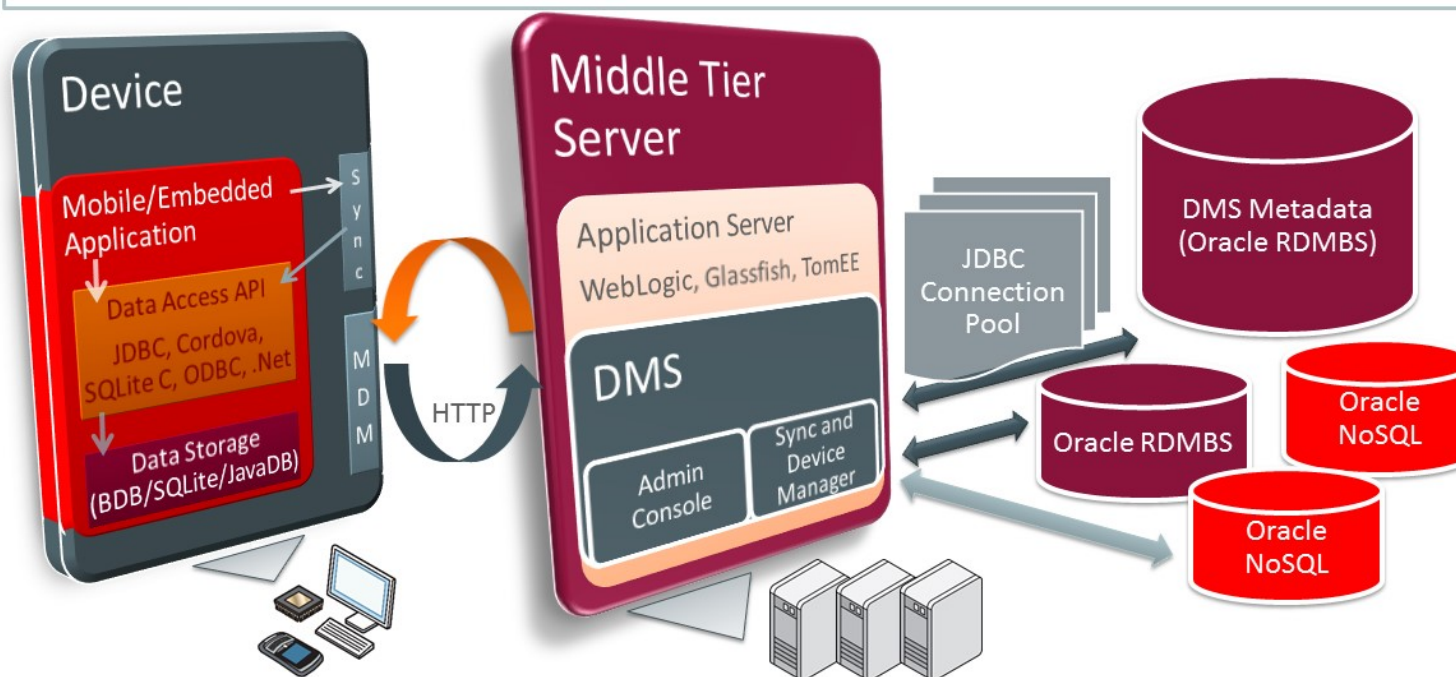
Ferramentas

- Oracle Database Mobile Server 12c
 - Aplicativo de sincronização avançada
 - Responsável pela sincronização de bancos remotos com uma base consolidada
- Projetado para redes conectadas
 - Se a conectividade for perdida, ele interrompe o funcionamento e retoma em momentos de conexão
- Características adicionais
 - Criptografia no cliente e na transmissão



Arquitetura

Database Mobile Server Architecture



Histórico

- Oracle JDeveloper
 - Biblioteca gratuita que objetiva simplificar o desenvolvimento de aplicações SOA e Java EE
 - Inclui o Oracle Mobile Application Framework (MAF)
 - Facilita a integração de fontes de dados remotas com dados móveis da aplicação
 - Implementa data caching e sincronização em operações conectadas



Arquitetura

- Web Services (SOAP, REST)
 - Obtenção e atualização para e do banco
- POJOs
 - Cache local dos dados para acesso quando desconectado
- Database
 - Banco de dados local (SQLite) para persistência da aplicação desconectada



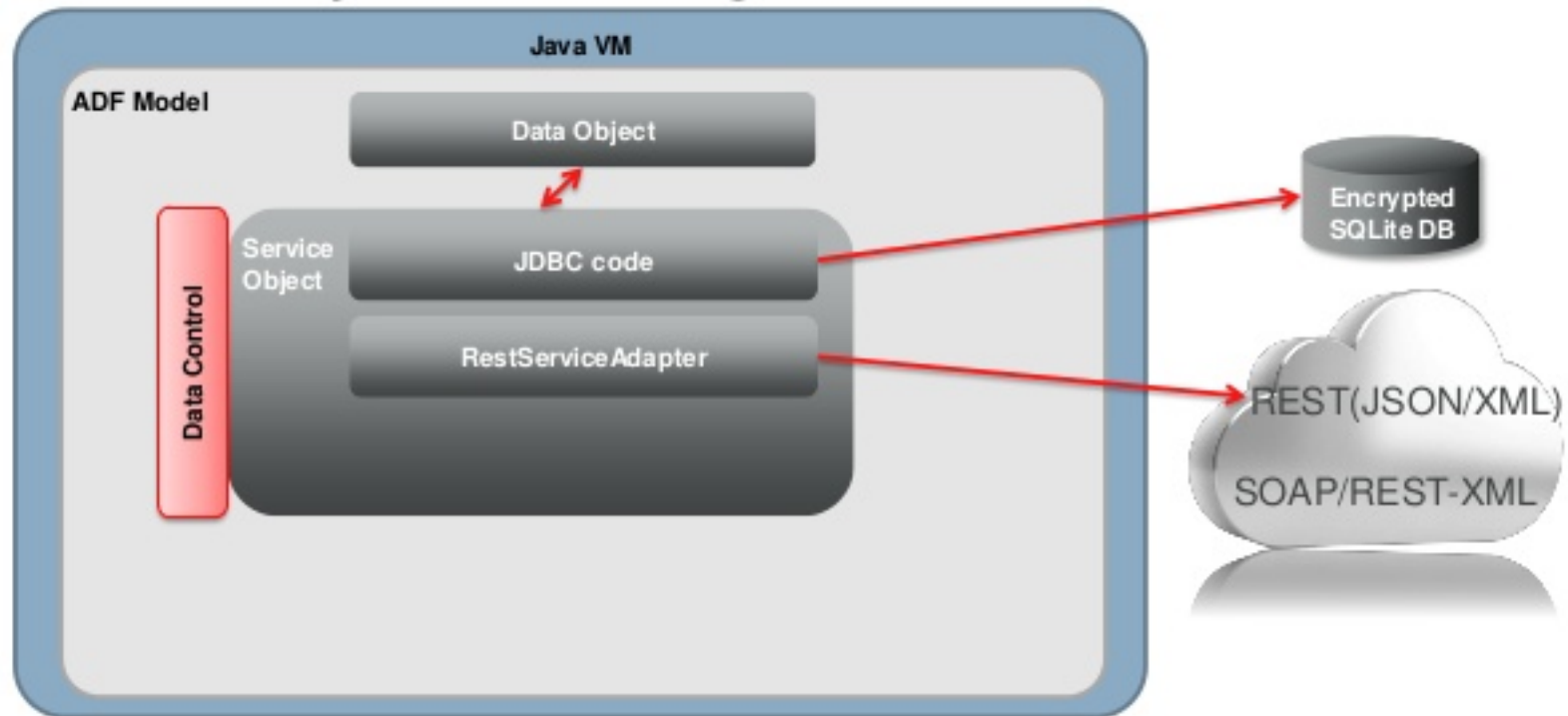
Modelo

- Objetos de Dados
 - Classes Java que mantêm os atributos de um objetos
 - Instâncias representam linhas de uma coleção
 - Podem fazer referências a outros objetos de dados em hierarquias complexas
- Objetos de serviços
 - Classes Java que oferecem CRUD
 - Retornam arrays de Objetos de Dados em seus métodos
 - São expostas ao controle de dados visual dos clientes



Modelo

Model Layer – Caching Data



ORACLE



Alternativas?



Persistência

- Arquivos
 - Solução tradicional
 - Oferecida pelo S.O.
 - Facilitad pelo Java
- BD Locais
 - Facilidade de acesso, diminuição da redundância
- DB Remotos
 - Solução para aplicações elaboradas
- Android
 - Provedores de conteúdo



Provedores de Conteúdo

- Objetos que armazenam dados de forma permanente
 - Pacote android.provider
- Como criar o seu próprio provedor?
 - Extendendo a classe ContentProvider
 - Também é possível utilizar os provedores existentes
- Localizados e acessados através de URIs
 - Cada dado que um provedor disponibiliza é acessado através de uma URI única



Provedores de Conteúdo

- Exemplos:
 - `android.provider.Contacts.Phones.CONTENT_URI`
 - `android.provider.Contacts.Photos.CONTENT_URI`
- Não são formas de armazenamento
 - Abstração para acesso aos dados
 - Podem utilizar, inclusive, o banco de dados SQLite



Usando um provedor

- Requisitos:
 - A URI que identifica o provedor
 - Os nomes dos campos a serem lidos ou escritos
 - Os tipos dos dados desses campos.
- Não são formas de armazenamento
 - Abstração para acesso aos dados
 - Podem utilizar, inclusive, o banco de dados SQLite



ContentProvider x SQLite

ContentProvider

```
private Cursor getEvents() {  
    return managedQuery(CONTENT_URI, FROM, null,  
        null, ORDER_BY);  
}
```

```
private void addEvent(String string) {  
    ContentValues values = new ContentValues();  
    values.put(TIME, System.currentTimeMillis());  
    values.put(TITLE, string);  
    getContentResolver().insert(CONTENT_URI, values);  
}
```

SQLiteDatabase

```
private Cursor getEvents() {  
    SQLiteDatabase db = events.getReadableDatabase();  
    Cursor cursor = db  
        .query(TABLE_NAME, FROM, null, null, null, null,  
            ORDER_BY);  
    startManagingCursor(cursor);  
    return cursor;  
}
```

```
private void addEvent(String string) {  
    SQLiteDatabase db = events.getWritableDatabase();  
    ContentValues values = new ContentValues();  
    values.put(TIME, System.currentTimeMillis());  
    values.put(TITLE, string);  
    db.insertOrThrow(TABLE_NAME, null, values);  
}
```



Criando um provedor

- Requisitos:
 - Declarar o provedor de conteúdo no arquivo manifesto
 - Especificar a forma de armazenamento (SQLite)
 - Estender a classe ContentProvider



Manifest

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="com.tyczj.bowling"
android:versionCode="1"
android:versionName="1.0" >

<uses-sdk android:minSdkVersion="11" />

<application
    android:icon="@drawable/ic_launcher"
    android:label="@string/app_name" >

    <activity android:name="Tabs">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>

    <activity android:name="BowlersList"></activity>
    <provider android:name="com.tyczj.bowling.BowlersDB"
        android:authorities="com.tyczj.bowling.bowlersdb">
    </provider>
</application>
```



ContentProvider

- Classe abstrata, com seis métodos abstratos
 - query()
 - insert()
 - update()
 - delete()
 - getType()
 - onCreate()
- São essas classes que abstraem o acesso aos dados



Conclusões

- Importância do SQLite
 - Banco de Dados Local ou distribuído
- Opção de arquivos
- Estruturas de acesso
 - DAO
 - Content Provider
- Content Provider é uma abstração da plataforma
 - Padronização do compartilhamento das informações
 - Content Provider da agenda, do SMS, de aplicações de terceiros etc.



Apresentações



11/06

- Aplicativo para Manutenção Automotiva
 - Daniela e Marcus Vinícius
- Agenda Acadêmica
 - Bruno Telles e Matheus Marinho
- Next Livros
 - João Paulo e Paulo Henrique
- Convite de Eventos
 - Rennan Toniolo e Sandro Coelho
- Agenda de Imóveis
 - Daniel Oliveira e Carlos Roberto



18/06

- Não Mora
 - Pedro Ivo, Victor Machado e Yuri
- BarWhere
 - Frederico Alves e Luidgi Sarto
- Transvita
 - Hygor, Leojayme e Marcelo Machado
- Status Baseado em Localização
 - Amanda e Paulo Vitor
- Tweetmapas
 - José Eduardo, Luiz Carlos e Rondinelle
- Qual é a Boa
 - Célio, Hugo, Thiago Caldeira e Thiago Mello

