



# ENEPEX

ENCONTRO DE ENSINO,  
PESQUISA E EXTENSÃO

8° ENEPE UFGD • 5° EPEX UEMS

## GERENCIAMENTO DE DADOS EM DISPOSITIVOS MÓVEIS

Guilherme Henrique Vieira Pereira<sup>1</sup>; Nilton César de Paula<sup>2</sup>

<sup>1</sup> Acadêmico de Ciência da Computação e bolsista de Iniciação Científica, e-mail: guilherme.hvp@gmail.com

<sup>2</sup> Professor e orientador do Curso de Ciência da Computação, e-mail: nilton@comp.uems.br

Endereço institucional: Universidade Estadual de Mato Grosso do Sul – UEMS

### RESUMO

Gerenciar dados em dispositivos móveis é basicamente organizar dados de uma pessoa no dia a dia, como por exemplo, os dados de um aluno de uma universidade, com auxílio de um smartphone, iphone, entre outros dispositivos móveis. O objetivo deste trabalho foi desenvolver uma aplicação que permitisse gerenciar dados em dispositivos móveis, ou seja, gerar dados e manipular os mesmos. Foram estudados, primeiramente, os tipos de gerenciadores de dados para dispositivos móveis mais utilizados hoje em dia. Também quais linguagens de programação estão sendo utilizadas para desenvolver aplicativos móveis. Considerando os resultados alcançados e os fatores utilizados para determinar o que era necessário, recomenda-se a utilização do IDE Eclipse, da linguagem JAVA, do banco de dados SQLite e ferramentas para o desenvolvimento Android. Conseqüentemente a utilização do sistema Android, por ser uma ferramenta gratuita e com facilidade de material para estudos.

**Palavras-chave:** Dispositivos móveis, SQLite, JAVA.

### INTRODUÇÃO

Os dados são recursos decisivos para determinar o trabalho de um gerenciador de banco de dados. É necessário esta tecnologia de gerenciamento de banco de dados para atender as necessidades de informação do usuário. E essas informações, valores de campos em uma base de dados, são geradas com frequência por estes mesmos usuários, a fim de

realizar consultas posteriormente, seja para consultar nomes de funcionários que trabalham em um determinado setor, ou para buscar os associados de um clube em um região específica.

Desenvolver banco de dados pequenos e pessoais é relativamente fácil. Entretanto, para desenvolver um banco de dados mais complexo, como para um empresa grande, é necessário realizar um projeto, onde terão analistas e administradores de banco de dados trabalhando em conjunto, melhorando assim a integridade e segurança dos dados. O desenvolvimento de uma aplicação acaba sendo mais simples do que a de uma base de dados, pois o mesmo realizará funções de busca, inserção e edição em uma base de dados já definida, ou seja, a complexidade de uma base de dados é maior por trabalhar diretamente com a organização dos mesmos. O gerenciamento de dados conta com o auxílio de um sistema de gerenciamento de banco de dados (SGBDs), ou seja, um conjunto de programas que permitem definir, editar e construir um base de dados.

O objetivo neste trabalho, utilizando ferramentas existentes, é o de gerar os dados para serem manipulados através de um dispositivo móvel por meio de uma aplicação simples. Em outras palavras, uma aplicação servirá de intermédio entre a base de dados e a manipulação desses dados, ou seja, ao inserir dados a partir desta aplicação, eles poderão ser consultados, editados ou mesmo apagados, o que de fato define o gerenciamento de dados em dispositivos móveis.

## **MATERIAIS E MÉTODOS**

A pesquisa foi realizada utilizando máquinas pessoais por um período e posteriormente os recursos do Laboratório de Sistemas Móveis (LSM). Esse laboratório implantando na UEMS foi criado com o apoio da Fundect e possui recursos computacionais como computadores, smartphones e tablets. Os equipamentos puderam satisfazer o desenvolvimento desta iniciação, com desempenho satisfatório, sendo assim, é possível realizar o desenvolvimento de aplicações e posteriormente fazer os devidos testes nos equipamentos portáteis disponíveis.

Em um período do projeto foram realizados apenas testes virtuais de aplicação, pois o acesso a aparelhos portáteis não estavam acessíveis. Com a implantação do LSM foi possível realizar os testes em aparelhos como o Motorola Moto G, desde a implementação até o teste final. Esse dispositivo possui o sistema Android KitKat 4.4, isso significa que possui a mais atualizada versão dos sistemas disponíveis para smartphones e o ADT (Android Development Tools) (Developer Android, 2014) que possui todas as ferramentas necessárias para o

desenvolvimento de aplicações hoje em dia. O Sistema Operacional padrão para o desenvolvimento de aplicações móveis é o Windows, porém as máquinas do LSM contam com os sistemas Windows 7 e Linux Mint, ou seja, se necessária a utilização de um ou de outro é possível ter acesso a ambas no laboratório.

Para os estudos foram feitas pesquisas em materiais na internet(Devmedia, 2014; StackOverflow, 2014; XDA Developers, 2014).

## RESULTADOS E DISCUSSÃO

Como resultado dos estudos tem-se o que foi necessário para o desenvolvimento da aplicação que gerencia um banco de dados para um dispositivo móvel e a explicação da aplicação móvel desenvolvida.

Primeiramente foi necessário realizar o download do ADT (Android Development Tools) no site developer.android.com que vem o IDE Eclipse e o Android SDK manager. O próximo passo para desenvolver uma aplicação móvel foi configurar os pacotes necessários usando o Android SDK manager. A versão utilizada no projeto foi a Android 4.4 Kitkat.

Com o ambiente pronto o próximo passo é desenvolver a aplicação móvel. A aplicação, chamada de newDados, acessa um banco de dados SQLite (SQLite, 2014) e possui as seguintes funcionalidades/classes: “BuscarPessoa”, “CadastroPessoa”, “EditarPessoa”, “Pessoa”, “PessoaListAdapter”, “RepositorioPessoa”, “RepositorioPessoaScript” e “SQLiteHelper”. A seguir as principais funcionalidades serão descritas.

O acesso ao banco de dados SQLite é realizado com o uso da extensão da classe "SQLiteOpenHelper". Na extensão é desenvolvido o código para criar (método scriptSQLCreate) e apagar (método scriptSQLDelete) uma tabela do banco de dados. A Figura 1 mostra a criação e exclusão e todo o acesso é via padrão SQL.

```
class SQLiteHelper extends SQLiteOpenHelper {  
  
    private static final String CATEGORIA = "livro";  
  
    private String[] scriptSQLCreate;  
    private String scriptSQLDelete;  
  
    SQLiteHelper(Context context, String nomeBanco, int versaoBanco, String[] scriptSQLCreate, String scriptSQLDelete) {  
        super(context, nomeBanco, null, versaoBanco);  
        this.scriptSQLCreate = scriptSQLCreate;  
        this.scriptSQLDelete = scriptSQLDelete;  
    }  
}
```

Figura 1. Criando e apagando um banco de dados SQLite.

Após criar o banco de dados pode-se inserir, atualizar e pesquisar informações no banco de dados. Essas funções são realizadas usando a classe “RepositorioPessoa” (Figura 2), que contém a estrutura do banco de dados e a implementação dos métodos de inserção, atualização e busca de dados.

```
public class RepositorioPessoa {  
  
    private static final String CATEGORIA = "dados";  
    private static final String NOME_BANCO = "dados_android";  
    public static final String NOME_TABELA = "pessoa";  
  
    protected SQLiteDatabase db;  
  
    public RepositorioPessoa(Context ctx) {  
        db = ctx.openOrCreateDatabase(NOME_BANCO, Context.MODE_PRIVATE, null);  
    }  
  
    protected RepositorioPessoa() {  
    }  
  
    public long salvar(Pessoa pessoa) {  
        long id = pessoa.id;  
  
        if (id != 0) {  
            atualizar(pessoa);  
        } else {  
            id = inserir(pessoa);  
        }  
    }  
}
```

**Figura 2.** Classe para incluir e alterar dados no SQLite.

O código para incluir e alterar dados no banco de dados SQLite está ilustrado na Figura 3. Os dados manipulados são o nome, cpf e idade de uma pessoa e o acesso no banco de dados é através dos métodos insert e update do SQL.

```
public long inserir(Pessoa pessoa) {  
    ContentValues values = new ContentValues();  
    values.put(Pessoas.NOME, pessoa.nome);  
    values.put(Pessoas.CPF, pessoa.cpf);  
    values.put(Pessoas.IDADE, pessoa.idade);  
  
    long id = inserir(values);  
    return id;  
}  
  
public long inserir(ContentValues valores) {  
    long id = db.insert(NOME_TABELA, "", valores);  
    return id;  
}  
  
public int atualizar(Pessoa pessoa) {  
    ContentValues values = new ContentValues();  
    values.put(Pessoas.NOME, pessoa.nome);  
    values.put(Pessoas.CPF, pessoa.cpf);  
    values.put(Pessoas.IDADE, pessoa.idade);  
  
    String _id = String.valueOf(pessoa.id);  
  
    String where = Pessoas._ID + "=?";  
    String[] whereArgs = new String[] { _id };  
  
    int count = atualizar(values, where, whereArgs);  
    return count;  
}  
  
public int atualizar(ContentValues valores, String where, String[] whereArgs) {  
    int count = db.update(NOME_TABELA, valores, where, whereArgs);  
    Log.i(CATEGORIA, "Atualizou [" + count + "] registros");  
    return count;  
}
```

**Figura 3.** Métodos da classe "RepositorioPessoa" para incluir e alterar dados no SQLite.

“BuscarPessoa” (parte do código na Figura 4) realiza uma busca pelo nome da pessoa, inserido pelo usuário, usando a função buscarPessoa(nomePessoa). Caso o nome seja

encontrado, os dados da pessoa, como CPF e idade são armazenados para serem apresentados ao usuário, como resposta a busca.

```
@Override
protected void onPause() {
    super.onPause();
    setResult(RESULT_CANCELED);

    finish();
}

public void onClick(View view) {

    EditText nome = (EditText) findViewById(R.id.campoNome);
    EditText cpf = (EditText) findViewById(R.id.campoCpf);
    EditText idade = (EditText) findViewById(R.id.campoIdade);

    String nomePessoa = nome.getText().toString();

    Pessoa pessoa = buscarPessoa(nomePessoa);

    if (pessoa != null) {
        cpf.setText(pessoa.cpf);
        idade.setText(String.valueOf(pessoa.idade));
    } else {
        cpf.setText("");
        idade.setText("");

        Toast.makeText(BuscarPessoa.this, "Nenhuma pessoa encontrada", Toast.LENGTH_SHORT).show();
    }
}

protected Pessoa buscarPessoa(String nomePessoa) {
    Pessoa pessoa = CadastroPessoa.repositorio.buscarPessoaPorNome(nomePessoa);
    return pessoa;
}
```

Figura 4. "BuscarPessoa" para pesquisar dados de uma pessoa usando o nome da pessoa.

Por fim, a classe "CadastroPessoa" (parte do código na Figura 5), tem por finalidade permitir a inclusão dos dados de uma pessoa, mas antes verifica se já não existe a pessoa no cadastro, ou seja, não permitindo a duplicidade de dados no banco de dados.

```
public void onClick(View view) {

    EditText nome = (EditText) findViewById(R.id.campoNome);
    EditText cpf = (EditText) findViewById(R.id.campoCpf);
    EditText idade = (EditText) findViewById(R.id.campoIdade);

    String nomePessoa = nome.getText().toString();

    Pessoa pessoa = buscarPessoa(nomePessoa);

    if (pessoa != null) {
        cpf.setText(pessoa.cpf);
        idade.setText(String.valueOf(pessoa.idade));
    } else {
        cpf.setText("");
        idade.setText("");

        Toast.makeText(BuscarPessoa.this, "Nenhuma pessoa encontrada", Toast.LENGTH_SHORT).show();
    }
}

protected Pessoa buscarPessoa(String nomePessoa) {
    Pessoa pessoa = CadastroPessoa.repositorio.buscarPessoaPorNome(nomePessoa);
    return pessoa;
}
```

Figura 5. Classe "CadastroPessoa" para inserir novos dados no banco de dados SQLite.

## CONCLUSÃO

Com os estudos e pesquisas realizadas para o desenvolvimento do projeto, foi possível compreender as novas ferramentas de desenvolvimento de aplicações usando banco de dados para dispositivos móveis. Possibilitando assim uma experiência nova e adquirir conhecimento sobre ferramentas atuais. Esta experiência nos incentiva a querer continuar trabalhando na área, para assim aprender mais sobre o assunto.

Outros projetos relacionados podem ser realizados, pois o desenvolvimento de aplicações para dispositivos móveis é uma área de grande amplitude e que engloba várias áreas, pois a tecnologia está em todos os lugares e com ela está todo o desenvolvimento em aplicações úteis e que agilizam trabalhos que antigamente eram mais demorados. Tornando tudo mais simples e fácil de ser trabalhado.

## AGRADECIMENTOS

À PIBIC-UEMS, pela concessão da bolsa, à FUNDECT pelos recursos para a criação do Laboratório de Sistemas Móveis (LSM) da UEMS. Também ao orientador pela sua orientação na condução dos estudos.

## REFERÊNCIAS

**Developer Android** Disponível em [developer.android.com](http://developer.android.com), 2014.

**Devmedia, cursos** Disponível em [devmedia.com.br](http://devmedia.com.br), 2014.

**SQLite** Disponível em [SQLite.org](http://SQLite.org), 2014.

**StackOverflow, retirar dúvidas** Disponível em [pt.stackoverflow.com](http://pt.stackoverflow.com), 2014.

**XDA Developers, pesquisas e exemplos** Disponível em [www.xda-developers.com](http://www.xda-developers.com), 2014.