
TRABALHO

APP CRUD ANDROID

Roger da Palma

OBJETIVO DO TRABALHO

O OBJETIVO DESTA PROJETO É DESENVOLVER UM APLICATIVO ANDROID QUE:

- GERENCIE INFORMAÇÕES BÁSICAS DOS ESTUDANTES, COMO NOME, CPF, TELEFONE, IDADE, STATUS DE ATIVIDADE E TIPO DE CURSO.
- PERMITA O GERENCIAMENTO DE FATURAS ASSOCIADAS A CADA ESTUDANTE, INCLUINDO O REGISTRO DE NOVOS PAGAMENTOS E A VISUALIZAÇÃO DE FATURAS ABERTAS.
- FACILITE A NAVEGAÇÃO ENTRE AS TELAS DE LISTAGEM DE ESTUDANTES, DETALHES DO ESTUDANTE E ADIÇÃO DE PAGAMENTOS.

BIBLIOTECAS E FERRAMENTAS UTILIZADAS

- **ANDROID SDK:** FRAMEWORK PRINCIPAL PARA O DESENVOLVIMENTO DO APLICATIVO ANDROID.
- **SQLITE:** BANCO DE DADOS LOCAL UTILIZADO PARA ARMAZENAR INFORMAÇÕES DE ESTUDANTES E FATURAS.
- **ANDROID RECYCLERVIEW:** UTILIZADO PARA EXIBIR LISTAS DE DADOS DE FORMA EFICIENTE E COM CAPACIDADE DE ROLAGEM (COMO LISTVIEW E GRIDVIEW).
- **TEXTVIEWNAME:** EXIBINDO MENSAGENS DAS VALIDAÇÕES E ETC! (TEXTALERT).
- **ANDROID ROOM:** PODERIA SER USADO PARA SIMPLIFICAR OPERAÇÕES DE BANCO DE DADOS, MAS NESTE PROJETO UTILIZAMOS SQLITEOPENHELPER PARA MANIPULAR DIRETAMENTE O BANCO DE DADOS SQLITE.

ARQUITETURA DO PROJETO

O PROJETO SEGUE UMA ARQUITETURA MVC SIMPLIFICADA:

- **MODEL:** CLASSES QUE REPRESENTAM A ESTRUTURA DOS DADOS, COMO STUDENT E PAGAMENTO.
- **VIEW:** XML LAYOUTS QUE DEFINEM A INTERFACE DO USUÁRIO, COMO ACTIVITY_MAIN.XML, STUDENT_ITEM.XML, ETC.
- **CONTROLLER:** ATIVIDADES ANDROID (MAINACTIVITY, STUDENTDETAILSACTIVITY, ADDEDITSTUDENTACTIVITY, ADDPAYMENTACTIVITY) E ADAPTADORES (STUDENTADAPTER) QUE CONTROLAM A LÓGICA DA INTERFACE DO USUÁRIO E A INTERAÇÃO COM O BANCO DE DADOS.

CLASSES DE ATIVIDADE (ACTIVITIES)

- **MAINACTIVITY.JAVA:** A ATIVIDADE PRINCIPAL QUE CARREGA E EXIBE A LISTA DE ESTUDANTES USANDO O RECYCLERVIEW E STUDENTADAPTER.
- **STUDENTDETAILSACTIVITY.JAVA:** EXIBE OS DETALHES DE UM ESTUDANTE ESPECÍFICO E TODAS AS SUAS FATURAS. PERMITE A ADIÇÃO DE NOVAS FATURAS.
- **ADDEDITSTUDENTACTIVITY.JAVA:** PERMITE ADICIONAR UM NOVO ESTUDANTE OU EDITAR UM EXISTENTE.
- **ADDPAYMENTACTIVITY.JAVA:** PERMITE ADICIONAR UMA NOVA FATURA PARA UM ESTUDANTE.

ADAPTADORES (ADAPTERS)

- **STUDENTADAPTER.JAVA:** ADAPTA OS DADOS DOS ESTUDANTES PARA SEREM EXIBIDOS EM UM RECYCLERVIEW. INCLUI LÓGICA PARA CONTAR E EXIBIR O NÚMERO DE FATURAS ABERTAS DE CADA ESTUDANTE.

BANCO DE DADOS (DATABASE)

- **STUDENTDATABASEHELPER.JAVA:** GERENCIA A CRIAÇÃO, ATUALIZAÇÃO E MANIPULAÇÃO DE DADOS NO BANCO DE DADOS SQLITE. CONTÉM MÉTODOS PARA ADICIONAR, EDITAR, EXCLUIR E RECUPERAR ESTUDANTES E FATURAS.

MODELOS DE DADOS (DATA MODELS)

- **STUDENT.JAVA:** MODELO DE DADOS PARA ARMAZENAR INFORMAÇÕES DO ESTUDANTE.
- **PAGAMENTO.JAVA:** MODELO DE DADOS PARA ARMAZENAR INFORMAÇÕES DE PAGAMENTO (FATURAS).

LAYOUTS XML

- **ACTIVITY_MAIN.XML:** LAYOUT PARA A TELA PRINCIPAL QUE EXIBE A LISTA DE ESTUDANTES.
- **ACTIVITY_STUDENT_DETAILS.XML:** LAYOUT PARA A TELA DE DETALHES DO ESTUDANTE.
- **ACTIVITY_ADD_EDIT_STUDENT.XML:** LAYOUT PARA ADICIONAR OU EDITAR INFORMAÇÕES DO ESTUDANTE.
- **ACTIVITY_ADD_PAYMENT.XML:** LAYOUT PARA ADICIONAR UMA NOVA FATURA.
- **STUDENT_ITEM.XML:** LAYOUT PARA CADA ITEM DA LISTA DE ESTUDANTES NO RECYCLERVIEW.

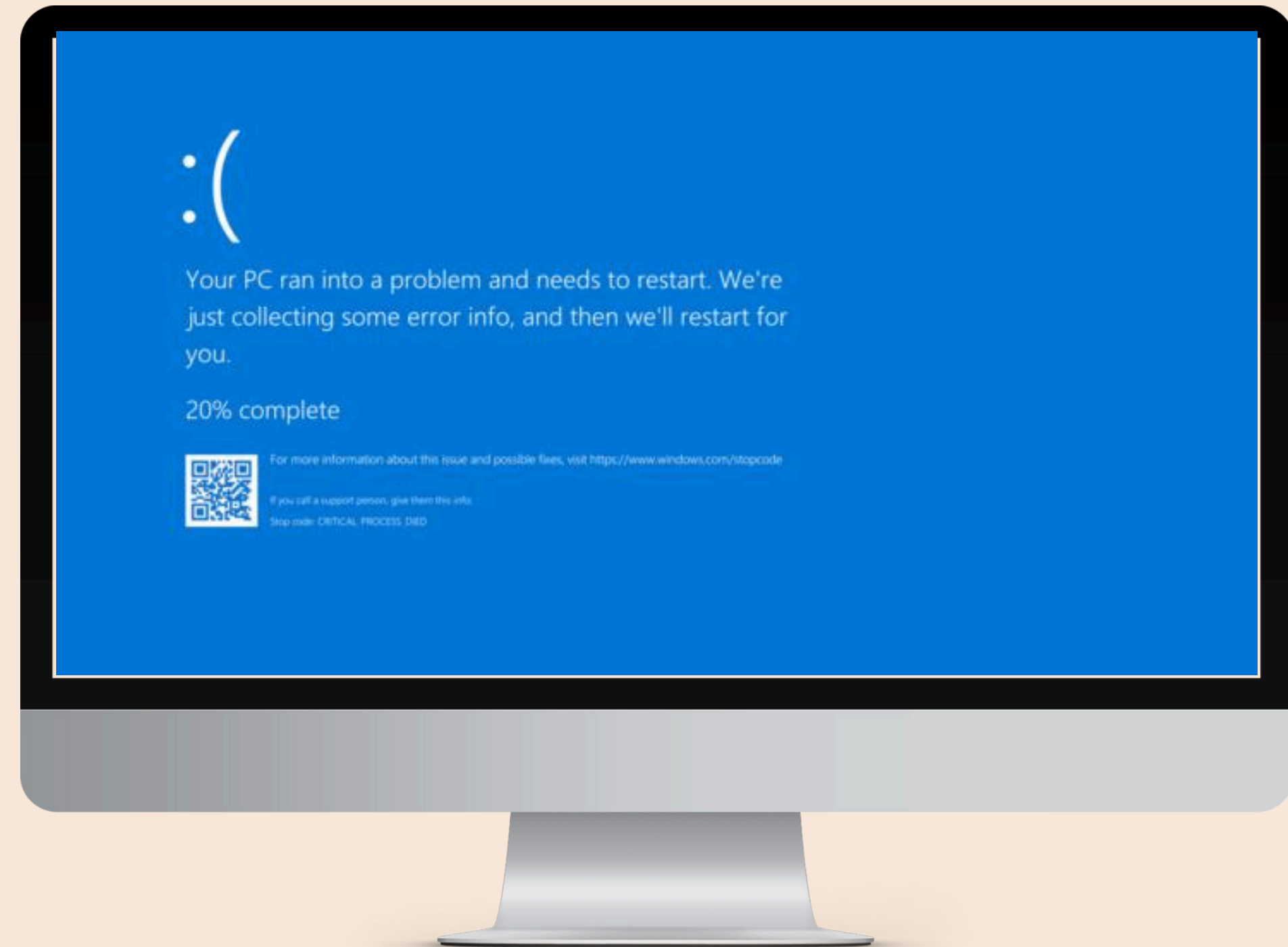
CODIGO

```
© StudentDetailsActivity.java </> activity_add_edit_student.xml </> activity_add_payment.xml © MainActivity.java x </> activity_main.xml
1 package com.example.myapplication;
2
3 > import ...
19
20 <> public class MainActivity extends AppCompatActivity {
21     private static final int REQUEST_CODE_ADD_STUDENT = 1; 1 usage
22     private static final int REQUEST_CODE_EDIT_STUDENT = 2; 1 usage
23
24     private RecyclerView recyclerView; 3 usages
25     private StudentAdapter studentAdapter; 6 usages
26     private List<Student> studentList; 4 usages
27     private FloatingActionButton fabAdd, fabDelete, fabEdit; 2 usages
28     private StudentDatabaseHelper studentDatabaseHelper; 3 usages
29
30     @Override
31     protected void onCreate(Bundle savedInstanceState) {
32         super.onCreate(savedInstanceState);
33         setContentView(R.layout.activity_main);
34
35         // Inicializando o helper do banco de dados
36         studentDatabaseHelper = new StudentDatabaseHelper(context);
37
38         // Inicializando as views
39         recyclerView = findViewById(R.id.recycler_view);
40         fabAdd = findViewById(R.id.fab);
41         fabDelete = findViewById(R.id.fab_delete);
42         fabEdit = findViewById(R.id.fab_refresh);
43
44         // Configurando o RecyclerView
45         recyclerView.setLayoutManager(new LinearLayoutManager(context));
46         studentList = loadStudentsFromDatabase(); // Carrega a lista de estudantes do banco de dados
47         studentAdapter = new StudentAdapter(context, studentList); // Inicializa o adapter com a lista de estudantes
48         recyclerView.setAdapter(studentAdapter); // Associa o adapter ao RecyclerView
49
50         // Listener para o botão de adicionar estudante
51         fabAdd.setOnClickListener(v -> {
52             Intent intent = new Intent(packageContext: MainActivity.this, AddEditStudentActivity.class);
53             startActivityForResult(intent, REQUEST_CODE_ADD_STUDENT);
54         });
55
56         // Listener para o botão de deletar estudante(s)
57         fabDelete.setOnClickListener(v -> {
```

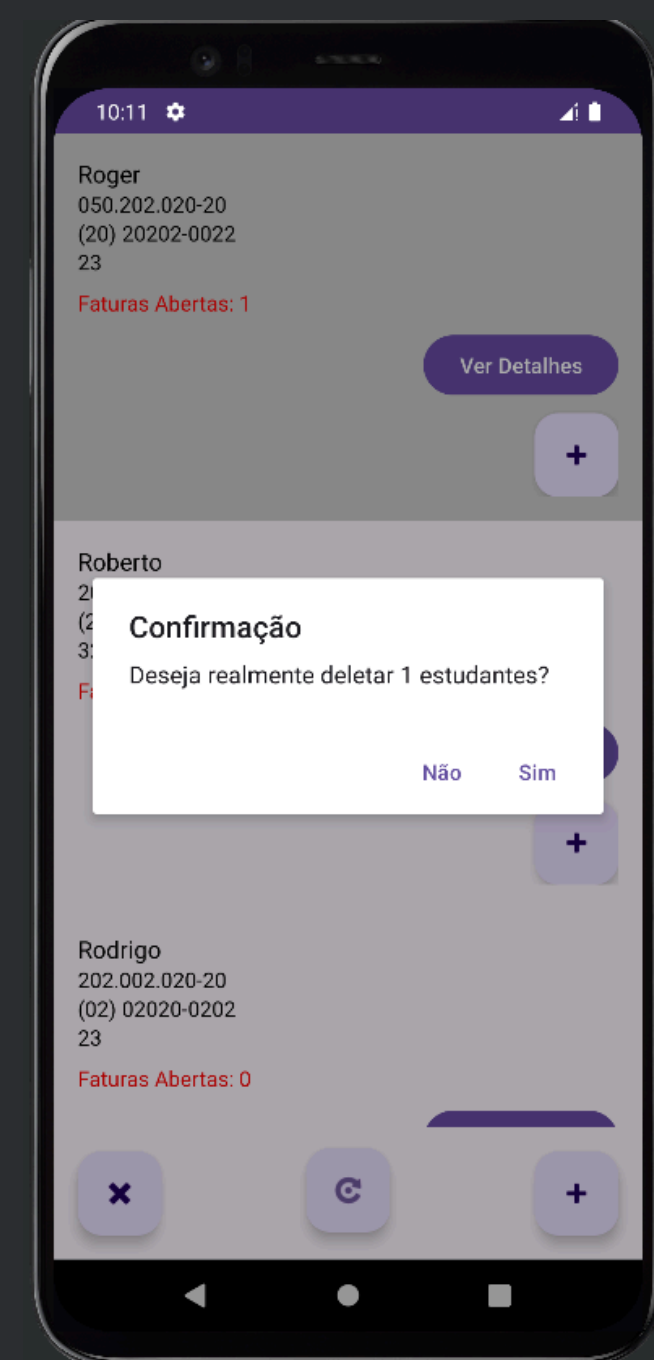
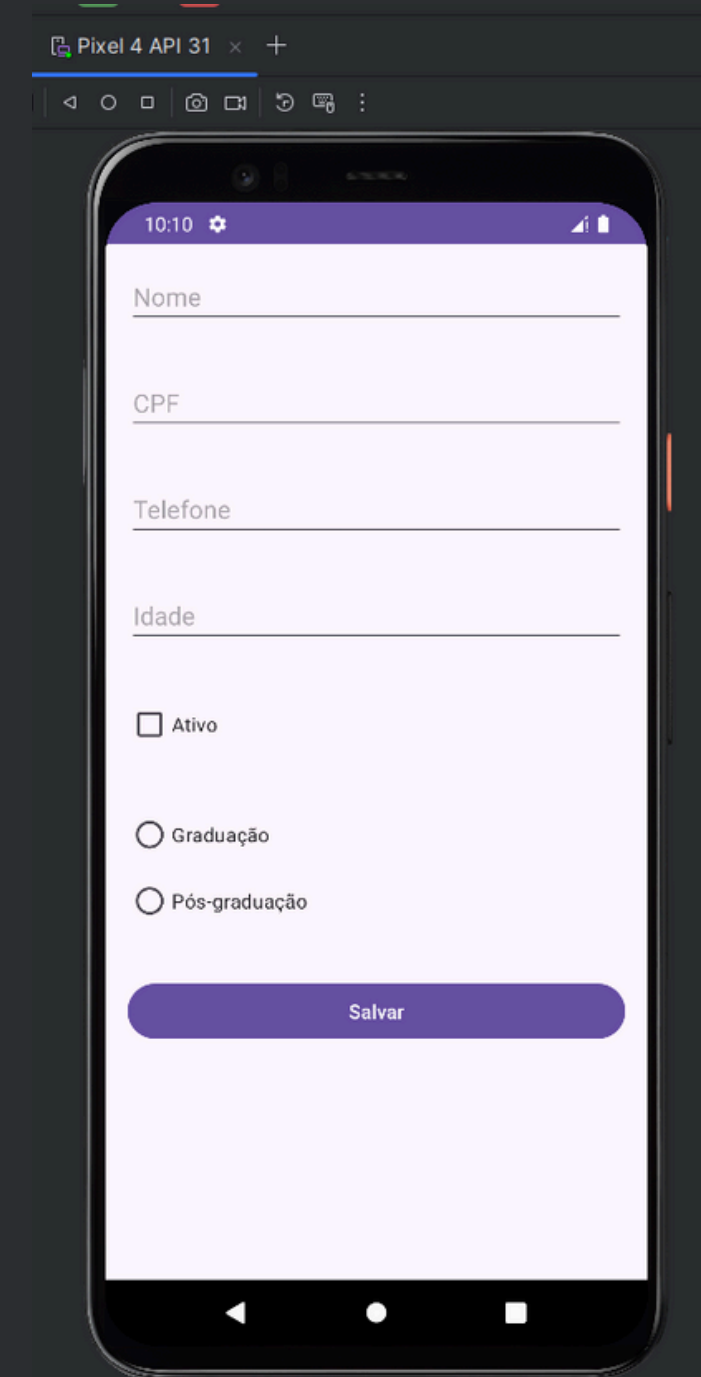
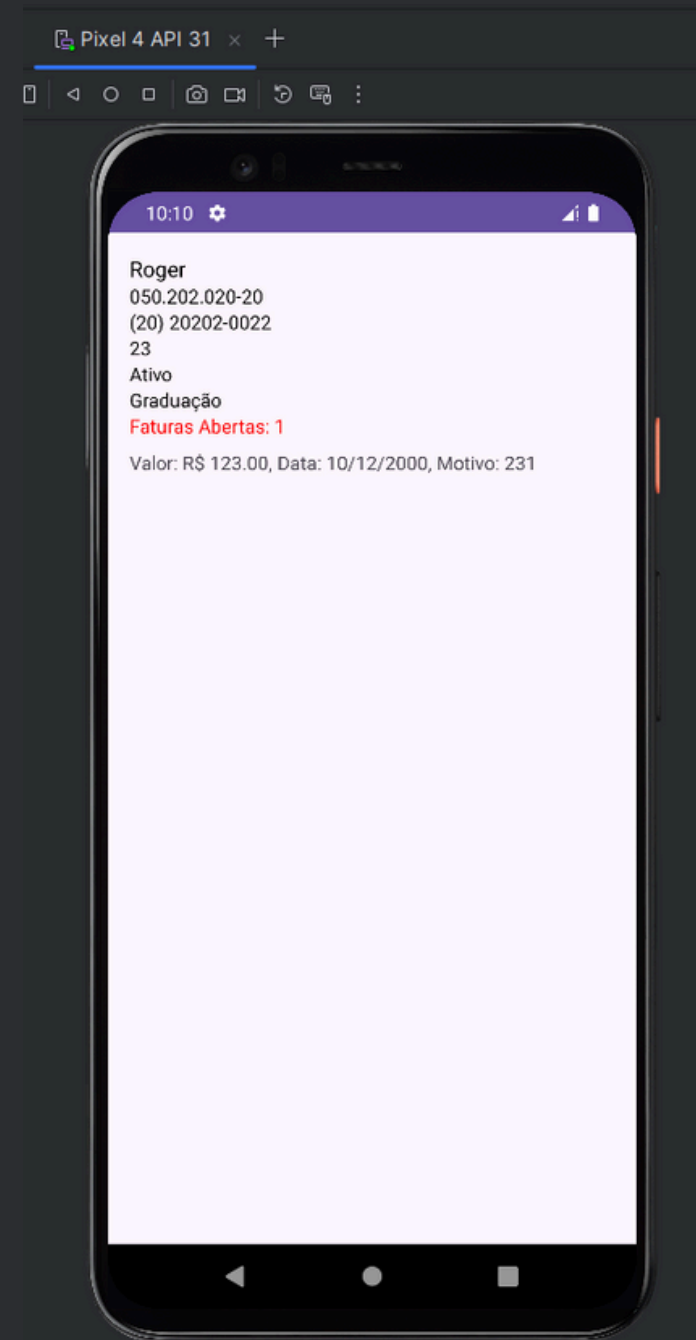
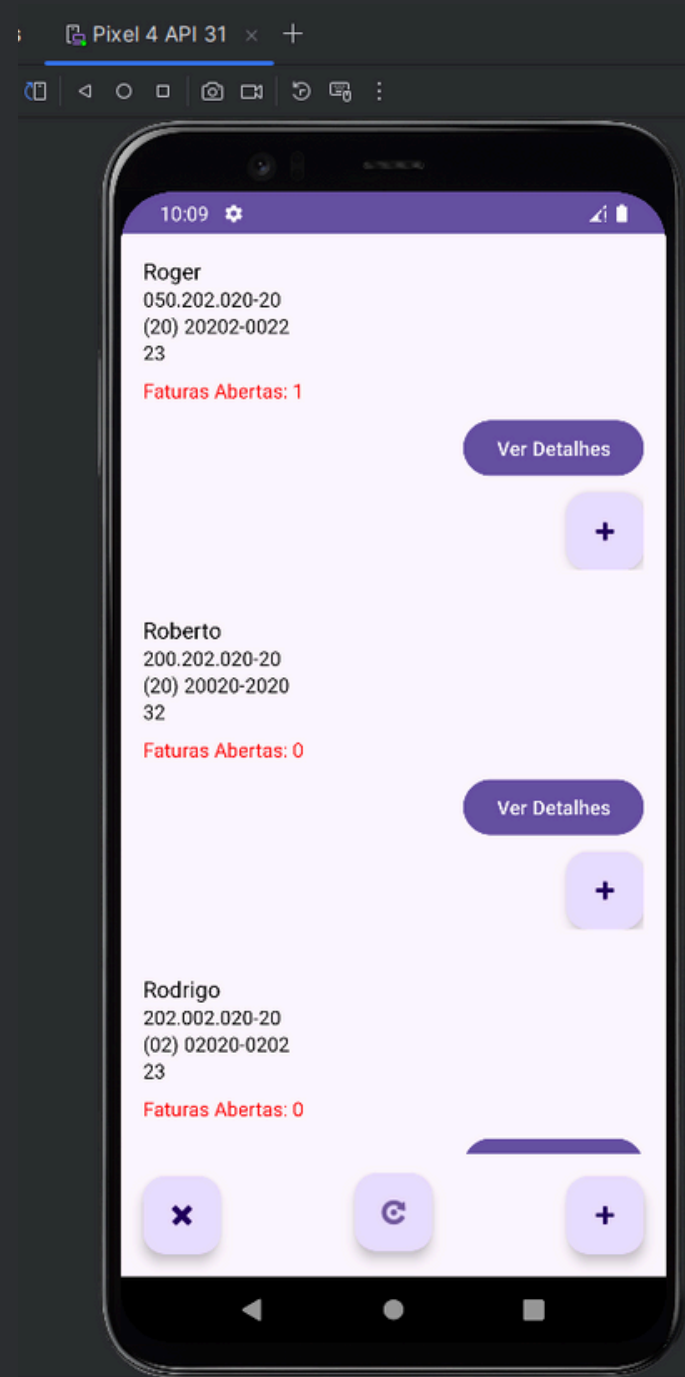
```
</> activity_add_edit_student.xml </> activity_add_payment.xml © MainActivity.java © StudentDatabaseHelper.java x </> activity_main.xml
1 package com.example.myapplication;
2
3 > import ...
12
13 public class StudentDatabaseHelper extends SQLiteOpenHelper { 10 usages
14     private static final String DATABASE_NAME = "studentss.db"; // Nome do banco de dados 1 usage
15     private static final int DATABASE_VERSION = 4; // Versão do banco de dados (incrementada para mudanças) 1 usage
16
17     // Nome da tabela e colunas para alunos
18     private static final String TABLE_STUDENTS = "students"; 7 usages
19     private static final String COLUMN_ID = "id"; 7 usages
20     private static final String COLUMN_NAME = "name"; 5 usages
21     private static final String COLUMN_CPF = "cpf"; 5 usages
22     private static final String COLUMN_PHONE = "phone"; 5 usages
23     private static final String COLUMN_AGE = "age"; 5 usages
24     private static final String COLUMN_ACTIVE = "active"; 5 usages
25     private static final String COLUMN_COURSE_TYPE = "course_type"; 5 usages
26
27     // Nome da tabela e colunas para pagamentos
28     private static final String TABLE_PAGAMENTOS = "pagamentos"; 7 usages
29     private static final String COLUMN_PAGAMENTO_ID = "id"; 4 usages
30     private static final String COLUMN_PAGAMENTO_ALUNO_ID = "aluno_id"; 7 usages
31     private static final String COLUMN_PAGAMENTO_VALOR = "valor"; 4 usages
32     private static final String COLUMN_PAGAMENTO_DATA = "data"; 4 usages
33     private static final String COLUMN_PAGAMENTO_DESCRICAO = "descricao"; 5 usages
34
35     public StudentDatabaseHelper(Context context) { 5 usages
36         super(context, DATABASE_NAME, factory: null, DATABASE_VERSION);
37     }
38
39     @Override
40     public void onCreate(SQLiteDatabase db) {
41         // Criação da tabela de alunos
42         String createStudentsTable = "CREATE TABLE " + TABLE_STUDENTS + " (" +
43             COLUMN_ID + " INTEGER PRIMARY KEY AUTOINCREMENT, " +
44             COLUMN_NAME + " TEXT, " +
45             COLUMN_CPF + " TEXT, " +
46             COLUMN_PHONE + " TEXT, " +
47             COLUMN_AGE + " INTEGER, " +
48             COLUMN_ACTIVE + " INTEGER DEFAULT 1, " +
49             COLUMN_COURSE_TYPE + " TEXT)";
```

DIFICULDADES

- IDENTIFICAR ERROS NO CÓDIGO: MUITAS DAS VEZES OS ERROS NÃO EXIBIAM CRASHES, SENDO NECESSÁRIO CONSULTAR OS LOGS PARA RASTREAR O ERRO COMETIDO.
- EXIBIÇÃO DE FATURAS REGISTRADAS (UTILIZAR DUAS TABELAS EM UMA SÓ): FOI CUSTOSO, POIS MUITAS VEZES NÃO ERAM EXIBIDAS NO LOCAL CERTO OU NEM ERAM EXIBIDAS.
- VERSÃO DO BANCO: ÀS VEZES ERA ESQUECIDO DE TROCAR A VERSÃO DO BANCO, CAUSANDO UM CRASH SEM "MOTIVO" APARENTE.



RESULTADOS



CONSIDERAÇÕES FINAIS

- FALTA DE BOAS PRATICAS LEVAM A ERROS BASICOS
- ERROS SEMPRE SAO EXIBIDOS EM LOGS, TESTAR O SISTEMA INCLUIR LOGS NO CODIGO PARA MELHOR ANALISE
- VEREFICAR VERSÃO DO BANCO, PODENDO A LEVAR UM ERRO SIMPLES POR ESTAR EM UMA VERSAO ERRADA

OBRIGADO!!!

REDES SOCIAIS :



Roger da Palma

• Desenvolvedor

Github: rogerdapalma

Email: rogerdapalma@gmail.com

TRABALHO

APP CRUD ANDROID

Roger da Palma
