

Inserindo UUID para alunos novos

Por fim, precisamos apenas editar o modo como estamos inserindo o id dos alunos, atualmente temos o seguinte código:

```
public void insere(Aluno aluno) {
    SQLiteOpenHelper db = getWritableDatabase();

    ContentValues dados = pegaDadosDoAluno(aluno);

    db.insert("Alunos", null, dados);
}
```

Observe que atualmente estamos enviando os dados que vem do método `pegaDadosDoAluno`. Se dermos uma olhada neste método temos o seguinte código:

```
@NonNull
private ContentValues pegaDadosDoAluno(Aluno aluno) {
    ContentValues dados = new ContentValues();
    dados.put("nome", aluno.getNome());
    dados.put("endereco", aluno.getEndereco());
    dados.put("telefone", aluno.getTelefone());
    dados.put("site", aluno.getSite());
    dados.put("nota", aluno.getNota());
    dados.put("caminhoFoto", aluno.getCaminhoFoto());
    return dados;
}
```

Veja que em nenhum momento estamos inserindo o id! Em outras palavras o SQLite gerava automaticamente quando fazia a chamada do `insert()`. Mas como ele fazia isso? Anteriormente o tipo de dado da nossa tabela de alunos era `INTEGER`, ou seja, ele já compreendia que os ids seriam incrementais, portanto, ele já gerava automaticamente! Agora temos como tipo de dados `CHAR(36)`, ou seja ele não tem nenhum padrão para gerar esses ids, portanto, precisamos inserir essa informação a partir de agora! Para isso basta apenas adicionar mais um `put()` dentro do método `pegaDadosDoAluno()` que vai ter como chave "id" e valor `aluno.getId()`.

Gerando UUID

Agora que o nosso método que cria o `ContentValues` está adicionando o campo `id` precisamos apenas gerar um UUID para o aluno antes de chamar esse método, portanto, adicione o código `aluno.setId(geraUUID())` antes de chamar o método `pegaDadosDoAluno()`.

Testando App

Agora que estamos gerando os UUIDs para alunos novos, basta apenas inserir um aluno novo e verificar no log, quando carregar a lista de alunos, se está aparecendo o UUID.

