

```
1 package com.example.agendario;
2 // seq-Junho-001
3 import android.app.Application;
4 import android.os.AsyncTask;
5
6 import androidx.lifecycle.LiveData;
7
8 import java.util.List;
9
10 public class QuestionRepository1 {
11
12     private QuestionDao mQuestionDao;
13     private DbRoomDatabase mDbRoomDatabase;
14     private LiveData<List<Question>> mAllQuestions;
15
16     LiveData<List<Question>> getAllQuestions(){
17         return mAllQuestions;
18     }
19
20     public QuestionRepository1(Application application){
21         mDbRoomDatabase = DbRoomDatabase.getDatabase(application);
22         mQuestionDao = mDbRoomDatabase.mQuestionDao();
23         mAllQuestions = mQuestionDao.getAllQuestions();
24     }
25
26     private static class InsertAsyncTask extends AsyncTask<Question, Void, Void>{
27         private QuestionDao mAsyncTaskDao;
28
29         InsertAsyncTask(QuestionDao dao){
30             mAsyncTaskDao=dao;
31         }
32
33         @Override
34         protected Void doInBackground(final Question... params){
35             mAsyncTaskDao.insertQuestion(params[0]);
36             return null;
37         }
38     }
39
40
41     private static class UpdateAsyncTask extends AsyncTask<Question, Void, Void>{
42         private QuestionDao mAsyncTaskDao;
43
44         UpdateAsyncTask(QuestionDao dao){
45             mAsyncTaskDao=dao;
46         }
47
48         @Override
49         protected Void doInBackground(final Question... params){
50             mAsyncTaskDao.updateQuestion(params[0]);
51             return null;
52         }
53     }
54
55
56     private static class DeleteAsyncTask extends AsyncTask<Question, Void, Void>{
57         private QuestionDao mAsyncTaskDao;
58
59         DeleteAsyncTask(QuestionDao dao){
60             mAsyncTaskDao=dao;
61         }
62
63         @Override
64         protected Void doInBackground(final Question... params){
65             mAsyncTaskDao.deleteQuestion(params[0]);
66             return null;
67         }
68     }
69 }
```

```
67     }
68
69 }
70
71 private static class DeleteAllAsyncTask extends AsyncTask<Void, Void, Void>{
72     private QuestionDao mAsyncTaskDao;
73
74     DeleteAllAsyncTask(QuestionDao dao){
75         mAsyncTaskDao=dao;
76     }
77
78     @Override
79     protected Void doInBackground(Void... voids){
80         mAsyncTaskDao.deleteAllQuestions();
81         return null;
82     }
83
84 }
85
86
87 public void insert(Question question){
88     new InsertAsyncTask(mQuestionDao).execute(question);
89 }
90
91 public void update(Question question){
92     new UpdateAsyncTask(mQuestionDao).execute(question);
93 }
94
95 public void delete(Question question){
96     new DeleteAsyncTask(mQuestionDao).execute(question);
97 }
98
99 public void deleteAll(){
100     new DeleteAllAsyncTask(mQuestionDao).execute();
101 }
102
103 }
104
105
106
107
108
109
110
111
112
113
114
115
```