```java
package com.example.agendiario;
// seq-Junho-001
import android.app.Application;
import android.os.AsyncTask;

import androidx.lifecycle.LiveData;

import java.util.List;

public class QuestionRepository1 {

    private QuestionDao mQuestionDao;
    private DbRoomDatabase mDbRoomDatabase;
    private LiveData<List<Question>> mAllQuestions;

    LiveData<List<Question>> getAllQuestions(){
        return mAllQuestions;
    }

    public QuestionRepository1(Application application){
        mDbRoomDatabase = DbRoomDatabase.getDatabase(application);
        mQuestionDao = mDbRoomDatabase.mQuestionDao();
        mAllQuestions = mQuestionDao.getAllQuestions();
    }

    private static class InsertAsyncTask extends AsyncTask<Question, Void, Void>{
        private QuestionDao mAsyncTaskDao;

        InsertAsyncTask(QuestionDao dao){
            mAsyncTaskDao=dao;
        }

        @Override
        protected Void doInBackground(final Question... params){
            mAsyncTaskDao.insertQuestion(params[0]);
            return null;
        }

    }

    private static class UpdateAsyncTask extends AsyncTask<Question, Void, Void>{
        private QuestionDao mAsyncTaskDao;

        UpdateAsyncTask(QuestionDao dao){
            mAsyncTaskDao=dao;
        }

        @Override
        protected Void doInBackground(final Question... params){
            mAsyncTaskDao.updateQuestion(params[0]);
            return null;
        }

    }

    private static class DeleteAsyncTask extends AsyncTask<Question, Void, Void>{
        private QuestionDao mAsyncTaskDao;

        DeleteAsyncTask(QuestionDao dao){
            mAsyncTaskDao=dao;
        }

        @Override
        protected Void doInBackground(final Question... params){
            mAsyncTaskDao.deleteQuestion(params[0]);
            return null;
        }

    }

    private static class DeleteAllAsyncTask extends AsyncTask<Void, Void, Void>{
        private QuestionDao mAsyncTaskDao;

        DeleteAllAsyncTask(QuestionDao dao){
            mAsyncTaskDao=dao;
        }

        @Override
        protected Void doInBackground(Void...  voids){
            mAsyncTaskDao.deleteAllQuestions();
            return null;
        }

    }


    public void insert(Question question){
        new InsertAsyncTask(mQuestionDao).execute(question);
    }

    public void update(Question question){
        new UpdateAsyncTask(mQuestionDao).execute(question);
    }

    public void delete(Question question){
        new DeleteAsyncTask(mQuestionDao).execute(question);
    }

    public void deleteAll(){
        new DeleteAllAsyncTask(mQuestionDao).execute();
    }

}
```