```java
package com.example.agendiario;

import android.content.Context;
import android.graphics.Color;

import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Filter;
import android.widget.Filterable;
import android.widget.TextView;

import androidx.recyclerview.widget.RecyclerView;

import java.util.ArrayList;
import java.util.List;

public class QuestionAdapter1 extends RecyclerView.Adapter<QuestionAdapter1.QuestionViewHolder> implements Filterable
{
    private List<Question> mQuestions;
    private List<Question> mQuestionFull;

    public interface OnItemClickListener{
        void onItemClick(Question question);
    }

    private OnItemClickListener mOnItemClickListener;

    private final LayoutInflater mInflater;

    @Override
    public QuestionViewHolder onCreateViewHolder(ViewGroup parent, int viewType){
        View itemView = mInflater.inflate(R.layout.recyclerview_item_questions, parent, false);
        return new QuestionViewHolder(itemView);
    }

    @Override
    public void onBindViewHolder(QuestionViewHolder holder, int position){
        Question current = mQuestions.get(position);
        holder.description.setText(current.getDescription());
        if(current.getStatus() == 0 ) {
            holder.description.setTextColor(Color.LTGRAY);
        } else {
            holder.description.setTextColor(Color.parseColor("#737373"));
        }
    }

    public class QuestionViewHolder extends RecyclerView.ViewHolder{
        private final TextView description;

        private QuestionViewHolder(View itemView){
            super(itemView);
            description = itemView.findViewById(R.id.text_recycler_name_question);

            itemView.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View v) {
                    int position = getAdapterPosition();
                    if( mOnItemClickListener != null && position != RecyclerView.NO_POSITION ){
                        mOnItemClickListener.onItemClick(mQuestions.get(position));
                    }
                }
            });

        }

    }


    @Override
    public int getItemCount(){
        if(mQuestions != null) {
            return mQuestions.size(); // quantidade elementos na lista
        } else {
            return 0;  // como a lista é vazia
        }
    }

    public void setQuestions(List<Question> questions){
        mQuestions = questions;
        mQuestionFull = new ArrayList<>(questions);
        notifyDataSetChanged();
    }

    public Question getQuestionAt(int positon){
        return mQuestions.get(positon);
    }

    public void setOnItemClickListener(OnItemClickListener listener){
        this.mOnItemClickListener = listener;
    }

    public QuestionAdapter1(Context context){
        mInflater = LayoutInflater.from(context);
    }

    private Filter questionFilter = new Filter() {
        @Override
        protected FilterResults performFiltering(CharSequence constraint) {
            List<Question> filteredList = new ArrayList<>();
            if(constraint == null || constraint.length() == 0){
                filteredList.addAll(mQuestionFull);
            } else {
                String filterPattern = constraint.toString().toLowerCase().trim();
                for(Question item : mQuestionFull){
                    if(item.getDescription().toLowerCase().contains(filterPattern)){
                        filteredList.add(item);
                    }
                }
            }
            FilterResults results = new FilterResults();
            results.values = filteredList;
            return results; // <<<<<<<<<
        }

        @Override
        protected void publishResults(CharSequence constraint, FilterResults results) {

        }
    };

    @Override
    public Filter getFilter(){
        return questionFilter;
    }

}
```