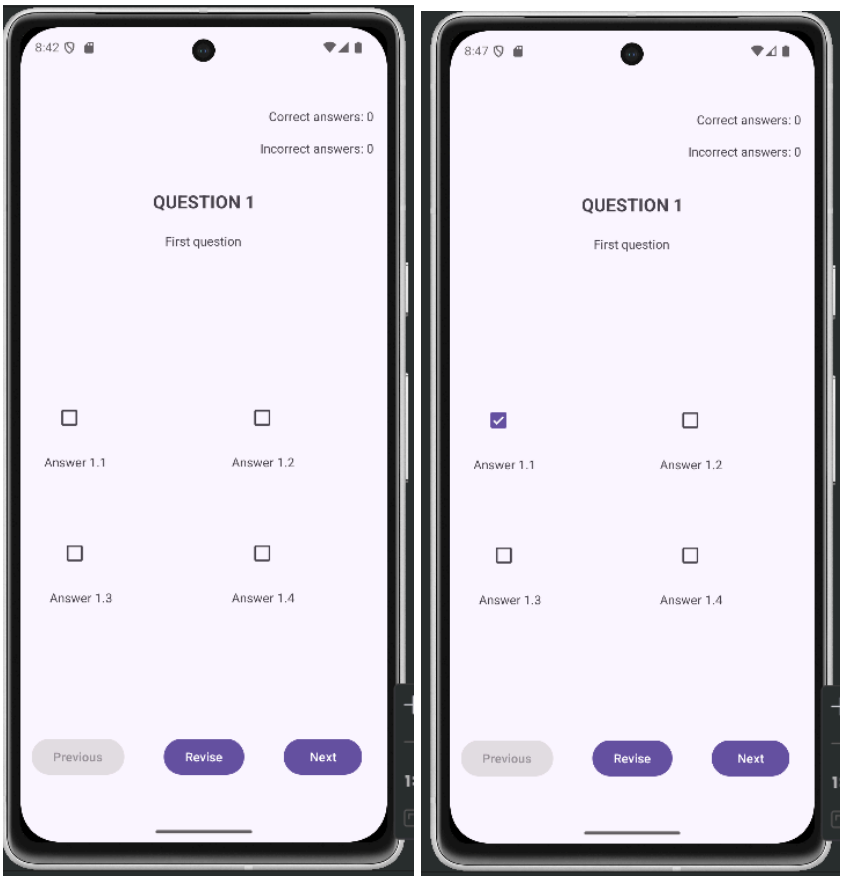
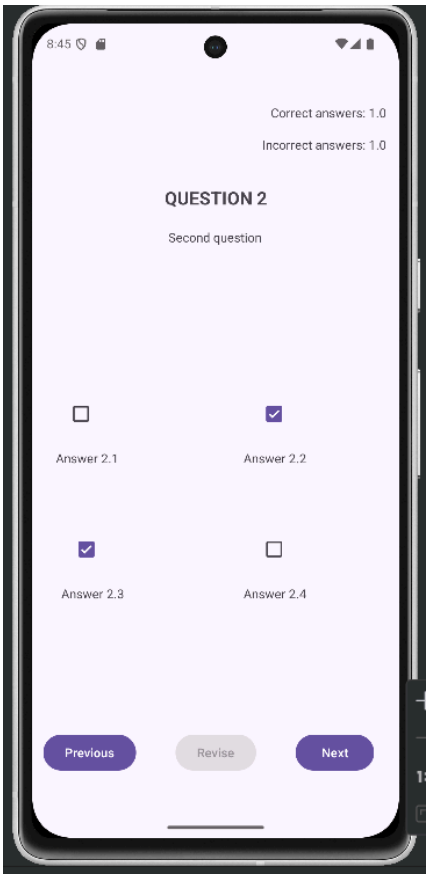
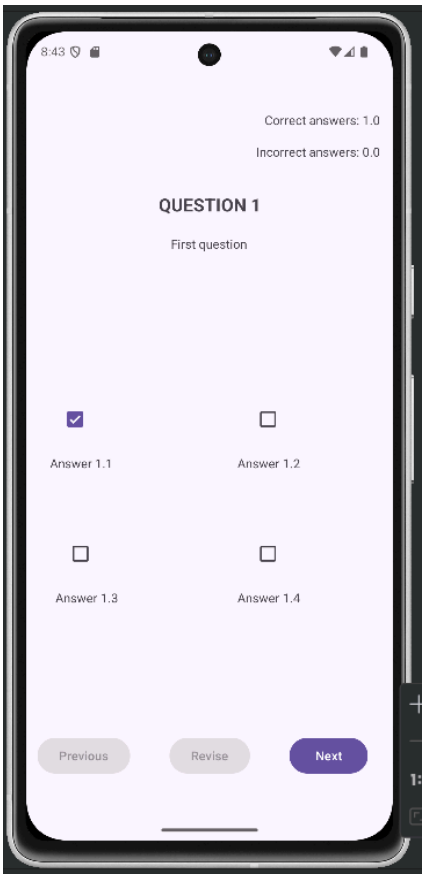
Tarea individual 7 - Juego de Trivia

Inicia la actividad con 4 CheckBox, 8 TextViews y 3 botones.
Si marcamos una opcion y pasamos a la siguiente pregunta sin revisarla, se quedara guardado como marcado, pero no puntuará. Si volvemos después se automarcará la que estaba seleccionada (imagen 4).
No deja volver a la pregunta anterior si no hay pregunta anterior, lo mismo con la ultima pregunta.



Si se revisa una pregunta deshabilita el boton de revisar.
Lo que hace al revisar es comprobar si cada CheckBox esta seleccionado, y lo compara con si la respuesta es correcta o no. Si aciertas todo, suma 1 punto.
Esta aplicación es escalable, lo único que habría que hacer para sumarle más preguntas es añadirlas al String[] questions, String[] answers, y poner cuáles son verdaderas y falsas en el boolean[] isCorrect.





Código:

```
16 ▷ </>  public class MainActivity extends AppCompatActivity {
17
18          private int actQuest = 1;  42 usages
19
20          private double correct = 0.0, incorrect = 0.0;  3 usages
21
22          private String[] questions = {  5 usages
23                  "First question",
24                  "Second question",
25                  "Third question"
26          };
27
28          private String[] answers = {  12 usages
29                  "Answer 1.1", "Answer 1.2", "Answer 1.3", "Answer 1.4",
30                  "Answer 2.1", "Answer 2.2", "Answer 2.3", "Answer 2.4",
31                  "Answer 3.1", "Answer 3.2", "Answer 3.3", "Answer 3.4"
32          };
33
34          private boolean[] isCorrect = {      //respuestas validas  4 usages
35                  true, false, false, false,
36                  true, false, true, false,
37                  true, false, false, true
38          };
```

```
40          private boolean[] isCheked = {      //ck en diferentes preguntas  16 usages
41                  false, false, false, false,
42                  false, false, false, false,
43                  false, false, false, false
44          };
45
46          private boolean[] corrected = {  4 usages
47                  false, false, false
48          };
49
50          private TextView tvNumQuestion, tvTitle, tvCor, tvIncor, tvA1, tvA2, tvA3, tvA4;  3 usages
51          private Button btnPrev, btnRev, btnNext;  5 usages
52          private CheckBox ck1, ck2, ck3, ck4;  4 usages
53
```

```java
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    EdgeToEdge.enable( $this$enableEdgeToEdge: this);
    setContentView(R.layout.activity_main);
    ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
        Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
        v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
        return insets;
    });
    tvCor = findViewById(R.id.tvCor);
    tvIncor = findViewById(R.id.tvIncor);


    tvTitle = findViewById(R.id.tvTitle);
    tvTitle.setText(questions[0]);
    tvNumQuestion = findViewById(R.id.tvNumQuestion);
    tvA1 = findViewById(R.id.tvA1);
    tvA1.setText(answers[0]);
    tvA2 = findViewById(R.id.tvA2);
    tvA2.setText(answers[1]);
    tvA3 = findViewById(R.id.tvA3);
    tvA3.setText(answers[2]);
    tvA4 = findViewById(R.id.tvA4);
    tvA4.setText(answers[3]);
```

```java
    btnPrev = findViewById(R.id.btnPrev);
    btnPrev.setOnClickListener(new View.OnClickListener(){
        @Override
        public void onClick(View v){
            actQuest = actQuest - 1;
            if(corrected[actQuest - 1] == false){
                btnRev.setEnabled(true);
            }else{
                btnRev.setEnabled(false);
            }
            tvNumQuestion.setText("QUESTION " + Integer.toString(actQuest));
            tvTitle.setText(questions[actQuest - 1]);
            tvA1.setText(answers[actQuest*4-4]);
            tvA2.setText(answers[actQuest*4-3]);
            tvA3.setText(answers[actQuest*4-2]);
            tvA4.setText(answers[actQuest*4-1]);

            ck1.setChecked(isCheked[actQuest*4-4]);
            ck2.setChecked(isCheked[actQuest*4-3]);
            ck3.setChecked(isCheked[actQuest*4-2]);
            ck4.setChecked(isCheked[actQuest*4-1]);

            if(actQuest == 1){
                btnPrev.setEnabled(false);
            }
```

```
                if(actQuest == questions.length - 1){
                    btnNext.setEnabled(true);
                }

            }
        });


        btnRev = findViewById(R.id.btnRev);
        btnRev.setOnClickListener(new View.OnClickListener(){
            @Override
            public void onClick(View v){

                if(isCheked[actQuest*4-4] == isCorrect[actQuest*4-4] &&
                        isCheked[actQuest*4-3] == isCorrect[actQuest*4-3] &&
                        isCheked[actQuest*4-2] == isCorrect[actQuest*4-2] &&
                        isCheked[actQuest*4-1] == isCorrect[actQuest*4-1]){
                    correct = correct + 1;
                }else{
                    incorrect = incorrect + 1;
                }
```

```
                tvCor.setText("Correct answers: " + Double.toString(correct));
                tvIncor.setText("Incorrect answers: " + Double.toString(incorrect));
                corrected[actQuest - 1] = true;
                StringBuilder sb = new StringBuilder();
                for (boolean b : corrected) {
                    sb.append(b ? "true " : "false ").append(" ");
                }
                String arrayContent = sb.toString().trim();
                Toast.makeText( context: MainActivity.this, arrayContent, Toast.LENGTH_SHORT).show();
                btnRev.setEnabled(false);

            }
        });


        btnNext = findViewById(R.id.btnNext);
        btnNext.setOnClickListener(new View.OnClickListener(){
            @Override
            public void onClick(View v){

                if(corrected[actQuest] == false){
                    btnRev.setEnabled(true);
                }else{
```

```
                btnRev.setEnabled(false);
            }


            actQuest = actQuest + 1;
            tvNumQuestion.setText("QUESTION " + Integer.toString(actQuest));
            tvTitle.setText(questions[actQuest - 1]);
            tvA1.setText(answers[actQuest*4-4]);
            tvA2.setText(answers[actQuest*4-3]);
            tvA3.setText(answers[actQuest*4-2]);
            tvA4.setText(answers[actQuest*4-1]);

            ck1.setChecked(isCheked[actQuest*4-4]);
            ck2.setChecked(isCheked[actQuest*4-3]);
            ck3.setChecked(isCheked[actQuest*4-2]);
            ck4.setChecked(isCheked[actQuest*4-1]);


            if(!(btnPrev.isEnabled())){
                btnPrev.setEnabled(true);
            }
```

```
            if(actQuest>=questions.length){
                btnNext.setEnabled(false);
            }

        }
    });


    ck1 = findViewById(R.id.ck1);
    ck1.setOnCheckedChangeListener((buttonView, isThisChecked) -> {
        isCheked[actQuest*4-4] = isThisChecked;
    });

    ck2 = findViewById(R.id.ck2);
    ck2.setOnCheckedChangeListener((buttonView, isThisChecked) -> {
        isCheked[actQuest*4-3] = isThisChecked;
    });
    ck3 = findViewById(R.id.ck3);
    ck3.setOnCheckedChangeListener((buttonView, isThisChecked) -> {
        isCheked[actQuest*4-2] = isThisChecked;
    });
```

```
        ck4 = findViewById(R.id.ck4);
        ck4.setOnCheckedChangeListener((buttonView, isThisChecked) -> {
            isCheked[actQuest*4-1] = isThisChecked;
        });

    }
}
```