

## Práctica 2

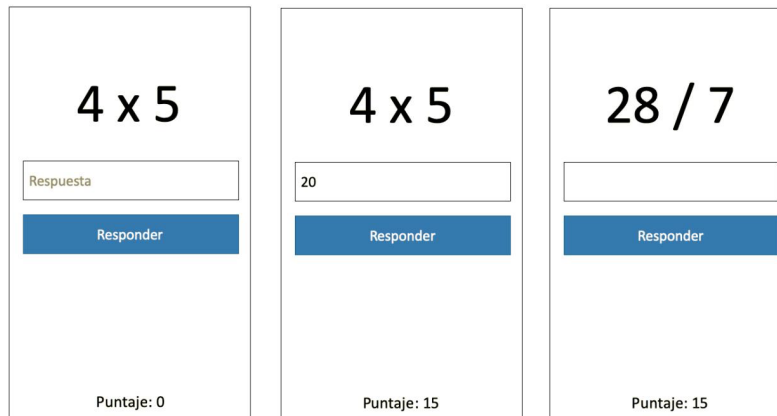
### Objetivo

En esta práctica debe realizar una aplicación Android capaz de interactuar con un usuario. Para eso debe usar los listeners `OnClickListener` y `onTouchListener`.

### Consigna

Icesi Games en asociación con Matileo lo contrata para que realice una aplicación que permita afiazar el conocimiento en aritmética de los estudiantes de Icesi.

Matileo decidió que la aplicación se llame Math Challenge y que consista en presentarle una serie de preguntas y que ofrezca un puntaje a partir de las respuestas.



The figure shows three sequential screens of the Math Challenge app. Each screen displays a math problem, a text input field for the answer, a blue 'Responder' button, and the current score at the bottom.

- Screen 1:** Problem is  $4 \times 5$ . The input field is empty with the placeholder text 'Respuesta'. The score is 'Puntaje: 0'.
- Screen 2:** Problem is  $4 \times 5$ . The input field contains the number '20'. The score is 'Puntaje: 15'.
- Screen 3:** Problem is  $28 / 7$ . The input field is empty. The score is 'Puntaje: 15'.

Figura 1. Pantalla de Math Challenge

Para las preguntas cree un banco de preguntas (arreglo de preguntas) de modo que vayan saliendo de forma aleatoria en la pantalla.

Para eso modele una clase `Pregunta` en la que considere tener tanto la pregunta como la respuesta, para que pueda hacer la validación directamente.

Si el jugador acierta recibe 5 puntos, si falla se le quitan 4 puntos.

### ¡AVANZADO!

1. Limite el juego para que dure 30 segundos. Este tiempo debe aparecer en la esquina superior derecha en todo momento.  
Una vez terminado el tiempo, puede volver a jugar si da click en el botón "Intentar de nuevo".
2. Logre que pulsando durante un 1.5 segundos sobre la pregunta pueda "pasar" y se muestre la siguiente pregunta.

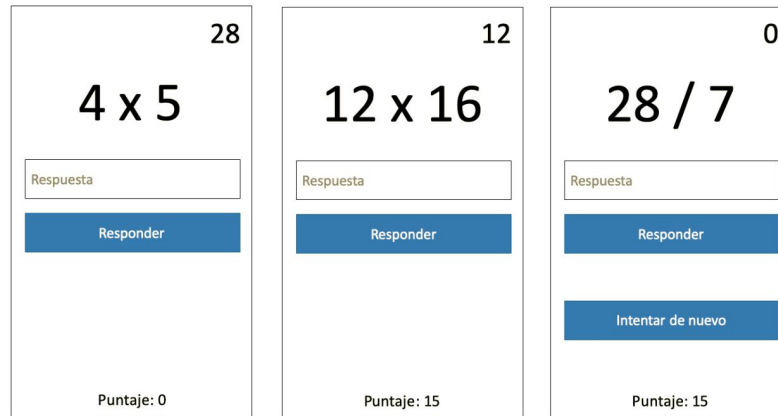


Figura 2. Contador y botón de intentar de nuevo

Note que el botón “Intentar de nuevo” aparece cuando temporizador finaliza. Usted puede hacer aparecer y desaparecer elementos visuales usando la propiedad visibility.



Esta propiedad puede ser VISIBLE, INVISIBLE (el botón aún permanece activo, solo que ahora es transparente) y GONE (es como si el botón ya no estuviera y no hay forma de darle click).

En código también puede cambiar la visibilidad usando

```
miBoton.setVisibility(View.GONE)
```

Para ocultar el botón

```
miBoton.setVisibility(View.VISIBLE)
```

Para hacer aparecer el botón