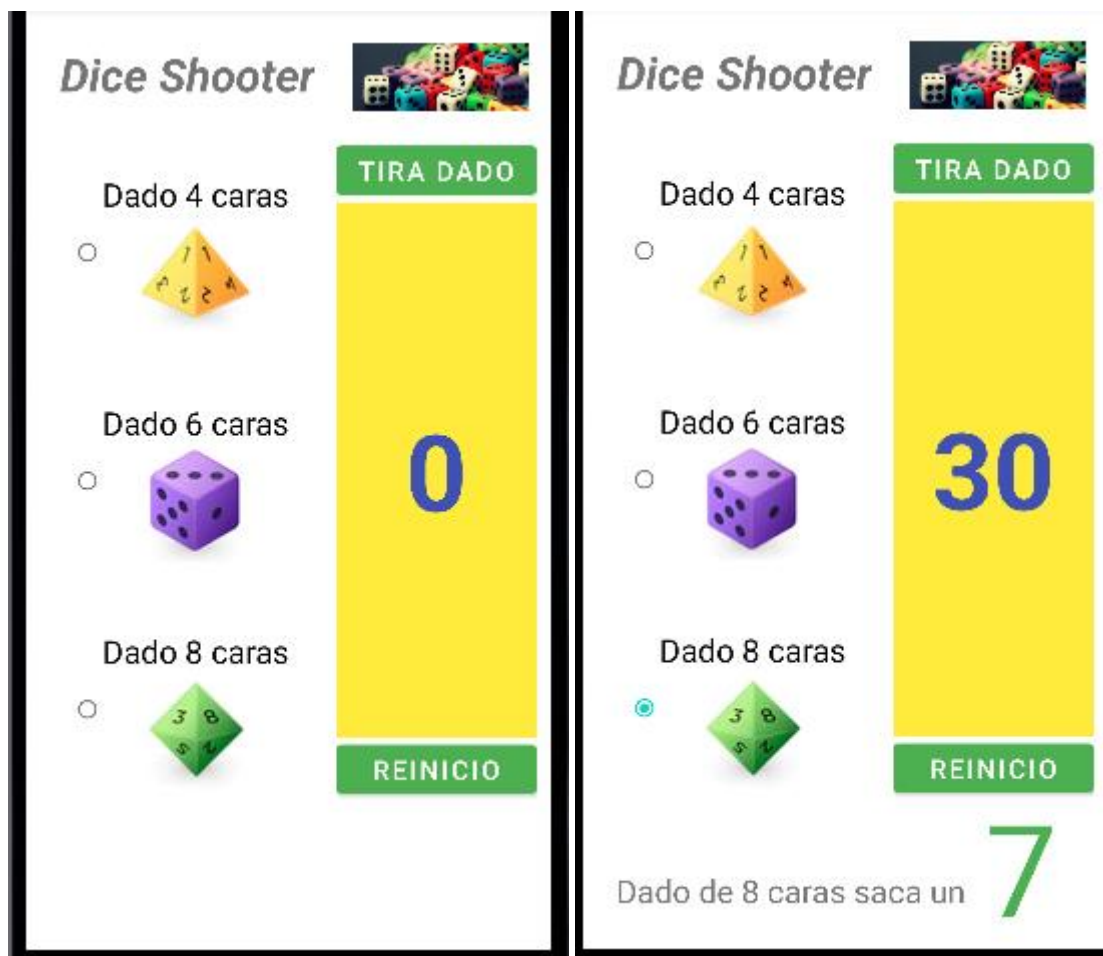


## Práctica 4 – Dados

En esta práctica desarrollaremos una sencilla interfaz gráfica de un **lanzador de dados** en Android Studio, para practicar el uso de eventos y la creación de nuevas actividades.

### Práctica

**1.** Desarrolla una app en Java con Android Studio (de nombre “*Dados xxx*”, siendo *xxx* las iniciales de tu nombre y apellidos) para ser utilizada con la API 31 de Android. La app simulará tiradas de dados, permitiendo al usuario elegir entre tirar dados de 4, 6 y 8 caras, mostrando el resultado de la tirada y el total de la partida (suma del valor de las tiradas):



La app tendrá las siguientes características:

- La interfaz de la app deberá utilizar de manera correcta y justificada al menos un **ConstraintLayout** y un **LinearLayout**. Las imágenes de la app se te proporcionarán y serán *logo.jpg*, *dado4caras.jpg*, *dado6caras.jpg* y *dado8caras.jpg*. El color de los botones y del número de la tirada es #4CAF50, el de la puntuación total es #3F51B5 y su fondo es #FFEB3B, aunque puedes utilizar colores parecidos (no los colores por defecto).

- b. La app tendrá como interfaz principal de inicio la de la primera imagen. El usuario puede seleccionar únicamente uno de los tres dados que se le ofrecen (Dado de 4 caras, Dado de 6 caras, Dado de 8 caras). Al pulsar el botón *Tira dado* únicamente realizará la tirada si el usuario ha seleccionado uno de los dados.
- c. Cuando el usuario pulse el **botón Tira dado** (habiendo seleccionado previamente un dado) se le mostrará un mensaje al usuario similar al de la segunda imagen ("*Dado de 8 caras saca un 7*"). Este texto no es fijo y se corresponderá con el número de caras que haya seleccionado el usuario y el valor aleatorio de la tirada (Si es un dado de 4 caras, puede ser del 1 al 4, si es de 6 caras del 1 al 6 y si es de 8 caras del 1 al 8). Debajo del botón Tira dado aparecerá la puntuación total de la partida, fruto de sumar a la puntuación actual la tirada del dado que se ha producido. Cuando el usuario pulse el **botón Reinicio** esta puntuación total se reiniciará a 0.

\* Método para calcular un número aleatorio entre dos valores enteros (incluyendo ambos) \*

```
private int getRandomNumber(int min, int max) {
    return (int) ((Math.random() * (max - min + 1)) + min);
}
```

- d. La app deberá contar con un diseño de pantalla específico para cuando el usuario visualice su móvil en orientación horizontal. El usuario deberá poder utilizar la totalidad de la app (ya sea haciendo scroll, o con una disposición que le permita ver todo en pantalla como por ejemplo en la siguiente imagen):



El estado de la app se deberá preservar cuando el usuario sale de la app para volver posteriormente o cuando cambia la orientación del móvil. Para lograr este comportamiento utilizaremos la clase Bundle (android.os) y sobrescribiremos el método onSaveInstanceState.

- e. La app deberá contar con una pantalla de login previa a esta donde el usuario podrá introducir un apodo, esta pantalla de login tendrá un botón donde solo si el usuario ha introducido un apodo correcto (más de 2 caracteres) la app pasará a la pantalla principal del lanzador de dados.

- f. La app deberá contar con una pantalla posterior a esta que se lanzará cuando el usuario llegue a 24 o más puntos. En esta pantalla posterior se mostrará un mensaje informando de sus tiradas de una manera similar a esta:

\_\_\_\_\_ (el apodo del usuario) ha logrado un total de \_\_\_\_ puntos en \_\_\_\_ tiradas:

\_\_\_\_ tiradas con el dado de 4 caras para lograr \_\_\_\_ puntos

\_\_\_\_ tiradas con el dado de 6 caras para lograr \_\_\_\_ puntos

\_\_\_\_ tiradas con el dado de 8 caras para lograr \_\_\_\_ puntos

Desde esta pantalla no se podrá volver a la pantalla principal de la app, únicamente a la pantalla de login para poder iniciar otra ronda de tiradas de dado con un nuevo usuario.

