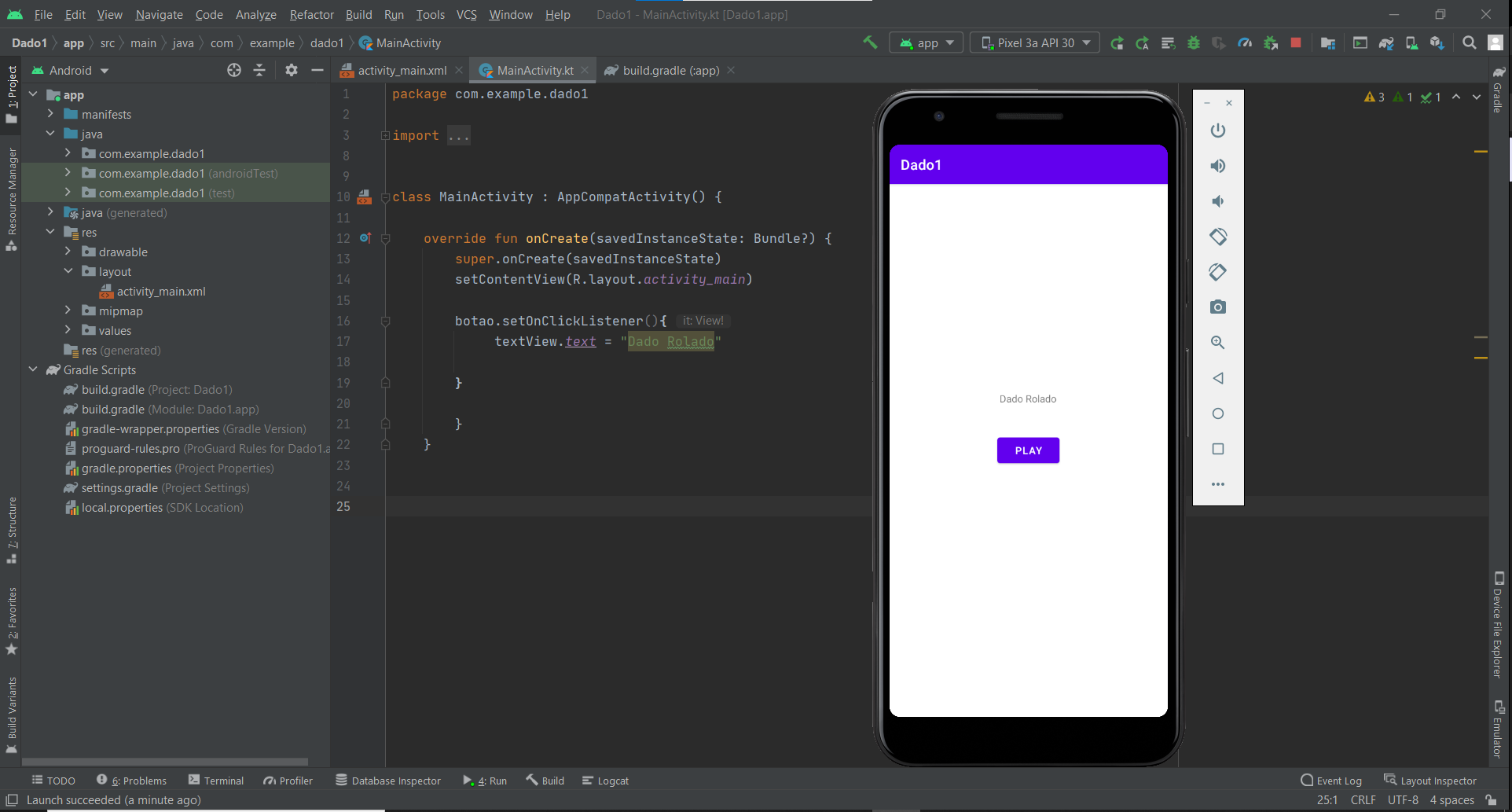


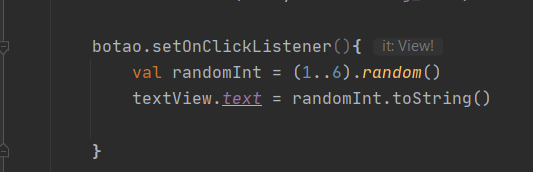
[5. Tarefa: Alterar o texto](https://developer.android.com/codelabs/kotlin-android-training-app-anatomy#4)

Nesta tarefa, você modifica o rollDice()método para alterar o texto no TextView. Para a primeira etapa, você altera esse texto de "Hello World!"para a string "Dice Rolled!". Para a segunda etapa, você exibe um número aleatório entre um e seis.

**Etapa 1: exibir uma string**



**Etapa 2: exibir um número aleatório**

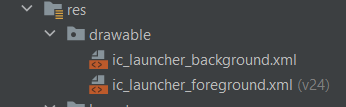


# [3. Tarefa: Adicionar e atualizar recursos de imagem](https://developer.android.com/codelabs/kotlin-android-training-images-compat#2)

você tinha um aplicativo que atualiza uma visualização de texto com um número entre 1 e 6 cada vez que o usuário toca em um botão. No entanto, o aplicativo é chamado DiceRoller, não 1-6 Number Generator, então seria bom se os dados realmente parecessem dados. Nesta tarefa, você adiciona algumas imagens de dados ao seu aplicativo. Então, em vez de atualizar o texto quando o botão é pressionado, você troca uma imagem diferente para cada resultado da rolagem.

# Etapa 1: adicione as imagens

1. Abra o projeto no Android Studio se ainda não estiver aberto.
2. Na visualização Projeto> Android, expanda a pasta **res** e, em seguida, expanda **drawable** .

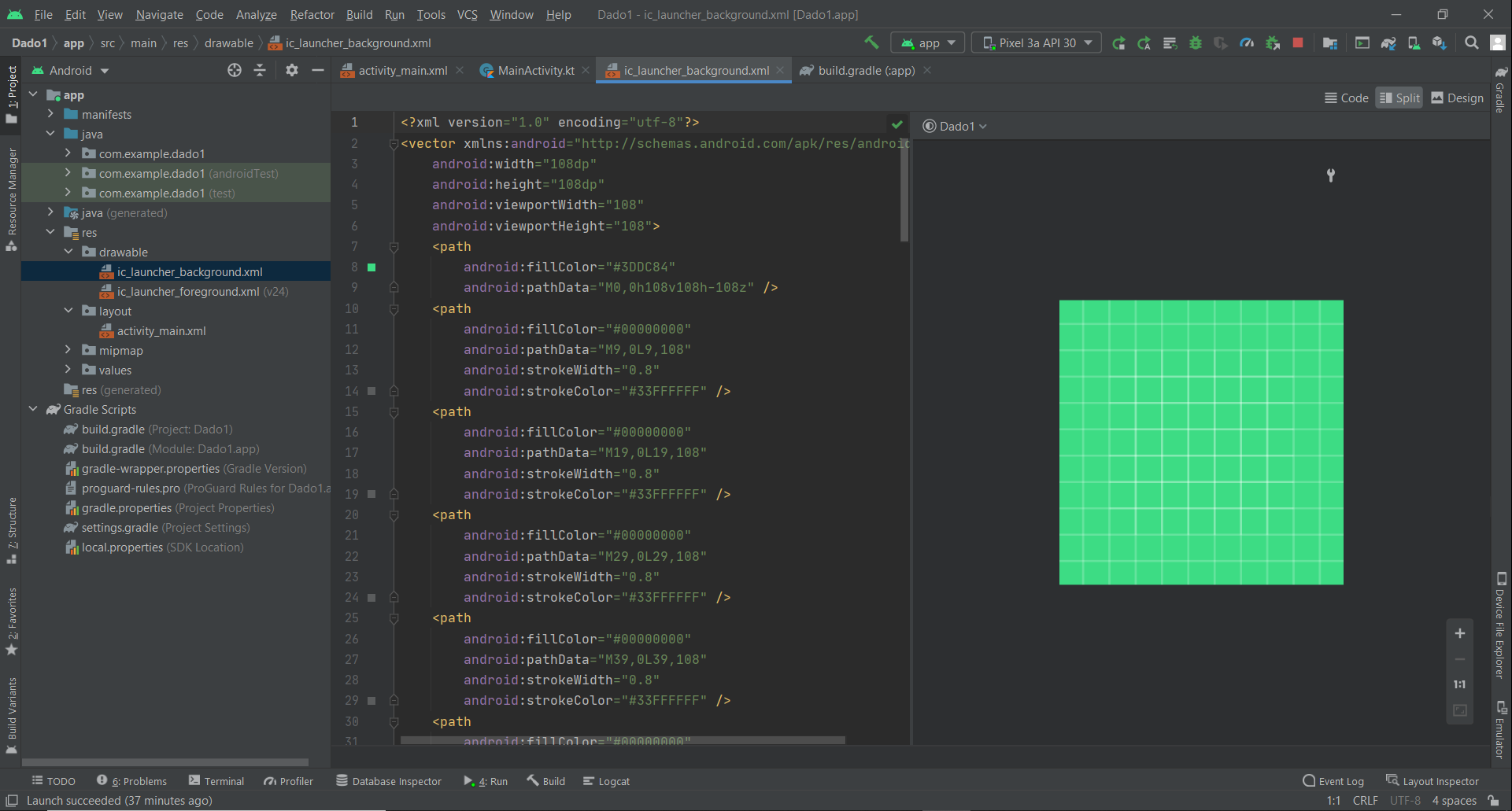


Seu aplicativo usa muitos recursos diferentes, incluindo imagens e ícones, cores, strings e layouts XML.

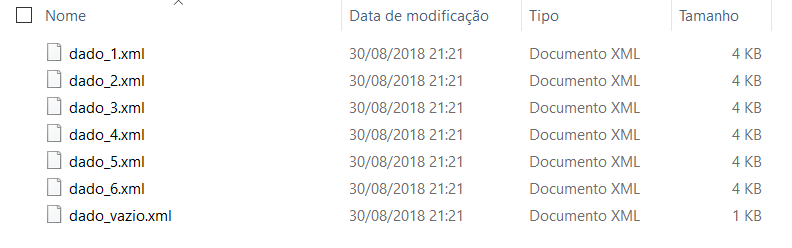
Todos esses recursos são armazenados na pasta res. A pasta drawable é onde você deve colocar todos os recursos de imagem para seu aplicativo.

Já na drawable você pode encontrar os recursos para os ícones do iniciador do aplicativo.

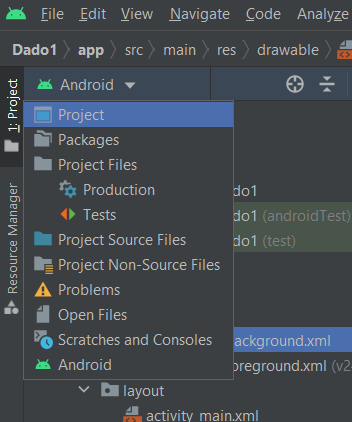
3. Clique duas vezes em **ic\_launcher\_background.xml** . Observe que esses são arquivos XML que descrevem o ícone como uma imagem vetorial. Os vetores permitem que suas imagens sejam desenhadas em muitos tamanhos e resoluções diferentes. As imagens de bitmap, como PNG ou GIF, podem precisar ser dimensionadas para dispositivos diferentes, o que pode resultar em alguma perda de qualidade. 4. Clique em **Visualizar** na coluna direita do editor XML para visualizar o drawable vetorial em formato visual.



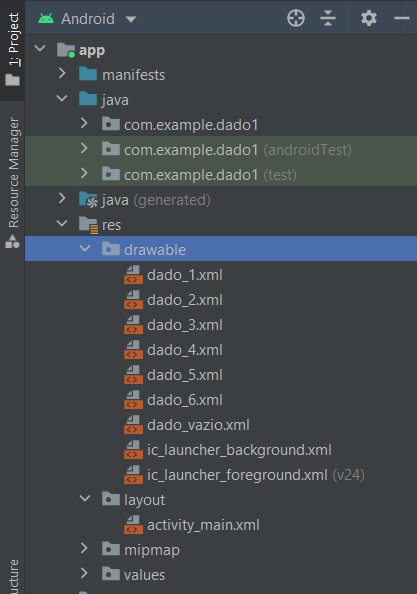
1. Baixe as imagens de dados para seu aplicativo na pasta ibagens.
2. Descompacte o arquivo. Você deve ter uma pasta de arquivos XML parecida com esta:



1. No Android Studio, clique no menu suspenso na parte superior da visualização do projeto que atualmente diz **Android** e escolha **Projeto** . A captura de tela abaixo mostra a estrutura do seu aplicativo no sistema de arquivos.



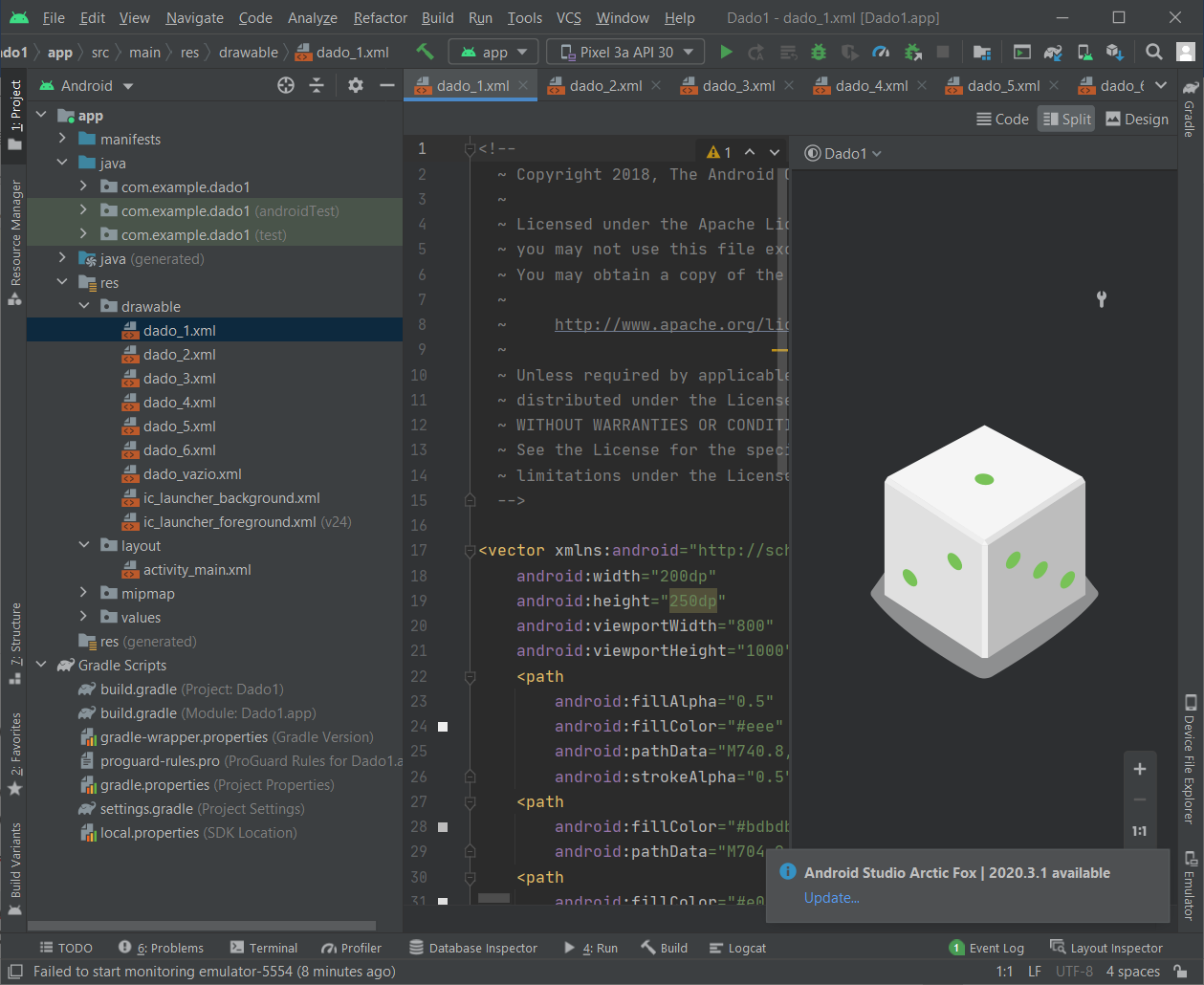
1. Expanda **DiceRoller> app> src> main> res> drawable** .
2. Arraste todos os arquivos XML individuais da pasta ibagens para o Android Studio e para a pasta **drawable** . Clique **OK** .



**Nota:** Certifique-se de soltar os arquivos na pasta **drawable** e não na pasta **drawable24** . Você aprenderá mais sobre esta pasta e para que as outras são usadas posteriormente.

Além disso, não inclua a pasta ibagens em si, apenas os arquivos. Arraste apenas os arquivos XML.

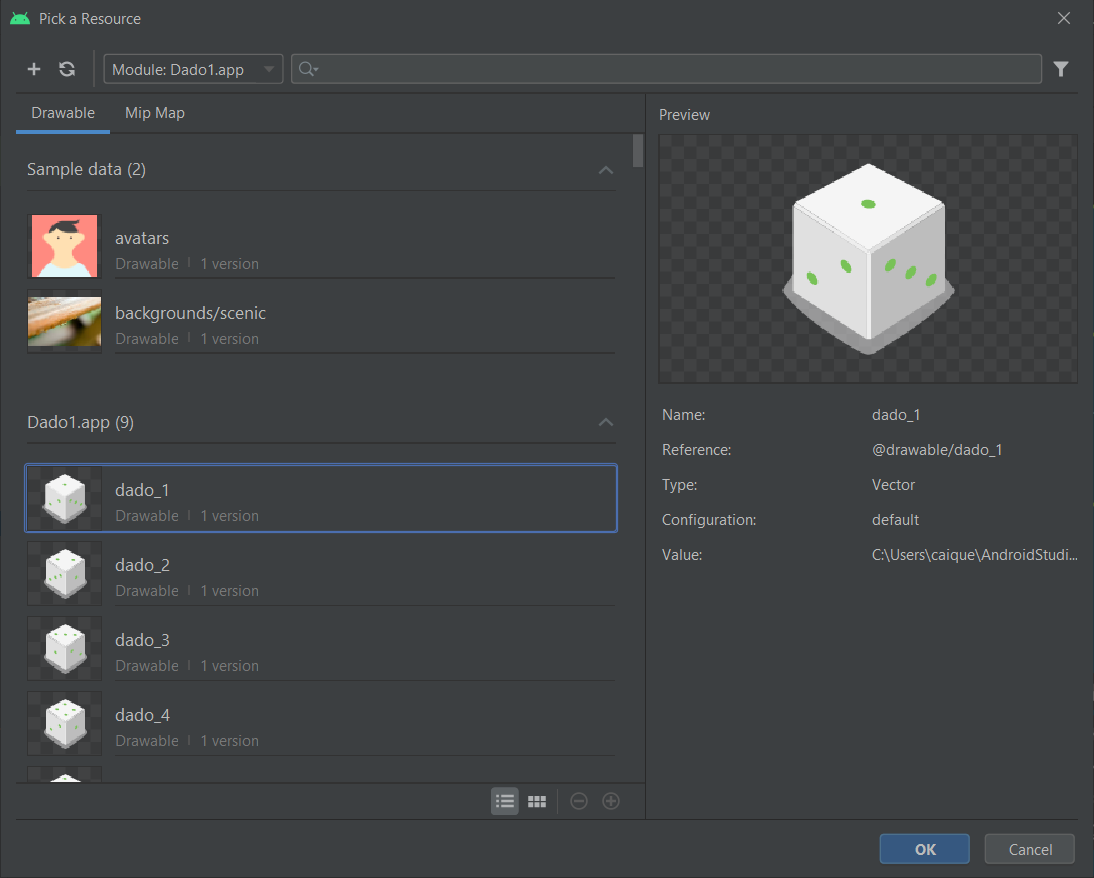
1. Mude o projeto de volta para a visualização do **Android** e observe que os arquivos XML de imagem de dados estão na pasta drawable.
2. Clique duas vezes dice\_1.xmle observe o código XML para esta imagem. Clique no botão **Visualizar** para obter uma visualização da aparência real desse drawable vetorial.

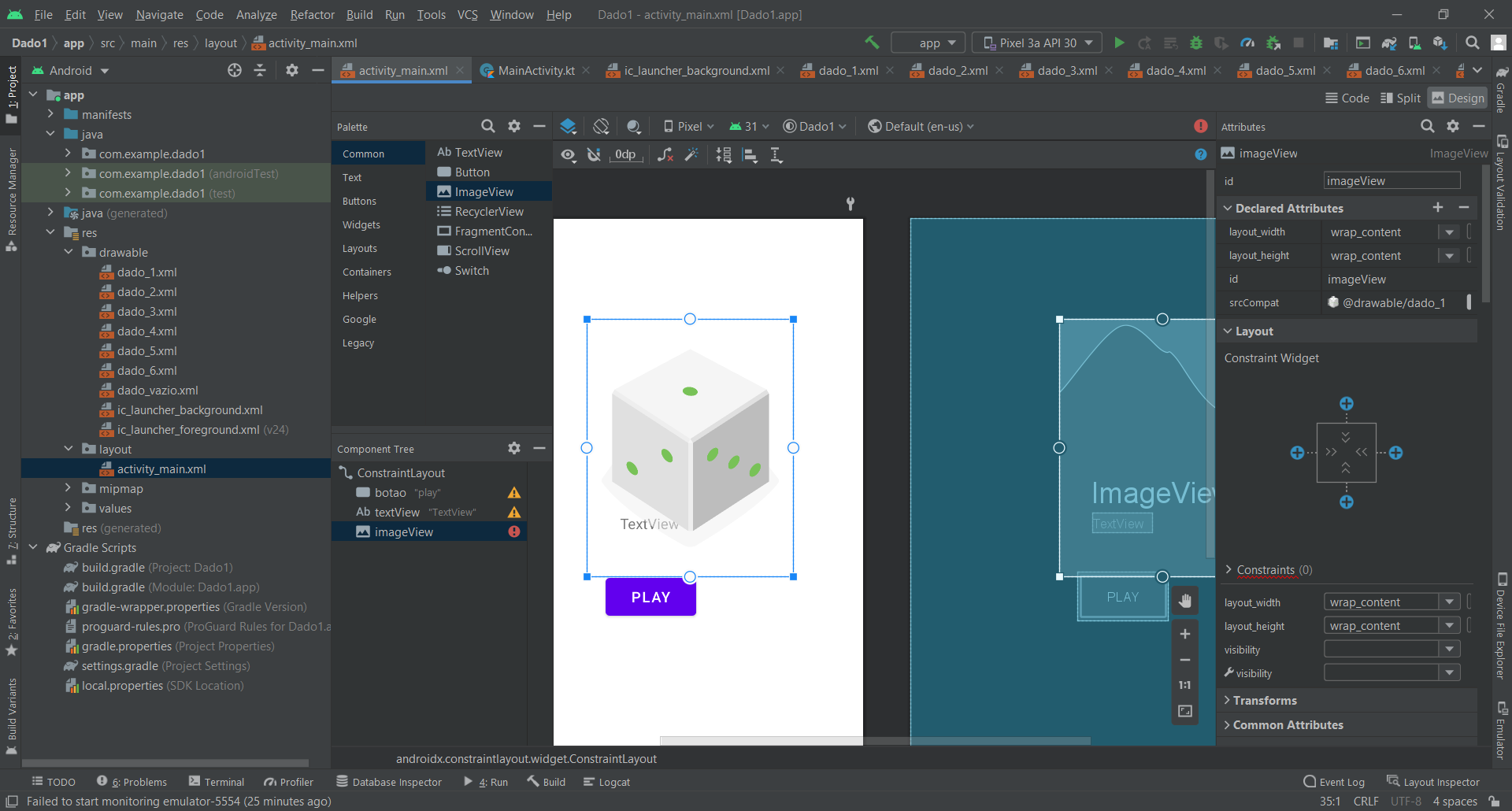


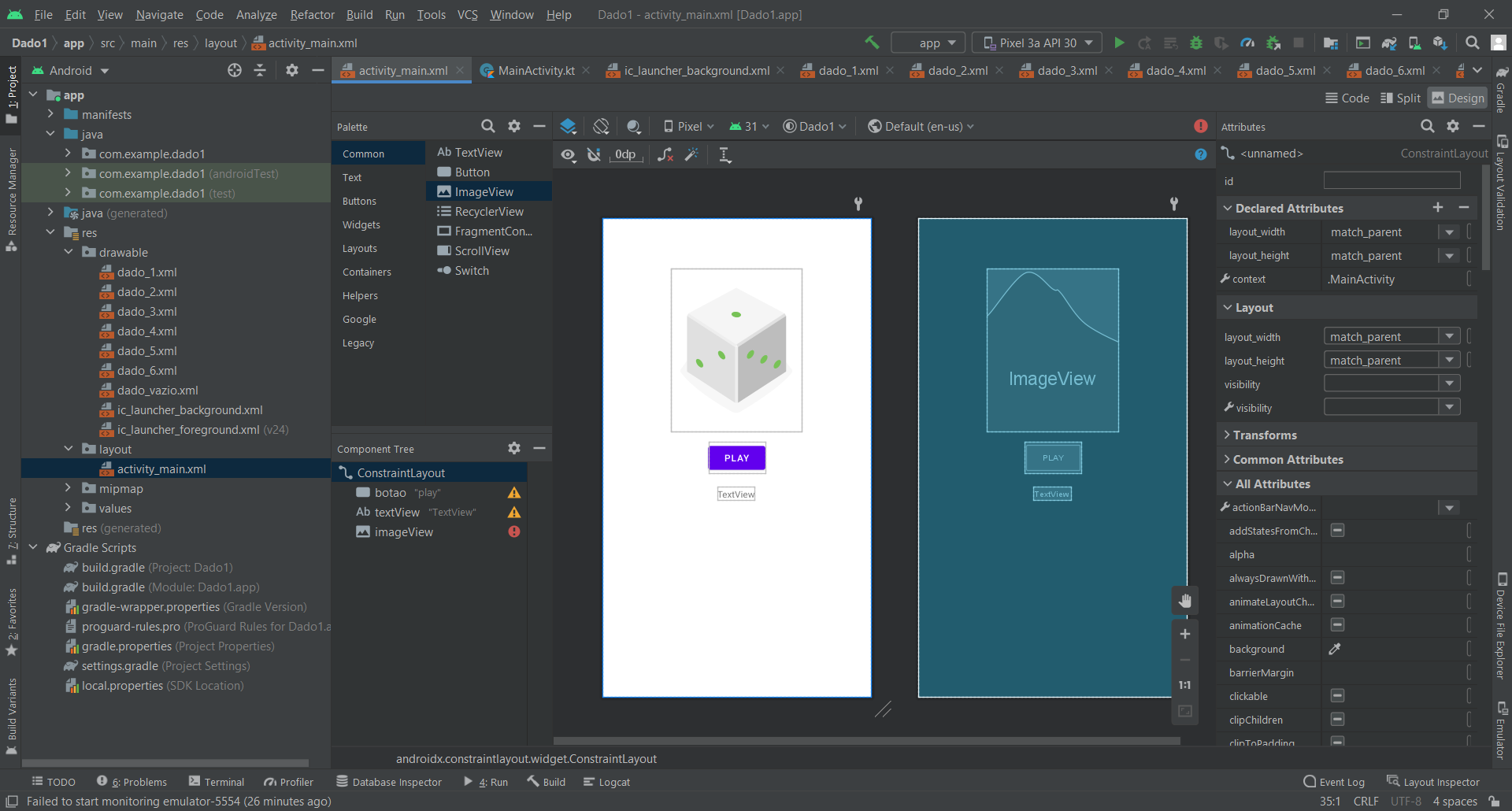
# Etapa 2: atualize o layout para usar imagens

Agora que você tem os arquivos de imagem de dados em sua pasta res/drawables, pode acessar esses arquivos a partir do layout e código do seu aplicativo.

1. Abra o activity\_main.xml de layout, se ainda não estiver aberto. Clique na guia **Texto** para visualizar o código XML do layout.
2. Adicione um <ImageView>







Use um ImageView para exibir uma imagem em seu layout. O único atributo novo para este elemento é android:src