

Docentes:

Sónia Luz, sonia.luz@ipleiria.pt
David Safadinho, david.safadinho@ipleiria.pt
Cátia Ledesma, catia.ledesma@ipleiria.pt

Ficha Prática 2

Guess a Number

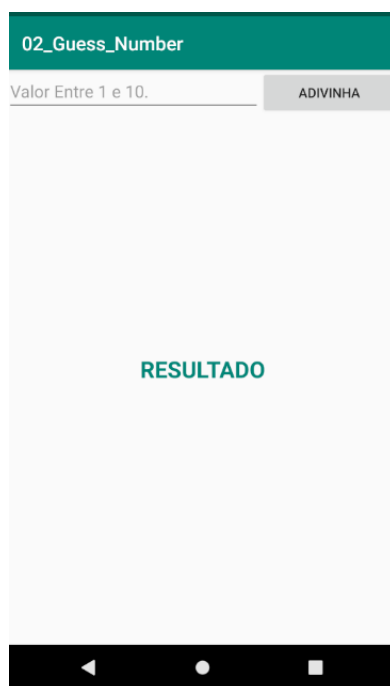
Objetivos da Ficha

- Criar um novo projeto “Adivinhar um número”
- Inserir e manipular componentes visuais: TextView, EditView e Button
- Entender as restrições dos tipos de layout

Introdução

Nesta ficha vamos criar uma aplicação que permita ao utilizador adivinhar um número gerado aleatoriamente. O utilizador irá receber indicações/dicas que permitem adivinhar o número mais facilmente.

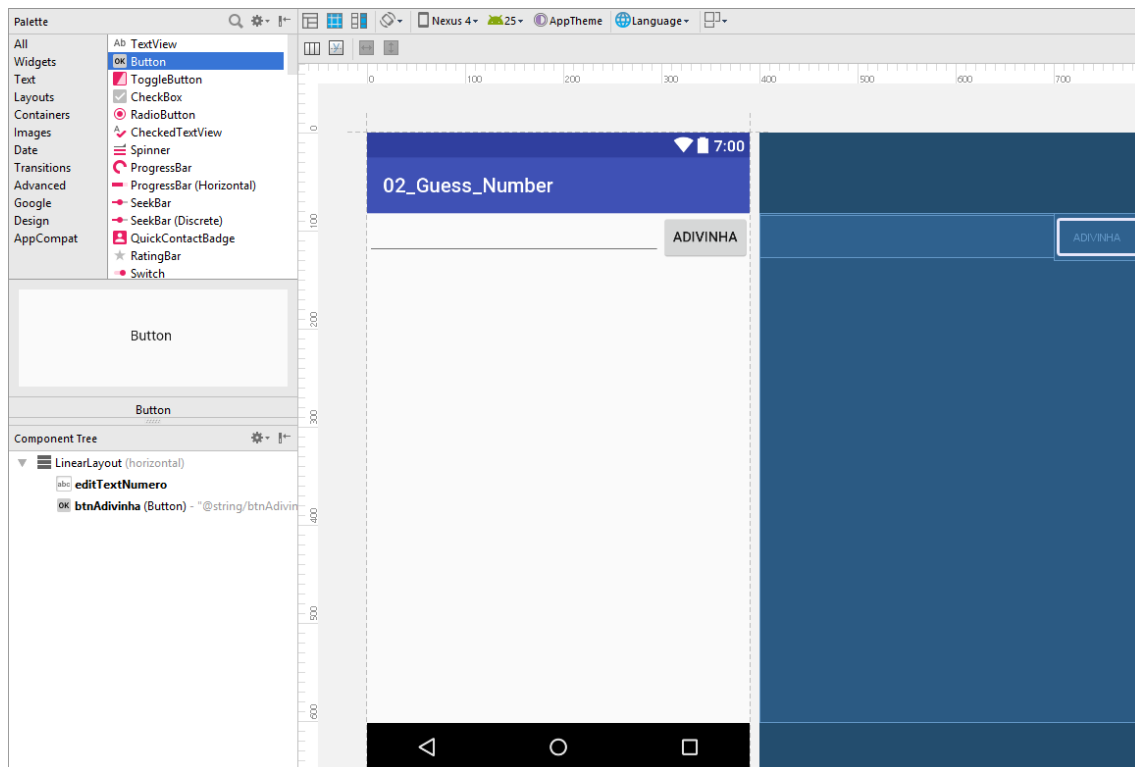
A aplicação “Guess a Number” terá um aspeto semelhante ao apresentado na figura seguinte.



Criar um novo projeto (Guess a Number)

1. Iniciar o IDE Android Studio;
2. Selecionar a opção **“New project”**;
3. Selecionar para que tipo de dispositivo iremos desenvolver esta aplicação;
 - 3.1. Neste caso será a opção para **“Phone and Tablet”**;
 - 3.2. Escolher como atividade uma **“Empty Activity”**
4. Configurar o nome **“Guess_Number”**, o package name **“amsi.dei.estg.ipleiria.pt”** e a pasta de destino da aplicação;
 - 4.1. **O nome e o caminho para a pasta de destino não podem conter espaços nem caracteres especiais;**
 - 4.2. Selecionar a linguagem de programação Java
 - 4.3. Escolher o SDK mínimo para a aplicação ser executada, API 21: Android 5.0 (Lollipop); de modo a contemplar aproximadamente 94,1% dos dispositivos móveis;
5. Após a aplicação criada podemos ver no IDE a estrutura do projeto e começar a definir a parte visual da aplicação;
6. No layout da atividade (**activity_main.xml**: se necessário devem *aceder através do Package Explorer, e no projeto em questão, selecionar a pasta “res” e a subpasta “layout”*) altere o tipo de layout de **“Constraint Layout”** para **“Linear Layout”** e elimine a TextView com o texto **“Hello World”**. Posteriormente adicione uma caixa de texto e um botão (arrastando os componentes da paleta para o layout da aplicação), a caixa de texto apenas deve aceitar números por isso deve adicionar o componente **“Edit Text Number”**;
7. Considerando o campo de texto devemos adicionar um id único **“etNumero”**, na propriedade **“ID”**;
8. De seguida devemos alterar as propriedades do botão:
 - 8.1. Devemos adicionar um id único **“btnAdivinha”**, na propriedade **“ID”**;
 - 8.2. Para cumprir as regras de arquitetura de desenho de aplicações para Android, deverá incluir um novo recurso (**String Resource**) com o texto pretendido. Com o botão selecionado, aceda ao tab de propriedades e escolha o botão **“...”** que se encontra à frente da propriedade **“Text”**. Na janela que lhe será apresentada deve adicionar um novo recurso utilizando **“+” > “String value”**. Crie um recurso com o nome **“btnAdivinha”** e com o conteúdo **“Adivinha”**;
 - 8.3. Na propriedade **onClick** deve adicionar o nome do método **“onClickAdivinha”**. Após isso deve ir ao ficheiro **activity_main.xml** e usando o **“Alt+Enter”** em cima do nome do método escolher a opção **“Create onClick event handler”**;

9. O layout da aplicação deverá ter o aspeto da figura seguinte;



10. Agora será necessário implementar as funcionalidades que permitem a um utilizador adivinhar um valor que foi gerado aleatoriamente pela aplicação;

10.1. Nesta classe, para acedermos ao campo de texto é necessário definir a variável de acesso ao componente, no método **onCreate()**;

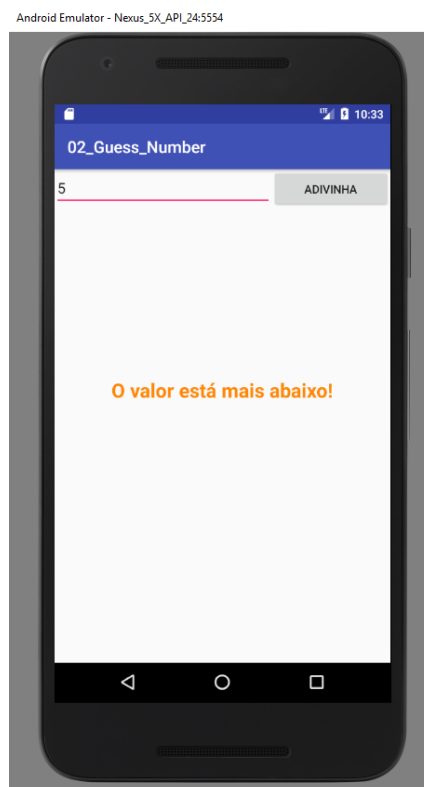
10.1.1. Para o conseguirmos fazer temos que definir a variável que referencia o elemento gráfico. Isto consegue-se recorrendo ao método **findViewById()** que a partir do identificador único de cada componente (**R.id**) os identifica;

10.2. De modo a testar a parte inicial e para conseguir ver o valor inserido pelo utilizador deve utilizar um componente do Android designado de **“Toast”**, no método **“onClickAdivinha”**;

10.3. Na classe **MainActivity.java** comece por criar um método que gere aleatoriamente um valor inteiro entre um valor mínimo e um máximo. Depois codifique os eventos e *listeners* responsáveis pela interação da aplicação;

10.3.1. Devemos definir o método para gerar aleatoriamente um valor inteiro entre dois valores recebidos por parâmetro. Este método será usado para gerar o valor que o utilizador irá adivinhar;

- 10.4. Para usar este método será necessário criar um atributo que guarde o valor gerado e que fique disponível para posterior comparação;
- 10.5. Para verificarmos se o utilizador acertou, ou não, no número gerado aleatoriamente será necessário alterar o método **“onClickAdivinha”**;
11. Após termos conseguido apresentar a informação utilizando um componente **“Toast”**, vamos alterar a aplicação de modo a que essa informação passe a ser apresentada através de um componente **“Text View”**. Proceda às alterações necessárias;
- 11.1. Com esta alteração quando o utilizador tenta acertar no valor gerado, o resultado será semelhante ao apresentado na figura seguinte, onde apresenta a informação se está abaixo, acima ou acertou no número gerado;



12. Para darmos mais pormenor à aplicação, vamos limitar o número de tentativas possíveis para um valor predefinido;
- 12.1. O resultado deverá ser semelhante ao apresentado nas figuras seguintes.
- 12.2. Será necessário criar um método para iniciar jogo cada vez que o número de tentativas é excedido ou quando o utilizador acerta no número.

