



ANDROID

NumberGuess App



Prof. Alessandro Brawerman

NumberGuess App

- O objetivo é fixar conceitos e propriedades relacionadas a botões, EditText e TextView.
- NumberGuess
 - Neste app o usuário deverá adivinhar um número sorteado de maneira aleatória, entre 1 e 100.
 - Ao acertar, o app mostra quantas tentativas foram necessárias



NumberGuess App

- Será necessário criarmos uma activity com os componentes de UI: EditText, TextView, Button.
- Vamos também testar alguns dos layouts disponíveis no design de activities.
- Ao final a activity deverá ser similar a ao lado.

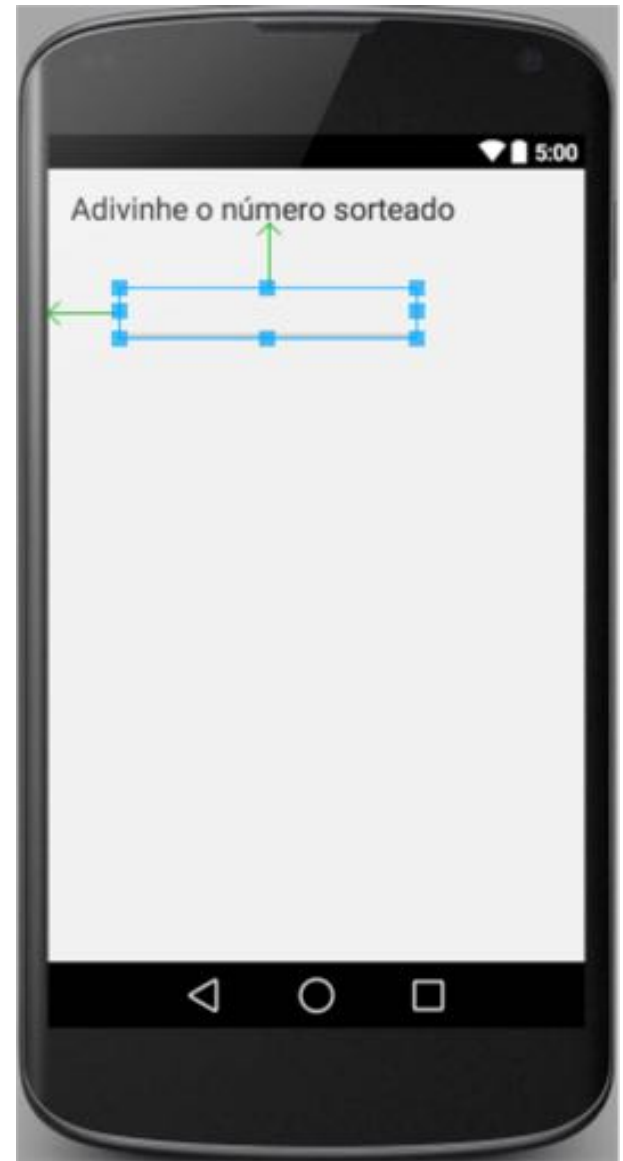


NumberGuess App

- Crie um novo projeto Android e nomeie-o NumberGuess.
- Escolha como target a API19.
- Selecione o template de Empty Activity.
- Nomeie a activity como MainActivity e o layout de activity_main
- Abra o arquivo de layout.
- Mude o layout para RelativeLayout.

NumberGuess App

- Remova o TextView do HelloWorld.
- Insira um TextView Large com o texto “Adivinhe o número sorteado”.
- Insira ainda um Text Field tipo Number para o usuário entrar com o número.
- Mude o id deste para input.



NumberGuess App

- Insira um botão para que possamos configurar a ação de “chute” do usuário.
- Mude o texto do botão para “Adivinhar”.
- No onClick do botão, chame o método guess.
- Por fim, insira um campo Text View de aparência média para apresentar a dica para o usuário.
- Deixe este campo sem texto e id = hint.



NumberGuess App

- Funcionamento

- Ao iniciar devemos sortear um número entre 1 e 100.
- A cada clique no botão devemos comparar o número que o usuário inseriu com o sorteado e informar:
 - Número sorteado é maior!
 - Número sorteado é menor!
 - Acertou. Você usou X tentativas.

NumberGuess App

- Comece criando os atributos int num, tries e um objeto Random r.
- No onCreate inicialize r e tries e sorteie o número aleatório.

```
public class MainActivity extends ActionBarActivity {  
  
    int num, tries;  
    Random r;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        r = new Random();  
        num = r.nextInt(100) + 1;  
        System.out.println(num);  
        tries = 0;  
    }  
}
```


NumberGuess App

- Crie o método guess, previamente vinculado ao clique do botão.
- Teste se existe algum tipo de dado no campo input do app.
 - Caso não haja, vamos mostrar um toast ao usuário.
 - Caso haja, prosseguimos com o código.
- Após o teste, se tudo estiver certo até o momento, devemos comparar o número sorteado com o fornecido e mostrar o resultado ao usuário.

NumberGuess App

```
public void guess (View view) {  
    EditText input = (EditText)findViewById(R.id.input);  
    TextView output = (TextView)findViewById(R.id.hint);  
  
    if (input.length() == 0) {  
        Toast msg = Toast.makeText(this, "Forneça um número!!!", Toast.LENGTH_LONG);  
        msg.show();  
    }  
    else {  
        String tip;  
        int inputNumber = Integer.parseInt(input.getText().toString());  
        tries++;  
        if (num > inputNumber)  
            tip = "Número sorteado é maior!";  
        else if (num < inputNumber)  
            tip = "Número sorteado é menor!";  
        else  
            tip = "Acertou! Você usou " + tries + " tentativas.";  
        output.setText(tip);  
    }  
}
```

NumberGuess App

Adivinhe o número sorteado

Adivinhar

Número sorteado é maior!

Forneça um número!!!

Adivinhe o número sorteado

10

Adivinhar

Número sorteado é maior!

Adivinhe o número sorteado

12

Adivinhar

Número sorteado é menor!

Adivinhe o número sorteado

11

Adivinhar

Acertou! Você usou 3 tentativas.