

Exercício 1

Neste exercício pretende-se a criação de um novo projeto com as seguintes características considerando as seguintes características:

- **Application Name:** FirstApp
- **Project Name:** FirstApp
- **Package Name:** pt.ipp.estg.firstApp
- **Minimum Required SDK:** API 23 Target SDK e Compile With: API 28

Com o projeto criado é pretendido a criação de um ecrã de aplicação semelhante à figura 1. Para o efeito deve usar o editor de layouts do android e escolher os componentes da paleta que melhor se adequem ao efeito desejado no layout na pasta res > layout > activity_main.xml.

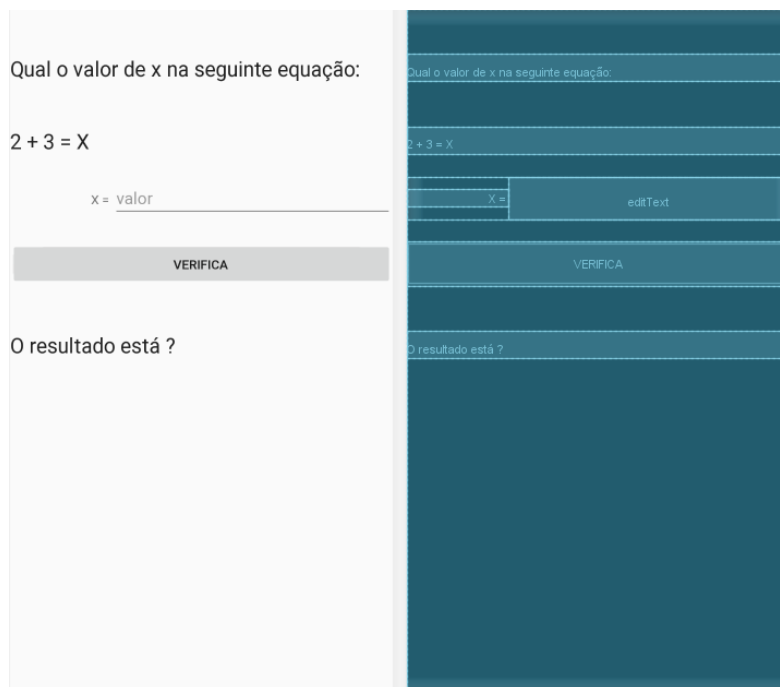


Figura 1 – Layout Android

Tendo em atenção ao enxerto de código 1 e 2, deve adicionar o código necessário para obter uma representação dos elementos layout TextView, EditText e Button como objetos. Estes excertos devem ser adicionados dentro do método onCreate() na atividade do projeto gerado.

```
TextView mResultado = findViewById(R.id.textView4);
EditText mEditText = findViewById(R.id.editText);
Button mButton = findViewById(R.id.button);
```

Script 1 - Obter referências de elementos do layout

```
mButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        mResultado.setText("O Resultado está correcto!");
    }
});
```

Script 2 - Adicionar comportamento a um botão em Android

Verificar que ao clicar no botão, o texto "O resultado está ?" é alterado para "O resultado está correcto!". Adicione o resto da lógica na aplicação de forma a validar se o input do utilizar está ou não correto de acordo com a expressão gerada.

Para efeitos de debug pode adicionar *prints* para o Logcat (consola de debug do Android) utilizando o método:

- `Log.d("TAG","Mensagem")`

O sistema de log é útil para obter feedback das ações executadas na aplicação. Implemente a função de log para que quando clicar no botão seja possível ver no logcat a mensagem tal como na figura 2.

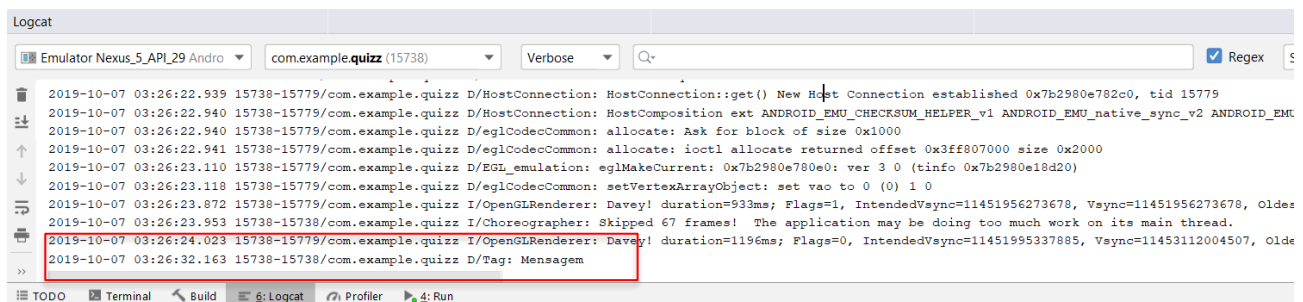


Figura 2 - Mensagem Logcat

Pode também fazer debug à aplicação usando o android studio. Para o efeito deve criar alguns breakpoints no código e clicar na função de debug, tal como na figura 3.

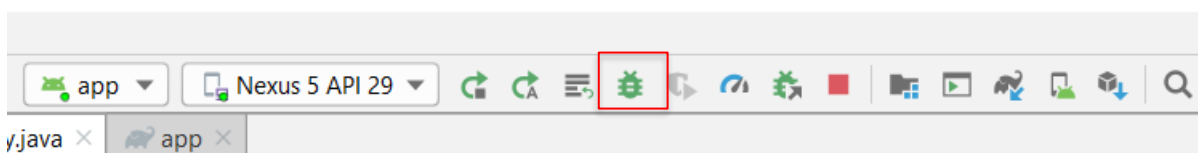


Figura 3 - Função de debug do android studio

De forma a tornar a nossa aplicação mais útil altere a lógica do programa de forma a que as perguntas sejam geradas de forma aleatória e dinâmica.

Crie um contador de respostas certas até à primeira resposta errada e de uma indicação de quantas respostas corretas seguidas o utilizador consegue alcançar.

Exercício 2

Neste exercício pretende-se a criação de um novo projeto com as seguintes características considerando as seguintes características:

- **Application Name:** Pool
- **Project Name:** poolApp
- **Package Name:** pt.ipp.estg.poolApp
- **Minimum Required SDK:** API 16 Target SDK e Compile With: API 28

Elementos capazes de gerar gráficos não estão por defeito disponíveis no editor de layouts do android studio. Contudo, é possível adicionar novos elementos ao editor de layouts por adição de dependências ao projeto na ferramenta gradle. Desta forma vamos adicionar uma biblioteca de gráficos ao projeto e adicionar um elemento gráfico na aplicação criada.

Para a adição da dependência, devemos editar o ficheiro gradle.build da nossa app e acrescentar a informação assinalada na figura 4.

```
repositories {  
    maven { url 'https://jitpack.io' }  
}  
  
dependencies {  
    implementation fileTree(dir: 'libs', include: ['*.jar'])  
    implementation 'androidx.appcompat:appcompat:1.0.2'  
    implementation 'androidx.constraintlayout:constraintlayout:1.1.3'  
    implementation 'com.github.PhilJay:MPAndroidChart:v3.1.0'  
  
    testImplementation 'junit:junit:4.12'  
  
    androidTestImplementation 'androidx.test.ext:junit:1.1.0'  
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.1.1'
```

Figura 4 - Dependências da biblioteca gráfica

No ficheiro de layouts da nossa aplicação na pasta res > layouts > activity_main.xml devemos na edição em modo texto adicionar uma view (elemento de layout em android) que caracteriza um gráfico gráfico line chart como no script 3. Neste exemplo está definido o layout completo, sendo o elemento responsável pelo gráfico dentro da tag com.github.mikephil.charting.charts.LineChart.

```
<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout  
    xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:app="http://schemas.android.com/apk/res-auto"  
    xmlns:tools="http://schemas.android.com/tools"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent">
```

```
<com.github.mikephil.charting.charts.LineChart
    android:id="@+id/chart"
    android:layout_width="match_parent"
    android:layout_height="200dp"
    android:layout_marginStart="16dp"
    android:layout_marginTop="16dp"
    android:layout_marginEnd="16dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.545"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<TextView
    android:id="@+id/textView"
    android:layout_width="89dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="48dp"
    android:text="Chart Sample!"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/chart" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

Script 3 - Elemento Line chart definido num layout em Android

Para preencher o gráfico é necessário adicionar pontos a um gráfico. Para adicionar um conjunto de 10 pontos aleatórios num gráfico pode utilizar o script 4 dentro do método onCreate() na actividade do seu projeto.

```
LineChart chart = (LineChart) findViewById(R.id.chart);

List<Entry> entries = new ArrayList<Entry>();
for (int i=0; i<10; i++) {
    entries.add(new Entry(i, (float) Math.random()));
}
LineDataSet dataSet = new LineDataSet(entries, "");
LineData lineData = new LineData(dataSet);

chart.setData(lineData);

chart.invalidate();
```

Script 4 - Preencher pontos num gráfico

Após estas alterações, deverá ser capaz de correr a aplicação e visualizar um gráfico com 10 pontos aleatórios, figura 5.

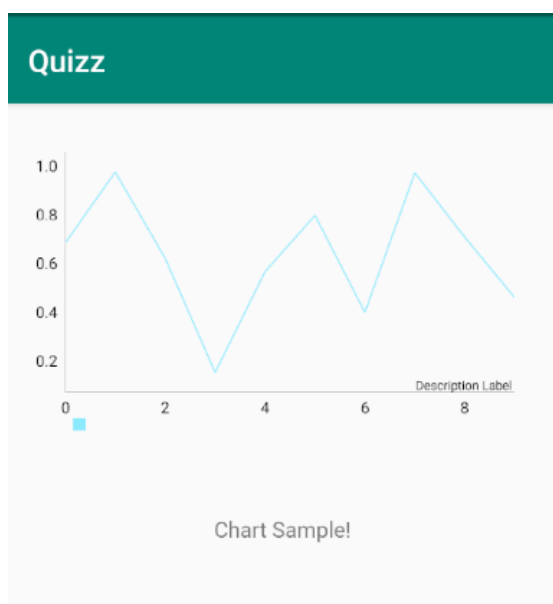


Figura 5 – Gráfico aleatório usando a biblioteca MPAndroidChart

Para melhor compreender a biblioteca gráfica utilizada, deve consultar a documentação oficial presente na página de internet: <https://weeklycoding.com/mpandroidchart/>.

Altere a aplicação de forma a ter um botão que quando carregado, gere um novo gráfico aleatório sem ser preciso sair da aplicação.

Explore outros gráficos da biblioteca, nomeadamente o PieChart e o BarChart.

Bibliografia Recomendada

Android developers - <https://developer.android.com/training/basics/firstapp>

MPAndroidChart - <https://weeklycoding.com/mpandroidchart/>