

Ficha do Aplicativo

Aplicativo Requisições HTTP

Resumo

Este aplicativo conecta-se a um servidor remoto através de requisições HTTP (Hiper Text Transfer Protocol).

Objetivos de Aprendizagem

1. Compreender os conceitos de conexão entre o aplicativo e um servidor remoto, utilizando requisições HTTP
2. Utilizar o conjunto de bibliotecas OkHTTP3
3. Utilizar a classe Picasso para carga e tratamento de imagens
4. Configurar dependências no Gradle
5. Liberar permissão de uso da internet no aplicativo, usando o AndroidManifest
6. Enviar requisições assíncronas a um servidor, obter e tratar a resposta a essas requisições
7. Carregar imagens a partir do Servidor, usando a classe Picasso
8. Carregar um Spinner a partir de dados obtidos no servidor, usando Adapter
9. Carregar um ListView a partir de dados nos formatos csv e json obtidos no servidor, usando Adapter

Dinâmica do Aplicativo

A tela inicial do aplicativo contém quatro botões que permitirão:

- a) Carregar uma página web a partir de uma url fornecida
- b) Carregar um arquivo texto disponível em um servidor remoto
- c) Carregar um spinner a partir das linhas de dados contidas em um arquivo texto no servidor. Essas linhas conterão nomes de imagens que poderão ser carregadas a partir da escolha de um item desse spinner
- d) Carregar um ListView a partir de um arquivo texto disponível no servidor, no formato CSV (Comma Separated Values). Este ListView será personalizado com a criação de um layout e um Adapter específicos.

Lista de Activities do Aplicativo

Nome	Layout
MainActivity.java	activity_main.xml
ObtemPaginaActivity.java	activity_obtem_pagina.xml
ObtemArqTextoActivity.java	activity_obtem_arqtexto.xml
ObtemImagemActivity.java	activity_obtem_imagem.xml
ObtemDadosActivity.java	activity_obtem_dados.xml
ObtemJsonActivity.java	Activity_obtem_json.xml

Drawables deste Aplicativo

Nome do arquivo	Dimensões	Tipo	Imagem
naocarregado.png	567 x 567	Imagem PNG	

Referências conceituais

Para um conteúdo sobre requisições HTTP acesse: <https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Messages>
<https://stackoverflow.com/questions/48555256/how-to-use-androidusescleartexttraffic-below-marshmallow-api-23>
<https://android-developers.googleblog.com/2016/04/protecting-against-unintentional.html>
<https://developer.android.com/guide/topics/manifest/application-element>

Para saber mais sobre a biblioteca de Classes OkHttp acesse: <https://square.github.io/okhttp>

Para saber mais sobre a biblioteca de Classes Picasso acesse: <https://square.github.io/picasso>

Para começar este aplicativo faça a criação de um novo projeto, a partir do zero. Em seguida faça as duas tarefas iniciais indicadas a seguir.

Tarefas iniciais – configuração de dependências para uso de Classes de terceiros

Neste aplicativo serão usadas classes feitas por terceiros. Isso significa que serão utilizadas classes que não são feitas pelo Google e, por consequência, não são automaticamente instaladas junto com o Android Studio.

No caso deste app vamos utilizar dois pacotes de classes

- OkHttp: contém os recursos necessários para executar requisições HTTP em um servidor disponível na internet;
- Picasso: contém os recursos necessários para buscar e carregar imagens a partir de endereços na internet;

Para utilizar classes assim é preciso abrir o projeto e configurar o que se costuma chamar de "dependências". Assim, precisamos informar ao Android Studio que essas dependências são necessárias e isso é feito no arquivo `build.gradle.kts` do módulo. Esse arquivo fica na pasta `{nome_do_projeto}.app`

Localize esse arquivo na pasta indicada e abra-o. Na seção dependencies acrescente estas duas linhas a seguir.

```
implementation("com.squareup.okhttp3:okhttp:4.12.0")
implementation("com.squareup.picasso:picasso:2.8")
```

Build.gradle.kts

```
plugins {
    alias(libs.plugins.android.application)
}

android {
    namespace = "com.ilp506.requisicoeshttp"
    compileSdk = 34

    defaultConfig {
        applicationId = "com.ilp506.requisicoeshttp"
        minSdk = 27
        targetSdk = 34
        versionCode = 1
        versionName = "1.0"

        testInstrumentationRunner = "androidx.test.runner.AndroidJUnitRunner"
    }

    buildTypes {
        release {
            isMinifyEnabled = false
            proguardFiles(
                getDefaultProguardFile("proguard-android-optimize.txt"),
                "proguard-rules.pro"
            )
        }
    }

    compileOptions {
        sourceCompatibility = JavaVersion.VERSION_1_8
        targetCompatibility = JavaVersion.VERSION_1_8
    }
}

dependencies {
    implementation(libs.appcompat)
    implementation(libs.material)
    implementation(libs.activity)
    implementation(libs.constraintlayout)

    implementation("com.squareup.okhttp3:okhttp:4.12.0")
    implementation("com.squareup.picasso:picasso:2.8")

    testImplementation(libs.junit)
    androidTestImplementation(libs.ext.junit)
    androidTestImplementation(libs.espresso.core)
}
```

Linhas a serem acrescentadas

Tarefas iniciais – configuração de permissão para uso de internet

Como este aplicativo fará requisições a um servidor web usando o protocolo HTTP, será necessário configurar a permissão de internet. Essa é uma tarefa bastante simples: basta abrir o arquivo `AndroidManifest` do aplicativo e incluir a linha a seguir no topo, antes da abertura da tag `<application ...`

```
<uses-permission android:name="android.permission.INTERNET" />
```

Resources

strings.xml

```
<resources>
  <string name="app_name">Requisições HTTP</string>
  <string name="btnpagina">Obter Página</string>
  <string name="btntexto">Obter Texto</string>
  <string name="btnimagem">Obter Imagem</string>
  <string name="btndados">Obter Dados</string>
  <string name="btnexecreqpag">Executa Requisição - Carrega página</string>
  <string name="btnexecreqtxt">Executa Requisição - Obtém texto</string>
  <string name="btnexecreqimg">Executa Requisição - Monta Spinner</string>
  <string name="btnexecreqdados">Executa Requisição - Carrega Dados</string>

  <string name="rotulo_reqpag">Página obtida</string>
  <string name="rotulo_reqtxt">Texto obtido do servidor</string>
  <string name="rotulo_reqimg">Spinner carregado a partir do servidor</string>
  <string-array name="vazio">
    <item>...</item>
  </string-array>
</resources>
```

colors.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <color name="black">#FF000000</color>
  <color name="white">#FFFFFFFF</color>
  <color name="cor_primaria">#0180B0</color>
  <color name="cor_fundo">#ACD7FF</color>
  <color name="cor_cxtexto">#D5EBFF</color>
  <color name="cor_texto">#2E0D70</color>
</resources>
```

Detalhamento das Activities

Layout: activity_main.xml

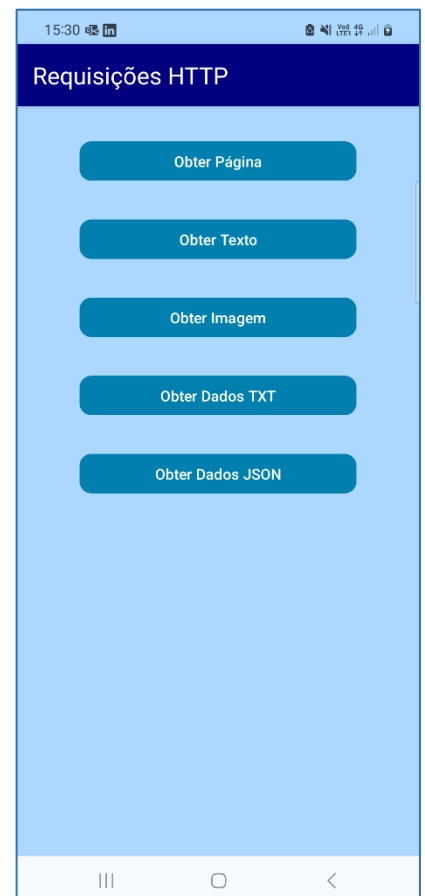
```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/cor_fundo"
    tools:context=".MainActivity">

  <Button
      android:id="@+id/btnObterPagina"
      android:layout_width="0dp"
      android:layout_height="wrap_content"
      android:layout_marginStart="64dp"
      android:layout_marginTop="32dp"
      android:layout_marginEnd="64dp"
      android:text="@string/btnpagina"
      app:cornerRadius="12dp"
      app:layout_constraintEnd_toEndOf="parent"
      app:layout_constraintHorizontal_bias="1.0"
      app:layout_constraintStart_toStartOf="parent"
      app:layout_constraintTop_toTopOf="parent" />

  <Button
      android:id="@+id/btnObterArqTexto"
      android:layout_width="0dp"
      android:layout_height="wrap_content"
      android:layout_marginStart="64dp"
      android:layout_marginTop="32dp"
      android:layout_marginEnd="64dp"
      android:text="@string/btntexto"
      app:cornerRadius="12dp"
      app:layout_constraintEnd_toEndOf="parent"
      app:layout_constraintHorizontal_bias="1.0"
      app:layout_constraintStart_toStartOf="parent"
      app:layout_constraintTop_toBottomOf="@+id/btnObterPagina" />

  <Button
      android:id="@+id/btnObterImagem"
      android:layout_width="0dp"
      android:layout_height="wrap_content"

  </resources>
```



```

        android:layout_marginStart="64dp"
        android:layout_marginTop="32dp"
        android:layout_marginEnd="64dp"
        android:text="@string/btnimagem"
        app:cornerRadius="12dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="1.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/btnObterArqTexto" />

<Button
    android:id="@+id/btnObterDadosTxt"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="64dp"
    android:layout_marginTop="32dp"
    android:layout_marginEnd="64dp"
    android:text="@string/btndadostxt"
    app:cornerRadius="12dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="1.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/btnObterImagem" />

<Button
    android:id="@+id/btnObterDadosJson"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="64dp"
    android:layout_marginTop="32dp"
    android:layout_marginEnd="64dp"
    android:text="@string/btndadosjson"
    app:cornerRadius="12dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="1.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/btnObterDadosTxt" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

Código: MainActivity.java

```

public class MainActivity extends AppCompatActivity {

    public static final String URL_RAIZ = "https://www.clictec.com.br/iperanga/reqhttp/";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main);
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
            return insets;
        });

        ActionBar actionBar = getSupportActionBar();
        ColorDrawable colorDrawable = new ColorDrawable(Color.parseColor("#000080"));
        actionBar.setBackgroundDrawable(colorDrawable);
        actionBar.setTitle(Html.fromHtml("<font color='#FFFFFF'>Requisições HTTP</font>", 0));

        setBtnsOnClick();
    }

    private void setBtnsOnClick() {
        Button btn;
        View.OnClickListener listener = new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                trataBtnOnClick(view);
            }
        };
        btn = findViewById(R.id.btnObterPagina);
        btn.setOnClickListener(listener);
        btn = findViewById(R.id.btnObterArqTexto);
        btn.setOnClickListener(listener);
        btn = findViewById(R.id.btnObterImagem);
        btn.setOnClickListener(listener);
        btn = findViewById(R.id.btnObterDadosTxt);
        btn.setOnClickListener(listener);
        btn = findViewById(R.id.btnObterDadosJson);
    }
}

```

```

        btn.setOnClickListener(listener);
    }

    private void trataBtnOnClick(View view) {
        if (view.getId() == R.id.btnObterPagina) {
            startActivity(new Intent(MainActivity.this, ObtemPaginaActivity.class));
        }
        else if (view.getId() == R.id.btnObterArqTexto) {
            startActivity(new Intent(MainActivity.this, ObtemArqTextoActivity.class));
        }
        else if (view.getId() == R.id.btnObterImagem) {
            startActivity(new Intent(MainActivity.this, ObtemImagemActivity.class));
        }
        else if (view.getId() == R.id.btnObterDadosTxt) {
            startActivity(new Intent(MainActivity.this, ObtemDadosActivity.class));
        }
        else if (view.getId() == R.id.btnObterDadosJson) {
            startActivity(new Intent(MainActivity.this, ObtemJsonActivity.class));
        }
    }
}

```

Layout: activity_obtem_pagina.xml

```

<?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:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/cor_fundo"
    tools:context=".ObtemPaginaActivity">

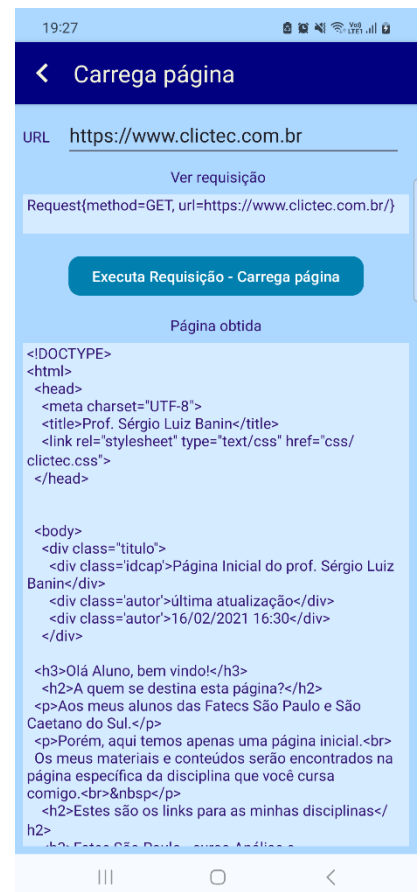
    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:text="URL"
        android:textColor="@color/cor_texto"
        app:layout_constraintBaseline_toBaselineOf="@+id/edtURL"
        app:layout_constraintStart_toStartOf="parent" />

    <EditText
        android:id="@+id/edtURL"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginTop="8dp"
        android:layout_marginEnd="16dp"
        android:ems="10"
        android:hint="Nome do Arquivo"
        android:inputType="text"
        android:text="https://www.clictec.com.br"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toEndOf="@+id/textView1"
        app:layout_constraintTop_toTopOf="parent"
        tools:text="https://www.clictec.com.br" />

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="8dp"
        android:text="Ver requisição"
        android:textColor="@color/cor_texto"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.507"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/edtURL" />

    <TextView
        android:id="@+id/txtVerRequisicao"
        android:layout_width="0dp"
        android:layout_height="40dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:layout_marginEnd="8dp"
        android:background="@color/cor_cxtexto"
        android:paddingStart="4dp"
        android:paddingEnd="4dp"

```



```

        android:textColor="@color/cor_texto"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView2"
        tools:text="txtVerRequisicao" />

<Button
    android:id="@+id/btnCarregaPagina"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="96dp"
    android:text="@string/btnexcreqpag"
    app:cornerRadius="12dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/edtURL" />

<TextView
    android:id="@+id/textView3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:text="@string/rotulo_reqpag"
    android:textColor="@color/cor_texto"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/btnCarregaPagina" />

<TextView
    android:id="@+id/txtVerPagina"
    android:layout_width="0dp"
    android:layout_height="0dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginBottom="8dp"
    android:background="@color/cor_cxtexto"
    android:paddingStart="4dp"
    android:paddingEnd="4dp"
    android:textColor="@color/cor_texto"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView3"
    tools:text="txtVerPagina" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

Código: ObtemPaginaActivity.java

```

public class ObtemPaginaActivity extends AppCompatActivity {

    private OkHttpClient reqHTTP;
    private EditText edtUrl;
    private TextView txtVerRequisicao;
    private TextView txtVerPagina;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_obtem_pagina);
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
            return insets;
        });

        ActionBar actionBar = getSupportActionBar();
        ColorDrawable colorDrawable = new ColorDrawable(Color.parseColor("#000080"));
        actionBar.setBackgroundDrawable(colorDrawable);
        actionBar.setTitle(Html.fromHtml("<font color='#F5F5DC'>Carrega página</font>", 0));
        actionBar.setHomeAsUpIndicator(R.drawable.ic_voltar);
        actionBar.setDisplayHomeAsUpEnabled(true);

        reqHTTP = new OkHttpClient();
        edtUrl = findViewById(R.id.edtURL);
        txtVerRequisicao = findViewById(R.id.txtVerRequisicao);
        configBtnCarregaPagina();
        configTxtVerPagina();
        configBtnCarregaPagina();
        configTxtVerPagina();
    }
}

```

```

    }

    @Override
    public boolean onOptionsItemSelected(@NonNull MenuItem item) {
        if (item.getItemId() == android.R.id.home) {
            finish();
            return true;
        }
        else
            return super.onOptionsItemSelected(item);
    }

    private void configBtnCarregaPagina() {
        Button btn = findViewById(R.id.btnCarregaPagina);
        btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                executaRequisicao();
            }
        });
    }

    private void configTxtVerPagina() {
        txtVerPagina = findViewById(R.id.txtVerPagina);
        txtVerPagina.setMovementMethod(new ScrollingMovementMethod());
        txtVerPagina.setOnLongClickListener(new View.OnLongClickListener() {
            @Override
            public boolean onLongClick(View view) {
                txtVerPagina.setText("");
                return false;
            }
        });
    }

    private void executaRequisicao() {
        String url = edtUrl.getText().toString().toLowerCase();
        Request request = new Request.Builder()
            .url(url)
            .build();
        txtVerRequisicao.setText(request.toString());
        Call call = reqHTTP.newCall(request);
        call.enqueue(new Callback() {
            @Override
            public void onFailure(@NonNull Call call, @NonNull IOException e) {
                runOnUiThread(new Runnable() {
                    @Override
                    public void run() {
                        txtVerPagina.setText("Falha de Conexão\n" + e.getMessage());
                    }
                });
            }

            @Override
            public void onResponse(@NonNull Call call, @NonNull Response response) throws IOException {
                runOnUiThread(new Runnable() {
                    @Override
                    public void run() {
                        try {
                            txtVerPagina.setText(response.body().string());
                            txtVerPagina.setScrollY(0);
                        }
                        catch (IOException e) {
                            txtVerPagina.setText("Erro no tratamento da mensagem - " + e.getMessage());
                        }
                    }
                });
            }
        });
    }
}

```

Layout: activity_obtem_arqtexto.xml

```
<?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:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/cor_fundo"
    tools:context=".ObtemArqTextoActivity">

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:text="Qual arquivo carregar?"
        android:textColor="@color/cor_texto"
        app:layout_constraintBaseline_toBaselineOf="@+id/edtArqTexto"
        app:layout_constraintStart_toStartOf="parent" />

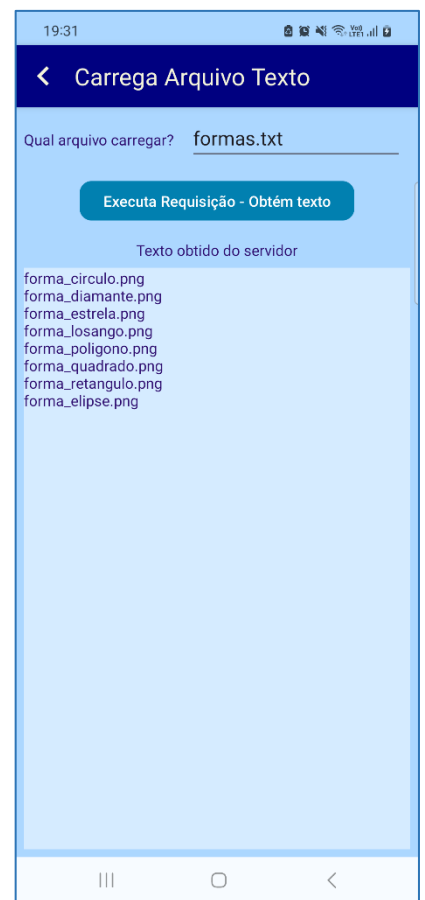
    <EditText
        android:id="@+id/edtArqTexto"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginTop="8dp"
        android:layout_marginEnd="16dp"
        android:ems="10"
        android:hint="Nome do Arquivo"
        android:inputType="text"
        android:text="formas.txt"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toEndOf="@+id/textView1"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/btnObtemArqTexto"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:text="@string/btnexcreqtxt"
        app:cornerRadius="12dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/edtArqTexto" />

    <TextView
        android:id="@+id/textView5"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:text="@string/rotulo_reqtxt"
        android:textColor="@color/cor_texto"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/btnObtemArqTexto" />

    <TextView
        android:id="@+id/txtTextoObtido"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:text="@string/txtTextoObtido"
        android:textColor="@color/cor_texto"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView5" />

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



```

android:layout_height="0dp"
android:layout_marginStart="8dp"
android:layout_marginTop="8dp"
android:layout_marginEnd="8dp"
android:layout_marginBottom="8dp"
android:background="@color/cor_cxtexto"
android:text="TextView"
android:textColor="@color/cor_texto"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/textView5" />

```

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

Código: ObtemArqTextoActivity.java

```

public class ObtemArqTextoActivity extends AppCompatActivity {

    private OkHttpClient reqHTTP;
    private EditText edtArqTexto;
    private TextView txtTextoObtido;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_obtem_arq_texto);
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
            return insets;
        });

        ActionBar actionBar = getSupportActionBar();
        ColorDrawable colorDrawable = new ColorDrawable(Color.parseColor("#000080"));
        actionBar.setBackgroundDrawable(colorDrawable);
        actionBar.setTitle(Html.fromHtml("<font color='#F5F5DC'>Carrega Arquivo Texto</font>", 0));
        actionBar.setHomeAsUpIndicator(R.drawable.ic_voltar);
        actionBar.setDisplayHomeAsUpEnabled(true);

        reqHTTP = new OkHttpClient();
        edtArqTexto = findViewById(R.id.edtArqTexto);
        configBtnObtemArqTexto();
        configTxtTextoObtido();
    }

    @Override
    public boolean onOptionsItemSelected(@NonNull MenuItem item) {
        if (item.getItemId() == android.R.id.home) {
            finish();
            return true;
        }
        else
            return super.onOptionsItemSelected(item);
    }

    private void configBtnObtemArqTexto() {
        Button btn = findViewById(R.id.btnObtemArqTexto);
        btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                executaRequisicao();
            }
        });
    }

    private void configTxtTextoObtido() {
        txtTextoObtido = findViewById(R.id.txtTextoObtido);
        txtTextoObtido.setMovementMethod(new ScrollingMovementMethod());
        txtTextoObtido.setOnLongClickListener(new View.OnLongClickListener() {
            @Override
            public boolean onLongClick(View view) {
                txtTextoObtido.setText("");
                return false;
            }
        });
    }
}

```

```

    });
}

private void executaRequisicao() {
    String url = MainActivity.URL_RAIZ + edtArqTexto.getText().toString().toLowerCase();
    Request request = new Request.Builder()
        .url(url)
        .build();

    Call call = reqHTTP.newCall(request);
    call.enqueue(new Callback() {
        @Override
        public void onFailure(@NotNull Call call, @NotNull IOException e) {
            runOnUiThread(new Runnable() {
                @Override
                public void run() {
                    txtTextoObtido.setText("Falha de conexão - " + e.getMessage());
                }
            });
        }

        @Override
        public void onResponse(@NotNull Call call, @NotNull Response response) throws IOException {
            runOnUiThread(new Runnable() {
                @Override
                public void run() {
                    try {
                        txtTextoObtido.setText(response.body().string());
                    } catch (IOException e) {
                        txtTextoObtido.setText("Erro no tratamento da mensagem - " + e.getMessage());
                    }
                }
            });
        }
    });
}
}
}

```

Layout: activity_obtem_imagem.xml

```

<?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:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/cor_fundo"
    tools:context=".ObtemImagemActivity">

    <TextView
        android:id="@+id/textView6"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:text="Qual arquivo carregar?"
        android:textColor="@color/cor_texto"
        app:layout_constraintBaseline_toBaselineOf="@+id/edtNomeImagens"
        app:layout_constraintStart_toStartOf="parent" />

    <EditText
        android:id="@+id/edtNomeImagens"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginTop="8dp"
        android:layout_marginEnd="16dp"
        android:ems="10"
        android:hint="Nome do Arquivo"
        android:inputType="text"
        android:text="formas.txt"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toEndOf="@+id/textView6"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/btnExecutaRequisicao"
        android:layout_width="wrap_content"

```



```

        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:text="@string/btnexcreqimg"
        app:cornerRadius="12dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/edtNomeImagens" />

<TextView
    android:id="@+id/textView7"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:text="@string/rotulo_reqimg"
    android:textColor="@color/cor_texto"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/btnExecutaRequisicao" />

<Spinner
    android:id="@+id/spnImagens"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="16dp"
    android:background="@color/cor_cxtexto"
    android:entries="@array/vazio"
    android:paddingTop="12dp"
    android:paddingBottom="12dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView7" />

<ImageView
    android:id="@+id/imgCarregada"
    android:layout_width="0dp"
    android:layout_height="0dp"
    android:layout_marginStart="16dp"
    android:layout_marginTop="16dp"
    android:layout_marginEnd="16dp"
    android:layout_marginBottom="16dp"
    app:layout_constraintBottom_toTopOf="@+id/txtStatus"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/spnImagens"
    app:srcCompat="@drawable/naocarregado" />

<TextView
    android:id="@+id/txtStatus"
    android:layout_width="0dp"
    android:layout_height="32dp"
    android:layout_marginStart="16dp"
    android:layout_marginEnd="16dp"
    android:layout_marginBottom="4dp"
    android:textSize="12sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

Código: ObtemImagemActivity.java

```

public class ObtemImagemActivity extends AppCompatActivity {

    private OkHttpClient reqHttp;

    private EditText edtNomeImagens;
    private TextView txtStatus;
    private Spinner spnImagens;
    private ImageView imgCarregada;

    private List<String> listaNomesImgs;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_obtem_imagem);
    }
}

```

```

ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
    Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
    v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
    return insets;
});

ActionBar actionBar = getSupportActionBar();
ColorDrawable colorDrawable = new ColorDrawable(Color.parseColor("#000080"));
actionBar.setBackgroundDrawable(colorDrawable);
actionBar.setTitle(Html.fromHtml("<font color='#F5F5DC'>Carrega Imagem</font>", 0));
actionBar.setHomeAsUpIndicator(R.drawable.ic_voltar);
actionBar.setDisplayHomeAsUpEnabled(true);

reqHttp = new OkHttpClient();
edtNomeImagens = findViewById(R.id.edtNomeImagens);
txtStatus = findViewById(R.id.txtStatus);
spnImagens = findViewById(R.id.spnImagens);
imgCarregada = findViewById(R.id.imgCarregada);

configBtnExecutaRequisicao();
}

@Override
public boolean onOptionsItemSelected(@NonNull MenuItem item) {
    if (item.getItemId() == android.R.id.home) {
        finish();
        return true;
    }
    else
        return super.onOptionsItemSelected(item);
}

private void configBtnExecutaRequisicao() {
    Button btn = findViewById(R.id.btnExecutaRequisicao);
    btn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            executaRequisicao();
        }
    });
}

private void executaRequisicao() {
    String url = MainActivity.URL_RAIZ + edtNomeImagens.getText().toString().toLowerCase();
    Request request = new Request.Builder()
        .url(url)
        .build();
    Call call = reqHttp.newCall(request);
    call.enqueue(new Callback() {
        @Override
        public void onFailure(@NotNull Call call, @NotNull IOException e) {
            runOnUiThread(new Runnable() {
                @Override
                public void run() {
                    txtStatus.setText("Falha de conexão - " + e.getMessage());
                }
            });
        }

        @Override
        public void onResponse(@NotNull Call call, @NotNull Response response) throws IOException {
            runOnUiThread(new Runnable() {
                @Override
                public void run() {
                    try {
                        montaSpnImagens(response.body().string());
                    }
                    catch (IOException e) {
                        txtStatus.setText("Falha durante carga dos dados.");
                    }
                }
            });
        }
    });
}

private void montaSpnImagens(String nomesImgs) {
    //this.nomesImgs = nomesImgs;
    txtStatus.setText(nomesImgs);
    listaNomesImgs = Arrays.asList(nomesImgs.split("\n"));
    ArrayAdapter<String> adaptador = new ArrayAdapter<>(
        this,
        android.R.layout.simple_spinner_item,

```

```

        listaNomesImgs
    );
    spnImagens.setAdapter(adaptador);
    setSpnImagensOnItemSelected();
}

private void setSpnImagensOnItemSelected() {
    spnImagens.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
        @Override
        public void onItemSelected(AdapterView<?> adapterView, View view, int position, long l) {
            carregaImagem(position);
        }

        @Override
        public void onNothingSelected(AdapterView<?> adapterView) {
            imgCarregada.setImageResource(R.drawable.nao_carregado);
            txtStatus.setText("");
        }
    });
}

private void carregaImagem(int i) {
    String nomeImagem = listaNomesImgs.get(i);
    String url = MainActivity.URL_RAIZ + nomeImagem;
    txtStatus.setText(url);
    Picasso.get()
        .load(url)
        .memoryPolicy(MemoryPolicy.NO_CACHE)
        .networkPolicy(NetworkPolicy.NO_CACHE)
        .into(imgCarregada);
}
}
}

```

Layout: activity_obtem_dados.xml

```

<?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:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/cor_fundo"
    tools:context=".ObtemDadosActivity">

    <Button
        android:id="@+id/btnCarregaDados"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="8dp"
        android:text="@string/btnexecreqdados"
        app:cornerRadius="12dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <ListView
        android:id="@+id/lvDados"
        android:layout_width="0dp"
        android:layout_height="0dp"
        android:layout_marginTop="8dp"
        android:layout_marginBottom="8dp"
        android:background="@color/cor_cxtexto"
        app:layout_constraintBottom_toTopOf="@+id/txtStatus"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/btnCarregaDados" />

    <TextView
        android:id="@+id/txtStatus"
        android:layout_width="0dp"
        android:layout_height="32dp"
        android:layout_marginStart="16dp"
        android:layout_marginEnd="16dp"
        android:layout_marginBottom="4dp"
        android:textSize="12sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent" />

```

19:37	
< Carrega Dados	
Executa Requisição - Carrega Dados	
1010 Jabaquara	Sobrado Geminado Valor venda: R\$ 430.000,00 Valor aluguel: R\$ 2.350,00 Área terreno: 75.6 e Área construída: 112.4 Dormitórios: 4 - Suítes: 1 - Vagas: 2
1020-Vendido Butantã	Casa Térrea não disponível para venda Valor aluguel: R\$ 2.500,00 Área terreno: 146.3 e Área construída: 110.0 Dormitórios: 3 - Suítes: 1 - Vagas: 2
1030 Pinheiros	Apartamento Valor venda: R\$ 550.000,00 não disponível para aluguel Área terreno: 0.0 e Área construída: 93.2 Dormitórios: 2 - Suítes: 2 - Vagas: 1
1040 Vila Maria	Terreno Valor venda: R\$ 320.000,00 não disponível para aluguel Área terreno: 230.0 e Área construída: 0.0 Dormitórios: 0 - Suítes: 0 - Vagas: 0
1050 Lapa	Apartamento Valor venda: R\$ 450.000,00 Valor aluguel: R\$ 1.890,00 Área terreno: 0.0 e Área construída: 112.0 Dormitórios: 3 - Suítes: 3 - Vagas: 1
1060 Jacañã	Apartamento Valor venda: R\$ 250.000,00
○ <	

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

Código: ObtemDadosActivity.java

```
public class ObtemDadosActivity extends AppCompatActivity {

    private OkHttpClient reqHttp;
    private ListView lvDados;
    private TextView txtStatus;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_obtem_dados);
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
            return insets;
        });
        ActionBar actionBar = getSupportActionBar();
        ColorDrawable colorDrawable = new ColorDrawable(Color.parseColor("#000080"));
        actionBar.setBackgroundDrawable(colorDrawable);
        actionBar.setTitle(Html.fromHtml("<font color='#F5F5DC'>Carrega Dados</font>", 0));
        actionBar.setHomeAsUpIndicator(R.drawable.ic_voltar);
        actionBar.setDisplayHomeAsUpEnabled(true);
        reqHttp = new OkHttpClient();
        lvDados = findViewById(R.id.lvDados);
        txtStatus = findViewById(R.id.txtStatus);
        configBtnCarregaDados();
    }

    @Override
    public boolean onOptionsItemSelected(@NonNull MenuItem item) {
        if (item.getItemId() == android.R.id.home) {
            finish();
            return true;
        }
        else
            return super.onOptionsItemSelected(item);
    }

    private void configBtnCarregaDados() {
        Button btn = findViewById(R.id.btnCarregaDados);
        btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                carregaDados();
            }
        });
    }

    private void carregaDados() {
        String url = MainActivity.URL_RAIZ + "dados.txt";
        Request request = new Request.Builder()
            .url(url)
            .build();
        Call call = reqHttp.newCall(request);
        call.enqueue(new Callback() {
            @Override
            public void onFailure(@NonNull Call call, @NonNull IOException e) {
                runOnUiThread(new Runnable() {
                    @Override
                    public void run() {
                        txtStatus.setText("Falha de conexão - " + e.getMessage());
                    }
                });
            }

            @Override
            public void onResponse(@NonNull Call call, @NonNull Response response) throws IOException {
                runOnUiThread(new Runnable() {
                    @Override
                    public void run() {
                        try {
                            exibeDados(response.body().string());
                        }
                        catch (IOException e) {
                            txtStatus.setText("Falha durante carga dos dados.");
                        }
                    }
                });
            }
        });
    }
}
```

```

    });
}
});
}

private void exibeDados(String dados) {
    List<String> ls;
    ls = Arrays.asList(dados.split("\n"));
    DadosAdapter dadosAdapter = new DadosAdapter(this, ls);
    lvDados.setAdapter(dadosAdapter);
}
}

```

Layout: activity_obtem_json.xml (o visual da tela é igual ao anterior, porém o id dos componentes é diferente)

```

<?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:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/cor_fundo"
    tools:context=".ObtemJsonActivity">

    <Button
        android:id="@+id/btnCarregaJson"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="8dp"
        android:text="@string/btnexecreqjson"
        app:cornerRadius="12dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <ListView
        android:id="@+id/lvDadosJson"
        android:layout_width="0dp"
        android:layout_height="0dp"
        android:layout_marginTop="8dp"
        android:layout_marginBottom="8dp"
        android:background="@color/cor_cxtexto"
        app:layout_constraintBottom_toTopOf="@+id/txtStatusJson"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/btnCarregaJson" />

    <TextView
        android:id="@+id/txtStatusJson"
        android:layout_width="0dp"
        android:layout_height="32dp"
        android:layout_marginStart="16dp"
        android:layout_marginEnd="16dp"
        android:layout_marginBottom="4dp"
        android:textSize="12sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

15:20	
< Carrega Dados	
Executa Requisição - Carrega JSON	
1010 Jabaquara	Sobrado Geminado Valor venda: R\$ 430.000,00 Valor venda: R\$ 2.350,00 Área terreno: 75.6 e Área construída: 112.4 Dormitórios: 4 - Suítes: 1 - Vagas: 2
1015 Freguesia do Ó	Casa de Fundos não disponível para venda Valor venda: R\$ 853,29 Área terreno: 87.9 e Área construída: 53.2 Dormitórios: 1 - Suítes: 0 - Vagas: 0
1020 Butantã	Casa Térrea não disponível para venda Valor venda: R\$ 2.500,00 Área terreno: 146.3 e Área construída: 110.0 Dormitórios: 3 - Suítes: 1 - Vagas: 2
1030 Pinheiros	Apartamento Valor venda: R\$ 550.000,00 não disponível para aluguel Área terreno: 100.0 e Área construída: 93.2 Dormitórios: 2 - Suítes: 2 - Vagas: 1
1040 Vila Maria	areat Valor venda: R\$ 320.000,00 não disponível para aluguel Área terreno: 230.0 e Área construída: 0.0 Dormitórios: 0 - Suítes: 0 - Vagas: 0
1050	Apartamento

Código: ObtemDadosActivity.java

```

public class ObtemJsonActivity extends AppCompatActivity {

    private OkHttpClient reqHttp;

    private ListView lvDadosJson;
    private TextView txtStatusJson;
    private ArrayList<Imovel> lstImoveis;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_obtem_json);
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
            return insets;
        });

        ActionBar actionBar = getSupportActionBar();
        ColorDrawable colorDrawable = new ColorDrawable(Color.parseColor("#000080"));
        actionBar.setBackgroundDrawable(colorDrawable);
        actionBar.setTitle(Html.fromHtml("<font color='#F5F5DC'>Carrega Dados</font>", 0));
        actionBar.setHomeAsUpIndicator(R.drawable.ic_voltar);
        actionBar.setDisplayHomeAsUpEnabled(true);

        reqHttp = new OkHttpClient();
        lvDadosJson = findViewById(R.id.lvDadosJson);
        txtStatusJson = findViewById(R.id.txtStatusJson);
        txtStatusJson.setMovementMethod(new ScrollingMovementMethod());
        configBtnCarregaDados();
    }

    @Override
    public boolean onOptionsItemSelected(@NonNull MenuItem item) {
        if (item.getItemId() == android.R.id.home) {
            finish();
            return true;
        }
        else
            return super.onOptionsItemSelected(item);
    }

    private void configBtnCarregaDados() {
        Button btn = findViewById(R.id.btnCarregaJson);
        btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                carregaDados();
            }
        });
    }

    private void carregaDados() {
        String url = MainActivity.URL_RAIZ + "dados_array.json";
        Request request = new Request.Builder()
            .url(url)
            .build();
        Call call = reqHttp.newCall(request);
        call.enqueue(new Callback() {
            @Override
            public void onFailure(@NotNull Call call, @NotNull IOException e) {
                runOnUiThread(new Runnable() {
                    @Override
                    public void run() {
                        txtStatusJson.setText("Falha de conexão - " + e.getMessage());
                    }
                });
            }

            @Override
            public void onResponse(@NotNull Call call, @NotNull Response response) throws IOException {
                runOnUiThread(new Runnable() {
                    @Override
                    public void run() {
                        try {
                            obtemImoveis(response.body().string());
                            exibeImoveis();
                        }
                        catch (IOException e) {
                            txtStatusJson.setText("Falha durante carga dos dados.");
                        }
                    }
                });
            }
        });
    }
}

```

```

    });
}
});
}

// Esta versão do método obterImoveis usa JSON contendo um array de elementos
private void obterImoveis(String json) {
    JSONObject objJSON; // objeto usado para acessar e manipular o JSON completo conforme recebido do
servidor
    JSONArray arrayJSON; // array contendo os vários imóveis retornados no objeto JSON acima
    JSONObject jsonImovel; // objeto usado para isolar um imóvel específico a partir do array arrayJSON
    Imovel objImovel; // objeto da nossa classe específica - Imovel - para carregar no ArrayList
que será usado para alimentar o ListView lvDadosJson
    lstImoveis = new ArrayList<>();
    try {
        objJSON = new JSONObject(json); // Esta opção é para quando o Json é um array nomeado, ou
seja, com este formato { "imoveis": [ ...conteúdo do array... ] }
        arrayJSON = objJSON.getJSONArray("imoveis");
        arrayJSON = new JSONArray(json); // Esta opção pode ser usada quando o Json é um array -
iniciando com [ e terminando com ] (abre e fecha colchetes)
        for (int i = 0; i < arrayJSON.length(); i++) {
            jsonImovel = arrayJSON.getJSONObject(i);
            objImovel = new Imovel(
                jsonImovel.getString("_id"),
                jsonImovel.getString("bairro"),
                jsonImovel.getString("tipo"),
                jsonImovel.getString("areat"),
                jsonImovel.getString("areac"),
                Integer.parseInt(jsonImovel.getString("dorms")),
                Integer.parseInt(jsonImovel.getString("suites")),
                Integer.parseInt(jsonImovel.getString("vagas")),
                Double.parseDouble(jsonImovel.getString("vlrvenda")),
                Double.parseDouble(jsonImovel.getString("vlraluga"))
            );
            lstImoveis.add(objImovel);
        }
    } catch (JSONException e) {
        txtStatusJson.setText(e.getMessage());
    }
}

private void exibeImoveis() {
    DadosJsonAdapter jsonAdapter = new DadosJsonAdapter(this, lstImoveis);
    lvDadosJson.setAdapter(jsonAdapter);
}
}

```

Para personalização do ListView lvDados contido no layout acima precisamos de um layout, que é o faixa_dados.xml e de um Adapter personalizado que é o DadosAdapter. Ambos estão a seguir.

layout: faixa_dados.xml

```

<?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">

    <androidx.constraintlayout.widget.Guideline
        android:id="@+id/guideline1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        app:layout_constraintGuide_percent="0.05" />

    <androidx.constraintlayout.widget.Guideline
        android:id="@+id/guideline2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        app:layout_constraintGuide_percent="0.95" />

    <TextView

```

```

        android:id="@+id/txtIdImovel"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="4dp"
        android:textColor="#4700D6"
        app:layout_constraintStart_toStartOf="@+id/guideline1"
        app:layout_constraintTop_toTopOf="parent"
        tools:text="txtIdImovel" />

<TextView
    android:id="@+id/txtTipoImovel"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_marginTop="4dp"
    android:gravity="end"
    android:textColor="#4700D6"
    app:layout_constraintEnd_toStartOf="@+id/guideline2"
    app:layout_constraintStart_toEndOf="@+id/txtIdImovel"
    app:layout_constraintTop_toTopOf="parent"
    tools:text="txtTipoImovel" />

<TextView
    android:id="@+id/txtBairroImovel"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="4dp"
    android:textColor="#4700D6"
    app:layout_constraintStart_toStartOf="@+id/guideline1"
    app:layout_constraintTop_toBottomOf="@+id/txtIdImovel"
    tools:text="txtBairroImovel" />

<TextView
    android:id="@+id/txtVlrVendaImovel"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_marginTop="4dp"
    android:gravity="end"
    android:textColor="#4700D6"
    app:layout_constraintEnd_toStartOf="@+id/guideline2"
    app:layout_constraintStart_toEndOf="@+id/txtIdImovel"
    app:layout_constraintTop_toBottomOf="@id/txtTipoImovel"
    tools:text="txtVlrVendaImovel" />

<TextView
    android:id="@+id/txtVlrAluguelImovel"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_marginTop="4dp"
    android:gravity="end"
    android:textColor="#4700D6"
    app:layout_constraintEnd_toStartOf="@+id/guideline2"
    app:layout_constraintStart_toEndOf="@+id/txtIdImovel"
    app:layout_constraintTop_toBottomOf="@id/txtVlrVendaImovel"
    tools:text="txtVlrAluguelImovel" />

<TextView
    android:id="@+id/txtComplemento"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:paddingBottom="8dp"
    android:gravity="center"
    android:text="txtComplemento"
    android:textColor="#4700D6"
    app:layout_constraintEnd_toStartOf="@+id/guideline2"
    app:layout_constraintStart_toStartOf="@+id/guideline1"
    app:layout_constraintTop_toBottomOf="@id/txtVlrAluguelImovel"
    tools:text="txtComplemento" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

Classe: DadosAdapter.java

```

public class DadosAdapter extends BaseAdapter {

    private Activity activity;

```

```

private List<String> dados;

public DadosAdapter(Activity activity, List<String> dados) {
    this.activity = activity;
    this.dados = dados;
}

@Override
public int getCount() {
    return dados.size();
}

@Override
public Object getItem(int position) {
    return dados.get(position);
}

@Override
public long getItemId(int position) {
    return position;
}

@Override
public View getView(int position, View view, ViewGroup parent) {
    if (view == null) {
        LayoutInflater inflater = activity.getLayoutInflater();
        view = inflater.inflate(R.layout.faixa_dado, parent, false);
    }
    String s;
    double d;

    s = dados.get(position);
    List<String> imovel = Arrays.asList(s.split(";"));

    TextView txt;
    txt = view.findViewById(R.id.txtIdImovel);
    txt.setText(imovel.get(0));
    txt = view.findViewById(R.id.txtBairroImovel);
    txt.setText(imovel.get(1));
    txt = view.findViewById(R.id.txtTipoImovel);
    txt.setText(imovel.get(2));
    txt = view.findViewById(R.id.txtVlrVendaImovel);
    d = Double.parseDouble(imovel.get(8)); // valor venda
    if (d != 0)
        txt.setText(String.format(Locale.getDefault(), "Valor venda: R$ %, .2f", d));
    else
        txt.setText("não disponível para venda");
    txt = view.findViewById(R.id.txtVlrAluguelImovel);
    d = Double.parseDouble(imovel.get(9)); // valor aluguel
    if (d != 0)
        txt.setText(String.format(Locale.getDefault(), "Valor aluguel: R$ %, .2f", d));
    else
        txt.setText("não disponível para aluguel");
    txt = view.findViewById(R.id.txtComplemento);
    s = "Área terreno: " + imovel.get(3) + " e Área construída: " + imovel.get(4) + "\n";
    s = s + "Dormitórios: " + imovel.get(5) + " - Suítes: " + imovel.get(6) + " - Vagas: " +
imovel.get(7);
    txt.setText(s);

    return view;
}
}

```

Classe: DadosJsonAdapter.java

```

public class DadosJsonAdapter extends BaseAdapter {

    private Activity activity;
    private ArrayList<Imovel> imoveis;

    public DadosJsonAdapter(Activity activity, ArrayList<Imovel> imoveis) {
        this.activity = activity;
        this.imoveis = imoveis;
    }

    @Override
    public int getCount() {
        return imoveis.size();
    }
}

```

```

@Override
public Object getItem(int position) {
    return imoveis.get(position);
}

@Override
public long getItemId(int position) {
    return position;
}

@Override
public View getView(int position, View view, ViewGroup parent) {
    if (view == null) {
        LayoutInflater inflater = activity.getLayoutInflater();
        view = inflater.inflate(R.layout.faixa_dado, parent, false);
    }
    TextView txt;
    Imovel imovel = imoveis.get(position);

    txt = view.findViewById(R.id.txtIdImovel);
    txt.setText(imovel.getCodImovel());
    txt = view.findViewById(R.id.txtBairroImovel);
    txt.setText(imovel.getBairro());
    txt = view.findViewById(R.id.txtTipoImovel);
    txt.setText(imovel.getTipo());
    txt = view.findViewById(R.id.txtVlrVendaImovel);
    if (imovel.getVlrvenda() != 0)
        txt.setText(String.format(Locale.getDefault(), "Valor venda: R$ %,2f", imovel.getVlrvenda()));
    else
        txt.setText("não disponível para venda");
    txt = view.findViewById(R.id.txtVlrAluguelImovel);
    if (imovel.getVlraluga() != 0)
        txt.setText(String.format(Locale.getDefault(), "Valor venda: R$ %,2f", imovel.getVlraluga()));
    else
        txt.setText("não disponível para aluguel");
    txt = view.findViewById(R.id.txtComplemento);
    String s = "Área terreno: " + imovel.getAreaT() + " e Área construída: " + imovel.getAreaC() + "\n";
    s = s + "Dormitórios: " + imovel.getDorms() + " - Suítes: " + imovel.getSuites() + " - Vagas: " +
    imovel.getVagas();
    txt.setText(s);

    return view;
}
}

```

Permissão para acessar a Internet e trafegar caracteres não criptografados

Este aplicativo precisa de permissão de uso de internet. Isto deve ser informado no arquivo AndroidManifest, através da cláusula de permissão conforme indicado a seguir.

Além disso, vamos acessar um endereço na internet através do protocolo HTTP, sem recurso de segurança. Por padrão o Android não permite esse tipo de acesso dos apps a endereços web. Porém, é possível alterar esse padrão e passar a permitir tal tipo de acesso. Essa alteração também é feita no AndroidManifest.xml e também está indicada a seguir. Durante a aula será demonstrado que quando o acesso é feito ao servidor através do protocolo HTTPS (S de Secure) essa liberação não é necessária.

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.ilp506.requisicoeshhttp">

    <uses-permission android:name="android.permission.INTERNET" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/Theme.RequisicoesHTTP"
        android:usesCleartextTraffic="true">
        <activity android:name=".ObtemDados" />
        <activity android:name=".ObtemImagem" />
        <activity android:name=".ObtemArqTexto" />
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

Esta linha concede permissão de acesso à Internet

usesCleartextTraffic = "true" permite trafegar texto não criptografado

Declaração de dependências no Gradle

Neste aplicativo usaremos dois conjuntos de classes externas à plataforma Android, são elas:

- OkHttp3 – que será utilizada para executar as requisições ao servidor
- Picasso – que será utilizada para trabalhar com as imagens que serão carregadas do servidor

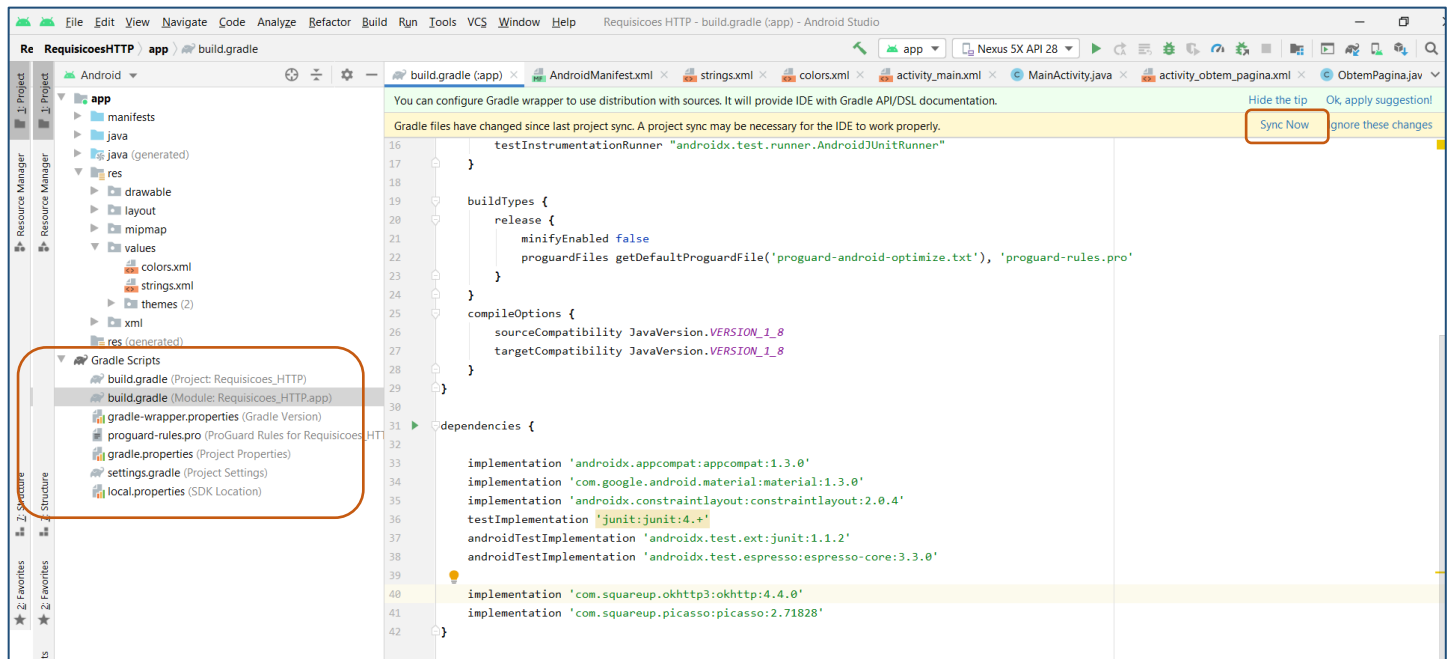
Ambas são projetos Open Source, modernas, simples de usar e seguras. Por esses motivos são amplamente usadas no mundo dos aplicativos mobile para Android.

Para possibilitar seu uso é necessário informar isso ao Android Studio. Essa informação será vinculada ao projeto (e não ao IDE) e deve ser especificada no arquivo build.gradle na seção *dependencies*., usando a diretiva *implementation*, conforme mostrado a seguir.

Para abrir o arquivo build.gradle abra o painel do projeto e expanda as opções sob o item Gradle Scripts. O segundo item da lista é o build.gradle do módulo, dê um duplo clique sobre o mesmo. O arquivo será aberto à direita. Localize a seção de dependências e acrescente estas duas linhas:

```
implementation 'com.squareup.okhttp3:okhttp:4.12.0'
implementation 'com.squareup.picasso:picasso:2.8'
```

Após digitar as dependências clique em Sync Now que apareça no topo da tela e está destacado na imagem.



build.gradle do módulo

```

plugins {
    alias(libs.plugins.android.application)
}

android {
    namespace = "com.ilp506.requisicoeshttp"
    compileSdk = 34

    defaultConfig {
        applicationId = "com.ilp506.requisicoeshttp"
        minSdk = 27
        targetSdk = 34
        versionCode = 1
        versionName = "1.0"

        testInstrumentationRunner = "androidx.test.runner.AndroidJUnitRunner"
    }

    buildTypes {
        release {
            isMinifyEnabled = false
            proguardFiles(
                getDefaultProguardFile("proguard-android-optimize.txt"),
                "proguard-rules.pro"
            )
        }
    }

    compileOptions {
        sourceCompatibility = JavaVersion.VERSION_1_8
        targetCompatibility = JavaVersion.VERSION_1_8
    }
}

dependencies {
    implementation(libs.appcompat)
    implementation(libs.material)
    implementation(libs.activity)
    implementation(libs.constraintlayout)
    implementation("com.squareup.okhttp3:okhttp:4.12.0")
    implementation("com.squareup.picasso:picasso:2.8")
    testImplementation(libs.junit)
    androidTestImplementation(libs.ext.junit)
    androidTestImplementation(libs.espresso.core)
}

```

Estas duas linhas especificam as dependências
(os números de versão podem variar, pois sofrem atualizações)