



CAMILA NIZER PORFIRIO

PROJETO: Documentação

PALHOÇA, SC

2023

SUMÁRIO

NOME DO APP OU PROJETO.....	3
COMO SURTIU A IDEIA.....	3
QUAL O BENEFÍCIO DO APP.....	3
IMPACTO NA SOCIEDADE.....	3
PONTOS NEGATIVOS.....	4
TEMPO DE HORAS TRABALHO (ESTIMATIVA PRÉVIA).....	4
QUAL FERRAMENTA/ MATERIAL DE TRABALHO.....	5
REFINAMENTO.....	5
LOCAIS DE PESQUISA / FERRAMENTAS.....	5
INDEPENDÊNCIA FUNCIONAL.....	5
REGISTRO DOS PRINT / TELAS / CÓDIGOS (PROGRESSO DO APLICATIVO)...	6
26.mar.2023.....	6
06.abr.2023.....	7
07.abr.2023.....	8
13.abr.2023.....	8
18.abr.2023.....	9
04.mar.2023 e 05.mar.2023.....	10
CÓDIGO APLICATIVO.....	12
MainActivity: Página inicial.....	12
MainActivity2: Página de Ajuda.....	20
MainActivitySobre: Página de informações.....	23
Principais comandos utilizados.....	26
REFERÊNCIAS.....	27

NOME DO APP OU PROJETO

Converte fácil - Conversor de moedas

COMO SURTIU A IDEIA

A ideia surgiu após uma viagem, onde era necessário realizar a pesquisa da cotação de moeda estrangeira com regularidade e foi possível constatar a necessidade de um aplicativo e todos os benefícios para controle financeiro consequentes do uso do mesmo.

A dificuldade surgiu quando a pesquisa da cotação era feita via Google e o preço do item foi calculado manualmente na calculadora, com intuito de saber o valor do item em reais e ter real controle do preço e custo em moeda nacional para avaliação e comparação com os preços disponibilizados no Brasil.

Deste modo, não há a “falsa sensação” de que todos os produtos oferecidos no exterior são mais acessíveis do que os nacionais, e consequentemente auxilia o turista e consumidor a avaliar melhor o que lhe é oferecido e se realmente vale a pena a compra.

QUAL O BENEFÍCIO DO APP

Facilidade para consulta de conversão de valores em moeda estrangeira com a cotação atualizada. Praticidade em ter as principais informações das moedas estrangeiras mais utilizadas em um só aplicativo e dispensando o cálculo e pesquisa manual das cotações e valores.

IMPACTO NA SOCIEDADE

A utilização do aplicativo facilita a consulta de valores em diferentes moedas, ocasionando assim maior controle financeiro para quem precisa. Pode ser utilizado em qualquer situação que demande a cotação de valores, seja para viagens, negócios, etc.

Atualmente, vivemos em uma sociedade cada vez mais globalizada. O acesso a moedas estrangeiras está cada vez mais popular, seja por investimentos ou até mesmo viagens e compras internacionais online.

Ter acesso rápido e simplificado à cotação atual de moedas estrangeiras garante real controle financeiro e consequentemente melhoria na vida financeira da sociedade.

PONTOS NEGATIVOS

Necessidade de internet para cotação atualizada de valores. Restrição a apenas algumas moedas estrangeiras.

Potenciais usuários geralmente são aqueles de classe média/alta, que utilizam moeda estrangeira, não sendo assim um aplicativo de uso da população de menor poder aquisitivo.

TEMPO DE HORAS TRABALHO (ESTIMATIVA PRÉVIA)

- Setup do ambiente de desenvolvimento: 2 horas;
- Criação do mockup de telas do aplicativo: 08 horas;
- Implementação da interface do aplicativo: 30 horas;
- Implementação da busca dos dados através API para cotações: 15 horas;

TEMPO DE HORAS TRABALHO (TEMPO EXECUTADO- APROXIMADO)

- Setup do ambiente de desenvolvimento: 5 horas (2h no ambiente de notebook, 3h para alteração do projeto para trabalho no desktop);
- Criação de mockup de telas do aplicativo: 03 horas;
- Implementação da interface do aplicativo: 20 horas;
- Implementação da busca dos dados através da API para cotações: 18 horas;

QUAL FERRAMENTA/ MATERIAL DE TRABALHO

- Software Justinmind, para criação de mockups de telas;
- Software Android Studio 2022.1.1 para implementação do aplicativo;
- Software Android Studio 2022.2.1 para implementação do aplicativo;
- Notebook, Intel Core I5, 8GB, SSD 256GB,
- Desktop, Intel Core I5, 16GB, SSD 256GB +1TB.

REFINAMENTO

Inicialmente o aplicativo disponibilizava apenas cinco tipos de moedas diferentes. Atualmente estão disponíveis onze moedas. Para melhor precisão, ao realizar a cotação, o aplicativo exibe a data e hora no qual foi realizada a cotação.

LOCAIS DE PESQUISA / FERRAMENTAS

Para o presente trabalho é necessário o uso de API para busca de cotação de moedas estrangeiras a fim de se fazer possível a conversão entre moedas, além de pesquisa em sites de documentação da linguagem para implementação.

INDEPENDÊNCIA FUNCIONAL

O aplicativo necessita de conexão com a internet para realização das cotações.

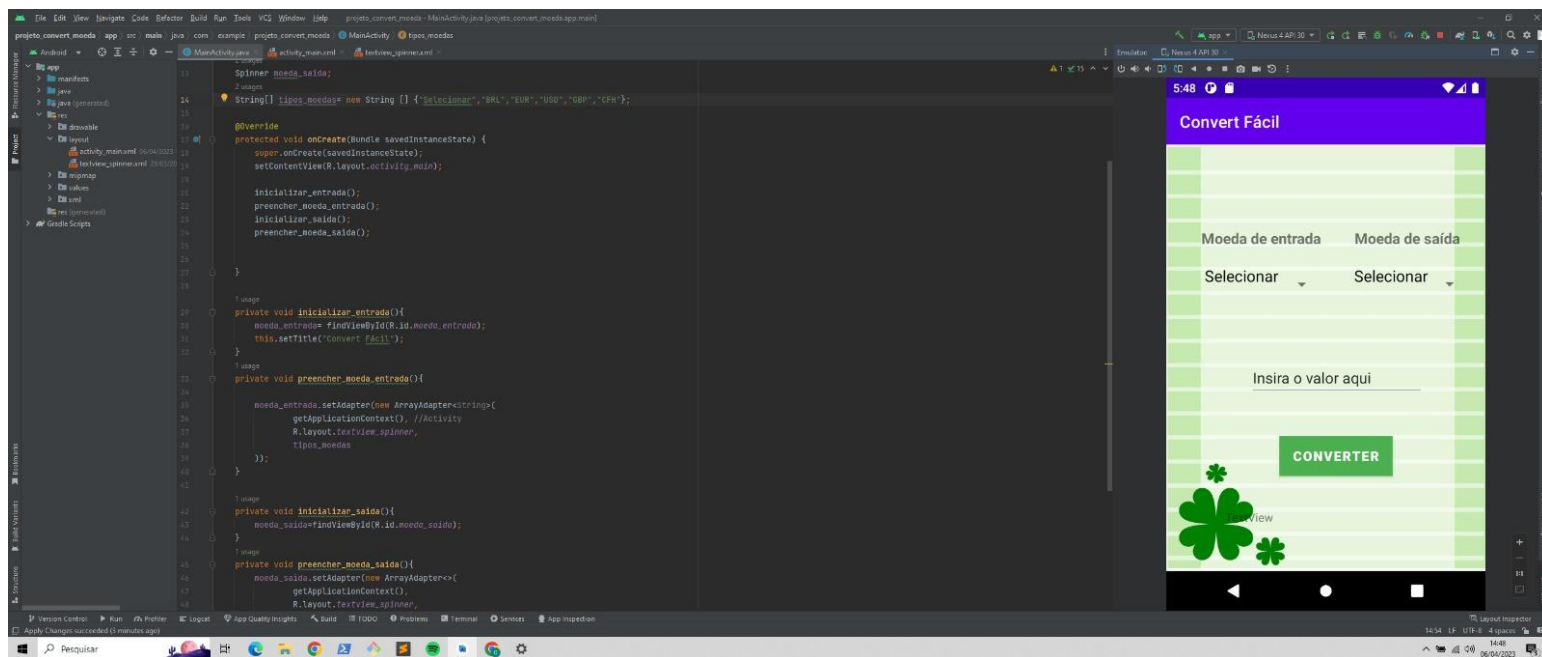
REGISTRO DOS PRINT / TELAS / CÓDIGOS (PROGRESSO DO APLICATIVO)

26.mar.2023

Mockup da tela inicial, conforme imagem abaixo.



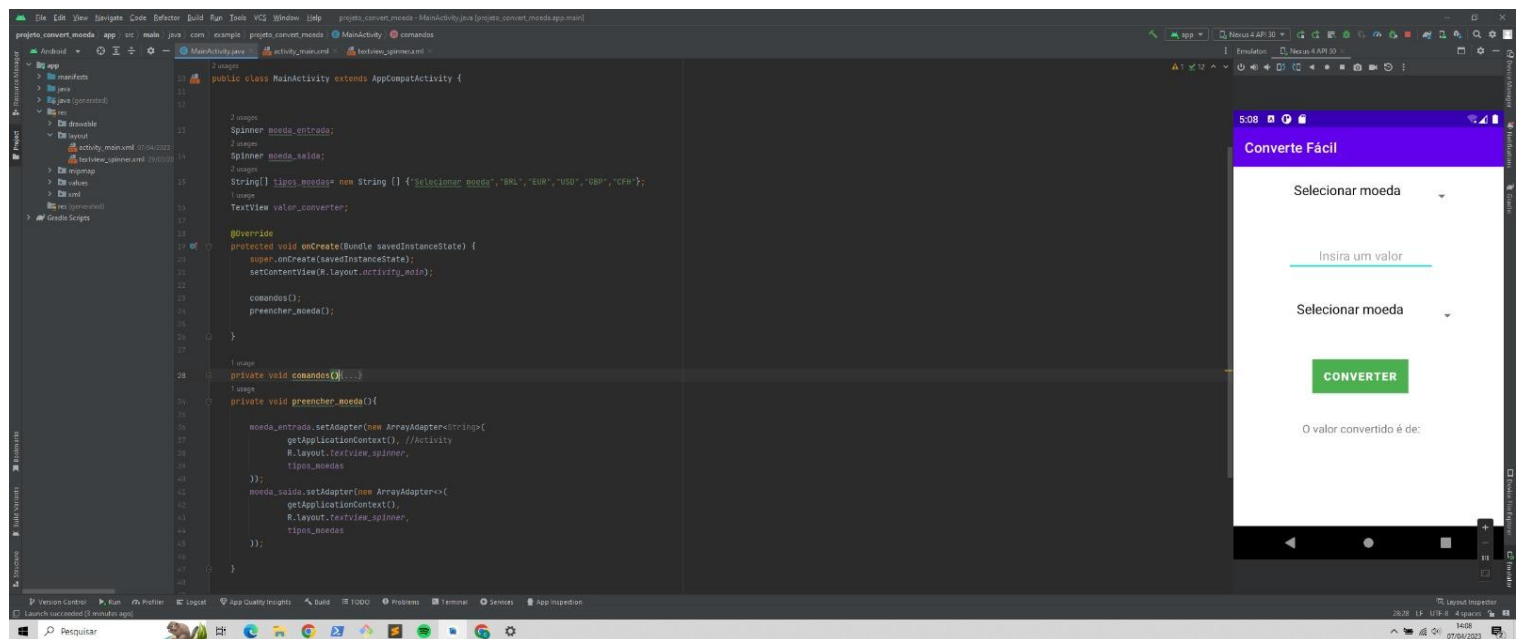
06.abr.2023



Inserção dos elementos na tela. Foram realizados testes para funcionamentos dos elementos inseridos. Nesta data as moedas disponibilizadas foram inseridas em um *array* (*tipos_moedas*). Logo após, criados dois *spinners* (para a entrada e saída) e inserido os elementos através do *array* disponibilizado.¹

¹ Disponível em: <<https://developer.android.com/develo/ui/views/components/spinner>>. Acesso em 08.mai.2023.

07.abr.2023



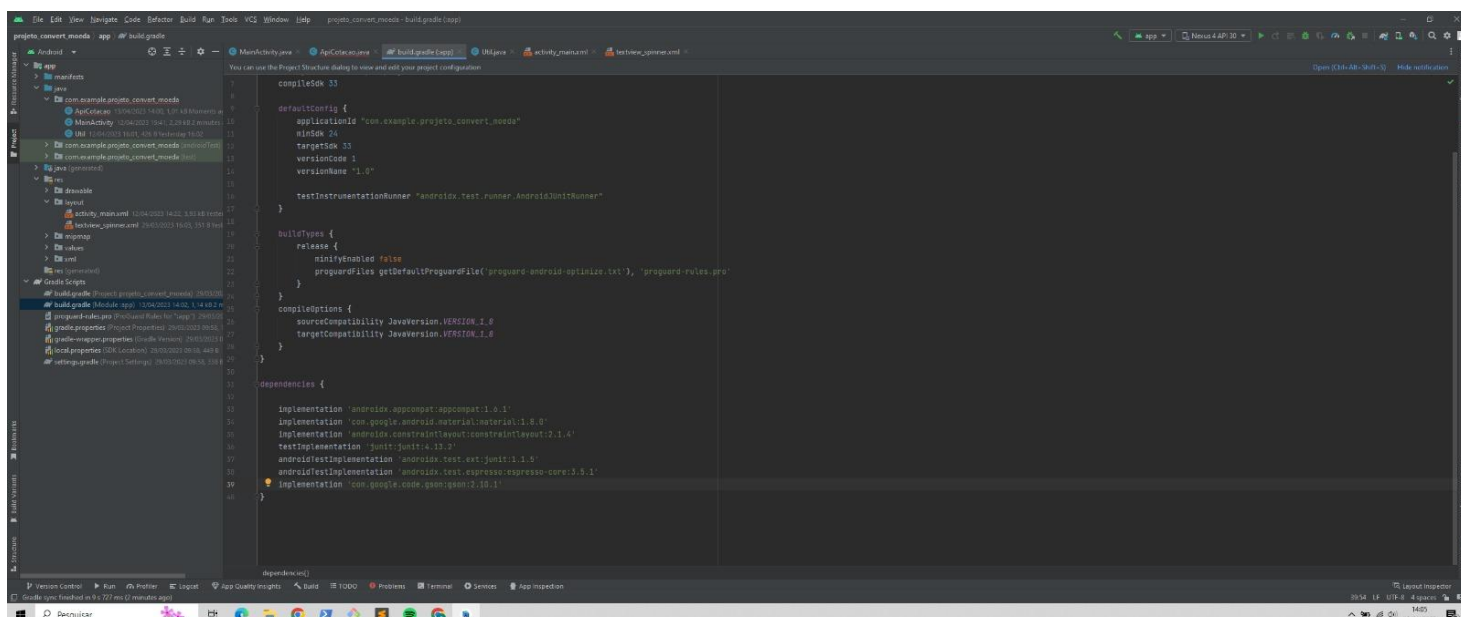
Nesta data foi feita a redução do código repetido. Antes haviam dois void que realizavam a mesma função (preencher o *spinner* com os valores do *arrays*). Após esta alteração foi também atribuído um atributo *hint*² e alterado para aceitar apenas números no *input*³ “Insira um valor”. Por fim, realizou-se a adaptação do projeto ao mockup.

13.abr.2023

Início da utilização da API para cotação de moedas. Foi necessário a instalação da biblioteca google GSON para conversão de objetos.

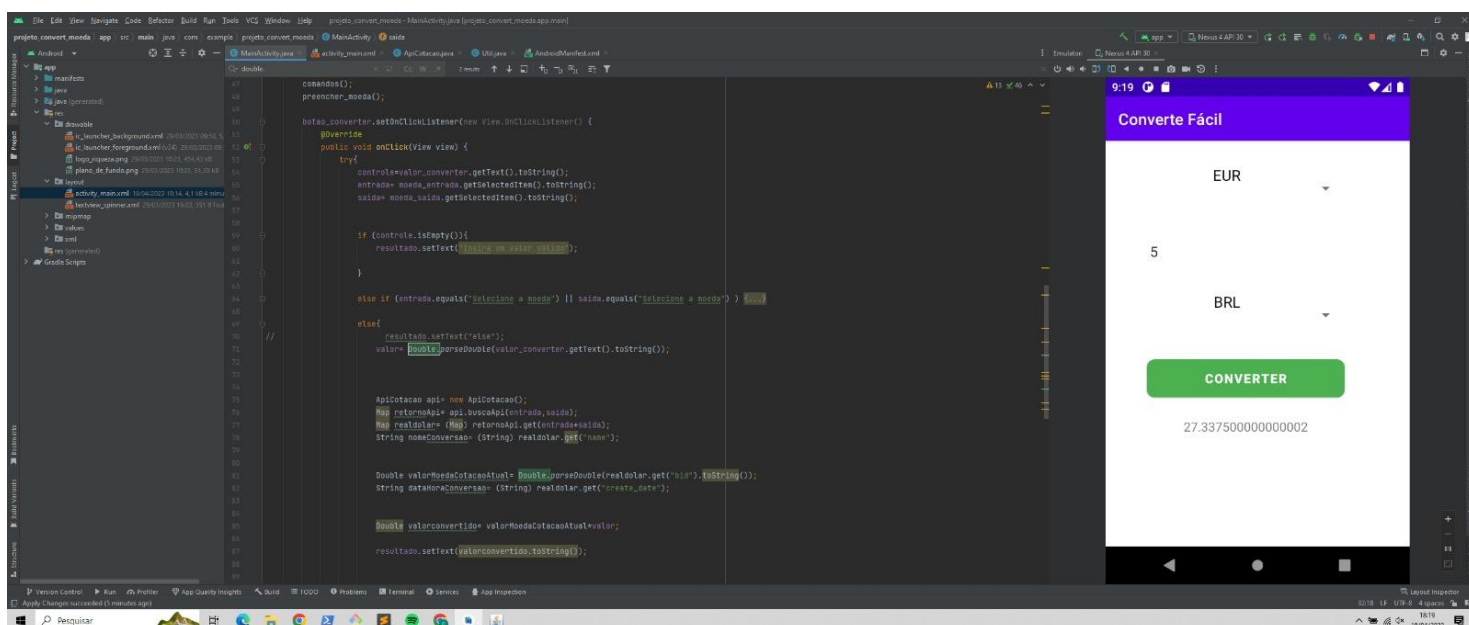
² Android:hint. “*Texto de dica a ser exibido quando o texto estiver vazio*”. Disponível em: <<https://developer.android.com/reference/android/widget/TextView>>. Acesso em 08.mai.2023

³ Especificação do método de entrada (*input*). Disponível em: <<https://developer.android.com/develop/ui/views/touch-and-input/keyboard-input/style>>. Acesso em 08.mai.2023

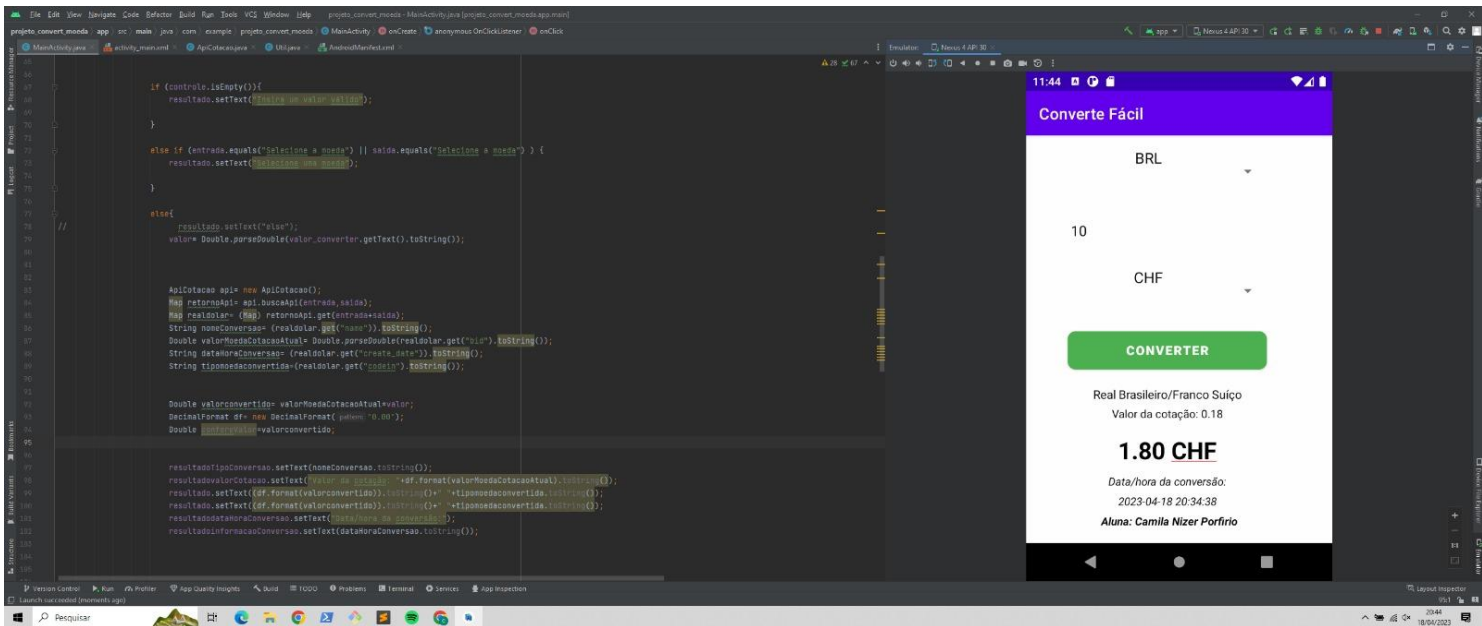


18.abr.2023

Inicialmente a API apresentava erro no retorno valor cotado. Após a instalação da biblioteca GSON e ajustes no Map, o aplicativo funcionou como esperado, conforme imagem abaixo.

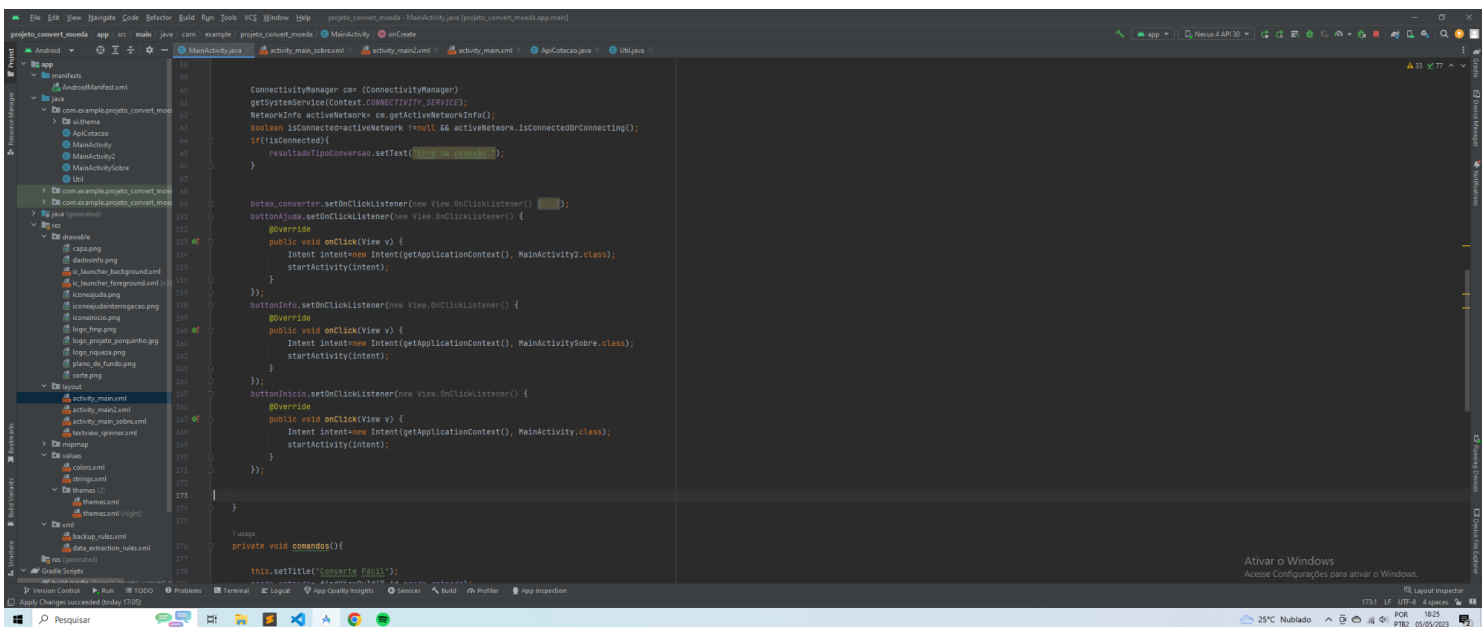


No mesmo dia foram realizados a inserção das demais Text Views para visualização de todas as informações:



04.mar.2023 e 05.mar.2023

Nesta data foram inseridos *ImageView*⁴ para alternar entre as *Activity*⁵ criadas, utilizando o objeto *Intent*⁶, como demonstrado:

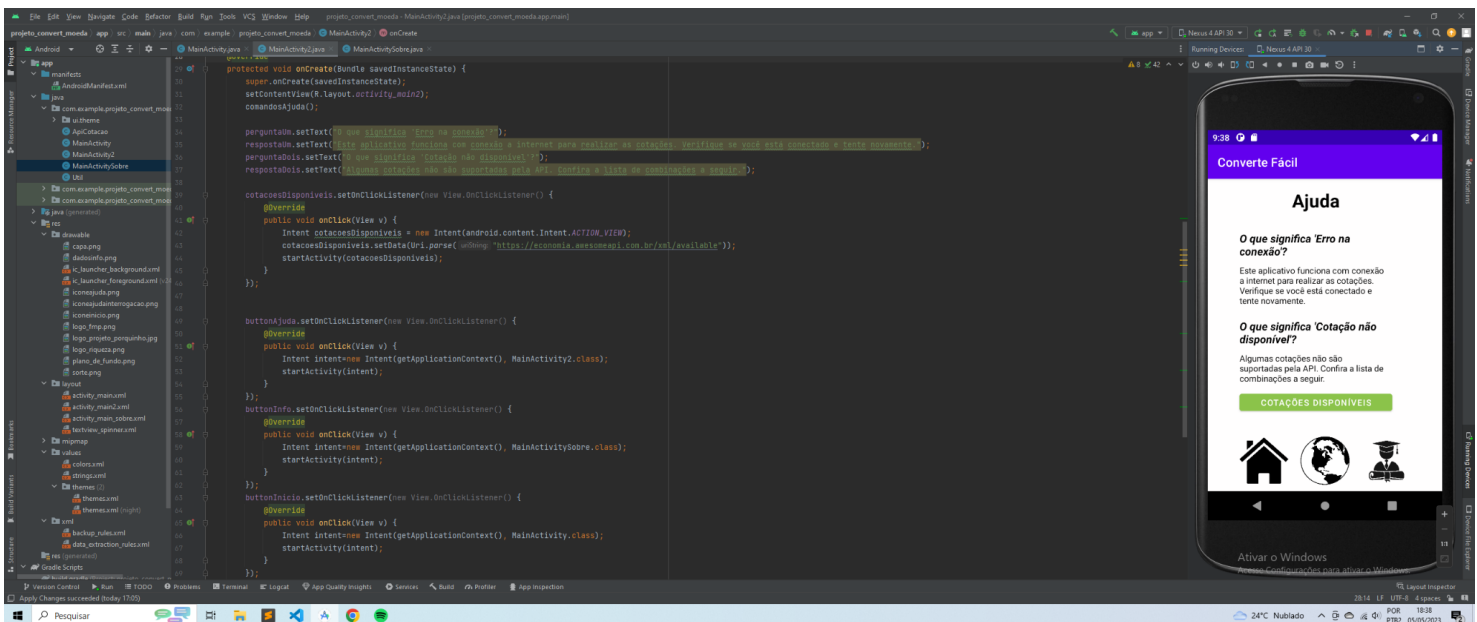


⁴ *ImageView*: Exibe recursos de imagem. Disponível em <https://developer.android.com/reference/android/widget/ImageView>. Acesso em 08.mai.2023.

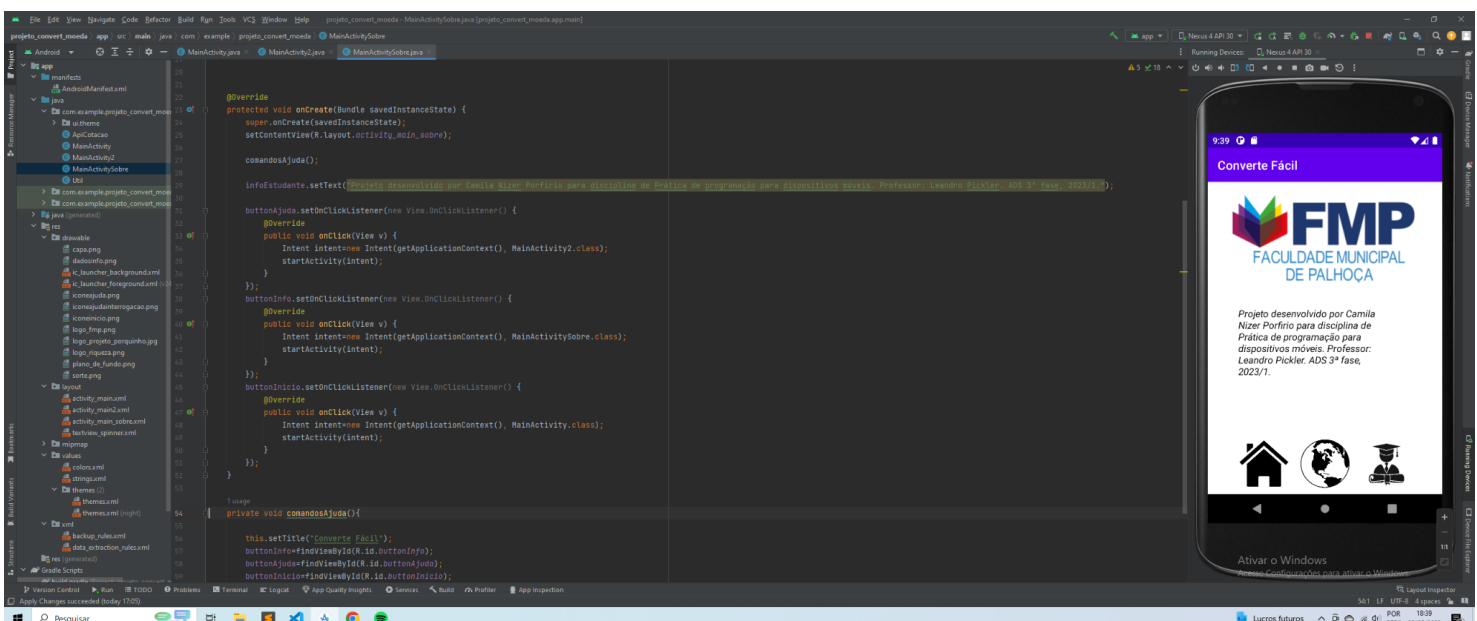
⁵ *Activity*: implementa parte da interface visual do usuário do aplicativo. Disponível em: <https://developer.android.com/guide/topics/manifest/activity-element>. Acesso em 08.mai.2023.

⁶ *Intent*: descrição abstrata de uma operação a ser executada. Disponível em: <https://developer.android.com/reference/android/content/Intent>. Acesso em 08.mai.2023.

Deste modo, foi criada uma *Activity* “Ajuda” para responder eventuais dúvidas dos usuários:



Do mesmo modo, a *Activity* “Sobre” disponibiliza os dados sobre a criação do projeto e responsáveis:



Urge destacar que atualmente todas as *Activities* interagem entre si, facilitando a interação do usuário com o aplicativo. Por fim, foram realizados ajustes na posição e tamanho dos elementos para melhor visualização.

CÓDIGO APLICATIVO

A seguir os principais códigos utilizados para desenvolvimento do aplicativo.

MainActivity: Página inicial

```
package com.example.projeto_convert_moeda;

import androidx.appcompat.app.AppCompatActivity;
import android.content.Context;
import android.content.Intent;
import android.net.ConnectivityManager;
import android.net.NetworkInfo;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.Spinner;
import android.widget.TextView;
import android.os.StrictMode;
import java.text.DecimalFormat;
import java.util.Map;

//Importação das bibliotecas para utilização do app.
```

```

public class MainActivity extends AppCompatActivity {
    Spinner moeda_entrada, moeda_saida;
    String[] tipos_moedas= new String [] {"Selecione a
moeda", "ARS", "AUD", "BRL", "CAD", "CHF", "DKK", "EUR", "GBP", "MXN", "
USD", "UYU"};
    EditText valor_converter;
    Button botao_converter;
    Double valor;
    String entrada, saida, controle;
                                                                    TextView
    resultado, resultadodataHoraConversao, resultadovalorCotacao, res
ultadoinformacaoConversao, resultadoTipoConversao;
    ImageView buttonInicio, buttonInfo, buttonAjuda;

    //Definição dos tipos de variáveis.

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        if (android.os.Build.VERSION.SDK_INT > 9) {
            StrictMode.ThreadPolicy policy =
                                                                    new
StrictMode.ThreadPolicy.Builder().permitAll().build();
            StrictMode.setThreadPolicy(policy);
        }
        //permite as Threads simultâneas no Android.

        comandos();
        preencher_moeda();
    }
}

```

```

        // Inicializa os métodos comandos (busca os Ids nos
arquivos xml, conectando o back e front do app) e
preencher_moeda (spinner)

        botao_converter.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                try{

controle=valor_converter.getText().toString();

                                                                 entrada=
moeda_entrada.getSelectedItem().toString();

                                                                 saida=
moeda_saida.getSelectedItem().toString();

                if (controle.isEmpty()){
                    //Se o usuário não digitar valor para
conversão
                    resultadoTipoConversao.setText("Insira
um valor válido");
                }

                else if (entrada.equals("Selecione a
moeda") || saida.equals("Selecione a moeda") ) {
                    //Se o usuário não selecionar moeda de
entrada ou saída

resultadoTipoConversao.setText("Selecione uma moeda");

                }
                else if(entrada.equals(saida)){

```

```

//Se o usuário selecionar moedas de
entrada e saída iguais

resultadoTipoConversao.setText("Selecione moedas
diferentes.");

    }
    else{

valor=
Double.parseDouble(valor_converter.getText().toString());

        if(valor<=0){
            //Se o usuário digitar valor igual
o menor que zero.

resultadoTipoConversao.setText("Insira um valor válido.");
        }
        else{

            //Executado se todos os campos
forem preenchidos corretamente

valor=
Double.parseDouble(valor_converter.getText().toString());

            ApiCotacao api= new ApiCotacao();

            //Chama a class ApiCotação

            Map retornoApi=
api.buscaApi(entrada,saida); //chama a Classe buscaApi
enviando as Strings parâmetros de entrada e saída

            Map realdolar= (Map)
retornoApi.get(entrada+saida); //recebe o MAP

            String nomeConversao=
(realdolar.get("name")).toString(); //busca no MAP o valor da
chave "name"

            Double valorMoedaCotacaoAtual=
Double.parseDouble(realdolar.get("bid").toString()); // busca
no MAP o valor da chave bid (valor da cotação para compra)

```

```

        String dataHoraConversao=
(realdolar.get("create_date")).toString(); // busca no MAP o
valor da chave "create_date" (data da conversão)

        String
tipomoedaconvertida=(realdolar.get("codein").toString()); //
busca no MAP o valor da chave "codein" (hora da conversão)

        Double valorconvertido=
valorMoedaCotacaoAtual*valor;

        DecimalFormat df= new
DecimalFormat("0.00");

//Utilizado para formatar o valor
com apenas duas casas após o ponto.

        resultadoTipoConversao.setText(nomeConversao.toString());

        resultadovalorCotacao.setText("Valor          da          cotação:
"+df.format(valorMoedaCotacaoAtual).toString());

        resultado.setText((df.format(valorconvertido)).toString()+"
"+tipomoedaconvertida.toString());

        resultado.setText((df.format(valorconvertido)).toString().repl
ace(".",",","")+ " "+tipomoedaconvertida.toString()); //replace:
usado para substituir o . por , no valor exibido.

        resultadodataHoraConversao.setText("Data/hora da conversão:");

        resultadoinformacaoConversao.setText(dataHoraConversao.toStrin
g());

    }

```



```

    }

    } catch (Exception erro){
        //Se retornar erro na API informa o erro ao
usuário

        resultadoTipoConversao.setText("Cotação não
disponível");

        resultadovalorCotacao.setText("");
        resultado.setText("");
        resultadodataHoraConversao.setText("");
        resultadoinformacaoConversao.setText("");


        //Verifica se há conexão com a internet e
caso não tenha informa o erro ao usuário

        ConnectivityManager cm=
        (ConnectivityManager)

        getSystemService(Context.CONNECTIVITY_SERVICE);

        NetworkInfo activeNetwork=
        cm.getActiveNetworkInfo();

        boolean isConnected=activeNetwork !=null &&
        activeNetwork.isConnectedOrConnecting();

        if(!isConnected){
            resultadoTipoConversao.setText("Erro na
conexão.");

            resultadovalorCotacao.setText("");
            resultado.setText("");
            resultadodataHoraConversao.setText("");

            resultadoinformacaoConversao.setText("");

        }
    }
}

```

```

        }
    }
});

//ImageView para alteração de páginas.

        buttonAjuda.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent=new
Intent(getApplicationContext(), MainActivity2.class);
        startActivity(intent);
    }
});

        buttonInfo.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent=new
Intent(getApplicationContext(), MainActivitySobre.class);
        startActivity(intent);
    }
});

        buttonInicio.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent=new
Intent(getApplicationContext(), MainActivity.class);
        startActivity(intent);
    }
});

```

```

    }

    private void comandos() {
        //renomeia o Title
        this.setTitle("Converte Fácil");

        //busca os Ids nos arquivos xml, conectando o back e
front do app
        moeda_entrada= findViewById(R.id.moeda_entrada);
        moeda_saida=findViewById(R.id.moeda_saida);
        valor_converter=findViewById(R.id.valor_a_converter);
        resultado = findViewById(R.id.resultado);
        botao_converter= findViewById(R.id.botao_converter);

        resultadoTipoConversao=findViewById(R.id.resultadoTipoConversa
o);

        resultadovalorCotacao=findViewById(R.id.valorCotacao);

        resultadodataHoraConversao=findViewById(R.id.dataHoraConversao
);

        resultadoinformacaoConversao=findViewById(R.id.informacaoConve
rsao);

        buttonInfo=findViewById(R.id.buttonInfo);
        buttonAjuda=findViewById(R.id.buttonAjuda);
        buttonInicio=findViewById(R.id.buttonInicio);
    }

    private void preencher_moeda() {

        //Insere o array no spinner de moeda de entrada e saída

        moeda_entrada.setAdapter(new ArrayAdapter<>(
            getApplicationContext(), //Activity

```

```

        R.layout.textview_spinner,
        tipos_moedas
    ));
    moeda_saida.setAdapter(new ArrayAdapter<>(
        getApplicationContext(),
        R.layout.textview_spinner,
        tipos_moedas
    ));
}
}

```

MainActivity2: Página de Ajuda

```

package com.example.projeto_convert_moeda;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;

public class MainActivity2 extends AppCompatActivity {

    TextView perguntaUm, perguntaDois, respostaUm, respostaDois;
}

```

```

ImageView buttonInicio,buttonInfo,buttonAjuda;
Button cotacoesDisponiveis;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main2);
    comandosAjuda();

    perguntaUm.setText("O que significa 'Erro na
conexão'?");

    respostaUm.setText("Este aplicativo funciona com
conexão a internet para realizar as cotações. Verifique se
você está conectado e tente novamente.");

    perguntaDois.setText("O que significa 'Cotação não
disponível'?");

    respostaDois.setText("Algumas cotações não são
suportadas pela API. Confira a lista de combinações a
seguir.");

    cotacoesDisponiveis.setOnClickListener(new
View.OnClickListener() {
        @Override
        public void onClick(View v) {
            //Botão com link para as cotações disponíveis
da API utilizada

            Intent cotacoesDisponiveis = new
Intent(android.content.Intent.ACTION_VIEW);

            cotacoesDisponiveis.setData(Uri.parse("https://economia.awesom
eapi.com.br/xml/available"));

            startActivity(cotacoesDisponiveis);
        }
    });
}

```

```

        //ImageView para alteração de páginas.
        buttonAjuda.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent=new
Intent(getApplicationContext(), MainActivity2.class);
                startActivity(intent);
            }
        });

        buttonInfo.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent=new
Intent(getApplicationContext(), MainActivitySobre.class);
                startActivity(intent);
            }
        });

        buttonInicio.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent=new
Intent(getApplicationContext(), MainActivity.class);
                startActivity(intent);
            }
        });
    }
}

```

//busca os Ids nos arquivos xml, conectando o back e front do app

```
private void comandosAjuda() {  
  
    this.setTitle("Converte Fácil");  
    perguntaUm= findViewById(R.id.perguntaUm);  
    perguntaDois=findViewById(R.id.perguntaDois);  
    respostaUm=findViewById(R.id.respostaUm);  
    respostaDois = findViewById(R.id.respostaDois);  
    buttonInfo=findViewById(R.id.buttonInfo);  
    buttonAjuda=findViewById(R.id.buttonAjuda);  
    buttonInicio=findViewById(R.id.buttonInicio);  
  
    cotacoesDisponiveis=findViewById(R.id.cotacoesDisponiveis);  
  
}  
}
```

MainActivitySobre: Página de informações

```
package com.example.projeto_convert_moeda;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.View;
```

```

import android.widget.ImageView;
import android.widget.TextView;

public class MainActivitySobre extends AppCompatActivity {

    ImageView buttonInicio,buttonInfo,buttonAjuda;
    TextView infoEstudante;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main_sobre);

        comandosAjuda();

        infoEstudante.setText("Projeto desenvolvido por Camila
Nizer Porfirio para disciplina de Prática de programação para
dispositivos móveis. Professor: Leandro Pickler. ADS 3ª fase,
2023/1.");

        //ImageView para alteração de páginas.
        buttonAjuda.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent=new
Intent(getApplicationContext(), MainActivity2.class);
                startActivity(intent);
            }
        });
    }
}

```



```

        buttonInfo.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent=new
Intent(getApplicationContext(), MainActivitySobre.class);
        startActivity(intent);
    }
});

        buttonInicio.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent=new
Intent(getApplicationContext(), MainActivity.class);
        startActivity(intent);
    }
});
}

private void comandosAjuda() {

    this.setTitle("Converte Fácil");
    buttonInfo=findViewById(R.id.buttonInfo);
    buttonAjuda=findViewById(R.id.buttonAjuda);
    buttonInicio=findViewById(R.id.buttonInicio);
    infoEstudante=findViewById(R.id.infoEstudante);

}
}

```

Principais comandos utilizados

()toString: *“Retorna uma representação de string do objeto. Em geral, o toString método retorna uma string que “representa textualmente” esse objeto”*⁷.

.equals: *“Indica se algum outro objeto é “igual” a este”*⁸.

BufferedReader : *“Lê o texto de um fluxo de entrada de caracteres, armazenando caracteres em buffer para fornecer uma leitura eficiente de caracteres, matrizes e linhas”*.⁹

Json: *“JSON é um padrão de formatação de dados para troca de informações entre sistemas. A palavra é um acrônimo de JavaScript Object Notation, mas, apesar de ter a linguagem de programação no nome, seu uso não se restringe apenas à ela”*¹⁰.

LinkedHashMap: *“ [...] lista encadeada que define a ordem de iteração, que normalmente é a ordem na qual as chaves foram inseridas no mapa [...]”*¹¹

Map: *“[...] é um objeto que mapeia valores para chaves, ou seja, através da chave consegue ser acessado o valor configurado, sendo que a chave não pode ser repetida ao contrário do valor, mas se caso tiver uma chave repetida é sobrescrito pela última chamada”*.¹²

Json: *“JSON é um padrão de formatação de dados para troca de informações entre sistemas. A palavra é um acrônimo de JavaScript Object Notation, mas, apesar de ter a linguagem de programação no nome, seu uso não se restringe apenas à ela”*¹³.

⁷ Disponível em: <[https://developer.android.com/reference/java/lang/Object#toString\(\)](https://developer.android.com/reference/java/lang/Object#toString())>. Acesso em 08.mai.2023

⁸ Disponível em: <[https://developer.android.com/reference/java/lang/Object#equals\(java.lang.Object\)](https://developer.android.com/reference/java/lang/Object#equals(java.lang.Object))>. Acesso em 08.mai.2023.

⁹ Disponível em: <<https://docs.oracle.com/javase/7/docs/api/java/io/BufferedReader.html>>. Acesso em: 08.mai.2023.

¹⁰ Disponível em: <<https://tecnoblog.net/responde/o-que-e-json-guia-para-iniciantes/>>. Acesso em: 08.mai.2023.

¹¹ Disponível em: <<https://developer.android.com/reference/java/util/LinkedHashMap>>. Acesso em 08.mai.2023.

¹² Disponível em: <<https://www.devmedia.com.br/conhecendo-a-interface-map-do-java/37463>>. Acesso em 08.mai.2023.

¹³ Disponível em: <<https://tecnoblog.net/responde/o-que-e-json-guia-para-iniciantes/>>. Acesso em: 08.mai.2023.

REFERÊNCIAS

ALMEIDA, Arthur dos Santos Almeida. **Consumindo uma API de maneira simples com Java.** Disponível em: <<https://arthur-almeida.medium.com/consumindo-uma-api-de-maneira-simples-com-java-2a386010e4b9>>. Acesso em 08.mai.2023.

ALURA. **Conversor de Moeda. Challenge de Java no ONE.** Disponível em: <<https://cursos.alura.com.br/forum/topico-conversor-de-moeda-challenge-de-java-no-one-271241>>. Acesso em 08.mai.2023.

ALURA. **Pegar valores do Spinner e RadioButton.** Disponível em: <<https://cursos.alura.com.br/forum/topico-pegar-valores-do-spinner-e-radiobutton-46755>>. Acesso em 08.mai.2023.

AWESOMEAPI. **API de cotações.** Disponível em: <<https://docs.awesomeapi.com.br/api-de-moedas>>. Acesso em 08.mai.2023.

BAELDUNG. **Convert JSON to a Map Using Gson.** Disponível em: <<https://www.baeldung.com/gson-json-to-map>>. Acesso em 08.mai.2023.

DEVELOPERS. **Connect to the network.** Disponível em: <<https://developer.android.com/training/basics/network-ops/connecting>>. Acesso em 08.mai.2023.

DEVELOPERS. **EditText.** Disponível em: <<https://developer.android.com/reference/android/widget/EditText>>. Acesso em 08.mai.2023.

DEVELOPERS. **TextView.** Disponível em: <<https://developer.android.com/reference/android/widget/TextView>>. Acesso em 08.mai.2023.

DEVELOPERS. **NetworkOnMainThreadException.** Disponível em: <<https://developer.android.com/reference/android/os/NetworkOnMainThreadException>>. Acesso em 08.mai.2023.

DEVELOPERS. **Specify the input method type.** Disponível em: <<https://developer.android.com/develop/ui/views/touch-and-input/keyboard-input/style>>. Acesso em 08.mai.2023.

DEVELOPERS. **Spinners.** Disponível em: <<https://developer.android.com/develop/ui/views/components/spinner>>. Acesso em 08.mai.2023.

DEVMEDIA. **Conhecendo a interface Map do Java.** Disponível em: <<https://www.devmedia.com.br/conhecendo-a-interface-map-do-java/37463>>. Acesso em 08.mai.2023.

DEVMEDIA. **Decimal Format.** Disponível em:
<<https://www.devmedia.com.br/forum/decimal-format/567155>>. Acesso em:
08.mai.2023.

DEVMEDIA. **Money Api:** Manipulando Dinheiro no Java. Disponível em:
<<https://www.devmedia.com.br/money-api-manipulando-dinheiro-no-java/31207>>.
Acesso em 08.mai.2023.

THECLUB. **Android** - Personalizando o “Spinner” Disponível em:
<<http://www.theclub.com.br/Restrito/Revistas/201403/andr1403.aspx>>. Acesso em
08.mai.2023.