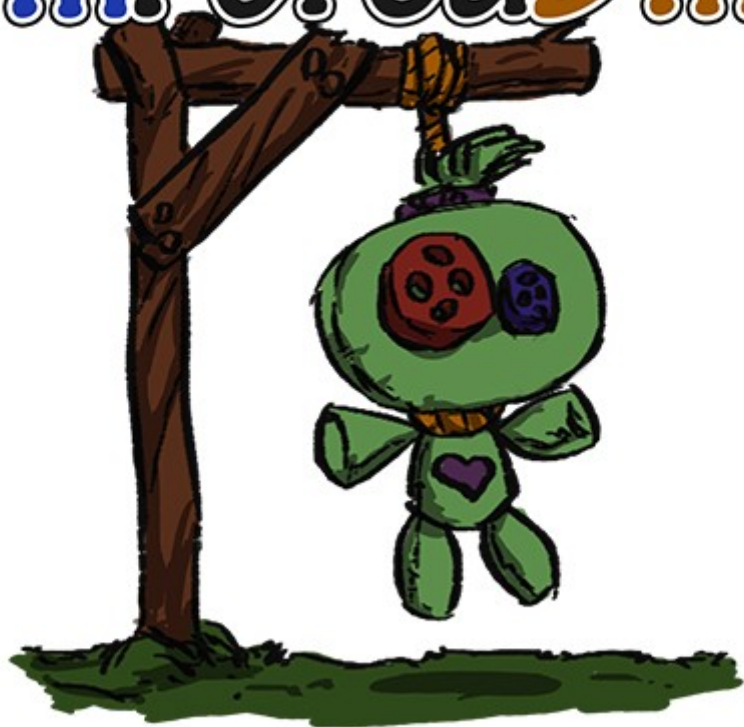


inFôrcaDin



© creative commons

Objetivo

Esta aplicação é resultado de uma ideia gerada para um trabalho de média para a Universidade Anhanguera, na disciplina “Desenvolvimento para Dispositivos Móveis”.

O objetivo do exercício era demonstrar os conhecimentos na criação de uma aplicação mobile, que utilizasse conexão com banco de dados e fosse algo mais do que um simples cadastro.

Já que estas eram as únicas regras e estávamos livres para criar, tivemos a ideia de fazer algo divertido. Assim nasceu **InForcaDin**, uma *app* onde o usuário pode criar suas próprias forcas para jogar com os amigos.

A ideia parte do conceito simples do conhecido jogo de **forca**, que é jogado em papel, com o diferencial que os desafios da forca devem possuir, além da palavra para ser descoberta, 5 dicas para serem exibidas uma a uma quando o jogador errar uma letra. E isso será organizado por categorias. Porque isso? Oras, porque a aplicação tratará de sortear uma das forcas criadas pelo usuário. Ou seja, nem mesmo quem criou o desafio da forca, saberá qual deles estará “na mesa”.

Sobre a aplicação

Esta aplicação também está licenciada sob *GNU/Creative Commons*, o que significa que pode ser usada, modificada e distribuída de forma livre.

Caso você seja desenvolvedor, pode acessar livremente o projeto no *Github* pelo link:

<https://github.com/arturguitelar/inforcadin>

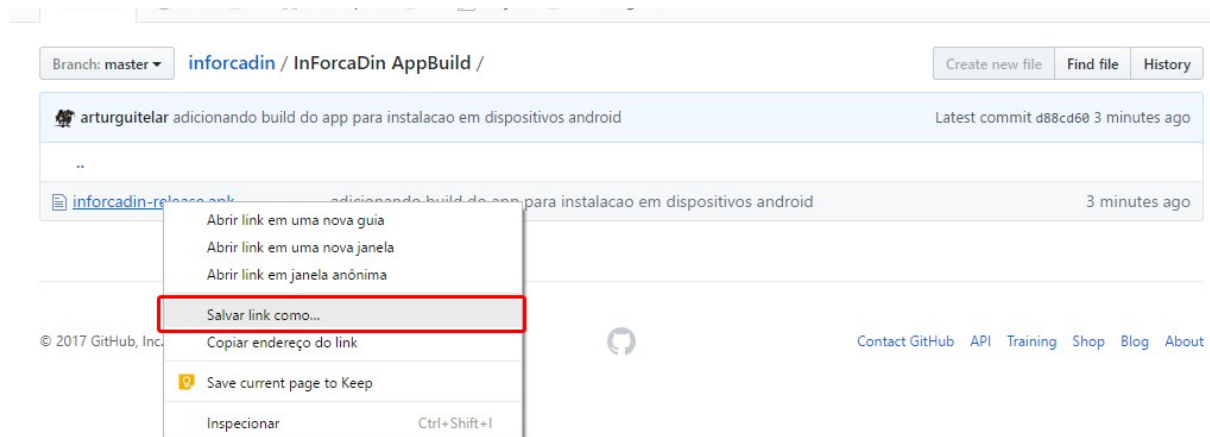
Você também pode criar modificações à vontade e criar seu próprio app através deste, caso seja de seu interesse (ou mesmo só dar uma espiada). Não há pretensão em continuar este projeto, visto que, para efeito didático, ele já cumpriu seu papel.

Isso também significa que, caso você encontre algum *bug*, fique à vontade para tentar corrigir e melhorar o código como quiser.

Caso você queira apenas baixar e se divertir um pouco, fique à vontade. Pode acessar este link:

<https://github.com/arturguitelar/inforcadin/tree/master/InForcaDin%20AppBuild>

Salvar o arquivo apk, como mostra a imagem abaixo:



Basta seguir o procedimento comum no seu aparelho Android para instalar aplicativos de terceiros.

Divirta-se! =]

Sobre este documento

Este documento não tem a pretensão de explicar como cada pedaço de código funciona, mas ele traz todos os códigos e telas para que o leitor possa dar uma olhada em como a coisa toda foi feita. A partir deste ponto será feita uma *listagem de código*, mostrando os códigos **xml** e **java** da aplicação.

Telas da aplicação e seus códigos de layout xml.



activity_main.xml

```
<?xml version="1.0" encoding="utf-8" ?>
<ScrollView 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"
    tools:context="trabalhoandroid.com.br.inforcadin.activities.MainActivity"
    android:orientation="vertical"
    android:layout_margin="@dimen/bordas">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">
        <ImageView
            android:layout_weight="0"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            app:srcCompat="@drawable/imagem_principal" />
        <Button
            android:id="@+id/btnJogarInicio"
            android:layout_weight="1"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:text="@string/btn_jogar" />
        <Button
            android:id="@+id/btnCriarInicio"
            android:layout_weight="1"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:text="@string/btn_criar" />
    </LinearLayout>
</ScrollView>
```

InForkDin

Categoria: Quadrinho



._. A ._. ._. A ._.

(0/6) Letras erradas:

A

CONFIRMAR LETRA

Dicas:

InForkDin

Categoria: Quadrinho



._. A ._. ._. A ._.

(0/6) Letras erradas:

A

CONFIRMAR LETRA

Dicas:

Você acertou!

InForkDin

Categoria: Quadrinho



._. A ._. ._. A ._.

(1/6) Letras erradas: D

D

CONFIRMAR LETRA

Dicas:

Você errou!

Gosta de botar medo.

InForkDin

Categoria: Quadrinho



._. A ._. ._. A ._.

(2/6) Letras erradas: D G

G

CONFIRMAR LETRA

Dicas:

Gosta de botar medo.

activity_ingame.xml

```
<?xml version="1.0" encoding="utf-8" ?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_margin="@dimen/bordas">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">
        <TextView
            android:id="@+id/txtCategorias"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Categoria:"
            android:layout_marginBottom="@dimen/bordas"/>
        <ImageView
            android:id="@+id/imgForca"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            app:srcCompat="@drawable/boneco_forca_01"
            android:layout_marginBottom="@dimen/bordas"/>
        <TextView
            android:id="@+id/txtPalavraForca"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginBottom="@dimen/margin_bottom_campos"
            android:text="palavra"
            android:textSize="30sp" />
        <TextView
            android:id="@+id/txtLetrasErradas"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="letras erradas"
            android:textSize="24sp"
            android:layout_marginBottom="@dimen/margin_bottom_campos"/>
        <Spinner
            android:id="@+id/spnEscolherLetra"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginBottom="@dimen/margin_bottom_campos"/>
        <Button
            android:id="@+id/btnConfirmarLetra"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Confirmar Letra"
            android:layout_marginBottom="@dimen/margin_bottom_campos"/>
        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Dicas:" />
        <ListView
            android:id="@+id/lvDicas"
            android:layout_width="match_parent"
            android:layout_height="140dp"/>
    </LinearLayout>
</ScrollView>
```



activity_vitoria.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.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"

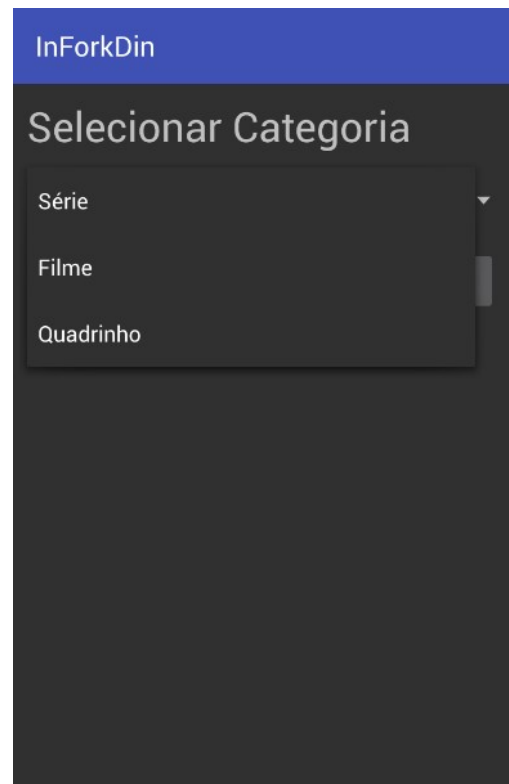
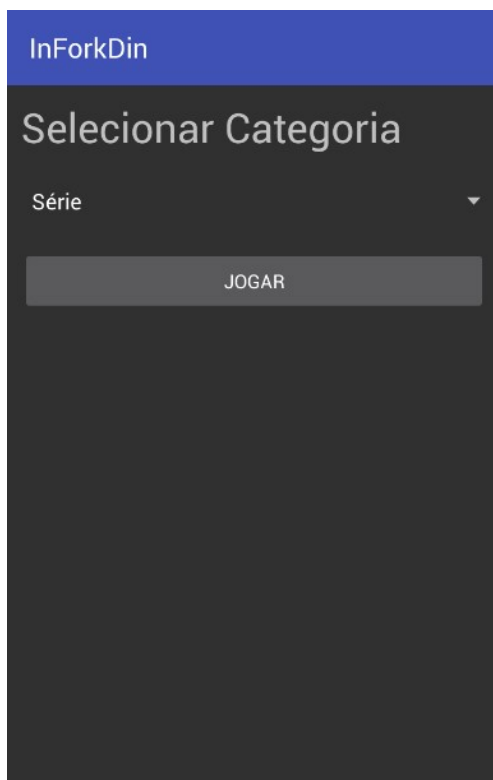
tools:context="trabalhoandroid.com.br.inforcadin.activities.VitoriaActivity">
    <ImageView
        android:id="@+id/imageView"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        app:srcCompat="@drawable/tela_de_vitoria"
        android:layout_marginLeft="8dp"
        app:layout_constraintLeft_toLeftOf="parent"
        android:layout_marginRight="8dp"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        android:layout_marginTop="24dp" />
    <Button
        android:id="@+id/btnJogarNovamente"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:text="@string/btn_jogar_novamente"
        android:layout_marginTop="24dp"
        app:layout_constraintTop_toBottomOf="@+id/imageView"
        android:layout_marginLeft="8dp"
        app:layout_constraintLeft_toLeftOf="parent"
        android:layout_marginRight="8dp"
        app:layout_constraintRight_toRightOf="parent" />
</android.support.constraint.ConstraintLayout>
```




activity_derrota.xml

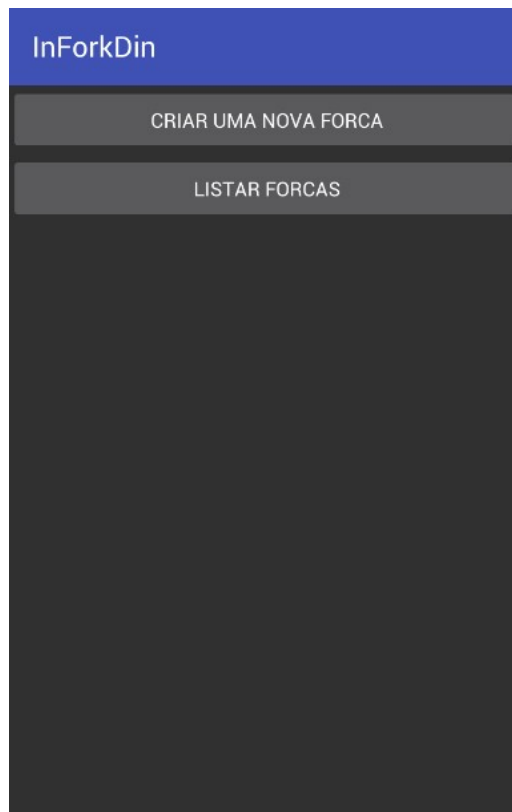
```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"

tools:context="trabalhoandroid.com.br.inforcadin.activities.DerrotaActivity">
    <ImageView
        android:id="@+id/imageView2"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        app:srcCompat="@drawable/tela_de_derrota"
        android:layout_marginLeft="8dp"
        app:layout_constraintLeft_toLeftOf="parent"
        android:layout_marginRight="8dp"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        android:layout_marginTop="24dp" />
    <Button
        android:id="@+id/btnJogarNovamente"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:text="@string/btn_jogar_novamente"
        android:layout_marginTop="16dp"
        app:layout_constraintTop_toBottomOf="@+id/imageView2"
        android:layout_marginRight="8dp"
        app:layout_constraintRight_toRightOf="parent"
        android:layout_marginLeft="8dp"
        app:layout_constraintLeft_toLeftOf="parent" />
</android.support.constraint.ConstraintLayout>
```



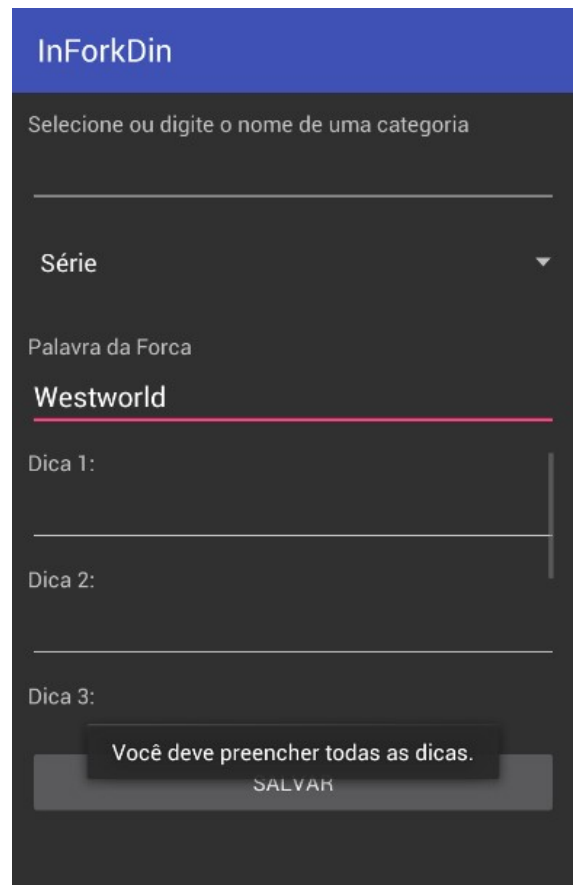
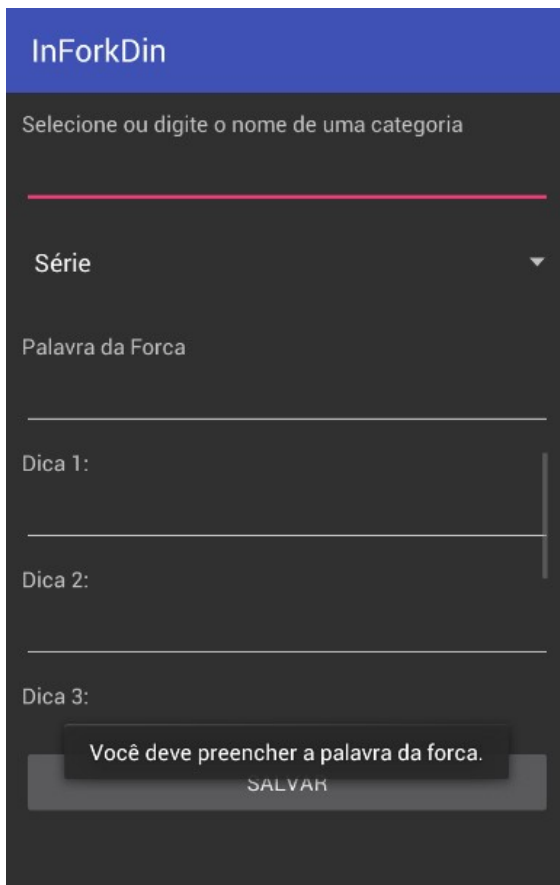
activity_seleciona_categoria.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_margin="@dimen/bordas">
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/txt_selecionar"
        android:textAlignment="center"
        android:textSize="30sp"
        android:layout_marginBottom="@dimen/margin_bottom_campos"/>
    <Spinner
        android:id="@+id/spinnerSelecionar"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:dropDownWidth="match_parent"
        android:visibility="visible"
        android:layout_marginBottom="@dimen/margin_bottom_campos" />
    <Button
        android:id="@+id/btnJogar"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/btn_jogar" />
</LinearLayout>
```



activity_escolha_nova_forca.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent">
    <Button
        android:id="@+id/btnCriarNovaForca"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/btn_nova_forca" />
    <Button
        android:id="@+id/btnListarForcas"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/btn_listar" />
</LinearLayout>
```



activity_cria_desafio.xml

```
<?xml version="1.0" encoding="utf-8" ?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_margin="@dimen/bordas">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical"
            android:layout_marginBottom="@dimen/margin_bottom_campos">
            <TextView
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:text="Selecione ou digite o nome de uma categoria" />
            <EditText
                android:id="@+id/etNovaCategoria"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginBottom="@dimen/margin_bottom_campos"/>
            <Spinner
                android:id="@+id/spnCategoria"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginBottom="@dimen/margin_bottom_campos"/>
        </LinearLayout>
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
```

```

        android:orientation="vertical"
        android:layout_marginBottom="@dimen/margin_bottom_campos">
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Palavra da Forca" />
        <EditText
            android:id="@+id/etPalavraForca"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:ems="10"
            android:inputType="textPersonName"/>
    </LinearLayout>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Dica 1:"/>
        <EditText
            android:id="@+id/etDica_1"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginBottom="@dimen/margin_bottom_campos"/>
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Dica 2:"/>
        <EditText
            android:id="@+id/etDica_2"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginBottom="@dimen/margin_bottom_campos" />
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Dica 3:"/>
        <EditText
            android:id="@+id/etDica_3"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginBottom="@dimen/margin_bottom_campos" />
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Dica 4:"/>
        <EditText
            android:id="@+id/etDica_4"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginBottom="@dimen/margin_bottom_campos" />
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Dica 5:"/>
        <EditText
            android:id="@+id/etDica_5"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginBottom="@dimen/margin_bottom_campos" />
    </LinearLayout>

```

```
<Button
    android:id="@+id/btnSalvar"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/btn_salvar"/>
<Button
    android:id="@+id/btnEditar"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/btn_alterar"
    android:visibility="gone"/>
<Button
    android:id="@+id/btnExcluir"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="excluir"
    android:visibility="gone"/>
</LinearLayout>
</ScrollView>
```

InForkDin	
Categoria	Palavra da força
Filme	Vinqadores
Filme	O Profissional
Filme	O Senhor dos Anéis
Série	Mr. Robot
Série	Todo mundo odeia o Chris
Série	Eu a patroa e as crianças
Quadrinho	Wolverine
Quadrinho	Homem-Aranha
Quadrinho	Batman
Série	Westworld

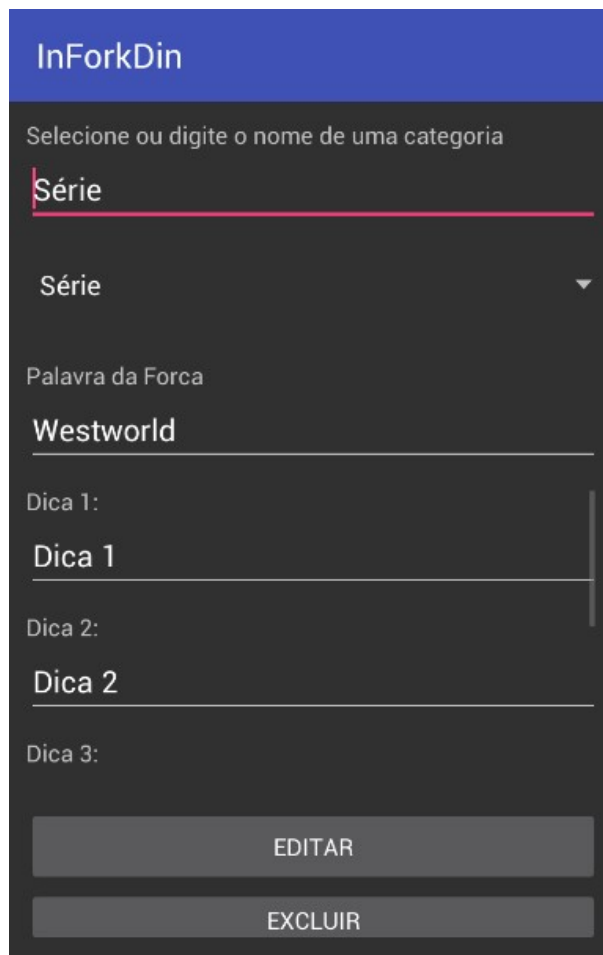
activity_lista_forcas.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="@dimen/bordas">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:layout_marginBottom="@dimen/margin_bottom_campos">
        <TextView
            android:layout_weight="1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/txt_categoria" />
        <TextView
            android:layout_weight="2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/txt_forca" />
    </LinearLayout>
    <ListView
        android:id="@+id/lvBuscar"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"/>
</LinearLayout>
```

Este é o xml utilizado para o preenchimento da lista na tela de listagem de forcas.

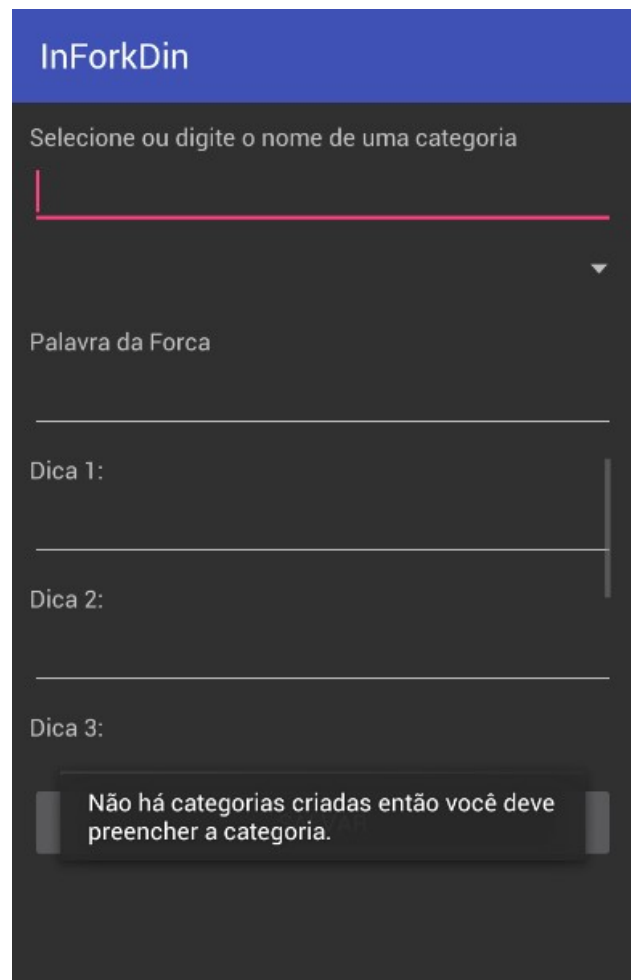
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="horizontal" android:layout_width="match_parent"
    android:layout_height="match_parent">
    <TextView
        android:id="@+id/txtCategoriaLV"
        android:layout_weight="2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:maxLength="100dp"
        android:text="@string/txt_categoria"
        android:layout_marginTop="@dimen/margin_bottom_campos"
        android:layout_marginBottom="@dimen/margin_bottom_campos"/>
    <TextView
        android:id="@+id/txtPalavraLV"
        android:layout_weight="1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:maxLength="200dp"
        android:text="@string/txt_forca"
        android:layout_marginTop="@dimen/margin_bottom_campos"
        android:layout_marginBottom="@dimen/margin_bottom_campos"/>
</LinearLayout>
```

A tela de edição de forcas utiliza o layout activity_cria_desafio.xml, que já foi mostrado previamente. A diferença é que o botão de “salvar” é trocado pelos botões “editar” e “excluir” através de código java.



The screenshot shows the 'InForkDin' app interface. At the top, there's a blue header with the text 'InForkDin'. Below it, a dark grey area contains the instruction 'Selecione ou digite o nome de uma categoria'. A text input field below this instruction contains the word 'Série' and has a red underline. Below the input field is a dropdown menu also showing 'Série' with a downward arrow. Further down, there's a label 'Palavra da Forca' followed by a text input field containing 'Westworld'. Below that are three hint sections: 'Dica 1:' with input 'Dica 1', 'Dica 2:' with input 'Dica 2', and 'Dica 3:' with an empty input field. At the bottom, there are two large grey buttons labeled 'EDITAR' and 'EXCLUIR'.

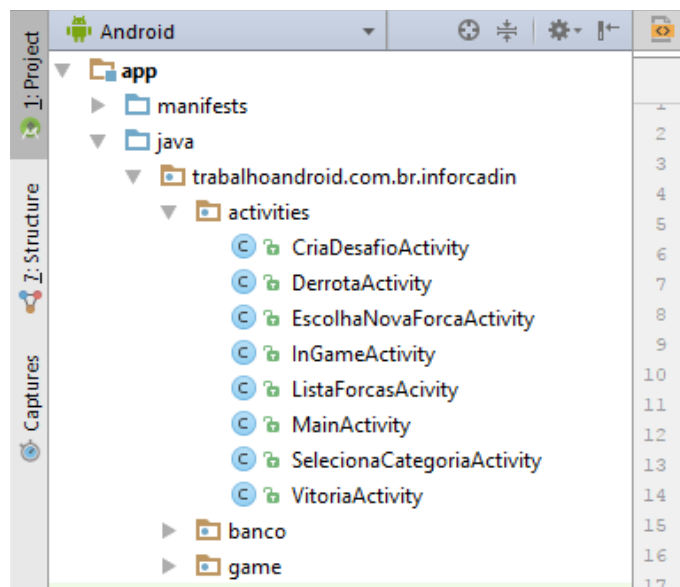
Caso não existam forcas no banco de dados, este deve ser o comportamento da aplicação na tela de início e na tela de criação de forcas.



Código Java

O código java foi organizado da seguinte forma:

- No diretório **activities** há o código relacionado às telas da aplicação e também (para fins de simplicidade) o código de funcionamento de algumas ações.



CriaDesafioActivity.java

```
package trabalhoandroid.com.br.inforcadin.activities;
import android.content.Intent;
import android.database.Cursor;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Spinner;
import android.widget.Toast;
import java.util.ArrayList;
import java.util.HashSet;
import java.util.Set;
import trabalhoandroid.com.br.inforcadin.R;
import trabalhoandroid.com.br.inforcadin.banco.*;
/**
 * Created by Artur on 19/05/2017.
 */
public class CriaDesafioActivity extends AppCompatActivity {
    private Spinner spnCategorias;
    private Button btnSalvar, btnEditar, btnExcluir;
    private EditText etNovaCategoria, etPalavraForca;
    private EditText[] dicas;
    private ArrayList<String> categorias;
    private ForcaDao crud;
    String codigo;
    Cursor cursorCodigo;
    // para armazenar os campos quando convertidos para string
    String pal, d1, d2, d3, d4, d5;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_cria_desafio);
        btnSalvar = (Button) findViewById(R.id.btnSalvar);
        btnEditar = (Button) findViewById(R.id.btnEditar);
        btnExcluir = (Button) findViewById(R.id.btnExcluir);
        etNovaCategoria = (EditText) findViewById(R.id.etNovaCategoria);
        etPalavraForca = (EditText) findViewById(R.id.etPalavraForca);
        dicas = new EditText[] {
            (EditText) findViewById(R.id.etDica_1),
            (EditText) findViewById(R.id.etDica_2),
            (EditText) findViewById(R.id.etDica_3),
            (EditText) findViewById(R.id.etDica_4),
            (EditText) findViewById(R.id.etDica_5)
        };
        spnCategorias = (Spinner) findViewById(R.id.spnCategoria);
        crud = new ForcaDao(getApplicationContext());
        codigo = getIntent().getStringExtra("codigo");
        preencherSpinner();
        if (getIntent().hasExtra("codigo")) {
            btnSalvar.setVisibility(View.GONE);
            btnEditar.setVisibility(View.VISIBLE);
            btnExcluir.setVisibility(View.VISIBLE);
            cursorCodigo = crud.buscarById(Integer.parseInt(codigo));

            etNovaCategoria.setText(cursorCodigo.getString(cursorCodigo.getColumnIndexOrThrow(ForcaHelperDao.getNomeCategoria())));

            etPalavraForca.setText(cursorCodigo.getString(cursorCodigo.getColumnIndexOrThrow(ForcaHelperDao.getPALAVRA())));
        }
    }
}
```

```

dicas[0].setText(cursorCodigo.getString(cursorCodigo.getColumnIndexOrThrow(ForcaHelperDao.getDica1())));

dicas[1].setText(cursorCodigo.getString(cursorCodigo.getColumnIndexOrThrow(ForcaHelperDao.getDica2())));

dicas[2].setText(cursorCodigo.getString(cursorCodigo.getColumnIndexOrThrow(ForcaHelperDao.getDica3())));

dicas[3].setText(cursorCodigo.getString(cursorCodigo.getColumnIndexOrThrow(ForcaHelperDao.getDica4())));

dicas[4].setText(cursorCodigo.getString(cursorCodigo.getColumnIndexOrThrow(ForcaHelperDao.getDica5())));
    }
    btnSalvar.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            salvar();
        }
    });
    btnEditar.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            atualizar();
        }
    });
    btnExcluir.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            excluir();
        }
    });
}

private void salvar() {
    String categoria = "";
    // caso o seletor de categorias esteja vazio, o usuário deve preencher
a categoria
    // se existir algo no seletor, o usuário pode selecionar
    if(isCampoVazio(etNovaCategoria)) {
        if( !categorias.isEmpty() ) {
            categoria = spnCategorias.getSelectedItem().toString();
        } else {
            Toast.makeText(CriaDesafioActivity.this, "Não há categorias
criadas então você deve preencher a categoria.", Toast.LENGTH_SHORT).show();
        }
    } else {
        categoria = etNovaCategoria.getText().toString();
    }
    // caso o campo forca não esteja vazio, verifica se todas as dicas estão
devidamente preenchidas
    if( !isCampoVazio(etPalavraForca) ) {
        if(isNotCamposDeDicasVazio()) {
            salvarDados(categoria);
            Toast.makeText(CriaDesafioActivity.this, "Forca salva com
sucesso.", Toast.LENGTH_SHORT).show();
            voltarParaActivity();
        } else {
            Toast.makeText(CriaDesafioActivity.this, "Você deve preencher
todas as dicas.", Toast.LENGTH_SHORT).show();
        }
    } else {
        Toast.makeText(CriaDesafioActivity.this, "Você deve preencher a

```

```

palavra da forca.", Toast.LENGTH_SHORT).show();
    }
}
private void atualizar() {
    String categoria = "";
    if (isCampoVazio(etNovaCategoria)) {
        categoria = spnCategorias.getSelectedItem().toString();
    } else {
        categoria = etNovaCategoria.getText().toString();
    }
    if (!isCampoVazio(etPalavraForca)) {
        if (isNotCamposDeDicasVazio()) {
            atualizarDados(categoria);
            Toast.makeText(CriaDesafioActivity.this, "Alterado.",
Toast.LENGTH_SHORT).show();
            voltarParaActivity();
        } else {
            Toast.makeText(CriaDesafioActivity.this, "Você deve preencher
todas as dicas.", Toast.LENGTH_SHORT).show();
        }
    } else {
        Toast.makeText(CriaDesafioActivity.this, "Você deve preencher a
palavra da forca.", Toast.LENGTH_SHORT).show();
    }
}
private void excluir() {
    crud.deletar(Integer.parseInt(codigo));
    Toast.makeText(CriaDesafioActivity.this, "Excluído.",
Toast.LENGTH_SHORT).show();
    voltarParaActivity();
}
private void salvarDados(String categoria) {
    String cat = categoria;
    converterCamposParaString();
    crud.inserir(cat, pal, d1, d2, d3, d4, d5);
}
private void atualizarDados(String categoria) {
    String cat = categoria;
    converterCamposParaString();
    crud.alterar(Integer.parseInt(codigo), cat, pal, d1, d2, d3, d4, d5);
}
private void converterCamposParaString() {
    pal = etPalavraForca.getText().toString();
    d1 = dicas[0].getText().toString();
    d2 = dicas[1].getText().toString();
    d3 = dicas[2].getText().toString();
    d4 = dicas[3].getText().toString();
    d5 = dicas[4].getText().toString();
}
private void preencherSpinner() {
    final Cursor cursor = crud.buscar();
    Set<String> catSet = new HashSet<>();
    categorias = new ArrayList<>();
    for (cursor.moveToFirst(); (!cursor.isAfterLast()); cursor.moveToNext())
{
        catSet.add(cursor.getString(1).toString());
    }
    for (String set : catSet) {
        categorias.add(set);
    }
    ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,
android.R.layout.simple_spinner_dropdown_item, categorias);
}

```

```

adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
    spnCategorias.setAdapter(adapter);
}
private boolean isCampoVazio(EditText campo) {
    return campo.getText().length() <= 0;
}
private boolean isNotCamposDeDicasVazio() {
    int contador = 0;
    for (EditText dica : dicas) {
        if (dica.getText().length() > 0) {
            contador++;
        }
    }
    return contador >= dicas.length;
}
private void voltarParaActivity() {
    Intent intent = new Intent(CriaDesafioActivity.this,
EscolhaNovaForcaActivity.class);
    startActivity(intent);
    finish();
}
}

```

DerrotaActivity.java

```

package trabalhoandroid.com.br.inforcadin.activities;
import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import trabalhoandroid.com.br.inforcadin.R;
public class DerrotaActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_derrota);
        Button btnJogarNovamente = (Button)
findViewById(R.id.btnJogarNovamente);
        btnJogarNovamente.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent it = new Intent(DerrotaActivity.this,
SelecionaCategoriaActivity.class);
                startActivity(it);
                finish();
            }
        });
    }
}

```

EscolhaNovaForcaActivity.java

```

package trabalhoandroid.com.br.inforcadin.activities;
import android.content.Intent;
import android.os.Bundle;
import android.support.annotation.Nullable;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
import trabalhoandroid.com.br.inforcadin.R;

```

```

/**
 * Created by Artur on 21/05/2017.
 */
public class EscolhaNovaForcaActivity extends AppCompatActivity {
    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_escolha_nova_forca);
        Button btnCriarNovaForca = (Button)
findViewById(R.id.btnCriarNovaForca);
        Button btnListarForcas = (Button) findViewById(R.id.btnListarForcas);
        btnCriarNovaForca.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent it = new Intent(EscolhaNovaForcaActivity.this,
CriaDesafioActivity.class);
                startActivity(it);
                finish();
            }
        });
        btnListarForcas.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent it = new Intent(EscolhaNovaForcaActivity.this,
ListaForcasActivity.class);
                startActivity(it);
                finish();
            }
        });
    }
}

```

InGameActivity.java

```

package trabalhoandroid.com.br.inforcadin.activities;
import android.content.Intent;
import android.os.Bundle;
import android.support.annotation.Nullable;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.*;
import java.util.ArrayList;
import java.util.List;
import trabalhoandroid.com.br.inforcadin.R;
import trabalhoandroid.com.br.inforcadin.game.DesafioForca;
/**
 * Created by Artur on 21/05/2017.
 */
public class InGameActivity extends AppCompatActivity {
    // elementos do layout
    ImageView imgForca;
    TextView txtCategorias, txtPalavraForca, txtLetrasErradas;
    Spinner spnEscolherLetra;
    Button btnConfirmarLetra;
    ListView lvDicas;
    // demais propriedades
    int contadorLetrasErradas;
    String letraEscolhida, msgPlacarErro, msgPlacarErroNumeros;
    List<String> listaLetrasValidas, palavraForcaArrayList,
placeholderArrayList, listaLetrasErradas, listaDicas;
    ArrayAdapter<String> adapterDicas;
    DesafioForca desafioForca;
}

```

```

@Override
public void onCreate(@Nullable Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_ingame);
    // pegando os elementos do layout
    imgForca = (ImageView) findViewById(R.id.imgForca);
    txtCategorias = (TextView) findViewById(R.id.txtCategorias);
    txtPalavraForca = (TextView) findViewById(R.id.txtPalavraForca);
    txtLetrasErradas = (TextView) findViewById(R.id.txtLetrasErradas);
    spnEscolherLetra = (Spinner) findViewById(R.id.spnEscolherLetra);
    btnConfirmarLetra = (Button) findViewById(R.id.btnConfirmarLetra);
    lvDicas = (ListView) findViewById(R.id.lvDicas);
    // pegando os dados necessários passados pela intent
    getElementosParaCriarForca();
    // inicializando variáveis de contagem, letras e listas
    contadorLetrasErradas = 0;
    listaLetrasValidas = listarLetrasValidas();
    palavraForcaArrayList = desafioForca.getArrayPalavraFormatado();
    msgPlacarErro = "Letras erradas: ";
    msgPlacarErroNumeros = "(0/6) ";
    // criando array para letras já erradas
    listaLetrasErradas = new ArrayList<>();
    // preenchendo os campos necessários
    txtCategorias.setText("Categoria: " + desafioForca.getCategoria());

    txtPalavraForca.setText(transformarPalavraPrimeiraContagem(palavraForcaArrayList));

    txtLetrasErradas.setText(msgPlacarErroNumeros + msgPlacarErro);
    // tratando do preenchimento do spinner
    spnEscolherLetra.setAdapter(preencherSpinner(listaLetrasValidas));
    // setando uma nova lista para armazenar as dicas
    listaDicas = new ArrayList<>();
    // setando adaptador para a lista de dicas
    adapterDicas = new ArrayAdapter<String>(InGameActivity.this,
        android.R.layout.simple_list_item_1, android.R.id.text1,
listaDicas);
    lvDicas.setAdapter(adapterDicas);
    btnConfirmarLetra.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            jogo();
        }
    });
}

private void getElementosParaCriarForca() {
    Intent it = getIntent();
    this.desafioForca = it.getParcelableExtra("forca");
}

// lógica do jogo em si
private void jogo() {
    this.letraEscolhida = spnEscolherLetra.getSelectedItem().toString();
    if(isLetraNaPalavra(letraEscolhida, palavraForcaArrayList)) {
        acertarLetra(letraEscolhida);
        if(isListasIguais(placeholderArrayList, palavraForcaArrayList)) {
            fimDeJogo("vitoria");
        }
    } else {
        if(contadorLetrasErradas >= 5)
            fimDeJogo("derrota");
        errarLetra(letraEscolhida);
    }
}

private String transformarPalavraPrimeiraContagem(List<String>

```

```

palavraArray) {
    // aqui guarda a quantidade de espaços para usar depois
    this.placeholderArrayList = new ArrayList<String>();
    String temp = "";
    for(String letra : palavraArray) {
        if(getListagemCaracteresInvalidos().contains(letra)) {
            this.placeholderArrayList.add(letra);
            temp += letra + " ";
        } else {
            this.placeholderArrayList.add("._.");
            temp += "._. ";
        }
    }
    return temp;
}

private boolean isLetraNaPalavra(String letra, List<String> palavra) {
    if(palavra.contains(letra))
        return true;
    return false;
}

private void acertarLetra(String letraEscolhida) {
    // verificando se a letra já foi usada antes
    if(isLetraUsada(letraEscolhida)) {
        Toast.makeText(InGameActivity.this, "Você já selecionou esta
letra", Toast.LENGTH_SHORT).show();
        return;
    }
    String novaPalavra = "";
    for(int i = 0; i < this.palavraForcaArrayList.size(); i++) {
        if (this.palavraForcaArrayList.get(i).equals(letraEscolhida)) {
            this.placeholderArrayList.set(i, letraEscolhida);
        }
    }
    for(String letra : this.placeholderArrayList)
        novaPalavra += letra + " ";
    this.txtPalavraForca.setText(novaPalavra);
    Toast.makeText(InGameActivity.this, "Você acertou!",
Toast.LENGTH_SHORT).show();
}

private boolean isListasIguais(List<String> listaA, List<String> listaB) {
    return listaA.equals(listaB);
}

private void errarLetra(String letraEscolhida) {
    // verificando se a letra já foi usada antes
    if(isLetraUsada(letraEscolhida)) {
        Toast.makeText(InGameActivity.this, "Você já selecionou esta
letra", Toast.LENGTH_SHORT).show();
        return;
    }
    atualizarPlacarErro(letraEscolhida);
    atualizarImagemForca();
    atualizarDicas();
    Toast.makeText(InGameActivity.this, "Você errou!",
Toast.LENGTH_SHORT).show();
}

private boolean isLetraUsada(String letra) {
    if(this.listaLetrasErradas.contains(letra)) {
        return true;
    }
    listaLetrasErradas.add(letra);
    return false;
}

private void atualizarPlacarErro(String letra) {

```



```

        this.contadorLetrasErradas++;
        this.msgPlacarErro += letra + " ";
        this.msgPlacarErroNumeros = "(" + contadorLetrasErradas + "/6) ";
        this.txtLetrasErradas.setText(msgPlacarErroNumeros + msgPlacarErro);
    }
    private void atualizarImagemForca() {
        switch (contadorLetrasErradas) {
            case 1 :
                imgForca.setImageResource(R.drawable.boneco_forca_02);
                break;
            case 2 :
                imgForca.setImageResource(R.drawable.boneco_forca_03);
                break;
            case 3 :
                imgForca.setImageResource(R.drawable.boneco_forca_04);
                break;
            case 4 :
                imgForca.setImageResource(R.drawable.boneco_forca_05);
                break;
            case 5 :
                imgForca.setImageResource(R.drawable.boneco_forca_06);
                break;
            default:
                break;
        }
    }
    private void atualizarDicas() {
        switch (contadorLetrasErradas) {
            case 1 :
                listaDicas.add(desafioForca.getDicas().get(0));
                adapterDicas.notifyDataSetChanged();
                break;
            case 2 :
                listaDicas.add(desafioForca.getDicas().get(1));
                adapterDicas.notifyDataSetChanged();
                break;
            case 3 :
                listaDicas.add(desafioForca.getDicas().get(2));
                adapterDicas.notifyDataSetChanged();
                break;
            case 4 :
                listaDicas.add(desafioForca.getDicas().get(3));
                adapterDicas.notifyDataSetChanged();
                break;
            case 5 :
                listaDicas.add(desafioForca.getDicas().get(4));
                adapterDicas.notifyDataSetChanged();
                break;
            default:
                break;
        }
    }
    // trata do preenchimento de listas de letras válidas e inválidas
    private List<String> listarLetrasValidas() {
        String[] validos = {"A", "B", "C", "D", "E", "F", "G", "H", "I", "J",
"K",
                        "L", "M", "N", "O", "P", "Q", "R", "S", "T", "U", "V", "W",
"X", "Y", "Z",
                        "0", "1", "2", "3", "4", "5", "6", "7", "8", "9",
                        "Á", "À", "Ã", "Â", "É", "Ê", "Í", "Ó", "Ô", "Ú", "Û", "Ç"};
        List<String> lista = arrayParaList(validos);
        return lista;
    }

```

```

    private List<String> getListagemCaracteresInvalidos() {
        String[] invalidos = {" ", "'", "\"", "!", "?", "@", "#", "$", "%",
        "",
                                "&", "+", "-", "*", "/", "^", "~", "`", " ",
                                "(", ")", "{", "}", "[", "]", " ", ".", ":";
        ";",
                                "|", "\\", "/", "a", "o", "o", "="};

        List<String> lista = arrayParaList(invalidos);
        return lista;
    }

    private List<String> arrayParaList(String[] arrayStrings) {
        List<String> lista = new ArrayList<>();
        for(int i = 0; i < arrayStrings.length; i++)
            lista.add(arrayStrings[i]);
        return lista;
    }

    // trata de preencher o spinner na inicialização
    private ArrayAdapter<String> preencherSpinner(List<String> lista) {
        ArrayAdapter<String> adapter = new ArrayAdapter<>(this,
        android.R.layout.simple_spinner_dropdown_item, lista);

        adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
        return adapter;
    }

    // tratando dos fins do jogo
    private void fimDeJogo(String resultado) {
        Intent it;
        if(resultado == "vitoria") {
            it = new Intent(InGameActivity.this, VitoriaActivity.class);
            startActivity(it);
        } else {
            it = new Intent(InGameActivity.this, DerrotaActivity.class);
            startActivity(it);
        }
        finish();
    }
}

```

ListaForcasActivity.java

```

package trabalhoandroid.com.br.inforcadin.activities;
import android.content.Intent;
import android.database.Cursor;
import android.os.Bundle;
import android.support.annotation.Nullable;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ListView;
import android.widget.SimpleCursorAdapter;
import trabalhoandroid.com.br.inforcadin.R;
import trabalhoandroid.com.br.inforcadin.banco.ForcaDao;
import trabalhoandroid.com.br.inforcadin.banco.ForcaHelperDao;
/**
 * Created by Artur on 21/05/2017.
 */
public class ListaForcasAcivity extends AppCompatActivity {
    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_lista_forcas);
        ForcaDao crud = new ForcaDao(getBaseContext());
    }
}

```

```

        final Cursor cursor = crud.buscar();
        String[] nomeCampos = new String[] {ForcaHelperDao.getNomeCategoria(),
ForcaHelperDao.getPALAVRA()};
        int[] idViews = new int[] {R.id.txtCategoriaLV, R.id.txtPalavraLV};
        SimpleCursorAdapter adaptador = new
SimpleCursorAdapter(getBaseContext(),
        R.layout.lista_de_forcas, cursor, nomeCampos, idViews, 0);
        ListView lista = (ListView) findViewById(R.id.lvBuscar);
        lista.setAdapter(adaptador);
        lista.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> parent, View view, int
position, long id) {
                String codigo;
                cursor.moveToPosition(position);
                codigo =
cursor.getString(cursor.getColumnIndexOrThrow(ForcaHelperDao.getID()));
                Intent intent = new Intent(ListaForcasAcivity.this,
CriaDesafioActivity.class);
                intent.putExtra("codigo", codigo);
                startActivity(intent);
                finish();
            }
        });
    }
}

```

MainActivity.java

```

package trabalhoandroid.com.br.inforcadin.activities;
import android.content.Intent;
import android.content.SharedPreferences;
import android.database.Cursor;
import android.preference.PreferenceManager;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;
import trabalhoandroid.com.br.inforcadin.R;
import trabalhoandroid.com.br.inforcadin.banco.ForcaDao;
import trabalhoandroid.com.br.inforcadin.banco.CriaDados;
public class MainActivity extends AppCompatActivity {
    public static final String KEY_PREFS_FIRST_LAUNCH = "first_launch";
    private Button btnIniciar, btnCriar;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        // criando algumas forcas para o app já ter algo no banco de dados
        quando for iniciado pela primeira vez
        checarPrimeiraVezECriarForcas();
        setContentView(R.layout.activity_main);
        btnIniciar = (Button) findViewById(R.id.btnJogarInicio);
        btnCriar = (Button) findViewById(R.id.btnCriarInicio);
        btnIniciar.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                ForcaDao crud = new ForcaDao(getBaseContext());
                Cursor cursor = crud.buscar();
                if(cursor.moveToFirst() == false) {
                    Toast.makeText(MainActivity.this, "Você precisa registrar
alguns desafios para brincar de forca.", Toast.LENGTH_SHORT).show();

```

```

        } else {
            Intent it = new Intent(MainActivity.this,
SelecionaCategoriaActivity.class);
            startActivity(it);
        }
    }
});
btnCriar.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent it = new Intent(MainActivity.this,
EscolhaNovaForcaActivity.class);
        startActivity(it);
    }
});
}

private void checarPrimeiraVezECriarForcas() {
    SharedPreferences pref =
PreferenceManager.getDefaultSharedPreferences(getApplicationContext());
    if (pref.getBoolean(KEY_PREFS_FIRST_LAUNCH, true)) {
        SharedPreferences.Editor editor = pref.edit();
        editor.putBoolean(KEY_PREFS_FIRST_LAUNCH, false);
        editor.commit();
        ForcaDao crud = new ForcaDao(getApplicationContext());
        CriaDados criaDados = new CriaDados(crud);
        criaDados.criar();
    }
}
}

```

SelecionaCategoriaActivity.java

```

package trabalhoandroid.com.br.inforcadin.activities;
import android.content.Intent;
import android.database.Cursor;
import android.os.Bundle;
import android.support.annotation.Nullable;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.Spinner;
import java.util.ArrayList;
import java.util.HashSet;
import java.util.List;
import java.util.Set;
import trabalhoandroid.com.br.inforcadin.R;
import trabalhoandroid.com.br.inforcadin.banco.ForcaDao;
import trabalhoandroid.com.br.inforcadin.banco.ForcaHelperDao;
import trabalhoandroid.com.br.inforcadin.game.DesafioForca;
/**
 * Created by Artur on 21/05/2017.
 */
public class SelecionaCategoriaActivity extends AppCompatActivity {
    private Spinner spnCategorias;
    private Button btnJogar;
    private ForcaDao crud;
    private ArrayList<String> categorias;
    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_seleciona_categoria);
    }
}

```

```

        spnCategorias = (Spinner) findViewById(R.id.spinnerSelecionar);
        btnJogar = (Button) findViewById(R.id.btnJogar);
        crud = new ForcaDao(getApplicationContext());
        preencherSpinner();
        btnJogar.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                DesafioForca forca = randomizarDesafio();
                Intent it = new Intent(SelecionaCategoriaActivity.this,
InGameActivity.class);
                it.putExtra("forca", forca);
                startActivity(it);
                finish();
            }
        });
    }

    private void preencherSpinner() {
        final Cursor cursor = crud.buscar();
        Set<String> catSet = new HashSet<>();
        categorias = new ArrayList<>();
        for(cursor.moveToFirst(); (!cursor.isAfterLast()); cursor.moveToNext())
        {
            catSet.add(cursor.getString(1).toString());
        }
        for(String set : catSet) {
            categorias.add(set);
        }
        ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,
android.R.layout.simple_spinner_dropdown_item, categorias);

        adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
        spnCategorias.setAdapter(adapter);
    }

    private DesafioForca randomizarDesafio() {
        DesafioForca desafioForca;
        String categoria = spnCategorias.getSelectedItem().toString();
        Cursor cursor = crud.buscarByCategoriaRandom(categoria);
        String cat =
cursor.getString(cursor.getColumnIndexOrThrow(ForcaHelperDao.getNomeCategoria()
));
        String palavra =
cursor.getString(cursor.getColumnIndexOrThrow(ForcaHelperDao.getPALAVRA()));
        List<String> dicas = new ArrayList<>();

        dicas.add(cursor.getString(cursor.getColumnIndexOrThrow(ForcaHelperDao.getDica1
())));

        dicas.add(cursor.getString(cursor.getColumnIndexOrThrow(ForcaHelperDao.getDica2
())));

        dicas.add(cursor.getString(cursor.getColumnIndexOrThrow(ForcaHelperDao.getDica3
())));

        dicas.add(cursor.getString(cursor.getColumnIndexOrThrow(ForcaHelperDao.getDica4
())));

        dicas.add(cursor.getString(cursor.getColumnIndexOrThrow(ForcaHelperDao.getDica5
())));

        desafioForca = new DesafioForca(cat);
        desafioForca.setPalavra(palavra);
        desafioForca.setDicas(dicas);
        return desafioForca;
    }

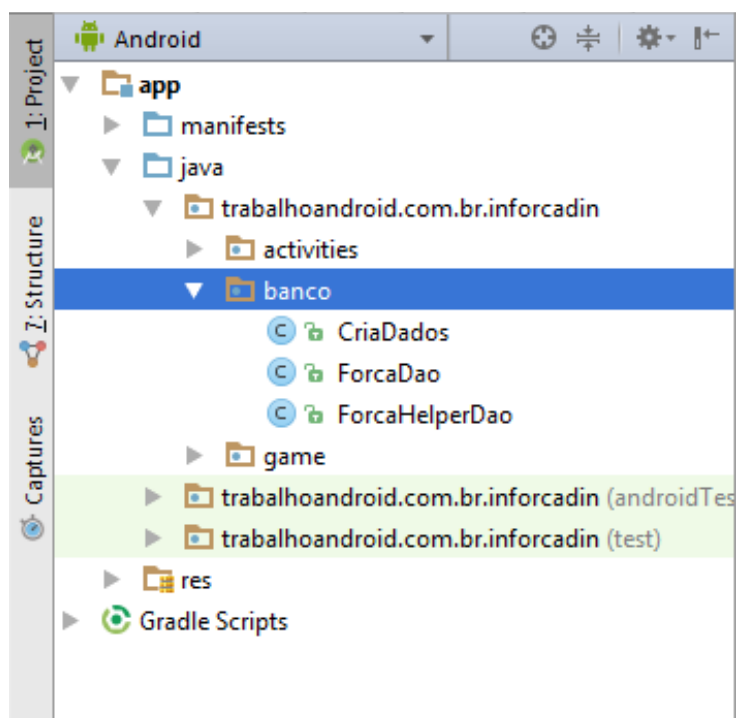
```

```
}
```

VitoriaActivity.java

```
package trabalhoandroid.com.br.inforcadin.activities;
import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import trabalhoandroid.com.br.inforcadin.R;
public class VitoriaActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_vitoria);
        Button btnJogarNovamente = (Button)
findViewById(R.id.btnJogarNovamente);
        btnJogarNovamente.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent it = new Intent(VitoriaActivity.this,
SelecionaCategoriaActivity.class);
                startActivity(it);
                finish();
            }
        });
    }
}
```

- No diretório banco há as classes responsáveis por conexão e manipulação do banco de dados (SQLite) e a classe Mockup para criação de dados padrão quando a aplicação é iniciada pela primeira vez.



CriaDados.java

```
package trabalhoandroid.com.br.inforcadin.banco;
/**
 * Created by Artur on 21/05/2017.
 */
public class CriaDados {
    ForcaDao forcaDao;
    public CriaDados(ForcaDao forcaDao) {
        this.forcaDao = forcaDao;
    }
    public void criar() {
        forcaDao.inserir("Filme",
            "Vingadores",
            "Lançado em 2012.",
            "Direção de Joss Whedon.",
            "A eterna luta entre o bem e o mal.",
            "Tem mais de um herói.",
            "Baseado em uma equipe de histórias em quadrinhos.");
        forcaDao.inserir("Filme",
            "O Profissional",
            "Lançado em 1995.",
            "Dirigido por Luc Besson",
            "No filme tem um assassino.",
            "Os vizinhos morrem.",
            "Um dos principais é uma menina.");
        forcaDao.inserir("Filme",
            "O Senhor dos Anéis",
            "Existe uma animação do mesmo tema de 1979.",
            "Não é uma Terra muito grande...",
            "Alguns habitantes são pequenos.",
            "Dirigido por Peter Jackson.",
            "No nome tem muitos mas o filme é sobre um só.");
        forcaDao.inserir("Série",
            "Mr. Robot",
            "É uma série meio dramática.",
            "A primeira temporada tem 10 episódios.",
            "Criada por Sam Esmail.",
            "O protagonista não é muito descolado.",
            "Assuntos Hackers.");
        forcaDao.inserir("Série",
            "Todo mundo odeia o Chris",
            "Conta a história de um famoso.",
            "Tem 4 temporadas.",
            "É sobre uma família.",
            "É de Humor.",
            "As pessoas não gostam muito dele.");
        forcaDao.inserir("Série",
            "Eu a patroa e as crianças",
            "Criada por Damon Wayans e Don Reo.",
            "É de Humor.",
            "Tem 5 temporadas.",
            "Criada em 2001.",
            "É uma sitcom.");
        forcaDao.inserir("Quadrinho",
            "Wolverine",
            "Ele é um pouco nervoso.",
            "Seus ossos devem ser pesados.",
            "Primeira aparição foi brigando contra um cara bem forte.",
            "Faz parte de um grupo de heróis.",
            "Se cura muito rápido.");
        forcaDao.inserir("Quadrinho",
            "Homem-Aranha",
            "Gosta de pular de um lado para o outro.",
```

```

        "Gosta de fazer piadas.",
        "Tem vermelho no uniforme.",
        "Sua primeira aparição foi em Amazing Fantasy nº15.",
        "Seu nome do meio é Benjamin, quem diria hein?");
forcaDao.inserir("Quadrinho",
        "Batman",
        "Gosta de botar medo.",
        "Inteligente até demais.",
        "Sabe várias artes-marciais.",
        "Às vezes é um tanto arrogante.",
        "Tem uma fundação com seu nome.");
    }
}

```

ForcaDao.java

```

package trabalhoandroid.com.br.inforcadin.banco;
import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
/**
 * Created by Artur on 20/05/2017.
 */
public class ForcaDao {
    private SQLiteDatabase db;
    private ForcaHelperDao banco;
    public ForcaDao(Context context) {
        banco = new ForcaHelperDao(context);
    }
    public String inserir(String categoria, String palavra,
        String dica1, String dica2, String dica3,
        String dica4, String dica5) {
        ContentValues valores;
        long resultado;
        db = banco.getWritableDatabase();
        valores = new ContentValues();
        valores.put(ForcaHelperDao.getNomeCategoria(), categoria);
        valores.put(ForcaHelperDao.getPALAVRA(), palavra);
        valores.put(ForcaHelperDao.getDica1(), dica1);
        valores.put(ForcaHelperDao.getDica2(), dica2);
        valores.put(ForcaHelperDao.getDica3(), dica3);
        valores.put(ForcaHelperDao.getDica4(), dica4);
        valores.put(ForcaHelperDao.getDica5(), dica5);
        resultado = db.insert(ForcaHelperDao.getTABELA(), null, valores);
        db.close();
        if(resultado == -1)
            return "Erro ao inserir registro.";
        else
            return "Registro inserido com sucesso!";
    }
    public Cursor buscar() {
        Cursor cursor;
        String[] campos = {ForcaHelperDao.getID(),
ForcaHelperDao.getNomeCategoria(), ForcaHelperDao.getPALAVRA()};
        db = banco.getReadableDatabase();
        cursor = db.query(ForcaHelperDao.getTABELA(), campos, null, null, null,
null, null, null);
        if(cursor != null)
            cursor.moveToFirst();
        db.close();
        return cursor;
    }
}

```



```

    }
    public Cursor buscarById(int id) {
        Cursor cursor;
        String[] campos = {ForcaHelperDao.getID(),
ForcaHelperDao.getNomeCategoria(),
        ForcaHelperDao.getPALAVRA(), ForcaHelperDao.getDica1(),
ForcaHelperDao.getDica2(),
        ForcaHelperDao.getDica3(), ForcaHelperDao.getDica4(),
ForcaHelperDao.getDica5()
        };
        String where = ForcaHelperDao.getID() + " = " + id;
        db = banco.getReadableDatabase();
        cursor = db.query(ForcaHelperDao.getTABELA(), campos, where, null,
null, null, null, null);
        if(cursor != null)
            cursor.moveToFirst();
        db.close();
        return cursor;
    }
    public Cursor buscarByCategoriaRandom(String categoria) {
        String sql = "select * from " + ForcaHelperDao.getTABELA() + " where "
+ ForcaHelperDao.getNomeCategoria()
        + " = '" + categoria + "' order by random() limit 1";
        db = banco.getReadableDatabase();
        Cursor cursor = db.rawQuery(sql, null);
        if(cursor != null)
            cursor.moveToFirst();
        db.close();
        return cursor;
    }
    public void alterar(int id, String categoria, String palavra,
        String dica1, String dica2, String dica3,
        String dica4, String dica5) {
        ContentValues valores;
        String where;
        db = banco.getWritableDatabase();
        where = ForcaHelperDao.getID() + " = " + id;
        valores = new ContentValues();
        valores.put(ForcaHelperDao.getNomeCategoria(), categoria);
        valores.put(ForcaHelperDao.getPALAVRA(), palavra);
        valores.put(ForcaHelperDao.getDica1(), dica1);
        valores.put(ForcaHelperDao.getDica1(), dica2);
        valores.put(ForcaHelperDao.getDica1(), dica3);
        valores.put(ForcaHelperDao.getDica1(), dica4);
        valores.put(ForcaHelperDao.getDica1(), dica5);
        db.update(ForcaHelperDao.getTABELA(), valores, where, null);
        db.close();
    }
    public void deletar(int id) {
        String where = ForcaHelperDao.getID() + " = " + id;
        db = banco.getReadableDatabase();
        db.delete(ForcaHelperDao.getTABELA(), where, null);
        db.close();
    }
}

```

ForcaHelperDao.java

```

package trabalhoandroid.com.br.inforcadin.banco;
import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

```

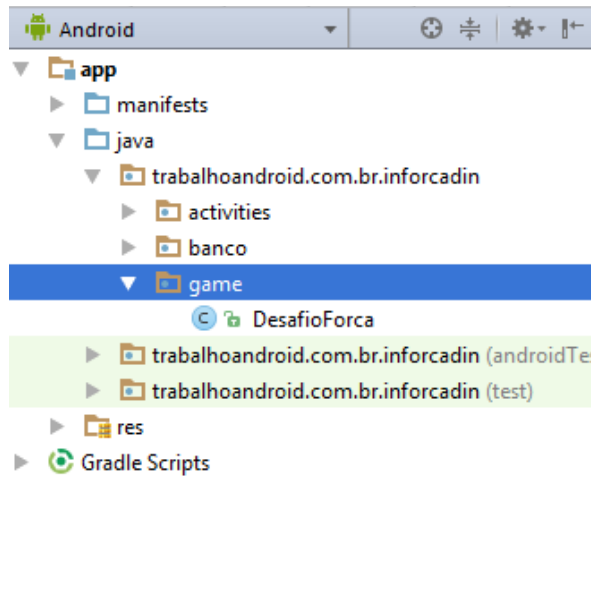
```

/**
 * Created by Artur on 20/05/2017.
 */
public class ForcaHelperDao extends SQLiteOpenHelper {
    private static final String NOME_BANCO = "inforcadin.db";
    private static final String ID = "_id";
    private static final int VERSAO = 1;
    private static final String TABELA = "forca";
    private static final String NOME_CATEGORIA = "nome_categoria";
    private static final String PALAVRA = "palavra";
    private static final String DICA_1 = "dica_1";
    private static final String DICA_2 = "dica_2";
    private static final String DICA_3 = "dica_3";
    private static final String DICA_4 = "dica_4";
    private static final String DICA_5 = "dica_5";
    public ForcaHelperDao(Context context) {
        super(context, NOME_BANCO, null, VERSAO);
    }
    @Override
    public void onCreate(SQLiteDatabase db) {
        String sql = "CREATE TABLE " + TABELA + "("
            + ID + " integer primary key autoincrement, "
            + NOME_CATEGORIA + " text, "
            + PALAVRA + " text, "
            + DICA_1 + " text, "
            + DICA_2 + " text, "
            + DICA_3 + " text, "
            + DICA_4 + " text, "
            + DICA_5 + " text"
            + ")";
        db.execSQL(sql);
    }
    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        db.execSQL("DROP TABLE IF EXISTS " + TABELA);
        onCreate(db);
    }
    public static String getID() {
        return ID;
    }
    public static String getTABELA() {
        return TABELA;
    }
    public static String getNomeCategoria() {
        return NOME_CATEGORIA;
    }
    public static String getPALAVRA() {
        return PALAVRA;
    }
    public static String getDica1() {
        return DICA_1;
    }
    public static String getDica2() {
        return DICA_2;
    }
    public static String getDica3() {
        return DICA_3;
    }
    public static String getDica4() {
        return DICA_4;
    }
    public static String getDica5() {
        return DICA_5;
    }
}

```

```
}  
}
```

- No diretório game há a classe de modelo de um objeto forca*.



* a ideia original era que esta pasta comportasse todas as classes de lógica do game em si, mas por questões de simplificação e quantidade de arquivos essas lógicas ficaram no arquivo InGameActivity.java.

DesafioForca.java

```
package trabalhoandroid.com.br.inforcadin.game;  
import android.os.Parcel;  
import android.os.Parcelable;  
import java.util.ArrayList;  
import java.util.List;  
/**  
 * Created by Artur on 18/05/2017.  
 */  
public class DesafioForca implements Parcelable {  
    private String categoria;  
    private String palavra;  
    private List<String> dicas;  
    public DesafioForca(String categoria) {  
        setCategoria(categoria);  
    }  
    protected DesafioForca(Parcel in) {  
        categoria = in.readString();  
        palavra = in.readString();  
        dicas = in.createStringArrayList();  
    }  
    public static final Creator<DesafioForca> CREATOR = new  
    Creator<DesafioForca>() {  
        @Override  
        public DesafioForca createFromParcel(Parcel in) {  
            return new DesafioForca(in);  
        }  
        @Override  
        public DesafioForca[] newArray(int size) {
```

```

        return new DesafioForca[size];
    }
};
public String getCategoria() {
    return categoria;
}
private void setCategoria(String categoria) {
    this.categoria = categoria;
}
public String getPalavra() {
    return palavra;
}
public void setPalavra(String palavra) {
    this.palavra = palavra;
}
public List<String> getDicas() {
    return dicas;
}
public void setDicas(List<String> dicas) {
    this.dicas = dicas;
}
public List<String> getArrayPalavraFormatado() {
    List<String> palavraArray = new ArrayList<>();
    for(int i = 0; i < getPalavra().length(); i++) {
        palavraArray.add("'" + getPalavra().toUpperCase().charAt(i));
    }
    return palavraArray;
}
@Override
public boolean equals(Object o) {
    if (this == o) return true;
    if (o == null || getClass() != o.getClass()) return false;
    DesafioForca desafioForca = (DesafioForca) o;
    if (!categoria.equals(desafioForca.categoria)) return false;
    if (palavra != null ? !palavra.equals(desafioForca.palavra) :
desafioForca.palavra != null)
        return false;
    return dicas != null ? dicas.equals(desafioForca.dicas) :
desafioForca.dicas == null;
}
@Override
public int hashCode() {
    int result = categoria.hashCode();
    result = 31 * result + (palavra != null ? palavra.hashCode() : 0);
    result = 31 * result + (dicas != null ? dicas.hashCode() : 0);
    return result;
}
@Override
public int describeContents() {
    return 0;
}
@Override
public void writeToParcel(Parcel dest, int flags) {
    dest.writeString(categoria);
    dest.writeString(palavra);
    dest.writeStringList(dicas);
}
}

```