**[Como criar uma simples câmera fotográfica em Kotlin](https://uware.com.br/como-criar-uma-simples-camera-fotografica-em-kotlin/)**

 14 de outubro de 2019

[Rodrigo Leutz](https://uware.com.br/author/rleutz/)

[Android](https://uware.com.br/category/android/) / [Kotlin](https://uware.com.br/category/kotlin/)

Nesse tutorial vou descrever Como criar uma simples câmera fotográfica em Kotlin para que tire uma foto e exiba em um campo de ImageView de uma forma bem simples.

Assim sendo usada a biblioteca [MediaStore](https://developer.android.com/reference/android/provider/MediaStore) do android.

**Como criar uma simples câmera fotográfica em Kotlin**

Primeiramente vamos começar criando um projeto vazio(Empty).

Depois vamos fazer umas mudanças no gradle no nivel de app adicionando a seguinte linha para utilizar o material design colocando um Floating Action Button.

**gradle/app**

**build.gradle(Module: app)**

dependencies {

...

implementation 'com.google.android.material:material:1.1.0-alpha10'

}

Vamos **sincronizar o gradle** para que ele inclua as bibliotecas do material design.

Logo após adicionarmos no gradle o material design vamos também adicionar no Manifest as permissões necessárias para poder executar a câmera. Insira as seguintes linhas acima da parte de application.

**AndroidManifest**

**AndroidManifest.xml**

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

Assim que tiver adicionado nossas linhas no gradle e no Manifest portanto podemos começar a programar e fazer com que nosso aplicativo bata fotos e as imprima em uma ImageView.

As fotos que são tiradas dessa maneira são de uma qualidade para **miniaturas**.

Se for necessário de uma qualidade maior então de uma olhada em [EXTRA\_OUTPUT](https://developer.android.com/reference/android/provider/MediaStore.html#ACTION_IMAGE_CAPTURE).

Assim sendo vamos a nossos blocos de programação para fazer a mágica acontecer.

**activity\_main**

**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:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".MainActivity">

<ImageView

android:id="@+id/imgPhoto"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:adjustViewBounds="true"

app:layout\_constraintTop\_toTopOf="parent"/>

<com.google.android.material.floatingactionbutton.FloatingActionButton

android:id="@+id/btnTakePhoto"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_margin="16dp"

android:src="@drawable/ic\_camera"

app:backgroundTint="#0000FF"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintBottom\_toBottomOf="parent"/>

</androidx.constraintlayout.widget.ConstraintLayout>

**MainActivity**

**MainActivity.kt**

package br.com.uware.camera

import android.Manifest

import android.app.Activity

import android.content.Intent

import android.content.pm.PackageManager

import android.graphics.Bitmap

import androidx.appcompat.app.AppCompatActivity

import android.os.Bundle

import android.util.Log

import android.widget.Toast

import androidx.core.app.ActivityCompat

import androidx.core.content.ContextCompat

import kotlinx.android.synthetic.main.activity\_main.\*

class MainActivity : AppCompatActivity() {

private var check: Boolean = false

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_main)

initPermissions()

btnTakePhoto.setOnClickListener {

if(check) {

val intent = Intent("android.media.action.IMAGE\_CAPTURE")

startActivityForResult(intent, 0)

}

else{

Toast.makeText(this, "Não tem permissão para usar a câmera.", Toast.LENGTH\_SHORT).show()

}

}

}

override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {

super.onActivityResult(requestCode, resultCode, data)

if (data != null && requestCode == 0 && resultCode == Activity.RESULT\_OK) {

val bundle: Bundle? = data.extras

if (bundle != null) {

val bitmap = bundle.get("data") as Bitmap

imgPhoto.setImageBitmap(bitmap)

}

}

}

// Permissions

private fun initPermissions(){

if(!getPermission()) setPermission()

else check = true

}

private fun getPermission(): Boolean{

return (ContextCompat.checkSelfPermission(this, Manifest.permission.CAMERA) == PackageManager.PERMISSION\_GRANTED)

}

private fun setPermission(){

val permissionsList = listOf<String>(

Manifest.permission.CAMERA

)

ActivityCompat.requestPermissions(this, permissionsList.toTypedArray(), 1)

}

override fun onRequestPermissionsResult(

requestCode: Int,

permissions: Array<out String>,

grantResults: IntArray

) {

when(requestCode){

1 -> {

if (grantResults.isEmpty() || grantResults[0] != PackageManager.PERMISSION\_GRANTED) {

Log.i("Permission: ", "Permission has been denied by user")

Toast.makeText(this, "Não tem permissão para usar a câmera.", Toast.LENGTH\_SHORT).show()

} else {

Log.i("Permission: ", "Permission has been granted by user")

check = true

}

}

}

}

}