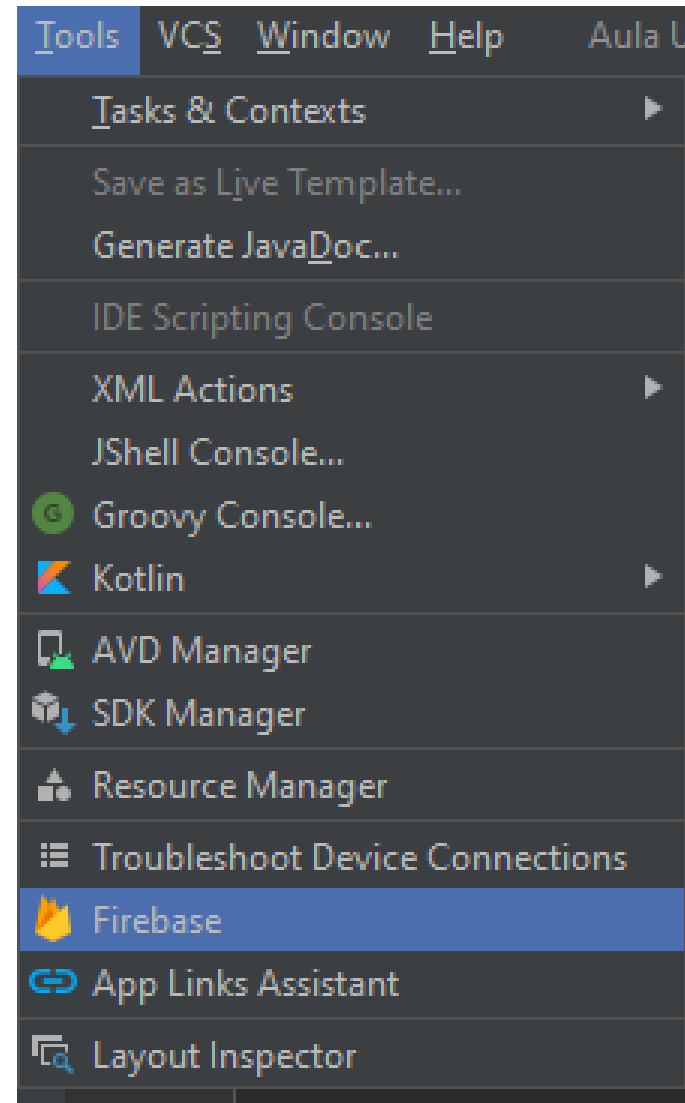


Enviar imagens para o firebase Storage

Prof. Me. Hélio Esperidião

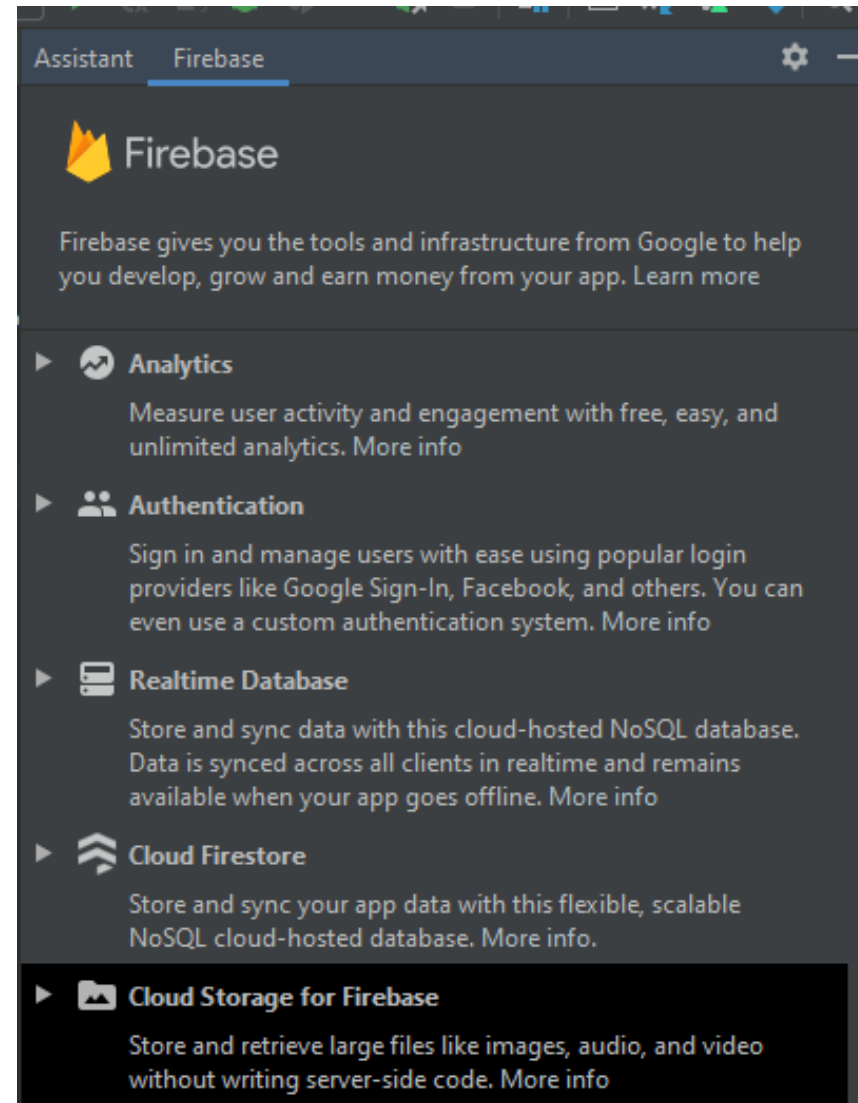
Configuração firebase

- Acesse o assistente do firebase no menu tools



Configuração firebase

- Selecione a opção cloud storage for firebase





Cloud Storage for Firebase

Store and retrieve large files like images, audio, and video without writing server-side code. [More info](#)



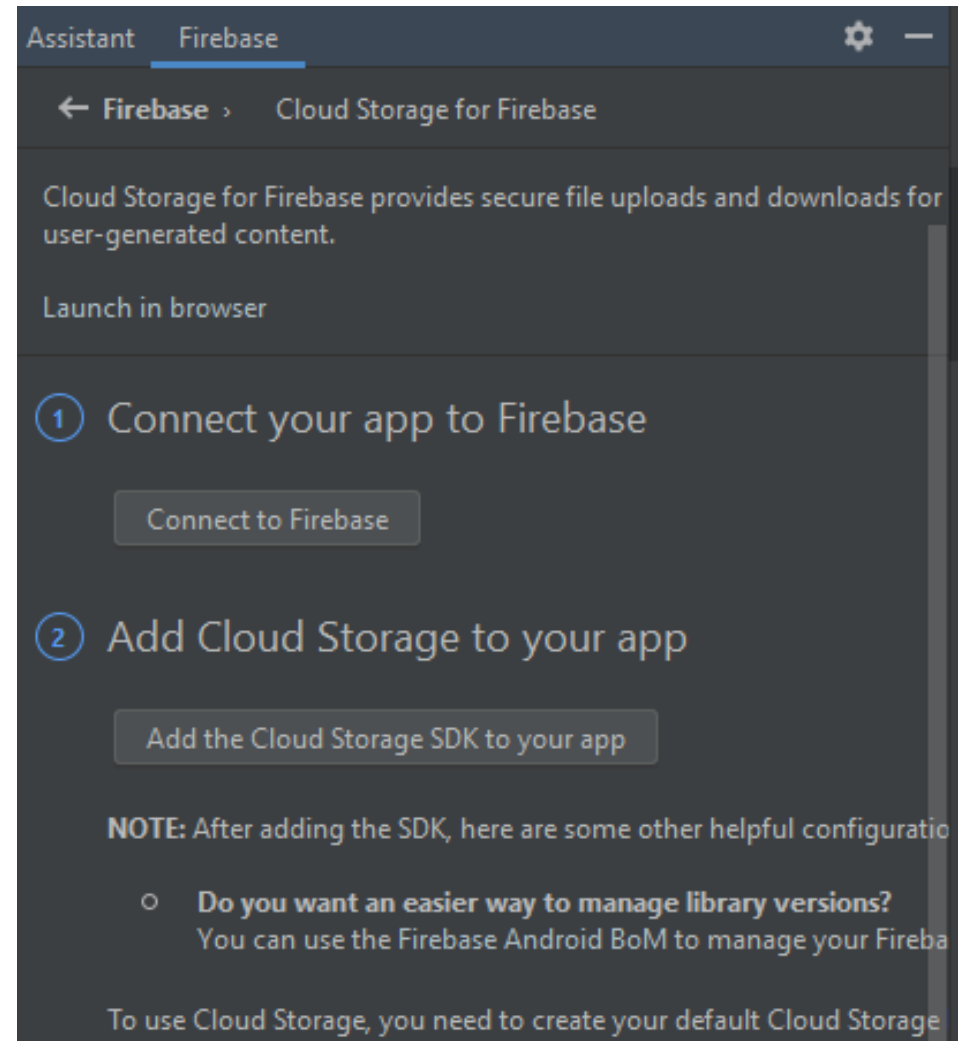
[Get started with Cloud Storage](#)



[Get started with Cloud Storage \[KOTLIN\]](#)

Conecte

- Conecte seu projeto kotlin a um projeto no fireabase





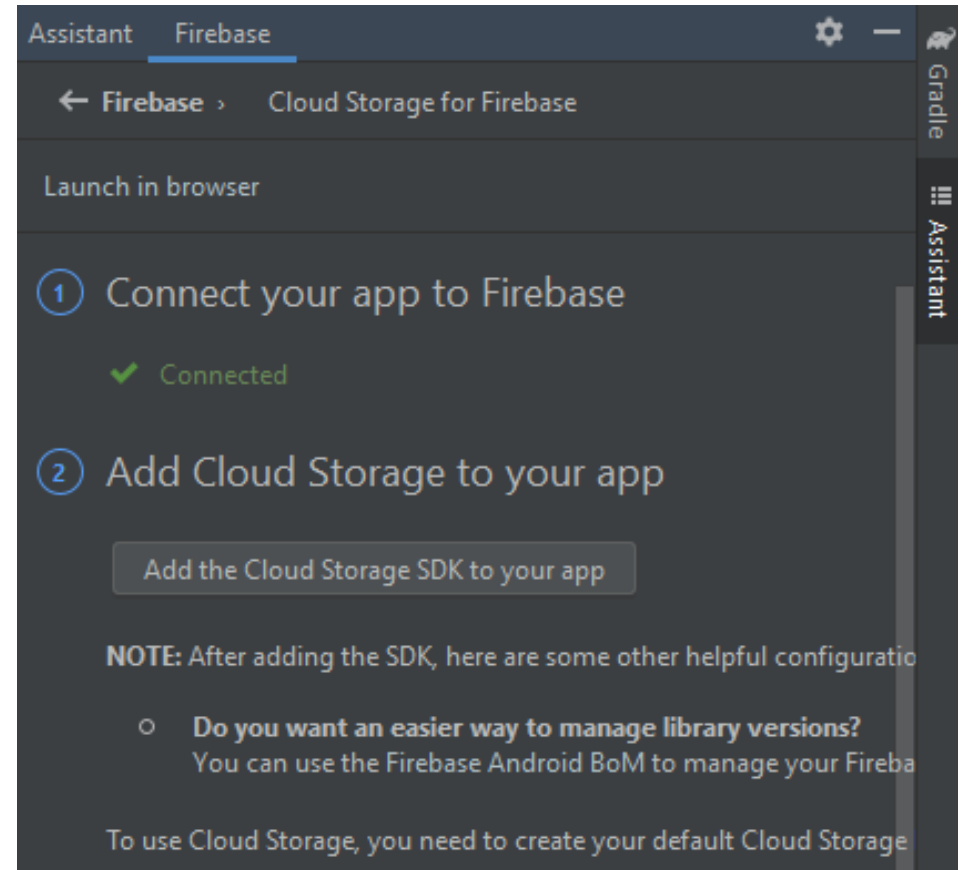
Seu app Android foi criado no Firebase.

Ele está pronto para ser conectado ao projeto do Android Studio.

[Conectar](#)

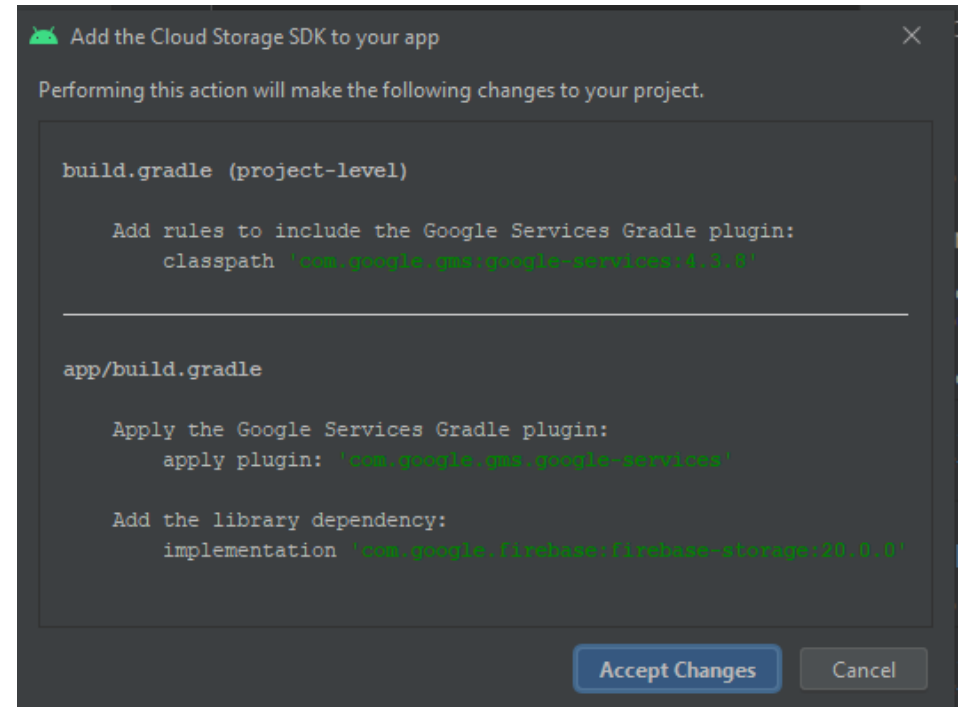
Configure o sdk

- Selecione a opção “add the cloud storage sdk to your app”

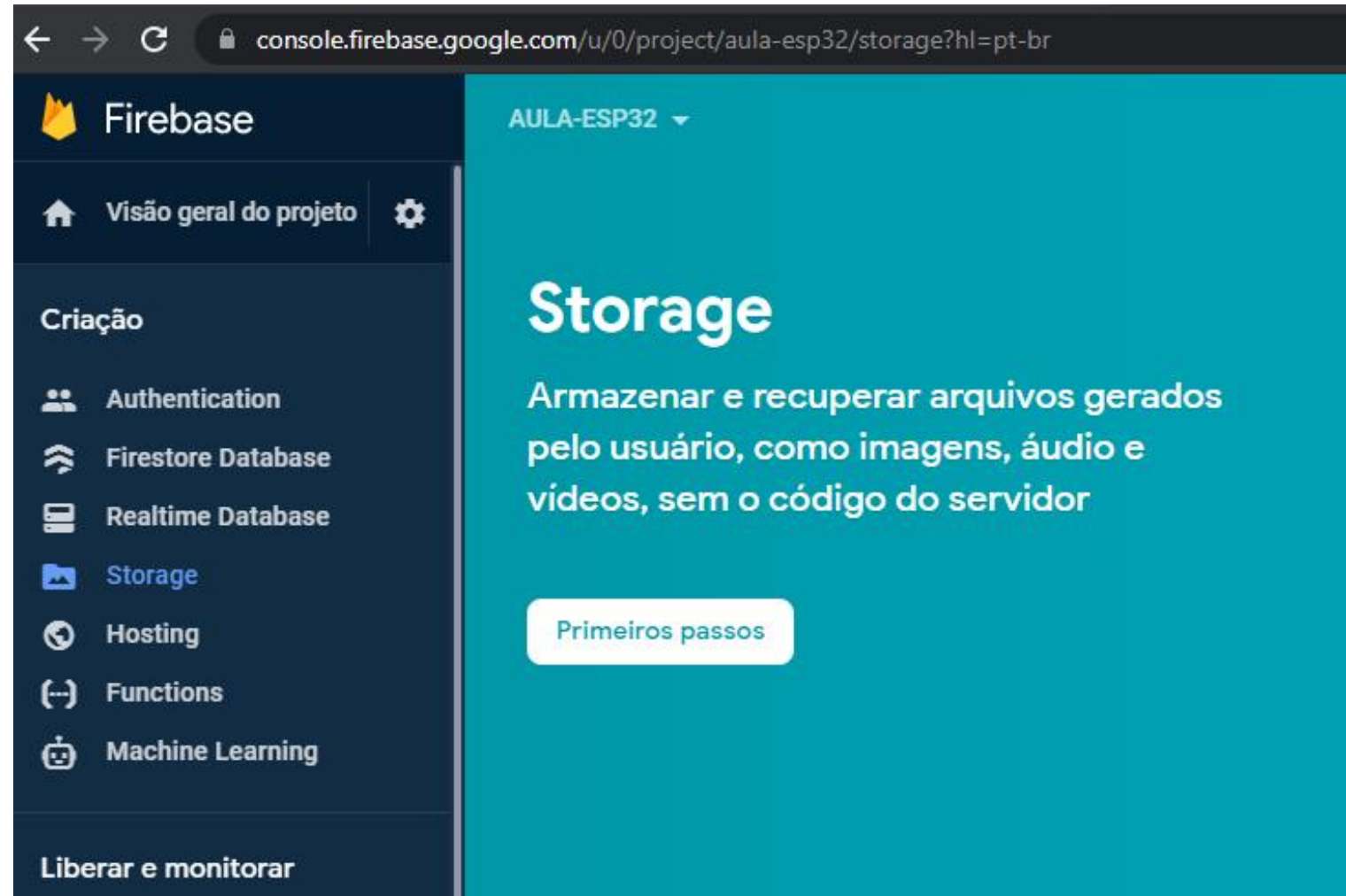


Aceite as mudanças

- Espere que o projeto seja configurado



Ative storage
no firebase



The screenshot shows the Firebase console interface. The browser address bar at the top displays the URL: `console.firebase.google.com/u/0/project/aula-esp32/storage?hl=pt-br`. The left sidebar features the Firebase logo and a navigation menu with the following items: "Visão geral do projeto" (with a home icon and settings gear), "Criação" (Creation), "Authentication", "Firestore Database", "Realtime Database", "Storage" (highlighted in blue), "Hosting", "Functions", and "Machine Learning". Below these is the section "Liberar e monitorar" (Release and monitor). The main content area has a teal background and displays the project name "AULA-ESP32" with a dropdown arrow. The title "Storage" is prominently shown, followed by the description: "Armazenar e recuperar arquivos gerados pelo usuário, como imagens, áudio e vídeos, sem o código do servidor". A white button labeled "Primeiros passos" (First steps) is positioned at the bottom of the main content area.

Configurações de acesso

Configurar o Cloud Storage

1 Regras seguras para o Cloud Storage

2 Definir local do Cloud Storage

Por padrão, suas regras permitem todas as leituras e gravações de usuários autenticados.

Após definir a estrutura de dados, será necessário criar regras para proteger seus dados. [Saiba mais](#)

```
service firebase.storage {
  match /b/{bucket}/o {
    match /{allPaths=**} {
      allow read, write: if request.auth != null;
    }
  }
}
```

Cancelar

Próxima



Configurar o Cloud Storage



Regras seguras para o Cloud Storage



Definir local do Cloud Storage

A configuração do local é o lugar em que o bucket e os dados do Cloud Storage padrão serão armazenados.



Não será possível alterar o local depois de configurá-lo. Esta configuração de local também será o local padrão do Cloud Firestore.

[Saiba mais](#)

Local no Cloud Storage

southamerica-east1

Os clientes do plano Blaze podem selecionar outros locais para intervalos adicionais

Cancelar

Concluir



Storage

Files Rules Usage

Proteja os recursos do Storage de abusos, como fraude de faturamento ou phishing [Configurar o App Check](#) ✕

[gs://aula-esp32.appspot.com](#) [Fazer upload do arquivo](#) + ⋮

<input type="checkbox"/>	Name	Tamanho	Tipo	Última modificação
Ainda não há arquivos aqui				

Regras de acesso



Proteja seus dados com regras que definem quem pode acessá-los e como eles são estruturados

[Ver os documentos](#)

Editar regras

Monitorar regras

alterações não publicadas

Publicar

Descartar

```
1 rules_version = '2';
2 service firebase.storage {
3   match /b/{bucket}/o {
4     match /{allPaths=**} {
5       allow read, write: if request.auth != null;
6     }
7   }
8 }
9
```

Modifique
para testar

Storage

Files Rules Usage

Editar regras

Monitorar regras



Proteja seus dados com regras que definem quem pode acessá-los e como eles são estruturados

```
1 rules_version = '2';  
2 service firebase.storage {  
3   match /b/{bucket}/o {  
4     match /{allPaths=**} {  
5       allow read, write;  
6     }  
7   }  
8 }  
9
```

Selecionar Imagem

```
private fun selecionarImagemGaleria() {  
    val intent = Intent()  
    intent.type = "image/*"  
    intent.action = Intent.ACTION_GET_CONTENT  
    val PICK_IMAGE_REQUEST = 22 //código para identificar resposta de intent  
    startActivityForResult(Intent.createChooser(intent, "Selecione"), PICK_IMAGE_REQUEST)  
}
```

downloadImage

```
fun downloadImage(caminho:String){
    val storageReference = FirebaseStorage.getInstance().reference.child(caminho)
    val localFile = File.createTempFile("image", "jpg")
    storageReference.getFile(localFile)
        .addOnSuccessListener { taskSnapshot ->
            val bitmap = BitmapFactory.decodeFile(localFile.absolutePath)
            var img: ImageView = findViewById(R.id.imageView)
            img.setImageBitmap(bitmap)
        }
        .addOnFailureListener { exception ->
            Log.e("TAG", "Error downloading image", exception)
        }
}
```


uploadImage

```
private fun uploadImage(caminhoArquivo: Uri) {
    if (caminhoArquivo != null) {
        val progressDialog = ProgressDialog(this)
        progressDialog.setTitle("Enviando...")
        progressDialog.show()
        var caminhoImagem = "images/" + UUID.randomUUID().toString()
        val ref: StorageReference = FirebaseStorage.getInstance().getReference().child(caminhoImagem)
        ref.putFile(caminhoArquivo!!).addOnSuccessListener {
            progressDialog.dismiss()
            Toast.makeText(this@MainActivity, "Imagem enviada", Toast.LENGTH_SHORT).show()
        }.addOnFailureListener { e -> progressDialog.dismiss()
            Toast.makeText(this@MainActivity, "Erro ao enviar " + e.message, Toast.LENGTH_SHORT).show()
        }.addOnProgressListener { taskSnapshot ->
            val progress = (100.0 * taskSnapshot.bytesTransferred / taskSnapshot.totalByteCount)
            progressDialog.setMessage("Enviando " + progress.toInt() + "%")
        }
    }
}
```

Evento disparado após usuário selecionar ou não imagem na galeria

```
override fun onActivityResult(requestCode: Int,resultCode: Int,data: Intent?) {
    super.onActivityResult( requestCode,resultCode,data )
    val PICK_IMAGE_REQUEST = 22 //código para identificar resposta de intent
    if (requestCode == PICK_IMAGE_REQUEST && resultCode == Activity.RESULT_OK && data != null && data.data != null) {
        var caminhoDaImagem = data.data
        try {
            val bitmap = MediaStore.Images.Media.getBitmap(contentResolver,caminhoDaImagem)
            var img: ImageView = findViewById(R.id.imgView)
            img.setImageBitmap(bitmap)
            if (caminhoDaImagem != null) {
                uploadImage(caminhoDaImagem)
            }
        }catch (e: IOException) {
            e.printStackTrace()
        }
    }
}
```

Storage



[Files](#)[Rules](#)[Usage](#)[Extensões](#) **NOVO**

Proteja os recursos do Storage de abusos, como fraude de faturamento ou phishing

[Configurar o App Check](#)

[gs://aula-paw.appspot.com](#)

[Fazer upload do arquivo](#)

<input type="checkbox"/>	Name	Tamanho	Tipo	Última modificação
<input type="checkbox"/>	 images/	—	Pasta	—
<input type="checkbox"/>	 eu.jpg	34.87 KB	image/jpeg	7 de mai. de 2023

 eu.jpg



Nome

[eu.jpg](#)

Tamanho

35.711 bytes

Tipo

image/jpeg