Agora vamos enviar essas imagens direto para o Storage

Storage = onde voce vai armazenar as imagens, para poder depois usar o link.

Firestore = onde vamos armazenar o a URL do usuário

Imagina o Facebook

Postamos nossa foto, ele guarda a imagem dentro do Storage

E guardamos a URL dentro do nosso perfil(Firestore)

Quando queremos usar pegamos a URL que esta dentro do nosso perfil e usamos ex: Picaso para trazer converter essa URL para imageView.

Pegando o local da imagem e armazenando

class UploadImagemActivity : AppCompatActivity() {  
  
 private val binding by *lazy* **{**ActivityUploadImagemBinding.inflate(*layoutInflater*)**}** /\*1 Primeiro vamos criar esse atributo \*/  
 private var uriImagemSelecionada: Uri? = null  
  
  
 private val abrirGaleria = registerForActivityResult(  
 */\*\* Oque esse codigo faz? ele espera abrir uma acitivty de resultado  
 \* Tinhamos tambem esse outro metodo, mas foi depreciado.  
 \* Tradução do registerForActivityResult  
 \* Registrar para um resultado de activity  
 \*  
 \* 2 Aqui dentro do metodo precisamos passar oque vai retornar como resultado  
 \* GetContent = usuario vai pegar um documento || eu vou pegar conteudos.  
 \*  
 \*  
 \* 3 uri -> esse é o endereço que esta nossa foto, é onde esta nossa foto na galeria.  
 \* Quando abrimos uma imagem, ela vai retornar o local onde esta essa imagem  
 \* ex: c://computador//alleph//imagem//fotopraia.png  
 \*  
 \* 4 Devemos fazer um teste na URI porque ela pode ser nula também  
 \* Se encontrar a imagem podemos usar ela, porque é bem simples.  
 \*/* ActivityResultContracts.GetContent()  
 ) **{** uri **->** if(uri != null) {  
 **/\*2 Aqui vamos passar esse atributo   
 \* Agora dentro de imagemSelecionada ja vamos ter o local de onde esta essa imagem  
 \* Agora vamos fazer o upload dessa imagem \*/  
 uriImagemSelecionada = uri**  
 binding.imageSelecionada.setImageURI(uri)  
 Toast.makeText(this, "Imagem selecionada.", Toast.*LENGTH\_SHORT*).show()  
 }else {  
 Toast.makeText(this, "Nenhuma imagem selecionada.", Toast.*LENGTH\_SHORT*).show()  
 }  
 **}** override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 *enableEdgeToEdge*()  
 setContentView(binding.*root*)  
 ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.*main*)) **{** v, insets **->** val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())  
 v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom)  
 insets  
 **}** binding.btnGaleria.setOnClickListener **{** abrirGaleria.launch("image/\*") // Mime Type  
  
  
// launch = executar - *todo aquele bloco de notas que fizemos la em cima.*// \* Mime Type = qual tipo de arquivo queremos exibir. (pegue a lista no google)  
// \* Voce pode pesquisar no google  
// \* Aqui no input ele espera receber uma string com o tipo de dados.  
// \*  
// \* Aqui por exemplo vamos trabalhar com imagem e todos os tipos de imagem  
// \* Então vai ficar image/\*\*  
// \* Se quiser trabalhar com um tipo de imagem ficaria  
// \* image.png  
// \*  
// \* Se quiser trabalhar com audio ficaria  
// \* audio/mpeg - audio/vorbis - audio/\* < para todos os tipos de video  
//  
  
  
 **}** }  
}

Agora vamos fazer o upload dessa imagem

# Agora vamos carregar dentro do Storege

Lembrar que primeiro precisamos implementar ele primeiro

Depois podemos criar seu atributo

//Armazenamento  
private val armazenamento by *lazy* **{** FirebaseStorage.getInstance()  
**}**

Agora podemos utilizar ele.

binding.btnUpload.setOnClickListener **{** uploadImagemStore()  
 **}**}  
  
private fun uploadImagemStore() {  
 *TODO*("Not yet implemented")  
}

Agora sim podemos utilizar ele, lembrar que temos um método a cima para pegar a imagem da galeria.

Atenção antes de usar o store, você precisa ir la no Firebase e ativar ele.

Ele vai servi como o banco de dados mesmo do Firebase firestore

Vamos trabalhar com pastas

Ex: fotos (pasta)

+ viagens (pasta)

Cabo\_frio.jpg (foto)

Buzios.jpg (foto)

+ carros (pasta)

Onix2022.png(foto)

Todo código explicado

Agora que você entendeu melhor, nossa estrutura vai ser assim

Fotos

<id\_usuario>

Foto.jpg

Mas não vamos passar o nome da foto assim, porque se a gente passar o nome da foto, toda vez que alguém enviar, ele vai sobrescrever a foto antiga.

Foto vamos deixar com um gerador de identificador único.

# Upload para dentro do Storage

binding.btnGaleria.setOnClickListener **{** abrirGaleria.launch("image/\*") // Mime Type  
  
  
// launch = executar - *todo aquele bloco de notas que fizemos la em cima.*// \* Mime Type = qual tipo de arquivo queremos exibir. (pegue a lista no google)  
// \* Voce pode pesquisar no google  
// \* Aqui no input ele espera receber uma string com o tipo de dados.  
// \*  
// \* Aqui por exemplo vamos trabalhar com imagem e todos os tipos de imagem  
// \* Então vai ficar image/\*\*  
// \* Se quiser trabalhar com um tipo de imagem ficaria  
// \* image.png  
// \*  
// \* Se quiser trabalhar com audio ficaria  
// \* audio/mpeg - audio/vorbis - audio/\* < para todos os tipos de video  
//  
  
  
 **}** binding.btnUpload.setOnClickListener **{** uploadImagemStore()  
 **}** }  
  
 private fun uploadImagemStore() {  
  
 // Pegando o ID do usuario, mas ele precisa logar  
 // E se voce precisar logar ele, voce pode inicar a principal activity de novo...  
 val idUsuario = FirebaseAuth.getInstance().*currentUser*?.*uid* if (idUsuario != null) {  
 // Primeiro devemos verificar se nossa URI = endereço da nossa imagem  
 // Ela não pode ser nula.  
  
 if (uriImagemSelecionada != null) {  
  
 /\*  
 fotos  
 viagens  
 + foto1.png  
 + foto2.png  
 \*/  
  
 /\* Esse modo e para ilustrar melhor  
 armazenamento  
 .getReference("fotos") // Criar a pasta fotos. (Se existir ele usa, se nao ele cria)  
 .child("viagens") // == FILHO - vamos definir um filho para a pasta  
 .child("foto1.pgn") // Agora vamos definir o nome do arquivo.  
 /\*  
 Atenção se voce nao quiser usar a pasta viagens, voce vai ter apenas a pasta fotos e o arquivo da foto.  
 Voce não é obrigado a criar a pasta "viagens"  
 \*/  
  
 \*/  
  
 val gerarNomeParaImagem = UUID.randomUUID().toString() // Agora vamos gerar um nome automatico para a imagem.  
  
 armazenamento.getReference("fotos")  
 .child(idUsuario)  
 .child(gerarNomeParaImagem) // Agora ele vai gerar sempre um nome diferente para cada foto, se tiver nome igual ele fica substituindo...  
  
 .putFile(uriImagemSelecionada!!) //Aqui estamos enviando putFile = arquivo, mas poderiamos enviar de varias formas ex: putBytes  
 */\*\* Atenção a uriImagemSelecionada é o caminho no celular do usuario  
 \* Quando da sucesso a gente retorna com o taks a URL dentro do Storage  
 \*/* .addOnSuccessListener **{** task **->** Toast.makeText(this, "Imagem enviada!", Toast.*LENGTH\_SHORT*).show()  
  
  
 */\*\* Se for sucesso ao salvar a imagem, podemos salvar essa imagem também dentro do firestor  
 \* Para futuramente a gente poder acessar essa imagem usando o picaso  
 \* Dessa forma estamos salvando a imagem dentro do proprio ID do usuario.  
 \*/* task.*metadata*?.*reference*?.*downloadUrl*?.addOnSuccessListener **{** urlFirebase **->** uploadImagemFirestore(urlFirebase!!)  
 Toast.makeText(this, "$urlFirebase", Toast.*LENGTH\_SHORT*).show()  
 **}  
  
  
 }**.addOnFailureListener **{** erro **->** Toast.makeText(this, "Falha ao enviar a imagem.", Toast.*LENGTH\_SHORT*).show()  
 **}** }  
 }  
  
  
}

Upload para dentro do Firebasestore

private fun uploadImagemFirestore(urlFirebase: Uri) {  
 // 1 primeiro ID do usuario logado  
 val idUsuarioLogado = FirebaseAuth.getInstance().*currentUser*?.*uid* if (idUsuarioLogado != null) {  
  
 // Instancia do banco de dados  
 val bancoDeDados = FirebaseFirestore.getInstance()  
  
 // Pegando imagem do usuario  
  
 val dados = *mapOf*(  
 "url\_imagem" *to* urlFirebase  
 )  
  
 bancoDeDados.collection("usuarios")  
 .document(idUsuarioLogado)  
 .update(dados)  
 .addOnSuccessListener **{** Toast.makeText(this, "Imagem carregada!", Toast.*LENGTH\_SHORT*).show()  
 **}**.addOnFailureListener **{** Toast.makeText(this, "ERRO: Não conseguimos guardar sua foto.", Toast.*LENGTH\_SHORT*).show()  
 **}** }  
 }  
}

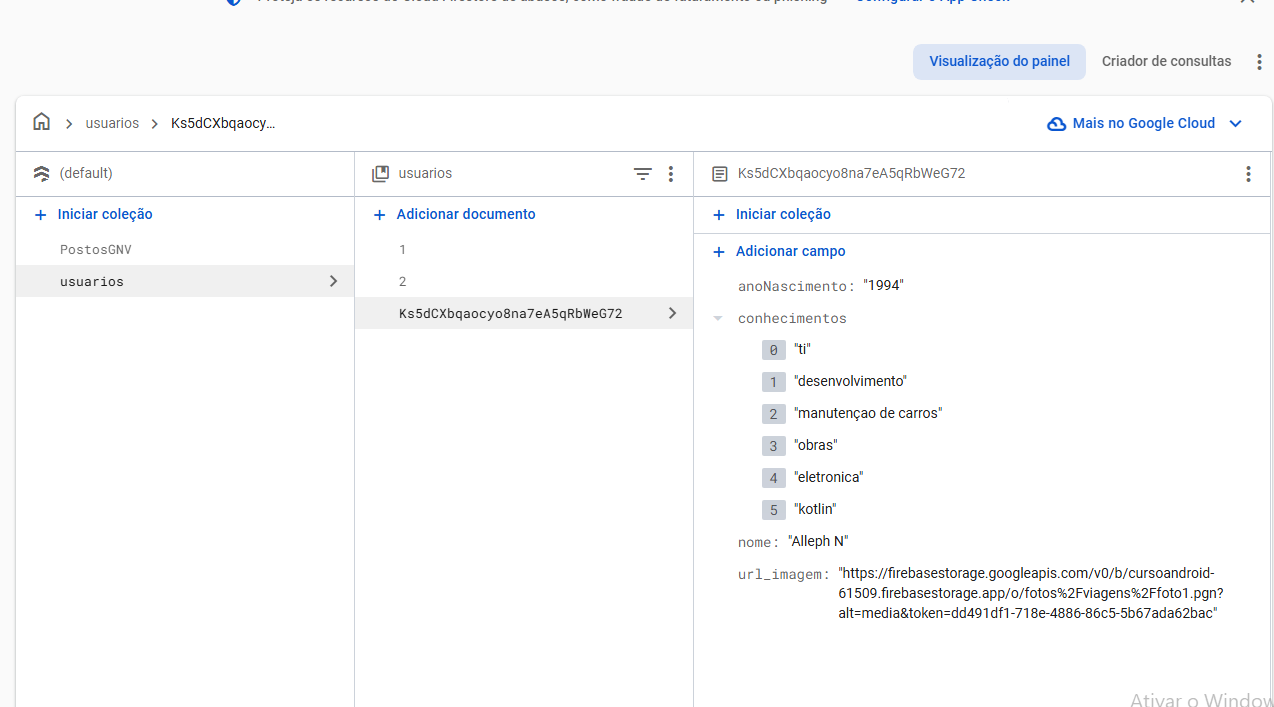
Agora imagina que você quer usar essa foto como foto de perfil.

Podemos usar o picaso para usar essa foto, vamos ter o local onde ela foi armazenada e vamos ter também o link dessa imagem, com o link podemos exibir essa imagem

Segue como ficou dentro do Storage

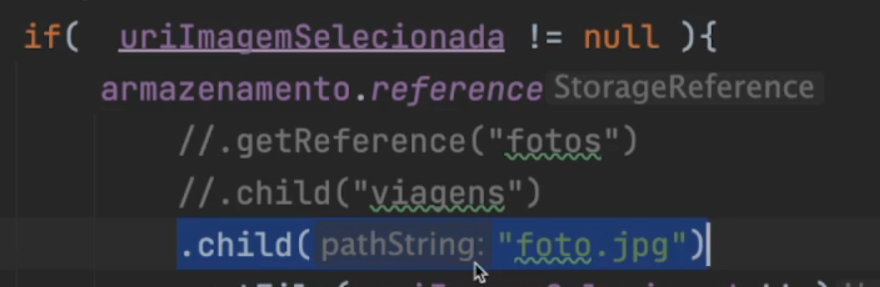


Segue como ficou dentro do firestore



Resumindo, adicionamos os recursos dentro do storage, e gravamos o link de cada um dentro do seu firestore.

# Salvando arquivo fora de pastas



Dessa forma iriamos salvar apenas a foto.jpg e não iria salvar dentro de nenhuma pasta.

Isso ficaria uma bagunça...