Desenvolvimento para Dispositivos Móveis

Aula 09

Recursos do dispositivo (câmera, GPS, sensores)

Prof. Msc. Francisco de Assis de Souza Rodrigues

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

🎯 Objetivos

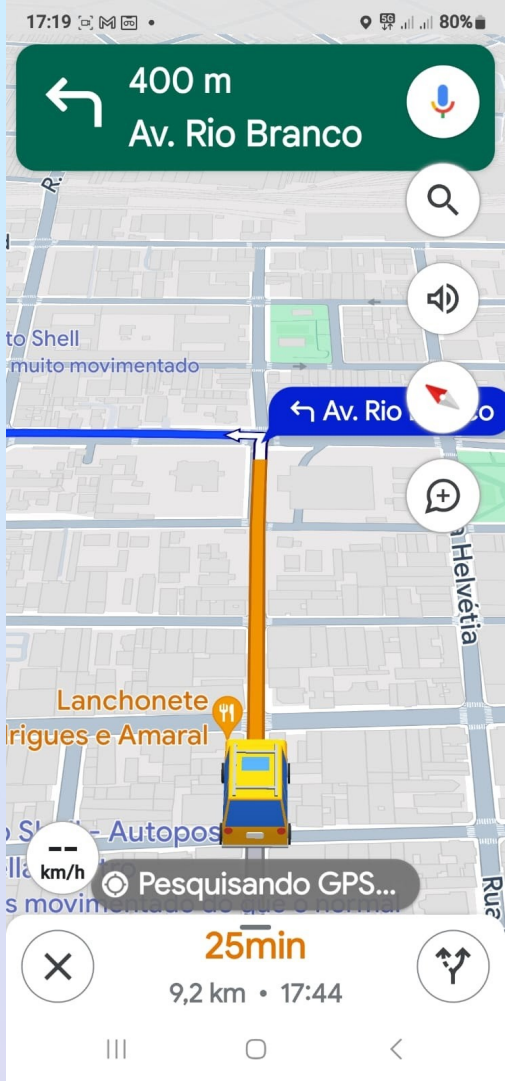
- Compreender como acessar e utilizar a câmera, localização (GPS) e sensores em dispositivos Android.
- Implementar funcionalidades básicas usando os principais recursos de hardware disponíveis no dispositivo.
- Discutir permissões de runtime e boas práticas de uso de recursos sensíveis.

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Introdução aos Recursos do Dispositivo

Antes de mergulhar no código, é fundamental entender que dispositivos Android possuem diversos recursos de hardware integrados — como câmera, GPS, acelerômetro, giroscópio, bússola, entre outros. Esses recursos são acessados por meio de APIs específicas da plataforma Android e frequentemente requerem permissões sensíveis do usuário.

- Qual o papel desses recursos em aplicações modernas?
- Exemplos de apps populares que fazem uso intenso desses recursos: **Instagram** (câmera), **Google Maps** (GPS), apps de saúde e jogos (sensores).



Visão Geral: Câmera, GPS e Sensores

- **Câmera:** utilizada para captura de fotos e vídeos.
- **GPS (Global Positioning System):** fornece a localização do dispositivo com base em satélites, redes Wi-Fi, torres de celular.
- **Sensores:** permitem detectar movimento, orientação e condições ambientais (ex: acelerômetro, giroscópio, luz, proximidade).

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Desafios

A compatibilidade de hardware é um dos grandes desafios no desenvolvimento Android devido à enorme variedade de dispositivos no ecossistema.

📱 1. Variedade de fabricantes e modelos
🧭 2. Sensores ausentes ou com comportamentos distintos
📸 3. APIs de câmera inconsistentes
🌍 4. Diferenças nos provedores de localização
🧩 5. Resoluções e orientações de tela
🔒 6. Políticas e permissões específicas do fabricante
🛠️ 7. Versões do Android diferentes
📦 8. Compatibilidade com o Google Play Services

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Desafios

📱 1. Variedade de fabricantes e modelos

Cada fabricante pode personalizar o Android e usar diferentes sensores, câmeras, GPS e componentes internos. Isso significa que:

- Um recurso pode estar presente em um dispositivo e ausente em outro (ex: sensor de proximidade).
- A qualidade da câmera ou precisão do GPS pode variar bastante.
- APIs que funcionam em um modelo podem se comportar de forma inesperada em outro.

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Desafios

🧭 2. Sensores ausentes ou com comportamentos distintos

Nem todos os dispositivos Android possuem os mesmos sensores. Exemplos:

- Acelerômetro pode não estar presente em alguns tablets ou dispositivos muito baratos.
- Sensor de luz pode estar ausente, ou mal calibrado.
- Alguns sensores podem retornar valores com precisão muito baixa ou ruído excessivo.

Dica: sempre verifique se o sensor existe antes de usá-lo:

```
val sensor = sensorManager.getDefaultSensor(Sensor.TYPE_LIGHT)
if (sensor != null) { /* usa o sensor */ }
```

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Desafios

📸 3. APIs de câmera inconsistentes

- Câmeras podem ter suporte limitado à API antiga (Camera1) ou total à Camera2.
- A CameraX resolve muitos desses problemas, mas dispositivos mais antigos podem ter drivers de câmera inconsistentes, o que causa travamentos ou falhas na captura.

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Desafios

🌍 4. Diferenças nos provedores de localização

Nem todos os dispositivos têm suporte completo ao Fused Location Provider (especialmente sem Google Play Services).

Em alguns dispositivos (como chineses sem Play Services), é necessário usar alternativas nativas.

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Desafios

🧩 5. Resoluções e orientações de tela

- Capturas de imagem podem variar em orientação (portrait/landscape), proporção ou resolução. Códigos que assumem uma proporção fixa tendem a quebrar.



UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Desafios

🔒 6. Políticas e permissões específicas do fabricante

Alguns fabricantes como Xiaomi, Huawei ou Samsung adicionam camadas extras de permissões ou restrições em segundo plano, o que pode impedir o funcionamento normal de sensores ou da localização quando o app está minimizado.



UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Desafios

🛠️ 7. Versões do Android diferentes

- O comportamento de permissões, sensores e APIs varia entre versões do Android.
    - Ex: permissões de localização em background mudaram a partir do Android 10.
    - APIs novas só estão disponíveis em versões recentes.

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Desafios

📦 8. Compatibilidade com o Google Play Services

- Nem todos os dispositivos possuem os serviços do Google, como a FusedLocationProviderClient. Isso é comum em dispositivos com Android AOSP (como Fire OS ou ROMs alternativas).

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

🔧 Estratégias de mitigação:

- Verificação de disponibilidade de hardware em tempo de execução.
- Uso de bibliotecas como CameraX e Play Services para maior compatibilidade.
- Testes em múltiplos emuladores e dispositivos físicos com perfis variados.
- Uso do Google Firebase Test Lab para rodar testes automatizados em diferentes dispositivos reais.
- Implementação de fallback para funcionalidades ausentes.

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

📊 Tabela de Disponibilidade Típica de Recursos em Dispositivos Android

Recurso / Sensor	Dispositivo de Entrada	Intermediário	Topo de Linha
📷 Câmera Frontal e Traseira	✅ Básica	✅ Boa	✅ Alta qualidade (múltiplas lentes, foco)
🎤 Microfone	✅ Simples	✅ Padrão	✅ Qualidade superior / múltiplos microfones
📍 GPS (GNSS)	✅ Presente	✅ Presente	✅ Alta precisão + suporte a múltiplos satélites
📡 Wi-Fi + Rede Celular	✅ 2.4GHz	✅ 2.4 + 5GHz	✅ 5GHz + Wi-Fi 6/7 + 5G
🎛️ Acelerômetro	✅ Presente	✅ Presente	✅ Presente
🎚️ Giroscópio	❌ Nem sempre	✅ Parcial	✅ Presente

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

📊 Tabela de Disponibilidade Típica de Recursos em Dispositivos Android

Recurso / Sensor	Dispositivo de Entrada	Intermediário	Topo de Linha
💡 Sensor de Luz Ambiente	❌ Ausente ou limitado	✅ Padrão	✅ Precisão alta
👆 Sensor de Proximidade	❌ Muitas vezes ausente	✅ Presente	✅ Preciso
🧭 Magnetômetro (bússola)	❌ Muitas vezes ausente	✅ Presente	✅ Preciso
🔋 Sensor de Temperatura/Bateria	✅ Simples	✅ Avançado	✅ Com gerenciamento térmico
🧠 NFC	❌ Raro	✅ Parcial	✅ Comum

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

📊 Tabela de Disponibilidade Típica de Recursos em Dispositivos Android

Recurso / Sensor	Dispositivo de Entrada	Intermediário	Topo de Linha
🕵️ Leitor de impressão digital	❌ Raro	✅ Comum	✅ Avançado (em tela)
🛰️ Suporte a Camera2 API	❌ Limitado	✅ Compatível	✅ Total (requerido pela CameraX)
📦 Google Play Services	✅ Comum, mas depende	✅ Comum	✅ Comum
🔐 Sensor biométrico facial	❌ Ausente	✅ Alguns modelos	✅ Comum

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Permissões no Android Manifest e em Tempo de Execução (Runtime)

No AndroidManifest.xml é necessário declarar permissões como:

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

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Permissões no Android Manifest e em Tempo de Execução (Runtime)

A partir do Android 6.0 (API 23), é obrigatório solicitar permissões perigosas em tempo de execução:

- Permissões perigosas exigem consentimento explícito do usuário.

- Exemplo de solicitação:

```
ActivityCompat.requestPermissions(this, arrayOf(Manifest.permission.CAMERA), REQUEST_CODE)
```

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Permissões no Android Manifest e em Tempo de Execução (Runtime)

Câmera

a) Abertura da câmera via Intent (MediaStore): Abordagem simples e rápida usando Intent implícita para acionar o app de câmera nativo:

- Receber resultado via onActivityResult e acessar Bitmap da imagem.

```
val cameraIntent = Intent(MediaStore.ACTION_IMAGE_CAPTURE)
startActivityForResult(cameraIntent, REQUEST_IMAGE_CAPTURE)
```

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Permissões no Android Manifest e em Tempo de Execução (Runtime)

Câmera

b) Armazenamento da imagem capturada:

- Utilizar MediaStore ou salvar manualmente a imagem em um arquivo temporário.

- **Exemplo:** criar um FileProvider e usar o caminho como URI para a imagem.

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Permissões no Android Manifest e em Tempo de Execução (Runtime)

Câmera

c) Uso da CameraX (avançado, opcional):

- Introdução ao CameraX, biblioteca moderna e fácil de usar para capturar fotos, vídeos e fazer previews com controle direto no app.

- Vantagens: mais controle, melhor compatibilidade entre dispositivos.

- Mencionar classes principais: Preview, ImageCapture, CameraProvider.

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Permissões no Android Manifest e em Tempo de Execução (Runtime)

Localização (GPS)

a) API de localização do Google Play Services:

- FusedLocationProviderClient: fornece acesso à localização com menor uso de bateria.

- Obter localização atual com:

```
fusedLocationClient.lastLocation
    .addOnSuccessListener { location ->
        // Usar latitude e longitude
    }
```

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Permissões no Android Manifest e em Tempo de Execução (Runtime)

Localização (GPS)

b) Requisição de permissões:

- Permissões necessárias:
    - android.permission.ACCESS_FINE_LOCATION (precisa de runtime permission)
    - android.permission.ACCESS_COARSE_LOCATION
- Gerenciar permissões com atenção ao Android 10+ (background location) e Android 11+ (precisa pedir permissões de localização separadas).

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Permissões no Android Manifest e em Tempo de Execução (Runtime)

Localização (GPS)

c) Obtenção da última localização conhecida:

- FusedLocationProviderClient pode retornar null se o GPS estiver desativado ou se a localização não estiver disponível.

- Alternativas: configurar LocationRequest com intervalos e escutar atualizações em tempo real.

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Permissões no Android Manifest e em Tempo de Execução (Runtime)

Sensores

a) SensorManager e tipos de sensores:

- SensorManager é a classe central para acessar sensores.
- Tipos comuns:
  - Acelerômetro: movimento em X, Y, Z
  - Giroscópio: rotação
  - Sensor de luz, proximidade, pressão
- Obter lista de sensores disponíveis:

```
val sensorManager = getSystemService(Context.SENSOR_SERVICE) as SensorManager
val sensors = sensorManager.getSensorList(Sensor.TYPE_ALL)
```

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Permissões no Android Manifest e em Tempo de Execução (Runtime)

Sensores

b) Registro e tratamento de eventos:

- Criar um SensorEventListener para responder a mudanças em sensores:

```
sensorManager.registerListener(sensorEventListener, accelerometer,
SensorManager.SENSOR_DELAY_NORMAL)
```

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Permissões no Android Manifest e em Tempo de Execução (Runtime)

Boas Práticas

a) Verificação de permissões:

- Verificar se a permissão foi concedida antes de executar qualquer funcionalidade sensível.

- Fornecer feedback útil se o usuário negar uma permissão.

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Permissões no Android Manifest e em Tempo de Execução (Runtime)

Boas Práticas

b) Uso consciente de bateria e privacidade:

- Minimizar chamadas desnecessárias ao GPS ou sensores.
- Não manter sensores ativos em segundo plano se não for essencial.
- Armazenar dados localmente com criptografia se forem sensíveis.

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Permissões no Android Manifest e em Tempo de Execução (Runtime)

Boas Práticas

c) Gerenciamento de ciclo de vida ao usar sensores:

- Ligar e desligar sensores de acordo com o ciclo de vida da activity ou fragment.

- CameraX e Location updates também devem seguir esse padrão para evitar vazamento de recursos ou apps rodando em segundo plano sem necessidade.

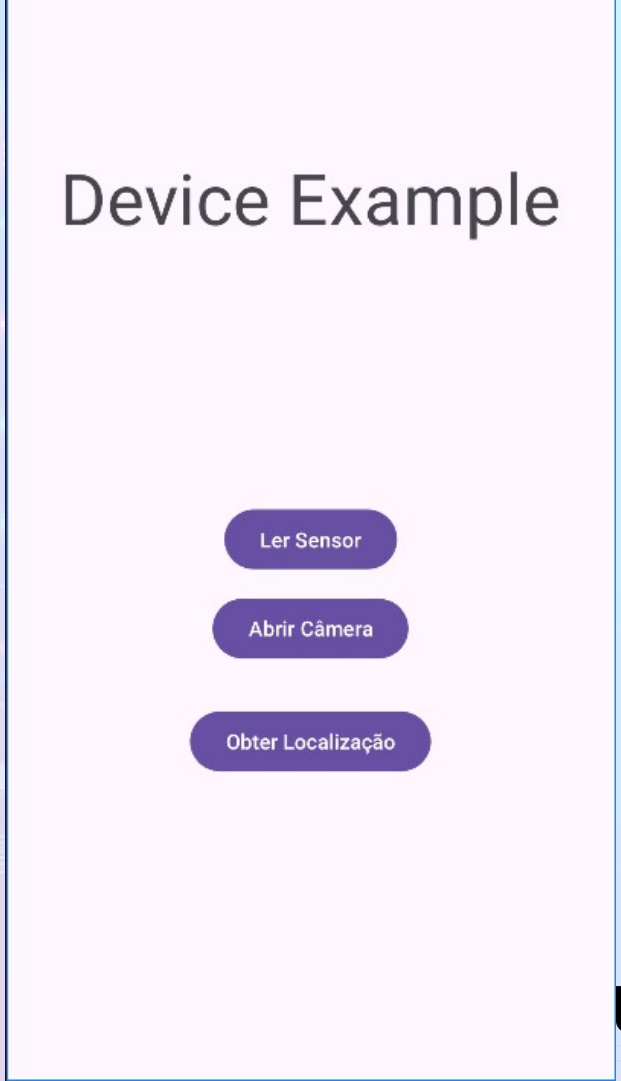
UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Prática

Projeto Android no Android Studio:

- Nome: “**DeviceExample**”

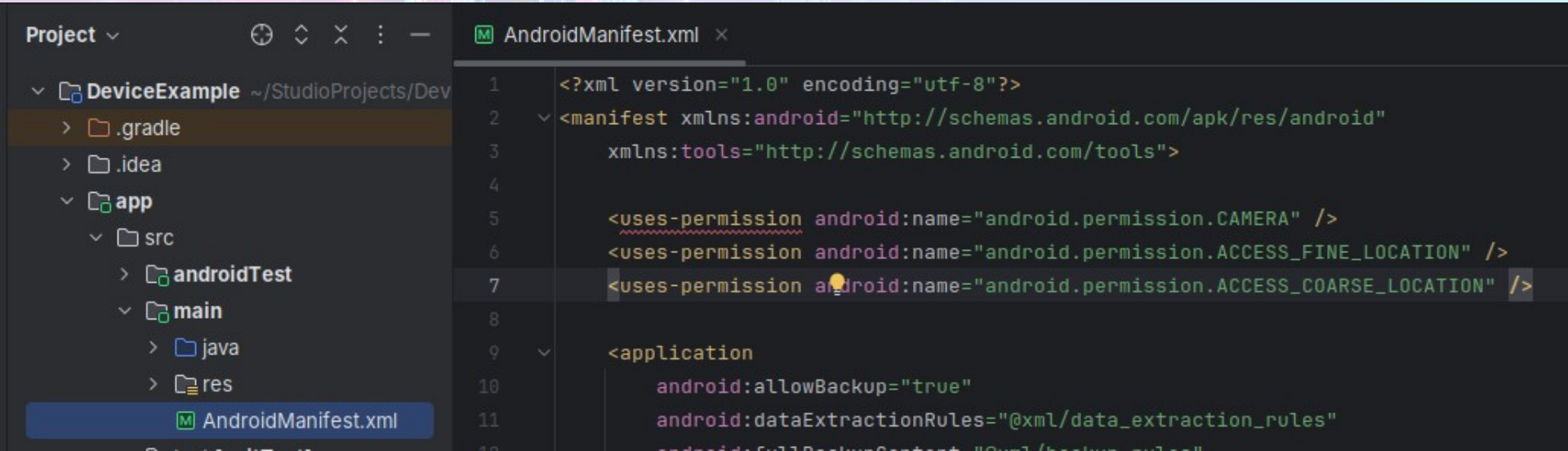
- Criação de 1 tela com três botões:
  a. “Abrir Câmera”,
  b. “Obter Localização”,
  c. “Ler Sensor”



UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Prática

Configurações do Android Manifest



Prática

```
24
25  class MainActivity : AppCompatActivity() {
26
27      private lateinit var fusedLocationClient: FusedLocationProviderClient
28      private lateinit var sensorManager: SensorManager
29      private var accelerometer: Sensor? = null
30      private val sensorListener = object : SensorEventListener {
31          override fun onSensorChanged(event: SensorEvent?) {
32              event?.values?.let {
33                  Log.d( tag: "Sensor",  msg: "X: ${it[0]}, Y: ${it[1]}, Z: ${it[2]}")
34              }
35          }
36
37          override fun onAccuracyChanged(sensor: Sensor?, accuracy: Int) {}
38      }
```

Prática

```
override fun onCreate(savedInstanceState: Bundle?) {
    ...

    val btnCamera = findViewById<Button>(R.id.btnCamera)
    val btnGPS = findViewById<Button>(R.id.btnGPS)
    val btnSensor = findViewById<Button>(R.id.btnSensor)

    fusedLocationClient = LocationServices.getFusedLocationProviderClient( activity: this)
    sensorManager = getSystemService(SENSOR_SERVICE) as SensorManager
    accelerometer = sensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER)

    btnCamera.setOnClickListener {
        if (ContextCompat.checkSelfPermission( context: this, Manifest.permission.CAMERA)
            != PackageManager.PERMISSION_GRANTED
        ) {
            ActivityCompat.requestPermissions(
                activity: this, arrayOf(Manifest.permission.CAMERA), REQUEST_IMAGE_CAPTURE
            )
        } else {
            openCam()
        }
    }
```

Prática

```
override fun onCreate(savedInstanceState: Bundle?) {
    ...

    btnGPS.setOnClickListener {
        if (ContextCompat.checkSelfPermission(
                context: this,
                Manifest.permission.ACCESS_FINE_LOCATION
            ) != PackageManager.PERMISSION_GRANTED
        ) {
            ActivityCompat.requestPermissions(
                activity: this,
                arrayOf(Manifest.permission.ACCESS_FINE_LOCATION),
                LOCATION_PERMISSION_REQUEST
            )
        } else {
            getLastLocation()
        }
    }
```

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Prática

```
override fun onCreate(savedInstanceState: Bundle?) {
    ...

    btnSensor.setOnClickListener {
        accelerometer?.let {
            sensorManager.registerListener(
                sensorListener,
                it,
                SensorManager.SENSOR_DELAY_NORMAL
            )
            Toast.makeText( context: this,  text: "Sensor ligado", Toast.LENGTH_SHORT).show()
        }
    }
}
```

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Prática

Fora do onCreate()

```
private fun getLastLocation() {
    fusedLocationClient.lastLocation.addOnSuccessListener { location: Location? ->
        location?.let {
            Log.d( tag: "GPS",  msg: "Lat: ${it.latitude}, Long: ${it.longitude}")
            Toast.makeText(
                context: this,
                text: "Localização: ${it.latitude}, ${it.longitude}",
                Toast.LENGTH_LONG
            ).show()
        }
    }
}
```

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Prática

Fora do onCreate

```
override fun onRequestPermissionsResult(
    requestCode: Int,
    permissions: Array<String>,
    grantResults: IntArray
) {
    super.onRequestPermissionsResult(requestCode, permissions, grantResults)
    if (requestCode == LOCATION_PERMISSION_REQUEST &&
        grantResults.isNotEmpty() &&
        grantResults[0] == PackageManager.PERMISSION_GRANTED) {
        getLastLocation()
    } else {
        Toast.makeText( context: this,  text: "Permissão de localização negada",
            Toast.LENGTH_SHORT).show()
    }
}
```

Prática

Fora do onCreate()

```
override fun onPause() {
    super.onPause()
    sensorManager.unregisterListener(sensorListener)
}

companion object {
    private const val REQUEST_IMAGE_CAPTURE = 1
    private const val LOCATION_PERMISSION_REQUEST = 2
}
```

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Prática

Fora do onCreate()

```
private fun openCam() {
    val intent = Intent(MediaStore.ACTION_IMAGE_CAPTURE)
    if (intent.resolveActivity(packageManager) != null) {
        startActivityForResult(intent, REQUEST_IMAGE_CAPTURE)
    } else {
        Toast.makeText( context: this,  text: "Nenhum app de câmera disponível",
            Toast.LENGTH_SHORT).show()
    }
}
```

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Prática

Fora do onCreate()

```
override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {
    super.onActivityResult(requestCode, resultCode, data)
    if (requestCode == REQUEST_IMAGE_CAPTURE && resultCode == RESULT_OK) {
        Toast.makeText( context: this,  text: "Imagem capturada!", Toast.LENGTH_SHORT).show()
    }
}
```

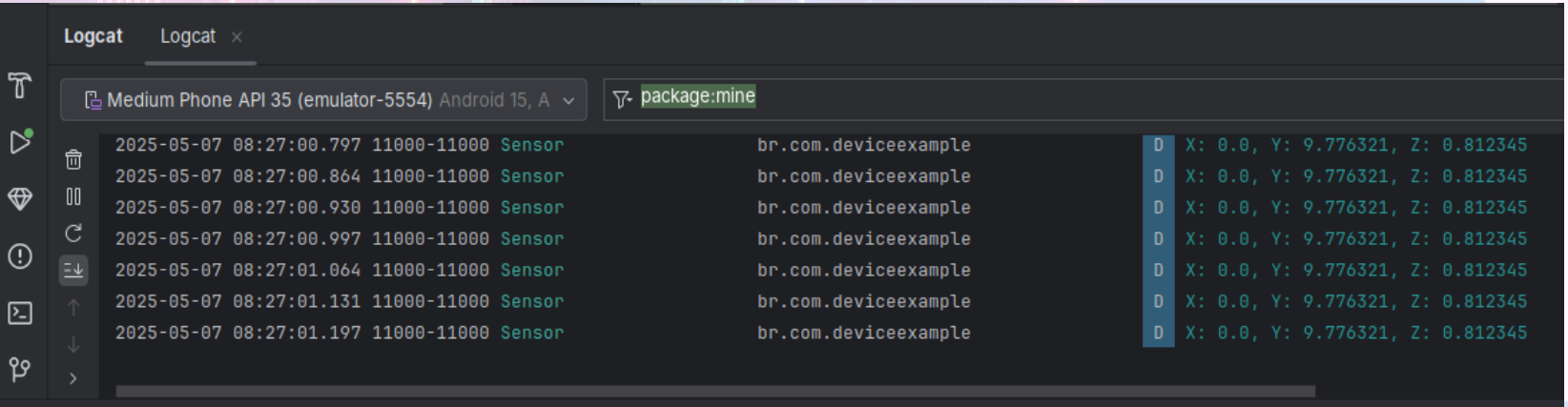
UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

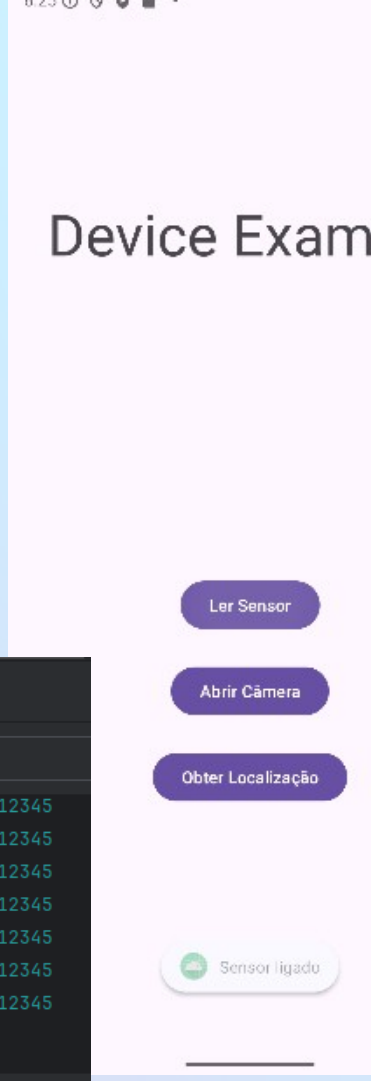
Prática

Execute seu App

Ao clicar no botão Ligar Sensor, obtemos o feedback visual do Toast.

E podemos observar os valores de no Logcat:

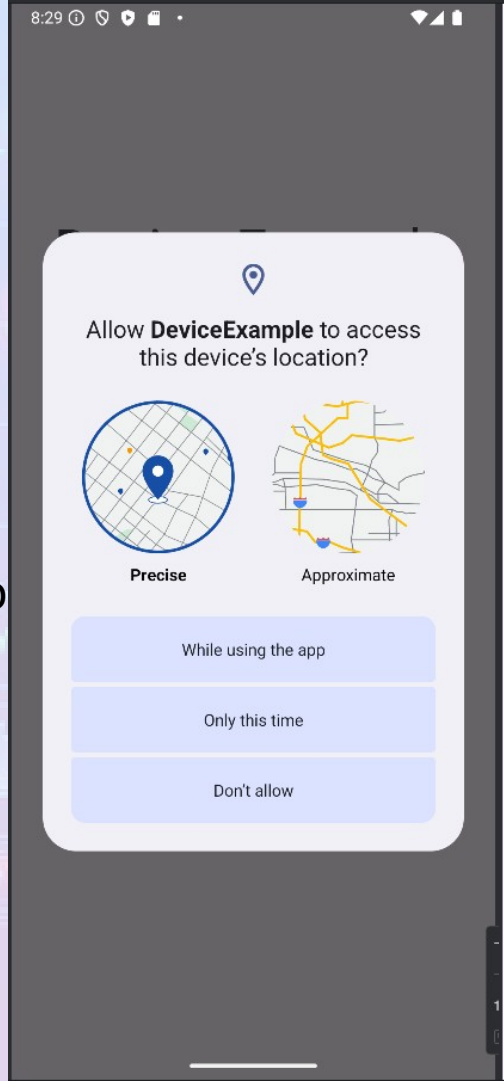


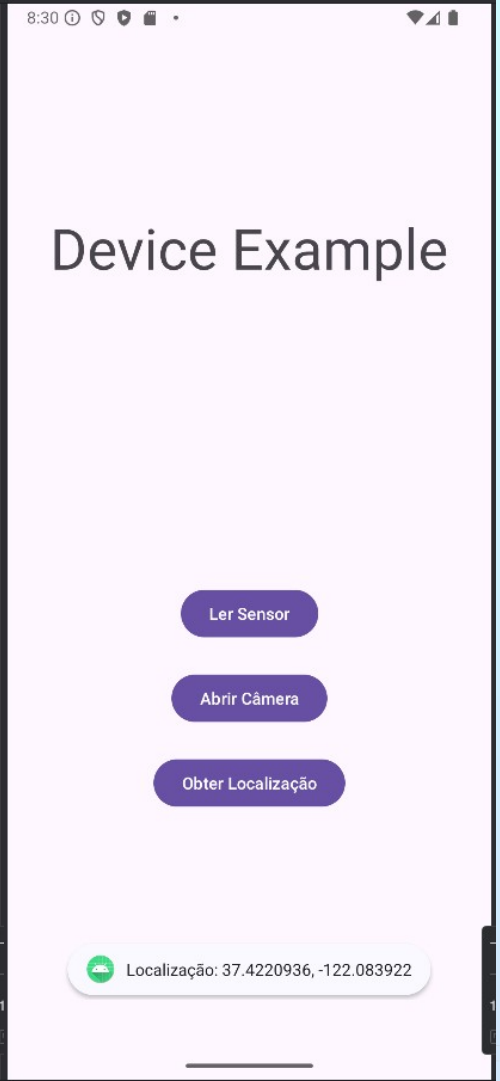


Prática

Ao clicar no botão Obter Localização o Android exibe solicitação de permissões de acesso.

Ao conceder a permissão, a localização é obtida em **Latitude** e **Longitude**.

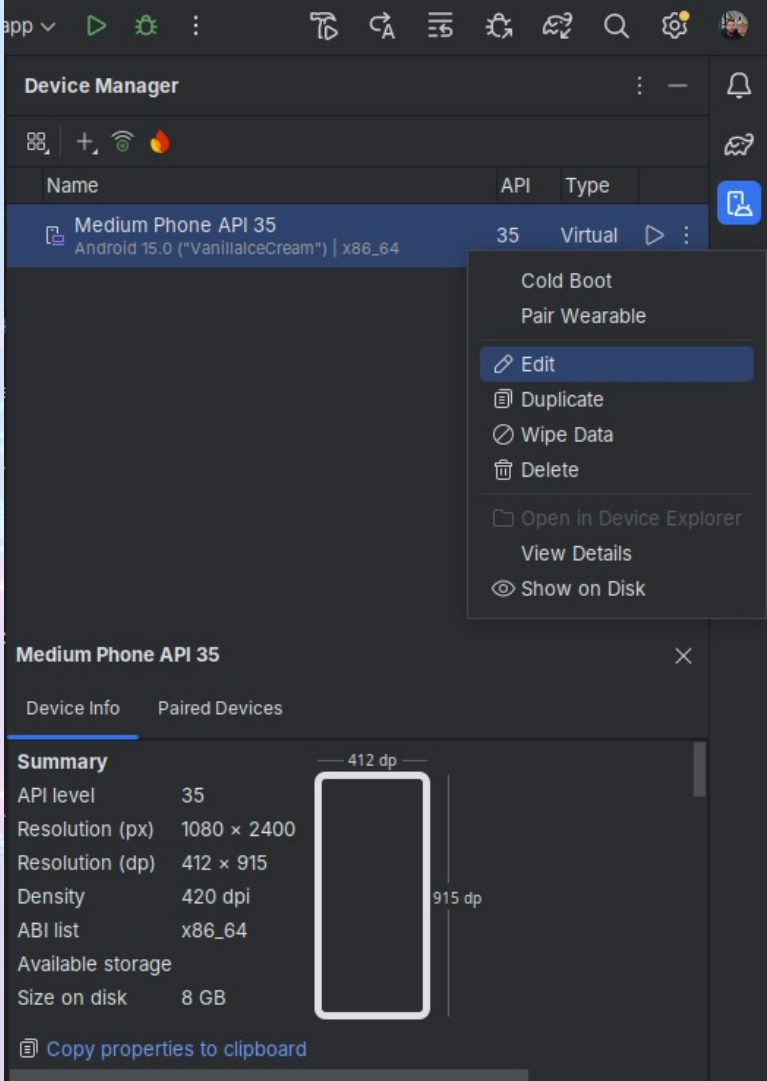




Prática

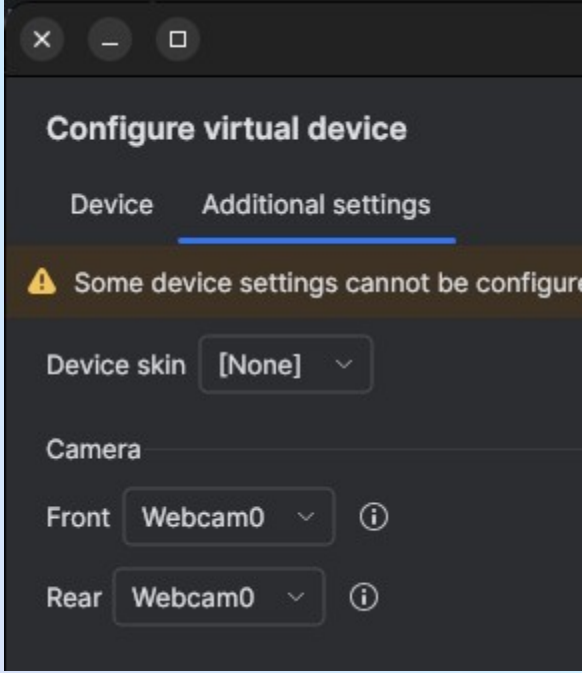
Para usar a câmera do seu computador no emulador do android você precisa configurar.

Abra o Device Manager e clique em editar.



Prática

Ao abrir a tela de Configurações do dispositivo virtual, clique na aba de Configurações Adicionais, na opção Câmera, selecione Webcam0 em Front e/ou Reat.



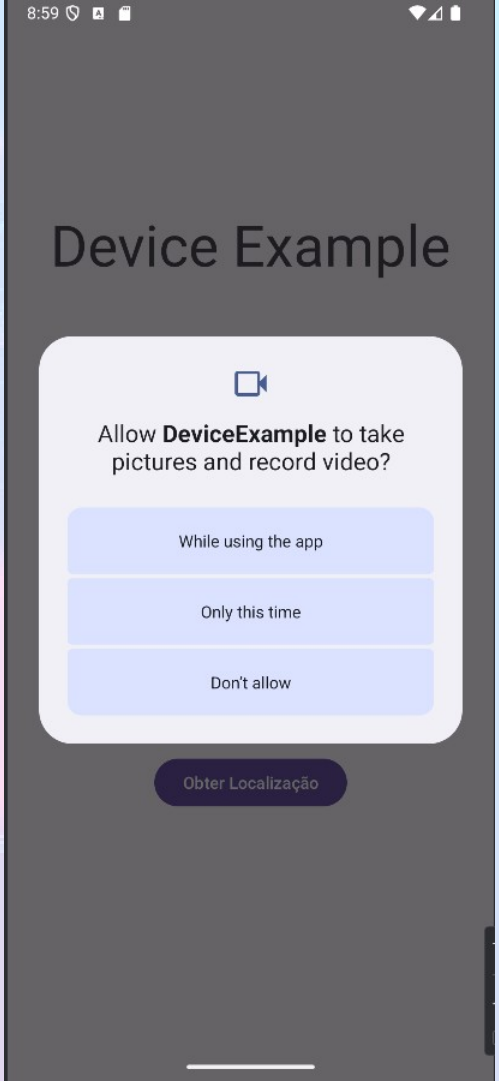
Agora seu dispositivo consegue usar a câmera para simulação de captura de foto.

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Prática

Ao clicar no botão Abrir Câmara, vai solicitar em tempo de execução permissão para acessar a câmera.

Se seu emulador tiver permissão você vê a câmera funcionar se não, uma mensagem aparecerá que não está disponível neste dispositivo, tente experimentar o app no device físico.



UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Github - Repositório do Exemplo

https://github.com/Mobile-Device-Development/DeviceExample



UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Bibliografia

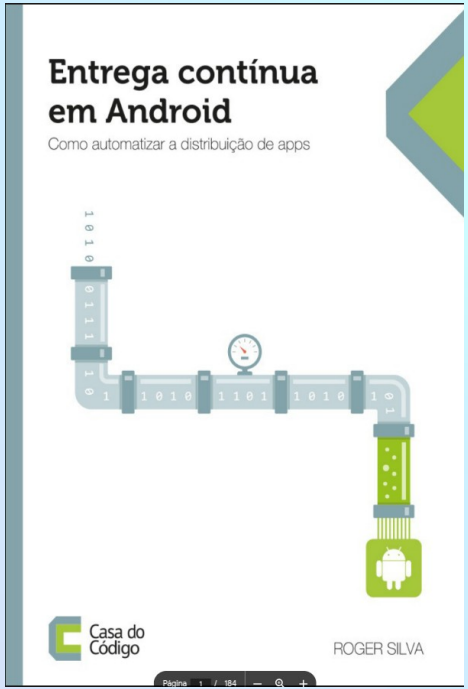
RESENDE, Kassiano. Kotlin com android: crie aplicativos de maneira fácil e divertida. São Paulo, SP: Casa do Código, 2018. E-book. Disponível em: https://plataforma.bvirtual.com.br. Acesso em: 26 fev. 2025.

https://plataforma.bvirtual.com.br/Acervo/Publicacao/213016

SILVA, Roger. Entrega contínua em android: como automatizar a distribuição de apps. São Paulo, SP: Casa do Código, 2016. E-book. Disponível em: https://plataforma.bvirtual.com.br. Acesso em: 26 fev. 2025.

https://plataforma.bvirtual.com.br/Acervo/Publicacao/212670

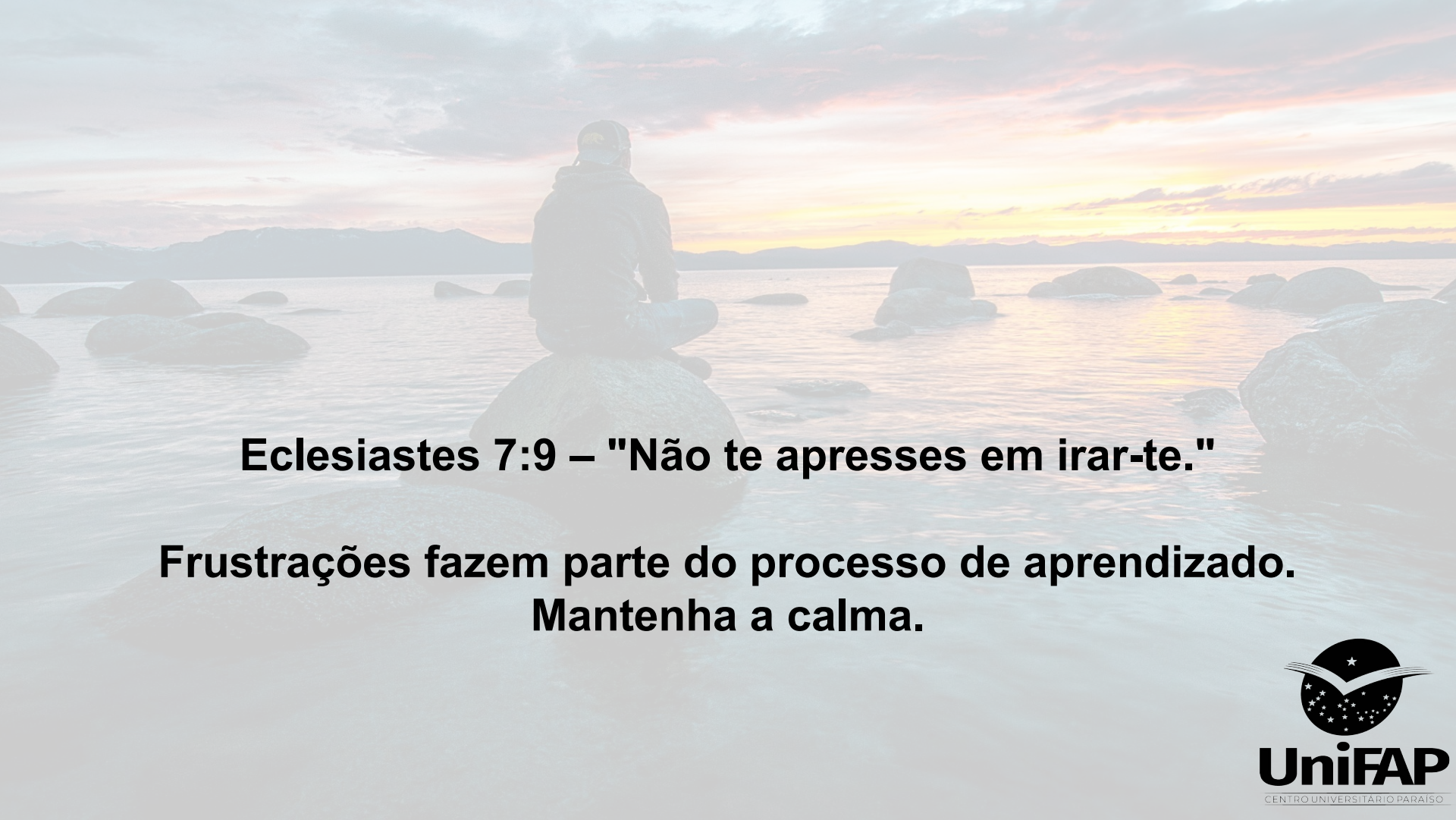




Perguntas?!



UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO



Eclesiastes 7:9 – "Não te apresses em irar-te."

Frustrações fazem parte do processo de aprendizado.
Mantenha a calma.

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO



Obrigado!

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO