



SMD0122 - PROGRAMAÇÃO PARA DISPOSITIVOS MÓVEIS - T01  
Prof. Windson Viana de Carvalho

# Autenticação Biométrica com Android

## Objetivo do Roteiro

Este roteiro tem como objetivo:

1. Apresentar o uso da API BiometricPrompt no Android.
2. Orientar os alunos a criarem três métodos diferentes de autenticação: digital, facial e senha/padrão.
3. Guiar o aluno a organizar o código a partir de um projeto-base inicial.

### 1. Introdução

#### O que é a Autenticação Biométrica no Android?

O Android oferece suporte à autenticação biométrica desde as versões mais antigas do sistema, com suporte inicial limitado à impressão digital. Com o tempo, a plataforma evoluiu para permitir o uso de múltiplos métodos de autenticação, como reconhecimento facial, íris e também credenciais do dispositivo (como senha, padrão ou PIN).

## 2

Para centralizar e padronizar o uso dessas autenticações, o Android introduziu a API `BiometricPrompt`, que permite solicitar autenticação do usuário de forma segura e unificada, oferecendo uma experiência nativa consistente.

O componente `BiometricPrompt` possui alguns elementos centrais que você deve entender:

**`BiometricManager`:** permite verificar se o dispositivo oferece suporte ao tipo de autenticação desejado (ex: digital, facial ou senha/padrão).

**`BiometricPrompt`:** é a classe que exibe a interface de autenticação e retorna o resultado (sucesso ou falha).

**`PromptInfo`:** define como será o diálogo de autenticação, incluindo o título, subtítulo e o tipo de autenticação permitida (`BIOMETRIC_STRONG`, `WEAK`, `DEVICE_CREDENTIAL`).

Então vamos partir para a prática aplicar e entender melhor alguns desses conceitos. O link do código inicial está aqui: [GitHub](#)

## 2. Atividade

Ao abrir o projeto fornecido, você verá que a autenticação por digital já é solicitada automaticamente. Sua tarefa será:

1. Remover a autenticação automática.
2. Criar adicionar **três botões** na interface da tela inicial.
3. Implementar a autenticação correta para cada botão, com base no tipo:

Tipo	Autenticador
Digital	BIOMETRIC_STRONG
Facial	BIOMETRIC_WEAK

Senha/Padrão	DEVICE_CREDENTIAL
--------------	-------------------

### Guia passo a passo:

#### *Etapa 1 – Abrir o app diretamente e criar os botões na tela*

Da forma como o código foi entregue a tela inicial só é exibida quando você consegue passar do processo de autenticação da digital com sucesso, para mudar isso e seguir na atividade você pode mover o setContentView para fora da função de “autenticação bem sucedida”, no início do seu onCreate, atualmente ele se encontra assim:

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

    executor = ContextCompat.getMainExecutor(this)
    biometricPrompt = BiometricPrompt(this, executor, object : BiometricPrompt.AuthenticationCallback() {
        override fun onAuthenticationSucceeded(result: BiometricPrompt.AuthenticationResult) {
            super.onAuthenticationSucceeded(result)
            runOnUiThread {
                Toast.makeText(applicationContext, "Autenticado com sucesso!", Toast.LENGTH_SHORT).show()
                setContentView(R.layout.activity_main)
            }
        }
    })
}
```

Edite o arquivo activity\_main.xml e adicione os botões necessários para a atividade:

```
<Button
    android:id="@+id/btn_digital"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Autenticar com Digital" />
```

Você também pode criar os botões usando somente a tela de design, mas lembre de fazer isso para cada um dos botões e mantenha text e IDs distintos.

#### *Etapa 2 – Criar métodos para cada tipo de autenticação*

No arquivo MainActivity.kt, crie três métodos separados:

```
private fun autenticarDigital() { ... }
private fun autenticarFacial() { ... }
private fun autenticarSenha() { ... }
```

Cada método deve:

- Usar o BiometricManager para verificar suporte.
- Criar um PromptInfo com o setAllowedAuthenticators(...).
- Chamar biometricPrompt.authenticate(promptInfo).

### *Etapa 3 – Associar os botões aos métodos*

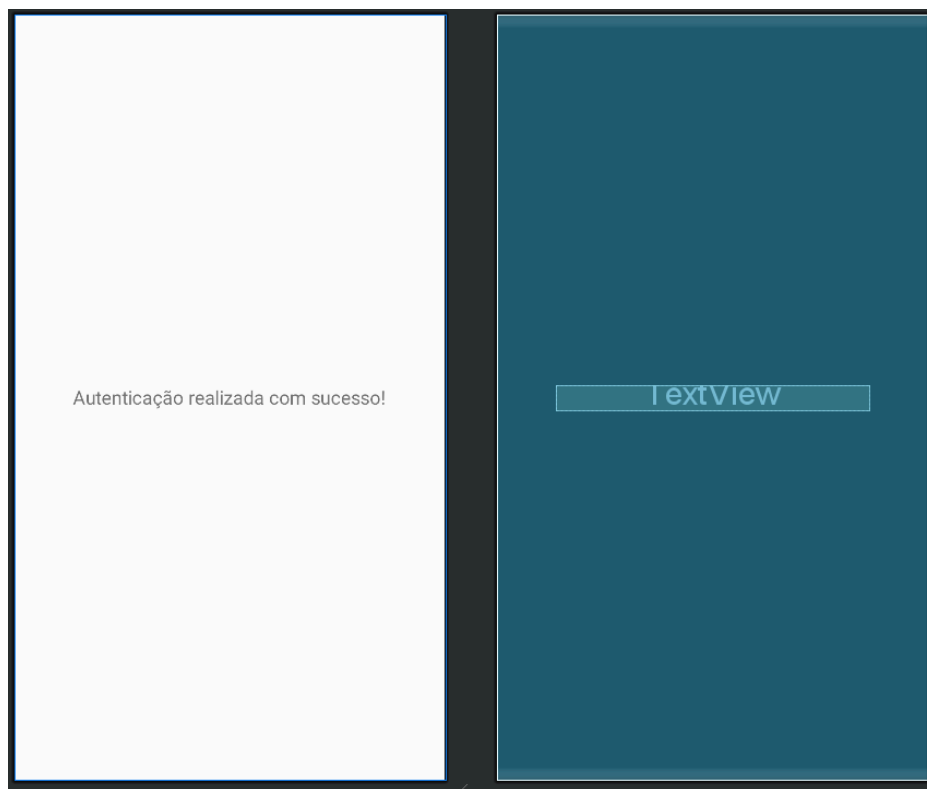
No onCreate, associe cada botão a sua função:

```
// Botão para autenticação por impressão digital
findViewById<Button>(R.id.btn_digital).setOnClickListener {
    autenticarDigital()
}

// Botão para autenticação facial (fraca)
findViewById<Button>(R.id.btn_facial).setOnClickListener {
    autenticarFacial()
}

// Botão para autenticação por senha ou padrão do dispositivo
findViewById<Button>(R.id.btn_senha).setOnClickListener {
    autenticarSenha()
}
}
```

No final, ao passar de qualquer uma das autenticações, você irá redirecionar o usuário para uma nova tela, para isso você pode usar a SecondActivity que já está no projeto:



Ao acessar essa tela a tarefa está concluída.

### Observações Importantes

- DEVICE\_CREDENTIAL exige que o dispositivo tenha senha, padrão ou PIN configurado.
- O botão de cancelar não deve ser usado com DEVICE\_CREDENTIAL, pois esse modo usa a tela de bloqueio do sistema.