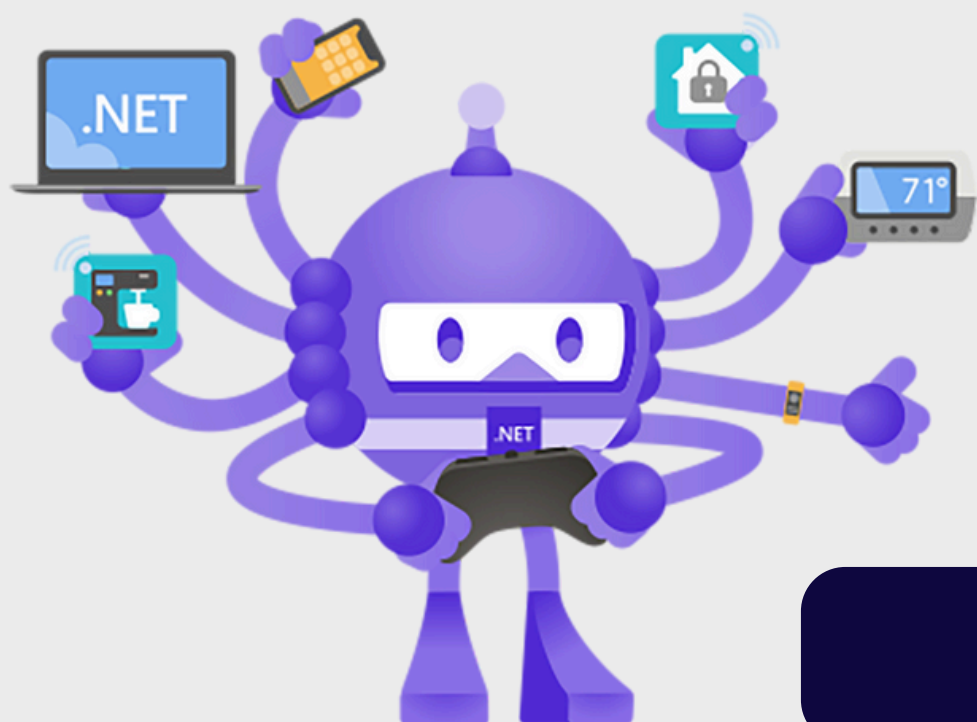


APP PARIDADE - Comparativo



.NET MAUI

X

ANDROID NATIVO



Equipe:

Fernanda Laudaes Silva

Julia Ragi Beltrão

Lucas Silva Ciacci

Sara Veríssimo Souza

Professor :

Pedro Palmuti

Curso:

Ciência da Computação - Mobile Technologies

OBJETIVO DO ESTUDO



Analisar

Analisar diferenças entre .NET MAUI e Android Nativo;



Avaliar

Avaliar produtividade, desempenho e manutenção;

- App desenvolvido com as mesmas features em ambas as plataformas:

- Lista filtrável (JSON local)
- Requisição HTTP (API IBGE)
- Geolocalização
- Tema Dark/Light
- Teste unitário simples

AMBIENTE E PROCEDIMENTOS

.NET MAUI (C#)

- IDE: Visual Studio 2022 (v17.8)
- SO: Windows 11 Pro 64 bits
- SDK: Android SDK 34 (Google APIs)
- Emulador: Pixel 5 (API 34)
- Build: Release (Hot Reload desativado)
- Linguagem: C#



Android Nativo (Kotlin)

- IDE: Android Studio Narwhal 3
- SO: Windows 10
- SDK: Android SDK 34 (minSdk 24, targetSdk 34)
- Emulador: Pixel 7 (API 34)
- Build: Release (Instant Run desativado)
- Linguagem: Kotlin 1.9.20

- **Lista filtrável com vinte itens locais:**

- **.NET MAUI :**

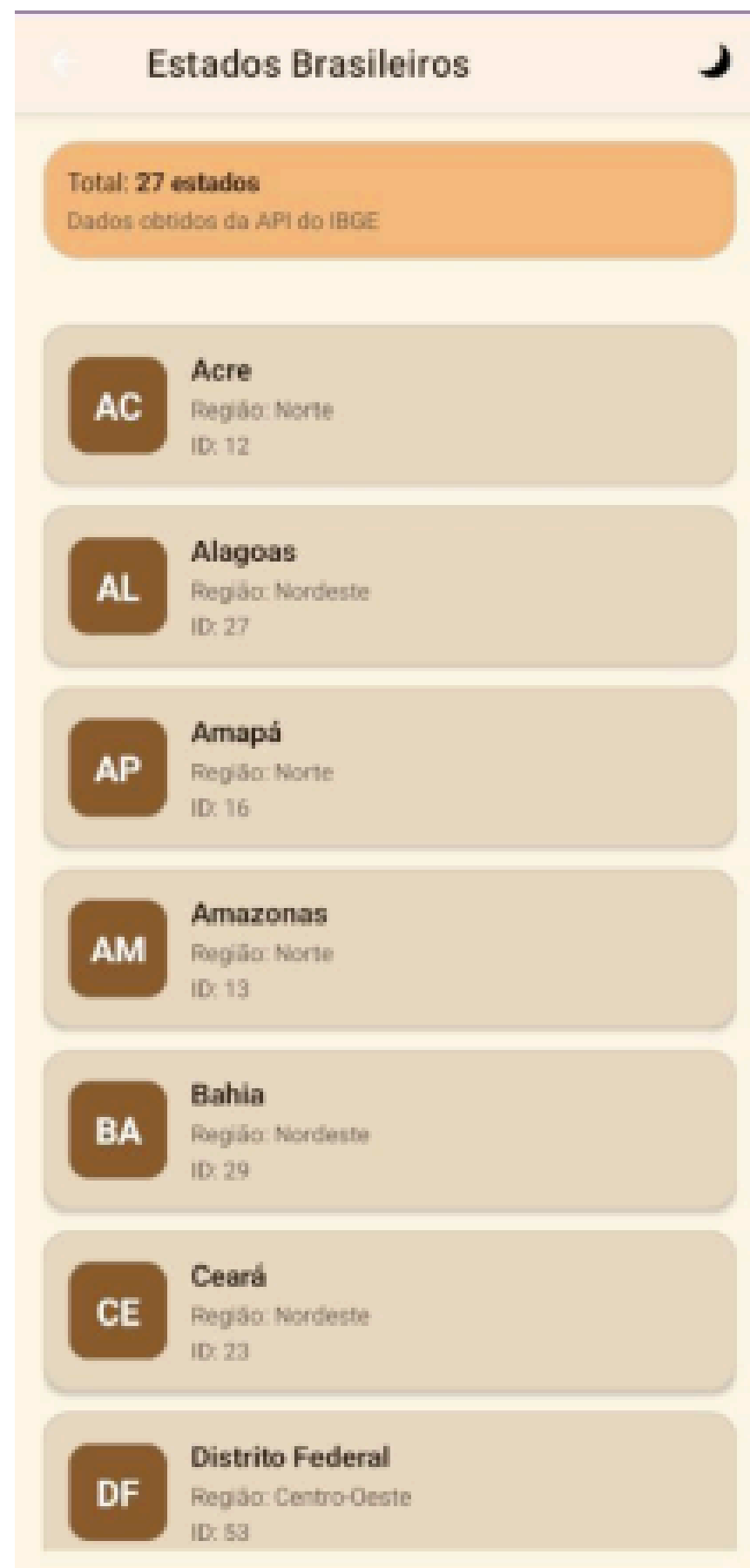


- **ANDROID NATIVO :**



- **Requisição HTTP para API pública (IBGE)**

- **NET MAUI :**



- **ANDROID NATIVO :**



- **Geolocalização:**

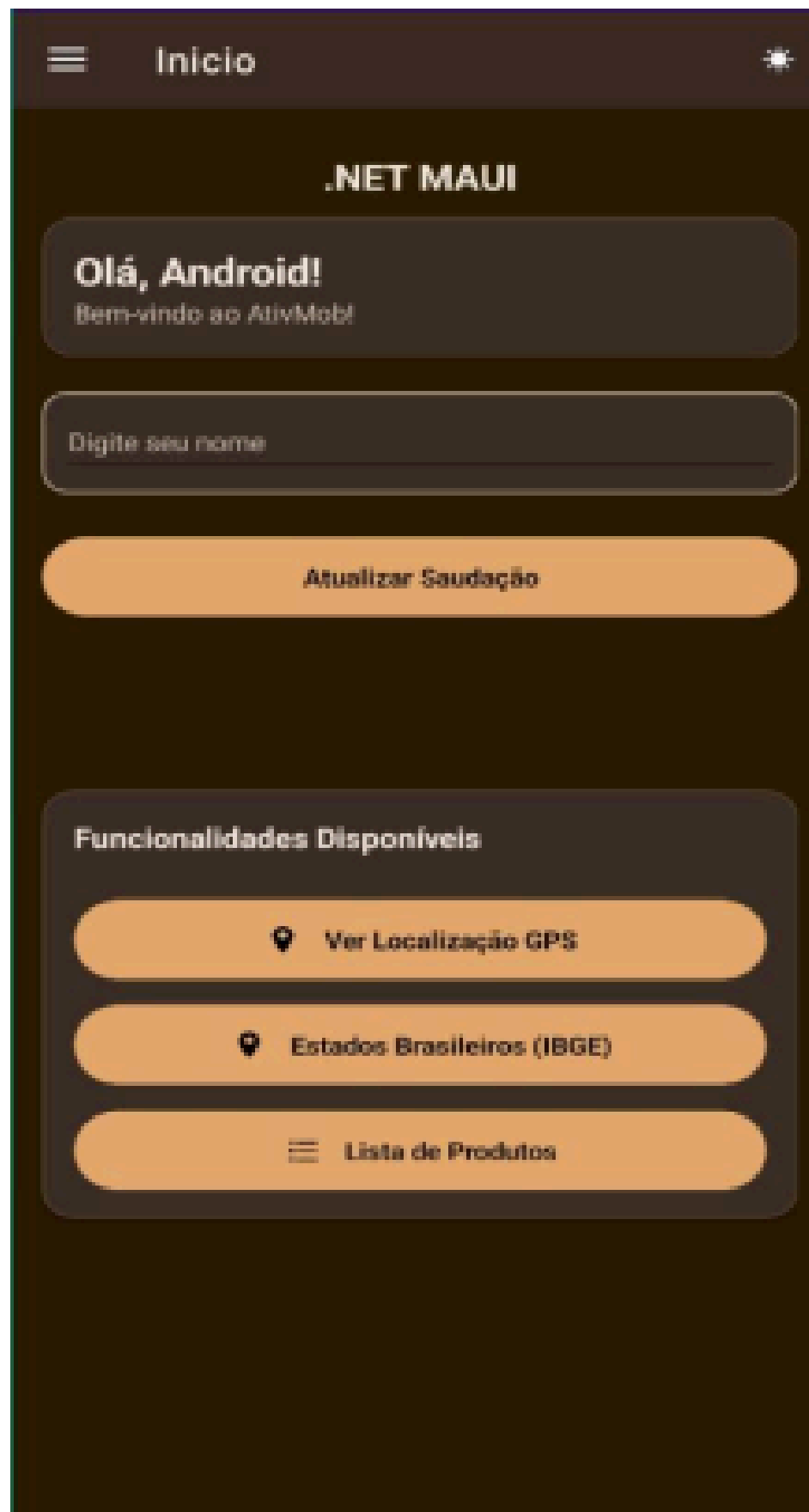
- **.NET MAUI :**



- **ANDROID NATIVO :**



- Tema Dark/Light :
 - .NET MAUI :

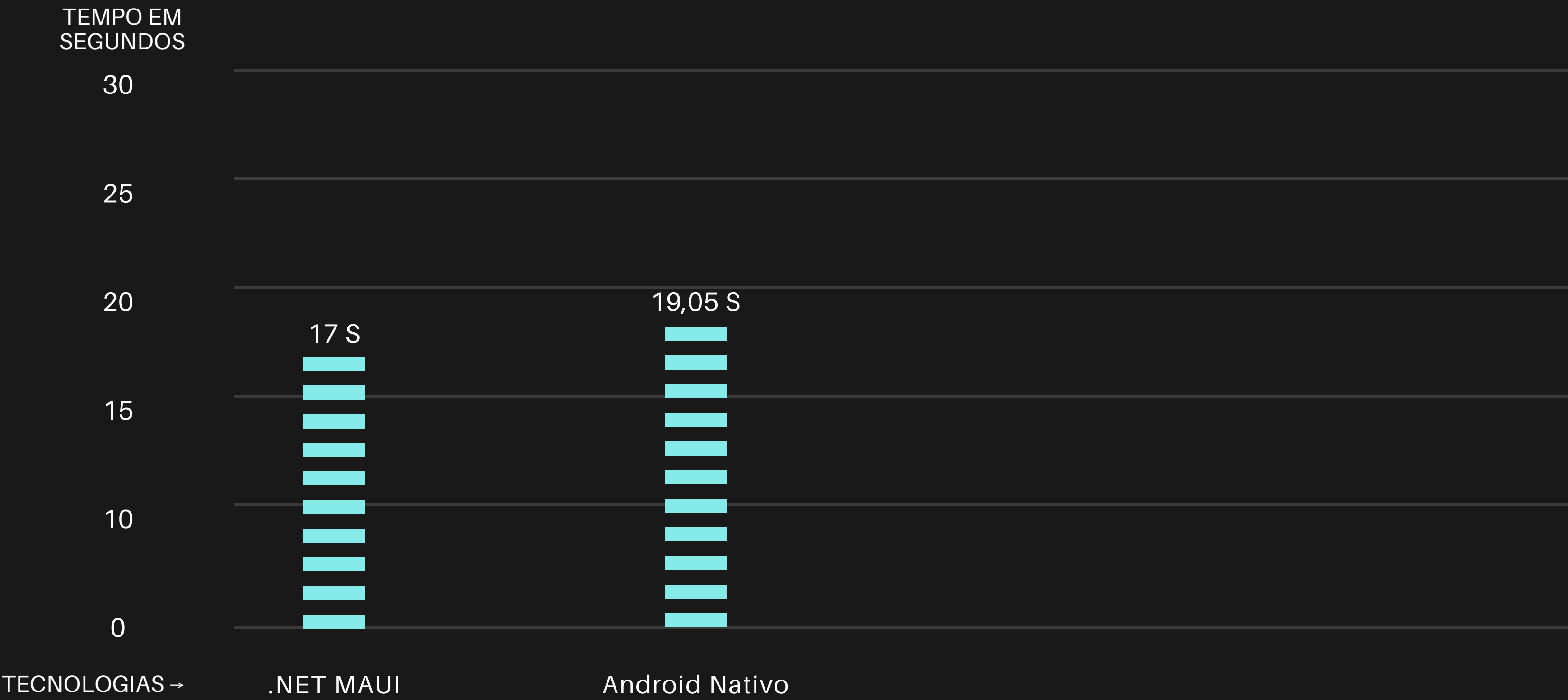


ANDROID NATIVO :



TEMPO DE COLD START

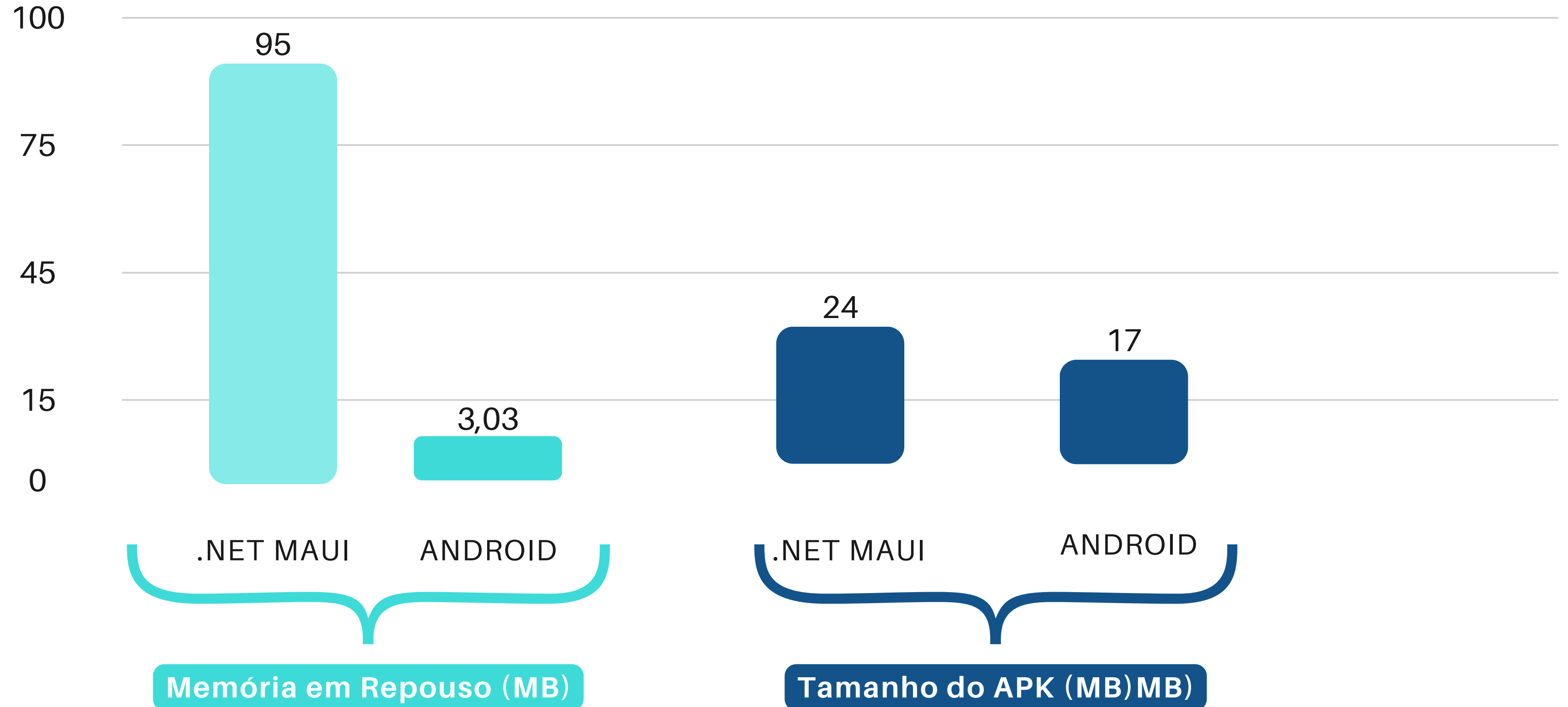
Objetivo: Mostrar a diferença de inicialização dos apps



EFICIÊNCIA DO APP

Objetivo: mostrar de forma visual quais tecnologias consomem menos recursos, combinando memória em repouso e tamanho do pacote final.

VALORES EM MB



RESUMO GERAL DAS MÉTRICAS

Métrica	.NET MAUI (C#)	ANDROID NATIVO (Kotlin)
Cold Start (s)	17,0	19,05
Tamanho do APK (MB)	24 MB	17 MB
Memória em repouso (MB)	95 MB	3,03 MB
Tempo de desenvolvimento (h)	6 h	6 h

ANÁLISE CRÍTICA

.NET MAUI

- Produtividade maior, ajustes rápidos (Hot Reload, C# conhecido);
- Porém → maior uso de memória e pacote maior;

ANDROID NATIVO

- Curva de aprendizado mais íngreme;
- Porém → desempenho mais enxuto, uso eficiente de recursos

Dilema: produtividade x eficiência

CONCLUSÃO & RECOMENDAÇÕES



.NET MAUI

Ideal para apps multiplataforma, foco em rapidez e manutenção unificada;



Android Nativo

ideal quando desempenho e eficiência são críticos;



AMBOS

Atenderam ao escopo proposto com sucesso;