



# BLOCKFACE

UM SISTEMA DE MONITORAMENTO FACIAL BASEADO EM BLOCKCHAINS

MAIS QUE UM SISTEMA DE MONITORAMENTO FACIAL, UM FACE MONITORING SYSTEM

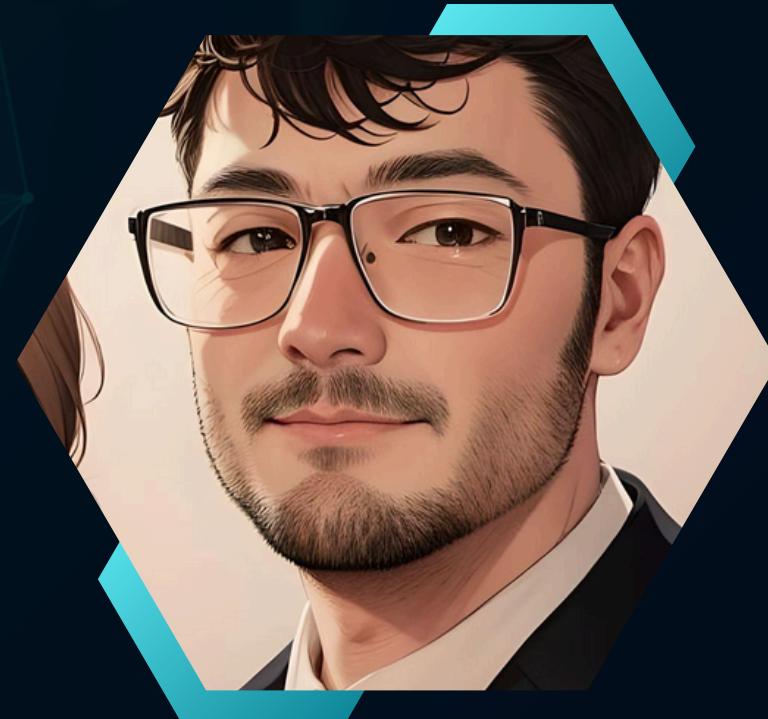
Novembro 2024



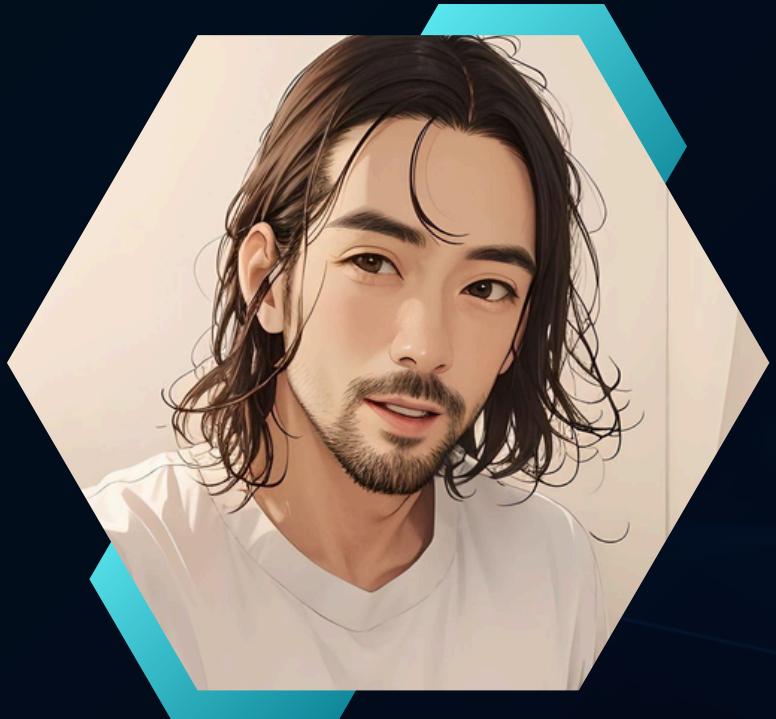
# NOSSO GRUPO



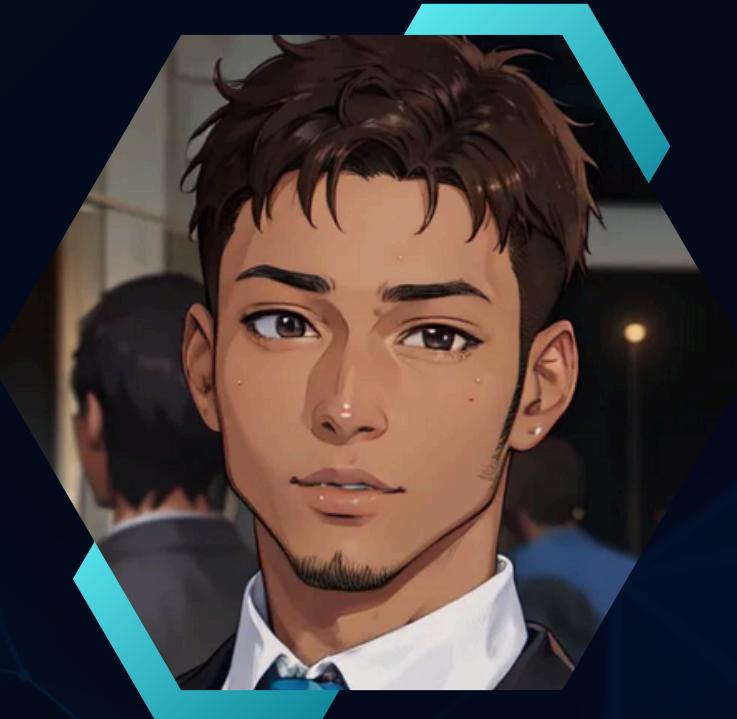
Marcos



Saulo



Bernardo



Super  
Eric

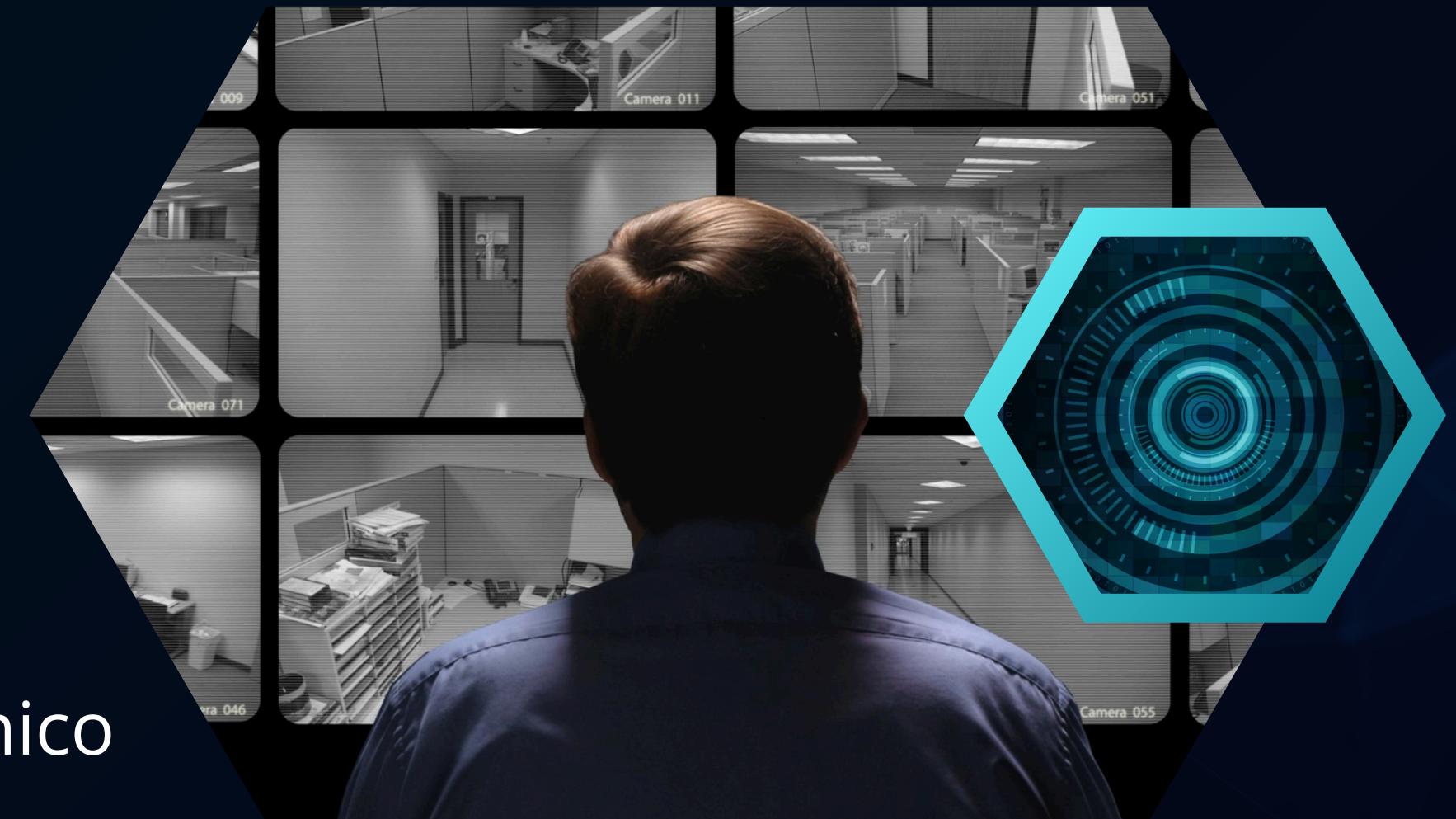


# O PROBLEMA

1 **Vulnerabilidades** a invasões cibernéticas.

2 **Risco de manipulação** de dados de **vídeo**.

3 Centralização dos dados, aumentando o **risco** de um único ponto de **falha**.





# OBJETIVOS

Sistema de **Segurança** de cameras robusto e confiável

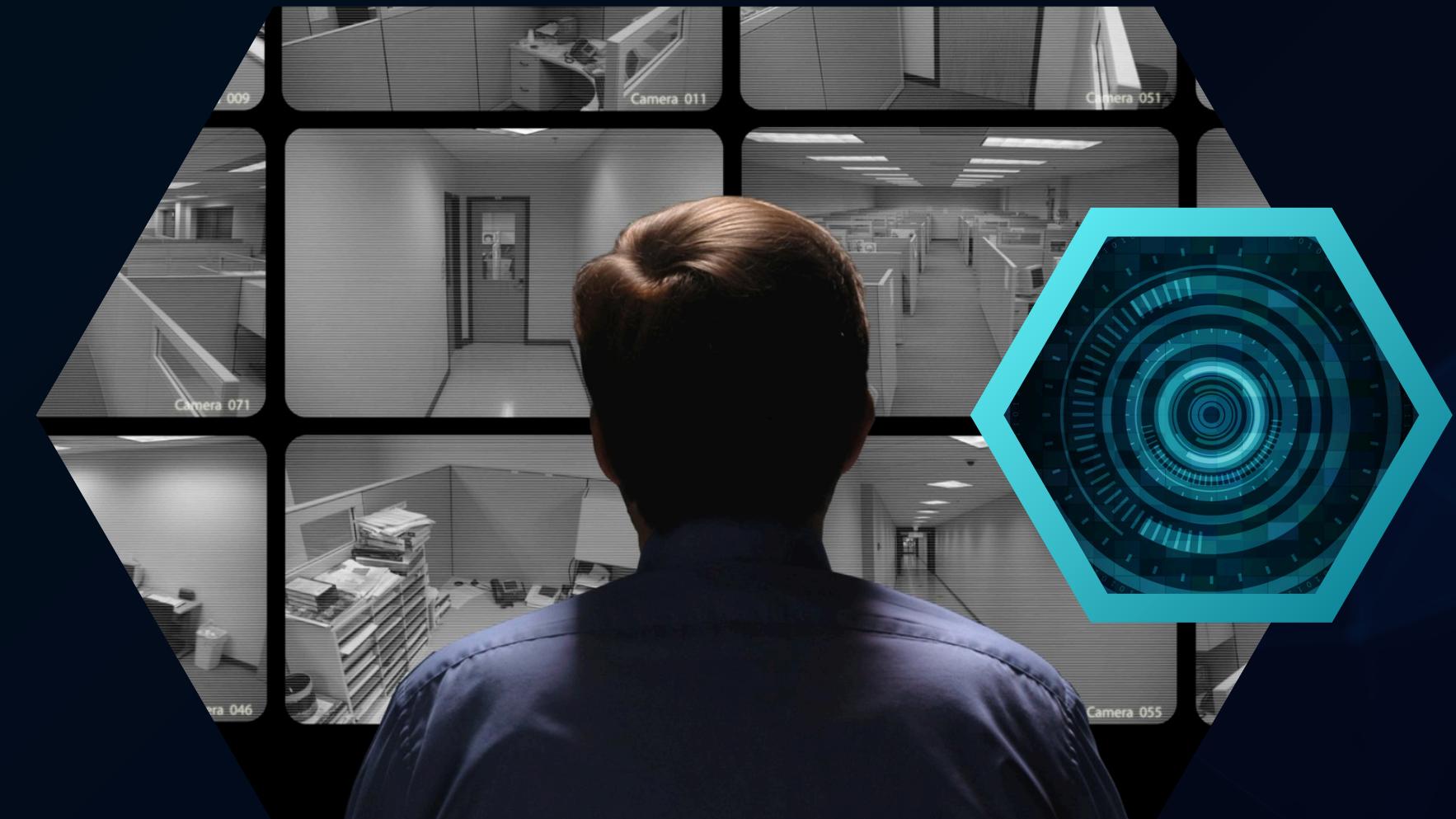
1

Base de dados **distribuída** e **descentralizada** baseada em **blockchain**

2

Implementar mecanismos de **auditoria transparente** para registros de vídeo.

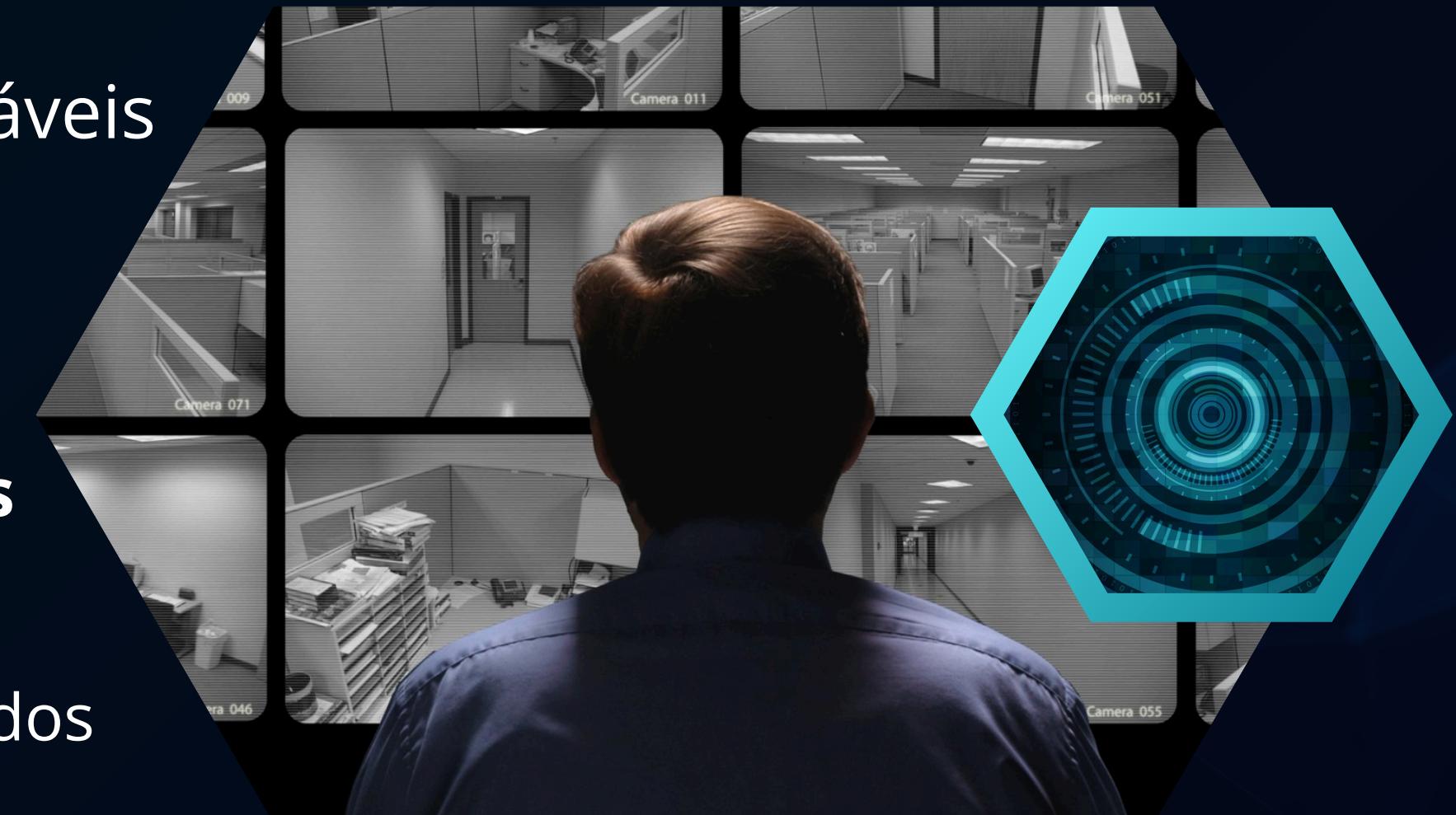
3





# JUSTIFICATIVA

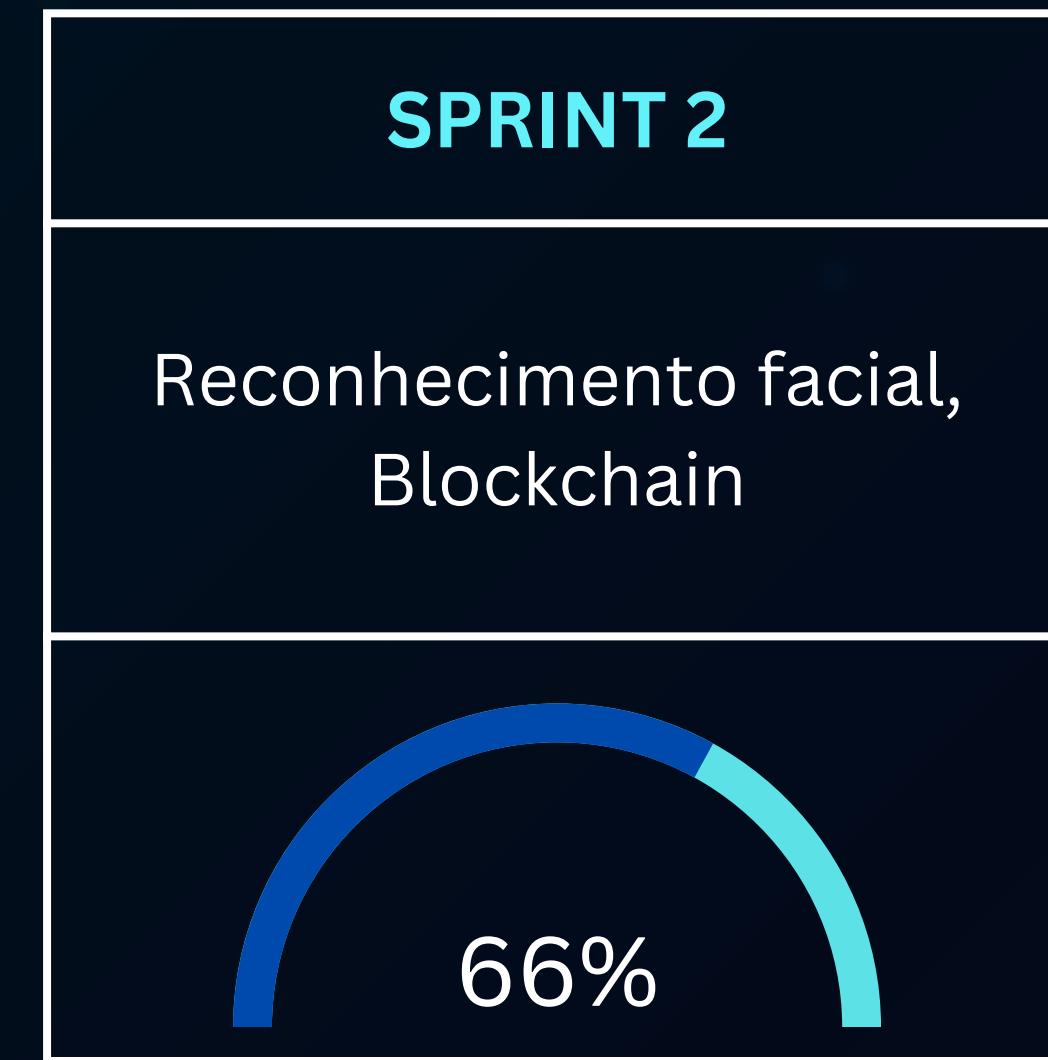
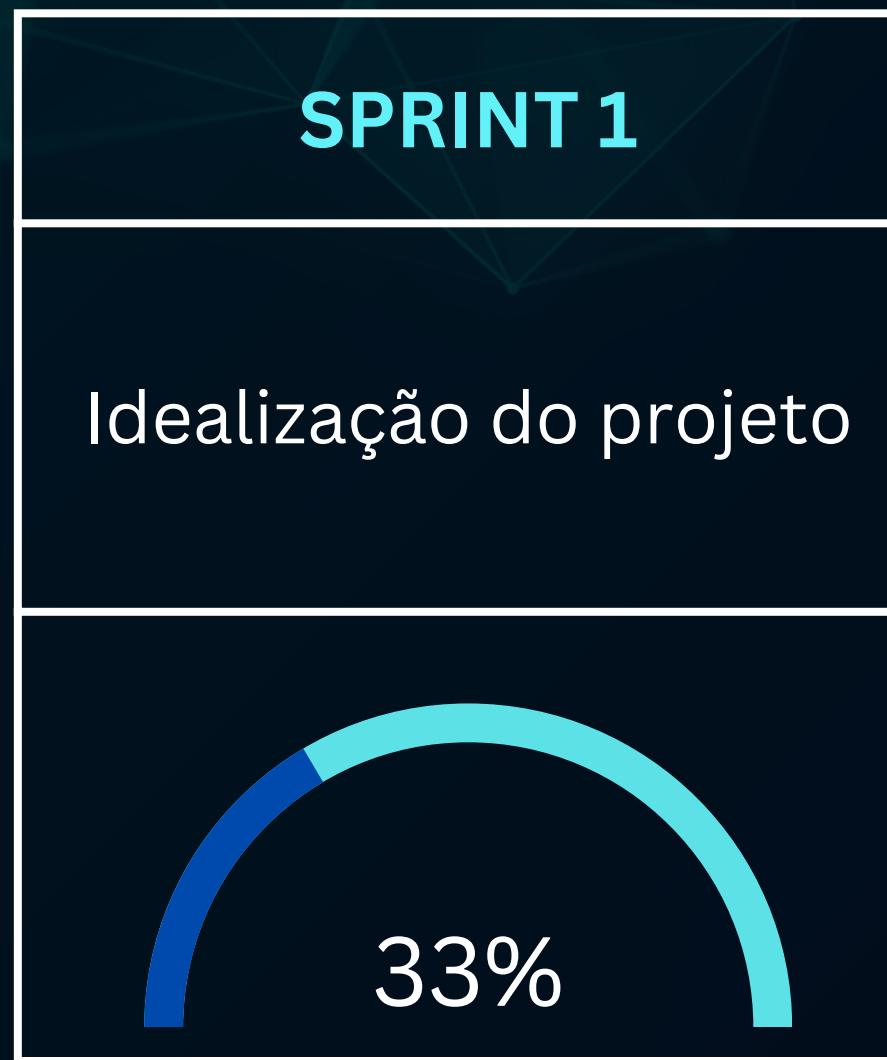
- 1** Segurança: Transações imutáveis e à prova de adulterações.
- 2** **Transparência:** Registros auditáveis por todas as partes interessadas.
- 3** Descentralização: Dados distribuídos que eliminam pontos únicos de falha.



# CRONOGRAMA

## > XP - Extreme Programming

- Orientado a testes
- Programação em par



# REVISÃO BIBLIOGRAFICA

**1 Surveillance Face Recognition Challenge**

1

Reconhecimento e pareamento de imagens de câmeras de segurança

**2 A review on face recognition systems: recent approaches and challenges**

2

Avanços recentes na área de reconhecimento de faces

**3 Enhancing Security and Privacy in Distributed Face Recognition Systems through Blockchain and GAN Technologies**

3

Sistema de câmeras de segurança com armazenamento dos dados em Blockchain

**4 Attendance System based on Blockchain and Face Recognition**

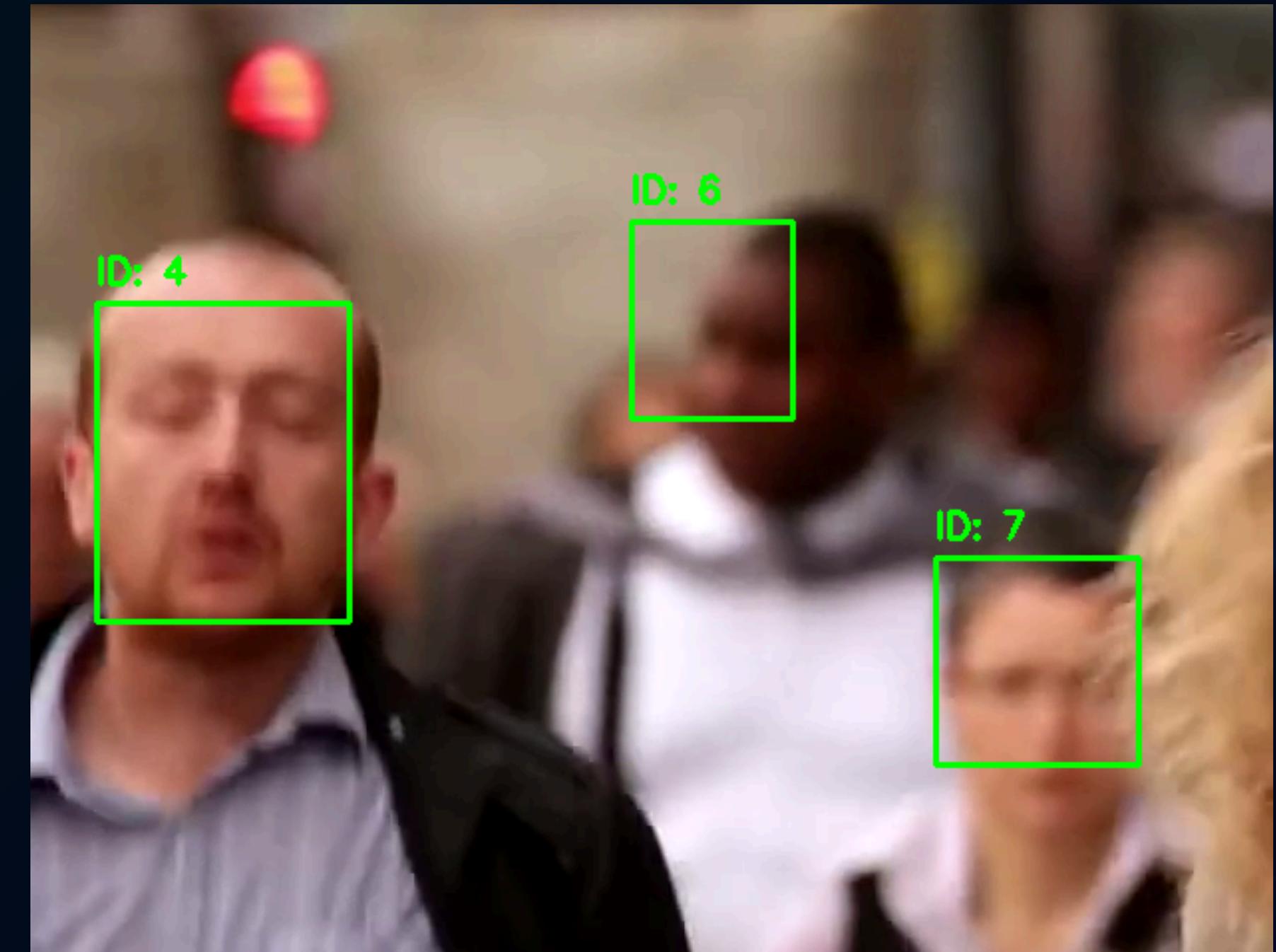
4

# DETEÇÃO

OpenCV / YOLO

1. **YOLO**
2. **NMS** (boxes duplicados)
3. **DeepSort** (rastreamento)

Salva o rosto depois de X frames  
ou quando perdermos a pessoa.



# FACENET

- PYTORCH
- Entrada do modelo: imagem vetorizada
- Retorno do modelo: embedding(incorporação)
  - um vetor de 128 dimensões que exibe uma representação da identidade da pessoa

# INTEGRAÇÃO YOLO & FACENET

- 1 Recorte da face a partir do vídeo da câmera (YOLO)
- 2 imagem da face é processada pelo FaceNet
- 3 Ao fim do processamento, as características (embedding) é salvo, assim como a imagem

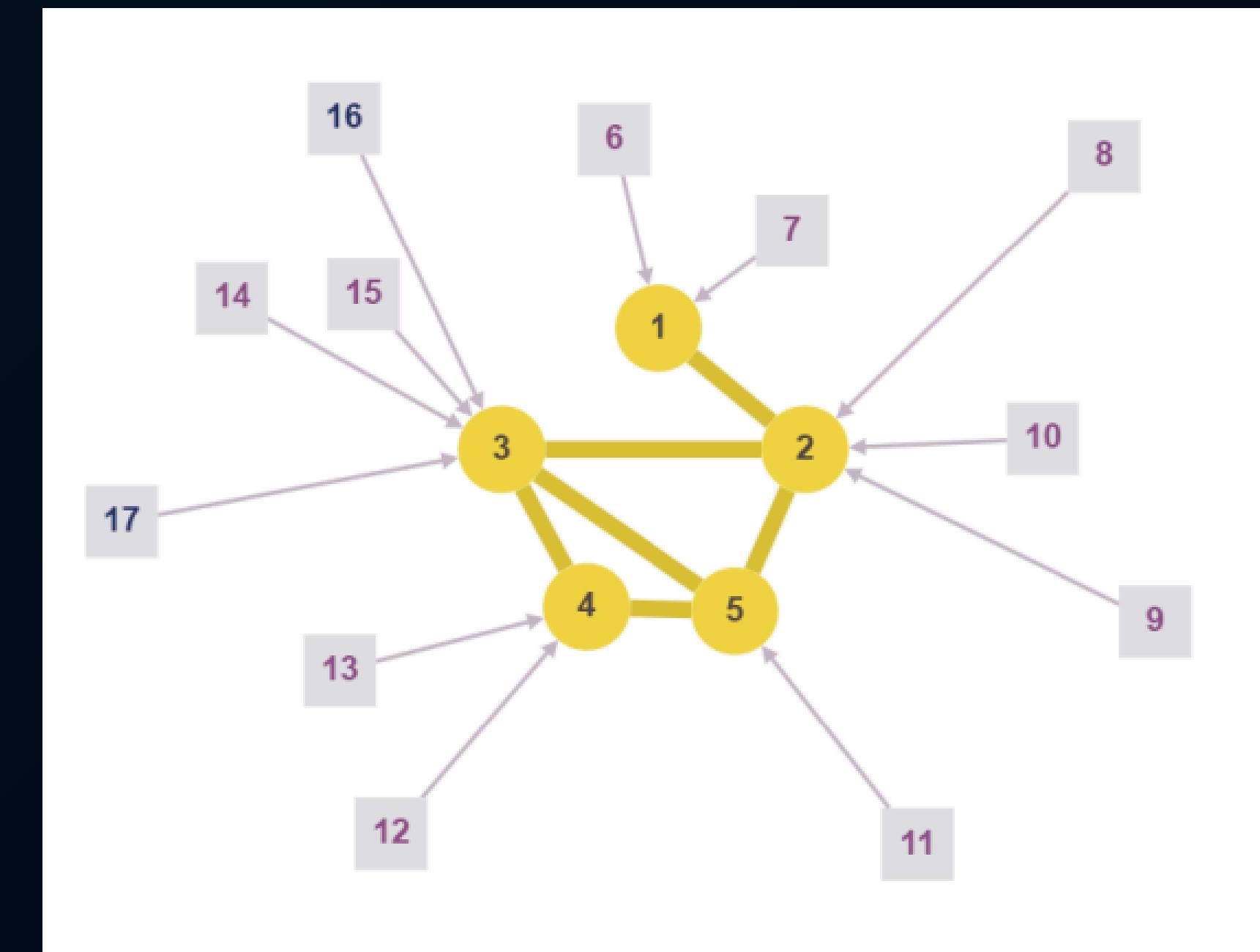
# BLOCKCHAIN

- **Carteira digital - Chaves**
- **Arquivos:**
  - **Imagens**
  - **Metricas faciais**



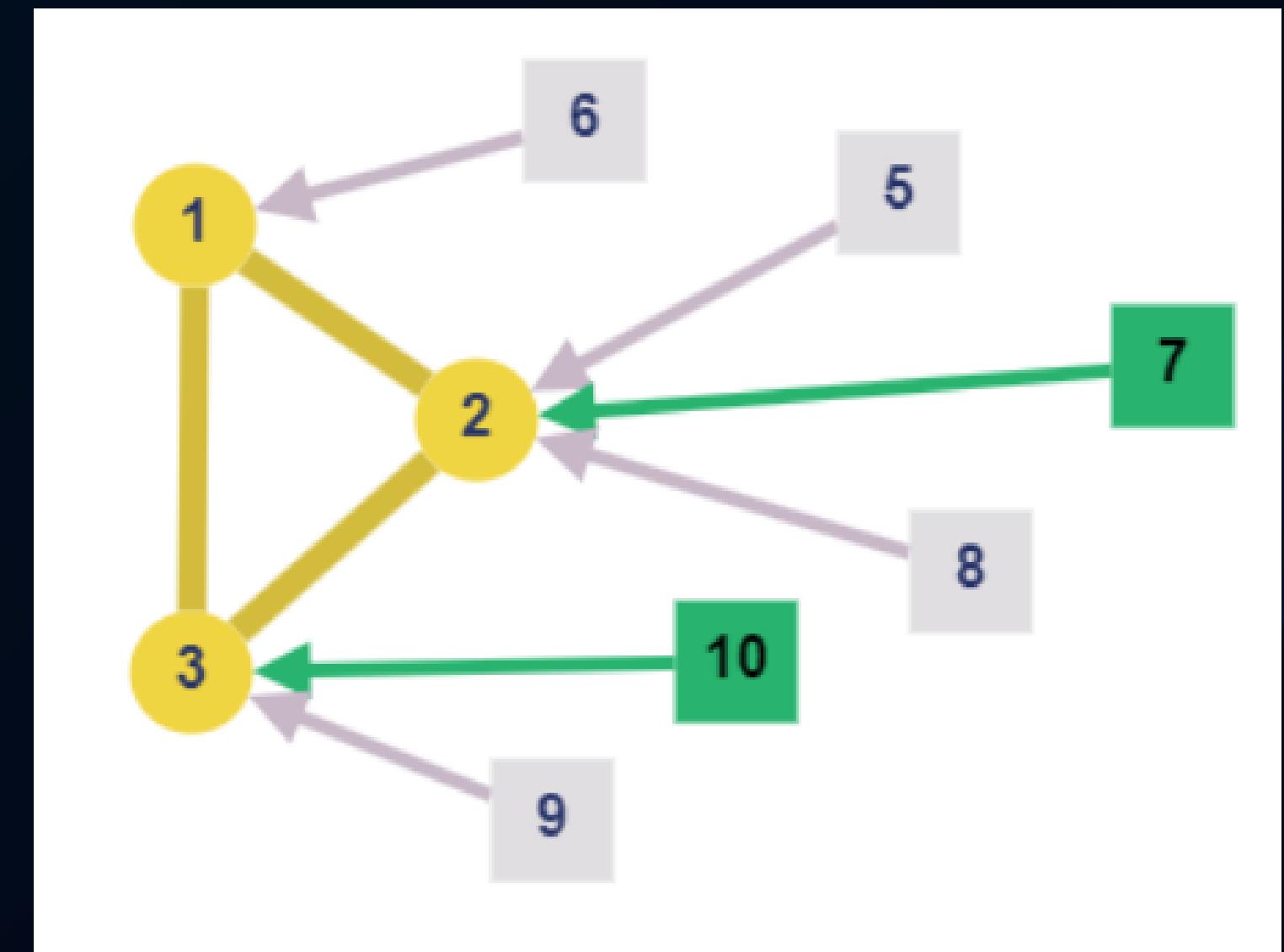
# BLOCKCHAIN

- Base de dados distribuída
- Grupo de líderes - Malha Parcial
- Nós subordinados - Estrela
- SHA-256 - Hash de blocos
- ECDSA - Assinatura Digital de Curvas Elípticas
- P2P - IPV6



# PROOF OF WORK

- Consenso Híbrido:
  - PoW - Proof of Work
  - PoA - Proof of Authority
- Processo de eleição
- Favorece computadores mais potentes



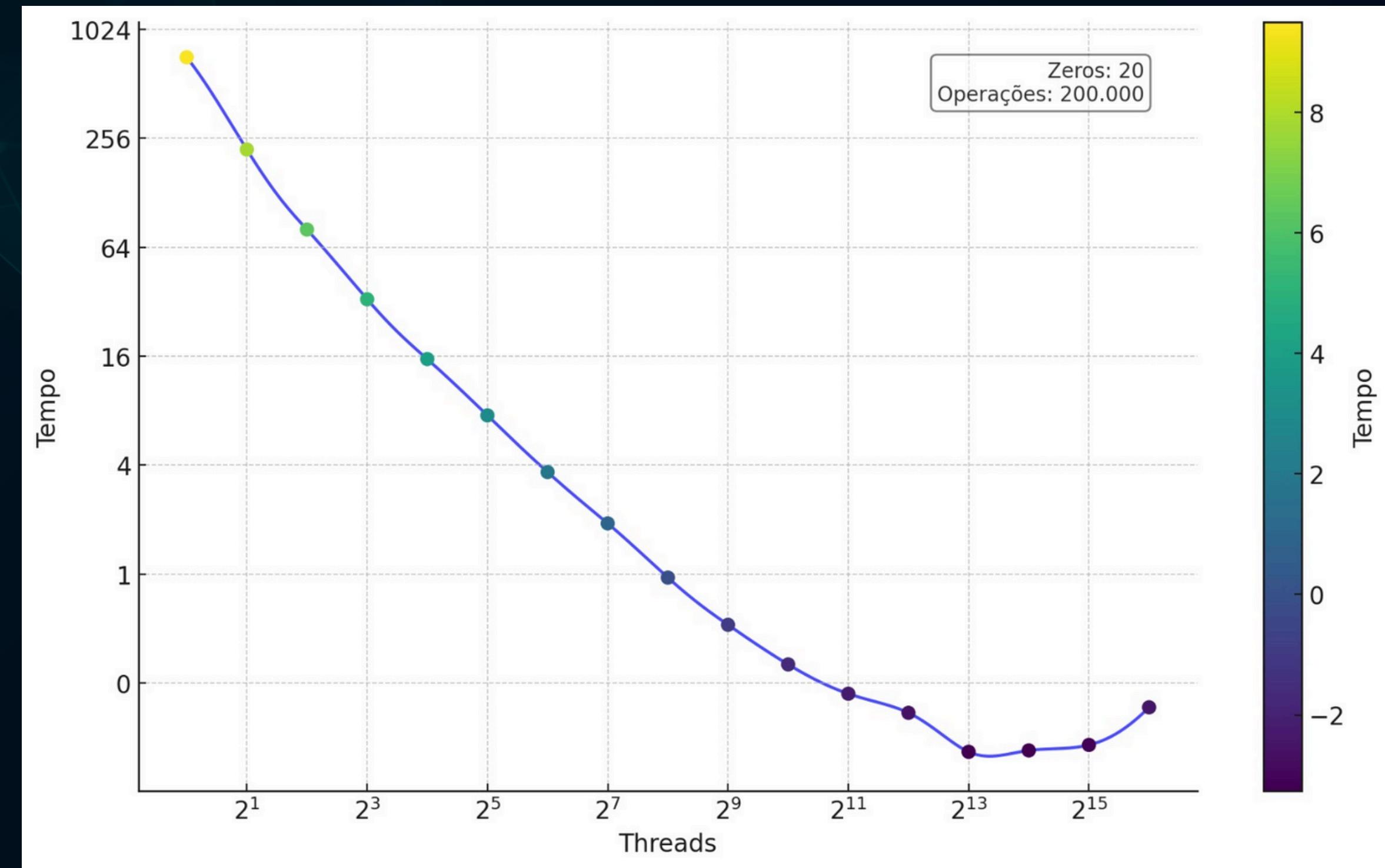
# METRICAS

- Sistema operacional: linux mint 21.1
- CPU: QUAD CORE MODEL-intel i5 - 11300H
- GPU: RTX 3050
- Memória RAM: 16GB

**USO DE GREENTHREADS**



# STRONG SCALING



# WEAK SCALING

- Fórmula usada:  $P = \frac{1}{2^n}$

Onde:

- P é a probabilidade de encontrar o número
- n é o número de zeros à esquerda.

# WEAK SCALING

- Fórmula usada:

$$N_t = \left\lceil \frac{E}{T} \right\rceil$$

Onde:

- $N_t$  é o número de threads necessárias
- $E$  é o número esperado de operações
- $T$  é o número de operações por thread

# WEAK SCALING

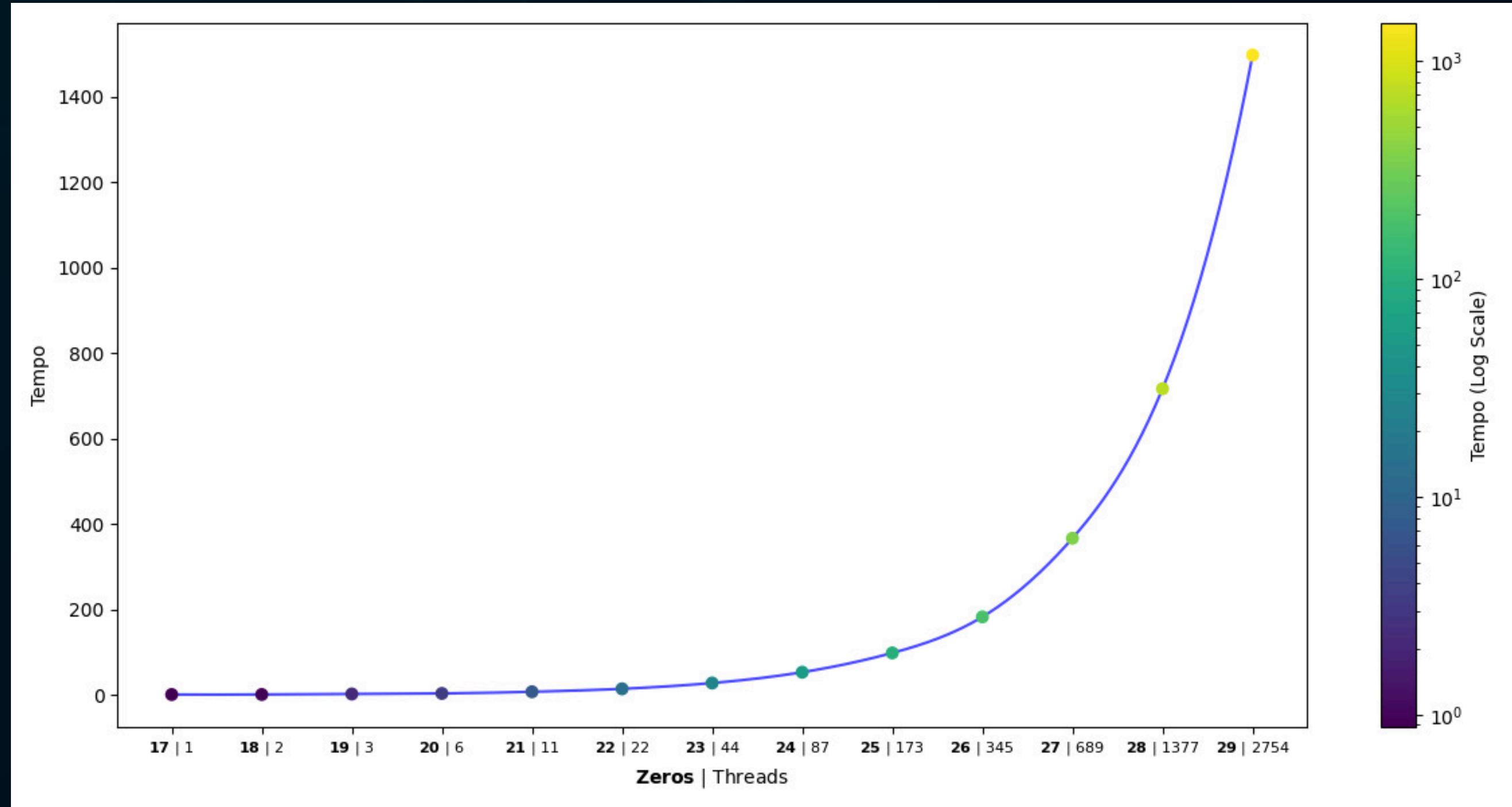
- Fórmula usada:

$$O = \frac{\ln(1-p)}{\ln(-1/2)}^n$$

Onde:

- O é o número de operações para P probabilidade
- P é a probabilidade de encontrar o número
- n é o número de zeros à esquerda.

# WEAK SCALING



Perca de  
eficiência em  
**25 zeros ou**  
**87 threads**



# RESULTADOS DA INTEGRAÇÃO

- 0% de Loss durante o treinamento do modelo FACENET
- 100% de acurácia do modelo treinado FACENET+QMUL-SurvFace
- O modelo manteve a mesma acurácia que foi proposto



# OBRIGADO!!!

