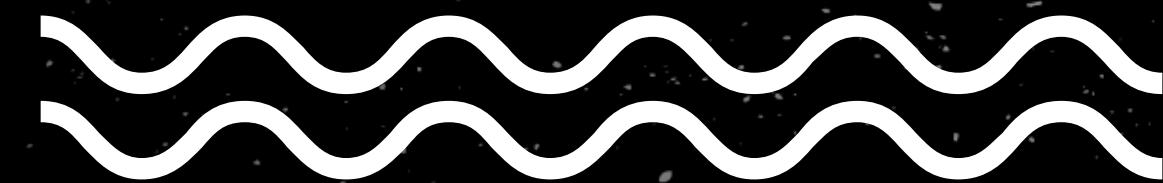




# COMPUTAÇÃO QUÂNTICA INTELIGENTE

Weverton Davi da Silva Farias  
Gabriel Amorim Crisóstomo



# SUMÁRIO

01

INTRODUÇÃO

Quem somos

02

MOTIVAÇÃO E PROPOSTA

Base de dados e Metodologia

03

PROCEDIMENTO

Desenvolvimento

04

CONCLUSÃO

# INTRODUÇÃO

- Weverton Davi da Silva Farias
- Análise de Desenvolvimento de Sistemas
- Universidade Maurício de Nassau
  
- Gabriel Amorim Crisóstomo
- Engenharia Eletrônica
- Universidade Federal de Pernambuco.

"Nature isn't classical, damn it,  
and if you want to make a  
simulation of nature, you'd  
better make it quantum  
mechanical."

— RICHARD FEYNMAN, 1981

(CONFERÊNCIA NO MIT SOBRE SIMULAÇÃO  
QUÂNTICA)



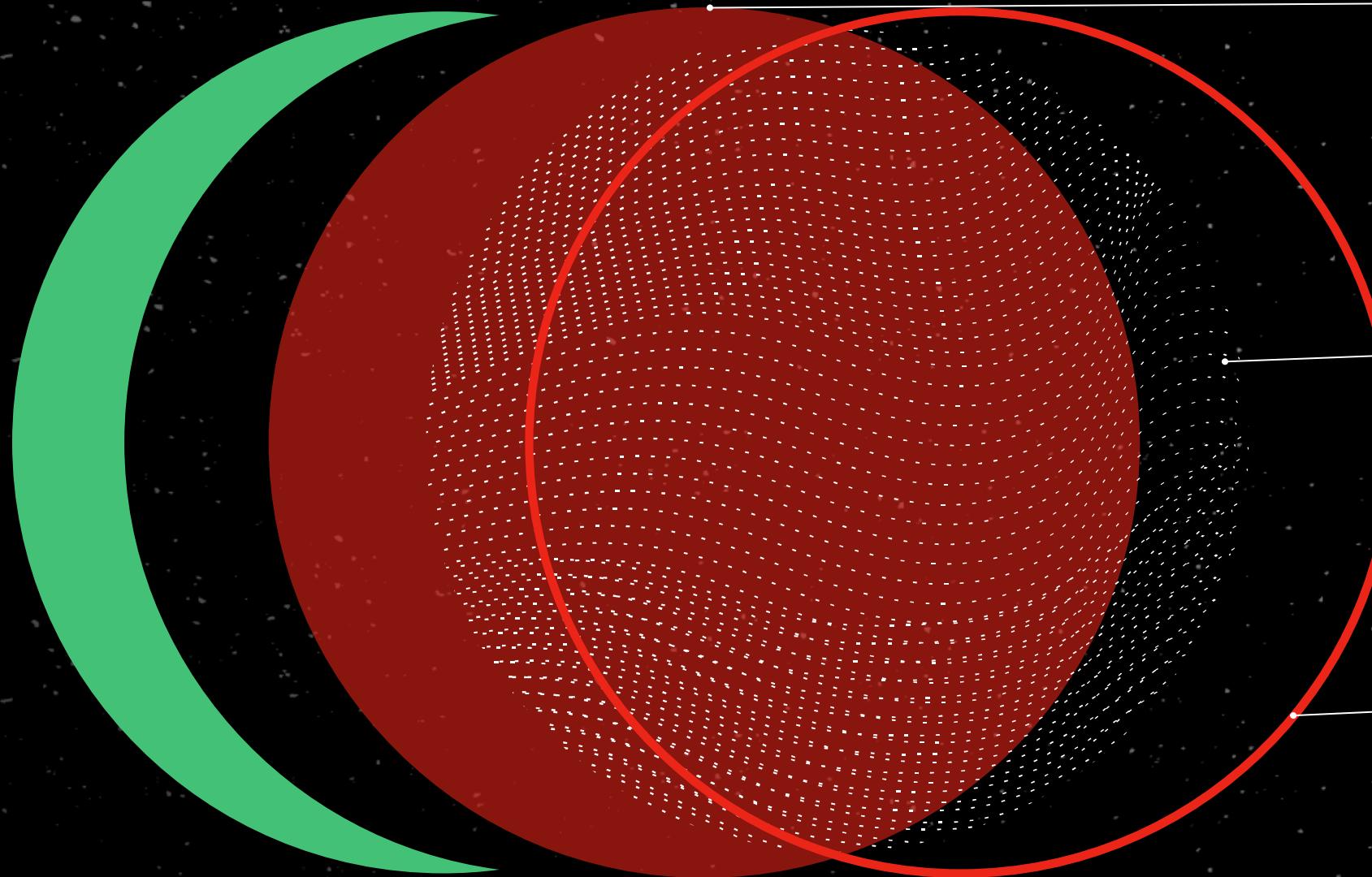


# MOTIVAÇÃO E PROPOSTA 02

Base de dados e metodologia escolhida



# METODOLOGIA E PROPOSTA



## MOTIVAÇÃO

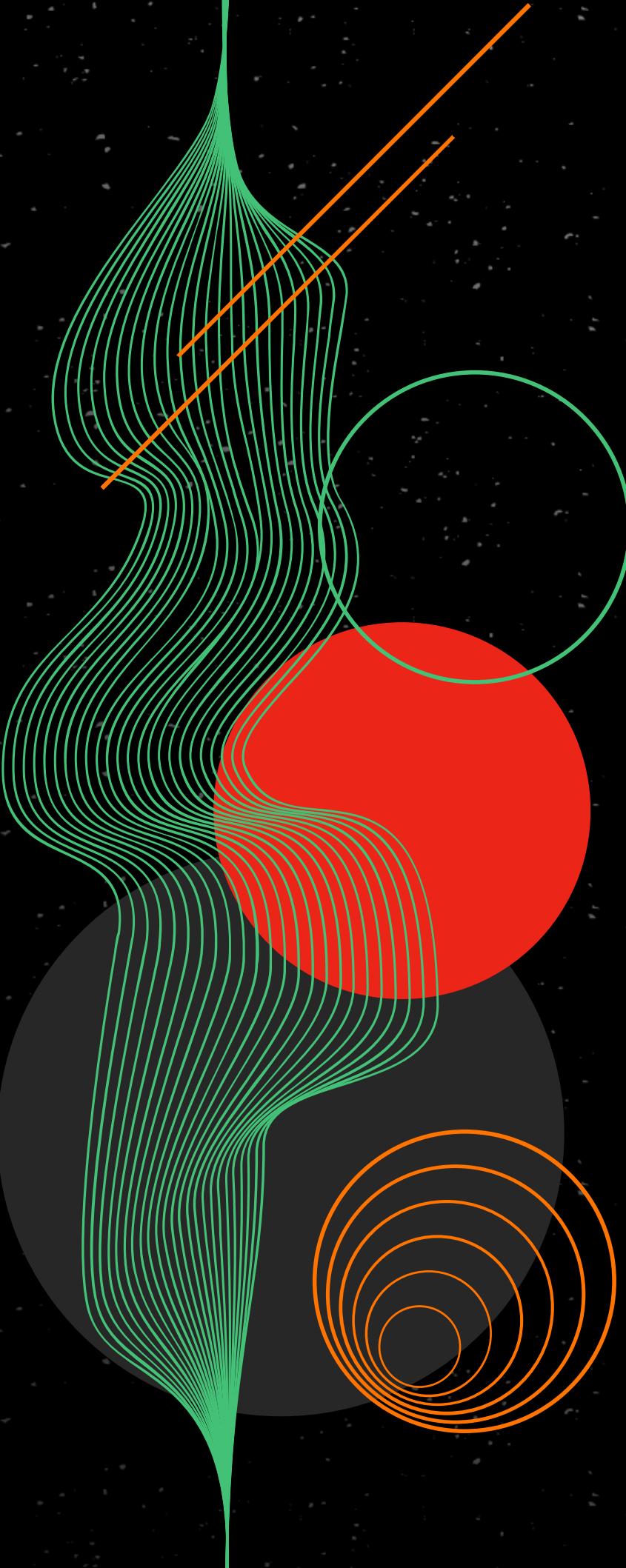
Conectar uma demanda pública  
a computação quântica  
inteligente

## METODOLOGIA

Análise comparativa de  
diferentes métodos de  
aprendizado de máquina

## BASE DE DADOS

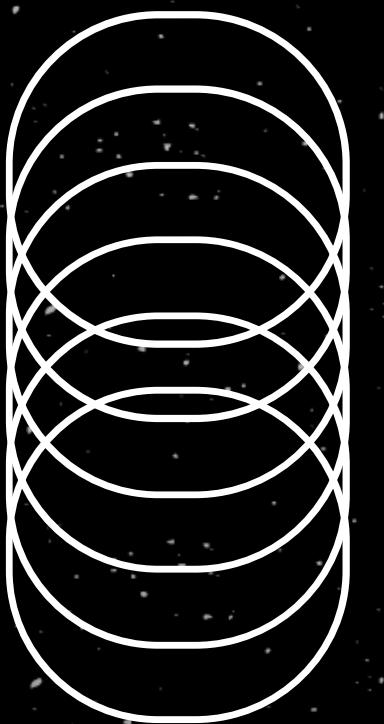
Pública e temática sobre saúde  
pública



# MOTIVAÇÃO

## Saúde Pública

- Desenvolvimento de Política Pública
- Melhoria de setores de saúde
- Expansão do método para outros conjuntos de dados
- Aplicação ampla do Quantum Machine Learning



# METODOLOGIA

CLASSIFICADOR  
VARIACIONAL  
QUÂNTICO

BAIXÍSSIMO CUSTO  
COMPUTACIONAL

BAIXA ACURÁCIA

CLASSIFICADOR  
MULTICLASSE

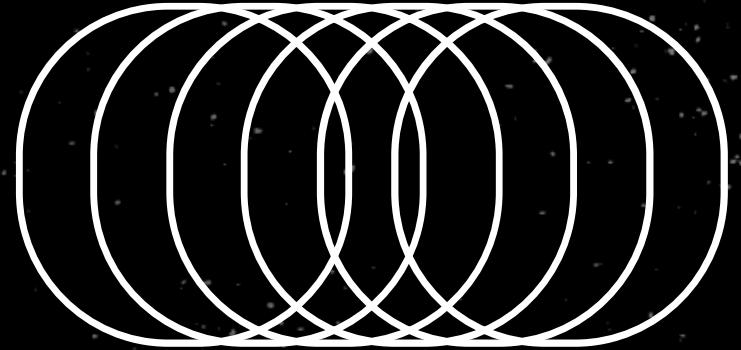
ALTO CUSTO  
COMPUTACIONAL

ALTA ACURÁCIA

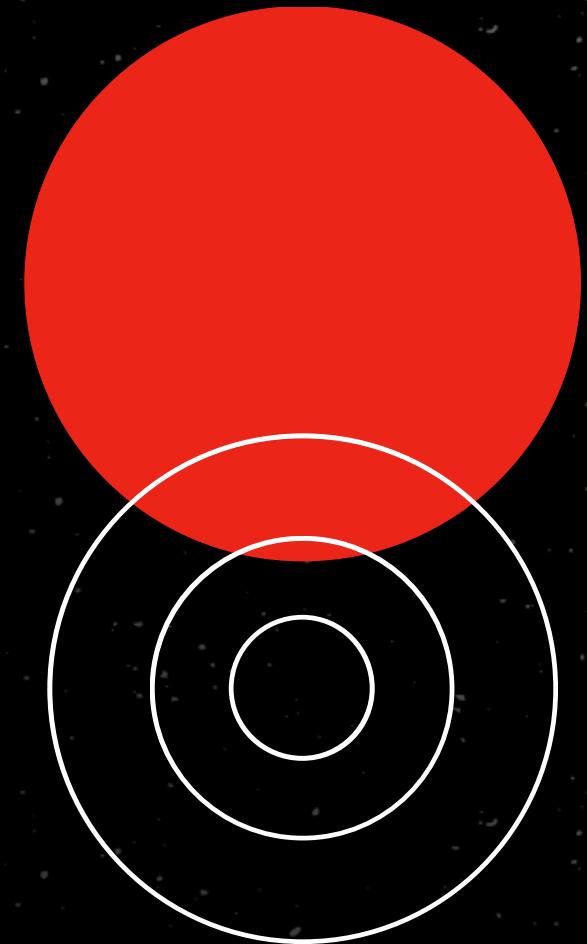
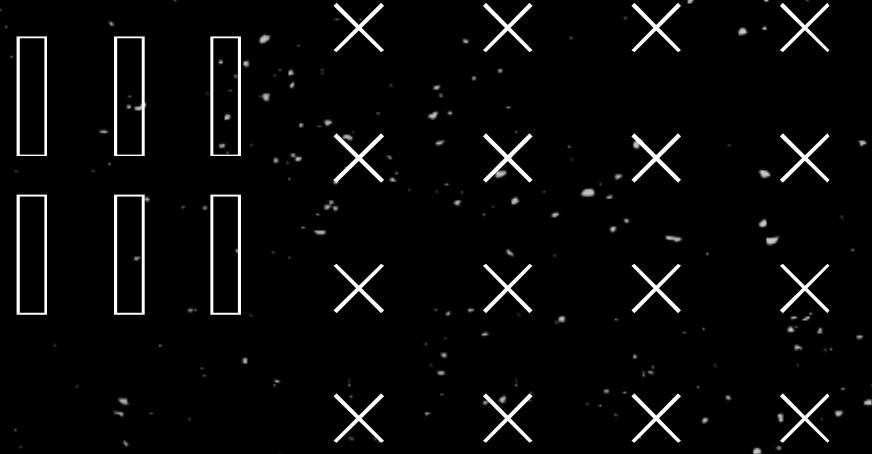
DATA  
REUPLOADING

BAIXO CUSTO  
COMPUTACIONAL

ACURÁCIA MÉDIA



# CIRCUITO QUÂNTICO VARIACIONAL



1

A versão mais básica:  
embute os dados, aplica  
camadas de  
entanglement + portas  
variacionais, e mede  $\langle Z \rangle$ .

3

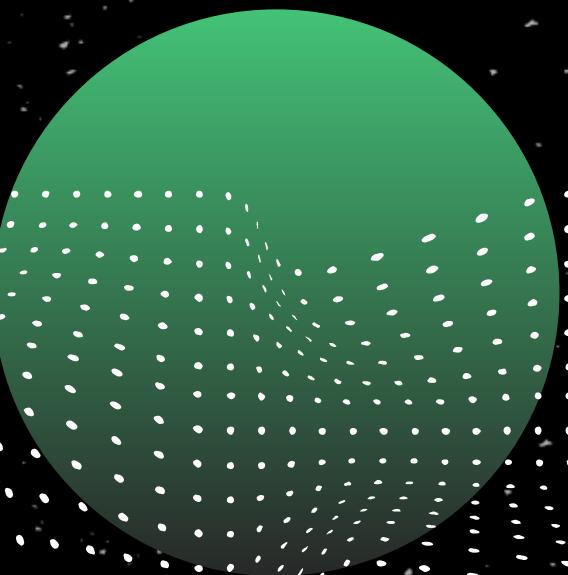
Ótimo como introdução  
a aprendizado de  
máquina quântico.

2

Tipicamente para tarefas  
binárias, mas pode ser  
estendido.

## CONTRAS

Pouca expressividade,  
Ruído em hardware real



# CLASSIFICADOR MULTICLASSE

1

Usa um único circuito, com medições em diferentes qubits ou observáveis correspondendo a cada classe.

2

A saída do circuito é um vetor de probabilidades (via softmax), como em classificadores clássicos.

3

Exige mais medições e observáveis – o que aumenta o custo experimental.

## CONTRAS

Escalabilidade ruim, Treinamento mais difícil, Hardware limitado

# DATA REUPLOADING

1

Inspirado em redes neurais profundas: permite inserir os dados várias vezes no circuito, entre portas parametrizadas.

2

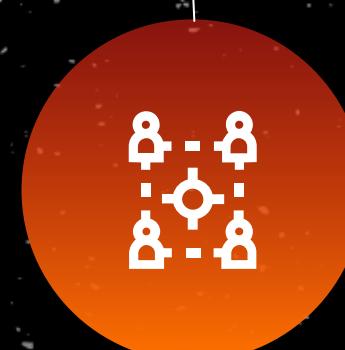
Aumenta a capacidade de representar funções mais complexas.

3

Pode ser usado tanto em classificação binária quanto multiclasse.

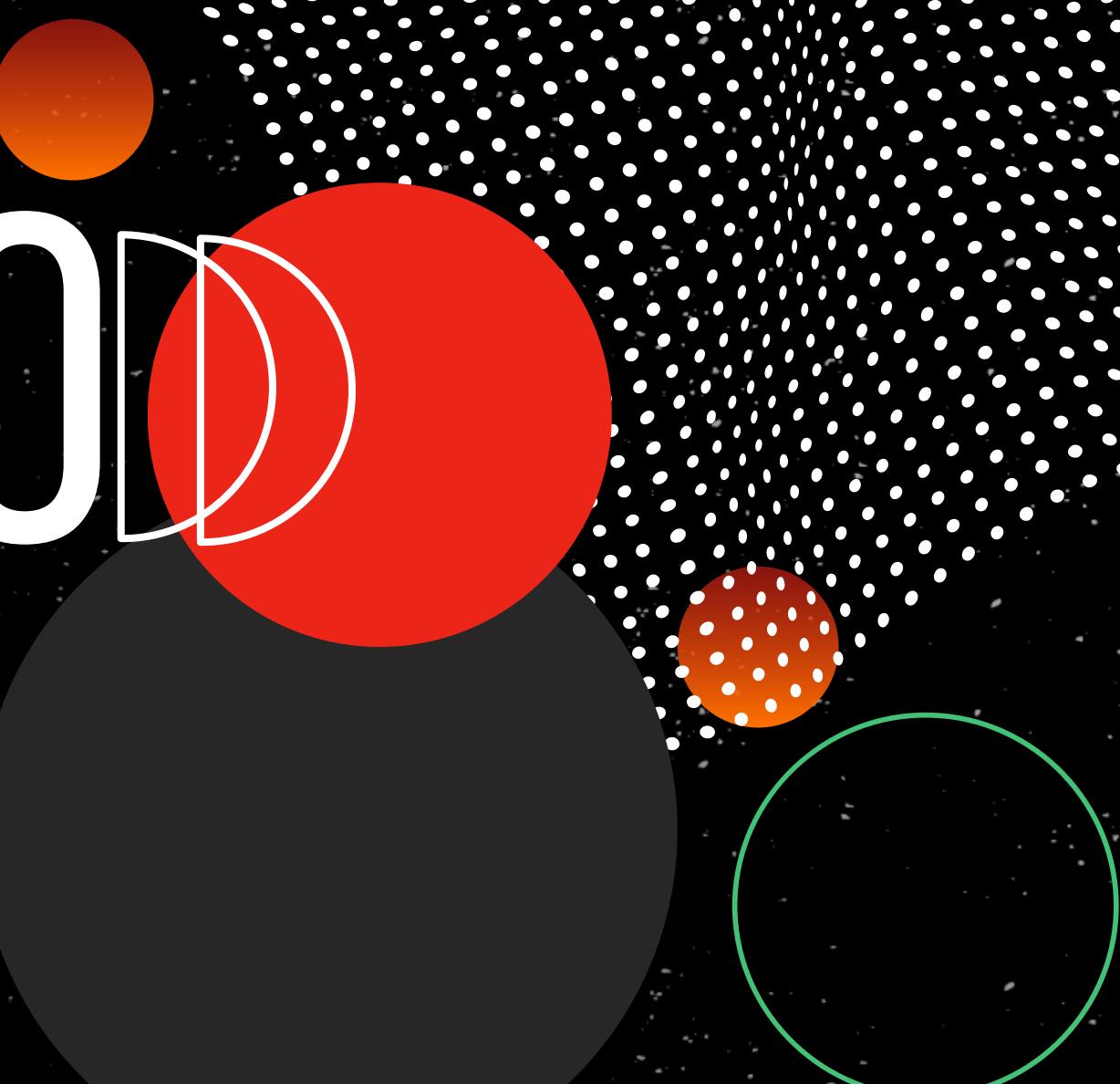
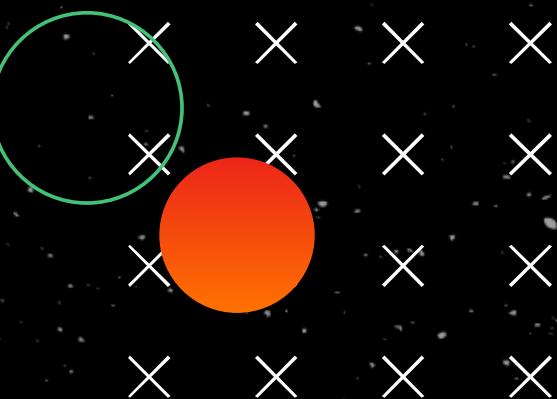
## CONTRAS

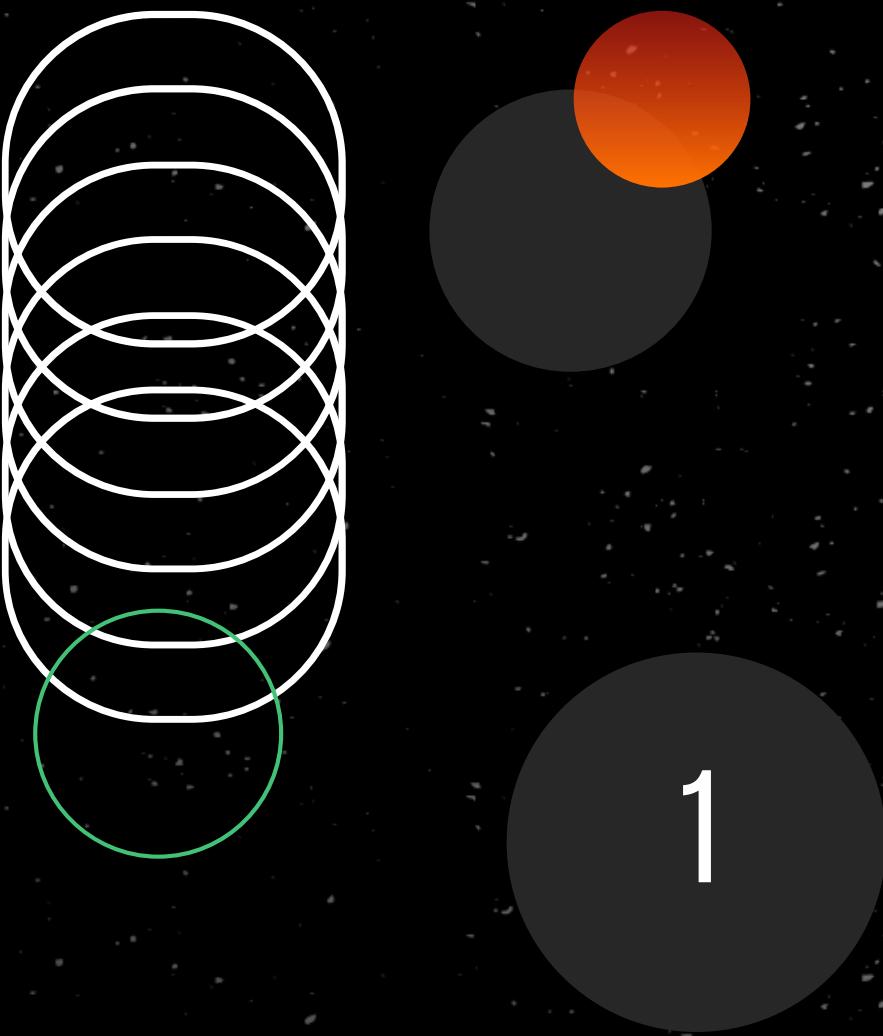
Circuitos mais profundos, Mais parâmetros, Risco de barren plateaus



03

# PROCEDIMENTO





## FONTE DE PESQUISA, DADOS E REFERÊNCIA

1

KAGGLE

Fonte de dados  
pública sobre  
transfusão de sangue  
para treinar o nosso  
sistema

2

PENNYLANE

Variational classifier

3

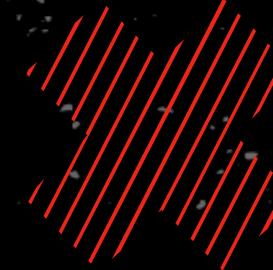
PENNYLANE

Multiclass margin  
classifier

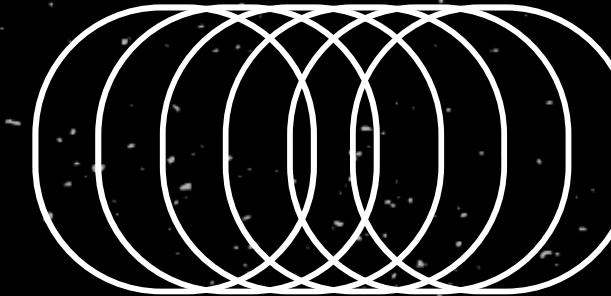
4

PENNYLANE

Data-reuploading  
classifier



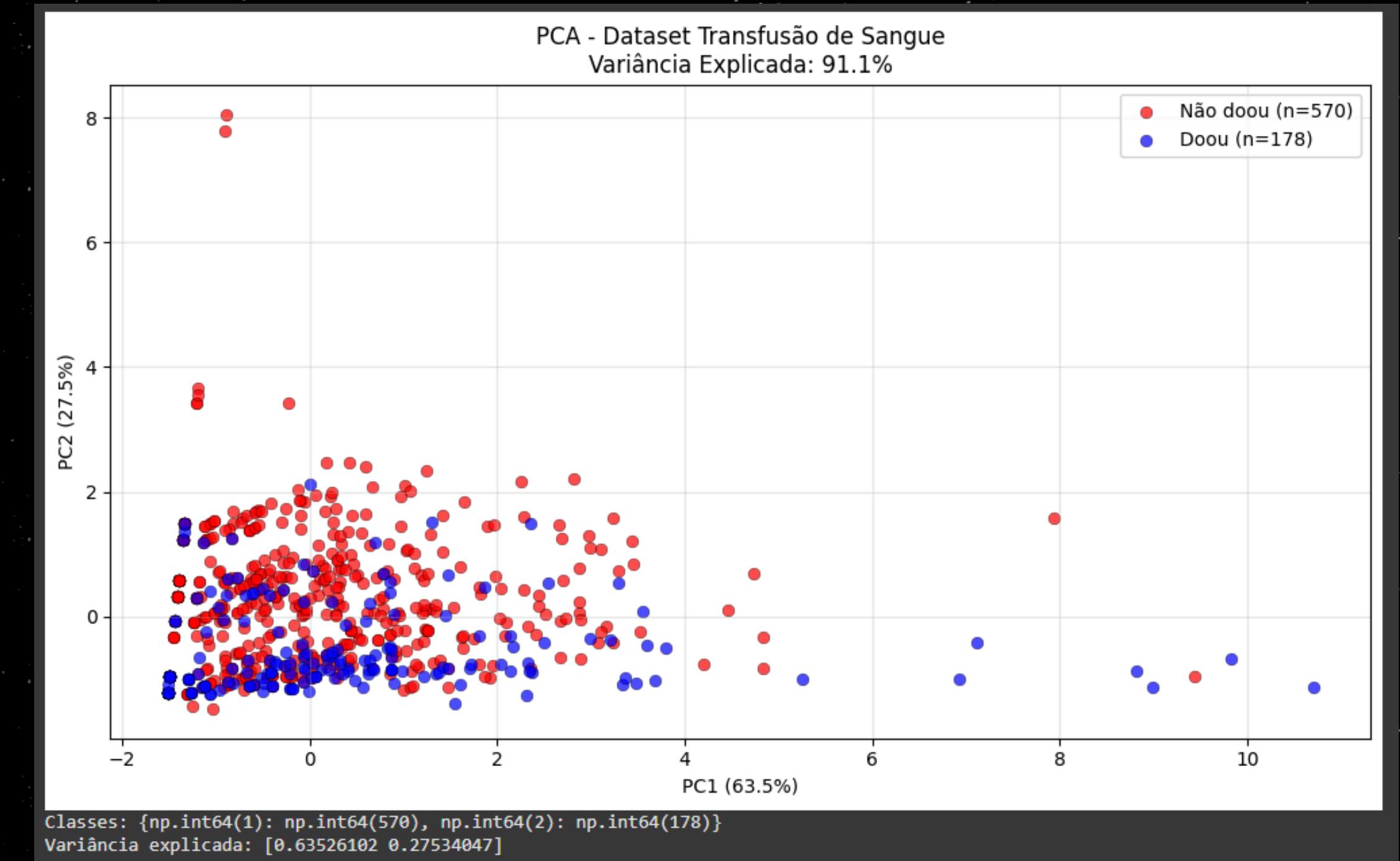
# CIRCUITO VARIACIONAL QUÂNTICO



48 SEGUNDOS DE  
IMPLEMENTAÇÃO

```
Step 10: Accuracy = 0.6873
Step 20: Accuracy = 0.6957
Step 30: Accuracy = 0.6472
Step 40: Accuracy = 0.6856
Step 50: Accuracy = 0.6806
```

Test Accuracy: 0.7000

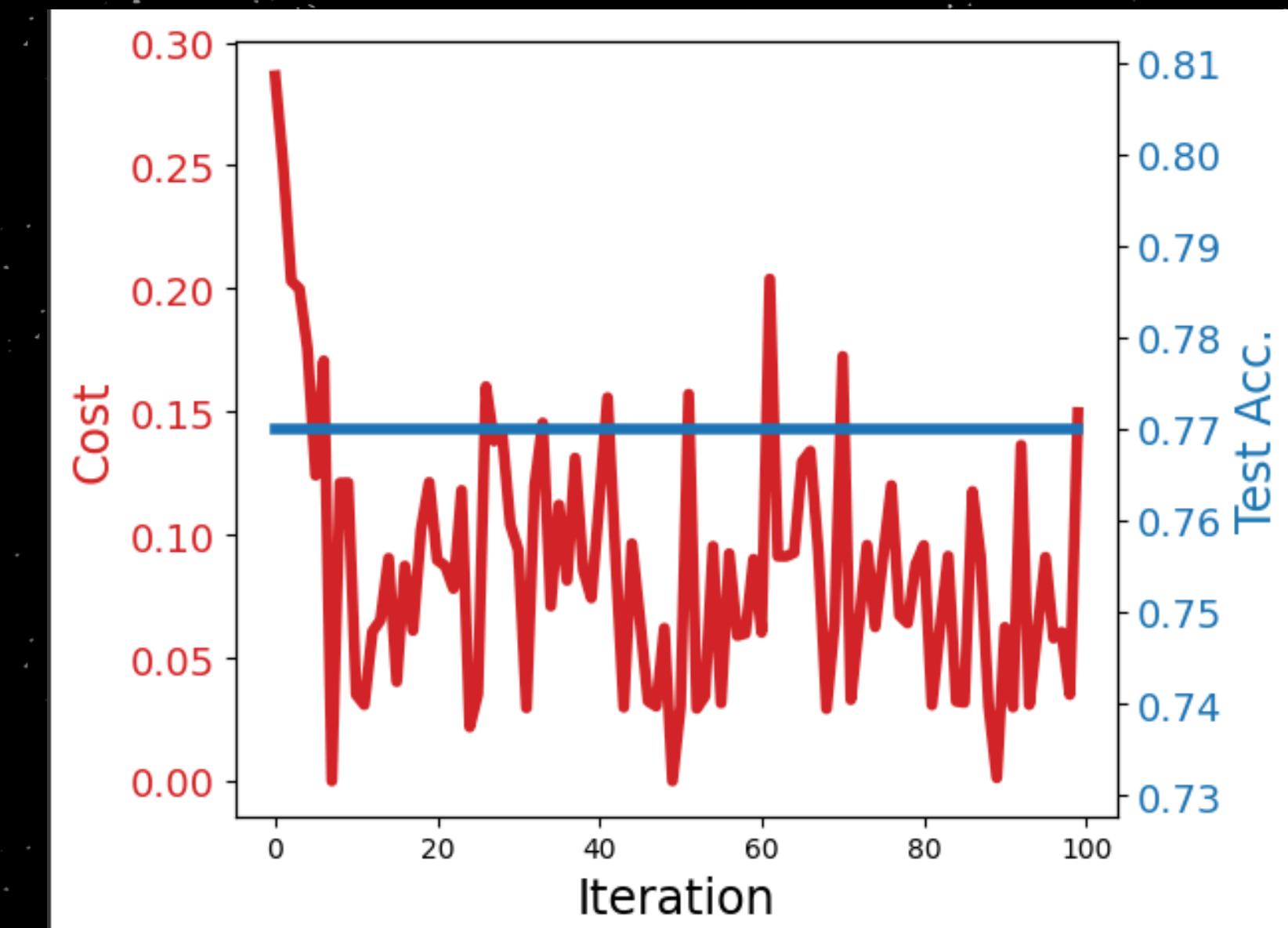


# MULTICLASS MARGIN CLASSIFIER

1 HORA E 25 SEGUNDOS DE IMPLEMENTAÇÃO

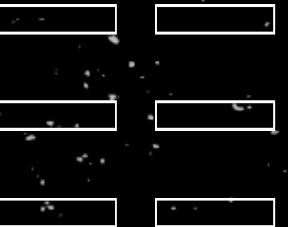
```
Iter: 81 | Cost: 0.0956556 | Acc train: 0.7593583 | Acc test: 0.7700535
Iter: 82 | Cost: 0.0308565 | Acc train: 0.7593583 | Acc test: 0.7700535
Iter: 83 | Cost: 0.0620595 | Acc train: 0.7593583 | Acc test: 0.7700535
Iter: 84 | Cost: 0.0913061 | Acc train: 0.7593583 | Acc test: 0.7700535
Iter: 85 | Cost: 0.0323315 | Acc train: 0.7593583 | Acc test: 0.7700535
Iter: 86 | Cost: 0.0318809 | Acc train: 0.7593583 | Acc test: 0.7700535
Iter: 87 | Cost: 0.1175840 | Acc train: 0.7593583 | Acc test: 0.7700535
Iter: 88 | Cost: 0.0912505 | Acc train: 0.7593583 | Acc test: 0.7700535
Iter: 89 | Cost: 0.0303741 | Acc train: 0.7593583 | Acc test: 0.7700535
Iter: 90 | Cost: 0.0012719 | Acc train: 0.7593583 | Acc test: 0.7700535
Iter: 91 | Cost: 0.0624079 | Acc train: 0.7593583 | Acc test: 0.7700535
Iter: 92 | Cost: 0.0299679 | Acc train: 0.7593583 | Acc test: 0.7700535
Iter: 93 | Cost: 0.1363375 | Acc train: 0.7593583 | Acc test: 0.7700535
Iter: 94 | Cost: 0.0309598 | Acc train: 0.7593583 | Acc test: 0.7700535
Iter: 95 | Cost: 0.0608788 | Acc train: 0.7593583 | Acc test: 0.7700535
Iter: 96 | Cost: 0.0908794 | Acc train: 0.7593583 | Acc test: 0.7700535
Iter: 97 | Cost: 0.0579968 | Acc train: 0.7593583 | Acc test: 0.7700535
Iter: 98 | Cost: 0.0604443 | Acc train: 0.7593583 | Acc test: 0.7700535
Iter: 99 | Cost: 0.0350583 | Acc train: 0.7593583 | Acc test: 0.7700535
Iter: 100 | Cost: 0.1495984 | Acc train: 0.7593583 | Acc test: 0.7700535
```

```
First X sample, original : tensor([2.0000e+00, 5.0000e+01, 1.2500e+04, 9.8000e+01])
First X sample, normalized: tensor([1.5999e-04, 3.9998e-03, 9.9996e-01, 7.8397e-03])
Num params: 111
```



# DATA-REUPLOADING CLASSIFIER

x x x x  
x x x x  
x x x x  
x x x x



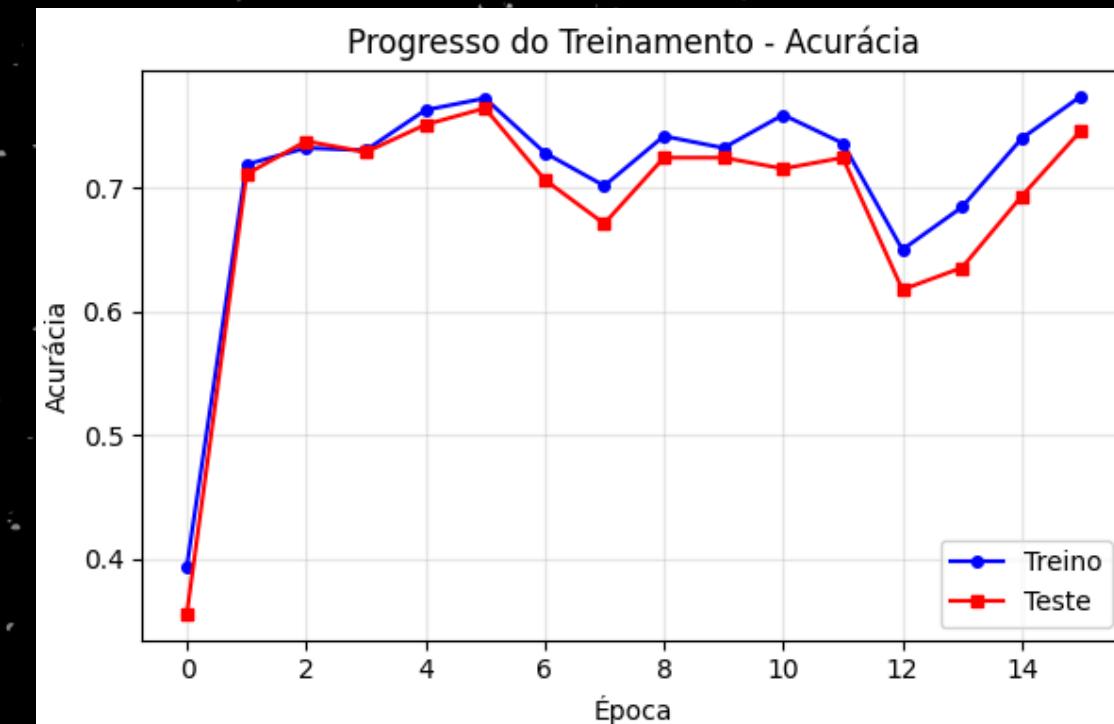
## 4 MINUTOS DE IMPLEMENTAÇÃO



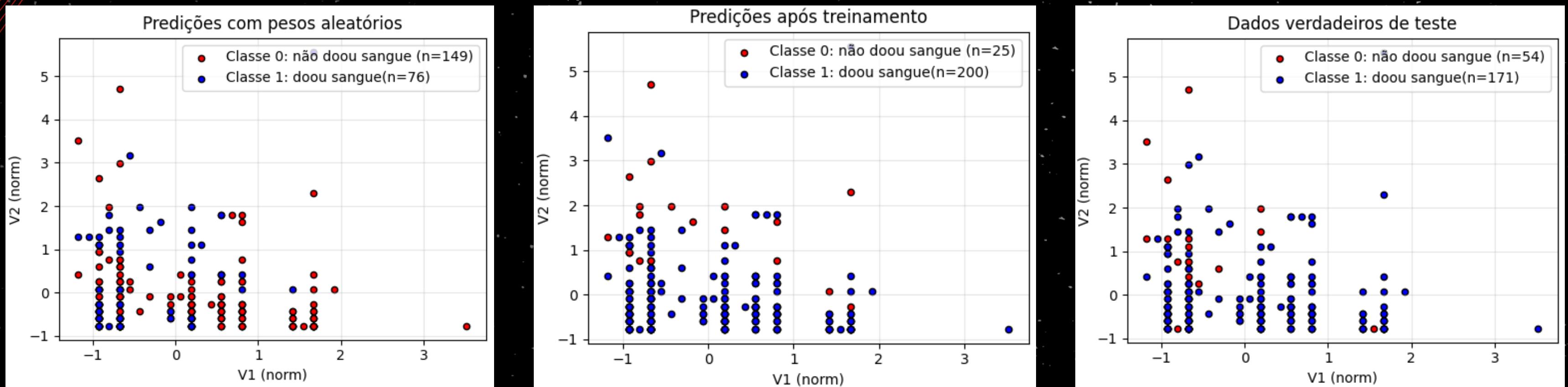
```
Camada 1: [ 5.18649045 -2.39881017  1.97251389]  
Camada 2: [-2.18699815 -3.23151363 -1.57032677]  
Camada 3: [-3.59710487  3.53512445  0.93855271]
```



```
Análise detalhada:  
Distribuição das classes verdadeiras: Counter({tensor(1, requires_grad=True): 171, tensor(0, requires_grad=True): 54})  
Distribuição das previsões: Counter({tensor(1, requires_grad=True): 200, tensor(0, requires_grad=True): 25})  
Matriz de confusão:  
TP: 157, TN: 11, FP: 43, FN: 14  
Precisão: 0.785  
Recall: 0.918  
F1-Score: 0.846
```



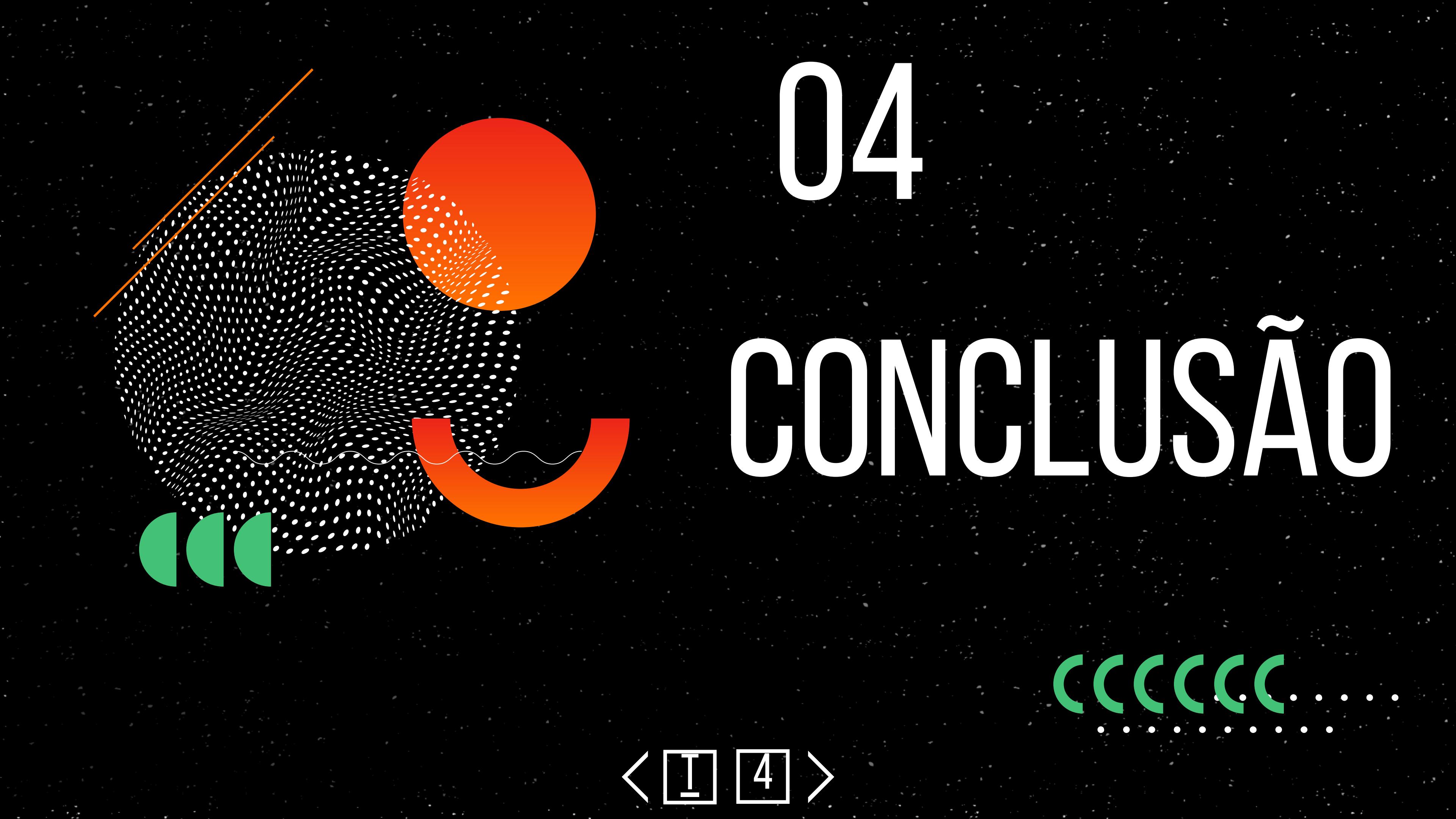
# DATA-REUPLOADING CLASSIFIER



Análise detalhada:

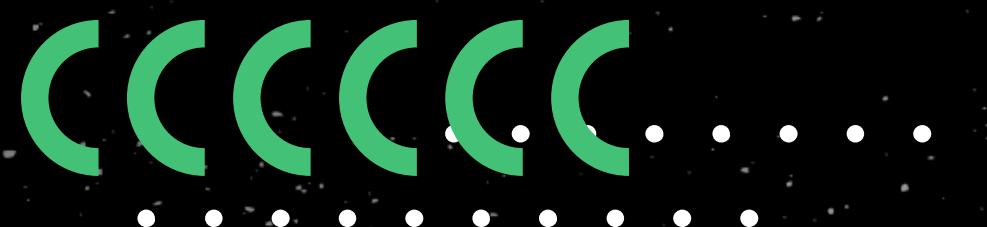
Distribuição das classes verdadeiras: Counter({tensor(1, requires\_grad=True): 171, tensor(0, requires\_grad=True): 54})

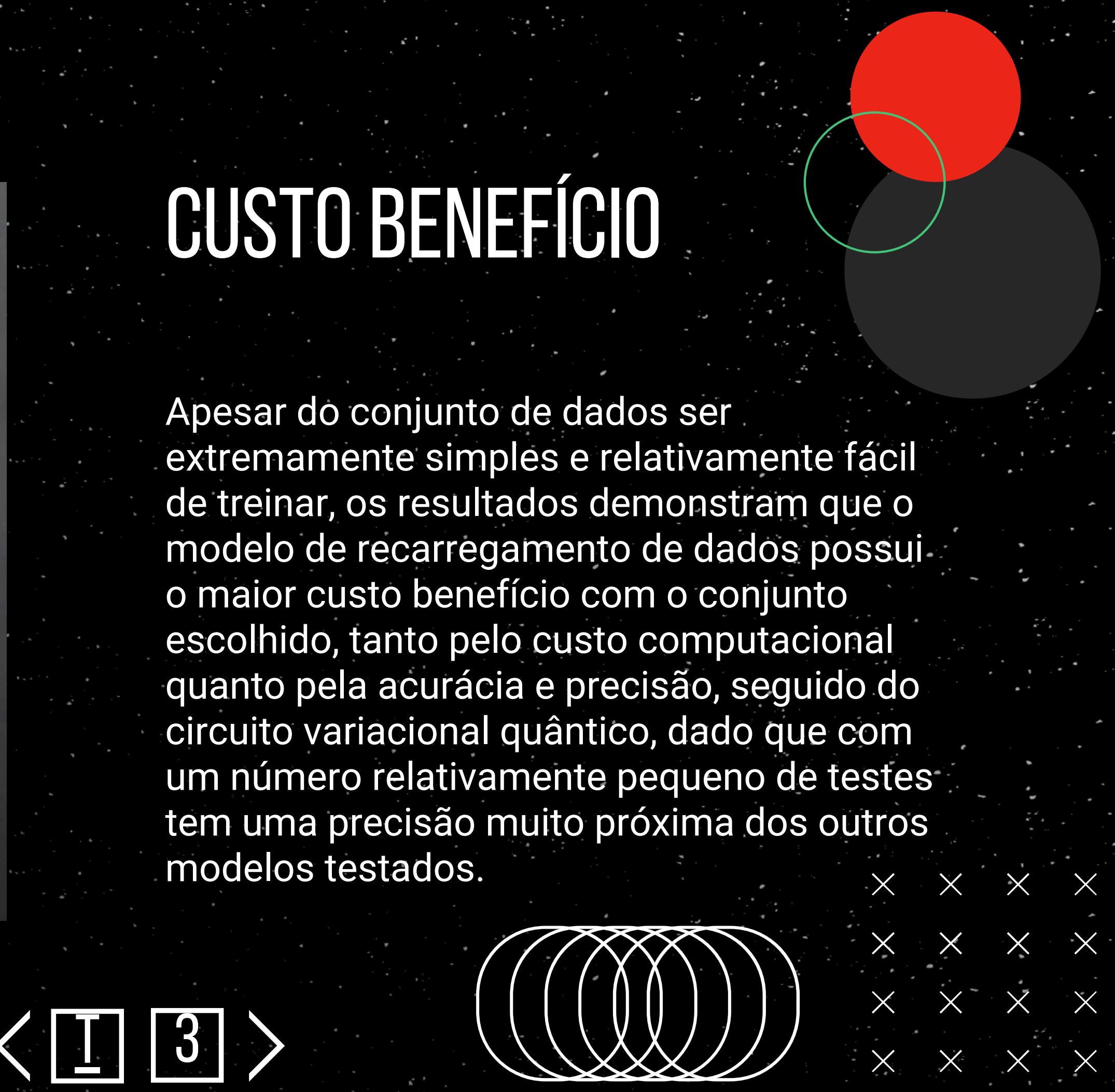
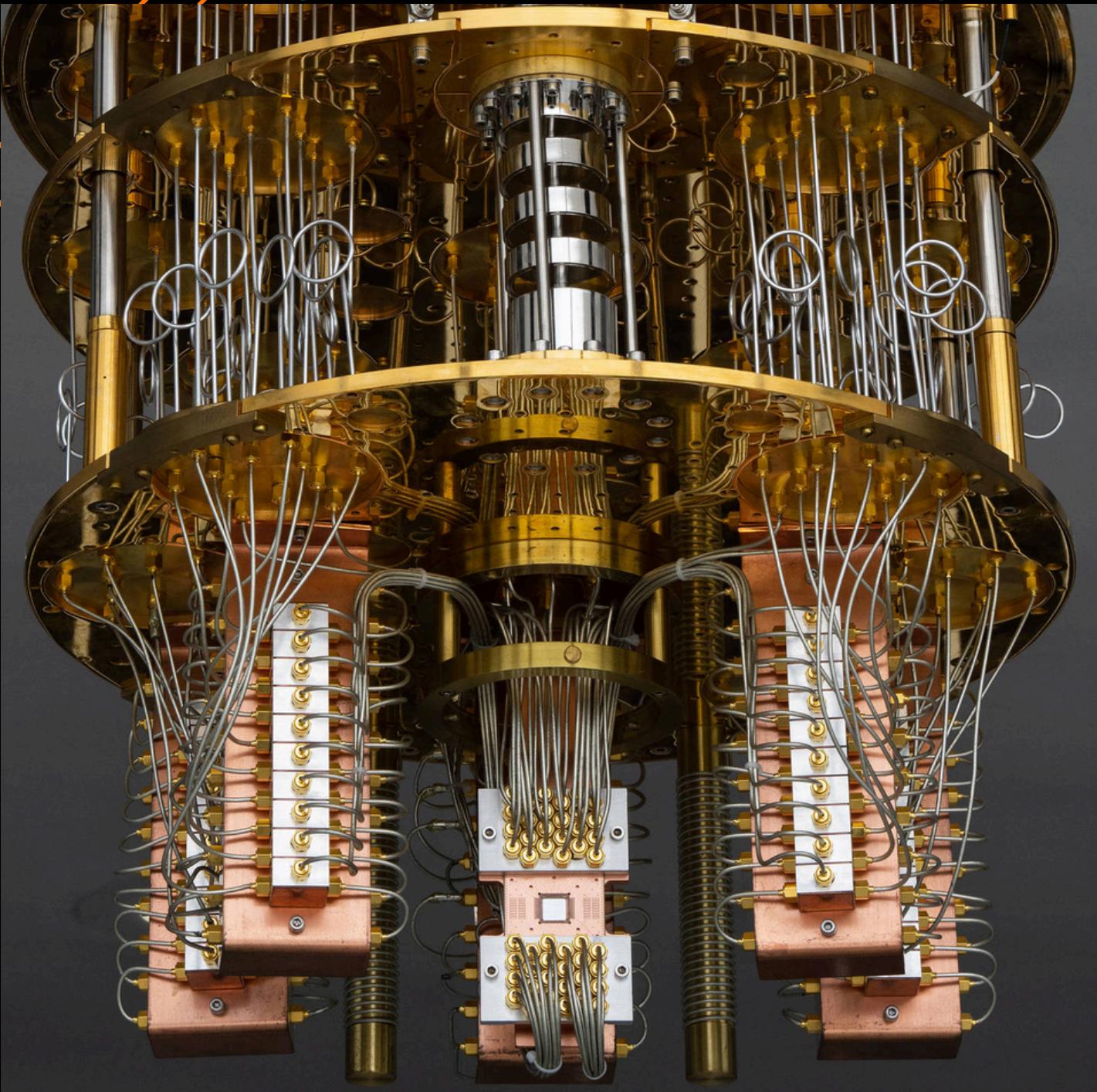
Distribuição das predições: Counter({tensor(1, requires grad=True): 200, tensor(0, requires grad=True): 25})



04

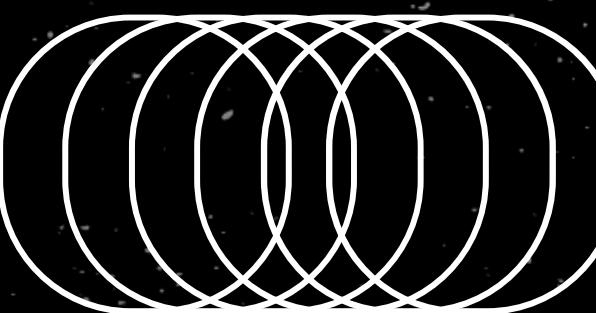
# CONCLUSÃO





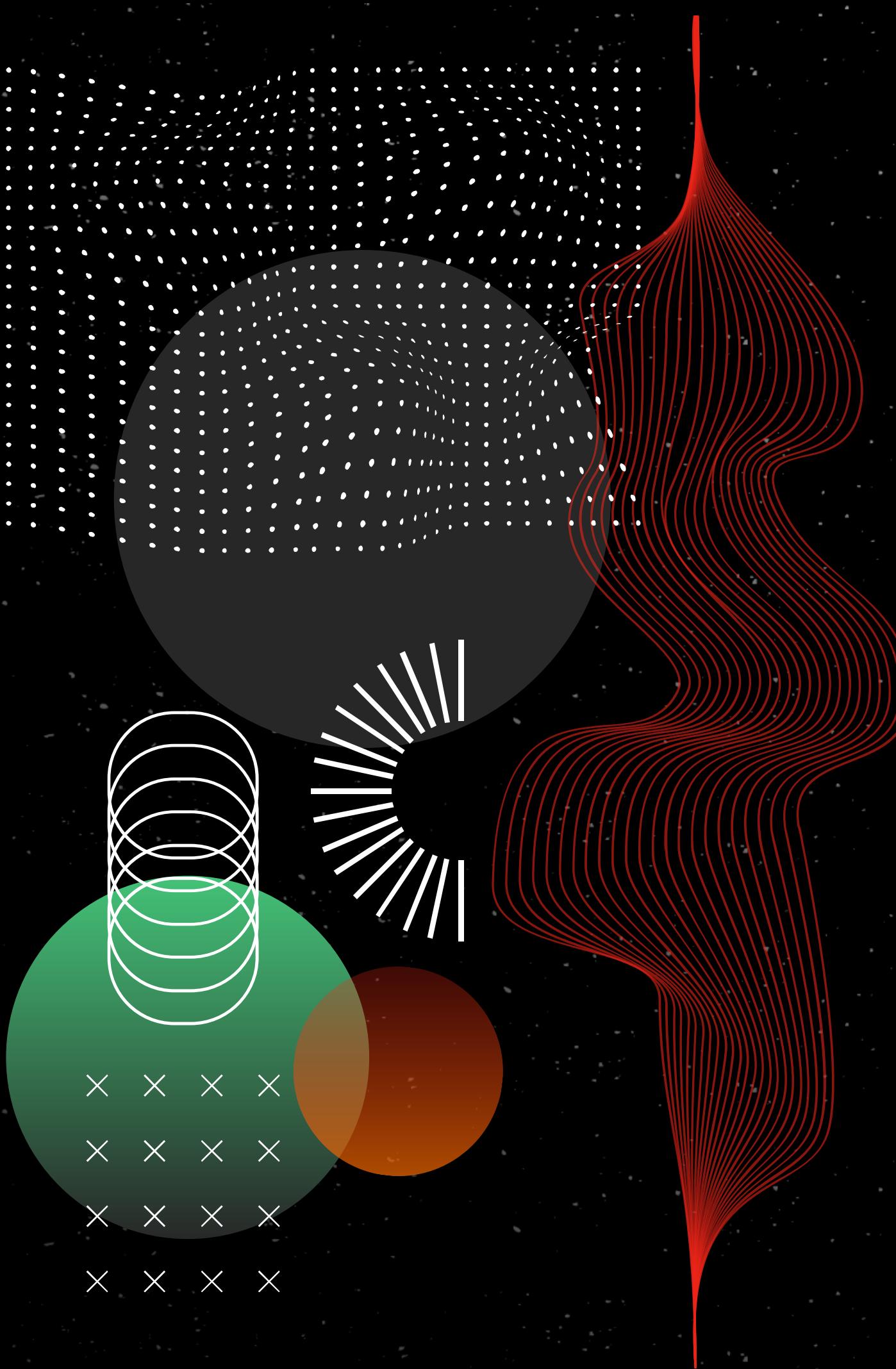
# CUSTO BENEFÍCIO

Apesar do conjunto de dados ser extremamente simples e relativamente fácil de treinar, os resultados demonstram que o modelo de recarregamento de dados possui o maior custo benefício com o conjunto escolhido, tanto pelo custo computacional quanto pela acurácia e precisão, seguido do circuito variacional quântico, dado que com um número relativamente pequeno de testes tem uma precisão muito próxima dos outros modelos testados.



# OBRIGADO!

Dúvidas, críticas, xingamentos ou questionamentos?



$$\left( \frac{-\hbar^2}{2m} \nabla^2 + V \right) \psi = i\hbar \frac{\partial \psi}{\partial t}$$

$$\Delta x_i \Delta p_i \geq \frac{\hbar}{2}$$

