# SAST vs IA

## Detecção de Vulnerabilidades em Aplicações Python

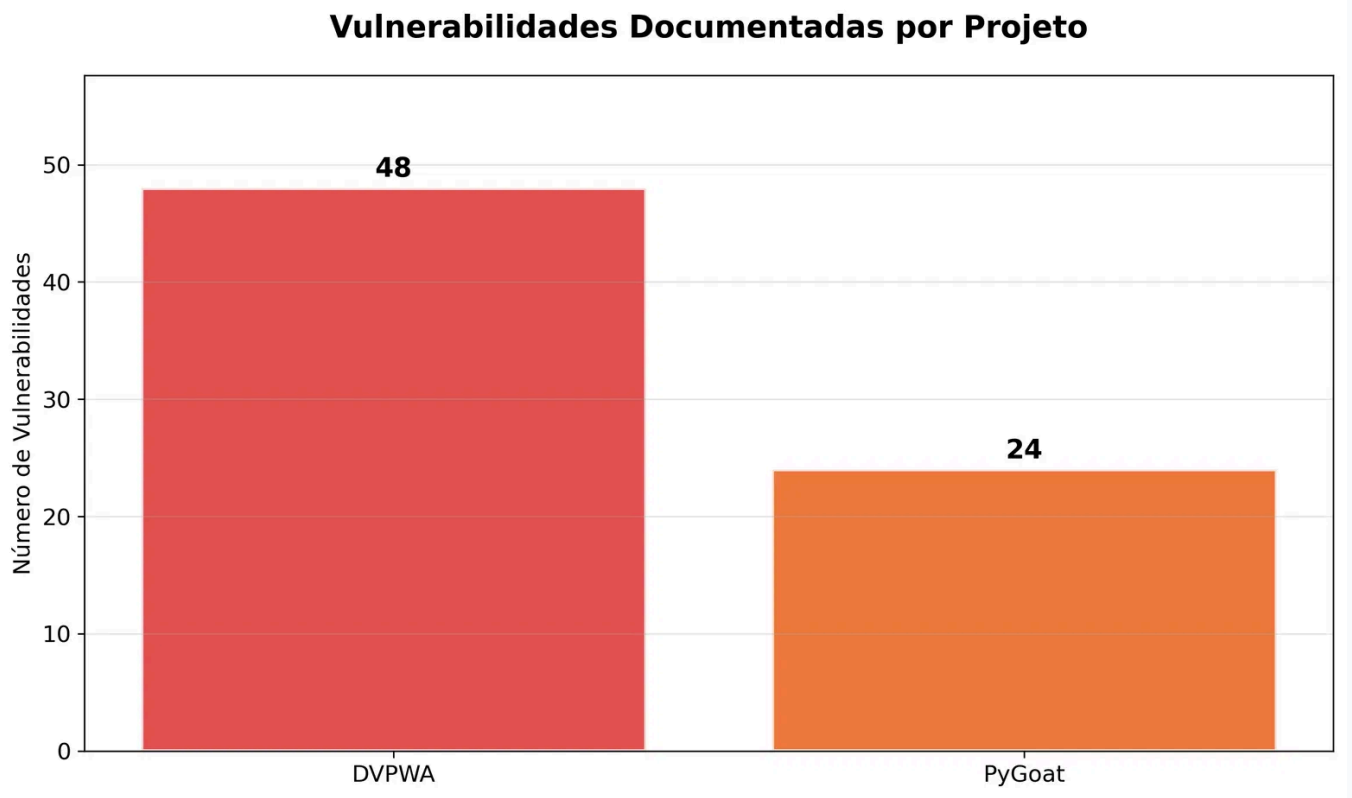
10 de outubro de 2025

# O Desafio da Detecção de Vulnerabilidades

Aplicações web modernas enfrentam ameaças constantes, mas as ferramentas de detecção são suficientes?

- **48 vulnerabilidades** documentadas no projeto DVPWA
- **24 vulnerabilidades** documentadas no projeto PyGoat
- Necessidade de avaliar a eficácia das ferramentas automatizadas
- Comparação entre abordagens tradicionais (SAST) e modelos de IA

*A questão central: Qual abordagem oferece melhor cobertura com menor taxa de falsos positivos?*



# Como Avaliamos: Métricas e Abordagem

## Ferramentas SAST:

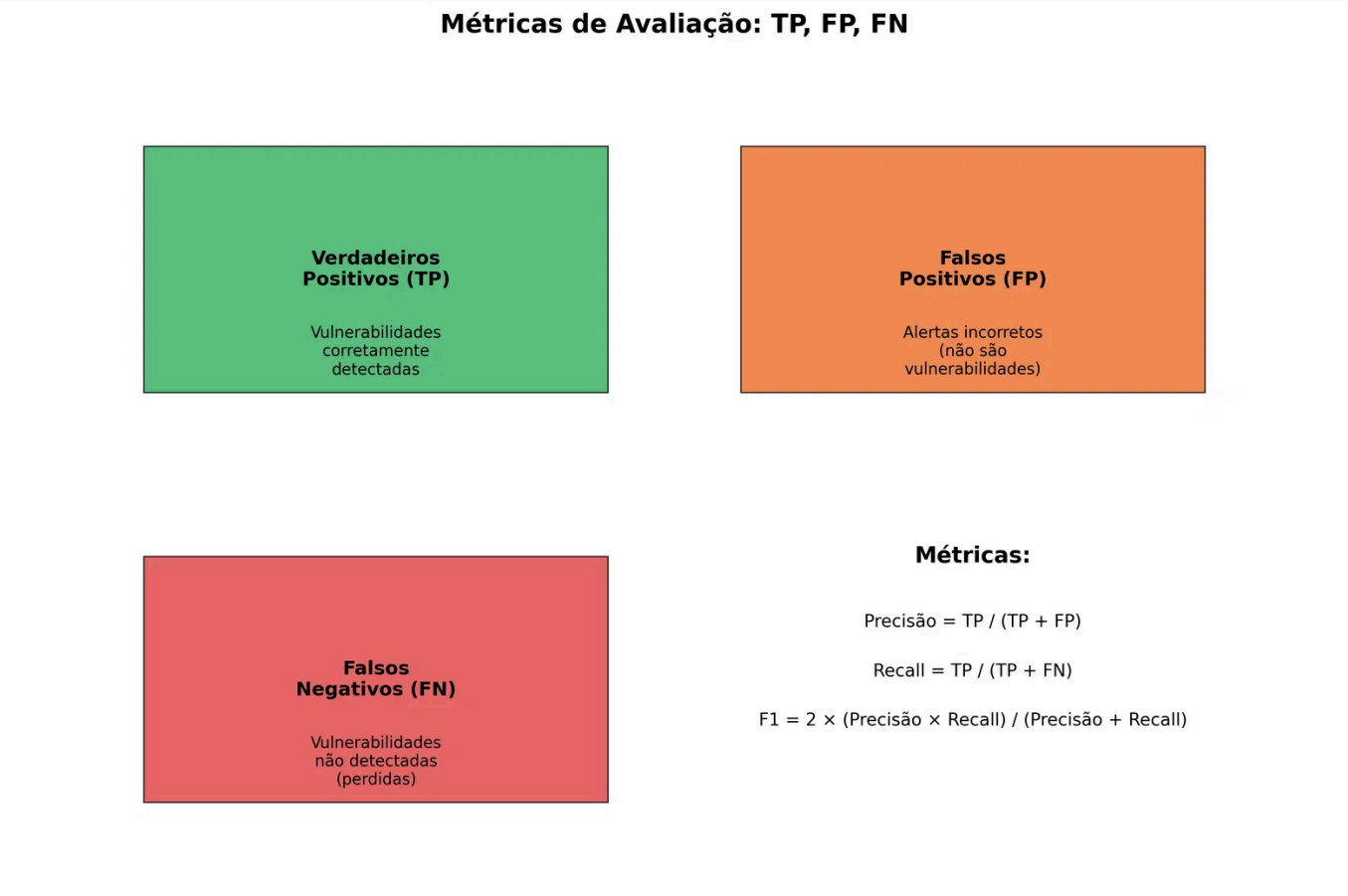
Bandit | Semgrep | Semgrep Pro

## Modelos de IA:

Gemini 2.5 Pro | Claude Sonnet 4.5 | GPT-5 | Grok-4

Avaliação baseada em:

- Groundtruths curados (48 e 24 vulnerabilidades)
- Correspondência de localização (arquivo:linha)
- Validação CWE quando disponível
- Métricas: Precisão, Recall e F1 Score



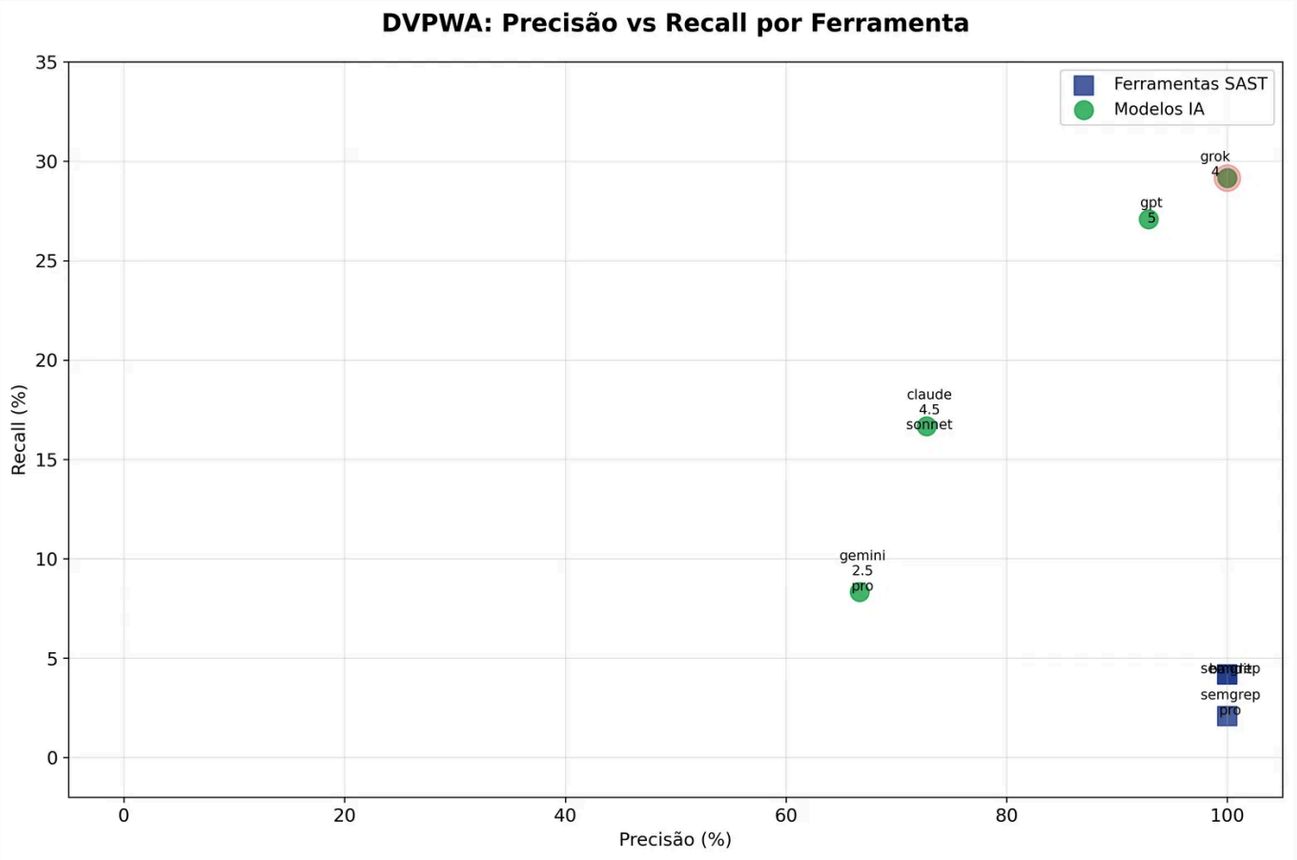
# DVPWA: Ferramentas SAST vs Modelos IA

O projeto DVPWA representa o **melhor cenário** para ferramentas automatizadas:

| Ferramenta | Precisão | Recall |
|---|---|---|
| SAST (média) | 100% | 2-4% |
| Gemini 2.5 Pro | 66,67% | 8,33% |
| Claude 4.5 | 72,73% | 16,67% |
| GPT-5 | 92,86% | 27,08% |
| **Grok-4** | 100% | 29,17% |

**Principais observações:**

- SAST: precisão perfeita, mas recall crítico
- Grok-4: melhor desempenho geral (7x superior ao SAST)
- Mesmo o melhor modelo deixou 71% das vulnerabilidades sem detecção

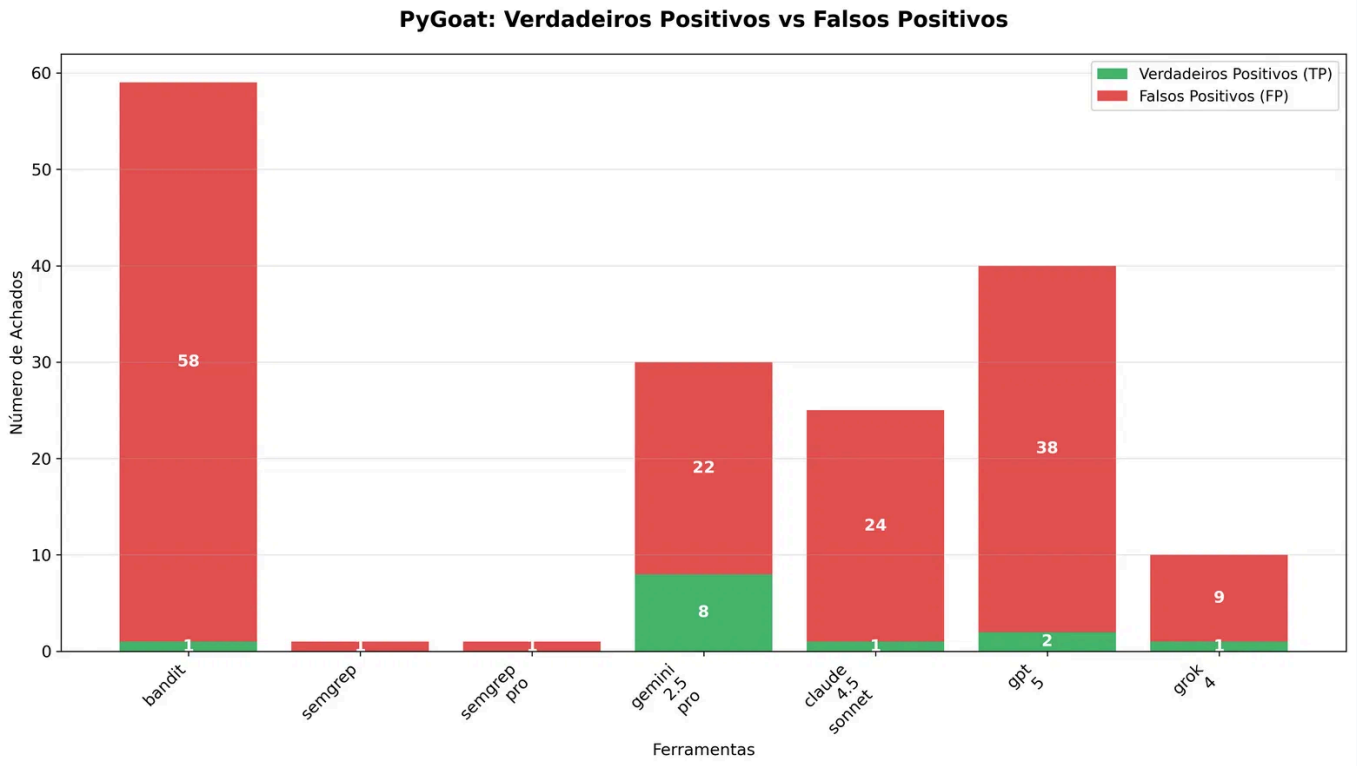


# PyGoat: Quando as Ferramentas Falham

> **Colapso generalizado de precisão**: Maioria das ferramentas com precisão <10% (vs 67-100% em DVPWA)

- **Bandit**: 58 falsos positivos em 59 achados — taxa de ruído inaceitável
- **Gemini 2.5 Pro**: Melhor performer (26,67% precisão, 33,33% recall)
- **Desempenho inconsistente** entre projetos evidencia forte dependência contextual

*O cenário PyGoat demonstra que mesmo as melhores ferramentas podem falhar drasticamente em contextos específicos.*



# O Trade-off Fundamental

## SAST: Abordagem Conservadora

Reporta apenas achados de altíssima confiança

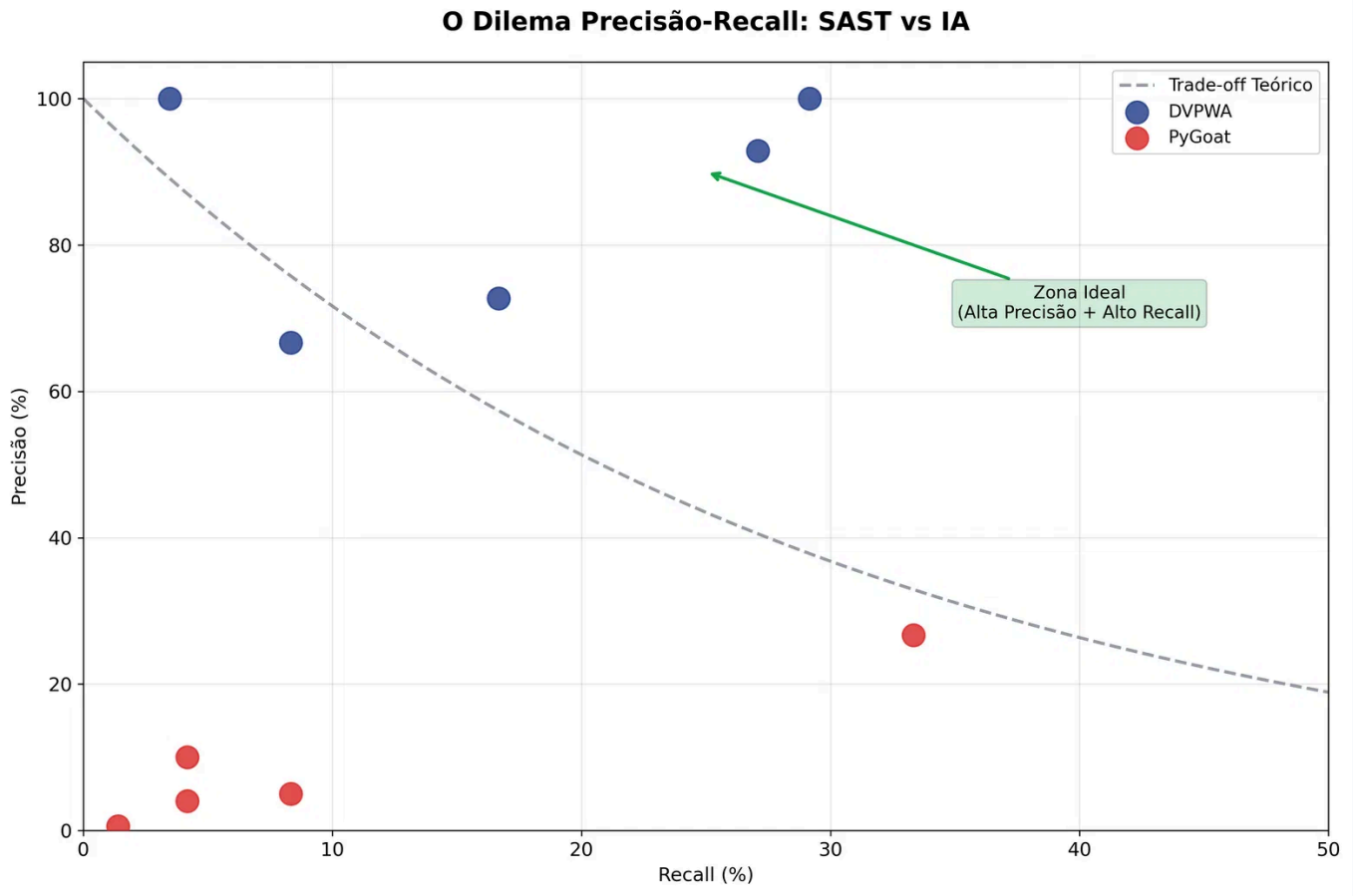
- Precisão perfeita (100%)
- Recall crítico (2-4%)
- Desenvolvedores confiam nos alertas

## IA: Abordagem Semântica

Capacidade de entender contexto e intenção

- Precisão variável (27-100%)
- Recall 4-7x superior
- Risco de fadiga de alertas

*"Baixo recall é mais perigoso que baixa precisão: vulnerabilidades não detectadas vão para produção."*




# Por que Baixo Recall é Mais Perigoso

## Falso Senso de Segurança

Vulnerabilidades não detectadas vão para produção enquanto equipes acreditam que o código é seguro.

## Teatro de Compliance

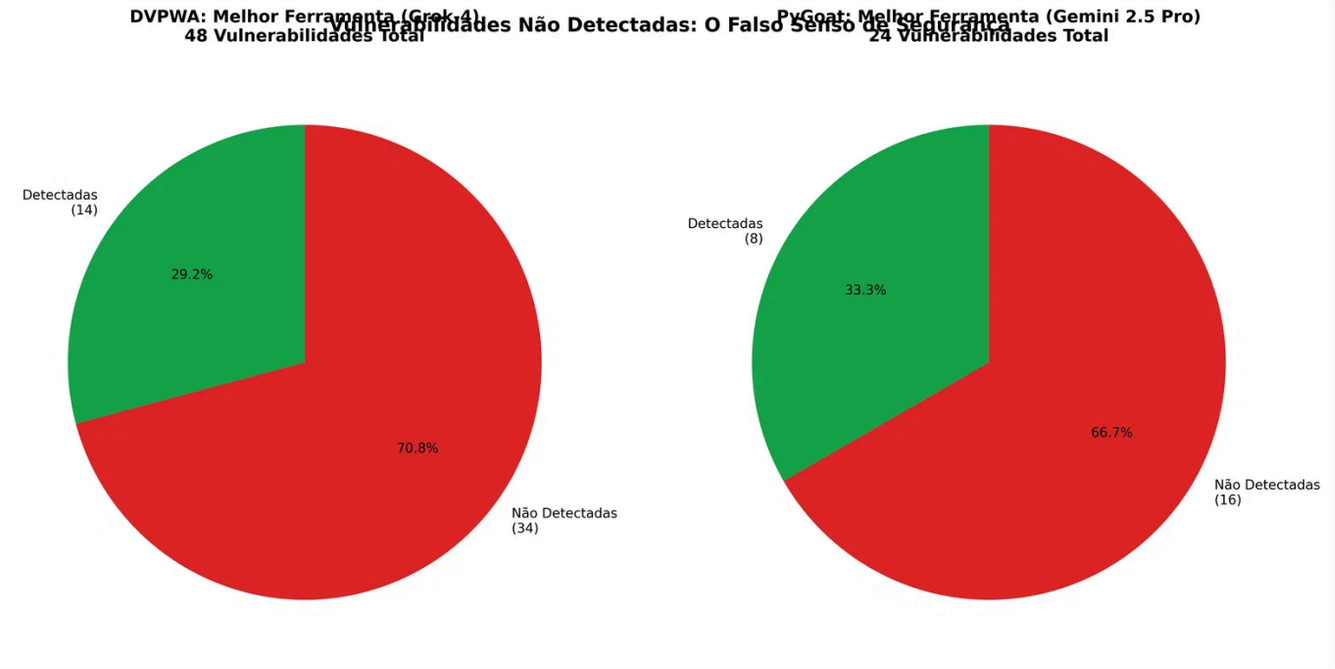
Organizações marcam "executamos ferramentas de segurança" sem alcançar segurança real.

## Recursos Mal Alocados

Equipes focam em corrigir 2-14 problemas detectados enquanto 34-46 outros permanecem.

## Confiança Injustificada

Stakeholders acreditam na segurança porque "as ferramentas não encontraram problemas".



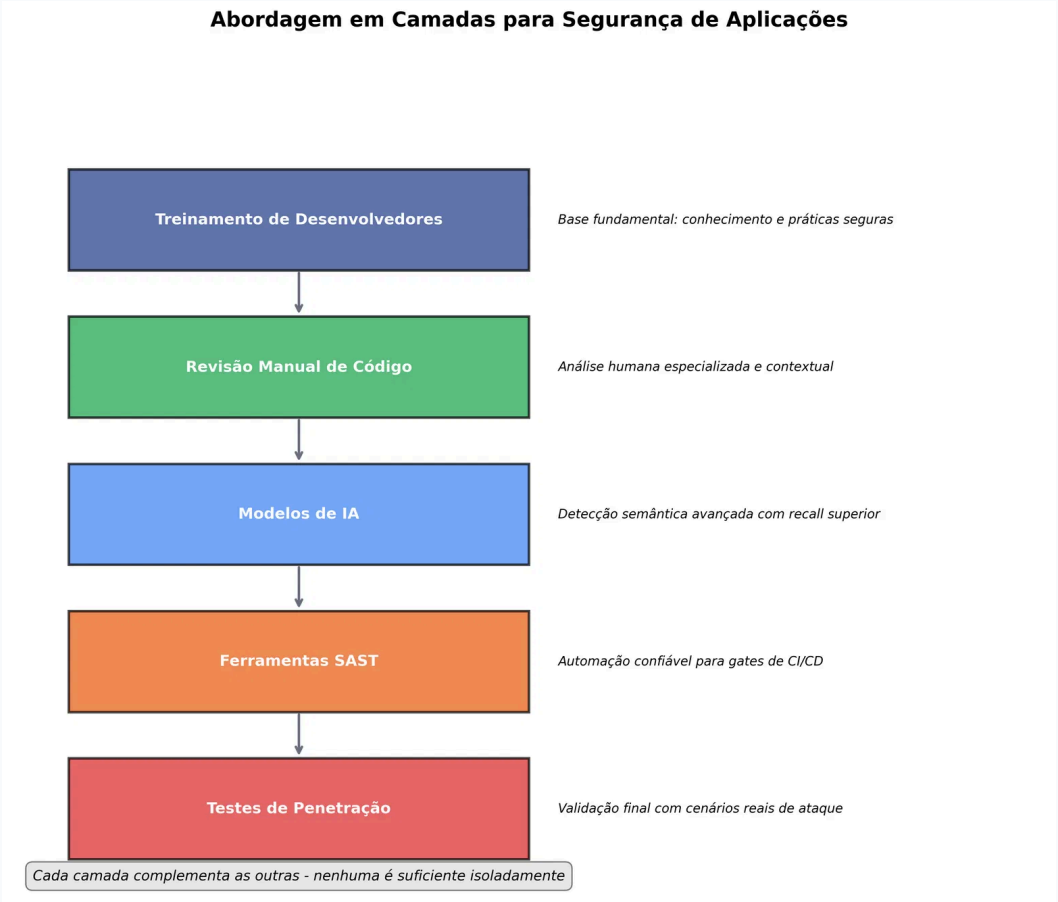
# Caminho para Segurança Efetiva

## Para Equipes de Segurança:

- Implementar abordagem em camadas: SAST + IA + revisão manual
- Medir e reportar recall, não apenas "achados remediados"
- Estabelecer expectativas realistas sobre capacidades

## Para Equipes de Desenvolvimento:

- Não ignorar achados de IA reflexivamente (>90% precisão)
- Fornecer feedback sobre falsos positivos
- Participar ativamente de revisões de segurança



# Conclusões e Próximos Passos