# Projeto Shamann – Implantação Técnica Completa

**Objetivo:**

Desenvolver um assistente modular de pentest altamente eficiente em ambientes com recursos limitados, utilizando IA leve baseada em regras e com decisões sempre validadas por humanos.

## Fases de Implantação

**FASE 1 – Configuração de Ambiente e Base Estrutural**

- Sistema Operacional: Kali Linux 2025.1 ou posterior (64-bit)  
- Ambiente Virtual Python: python3.12 -m venv venv  
- Bibliotecas Base: rich, typer, sqlite-utils, pandas  
- Estrutura de Pastas:  
 shamann/  
 ├── core/  
 ├── modules/  
 ├── ai\_oracle/  
 ├── db/  
 ├── logs/  
 ├── output/  
 ├── shamann.py  
 ├── requirements.txt

**FASE 2 – Implementação Modular**

- Guardiões: scripts autônomos para ferramentas como nmap, whois, netcat, etc.  
- Mestre: orquestrador entre os módulos, logs e controle de execução.  
- Oráculo: IA leve baseada em regras SQLite, sem ações autônomas.

**FASE 3 – Integração da Lógica Semântica (IA Leve)**

- Banco de Conhecimento: SQLite com tabelas de vulnerabilidades, regras e ações.  
- Motor de Inferência: baseado em lógica condicional simples (ex: IF porta 21 aberta THEN risco FTP).

**FASE 4 – Interface e Usabilidade**

- CLI Interativa com Typer  
- Saída formatada com Rich  
- Exportação em JSON, TXT e relatório final em Markdown.

**FASE 5 – Expansão e Atualizações**

- Novos módulos Guardiões facilmente adicionáveis.  
- Atualizações seguras com rollback local.  
- Shamann: módulo de defesa passiva e manutenção de integridade.

## Controle de Versões e Compatibilidade

- Python 3.12  
- SQLite 3.35+  
- Typer 0.9.0  
- Rich 13.7.0  
- Kali Linux 2025.1

## Visão Final

Este projeto é ideal para:  
- Infraestruturas antigas ou com hardware limitado.  
- Operações em ambientes isolados.  
- Educação e treinamento em segurança da informação.  
- Profissionais que valorizam controle manual com assistência inteligente.