OFICINA DO BIRUTIA



Engenharia reversa com IA (só em sistemas seus)

O REA conecta agentes de IA a ferramentas que analisam programas, sites e aplicativos. Veja o que ele faz e, principalmente, onde o uso é permitido.

Engenharia reversa é desmontar um produto pronto para entender como ele funciona, como um mecânico que abre um motor sem o manual. O projeto REA (Reverse Engineer Anything, repositório morluto/rea, licença MIT, ativo no GitHub) conecta agentes de IA a ferramentas de análise de programas, aplicativos, sites e outros alvos. Segundo a referência, cada conclusão vem com prova e com o que ficou sem confirmar. Este guia é uma apresentação; o uso é permitido apenas em software seu ou que você esteja autorizado a examinar.

Quando é permitido

O próprio projeto afirma servir para pesquisa legal, educação, interoperabilidade, preservação de software e análise de software que você tem direito ou autorização de examinar. Exemplos:

Auditar o seu próprio aplicativo (o que entrou no pacote, se há chaves expostas).

Documentar um sistema antigo da sua empresa, com autorização.

Estudar como a sua própria página se comporta.

Quando não é permitido

Quebrar licenças, usar chaves roubadas ou compartilhadas, burlar DRM ou proteção, copiar código protegido de outro programa e invadir sistemas sem autorização. A lei do seu país e a licença do software podem limitar a engenharia reversa. Ter o arquivo no computador não dá permissão para tudo. Ter um relatório ou pseudocódigo não dá direitos sobre o programa: a sua versão deve ter código novo, seu.

Passo a passo resumido

1. Leia o README e o aviso legal do projeto.
2. Instale conforme a documentação (exige Node.js recente e, para programas compilados, uma ferramenta de análise com licença regular; há uma opção gratuita).
3. Rode somente em alvos seus ou autorizados.
4. Analise programas desconhecidos em ambiente isolado (máquina virtual).

Pedidos de exemplo (somente sistemas seus)

Pedido: Auditar o meu aplicativo

> Analise o pacote do meu próprio aplicativo em [caminho]. Quero saber quais módulos e dependências entraram, se há algum arquivo, chave ou texto que não deveria estar ali e o que pesa mais. Liste cada achado com a prova e o que corrigir primeiro. O aplicativo é meu.

Pedido: Documentar sistema legado

> Temos um sistema antigo da empresa em [caminho], sem o código-fonte, e tenho autorização da empresa para analisá-lo. Mapeie as funções principais, os arquivos e bancos que ele lê e grava e as dependências. Monte uma documentação em Markdown com uma seção por módulo, a prova de cada ponto e a lista do que ainda é desconhecido.

Avisos

Decompilação e análise com IA podem vir incompletas ou erradas; confira a prova.

Análises de execução abrem o programa de verdade: use ambiente isolado.

A responsabilidade legal é de quem usa.

Checklist final

[ ] O alvo é meu ou tenho autorização por escrito
[ ] Li o aviso legal do projeto
[ ] Uso ferramentas com licença regular
[ ] Rodei em ambiente isolado

Quem fez

REA, de morluto (github.com/morluto/rea). Curadoria da Oficina do BirutIA, com apoio do material de Guilherme Morais (@guilhermemorais.ia), consultado como referência.

Oficina do BirutIA - material gratuito. Conferido em 28/10/2026. As ferramentas de IA mudam com frequência.

Quer automações para a sua empresa?
O BirutIA e a MLM Produções podem ajudar.
www.mlmproducoes.com.br