

# Referência Rápida do OpenAI Codex CLI

Uso do CLI, modos, configuração, comandos principais

## Início Rápido

### Instalar

```
npm install -g @openai/codex
```

### Autenticação

```
export OPENAI_API_KEY="sk-..."
```

Defina no perfil do shell ou no arquivo .env

### Primeiro Uso

```
codex "explain this project"
codex "add input validation to app.py"
```

## Comandos

### Interativo (Padrão)

```
codex # start interactive
session
codex "fix the login bug" # with initial prompt
```

### codex exec

```
codex exec "write unit tests for utils.py"
codex exec "refactor db.py to use async"
```

Não-interativo — executa a tarefa até o fim

### codex review

```
codex review # review staged
changes
codex review --diff HEAD~3 # review last 3
commits
```

## Modos

### Modos de Aprovação

<b>suggest</b>	Mostrar mudanças propostas, requerer aprovação para cada edição e comando
<b>auto-edit</b>	Aplicar edições automaticamente, requerer aprovação para comandos shell
<b>full-auto</b>	Aplicar edições e executar comandos sem aprovação

### Definir o Modo

```
codex --approval-mode suggest "add tests"
codex --approval-mode auto-edit "refactor"
codex --approval-mode full-auto "fix lint"
```

## Configuração

### Arquivo de Config

```
# ~/.codex/config.yaml
model: o4-mini
approval_mode: suggest
providers:
  - name: openai
    api_key_env: OPENAI_API_KEY
```

Overrides por projeto: .codex/config.yaml na raiz do projeto

### Instruções do Projeto

```
# AGENTS.md (in project root)
- Run tests with: uv run pytest
- Use ruff for linting
- Never modify migration files
```

## Opções de Config Úteis

<b>model</b>	Modelo a usar (ex.: <b>o4-mini</b> , <b>o3</b> )
<b>approval_mode</b>	Modo de aprovação padrão
<b>providers</b>	Config do provedor API (chave, URL base)
<b>history</b>	Salvar histórico da conversa: <b>true</b> / <b>false</b>

## Sandboxing

### Como Funciona

O Codex executa comandos em ambiente sandbox para evitar efeitos colaterais indesejados. Acesso à rede desabilitado por padrão. Escritas de arquivo restritas ao diretório do projeto.

## Opções de Sandbox

<b>macOS</b>	Apple Seatbelt (padrão, embutido)
<b>Linux</b>	Sandbox baseado em Docker
<b>--full-auto</b>	Exige sandbox habilitado
<b>--dangerously-auto-approve</b>	Ignorar sandbox (não recomendado)

## Dicas

### Prompts Eficazes

<b>Seja específico</b>	"add retry logic to fetch_data()" > "improve code"
<b>Referencie arquivos</b>	"fix the bug in src/auth.py" delimita o escopo
<b>Declare restrições</b>	"don't change the public API" define limites
<b>Itere</b>	Continue refinando na mesma sessão

### Padrões de Workflow

```
# Explore → plan → execute
codex "explain the auth module"
codex "plan how to add OAuth support"
codex --approval-mode auto-edit "add OAuth"
```

```
# Review before commit
git add -A
codex review
```

### Flags Comuns

<b>--model, -m</b>	Override do modelo para esta sessão
<b>--approval-mode</b>	Definir modo de aprovação
<b>--quiet, -q</b>	Saída mínima
<b>--no-project-doc</b>	Ignorar AGENTS.md