EVO

ENHANCED VIRTUAL OPERATOR

Ivo Faria

1 DE FEVEREIRO DE 2026

IVO FARIA

1. Visão e Objetivos

O EVO (Enhanced Virtual Operator) é um operador virtual local, inspirado em assistentes de ficção científica, mas concebido com princípios realistas, conservadores e defensáveis do ponto de vista técnico e ético. Não visa omnisciência: prioriza previsibilidade, privacidade e controlo explícito. A sua presença deve parecer humana e discreta: fala quando chamado, confirma antes de agir e mantém um HUD minimalista para feedback visual.

Objetivos de alto nível:

- Reduzir fricção homem–máquina no dia a dia.

- Centralizar e padronizar ações recorrentes em Windows.

- Estabelecer uma base modular robusta para expansão futura.

- Fornecer um projeto técnico demonstrável e auditável.

2. Requisitos

Funcionais:

- Ativar por wake word dedicado (sensibilidade configurável) com VAD local para detetar fala.

- Transcrever (STT, Whisper local) e sintetizar (TTS, Piper/SAPI) com latência baixa.

- Interpretar intenções por regras determinísticas (regex/ontologia), com fallback LLM local quando necessário.

- Executar ações do sistema (bloquear, suspender, hibernar) mediante confirmação explícita.

- Fornecer HUD/overlay PySide6 para estados e mensagens curtas.

Não funcionais:

- Privacidade por defeito (processamento local; sem telemetria pessoal).

- Previsibilidade e repetibilidade de comandos.

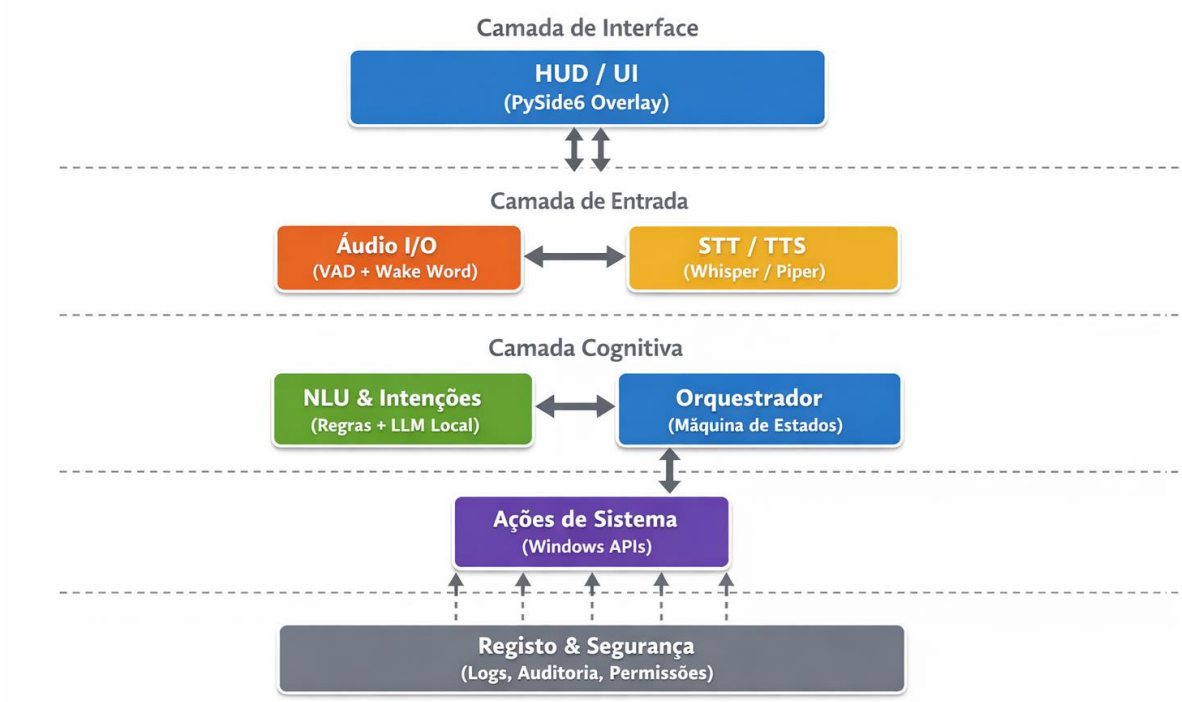
- Eficiência: footprint reduzido de CPU/GPU e gestão de energia.

- Observabilidade: registos de eventos e falhas com rotação.

- Modularidade e interfaces simples entre componentes.

3. Arquitetura de Alto Nível

Arquitetura modular e síncrona com orquestrador central. Os módulos principais são: HUD/UI, Áudio I/O, STT/TTS, NLU/Intenções, Orquestrador e Ações de Sistema, apoiados por Registo/Segurança. As fronteiras entre módulos são estáveis e baseadas em interfaces simples (contratos).



4. Componentes e Interfaces

HUD/UI (PySide6)

- Overlay transparente, sempre no topo, não interfere com janelas.

- Estados (listening/thinking/speaking) e mensagens curtas adaptativas (contraste).

- Entrada de texto opcional para fallback.

Áudio I/O

- Captura do microfone com supressão básica de ruído.

- VAD local (janela pequena) para delimitar fala.

- Wake word dedicado com cooldown e threshold ajustáveis.

STT

- Whisper local (modelo ajustado ao hardware).

- Normalização do texto (pontuação, capitalização leve).

TTS

- Piper/SAPI com voz neutra; pausas naturais e taxa configurável.

- Pré-carregamento de voz para reduzir latência.

NLU/Intenções

- Regras determinísticas (regex/slots/ontologia).

- Fallback LLM local para inputs fora do domínio.

- Saída: {intencao, parametros, confianca}.

Orquestrador

- Máquina de estados com janela temporal de conversa e timeouts.

- Política Pedido→Confirmação→Execução para ações críticas.

Ações de Sistema

- APIs Windows (bloquear/suspender/hibernar).

- Princípio do menor privilégio.

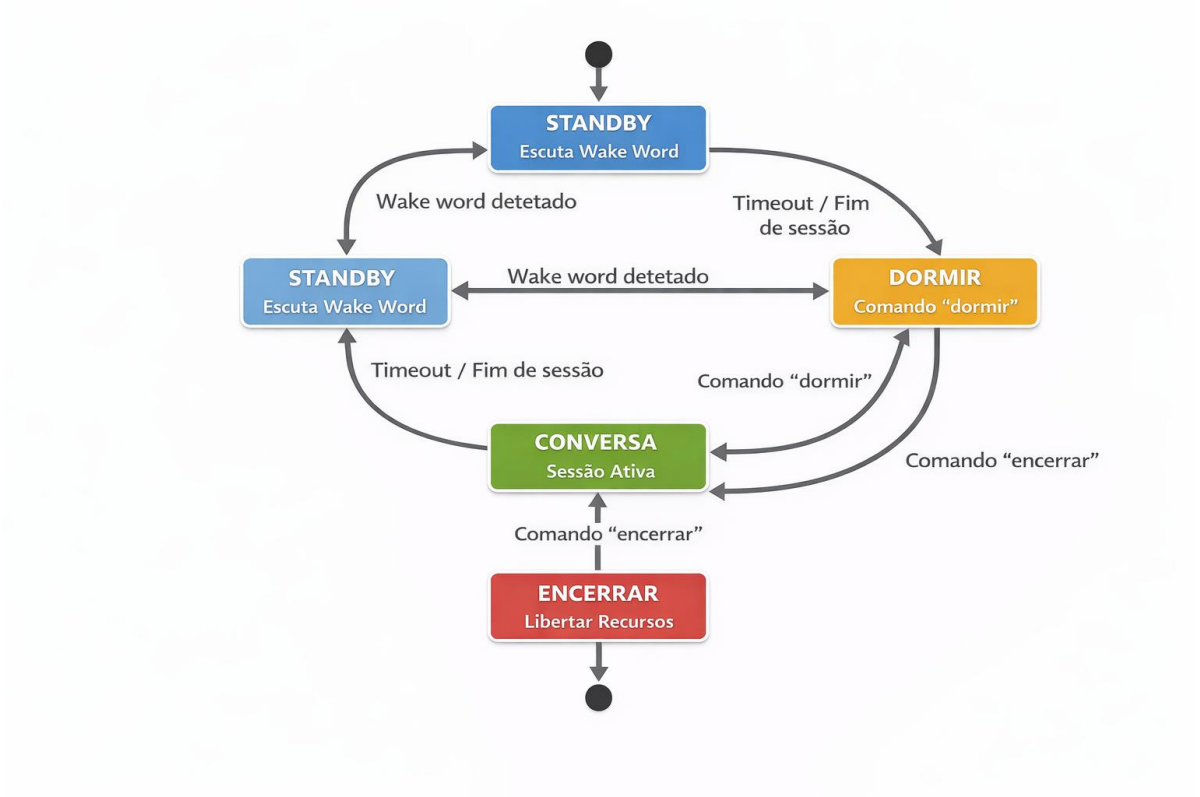
Registo & Segurança

- Eventos críticos e erros, sem dados pessoais.

- Fail-safe por defeito.

5. Estados e Fluxos

Estados principais: STANDBY (apenas wake), CONVERSA (janela ativa), DORMIR (sem escuta), ENCERRAR (libertação de recursos). Transições por wake, timeouts e intenções explícitas.



6. Pipeline de Áudio e Linguagem

Sequência: Microfone → VAD → wake word → STT → NLU → Orquestrador → TTS → Altifalante. Cada etapa é substituível e configurável.

7. Experiência de Utilizador (UX) e HUD

- Comunicação natural e concisa; sem textos longos por defeito.

- Feedback visual mínimo (estados, legendas curtas).

- Tipografia Segoe UI; contraste dinâmico com sombra/contorno.

- Pausas e confirmações verbais claras; repetição apenas quando pedida.

8. Pseudocódigo — Orquestração Global

```
INICIAR:
  carregar_config()
  iniciar_logger()
  iniciar_HUD()
  iniciar_audio()
  estado ← STANDBY

LOOP PRINCIPAL:
  enquanto ativo:
    se estado == STANDBY:
      se wake_word_detectado():
        abrir_contexto_conversa()
        estado ← CONVERSA
        reiniciar_temporizador()

    se estado == CONVERSA:
      se fala_disponivel() e VAD_ativo():
        texto ← STT_transcrever()
        se texto:
          intencao, params, conf ← NLU_interpretar(texto)
          resposta, acao ← DECIDIR(intencao, params)
          TTS_falar(resposta)
          se acao in {HIBERNAR, SUSPENDER, BLOQUEAR}:
            se confirmar(): executar_acao(acao)
```

```
        reiniciar_temporizador()

      se temporizador_expirou(): estado ← STANDBY


    se intencao == DORMIR: desativar_escuta(); estado ← DORMIR

    se estado == DORMIR e wake_word_detectado(): ativar_escuta(); estado ← CONVERSA

    se intencao == ENCERRAR_EVO: libertar_recursos(); terminar()
```

9. Pseudocódigo — Módulos

```
AUDIO:

  iniciar_dispositivo()

  enquanto ativo:

    frame ← ler_buffer()

    if VAD(frame): acumular()

    if wake(frame): sinalizar_wake()


STT:

  audio → whisper.transcrever(lang='pt') → texto

  return normalizar(texto)


NLU:

  if regra_bate(texto): return (INTENCAO, PARAMS, 1.0)

  else: return LLM_local(texto)
```

```
ACOES:
  verificar_permissoes()
  if acao == HIBERNAR: chamar_API_hibernar()
  if acao == BLOQUEAR: chamar_API_bloquear()
  registar(evento)
```

10. Configuração Sugerida (YAML-like)

```
config:
  audio_device: default
  vad_threshold: 0.6
  wakeword:
    model: 'evo.wake'
    sensitivity: 0.55
  stt:
    engine: whisper
    model: small-pt
  tts:
    engine: piper
    voice: 'pt-neutral'
  timeout_conversa_ms: 6000
  confirmacao_requerida: [HIBERNAR, SUSPENDER, BLOQUEAR]
  logs:
    nivel: INFO
    rotacao_MB: 10
    manter_ficheiros: 5
```

11. Plano de Testes e Métricas

- Wake word: medir FP/FN em 60 min (silencioso/ruidoso).

- Latência: wake→resposta (p95) em microfones diferentes.

- WER STT PT-PT com amostras multi-sotaque.

- Pedido→Confirmação→Execução em ações críticas.

- Execução 24h sem crash/mem leaks.

12. Segurança e Políticas

- Princípio do menor privilégio em todas as interações com o SO.

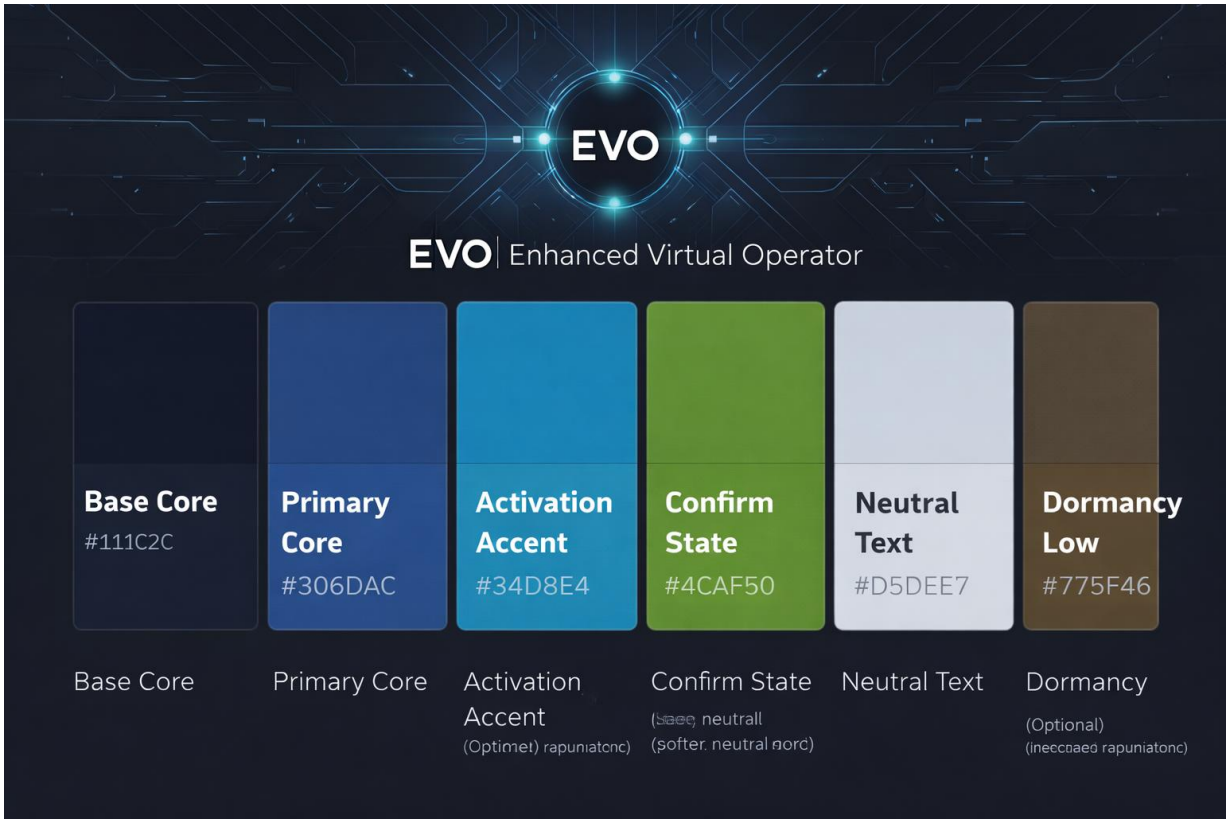
- Sem automação silenciosa; confirmar sempre antes de ações críticas.

- Logs apenas de eventos técnicos; sem dados pessoais.

- Fail-safe: em dúvida, não executar.

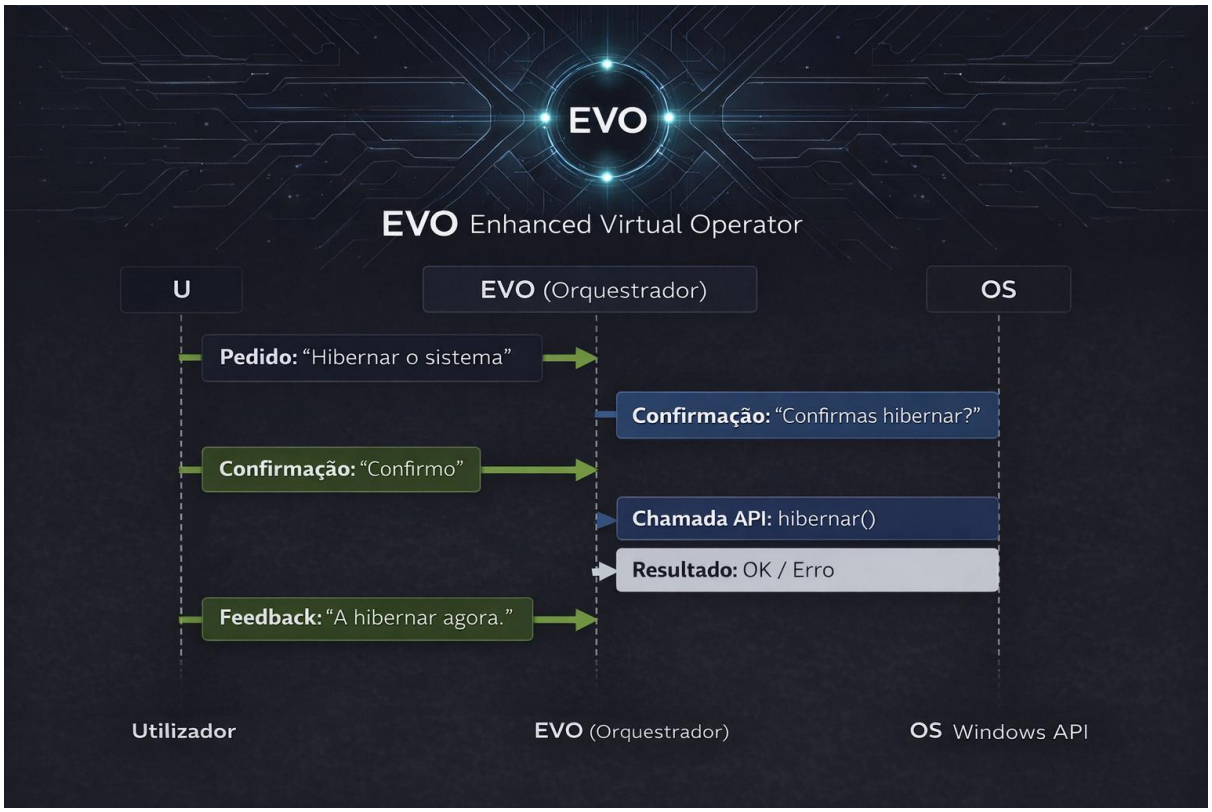
13. Identidade Visual EVO

Tom tecnológico, sério e preciso. Tipografia Segoe UI (Semibold para títulos) e Consolas para código. Usar o logótipo original (PNG) tal como fornecido; evitar distorções, contornos excessivos e fundos ruidosos.



14. Exemplo de Sequência — Ação Crítica

Fluxo Pedido → Confirmação → Execução para “hibernar”.



15. Roadmap de Evolução

- Perfis de voz e persona (voz, formalidade, tempo de resposta).

- Integração com domótica local (Home Assistant) via intents controladas.

- Tools/plug-ins para automações seguras em sandbox.

- Migração opcional para .NET no front-end; núcleo Python.

- Suporte multi-OS (Linux headless) e CLI.

16. Anexo — Intenções e Comandos

- "bloquear sessão" → BLOQUEAR

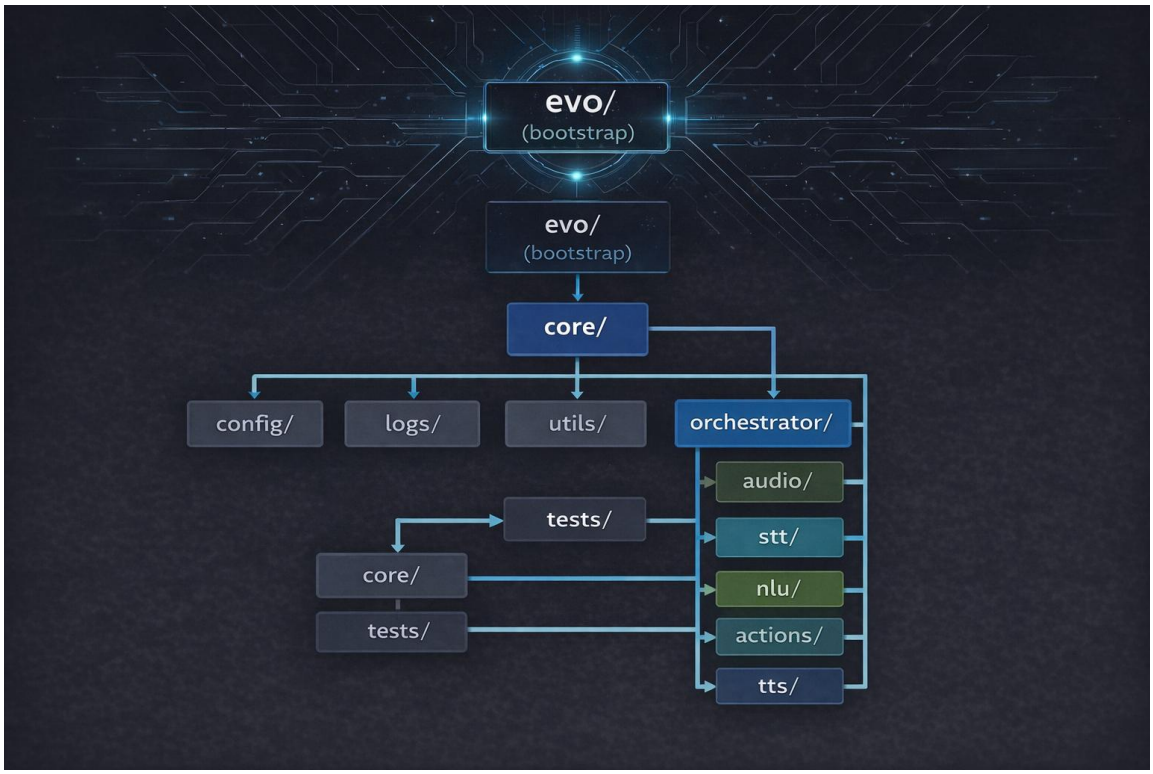
- "podes hibernar o PC" → HIBERNAR (confirmação obrigatória)

- "põe-te a dormir" → DORMIR

- "termina" → ENCERRAR_EVO

17. Arquitetura por Ficheiros e Diretórios

Estrutura sugerida do projeto e responsabilidades por pasta:




```
evo/
├─ core/            # __init__, ciclo de vida da app, carregador de config
├─ audio/           # captura, VAD, wake word
├─ stt/             # wrappers de Whisper
├─ tts/             # wrappers de Piper/SAPI
├─ nlu/             # regras e fallback LLM local
├─ orchestrator/    # máquina de estados, timers
├─ actions/         # chamadas a APIs Windows
├─ ui/              # PySide6 HUD/overlay
├─ config/          # ficheiros .yaml/.json
```

```
├─ logs/            # rotação e formatos
├─ tests/           # unit/integration/e2e
└─ utils/           # helpers partilhados
```

18. Orçamento de Latência (Objetivos por Etapa)

Etapa	Alvo (ms)	Notas
VAD (frame detect)	<= 20 ms	Amostras de 10–20 ms; decisão incremental
Wake word	<= 50 ms	Modelo pequeno; janela deslizante
STT (Whisper small-pt)	150–300 ms	Depende de CPU/GPU e chunk size
NLU (regras)	<= 10 ms	Regex/slots determinísticos
NLU (LLM local fallback)	80–200 ms	Prompt curto e contexto reduzido
Orquestração	<= 5 ms	Enfileirar/decidir
TTS (Piper/SAPI)	120–250 ms	Pré-carregar voz; streaming se possível
Total p95	<= 650–800 ms	Do wake à primeira sílaba sintetizada

19. Esquema Formal de Configuração

YAML recomendado com chaves obrigatórias/opcionais e defaults; JSON Schema opcional para validação automática.

```
config:
```

```
audio:
  device: default           # obrigatório
  sample_rate: 16000        # default
vad:
  threshold: 0.6            # obrigatório (0..1)
  frame_ms: 20              # default
wakeword:
  model: 'evo.wake'         # obrigatório
  sensitivity: 0.55         # default
  cooldown_ms: 800          # default
stt:
  engine: whisper           # obrigatório
  model: small-pt           # obrigatório
  chunk_ms: 800             # opcional
tts:
  engine: piper             # obrigatório
  voice: 'pt-neutral'       # obrigatório
  speed: 1.0                # opcional
nlu:
  mode: 'rules+llm'         # obrigatório
  ruleset: 'pt.yml'         # obrigatório se mode inclui rules
  llm_model: 'local-7b'     # opcional
  max_tokens: 64            # opcional
orchestrator:
  window_ms: 6000           # obrigatório
  idle_timeout_ms: 3000     # opcional
security:
```

```
    confirm_required: [HIBERNAR, SUSPENDER, BLOQUEAR]

  logs:

    level: INFO

    rotation_mb: 10

    keep_files: 5
```

```
{

  "$schema": "https://json-schema.org/draft/2020-12/schema",

  "type": "object",

  "properties": {

    "config": {

      "type": "object",

      "required":
["audio","vad","wakeword","stt","tts","nlu","orchestrator","security","logs"],

      "properties": {

        "audio":
{"type":"object","properties":{"device":{"type":"string"},"sample_rate":{"type
":"integer"}}},

        "vad":
{"type":"object","properties":{"threshold":{"type":"number"},"frame_ms":{"type
":"integer"}}},

        "wakeword":
{"type":"object","properties":{"model":{"type":"string"},"sensitivity":{"type"
:"number"},"cooldown_ms":{"type":"integer"}}},

        "stt":
{"type":"object","properties":{"engine":{"type":"string"},"model":{"type":"str
ing"},"chunk_ms":{"type":"integer"}}},

        "tts":
{"type":"object","properties":{"engine":{"type":"string"},"voice":{"type":"str
ing"},"speed":{"type":"number"}}},
```

"nlu":
{"type":"object","properties":{"mode":{"type":"string"},"ruleset":{"type":"str
ing"},"llm_model":{"type":"string"},"max_tokens":{"type":"integer"}}},

"orchestrator":
{"type":"object","properties":{"window_ms":{"type":"integer"},"idle_timeout_ms
":{"type":"integer"}}},

"security":
{"type":"object","properties":{"confirm_required":{"type":"array","items":{"ty
pe":"string"}}}},

"logs":
{"type":"object","properties":{"level":{"type":"string"},"rotation_mb":{"type"
:"integer"},"keep_files":{"type":"integer"}}}

}

}

}

}

20. Plano de Logs e Observabilidade

Formato de linha (texto):

[YYYY-MM-DD HH:MM:SS.mmm] LEVEL=INFO SRC=orchestrator EVT=intent_decoded INTENCAO=HIBERNAR CONF=0.93

Campos mínimos: timestamp | level | source | evento | detalhes (k=v)

Rotação: max 10 MB, manter 5, compressão opcional (.gz)

Eventos: voice_activity_start | wakeword_detected | stt_ok | stt_fail | intent_decoded | action_confirm_request | action_executed | action_denied

Export: ficheiro local (por defeito); stdout opcional; integração futura com ELK/Seq.

21. Guia de Deploy em Windows

```
# Instalação local (PowerShell)

python -m venv .venv

.venv\Scripts\activate

pip install -r requirements.txt


# Arranque automático (Task Scheduler)

# 1) Task Scheduler > Create Task

# 2) Triggers: At log on

# 3) Actions: Start a program -> Program/script:
C:\...\.venv\Scripts\python.exe

#    Add arguments: -m evo.main

#    Start in: C:\...\projeto

# 4) Conditions: ajustar opções de energia conforme necessidade


# Privilégios: executar com conta padrão; evitar admin
```

22. Plano de QA e Casos de Teste

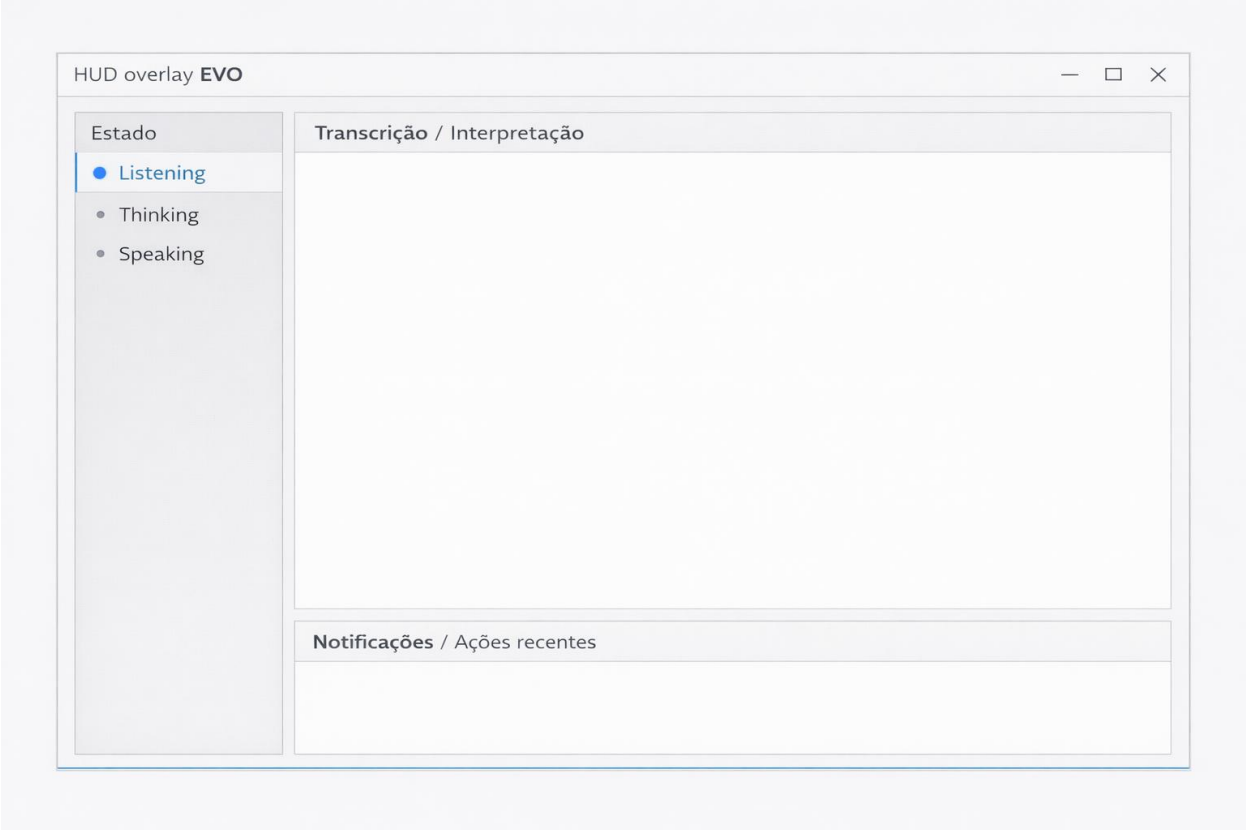
Área	Métrica/Objeto	Cenário	Critério de Aceitação
Wake word	Taxa FP/FN (60 min)	Silêncio / TV / conversas	FP < 1/h; FN aceitável
Latência	Wake→Resposta (p95)	Mic USB/onboard	≤ 800 ms

STT WER	WER PT-PT	Amostras multi-sotaque	≤ meta interna
Ações críticas	Pedido→Confirmação→Execução	Hibernar/Bloquear/Suspender	Nunca sem confirmação
Resiliência	Run 24h	Carga normal/ruído	Sem crash/mem leak

23. Wireframes de UX/HUD

HUD — tema escuro (wireframe):

HUD — tema claro (wireframe):



24. Riscos e Mitigações

- Latência acima do alvo → quantização/otimização de modelos e buffering.

- Falsos positivos de wake → ajustar sensibilidade/cooldown, filtros de ruído.

- WER elevado → dicionário custom, limpeza de ruído, escolha de modelo adequado.

- Falhas em ações críticas → confirmações reforçadas, retries com backoff e cancelamentos seguros.

- Consumo excessivo → profiling de CPU/GPU e gestão de energia.