



1 Agentes de Voz

Cada agente ganha um endereço SIP próprio:
`agt_xxxx@sip.sippulse.ai` — para o Asterisk, é só mais um ramal.

A PJSIP com contact fixo RECOMENDADO

```
; pjsip.conf
[agente-ia]
type=endpoint
context=from-internal
disallow=all
allow=a law, ulaw
aors=agente-ia
dtmf_mode=rfc4733
```

```
[agente-ia]
type=aor
contact=sip:agt_xxxx@sip.sippulse.ai
```

```
; extensions.conf (tT habilita transf.)
exten => 4000,1,Dial(PJSIP/agente-ia,,tT)
```

B SipPulse AI registra no PABX

Ideal atrás de NAT ou sem trunk: crie um ramal comum (endpoint + auth + aor, `max_contacts=1`) e cadastre servidor, usuário e senha no painel SipPulse AI. A plataforma faz o **REGISTER** e o agente atende como ramal registrado: `Dial(PJSIP/4000,,tT)`.

C chan_sip LEGADO · até Asterisk 20

```
; sip.conf (removido no Asterisk 21+)
[sippulse]
type=peer
host=sip.sippulse.ai
fromuser=agt_xxxx
dtmfmode=rfc2833
```

```
; extensions.conf
exten => 4000,1,Dial(SIP/agt_xxxx@sippulse,,tT)
```

2 Variáveis nos cabeçalhos SIP

Personalize cada chamada com dados do seu CRM ou banco antes do Dial. Valores **sempre em Base64** (aceita acentos e quebras de linha).

X-Additional-Instructions

Acrescenta contexto ao prompt do agente **só nesta chamada**: nome do cliente, fatura, status do pedido.

X-Initial-Message

Substitui a saudação inicial do agente por uma frase sua.

X-Uid

Define o ID da sessão, para casar a conversa com seu ticket ou CRM depois.

```
; extensions.conf — busque no CRM e injete
exten => 4000,1,NoOp(${CALLERID(num)})
same => n,Set(INSTR=0 cliente se chama
    ${NOME}, a fatura de R$ ${VALOR}
    vence em ${VENC})
same => n,Set(PJSIP_HEADER(add,
    X-Additional-Instructions)=
    ${BASE64_ENCODE(${INSTR}}))
same => n,Set(SAUD=01á ${NOME},
    aqui é a Ana da Acme)
same => n,Set(PJSIP_HEADER(add,
    X-Initial-Message)=
    ${BASE64_ENCODE(${SAUD}}))
same => n,Set(PJSIP_HEADER(add,X-Uid)=
    ${BASE64_ENCODE(${UNIQUEID}}))
same => n,Dial(PJSIP/agente-ia,,tT)
```

- No `chan_sip` use `SIPAddHeader(X-Uid: ${BASE64_ENCODE(${UNIQUEID}})`
- **Discador próprio?** Transfira a chamada atendida ao agente com esses mesmos cabeçalhos — ou disque pela API de outbound

3 Gravação com Transcrição

- 1 Grave em estéreo · cliente e atendente em canais separados

```
exten => s,1,MixMonitor(${UNIQUEID}.wav,D)
same => n,Dial(PJSIP/${RAMAL})
```

- 2 Transcreva em lote com Python no cron · a cada 5 min

```
# /usr/local/bin/transcrever.py
# cron: */5 * * * * asterisk transcrever.py
import requests, pathlib, time
GRAV = pathlib.Path(
    "/var/spool/asterisk/monitor")
for wav in GRAV.glob("*.wav"):
    saida = wav.with_suffix(".json")
    if saida.exists(): continue
    if time.time() - wav.stat().st_mtime < 60:
        continue # gravação em andamento
r = requests.post(
    "https://api.sippulse.ai"
    "/v1/asr/transcribe",
    headers={"api-key": "SUA_API_KEY"},
    files={"file": open(wav, "rb")},
    data={"model": "pulse-telephony",
        "response_format":
            "stereo_diarization",
            "vad_preset": "telephony",
            "language": "pt"})
saida.write_text(r.text)
```

- 3 Analise · sentimento, objeções, tom e interesse do cliente

- **pulse-telephony**: o melhor modelo SipPulse para áudio telefônico, diarização por canal 100% precisa
- WAV, MP3, PCM e OGG até 25 MB · **sem cron**: 3º argumento do MixMonitor dispara o script ao desligar

4 Texto para Fala (TTS)

- 1 No dialplan, um AGI recebe o texto da chamada

```
exten => 5000,1,Answer()
same => n,AGI(tts.py,01á ${CALLERID(name)}
    seu boleto vence amanhã)
```

- 2 O AGI gera, cacheia e toca o áudio · sem System() nem shell

```
# /var/lib/asterisk/agi-bin/tts.py
import sys, requests, hashlib
import subprocess, pathlib
texto = " ".join(sys.argv[1:])
while sys.stdin.readline().strip(): pass
som = pathlib.Path(
    "/var/lib/asterisk/sounds/tts") / \
    hashlib.md5(texto.encode()).hexdigest()
if not som.with_suffix(".wav").exists():
    # cache: gera só na primeira vez
r = requests.post(
    "https://api.sippulse.ai"
    "/v1/tts/generate",
    headers={"api-key": "SUA_API_KEY"},
    json={"input": texto,
        "model": "azure-tts",
        "voice": "pt-BR-FranciscaNeural",
        "response_format": "wav"})
bruto = som.with_suffix(".raw.wav")
bruto.write_bytes(requests.get(
    r.json()["download"]).content)
subprocess.run(["sox", str(bruto),
    "-r", "8000", "-c", "1",
    str(som.with_suffix(".wav"))])
print(f'STREAM FILE tts/{som.name} ""')
sys.stdout.flush()
```

- Vozes PT-BR de múltiplos provedores · liste em [GET /v1/tts/voices](#)
- **Cache por hash do texto**: frases repetidas tocam na hora, sem latência nem custo extra

● **Transferência** para filas, ramais e números externos (SIP REFER) · habilite com **tT** no Dial

● **DTMF bidirecional** · envia para URAs e recebe CPF/PIN (RFC 4733)

● **Latência** inferior a 500 ms



Comece agora: crie um agente e aponte seu Asterisk para

`agt_xxxx@sip.sippulse.ai`

`sippulse.ai` · docs.sippulse.ai · api.sippulse.ai · cobrança por consumo, tipicamente abaixo de R\$ 0,75/min