



1 Agentes de Voz

O agente é um destino SIP:

`agt_xxxx@sip.sippulse.ai`. Um `bridge` resolve.

A Bridge direto RECOMENDADO

```
<!-- dialplan/default/agente.xml -->
<extension name="agente-ia">
  <condition field="destination_number"
    expression="^4000$">
    <action application="bridge"
      data="sofia/external/
        agt_xxxx@sip.sippulse.ai"/>
  </condition>
</extension>
```

B SipPulse AI registra no FreeSWITCH

```
<!-- directory/default/4000.xml -->
<include>
  <user id="4000">
    <params>
      <param name="password"
        value="senha-forte"/>
    </params>
  </user>
</include>
```

Cadastre servidor, usuário e senha no painel SipPulse AI (registro externo) e chame com `bridge user/4000`.

- DTMF RFC 2833 e transferência via REFER já são o padrão do sofia
- Recarregue com `reloadxml` e teste discando **4000**

2 Variáveis nos cabeçalhos SIP

Toda variável `sip_h_X-*` vira cabeçalho SIP na saída.

Valores em Base64 — um Lua curto resolve. Quebras de linha são só visuais.

```
-- scripts/agente.lua
-- dialplan: <action application="lua"
--   data="agente.lua ${caller_id_number}"/>
local function b64(s)
  local p = io.popen("printf %s ' "
    .. s .. "' | base64 -w0")
  local r = p:read("*a"); p:close()
  return r
end
-- busque no CRM pelo argv[1] e injete
local instr = "0 cliente se chama "
.. nome .. ", fatura vence " .. venc
session:setVariable(
  "sip_h_X-Additional-Instructions",
  b64(instr))
session:setVariable("sip_h_X-Initial-Message",
  b64("Olá " .. nome .. ", aqui é a Ana"))
session:setVariable("sip_h_X-Uid",
  b64(session:getVariable("uuid")))
session:execute("bridge",
  "sofia/external/agt_xxxx@sip.sippulse.ai")
```

- `X-Additional-Instructions` soma contexto ao prompt só nesta chamada · `X-Initial-Message` troca a saudação · `X-Uid` casa com seu CRM
- Sem Lua? Sem Base64 não há acentos: prefira o script acima

3 Gravação com Transcrição

O FreeSWITCH grava **estéreo nativo** — sem sox, sem conversão: é o cenário perfeito para diarização.

1 Grave em estéreo · uma linha antes do bridge

```
<action application="set"
  data="RECORD_STEREO=true"/>
<action application="record_session"
  data="${recordings_dir}/${uuid}.wav"/>
```

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

```
# /usr/local/bin/transcrever.py
import requests, pathlib, time
REC = pathlib.Path(
  "/var/lib/freeswitch/recordings")
for wav in REC.rglob("*.wav"):
  saida = wav.with_suffix(".json")
  if saida.exists(): continue
  if time.time()-wav.stat().st_mtime < 60:
    continue
  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)
```

- `pulse-telephony`: diarização por canal 100% precisa · até 25 MB por arquivo
- Depois, análise estruturada: `POST /v1/structured-analyses/{id}/execute`

4 Texto para Fala (TTS)

Um Lua com cache toca áudio dinâmico em URAs e anúncios, direto do dialplan.

```
-- scripts/tts.lua · dialplan:
-- <action application="lua"
--   data="tts.lua Olá ${caller_id_name}"/>
local api = freeswitch.API()
local texto = table.concat(argv, " ")
local h = api:execute("md5", texto)
local f = "/usr/share/freeswitch"
  .. "/sounds/tts/" .. h .. ".wav"
local t = io.open(f)
if not t then -- cache: gera 1 vez
  os.execute("/usr/local/bin/tts.sh "
    .. texto .. " " .. f)
else t:close() end
session:answer()
session:streamFile(f)
```

```
# /usr/local/bin/tts.sh "texto" saida.wav
URL=$(curl -s -X POST \
  https://api.sippulse.ai/v1/tts/generate \
  -H "api-key: SUA_API_KEY" \
  -H "Content-Type: application/json" \
  -d '{"input":"'${1}',
    "model":"'azure-tts"',
    "voice":"'pt-BR-FranciscaNeural'",
    "response_format":"'wav'"}' \
  | jq -r .download)
curl -s "$URL" | sox - -r 8000 -c 1 "$2"
```

- Cache por hash (md5 do próprio FreeSWITCH): frases repetidas tocam na hora
- Vozes PT-BR de múltiplos provedores: `GET /v1/tts/voices`

● Estéreo nativo · RECORD_STEREO=true e a diarização é perfeita

● Sem MRCP, sem módulo extra · só SIP, Lua e REST

● Latência inferior a 500 ms



Comece agora: crie um agente e faça o bridge do seu FreeSWITCH em minutos

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