

API Biziva Guia de integração

Edição 17/10/2026 | Para você e para a IA que desenvolve seu sistema

Baixe. Envie à sua IA. Integre com orientação.

Este documento explica como conectar seu site, aplicativo ou sistema à API Biziva para consumir texto, imagens, áudio, vídeo e agentes com documentos da empresa.

Como usar, mesmo sem saber programar

1. Baixe este PDF no API empresarial da Biziva.
2. Anexe o PDF ao ChatGPT, Claude, Cursor ou outra IA que ajude no desenvolvimento do seu sistema.
3. Copie o pedido pronto da próxima página e diga o que você quer integrar primeiro.
4. Dê à IA acesso ao projeto ou peça que ela indique quais arquivos você deve editar.
5. Crie sua chave no painel e configure-a somente no servidor. Não cole a chave no chat da IA.
6. Peça o teste de conexão antes de autorizar operações pagas.

O que você precisa

Uma conta da API, chave ativa com permissões adequadas, saldo para operações pagas e acesso ao código ou ao desenvolvedor do seu sistema. O documento orienta a integração; a IA ainda precisa do projeto e de um ambiente onde possa trabalhar.

BASE RECOMENDADA

<https://api.biziva.com.br/v1>

CONTA E CHAVES

<https://api.biziva.com.br/console>

DOCUMENTAÇÃO ATUAL

<https://api.biziva.com.br/docs>

Compatibilidade principal: Chat Completions no formato OpenAI. Recursos adicionais têm contratos próprios descritos neste guia. Nem todo recurso de um SDK externo é suportado.

01 | Pedido pronto para a sua IA

Anexe este PDF e copie a mensagem abaixo. Troque somente a descrição do sistema e os recursos desejados. Não inclua a chave real.

Quero integrar a API Biziva ao meu sistema.

Leia este PDF como referência técnica, e não como acesso ao projeto.

Meu sistema é: [descreva seu site/app e sua linguagem, se souber].

Quero começar com: [chat, imagens, voz, documentos ou outro recurso].

1. Inspecione o projeto e identifique backend, frontend e autenticação.
2. Use a base `https://api.biziva.com.br/v1`.
3. Crie `BIZIVA_API_KEY` como segredo no servidor. Não peça que eu cole a chave no chat, no código público, no navegador ou nos logs.
4. Implemente primeiro GET `/models` para testar conexão sem gerar IA.
5. Use as rotas e os formatos deste PDF; não invente endpoints.
Para chat compatível, use POST `/chat/completions`, não `/responses`.
6. Use um ID idempotente persistente por operação lógica. Em uma repetição, preserve a mesma chave e exatamente o mesmo conteúdo.
7. Trate 202 como tarefa em processamento. Consulte a `status_url` retornada. Não gere outra operação para resolver uma incerteza.
8. Persista ID do pedido, estado e recibo. Mostre ao usuário custo em BRL pelo campo `charged_brl`; não use só `charged_cents`.
9. Implemente erros em português, timeout e consultas com espera crescente. Nunca faça repetição paga indiscriminada.
10. Mantenha os outros provedores do sistema funcionando.
11. Escreva testes locais com respostas simuladas e sem custo real.
12. Mostre os arquivos alterados, como configurar o segredo e como executar. Separe o teste gratuito de conexão dos testes pagos.

Antes de executar qualquer teste pago, informe operação e orçamento.

Se faltar acesso ao projeto, explique o passo necessário.

Não prometa integração concluída sem testar o fluxo ponta a ponta.

Se você usa uma ferramenta sem código

Procure um conector OpenAI compatível que permita URL personalizada. Configure a base recomendada e o modelo biziva-auto para chat. Se a ferramenta exigir uma chave iniciada por sk-, não altere sua chave por conta própria: essa compatibilidade depende da conta. Se ela não aceitar URL própria, será necessário um conector ou backend intermediário.

02 | Conexão, chave e permissões

URLs e arquitetura

Use `https://api.biziva.com.br/v1` nas integrações novas. O prefixo `/api/v1` no mesmo domínio também é aceito. A integração legada `https://biziva.com.br/api/v1` continua compatível. Prefira o domínio independente para novas configurações. Confirme conta, chave e saldo na origem usada; não presuma migração automática entre ambientes. O domínio não elimina todas as dependências compartilhadas de infraestrutura.

Fluxo recomendado: seu navegador/app chama seu backend; seu backend chama a Biziva. Autentique seus próprios usuários e aplique limites por usuário no seu sistema. Uma chave da empresa não deve ser distribuída aos clientes finais.

Autenticação

```
Authorization: Bearer SUA_CHAVE_BIZIVA
Content-Type: application/json
Idempotency-Key: pedido-unico-0001
```

Formato normal da chave: `bzv_` seguido de 64 caracteres hexadecimais. `Authorization: Bearer` é o padrão recomendado para todas as rotas. Algumas rotas aceitam `x-api-key`, mas não dependa desse cabeçalho para cobertura universal. O prefixo adicional `sk-` é uma exceção de compatibilidade por conta, não uma regra global.

Permissão	Uso
<code>catalog:read</code>	Consultar modelos, capacidades e preços.
<code>usage:read</code>	Consultar consumo e saldo.
<code>services:create</code>	Executar serviços de IA; operações pagas precisam de saldo.

A chave completa aparece uma única vez. Defina validade, revogue quando necessário e mantenha ambientes separados. Uma chave expirada, revogada ou sem permissão não deve ser recuperada copiando credenciais de outra conta.

Primeiro teste: somente conexão

```
curl https://api.biziva.com.br/v1/models \
-H "Authorization: Bearer $BIZIVA_API_KEY"

curl https://api.biziva.com.br/v1/features \
-H "Authorization: Bearer $BIZIVA_API_KEY"
```

Essas consultas não geram conteúdo de IA. HTTP 200 com catálogo confirma autenticação e acesso ao catálogo; não comprova que toda modalidade paga pode ser executada com o saldo e as permissões atuais.

03 | Texto e conversa

Requisição mínima

```
curl https://api.biziva.com.br/v1/chat/completions \
-H "Authorization: Bearer $BIZIVA_API_KEY" \
-H "Content-Type: application/json" \
-H "Idempotency-Key: chat-pedido-0001" \
-d '{"model": "biziva-auto", "messages": [
  {"role": "system", "content": "Responda de forma breve em português."},
  {"role": "user", "content": "Explique como organizar meus atendimentos."}
], "max_tokens": 500, "stream": false}'
```

Leia o texto em `choices[0].message.content`; `finish_reason` informa o encerramento; `usage` contém os tokens medidos. Campos adicionais aparecem em biziva, incluindo o recibo. Valores e IDs de exemplos são ilustrativos, não tarifas contratadas.

Perfil	Quando usar
biziva-auto	Classificação e seleção automática.
biziva-fast	Texto simples e respostas curtas.
biziva-smart	Tarefas mais exigentes; não equivale a qualquer modelo específico.
biziva-vision	Entrada com imagens ou arquivos aceitos pela rota.
Modelo exato do catálogo	Somente ID aceito e disponível no catálogo da conta.

Streaming e ferramentas

`stream:true` retorna SSE no contrato Chat Completions: eventos `data:` com chunks e encerramento `data: [DONE]`. Solicite `stream_options:{include_usage:true}` quando precisar do uso final. Trate erros após início do fluxo e não repita uma geração automaticamente se o navegador perder a conexão. Reconstrua o texto pelos deltas e persista o ID lógico no backend.

POST `/messages` oferece compatibilidade no formato Anthropic. Não presuma suporte integral a todas as APIs Anthropic/OpenAI. Para function calling e JSON estruturado, escolha modelo compatível no catálogo e valide a resposta no servidor. Não use `/responses` ou `Assistants` como se fossem aliases públicos da Biziva.

Limites e modelos

O chat aceita até 200 mensagens e uma escolha por chamada (n maior que 1 não é suportado). Limites de entrada, saída, contexto e modalidades variam por modelo e política da conta. Consulte `GET /models/{id}`; não anuncie 1 milhão de tokens ou 128 mil de saída para todas as rotas automáticas.

04 | Cliente de integração em JavaScript

Exemplo para Node.js com fetch nativo, executado no servidor. Não use este arquivo como código de navegador. Gere e guarde o ID antes do primeiro POST.

```
const BASE = "https://api.biziva.com.br/v1";

export async function biziva(path, {method="GET", body, requestId={}}) {
  const key = process.env.BIZIVA_API_KEY;
  if (!key) throw new Error("Configure BIZIVA_API_KEY no servidor.");
  const headers = {Authorization: "Bearer " + key};
  if (body !== undefined) headers["Content-Type"] = "application/json";
  if (requestId) headers["Idempotency-Key"] = requestId;
  const response = await fetch(BASE + path, {
    method, headers,
    body: body === undefined ? undefined : JSON.stringify(body),
    signal: AbortSignal.timeout(120000)
  });
  const data = await response.json();
  if (!response.ok) {
    const error = new Error(data.error?.message || "Falha na API.");
    error.status = response.status;
    error.details = data.error;
    throw error;
  }
  return {status: response.status, data};
}

// Sem custo de geração:
const catalog = await biziva("/models");

// Operação paga: pedido.id deve vir da sua base e ser reutilizado.
const result = await biziva("/chat/completions", {
  method: "POST", requestId: pedido.id,
  body: {model:"biziva-auto", messages:[
    {role:"user",content:pedido.texto}
  ], max_tokens:500, stream:false}
});
const texto = result.data.choices[0].message.content;
const recibo = result.data.biziva;
// Salve texto, recibo e pedido.id, sem gravar a chave nos logs.
```

O exemplo espera JSON e não implementa SSE, multipart ou arquivo binário. Use um adaptador separado para áudio, uploads e streaming. Em timeout, mantenha o ID e consulte/reenvie somente pelo contrato idempotente; não crie um novo pedido só para tentar novamente.

05 | Cobrança, saldo e recargas

A API usa carteira pré-paga, separada da mensalidade da plataforma. O pedido pode reservar um valor máximo e depois liquidar o uso medido. Reserva não é custo final. O saldo não utilizado é liberado quando o fluxo confirma a liquidação ou devolução.

Cobrança fracionária

Campo	Interpretação
biziva.charged_brl	Valor efetivo do consumo em reais; preserve casas decimais.
biziva.charged_cents	Débito inteiro em centavos. Pode ser zero mesmo havendo consumo fracionário.
biziva.billed_micros	Unidades contábeis quando presentes: 300.000 unidades = R\$1.
biziva.price_precision	fractional-v1 ou precisão legada do recibo.

Nunca calcule receita usando apenas `charged_cents/100` em operações fracionárias. Armazene `billed_micros` como inteiro, quando disponível; converta para reais dividindo por 300.000. Não use arredondamento de exibição como livro financeiro.

```
GET /usage
```

```
Authorization: Bearer SUA_CHAVE_BIZIVA
```

```
# Consulta saldo e reservas/consumos da conta.
```

```
# availableBalanceBrl representa o saldo disponível com frações.
```

```
POST /topups
```

```
{"amountCents":10000,"email":"financeiro@suaempresa.com"}
```

```
GET /topups/TOPUP_ID
```

Recarga padrão via Pix: R\$ 100 a R\$ 5.000, conforme política da conta. `amountCents` é inteiro em centavos. Criar Pix não comprova pagamento: espere status `credited`. Se já pagou e a consulta falhar, não gere outra recarga para solucionar a incerteza.

Preços e cotação

GET /pricing informa a política atual. Texto, busca e embeddings seguem o multiplicador de fornecimento informado; imagem, vídeo e áudio têm política própria. Hoje: 1, 5x texto e 1, 3x mídia sobre o custo aplicável, convertidos para BRL pelo servidor. Tarifas e câmbio não devem ser fixados no seu código.

Perfis de texto: `biziva_profile` automatic, exact ou economic. Modelo exato mantém o contrato do catálogo. O perfil `economic` exige elegibilidade e custos verificados da conta; pode estar indisponível. POST /pricing/quote oferece cotação estimada, não garantia geral de economia. Consulte a resposta atual antes de oferecer o recurso.

06 | Idempotência e tratamento de erros

Envie Idempotency-Key de 8 a 100 caracteres: letras, números, hífen ou sublinhado. Um UUID é adequado. Use uma chave por operação lógica e preserve o mesmo conteúdo na repetição. Não reutilize o mesmo ID para serviço ou parâmetros diferentes. Registre estado, ID público e ID enviado.

Um replay concluído reaproveita o resultado conforme o contrato. HTTP409 pode indicar conteúdo diferente, pedido em andamento ou falha/resultado incerto. Leia `error.type` e `error.message`. Não trate todo409 como autorização para trocar de chave.

HTTP	Conduta do integrador
400 / 422	Corrigir campo, formato, mídia ou contrato recusado.
401	Conferir chave, validade e revogação.
402	Saldo/reserva insuficiente ou orçamento configurado. Mostrar motivo.
403	Permissão, papel ou recurso indisponível para essa chave.
404 / 410	ID inexistente, recurso não publicado, arquivo removido ou expirado.
409	Conflito, pedido existente ou resultado incerto: consultar antes de agir.
413	Reduzir arquivo/corpo para o limite da rota.
423	Execuções pausadas por controle da empresa.
429	Limite de uso, simultaneidade, orçamento ou proteção de gastos.
502 / 503 / timeout	Não afirmar custo zero. Consultar o pedido e conciliar antes de repetir.

Formato e decisões

```
{"error":{"message":"Saldo insuficiente","code":402}}
```

Algumas famílias têm envelope próprio e não retornam todos esses campos. Trate primeiro o status HTTP e depois os campos presentes. Não exponha erros brutos de fornecedor ou segredos ao cliente final.

Use espera crescente para consultas GET, limite de tentativas e cancelamento da espera no seu app. Se a execução estiver em `reconciliation_pending` ou resultado incerto, mantenha o pedido e apresente suporte/pendência. Um novo POST pago com ID novo é uma nova operação, não uma consulta.

07 | Imagens: criação e edição

```
POST /images/generations
{"model":"biziva-image","prompt":"Uma sala de reunião iluminada",
 "size":"1024x1024","n":1,"response_format":"url"}
```

```
POST /images/edits
Content-Type: multipart/form-data
prompt=Preserve a pessoa e altere o fundo para um escritório.
image[]=@foto1.jpg
image[]=@foto2.jpg
size=1024x1536
```

Use prompt de até 3.200 caracteres, n de 1 a 4 e formato suportado. Exemplos de size:1024x1024, 1024x1536 e1536x1024. Edição também aceita JSON com images:[URLs ou data URIs]; até 5 referências, cada uma até 10 MB. É necessário ao menos uma referência em edits.

biziva-image é a opção padrão; biziva-image-premium ou quality:hd solicitam a modalidade premium. O roteamento considera pedido e conta. Não prometa equivalência facial perfeita nem que o mesmo motor será usado em toda chamada automática.

Como receber e guardar

A resposta normal traz data[].url ou data[].b64_json conforme response_format e um recibo biziva. Baixe as imagens e salve no armazenamento do seu sistema; URLs são temporárias. Não faça a integração depender da retenção da URL.

Recomendado: processamento assíncrono

```
POST /images/generations
Prefer: respond-async
Idempotency-Key: imagem-pedido-0001
{"model":"biziva-image","prompt":"Sala iluminada",
 "response_format":"url","n":1,"async":true}

# HTTP202: guarde id/status_url retornados.
GET /images/requests/REQUEST_ID
```

Assíncrono de imagem usa sua própria rota de consulta, não /tasks. Use a status_url devolvida, pois o prefixo pode vir como /api/v1. Nesse modo, use URL; b64_json não é aceito. Limite atual:20 pedidos pendentes por conta. Não repita uma geração falha ou incerta com novo ID automaticamente.

Não existem rotas públicas de feedback/avaliação de qualidade do cliente nesse contrato. Não implemente /feedback ou /quality a partir de versões antigas do guia.

08 | Tarefas, arquivos e acompanhamento

Documentos, agentes, buscas, música, efeitos, voz personalizada e operações ampliadas de vídeo usam tarefas. POST de execução retorna 202 com id e status_url. Guarde-os e consulte o recurso até um estado terminal. O processo não terminou só porque o servidor aceitou a requisição.

```
GET /tasks/TASK_ID
GET /tasks
DELETE /tasks/TASK_ID
```

Estado	O que mostrar
preparing / queued	Preparando / na fila.
running / submitted	Executando / aguardando o fornecedor.
awaiting_approval	Ação externa precisa de aprovação exata.
paused	Empresa ou agente pausado; aguardar retomada.
succeeded	Exibir o resultado no campo result.
failed / cancelled	Mostrar falha ou cancelamento; verificar etapas já cobradas.
reconciliation_pending	Resultado incerto: não disparar novamente automaticamente.

Cancelamento só é aceito quando o contrato permite, por exemplo queued ou awaiting_approval. Etapas já realizadas podem ter cobrança. Uma pausa de agente impede a próxima etapa, não desfaz a execução externa que já começou.

Uploads e downloads de mídia

```
POST /media-assets
Content-Type: audio/wav
Content-Length: TAMANHO_EM_BYTES
[corpo binário]
```

```
POST /media-assets?seconds=5
Content-Type: video/mp4
Content-Length: TAMANHO_EM_BYTES
[corpo MP4 binário]
```

```
GET /media-assets/ASSET_ID
```

Não envie JSON ou base64 nessa rota binária. Áudio e imagem: até 16 MB; vídeo MP4: até 64 MB e 60 segundos, com duração declarada e áudio válido. Rotas específicas podem impor limite menor. Assets são privados por conta e expiram em 7 dias. Baixe e mantenha a cópia necessária no seu sistema.

09 | Áudio e vetores de texto

Transcrição

```
curl https://api.biziva.com.br/v1/audio/transcriptions \
-H "Authorization: Bearer $BIZIVA_API_KEY" \
-H "Idempotency-Key: transcricao-pedido-0001" \
-F "file=@audio.mp3" -F "language=pt"
```

Arquivo até 25 MB. A resposta padrão é JSON com text e recibo quando presente; response_format:text solicita texto simples. Registre o custo pelos campos/cabeçalhos retornados no formato escolhido.

Texto para voz

```
POST /audio/speech
{"model":"biziva-tts","input":"Olá, como posso ajudar?",
 "voice":"alloy","response_format":"mp3"}
```

A resposta é binária; salve o arquivo, não tente lê-la como JSON. input até 4.096 caracteres. Formatos:mp3, opus, aac, flac e wav. Vozes aceitas incluem alloy, ash, ballad, coral, echo, fable, onyx, nova, sage, shimmer, verse, marin e cedar. Valide conforme o modelo escolhido; instruções e formatos podem restringir rotas.

Os cabeçalhos de cobrança e ID devem ser preservados no backend. Use os campos de precisão fracionária quando disponíveis; o cabeçalho inteiro em centavos sozinho pode não representar todo o consumo efetivo.

Embeddings

```
POST /embeddings
{"model":"biziva-embed","input":["texto 1","texto 2"],
 "dimensions":256}
```

Até32 textos por chamada. A resposta traz data[].embedding, índice e uso. dimensions depende do modelo. Um índice vetorial precisa manter a mesma dimensão/modelo; não misture embeddings incompatíveis. A base de conhecimento gerenciada da Biziva já cuida do índice do seu próprio contrato.

Catálogo

GET /models lista os perfis disponíveis. GET /audio/models informa modalidades e tarifas atuais para música, efeitos e voz personalizada. Esses recursos não usam o corpo de /audio/speech.

10 | Música, efeitos e voz personalizada

```
POST /audio/music
{"prompt":"Piano suave para apresentação","seconds":30,
 "instrumental":true}
```

```
POST /audio/effects
{"prompt":"Um sino suave com reverberação","seconds":2}
```

Música: 5 a 120 segundos. Efeitos: 0, 5 a 22 segundos. Retornam tarefa 202; consulte /tasks/ID e baixe o asset indicado. Consulte /audio/models para tarifas vigentes; o tempo solicitado não substitui o recibo final.

Cadastrar uma voz autorizada

```
POST /voices
{"name":"Minha voz autorizada"}

# A resposta devolve VOICE_ID e a frase do desafio.
# Grave a frase com a voz autorizada e envie em /media-assets.
POST /voices/VOICE_ID/verify
{"asset_id":"ASSET_ID","consent":true,"rights_holder":true}

# Após estado verified:
POST /audio/clones
{"voice_id":"VOICE_ID","text":"Mensagem curta autorizada.",
 "language":"portuguese"}

GET /voices
DELETE /voices/VOICE_ID
```

Envie Idempotency-Key na verificação e nas gerações. O desafio verifica a presença da frase no áudio, não comprova identidade civil ou autoria. Não envie voz de terceiros sem autorização. Revogar a voz, remover ou expirar a referência impede seu uso posterior.

Texto para voz personalizada: até 300 caracteres. Idiomas expostos em inglês: portuguese, english, spanish, french, german e italian. Arquivos de referência expiram em 7 dias; renove o fluxo autorizado quando necessário. O texto comum de /audio/speech não exige cadastro de clone.

11 | Vídeo com áudio

Geração padrão e ampliada

```
POST /videos
{"model":"biziva-video","prompt":"Uma praça ao entardecer",
 "audio_prompt":"Ambiente natural da praça"}
```

```
POST /videos/extended
{"prompt":"Uma praça ao entardecer","seconds":10,
 "resolution":"1080p","aspect_ratio":"16:9",
 "audio_prompt":"Ambiente natural da praça"}
```

A rota padrão /videos gera aproximadamente 5 segundos em 720p e tem consulta própria em /videos/ID. A ampliada aceita 5, 10 ou 15 segundos e 720p/1080p, usando /tasks/ID. Ambas exigem áudio. Guarde e use a status_url/rota de consulta do retorno, sem misturar os dois contratos.

Consulte GET /videos/models para tarifas e limites. Para vídeo ampliado com imagem inicial, envie PNG/JPEG em /media-assets e inclua image_asset_id. O resultado passa por verificação de áudio; recusa por silêncio não autoriza regenerar automaticamente.

Compor cliques e substituir narração

```
POST /videos/edits
{"video_asset_id":"ASSET_1","clips":[
 {"asset_id":"ASSET_1"}, {"asset_id":"ASSET_2"}]}
```

```
POST /videos/dubs
{"video_asset_id":"ASSET_ID","text":"Roteiro fornecido.",
 "voice":"alloy"}
```

Composição: até 10 cliques e 60 segundos no total, preservando áudio. Dublagem por roteiro gera a voz do texto informado e substitui a faixa sonora. Não faz sincronização labial. A narração precisa ter duração compatível. Voz e composição são etapas distintas e podem ter cobrança individual.

Traduzir um vídeo existente

```
POST /videos/translations
{"video_asset_id":"ASSET_ID","target_language":"en",
 "voice":"alloy"}
```

Upload MP4 de até 16 MB e 60 segundos, com áudio. Idiomas: pt, en, es, fr, de, it. Transcreve, traduz e cria nova narração; não clona automaticamente a voz do falante e não faz lip sync. Consulte a tarefa pai e, se informado, child_task_id. Etapas concluídas mantêm seus recibos.

12 | Voz em tempo real

Realtime tem um protocolo próprio de WebSocket. Não é uma chamada HTTP de texto nem um conector WebRTC pronto. Crie a sessão no seu backend e entregue somente o ticket temporário ao cliente autorizado.

```
POST /realtime/sessions
Idempotency-Key: sessao-voz-0001
{"model":"biziva-realtime","voice":"alloy",
 "instructions":"Atenda em português de forma breve."}
```

```
# Guarde id, websocket_url, ticket, ticket_expires_at,
# protocols, limits e reservation_brl retornados.
```

```
GET /realtime/sessions/SESSION_ID
DELETE /realtime/sessions/SESSION_ID
```

Abertura: use `websocket_url` e os subprotocolos exatamente como retornados, por exemplo `biziva-realtime` e `biziva-ticket.TICKET`. O ticket expira em 60 segundos, é de uso único e não é reemitido no replay. Nunca envie a chave permanente da API para o navegador.

Áudio e eventos

```
// Exemplo conceitual no cliente, depois de receber a sessão do backend:
const ws = new WebSocket(session.websocket_url, session.protocols);
```

```
// PCM16, 24 kHz, mono, em base64:
{"type":"input_audio_buffer.append","audio":"BASE64_PCM16"}
{"type":"input_audio_buffer.commit"}
{"type":"response.create"}
{"type":"response.cancel"}
{"type":"input_audio_buffer.clear"}
```

Limites atuais: 120 segundos de sessão, 30 segundos de áudio de entrada, 6 turnos e 512 tokens de saída por resposta; até 2 sessões pendentes por conta. Consulte `limits`, pois o contrato pode evoluir. O backend reserva saldo e liquida o uso medido ao encerrar. Não confunda a reserva com preço fixo por sessão.

Implemente captura e conversão de áudio, reprodução dos deltas, botão de encerrar, tratamento de fechamento e consulta do recibo final. Não reutilize ticket usado nem crie sessões em ciclo automático após desconexão.

13 | Documentos e respostas com fontes

```
POST /knowledge-bases
{"name":"Documentos da minha empresa"}

POST /knowledge-bases/KB_ID/documents
Idempotency-Key: manual-atendimento-0001
{"name":"Manual.txt","text":"Texto completo do manual..."}

POST /knowledge-bases/KB_ID/answers
Idempotency-Key: pergunta-manual-0001
{"query":"Qual é o prazo de atendimento?","top_k":5}

POST /knowledge-bases/KB_ID/search
{"query":"prazo de atendimento","top_k":5}
```

Espere a tarefa de indexação concluir e o documento ficar ready antes de consultar. Resposta de answers inclui answer e fontes com documento, página e trecho; marcadores como[S1] referenciam as fontes retornadas. Validação do identificador não certifica que toda afirmação da IA esteja correta.

Formatos e limites

TXT, Markdown, CSV, JSON, PDF com texto, DOCX e XLSX. Binários:JSON com name, mime e data_base64; até 2 MB. PDF com texto:até 100 páginas. Até80 trechos por documento, 1.000 trechos por base e20 bases por conta. DOCX/XLSX extraem conteúdo textual; não são renderizadores completos de layout nem motores de fórmulas/macros.

```
GET /knowledge-bases
GET /knowledge-bases/KB_ID/documents
GET /knowledge-bases/KB_ID/documents/DOC_ID
DELETE /knowledge-bases/KB_ID/documents/DOC_ID
DELETE /knowledge-bases/KB_ID
```

OCR: importação paga e revisão

```
POST /document-imports
{"name":"Contrato digitalizado.pdf","mime":"application/pdf",
 "data_base64":"BASE64_DO_ARQUIVO"}
```

OCR aceita PDF, PNG ou JPEG até 2 MB; PDF até 10 páginas. Retorna tarefa de extração com texto para revisão humana, review_required:true e indexed:false. Revise antes de enviar o texto para a base. O OCR não indexa automaticamente nem deve ser acionado silenciosamente no lugar de extração comum.

Textos indexados são enviados ao fornecedor de embeddings; trechos selecionados podem ser enviados ao modelo de resposta. Dados são isolados por conta. Só envie documentos que você pode processar para essa finalidade.

14 | Agentes, rascunhos e ações externas

```
POST /agents
{"name":"Atendimento e documentos","model":"biziva-smart",
 "knowledge_base_ids":["KB_ID"],
 "instructions":"Consulte os documentos e produza rascunhos."}
```

```
POST /agents/AGENT_ID/runs
Idempotency-Key: atendimento-run-0001
{"input":"Consulte o manual e prepare uma resposta.",
 "max_steps":6,"max_cost_brl":15}
```

Até 6 etapas. O orçamento limita compromissos por execução e etapas; consulta à base também consome. Agentes podem pesquisar documentos, criar rascunhos Markdown e abrir tickets internos. Não têm acesso irrestrito aos sistemas da empresa.

```
GET /agents
DELETE /agents/AGENT_ID
GET /documents/DOC_ID
DELETE /documents/DOC_ID
GET /tickets
```

Conectar uma ferramenta HTTP da sua empresa

```
POST /agent-tools
{"name":"Criar demanda no CRM",
 "url":"https://meu-sistema.com/biziva/actions",
 "authorization":"Bearer SEGREDO_DO_MEU_SISTEMA"}

# Inclua tool_ids:["TOOL_ID"] no agente.
# Uma ação externa para em awaiting_approval.
POST /tasks/RUN_ID/approve
{"approval_id":"ID_EXIBIDO","hash":"HASH_EXIBIDO",
 "approved":true}
```

Revise URL, método, corpo e hash da ação antes de aprovar. Seu endpoint recebe uma chave idempotente e também precisa deduplicá-la. A autorização fica criptografada; não a coloque no frontend ou em logs. Só destinos HTTPS públicos são aceitos; redirects não são seguidos. Uma execução incerta não é reenviada automaticamente.

Abrir ticket não envia WhatsApp, e-mail ou Slack. Integração com CRM/GitHub/outros canais exige cadastro e autorização próprios. O console oferece um assistente para configurar ações GitHub de issue ou dispatch; não há acesso automático a repositórios do cliente.

15 | Pesquisa na web e lotes

Pesquisa com fontes

```
POST /web-search
Idempotency-Key: pesquisa-0001
{"query":"Pergunta para pesquisar...",
 "allowed_domains":["exemplo.com"]}
```

allowed_domains é opcional, até 20 domínios. A tarefa retorna resposta e referências de fontes. Busca e tokens entram no custo da execução; não interprete a tarifa da ferramenta como preço completo da resposta.

Lote com acompanhamento individual

```
POST /batches
Idempotency-Key: lote-pedidos-0001
{"items":[
  {"custom_id":"texto-001","operation":"chat","body":{"
    "messages":[{"role":"user","content":"Olá"}],"max_tokens":500}},
  {"custom_id":"vetor-001","operation":"embed","body":{"
    "input":"Texto de exemplo","dimensions":256}}
], "webhook_id":"HOOK_ID_OPCIONAL"}
```

Até 25 itens e custom_id único dentro do lote. Operações aceitas: chat, embed, speech, web-search e knowledge-answer. Chat de lote: até 30 mensagens, 2.000 tokens de saída e sem streaming. knowledge-answer recebe kb_id e query no body.

A tarefa final contém manifesto por item com status e result_url. Baixe cada resultado usando a chave da conta. Um item com falha não obriga repetir todo o lote. Cada etapa tem cobrança própria; não existe desconto automático equivalente ao Batch de outros fornecedores.

```
GET /tasks/BATCH_ID
GET /tasks/BATCH_ID/items/CUSTOM_ID
```

Resultado e arquivo

URLs retornadas como caminhos /api/v1/... devem ser resolvidas no domínio api.biziva.com.br, não concatenadas cegamente com /v1. Valide o domínio antes de anexar Authorization. Não encaminhe sua chave a URLs de fornecedores ou de terceiros.

16 | Webhooks: notificações de tarefas

```
POST /webhooks
{"url":"https://meu-sistema.com/webhooks/biziva"}
```

```
GET /webhooks
DELETE /webhooks/HOOK_ID
```

Guarde `signing_secret` no servidor: ele aparece uma única vez. Informe `webhook_id` nos pedidos compatíveis. A notificação informa ID e estado; consulte o resultado privado com sua chave. Não confie apenas no corpo de um callback para atribuir saldo ou liberar ações.

Validar a assinatura

Cabeçalhos:
`webhook-id`
`webhook-timestamp`
`webhook-signature`

Mensagem assinada:
`TIMESTAMP + "." + CORPO_JSON_EXATO`

Assinatura:
`v1=HEX_MINUSCULO(HMAC_SHA256(signing_secret, mensagem))`

Leia o corpo bruto antes de fazer `JSON.parse`. Verifique HMAC com comparação segura, confira a idade do timestamp e deduplique `webhook-id`. Grave o evento antes de responder; devolva `2xx` rapidamente e processe em segundo plano. Não siga redirects de callback.

Entrega ao menos uma vez, com até 8 tentativas e espera crescente. Por isso, o mesmo evento pode chegar repetido. Sem deduplicação, seu sistema pode mostrar resultados ou disparar ações duas vezes mesmo que a Biziva cobre a geração uma só vez.

Sem servidor público para callbacks?

Use polling de `status_url` com espera crescente e limite de tempo na interface. O fim da espera do navegador não deve cancelar ou duplicar silenciosamente o pedido no backend. A tarefa pode concluir mais tarde; mantenha uma tela de histórico para recuperar o resultado.

17 | Gestão, segurança e integrações

Controle da empresa

A Central da empresa em <https://api.biziva.com.br/central> permite gerir bases, agentes, tarefas, rascunhos e controles. A sessão da conta da API é própria. Não suponha que entrar na plataforma Biziva autentique automaticamente o console independente.

```
GET /company-control
GET /evaluations
POST /evaluations
DELETE /evaluations/EVALUATION_ID
```

Pausas gerais, limites por agente e papéis de chave são configurados pela sessão do proprietário na Central. Não tente alterá-los com uma chave comum. Papéis:viewer(leitura), operator(operação limitada) e manager(gestão conforme escopos). O orçamento diário compromete máximos conservadores, não representa só gasto final; o dia usa Brasília.

Casos de avaliação de agentes registram pergunta e trechos esperados. A execução manual paga fica disponível na Central. A verificação é textual, não certificação semântica ou aprovação automática de modelo. Isso não reativa os antigos endpoints públicos de avaliação das entregas.

OAuth e terceiros

Use API key para integração direta. Conexões OAuth só funcionam com cliente previamente cadastrado, escopos e callback aprovado. Não invente client_id, redirect_uri ou parceiro habilitado. Disponibilidade de um fluxo OAuth não comprova que uma ferramenta externa já esteja homologada.

WhatsApp e alertas de conta

O console oferece cadastro de telefone e preferências de avisos. O vínculo e o consentimento pertencem ao titular e podem exigir confirmação pelo próprio WhatsApp. Não use esse recurso como endpoint genérico de disparo para clientes. Chamadas da API, documentos e agentes não enviam mensagens externas automaticamente.

O que registrar no seu backend

Registre conta interna, ID lógico, endpoint, modelo/perfil solicitado, estado, tempo, status HTTP e recibo. Evite gravar prompts e documentos por padrão quando desnecessários. Nunca registre segredo, ticket de sessão, token de terceiro ou Authorization. Exponha ao cliente apenas informações da conta dele.

No painel, diagnóstico e alertas ajudam o suporte. Eles não substituem os logs e limites do integrador. Limites opcionais podem retornar 429 e precisam ser tratados sem loops de retry.

18 | Checklist de entrega e referência

Antes de chamar de integrado

- Conexão testada com GET /models e permissões corretas.
- Chave somente no servidor; nenhum segredo em bundle, Git, logs ou chat da IA.
- Uma operação paga de teste aprovada pelo responsável, com teto explícito.
- ID idempotente persistido e teste de reenvio sem duplicidade.
- 202, polling e resultados incertos tratados sem novo disparo automático.
- Streaming, multipart e downloads testados separadamente, se utilizados.
- Saldo e recibo fracionário exibidos corretamente.
- Arquivos temporários copiados para seu storage quando necessário.
- 401, 402, 403, 409, 423, 429, 502/503 e timeout com mensagens compreensíveis.
- Webhooks com assinatura e deduplicação; ações externas com aprovação.
- Outros provedores e fluxos existentes do seu sistema preservados.

Onde conferir mudanças

Catálogo autenticado:GET /models e /models/{id}. Capacidades:GET /features. Política de preço:GET /pricing. Consumo/saldo:GET /usage. Mídia:GET /videos/models e /audio/models. Documentação:<https://api.biziva.com.br/docs>. Conta:<https://api.biziva.com.br/console>.

Notas desta edição

Este guia substitui o PDF de 14/10/2026. Foram retiradas informações particulares de clientes e recomendações antigas de que todo502 não cobra ou deve ser repetido. Foram acrescentados cobrança fracionária, domínio independente, imagens assíncronas, vídeo com áudio, realtime, bases, DOCX/XLSX, OCR pago, agentes, aprovações, tradução de vídeo, lotes, webhooks e controles da empresa.

Referência verificada nos contratos publicados em 17/10/2026. Recursos podem depender de saldo, escopos, políticas da conta e disponibilidade do fornecedor. A IA que recebe este documento deve consultar o catálogo atual e o projeto antes de implementar. Nenhuma credencial ou configuração particular de cliente está incluída.