Pagaleve In-Store API

Documentação técnica para integração de Parcelamento via Pix em lojas físicas (PDV).

v1.0 Abril 2026 Sandbox disponível

O que é esta integração?

A integração In-Store permite que estabelecimentos físicos ofereçam **Parcelamento via Pix** no caixa. O lojista identifica o cliente pelo CPF, envia os termos de aceite via WhatsApp e exibe um QR Code Pix para pagamento da 1ª parcela. As demais parcelas são cobradas automaticamente pela Pagaleve.

Telas que compõem o fluxo no PDV

ETAPA	ENDPOINT PRINCIPAL	OBJETIVO
1. Autenticação	`POST /v1/authentication`	Obter Bearer Token (JWT) para autorizar as chamadas
2. Análise de crédito	`POST /v1/machine-integration/checkouts`	Coletar CPF e verificar elegibilidade ao Parcelamento via Pix
2.1 Análise de crédito (multi oferta)	`POST /v1/machine-integration/generate-plan-from-simulation`	Quando há mais de uma simulação retornada no passo 2
3. Envio dos termos	`POST /v1/machine-integration/send-terms`	Enviar termos de aceite via WhatsApp ao cliente
4. Pagamento (QR Code)	`GET /v1/checkouts/{id}`	Consultar o checkout para acompanhar aceite, autorização ou conclusão
5. Sucesso	`Webhook / GET /v1/checkouts/{id}`	Confirmar `state = AUTHORIZED/COMPLETED`
6. Criar Pagamento	`POST /v1/payments`	Criar a autorização ou captura do pagamento usando o checkout autorizado

Pré-requisitos

O que você precisa ter antes de iniciar a integração.

Conta e credenciais

É necessário ter uma conta aprovada na Pagaleve. Após aprovação, você receberá:

VARIÁVEL	DESCRIÇÃO	USO
`API_KEY`	Chave pública do lojista	Campo `username` no login
`API_SECRET`	Chave secreta do lojista	Campo `password` no login

VARIÁVEL	DESCRIÇÃO	USO
`BEARER_TOKEN`	JWT gerado pela autenticação	Header `Authorization` em todas as chamadas

Checklist inicial

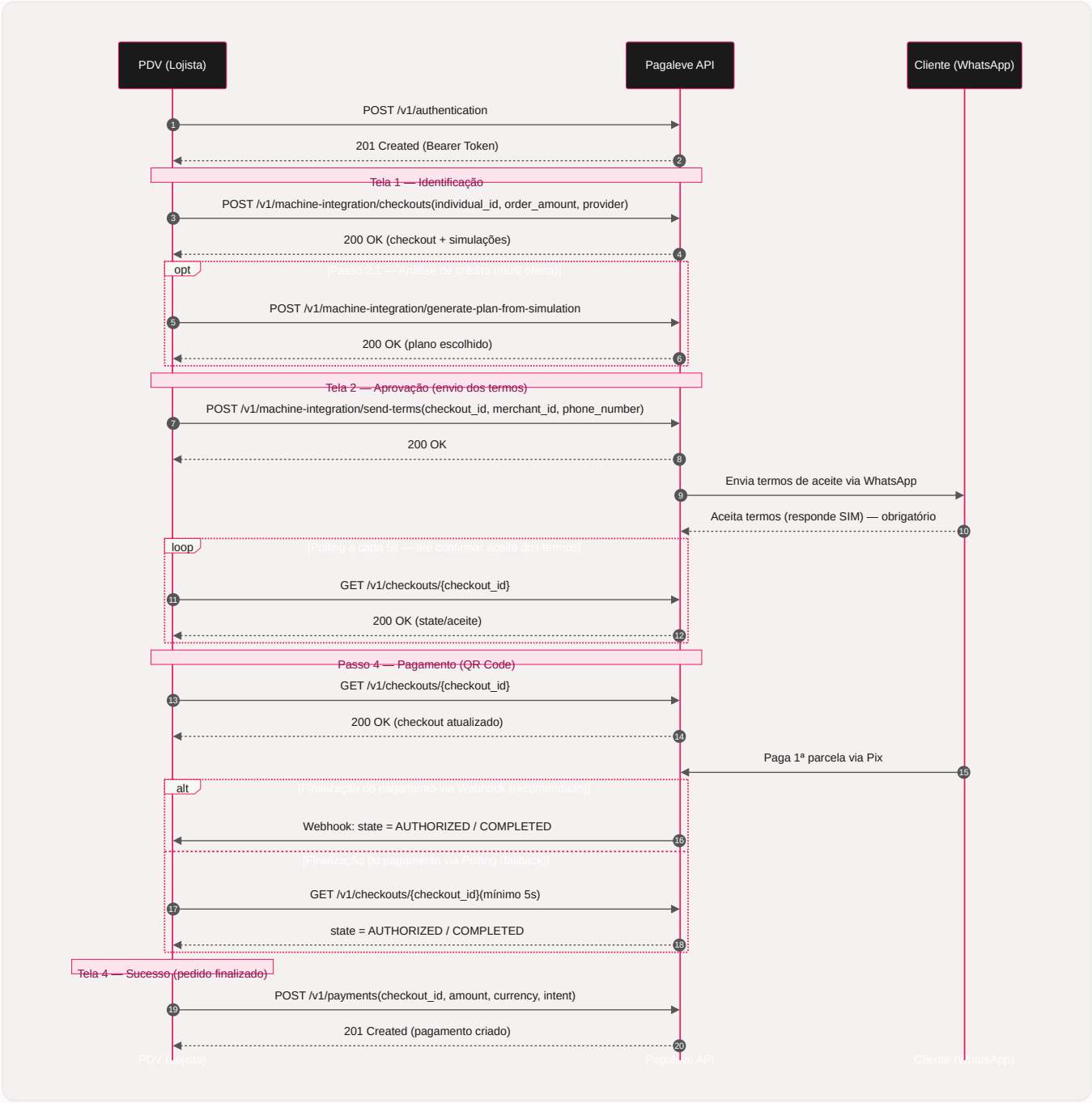
- [ ] Conta aprovada na Pagaleve
- [ ] API Key e API Secret em mãos
- [ ] URL de webhook HTTPS configurada
- [ ] Ambiente sandbox ativo para testes
- [ ] Acesso ao simulador de pagamento Pagaleve

Fluxo completo

Visão sequencial de todas as etapas, da identificação do cliente à confirmação do pagamento Parcelamento via Pix.

Diagrama de sequência

Interação completa entre **PDV**, **Pagaleve** e **Cliente (WhatsApp)** ao longo do fluxo In-Store.



Fluxo passo a passo

1 **Autenticação — obter Bearer Token (JWT)** `POST /v1/authentication` Recolher

- Primeira chamada do fluxo para autenticar o PDV na API Pagaleve.
- Retorna o token Bearer usado nos endpoints seguintes.

↓

2 **Análise de crédito — CPF + valor do pedido** `POST /v1/machine-integration/checkouts` Recolher

- Cria o checkout In-Store e valida a elegibilidade do cliente.

- O checkout_id retornado deve ser usado nos próximos passos.

↓

2.1 **Análise de crédito (multi oferta)** `POST /v1/machine-integration/generate-plan-from-simulation` Recolher

- Usado quando `simulations[]` retorna múltiplas ofertas de parcelamento para o cliente.
- Ao selecionar uma oferta, envie `checkout_id`, `customer_id` e `simulation_id` para gerar o plano definitivo.

↓

3 **Envio dos termos** `POST /v1/machine-integration/send-terms` Recolher

- Envia os termos e condições para o telefone do cliente.
- O cliente deve responder `SIM` no WhatsApp para autorizar o parcelamento.
- Após o envio, use polling em `GET /v1/checkouts/{checkout_id}` a cada 5 segundos até confirmar o aceite.

↓

4 **Pagamento (QR Code)** `GET /v1/checkouts/{id}` Recolher

- Consulta o checkout informado pelo `id`.
- Use para acompanhar o estado do checkout e confirmar quando ele estiver `AUTHORIZED` ou `COMPLETED`.

↓

5 **Sucesso** `Webhook / GET /v1/checkouts/{checkout_id}` Recolher

- Exibir sucesso quando o polling retornar `state === "AUTHORIZED"` ou `state === "COMPLETED"`.
- Use polling com intervalo mínimo de 5 segundos quando o parceiro não consumir webhooks.
- As demais parcelas são cobradas automaticamente via WhatsApp pela Pagaleve.

↓

6 **Criar Pagamento** `POST /v1/payments` Recolher

- Cria a autorização ou captura do pagamento para o checkout informado.
- Use `Idempotency-Key` em retentativas para evitar duplicidade.

Dependência obrigatória: **1 → 2 → (2.1) → 3 → 3.1 (aceite) → 3.2 (polling a cada 5s) → 4 → 5 → 6**. O `checkout_id` retornado em 2 é usado nos passos seguintes. As demais parcelas são cobradas automaticamente via WhatsApp pela Pagaleve.

> **Importante:** não há webhook para criação do checkout nem para aceite dos termos. A criação é confirmada pela resposta do próprio endpoint, e o aceite dos termos deve ser identificado por polling em `GET /v1/checkouts/{checkout_id}`, executado **a cada 5 segundos**. Para a confirmação do pagamento, prefira **Webhook** quando disponível; use polling apenas como fallback.

Passo 1 — Autenticação

Autentica a aplicação parceira e retorna o token Bearer usado nas próximas chamadas.

`POST` `/v1/authentication`

Endpoint no padrão da documentação pública da Pagaleve. Envie as credenciais fornecidas pela Pagaleve no corpo da requisição.

BASE URL — SANDBOX

```
https://sandbox-api.pagaleve.io/v1/authentication
```

HEADERS

```
accept: application/json
content-type: application/json
```

REQUEST BODY

```
{
  "username": "{{API_KEY}}",
  "password": "{{API_SECRET}}"
}
```

CAMPO	TIPO	REQ.	DESCRIÇÃO
username	string	Sim	E-mail ou API key fornecida pela Pagaleve para acesso à API
password	string	Sim	Senha ou API Secret fornecida pela Pagaleve para acesso à API

EXEMPLO CURL

```
curl --request POST \
  --url https://sandbox-api.pagaleve.io/v1/authentication \
  --header 'accept: application/json' \
  --header 'content-type: application/json' \
  --data '{"username":"{{API_KEY}}","password":"{{API_SECRET}}"}'
```

Respostas possíveis

201 Autenticação realizada com sucesso. Use o token retornado como `Authorization: Bearer {{token}}`.	403 Problema no processo de autenticação, como usuário inexistente ou conflito de e-mail e/ou senha.

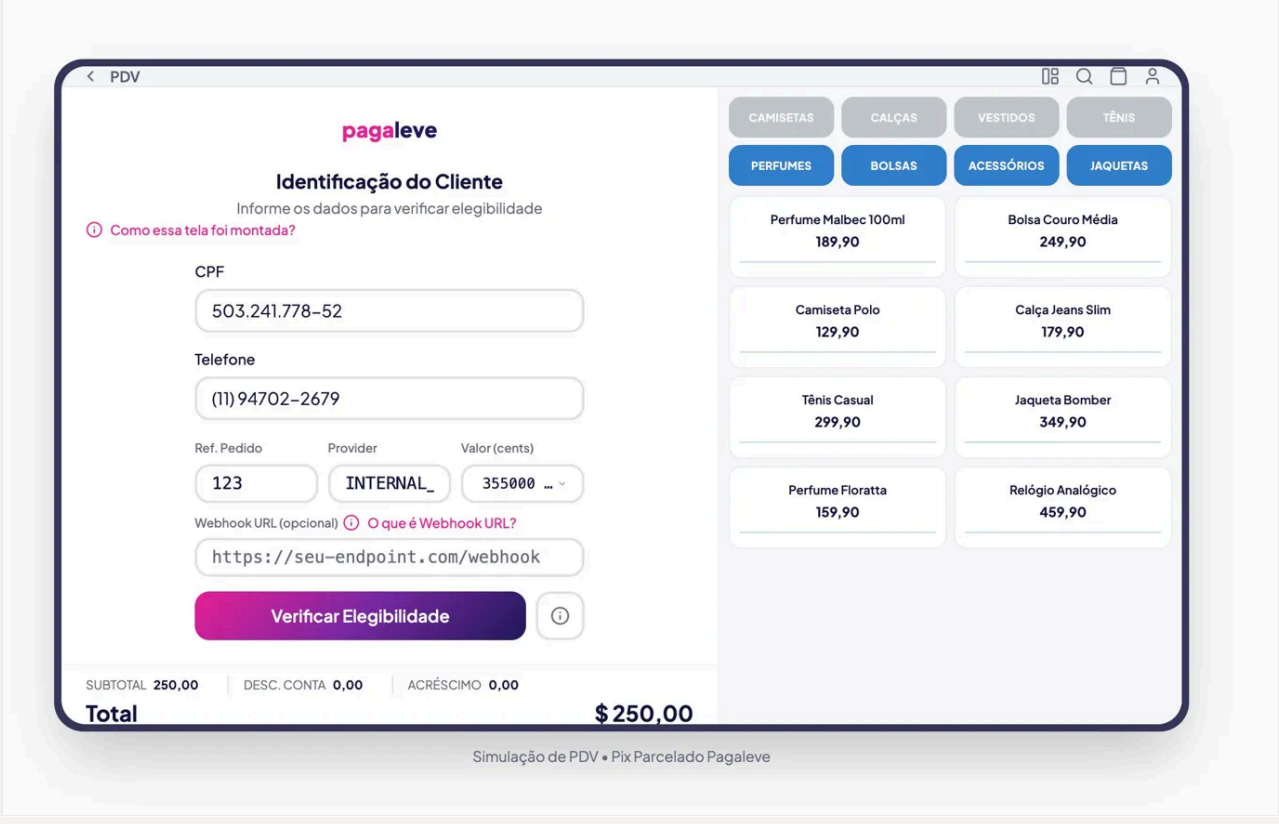
> Armazene o token com segurança e renove a autenticação quando necessário. Todas as próximas chamadas protegidas exigem o header `Authorization`.

Passo 2 — Análise de crédito

Cria o checkout In-Store coletando o CPF do cliente e os dados do pedido para verificar elegibilidade ao Parcelamento via Pix.

TELA 1 — IDENTIFICAÇÃO DO CLIENTE

`POST /v1/machine-integration/checkouts`



Simulação de PDV • Pix Parcelado Pagaleve

Coleta CPF, telefone, referência e valor do pedido. Ao clicar em **Verificar Elegibilidade**, o PDV chama o endpoint para criar o checkout e validar a aprovação do cliente.

`POST` `/v1/machine-integration/checkouts`

> Headers obrigatórios: `Authorization: Bearer {{token}}` e `Idempotency-Key: {{uuid-v4}}`. Em retentativas, reutilize o mesmo Idempotency-Key.

HEADERS OBRIGATÓRIOS

```
Authorization: Bearer {{token}}
Idempotency-Key: {{uuid-v4}}
Content-Type: application/json
```

REQUEST BODY

```
{
  "individual_id":    "50324177852",
  "order_amount":     25000,
  "order_reference":  "123",
  "provider":         "INTERNAL_INVITE",
  "intent":           "CAPTURE",
  "webhook_url":      "https://seu-dominio.com/webhooks/pagaleve"
}
```

CAMPO	TIPO	REQ.	DESCRIÇÃO
individual_id	string	Sim	CPF do cliente (somente dígitos)
order_amount	integer	Sim	Valor do pedido em centavos (R$ 250,00 = 25000)
order_reference	string	Sim	Referência interna do pedido
provider	string	Sim	Provedor de integração (ex: VTEX, INTERNAL_INVITE)
intent	string	Sim	Deve ser enviado como CAPTURE (autoriza e captura automaticamente).
webhook_url	string (URL)	Sim	Endpoint HTTPS do lojista onde a Pagaleve enviará as notificações de status do checkout (pagamento aprovado, expirado, cancelado, etc.).

RESPONSE — OFERTA ÚNICA (200)

```
{
  "checkout": {
    "id":          "chk_a1b2c3d4...",
    "merchant_id": "mer_xyz789",
    "order": {
      "amount":    25000,
      "reference": "123"
    },
    "plan": {
      "current_installment": {
        "qr_code": { "code": "00020126580014br.gov.bcb.pix..." }
      }
    }
  },
  "simulations": [{ "installments": 6, "amount": 62721 }]
}
```

Respostas possíveis

200 **1 simulação** — oferta única. Próximo passo: tela de Aprovação (send-terms).

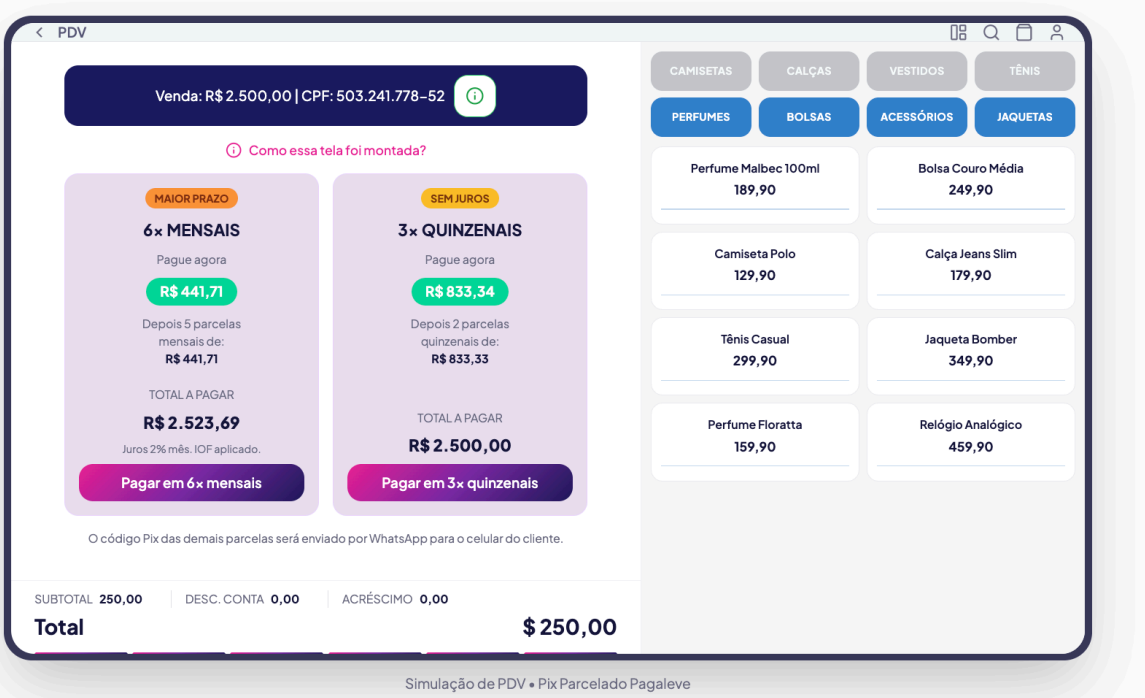
200 **Múltiplas simulações** — exibir tela de multi-oferta para o cliente escolher.

400 `DecisionPreApproveException` — cliente não aprovado.

Passo 2.1 — Análise de crédito (multi oferta)

Quando o passo 2 retorna múltiplas simulações, o cliente escolhe uma oferta de parcelamento e este endpoint gera o plano definitivo.

TELA — MULTI-OFERTA `simulations[]`



Simulação de PDV • Pix Parcelado Pagaleve

Cada card de oferta é montado a partir de um item do array **simulations[]** retornado no endpoint de Análise de crédito.

📦 Como montar a tela de Multi-Oferta

Os dados vêm do array `simulations` retornado na resposta do endpoint de criação de checkout In-Store (`POST /v1/machine-integration/checkouts`). Cada item em `simulations[]` representa uma oferta de parcelamento disponível para o cliente.

CAMPO	DESCRIÇÃO
`id`	Identificador da simulação usado na confirmação.
`number_of_installments`	Quantidade de parcelas.

CAMPO	DESCRIÇÃO
`rate`	Taxa de juros mensal. Quando for `0`, a oferta é sem juros.
`amount`	Valor total a pagar, com juros quando houver.
`order_amount`	Valor original do pedido.
`current_installment.amount`	Valor da 1ª parcela exibida como “Pague agora”.
`installments[]`	Detalhamento de todas as parcelas.
`iof`	Valor do IOF quando há juros.

POST `/v1/machine-integration/generate-plan-from-simulation`

Ao selecionar uma oferta, chame este endpoint com `checkout_id`, `customer_id`, `simulation_id` e `intent`.

REQUEST BODY

```
{
  "checkout_id":   "chk_a1b2c3d4...",
  "customer_id":   "cus_12345...",
  "simulation_id": "sim_12345...",
  "intent":        "CAPTURE"
}
```

EXEMPLO DE LEITURA DAS SIMULAÇÕES

```
response.simulations.forEach(sim => {
  console.log(sim.number_of_installments + "x");
  console.log("Taxa:", sim.rate + "%");
  console.log("1ª parcela:", sim.current_installment.amount);
});

// Ao selecionar
const chosen = simulations[selectedIndex];
generatePlan({ simulation_id: chosen.id });
```

200 OK

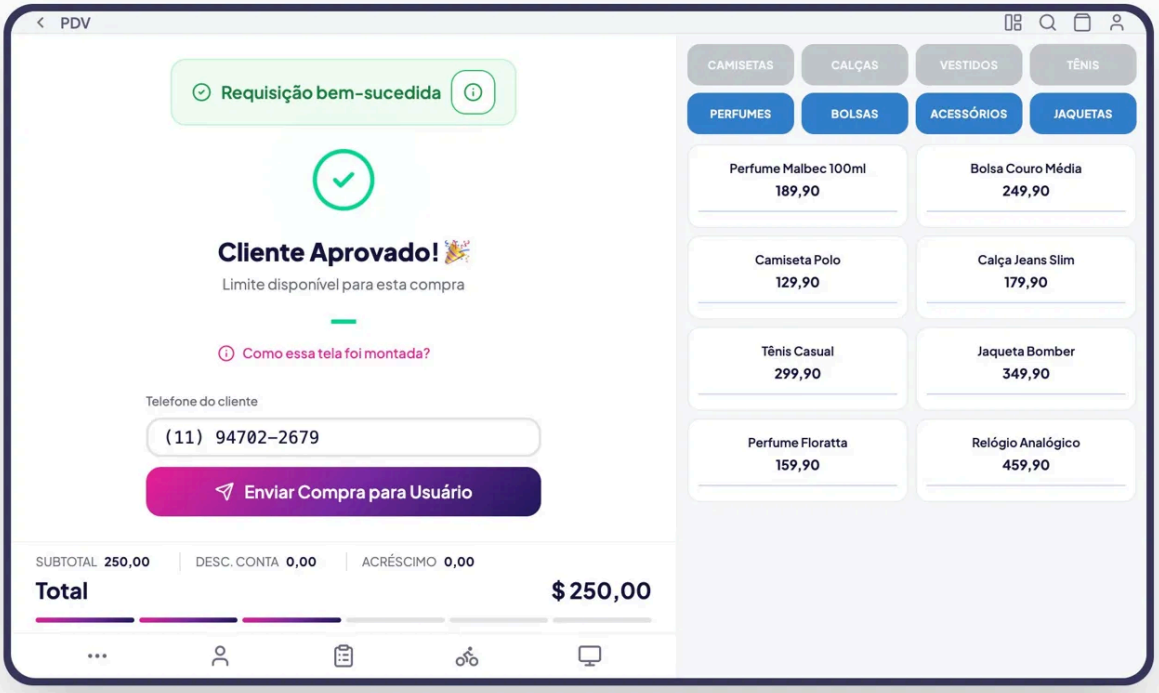
```
{
  "current_installment": {
    "qr_code": {
      "code": "00020126580014br.gov.bcb.pix..."
    }
  },
  "installments": 6,
```

```
  "amount":       62721
}
```

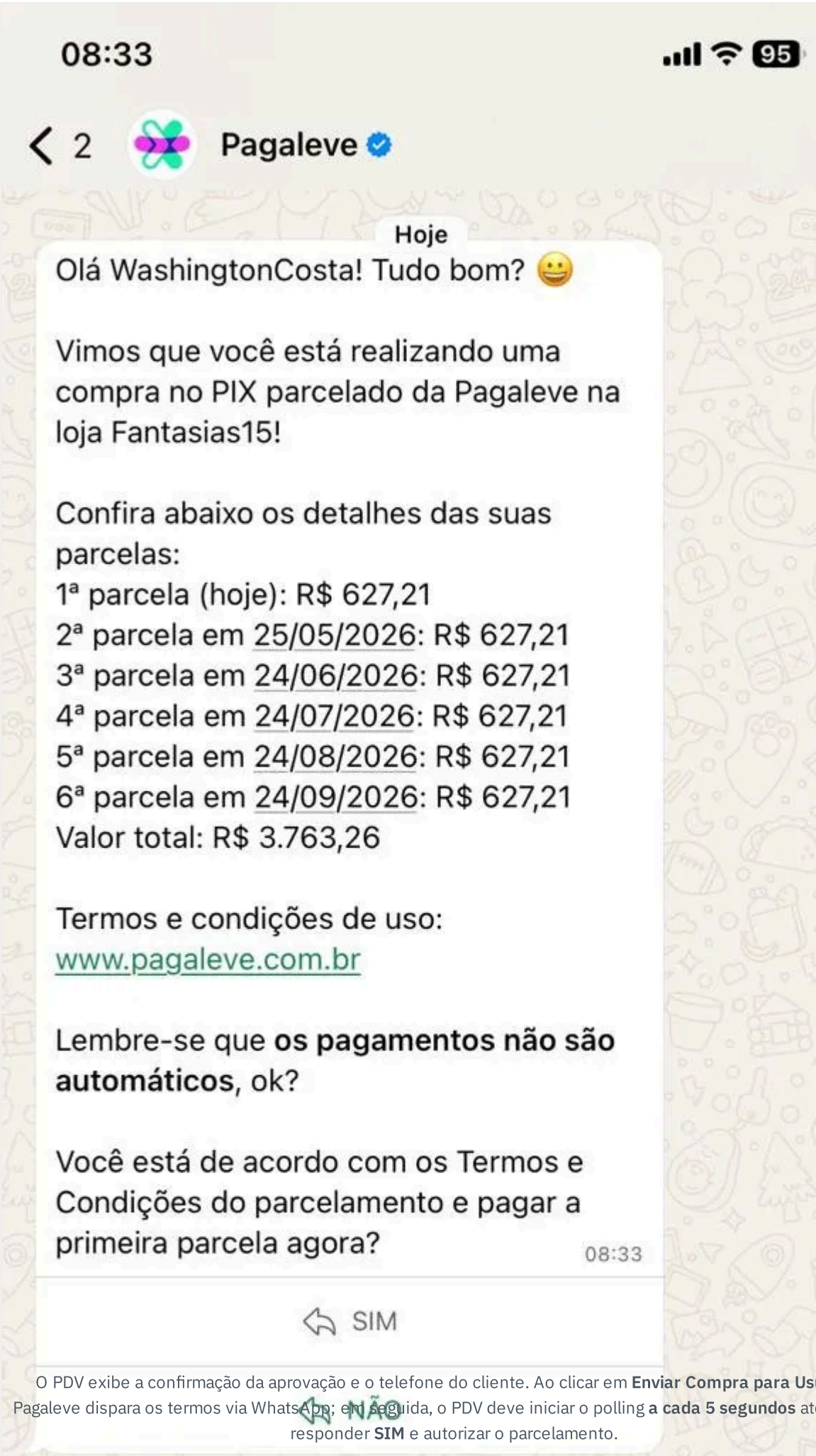
Passo 3 — Envio dos termos

Envia os termos de aceite ao celular do cliente via WhatsApp/SMS. O cliente precisa responder "SIM" para autorizar o parcelamento.

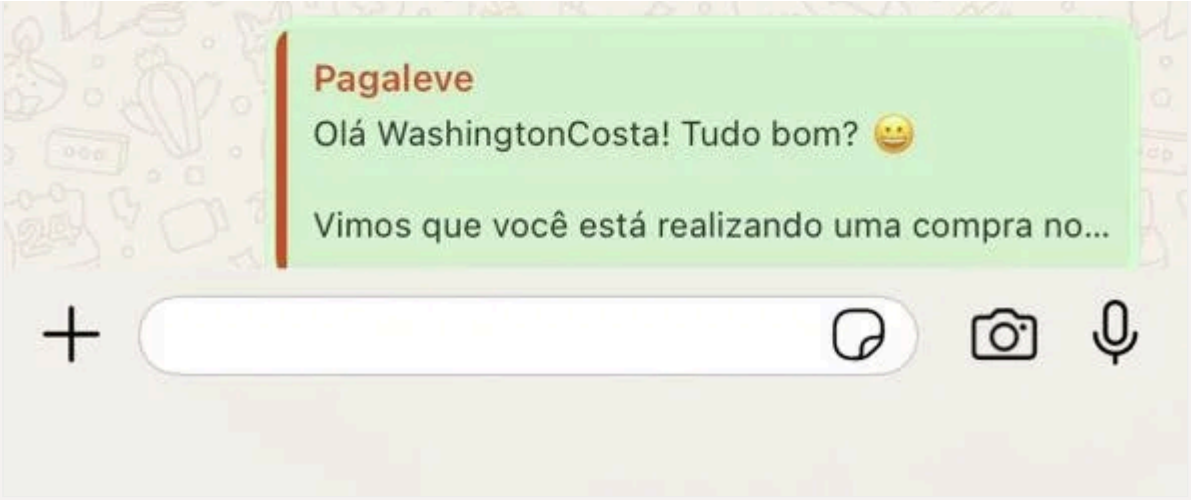
TELA 2 — APROVAÇÃO · ENVIAR COMPRA PARA O CLIENTE `POST /v1/machine-integration/send-terms`



Simulação de PDV • Pix Parcelado Pagaleve



O PDV exibe a confirmação da aprovação e o telefone do cliente. Ao clicar em **Enviar Compra para Usuário**, a Pagaleve dispara os termos via WhatsApp; em seguida, o PDV deve iniciar o polling **a cada 5 segundos** até o cliente responder **SIM** e autorizar o parcelamento.



> O valor aprovado a ser exibido na tela vem de `checkout.order.amount` da resposta do passo 2.

`POST` `/v1/machine-integration/send-terms`

Disparado ao clicar em "Enviar Compra para Usuário" na tela de Aprovação.

REQUEST BODY

```
{
  "checkout_id":  "chk_a1b2c3d4...",
  "merchant_id":  "mer_xyz789",
  "phone_number": "+5511947022679"
}
```

CAMPO	TIPO	REQ.	DESCRIÇÃO
`checkout_id`	string	Sim	ID do checkout criado no passo 2
`merchant_id`	string	Sim	ID do merchant (vem da resposta do checkout)
`phone_number`	string	Sim	Telefone com prefixo internacional (ex: +5511999999999)

`200 OK`

> Em caso de sucesso (200), o cliente recebe os termos no celular via WhatsApp/SMS, com detalhamento das parcelas, datas de cobrança e valor total. Após essa chamada, o PDV deve consultar `GET /v1/checkouts/{checkout_id}` **a cada 5 segundos** até identificar o aceite. Ao responder **SIM**, o fluxo segue para o pagamento da 1ª parcela via QR Code Pix.

Como identificar que o cliente aceitou os termos?

Após o envio dos termos, o PDV deve consultar o checkout **a cada 5 segundos** para verificar se o cliente respondeu **SIM** no WhatsApp. Não existe webhook para aceite dos termos; por isso, essa validação acontece exclusivamente por polling.

`GET` `/v1/checkouts/{checkout_id}`

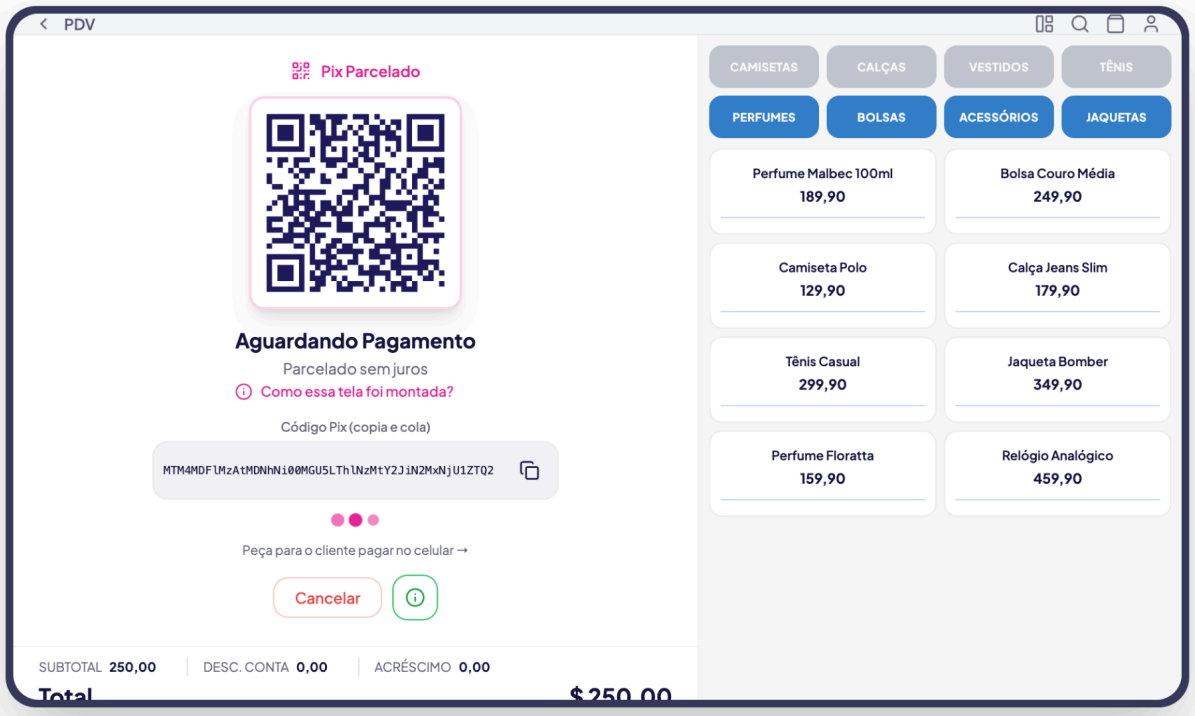
Execute a consulta **a cada 5 segundos** até identificar que os termos foram aceitos e o fluxo pode avançar para o QR Code.

> Nunca faça polling em intervalo inferior a 5 segundos.

Passo 4 — Pagamento (QR Code)

Consulta os dados de um checkout pelo ID para acompanhar o estado do fluxo.

TELA 3 — QR CODE · AGUARDANDO PAGAMENTO `GET /v1/checkouts/{id}`



Simulação de PDV • Pix Parcelado Pagaleve

O PDV renderiza o **QR Code Pix** e o código copia-e-cola retornados pelo checkout de pagamento. Os dados da tela vêm do pedido, do cliente e do plano confirmado nos passos anteriores.

`GET` `/v1/checkouts/{id}`

Endpoint no padrão da documentação pública de consulta de checkout. Use para consultar o checkout informado e acompanhar estados como aceite, autorização ou conclusão.

BASE URL — SANDBOX

```
https://sandbox-api.pagaleve.io/v1/checkouts/{id}
```

HEADERS

```
Authorization: Bearer {{token}}
Subsidiary-Id: {{subsidiary_id}}
accept: application/json
```

HEADER	TIPO	REQ.	DESCRIÇÃO
Authorization	string	Sim	Token de autorização Bearer
Subsidiary-Id	string	Não	Para uma holding consultar dados de uma subsidiária, esse campo deve ser preenchido com o ID da subsidiária.

PARÂMETRO	TIPO	REQ.	DESCRIÇÃO
id	string	Sim	ID do checkout consultado.

EXEMPLO CURL

```
curl --request GET \
  --url https://sandbox-api.pagaleve.io/v1/checkouts/{{checkout_id}} \
  --header 'accept: application/json' \
  --header 'Authorization: Bearer {{token}}'
```

Respostas possíveis

`200` Checkout encontrado e retornado com sucesso.

`404` Checkout com o ID informado não encontrado.

Acompanhamento do pagamento

A finalização deve ocorrer preferencialmente via **Webhook**, informado em `webhook_url`. O polling em `GET /v1/checkouts/{checkout_id}` deve ser usado apenas como fallback e nunca com intervalo inferior a 5 segundos.

Simulação de pagamento (sandbox)

> Para testar, copie o código Pix exibido no PDV e cole na aba **"Simular Pagamento"** dentro das configurações (⚙) do ambiente de staging. Isso confirma o pagamento sem necessidade de Pix real.

SANDBOX — SIMULAR PAGAMENTO `Staging`

Configurações

Gerencie credenciais e operações de sandbox.

Credenciais | Simular Pagamento | Remover Usuário

Simular Pagamento

Cole o código do QR Code (PIX) para simular o pagamento no ambiente de staging.

CÓDIGO QR CODE (PIX)

```
MDFlMzAtMDNhNi00MGU5LThlNzMtY2JiN2MxNjU1ZTQ2
```

Simular Pagamento

Configurações

Gerencie credenciais e operações de sandbox.

Credenciais | Simular Pagamento | Remover Usuário

Simular Pagamento

Cole o código do QR Code (PIX) para simular o pagamento no ambiente de staging.

CÓDIGO QR CODE (PIX)

```
MTM4MDFlMzAtMDNhNi00MGU5LThlNzMtY2JiN2MxNjU1
```

Pagamento simulado com sucesso!


```
{
  "id": "aa4838e1-d405-4b8b-a0b0-033a65dc79a4",
  "amount": 62721,
  "individualId": "undefined",
  "timestamp": "2026-04-24T11:43:31.472Z",
  "updatedAt": "2026-04-24T11:43:31.754Z",
  "endToEndId": "13801e30-03a6-40e9-8e73-cbb7c1655e46",
  "qrCodeExternalId": "13801e30-03a6-40e9-8e73-cbb7c1655e46",
  "externalId": "13801e30-03a6-40e9-8e73-cbb7c1655e46",
  "name": "Milionario e José Rico",
```


Simular Pagamento

Cole o código Pix copiado da Tela 3 e clique em **Simular Pagamento**. Em sucesso, o webhook confirma o pagamento e o PDV avança para a Tela 4 — Sucesso.

Passo 5 — Sucesso

Exibida quando o polling confirma `state = AUTHORIZED` ou `state = COMPLETED`.

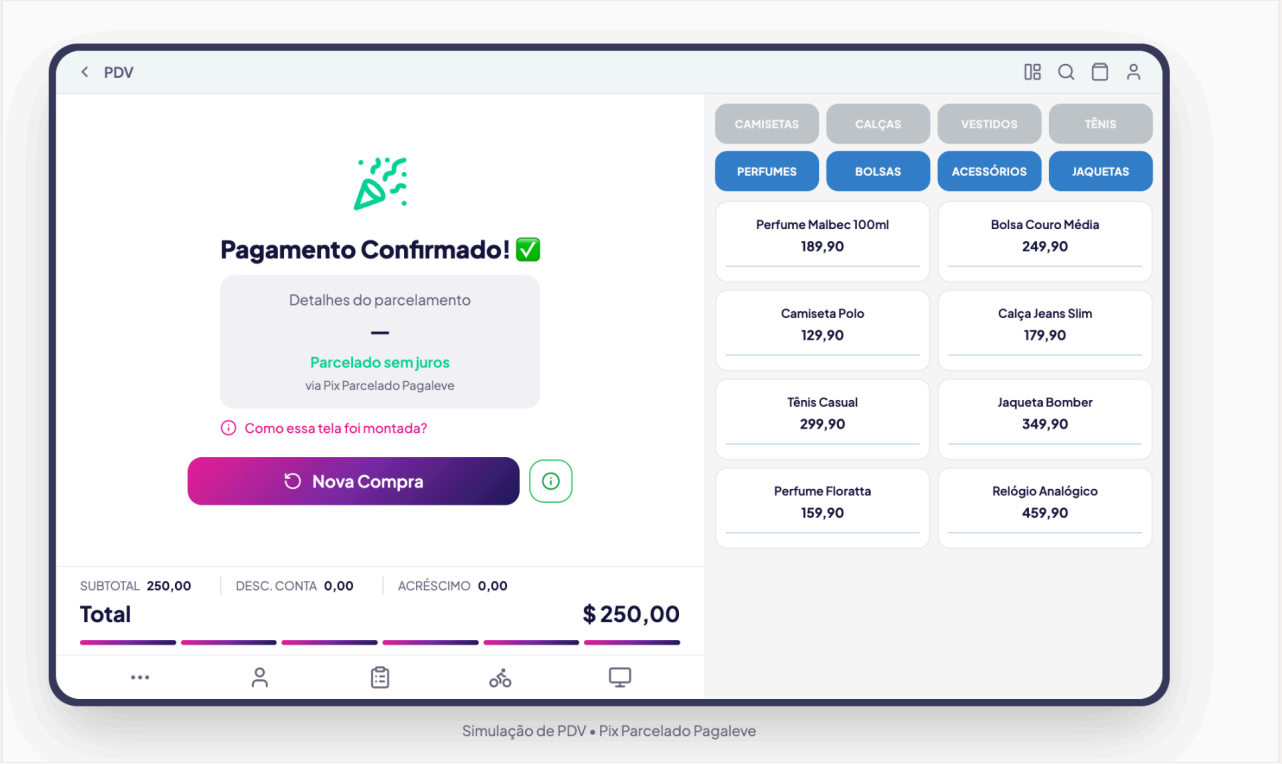
📦 Como montar a tela de Sucesso

Detalhes de integração para parceiros.

Quando exibir?

Exiba esta tela quando o polling de `GET /v1/checkouts/{checkout_id}` retornar `state === "AUTHORIZED"` ou `state === "COMPLETED"`.

TELA 4 — PAGAMENTO CONFIRMADO `state = AUTHORIZED / COMPLETED`



Simulação de PDV • Pix Parcelado Pagaleve

Tela final exibida quando o pagamento da 1ª parcela é confirmado. O valor total e a quantidade de parcelas são lidos de `checkout.order.amount` e `checkout.ccb_terms_and_condition.numberOfInstallments`.

Dados exibidos

CAMPO	CAMINHO NA RESPOSTA	USO
Valor total	`checkout.order.amount`	Valor total da compra (em centavos)
Número de parcelas	`checkout.ccb_terms_and_condition.numberOfInstallments`	Quantidade de parcelas do plano

> **Fluxo concluído.** Neste ponto, o pagamento da 1ª parcela foi confirmado. As demais parcelas serão cobradas

automaticamente via WhatsApp.

Passo 6 — Criar Pagamento

Cria a autorização ou captura do pagamento a partir do checkout autorizado.

`POST` `/v1/payments`

Use este endpoint para criar o pagamento informando o pedido, valor, moeda e intenção da operação.

BASE URL — SANDBOX

```
https://sandbox-api.pagaleve.io/v1/payments
```

HEADERS

```
Authorization: Bearer {{token}}
Idempotency-Key: {{uuid-v4}}
Subsidiary-Id: {{subsidiary_id}}
accept: application/json
content-type: application/json
```

HEADER	TIPO	REQ.	DESCRIÇÃO
Authorization	string	Sim	Token de autorização Bearer.
Idempotency-Key	string	Não	Chave para segurança em retentativas, evitando executar a mesma operação duas vezes.
Subsidiary-Id	string	Não	Para holdings consultarem dados de uma subsidiária, informe o ID da subsidiária.

REQUEST BODY

```
{
  "amount": 25000,
  "checkout_id": "chk_a1b2c3d4...",
  "currency": "BRL",
  "intent": "AUTH",
  "reference": "PEDIDO-123"
}
```

CAMPO	TIPO	REQ.	DESCRIÇÃO
amount	number	Sim	Valor em centavos. Deve ser maior ou igual a 1.
checkout_id	string	Sim	ID do pedido.
currency	string	Sim	Moeda utilizada. Valores permitidos: BRL ou USD.

CAMPO	TIPO	REQ.	DESCRIÇÃO
intent	string	Sim	AUTH para autorização com captura posterior, ou CAPTURE para autorizar e capturar automaticamente.
reference	string	Não	ID de referência utilizado na raiz do checkout.

EXEMPLO CURL

```
curl --request POST \
  --url https://sandbox-api.pagaleve.io/v1/payments \
  --header 'accept: application/json' \
  --header 'content-type: application/json' \
  --header 'Authorization: Bearer {{token}}' \
  --header 'Idempotency-Key: {{uuid-v4}}' \
  --data '{{request_body}}'
```

Respostas possíveis

- **201** Pagamento criado com sucesso.
- **404** Checkout informado não encontrado.
- **409** Checkout já foi iniciado.
- **412** Checkout já foi completado ou não está autorizado.
- **422** Valor do pagamento diferente do valor autorizado.

Passo 6.1 — Reembolsar Pagamento Capturado

Reembolsa total ou parcialmente um pagamento já capturado. Suporta múltiplos reembolsos até o valor total capturado.

POST `/v1/payments/{id}/refund`

Use este endpoint para reembolsar um pagamento capturado. Podem ser feitos reembolsos totais, parciais ou múltiplos, respeitando o valor total já capturado.

BASE URL — SANDBOX

```
https://sandbox-api.pagaleve.io/v1/payments/{id}/refund
```

HEADERS

```
Authorization: Bearer {{token}}
Idempotency-Key: {{uuid-v4}}
Subsidiary-Id: {{subsidiary-id}} (opcional)
```

HEADER	TIPO	REQ.	DESCRIÇÃO
Authorization	string	Sim	Token Bearer.
Idempotency-Key	string	Não	Evita reprocessar a mesma operação em retentativas.
Subsidiary-Id	string	Não	Obrigatório apenas quando uma holding consulta em nome de uma subsidiária.

PARÂMETRO	TIPO	REQ.	DESCRIÇÃO
id	string	Sim	ID do pagamento capturado que será reembolsado.

REQUEST BODY

```
{
  "amount": 2000,
  "description": "Motivo do reembolso",
  "reference": "ref-loja-123",
  "reason": "REQUESTED_BY_CUSTOMER"
}
```

CAMPO	TIPO	REQ.	DESCRIÇÃO
amount	number	Sim	Valor a reembolsar, em centavos.
description	string	Não	Observações sobre o reembolso.
reference	string	Não	ID do reembolso vindo do lojista.
reason	string (enum)	Não	DUPLICATE, FRAUDULENT ou REQUESTED_BY_CUSTOMER.

EXEMPLO CURL

```
curl --request POST \
  --url https://sandbox-api.pagaleve.io/v1/payments/{id}/refund \
  --header 'Authorization: Bearer {{token}}' \
  --header 'Content-Type: application/json' \
  --data '{"amount": 2000, "reason": "REQUESTED_BY_CUSTOMER"}'
```

RESPOSTA DE SUCESSO (200)

```
{
  "amount": 2000,
  "id": "3c0db7b0-c145-4af6-aecd-82fb964665a4",
  "timestamp": "2022-12-09T13:36:00.489Z",
  "reference": "PD2L0046"
}
```

Respostas possíveis

200 Reembolso processado com sucesso.

400 Requisição inválida.

404	Pagamento não encontrado.
409	Já existe um reembolso em processamento para esse payment_id ou para essa idempotency_key.
412	Reembolso não permitido para o status atual do pagamento.

> Para consultar os reembolsos já realizados de um pagamento, use o endpoint complementar `GET /payments/{id}/refunds`, que retorna a lista de reembolsos associados ao `payment_id`.

Passo 7 — Listar Subsidiárias

Lista as subsidiárias associadas à holding autenticada.

`GET` `/v1/merchant-search/list-subsidiaries`

Use este endpoint para listar todas as subsidiárias vinculadas à holding autenticada, com filtros opcionais de período e paginação.

BASE URL — SANDBOX

```
https://sandbox-api.pagaleve.io/v1/merchant-search/list-subsidiaries
```

HEADERS

```
Authorization: Bearer {{token}}
```

HEADER	TIPO	REQ.	DESCRIÇÃO
Authorization	string	Sim	Token Bearer.

PARÂMETRO	TIPO	REQ.	DESCRIÇÃO
page	integer	Não	Número de página para iniciar (>= 0).
size	integer	Não	Número máximo de itens a serem retornados (limitado em 200).
start_date	string	Não	Data inicial no formato ISO (UTC-0).
end_date	string	Não	Data final no formato ISO (UTC-0).
timezone	string	Não	Timezone usado para o filtro de datas.

EXEMPLO CURL

```
curl --request GET \
  --url 'https://sandbox-api.pagaleve.io/v1/merchant-search/list-subsidiaries?page=0&size=20' \
```

```
  --header 'Authorization: Bearer {{token}}'
```

RESPOSTA DE SUCESSO (200)


```
{
  "items": [
    {
      "id": "bd0e0361-8df5-4888-bd58-72ef786ed9b4",
      "business_name": "Exemplo Store",
      "business_number": "12345678000190"
    }
  ]
}
```


Respostas possíveis

200	Lista de subsidiárias retornada com sucesso.
400	O merchant autenticado não é uma holding company (IllegalHoldingCompanyStateException).
404	Merchant com o id informado não encontrado (MerchantNotFoundException).

> Este endpoint só retorna resultados quando o merchant autenticado é do tipo holding company. Para consultar dados de uma subsidiária específica em outras chamadas, utilize o header `Subsidiary-Id` com o id retornado aqui.

Mapeamento técnico

Todos os endpoints em ordem, com dependências e obrigatoriedade.

#	ETAPA	MÉTODO	ENDPOINT	DEPENDE DE	REQ.
1	Autenticação	POST	/v1/authentication	—	Sim
2	Análise de crédito	POST	/v1/machine-integration/checkouts	Token #1	Sim
2.1	Análise de crédito (multi oferta)	POST	/v1/machine-integration/generate-plan-from-simulation	checkout_id #2	Condicional
3	Envio dos termos	POST	/v1/machine-integration/send-terms	checkout_id + merchant_id	Sim
3.1	Aceite + polling dos termos a cada 5s	GET	/v1/checkouts/{checkout_id}	send-terms #3	Sim

#	ETAPA	MÉTODO	ENDPOINT	DEPENDE DE	REQ.
4	Pagamento (QR Code)	GET	/v1/checkouts/{id}	Aceite #3.1	Sim
5	Sucesso	—	inbound lojista (webhook_url)	Config no #4	Preferencial
5b	Polling de fallback	GET	/v1/checkouts/{checkout_id}	checkout #4	Fallback
6	Criar Pagamento	POST	/v1/payments	checkout autorizado #5	Sim

Ambientes & URLs

URL base oficial dos endpoints Pagaleve. Sempre use sandbox para desenvolvimento e testes.

API Pagaleve

SANDBOX

`https://sandbox-api.pagaleve.io`

Use para desenvolvimento, homologação e simulação de pagamento

PRODUÇÃO

`https://api.pagaleve.com.br`

Somente após validação em sandbox

Exemplo de URL completa

```
POST https://sandbox-api.pagaleve.io/v1/machine-integration/checkouts
```

HTTP status codes

200 **Success** — Success

400 **Bad Request** — Requisição inválida

401 **Unauthorized** — Falha na autenticação

403 **Forbidden** — Sua aplicação não está autorizada a acessar o recurso

404 **Not Found** — Requisição válida, mas o item não foi encontrado

409 **Conflict** — Solicitação aceita, mas em conflito com as regras de negócios

500 **Internal Server Error** — Um erro interno ocorreu

Webhook

Notificações assíncronas enviadas pela Pagaleve à URL informada em `webhook_url` no checkout de pagamento.

> Direção: Pagaleve → Servidor do lojista (POST para a `webhook_url` enviada no `/v1/checkouts`). Use webhook para confirmação de pagamento; o aceite dos termos deve ser verificado por polling.

PAYLOAD RECEBIDO `INFERIDO`

```
{
  "event":           "CHECKOUT_AUTHORIZED",
  "checkout_id":     "chk_a1b2c3d4...",
  "state":           "AUTHORIZED",
  "order_amount":    25000,
  "order_reference": "123",
  "individual_id":   "50324177852",
  "timestamp":       "2026-04-24T11:43:31.472Z"
}
```

Eventos disponíveis

EVENTO	QUANDO
`CHECKOUT_AUTHORIZED`	1ª parcela paga via Pix (`state = AUTHORIZED`)
`CHECKOUT_COMPLETED`	Fluxo concluído (`state = COMPLETED`)
`CHECKOUT_CANCELLED`	Cancelado pelo lojista ou cliente
`CHECKOUT_EXPIRED`	QR Code expirou sem pagamento

> Não há evento de webhook para checkout criado nem para aceite de termos nesta jornada. Para saber se o checkout foi criado, use a própria resposta da criação; para saber se os termos foram aceitos, use polling em `GET /v1/checkouts/{checkout_id}`.

> Responda sempre com HTTP 200 em menos de 5 segundos. Processe de forma assíncrona. O mesmo evento pode ser entregue mais de uma vez — implemente idempotência.

Checklist go-live

Itens obrigatórios antes de ativar em produção.

- [ ] Credenciais de produção (API_KEY / API_SECRET) configuradas em variáveis de ambiente
- [ ] Webhook HTTPS configurado e enviado em `webhook_url` no checkout de pagamento
- [ ] Renovação automática do Bearer Token antes dos 60 min

- [ ] Idempotency-Key único (UUID v4) por chamada de `/checkouts`
- [ ] Tratamento dos estados: CREATED, PENDING, AUTHORIZED, COMPLETED, CANCELLED, EXPIRED
- [ ] Tela de multi-oferta implementada para resposta com múltiplas simulações
- [ ] Polling do aceite dos termos a cada 5s, com timeout máximo configurado
- [ ] Tratamento de `DecisionPreApproveException` (cliente não aprovado)
- [ ] Telefone do cliente formatado com prefixo `+55` antes do send-terms
- [ ] QR Code Pix renderizado a partir de `qr_code.code` com botão "copia e cola"
- [ ] Logs de todas as chamadas para auditoria
- [ ] Testes no sandbox usando "Simular Pagamento" das configurações