

Documentación Integración Khipu a través de llamadas a la API

Crear cobro

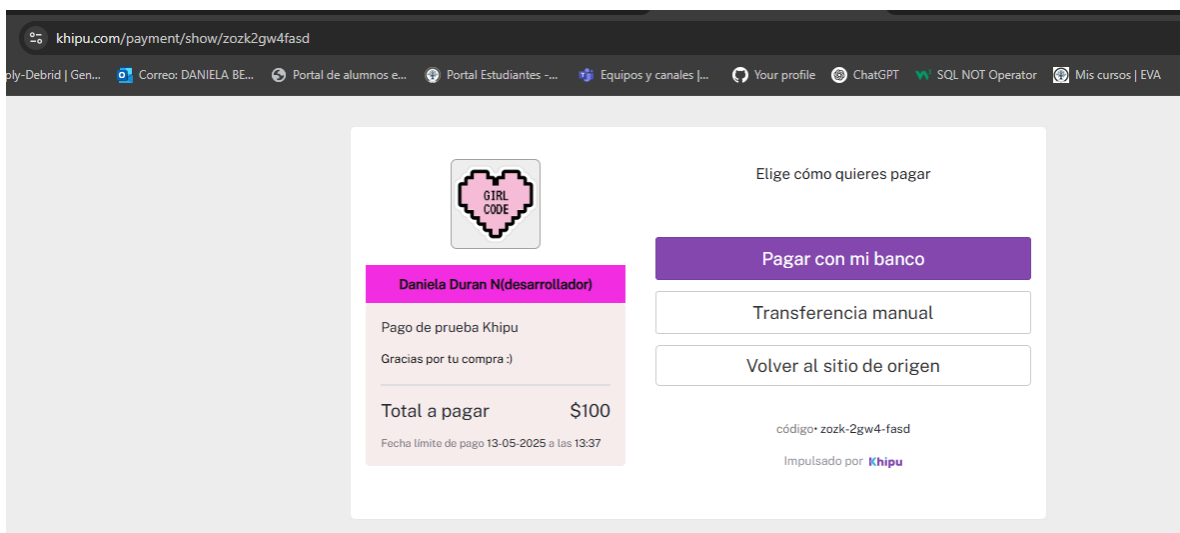
```
crear_cobro.py X
crear_cobro.py > ...
1  import requests
2
3  # Mis credenciales reales
4  API_KEY = 'f3df785a-3e8d-4070-a114-27abe464f270'
5  RECEIVER_ID = '497950'
6
7  url = 'https://payment-api.khipu.com/v3/payments'
8
9  headers = {
10     'x-api-key': API_KEY,
11     'Content-Type': 'application/json'
12 }
13
14 data = {
15     "receiver_id": RECEIVER_ID,
16     "subject": "Pago de prueba Khipu",
17     "amount": 100,
18     "currency": "CLP",
19     "transaction_id": "pedido-123456",
20     "payer_email": "cliente@correo.com",
21     "return_url": "https://tusitio.com/exito",
22     "cancel_url": "https://tusitio.com/cancelado",
23     "notify_url": "https://tusitio.com/webhook",
24     "custom": "Este es un pago de prueba",
25     "body": "Gracias por tu compra :)"
26 }
27
28 response = requests.post(url, headers=headers, json=data)
29
30 print("Código de estado HTTP:", response.status_code)
31 print("Respuesta del servidor:", response.text)
32
```

Creación de pago. Código exitoso

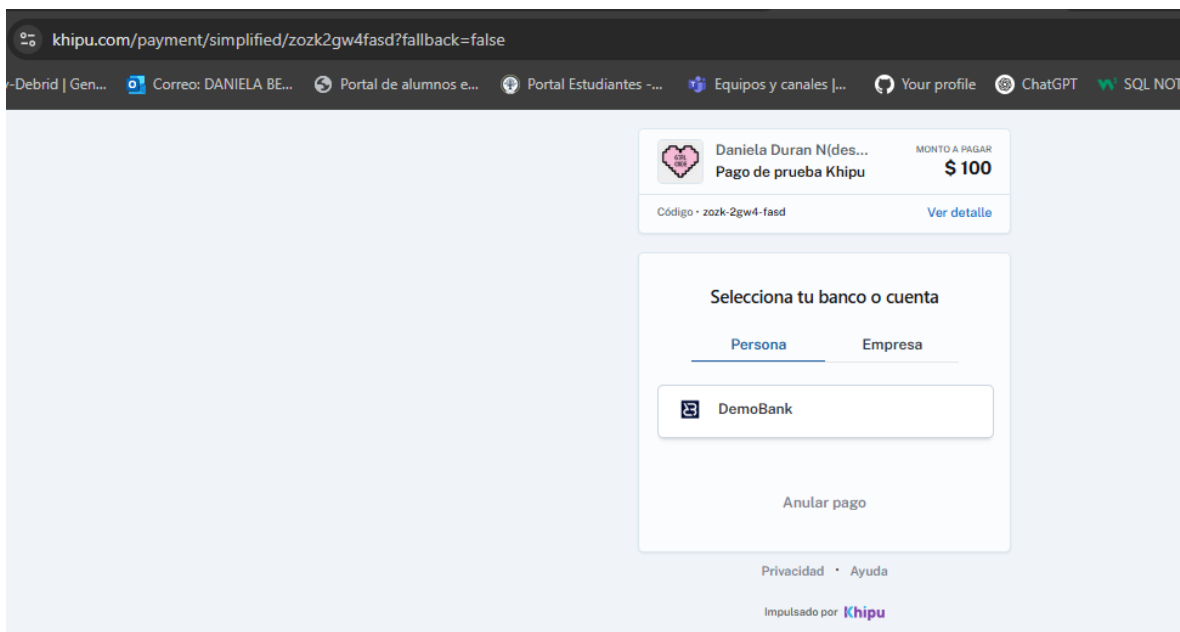
```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Código de estado HTTP: 403
Respuesta del servidor:
Respuesta del servidor:
PS C:\Users\Usuario\Desktop\Khipu_testing> python crear_cobro.py
PS C:\Users\Usuario\Desktop\Khipu_testing> python crear_cobro.py
Código de estado HTTP: 200
Código de estado HTTP: 200
Respuesta del servidor: {"app_url":"khipu://pos/zozk2gw4fasd","payment_id":"zozk2gw4fasd","payment_url":"https://khipu.com/payment/info/zozk2gw4fasd","ready_for_termin
al":false,"simplified_transfer_url":"https://app.khipu.com/payment/simplified/zozk2gw4fasd","transfer_url":"https://khipu.com/payment/manual/zozk2gw4fasd"}
PS C:\Users\Usuario\Desktop\Khipu_testing> []
```

Probamos URL devuelta por la terminal

<https://khipu.com/payment/info/zozk2gw4fasd>




Opción pagar con mi banco:



Elegimos opción demoBank nos redirige solicitud de credenciales para ingresar a DemoBank:

khipu.com/payment/simplified/zozk2gw4fasd?fallback=false

y-Debrid | Gen... Correo: DANIELA BE... Portal de alumnos e... Portal Estudiantes ~... Equipos y canales [...] Your profile ChatGPT SQL NO



Daniela Duran N(des...

MONTO A PAGAR

Pago de prueba Khípu

\$ 100

Código • zozk-2gw4-fasd

Ver detalle

Ingresar a tu banco

DemoBank

Usa la clave de DemoBank, no la de tu App.

RUT

Ej. 12.345.678-K

Clave

Clave DemoBank

Usa 1234

Recordar credenciales

Al continuar tu pago estás aceptando las condiciones de uso del servicio de Khípu

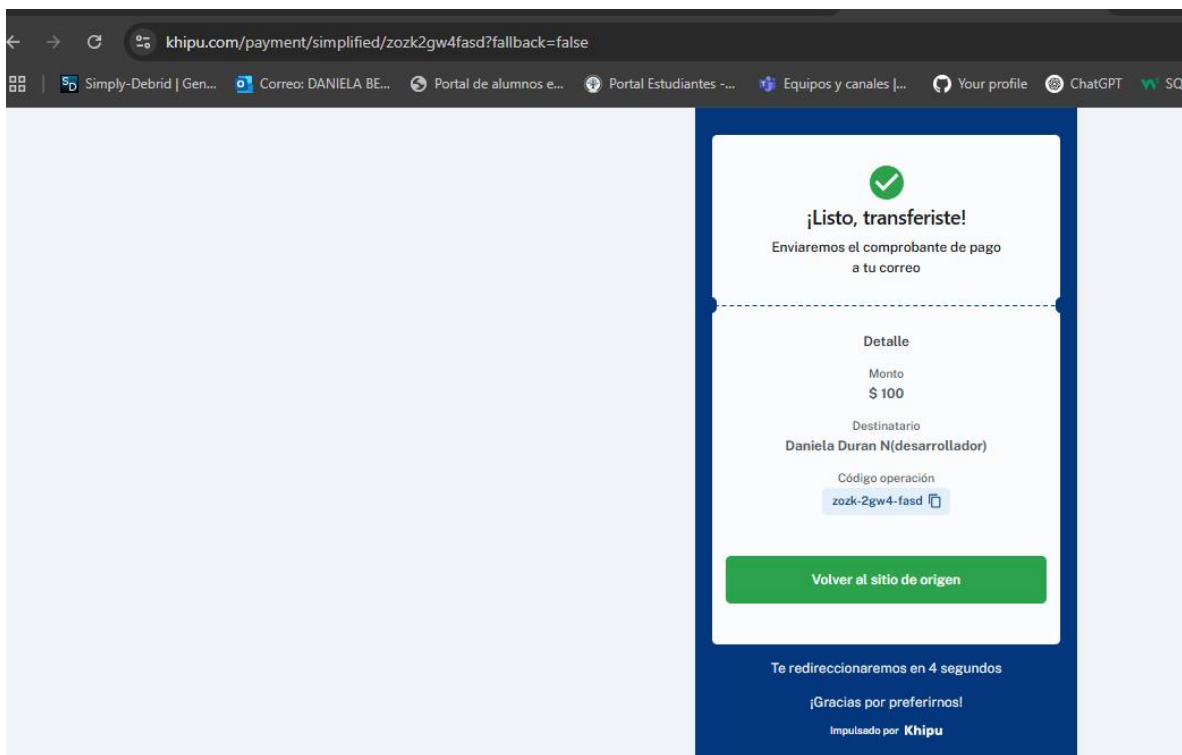
Ingresar

Anular pago

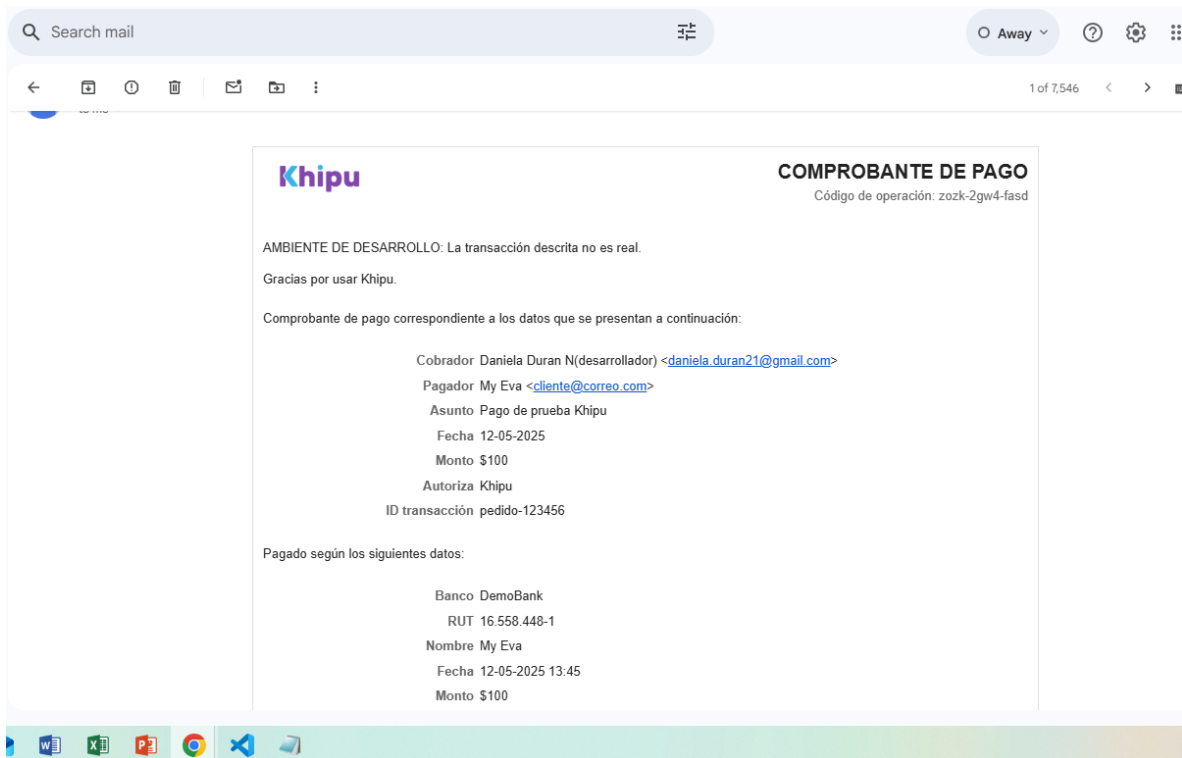
Privacidad • Ayuda

Impulsado por Khípu

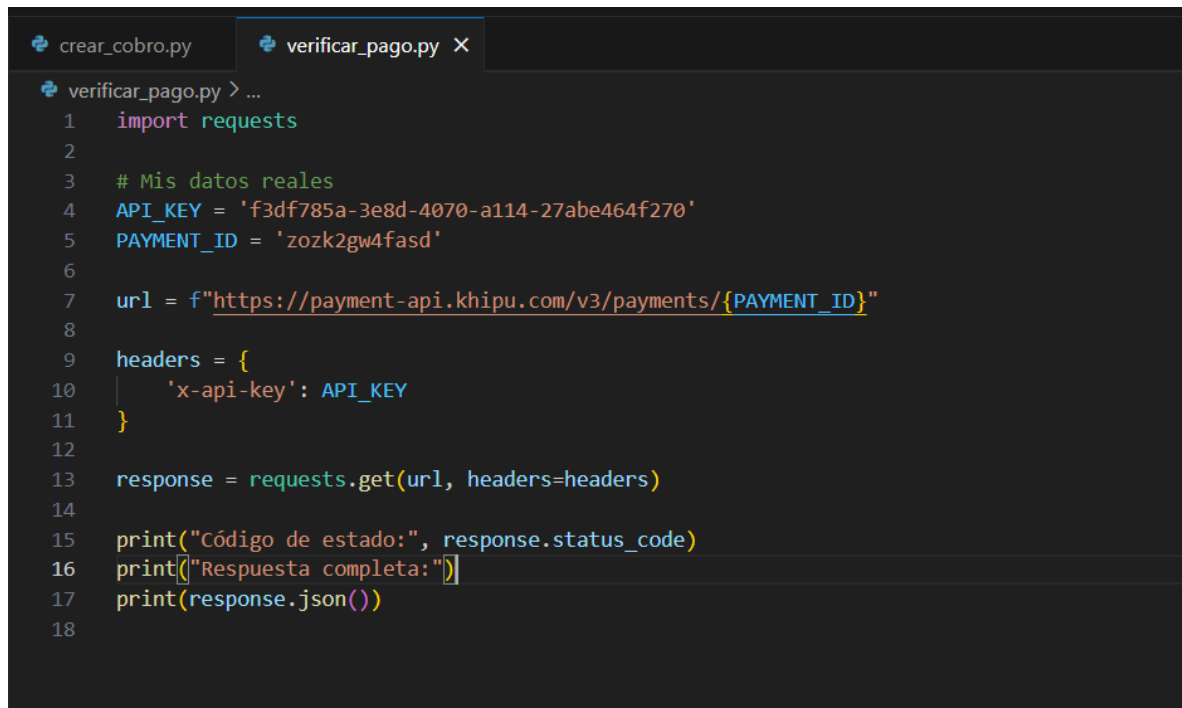
Ingreso datos y transferencia exitosamente:



Me llega un mensaje de correo notificando el pago exitoso:



Para verificar pago creamos verificar_pago.py



```
1 import requests
2
3 # Mis datos reales
4 API_KEY = 'f3df785a-3e8d-4070-a114-27abe464f270'
5 PAYMENT_ID = 'zozk2gw4fasd'
6
7 url = f"https://payment-api.khipu.com/v3/payments/{PAYMENT_ID}"
8
9 headers = {
10     'x-api-key': API_KEY
11 }
12
13 response = requests.get(url, headers=headers)
14
15 print("Código de estado:", response.status_code)
16 print("Respuesta completa:")
17 print(response.json())
18
```

Respuesta en la terminal con un código 200 y status: done nos confirma el paso exitoso



```
PS C:\Users\Usuario\Desktop\Khipu_testing> python verificar_pago.py
Código de estado: 200
Respuesta completa:
Código de estado: 200
Código de estado: 200
Código de estado: 200
Código de estado: 200
Respuesta completa:
{'amount': '100.0000', 'app_url': 'khipu:///pos/zozk2gw4fasd', 'attachment_urls': [], 'bank': 'DemoBank', 'bank_account_number': '000000000000540321', 'bank_id': 'Bawd', 'ready_for_terminal': False, 'simplified_transfer_url': 'https://app.khipu.com/payment/simplified/zozk2gw4fasd', 'transfer_url': 'https://khipu.com/payment/manual/zozk2gw4fasd'}
Código de estado: 200
Respuesta completa:
{'amount': '100.0000', 'app_url': 'khipu:///pos/zozk2gw4fasd', 'attachment_urls': [], 'bank': 'DemoBank', 'bank_account_number': '000000000000540321', 'bank_id': 'Bawd', 'ready_for_terminal': True, 'receipt_url': 'https://s3.amazonaws.com/notifications.khipu.com/CPKH-1205251345-zozk2gw4fasd.pdf', 'receiver_id': '497950', 'responsible_user_email': 'daniela.duran21@gmail.com', 'return_url': 'https://tusitio.com/exito', 'send_email': False, 'send_reminders': False, 'simplified_transfer_url': 'https://app.khipu.com/payment/simplified/zozk2gw4fasd', 'status': 'done', 'status_detail': 'normal', 'subject': 'Pago de prueba Khipu', 'third_party_authorization_details': '', 'transaction_id': 'pedido-123456', 'transfer_url': 'https://khipu.com/payment/manual/zozk2gw4fasd'}
```

Mejoramos el aspecto visual de verificar_pago.py con JSON

```
crear_cobro.py  verificar_pago.py X

verificar_pago.py > ...
1  import requests
2  import json
3
4  API_KEY = 'f3df785a-3e8d-4070-a114-27abe464f270'
5  PAYMENT_ID = 'zozk2gw4fasd'
6
7  url = f"https://payment-api.khipu.com/v3/payments/{PAYMENT_ID}"
8
9  headers = {
10     'x-api-key': API_KEY
11 }
12
13 response = requests.get(url, headers=headers)
14
15 print("Código de estado:", response.status_code)
16
17 respuesta = response.json()
18 print("Respuesta formateada:")
19 print(json.dumps(respuesta, indent=4, ensure_ascii=False))
20
21
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Respuesta formateada:
{
  "amount": "100.0000",
  "app_url": "khipu:///pos/zozk2gw4fasd",
  "attachment_urls": [],
  "bank": "DemoBank",
  "bank_account_number": "000000000000540321",
  "bank_id": "Bawdf",
  "body": "Gracias por tu compra :)",
  "cancel_url": "https://tusitio.com/cancelado",
  "conciliation_date": "2025-05-12T00:00:00.000Z",
  "currency": "CLP",
  "custom": "Este es un pago de prueba",
  "discount": "0.0000",
  "expires_date": "2025-05-13T17:37:26.393Z",
  "funds_source": "not-available",
  "notification_token": "64daf71624af11009d2788f9b89da7a70a9c6c4462d7fff9c709a802ce582f93",
  "notify_api_version": "3.0",
  "notify_url": "https://tusitio.com/webhook",
  "out_of_date_conciliation": false,
  "payer_email": "cliente@correo.com",
  "payer_name": "Cobrador de desarrollo #497.950",
  "payment_id": "zozk2gw4fasd",
  "payment_method": "simplified transfer",
  "payment_url": "https://khipu.com/payment/info/zozk2gw4fasd",
  "personal_identifier": "16.558.448-1",
  "picture_url": "",
  "ready_for_terminal": true,
  "receipt_url": "https://s3.amazonaws.com/notifications.khipu.com/CPKH-1205251345-zozk2gw4fasd.pdf",
  "receiver_id": 497950,
  "responsible_user_email": "daniela.duran21@gmail.com",
  "return_url": "https://tusitio.com/exito",
  "send_email": false,
  "send_reminders": false,
  "simplified_transfer_url": "https://app.khipu.com/payment/simplified/zozk2gw4fasd",
  "status": "done",
  "status_detail": "normal",
  "subject": "Pago de prueba Khipu",
  "third_party_authorization_details": "",
  "transaction_id": "pedido-123456",
  "transfer_url": "https://khipu.com/payment/manual/zozk2gw4fasd"
}
```

Lo mismo para crear_cobro.py nos ordena mucho mejor la vista en la terminal.

```
crear_cobro.py > ...
1  import requests
2  import json
3
4  # Mis credenciales reales
5  API_KEY = 'f3df785a-3e8d-4070-a114-27abe464f270'
6  RECEIVER_ID = '497950'
7
8  url = 'https://payment-api.khipu.com/v3/payments'
9
10 headers = {
11     'x-api-key': API_KEY,
12     'Content-Type': 'application/json'
13 }
14
15 data = {
16     "receiver_id": RECEIVER_ID,
17     "subject": "Pago de prueba Khipu",
18     "amount": 100,
19     "currency": "CLP",
20     "transaction_id": "pedido-123456",
21     "payer_email": "cliente@correo.com",
22     "return_url": "https://tusitio.com/exito",
23     "cancel_url": "https://tusitio.com/cancelado",
24     "notify_url": "https://tusitio.com/webhook",
25     "custom": "Este es un pago de prueba",
26     "body": "Gracias por tu compra :)"
27 }
28
29 response = requests.post(url, headers=headers, json=data)
30
31 print("Código de estado HTTP:", response.status_code)
32
33 respuesta = response.json()
34 print("Respuesta formateada:")
35 print(json.dumps(respuesta, indent=4, ensure_ascii=False))
36
```

```
or directory
PS C:\Users\Usuario\Desktop\Khipu_testing> python crear_cobro.py
Código de estado HTTP: 200
Respuesta formateada:
{
  "app_url": "khipu:///pos/gn1yx9qhiyxj",
  "payment_id": "gn1yx9qhiyxj",
  "payment_url": "https://khipu.com/payment/info/gn1yx9qhiyxj",
  "ready_for_terminal": false,
  "simplified_transfer_url": "https://app.khipu.com/payment/simplified/gn1yx9qhiyxj",
  "transfer_url": "https://khipu.com/payment/manual/gn1yx9qhiyxj"
}
PS C:\Users\Usuario\Desktop\Khipu_testing> |
```

En resumen:

1. Se creó cuenta en modo desarrollador en el portal de Kiphu siguiendo las instrucciones señaladas en el kit de implementación, se utilizó DemoBank como banco de pruebas y se obtuvieron las credenciales receiver_id y API key válida para llamadas a la API (x-api-key).
2. Se implementó Python en la llamada POST/PAYMENTS para crear un cobro. No se utilizaron herramientas manuales desde el portal para generar datos. Se verificó que la integración responde con código HTTP 200 en la terminal.
3. Simulación de pago, se generó un payment_url válido que se probó en el navegador y se simularon pagos con Demobank.
4. Se verifica el estado del pago al crear un segundo script verificar_pago.py para consultar GET/payments/{payment_id} de lo cual la API respondió con estado "done" lo que indica que el pago fue simulado con éxito.

Se mejoraron errores como 403 Forbidden y se corrigieron los headers y también se añadió formato JSON para una mayor legibilidad en la terminal con las respuestas.