

Serial Port Integration Guide — TerpaPay

Guía práctica para programadores que necesitan conectar un sistema externo a TerpaPay a través del puerto serial (COM).

1. Conexión Física

Conecta la terminal PAX a tu computadora mediante un cable USB. La terminal se presenta como un puerto serial virtual.

Parámetro	Valor
Baudrate	9600
Data bits	8
Parity	None
Stop bits	1
Flow control	None
Line ending	\r\n (CR+LF)

En Linux el dispositivo suele ser `/dev/ttyACM0`. En Windows será un `COMx`.

2. Protocolo de Mensajes

Formato

Todos los mensajes son texto plano, una línea por comando, terminados con `\r\n`:

```
<comando>\r\n
```

No se requiere framing binario ni prefijos de longitud. Se pueden enviar múltiples comandos en secuencia (modo ráfaga).

Normalización

El parser de TerpaPay aplica `trim()` y `lowercase()` a cada mensaje. El input JSON es rechazado con NACK.

3. Comandos Soportados

Comandos de Pago

Comando	Descripción	Ejemplo
p{MONTO}	Inicia cobro; acepta <code>itemId</code> opcional con <code>p25.50</code> , <code>ITEM-01</code> o <code>p(25.50, ITEM-01)</code>	<code>p25.50\r\n</code>
cancel	Cancela cobro en curso	<code>cancel\r\n</code>
void	Reversa última transacción	<code>void\r\n</code>

Formato del monto:

- Regex: `^p\s*(?\\s*([0-9]+(?:\\.[0-9]+)?)\\s*)?$', ^p\s*([0-9]+(?:\\.[0-9]+)?)\\s*,\\s*(^[,]*?)\\s*$ o ^p\s*(\\s*([0-9]+(?:\\.[0-9]+)?)\\s*,\\s*(^[,]*?)\\s*)$`
- Decimal solo con punto; la coma siempre separa `itemId`: `p25.50`, `p100`, `p2.14,5`, `p1500,99`, `P(1.00)`, `p(25.50, ITEM-01)`, `P(125,50)`
- Si quieres cobrar tres con catorce, usa `p3.14`; `p3,14` significa monto 3 con `itemId=14`.
- Múltiples comas son inválidas: `p2,2,2`.
- `itemId` es opcional, no puede venir vacío y se limita a 100 caracteres.

Comandos de Control

Comando	Descripción
activePayment	Habilita procesar pagos, cancel y void
desactivePayment	Deshabilita procesar pagos, cancel y void
ping	Verifica conectividad

Comandos de Consulta

Comando	Descripción
status	Estado completo (device, transacción, conectividad, última tx)
device_state	Estado operativo interno
connectivity	Estado de conexión a internet y al root
last_transaction_status	Última transacción recordada en memoria

Comandos de Protocolo

Comando	Descripción
ack	Confirmación a nivel protocolo; el parser la acepta pero no genera efecto
nack	Confirmación negativa a nivel protocolo; el parser la acepta pero no genera efecto

4. Formato de Respuestas

Todas las respuestas son JSON en una sola línea. Ejemplos:

Cobro exitoso:

```
{"transactionId":"uuid","amount":25.50,"chargedAmount":25.50,"status":"APPROVED","date":"2026-06-10T12:30:00","terminal":"SN123456","itemId":"ITEM-01"}
```

Cobro rechazado:

```
{"transactionId":"","amount":25.50,"chargedAmount":0.0,"status":"NOT_APPROVED","date":"2026-06-10T12:30:00","terminal":"SN123456","itemId":"ITEM-01"}
```

Void (comportamiento actual):

```
{"transactionId":"","amount":0.0,"chargedAmount":0.0,"status":"APPROVED","date":"2026-06-10T12:30:00","terminal":"SN123456","itemId":""}
```

No existe un payload dedicado `voidOk` o `voidFail` en la versión actual.

Cancelación:

```
{"type":"cancelOk","data":{"command":"cancel","message":"cancel ok"}}
```

Ping:

```
{"type":"pong","data":null}
```

Status:

```
{"type":"status","data":{"deviceState":"AVAILABLE","transactionState":"IDLE","paymentEventsEnabled":true,"connectivity":{"internetConnected":true,"rootConnected":true},"lastTransaction":{"status":"NONE","payload":null}}}
```

Device State:

```
{"type":"deviceState","data":{"state":"AVAILABLE","transactionState":"IDLE","paymentEventsEnabled":true}}
```

Connectivity:

```
{"type":"connectivity","data":{"internetConnected":true,"rootConnected":true}}
```

Last Transaction:

```
{"type":"lastTransactionStatus","data":{"status":"PAYMENT_APPROVED","payload":{...}}}
```

NACK:

```
{"type":"nack","data":{"reason":"Payment events are disabled","command":"p"}}
```

ACK automático:

La versión actual no emite un ack automático al recibir p, status, device_state, connectivity o last_transaction_status. Si el comando es válido, TerpaPay responde directamente con el payload final correspondiente.

ActivePayment:

```
{"type":"activePayment","data":{"command":"activePayment","enabled":true,"message":"payment events enabled"}}
```

DesactivePayment:

```
{"type":"desactivePayment","data":{"command":"desactivePayment","enabled":false,"message":"payment events disabled"}}
```

5. Flujo de Cobro Típico

```
PC > p(25.50)\r\n
(usuario inserta tarjeta en terminal PAX)
(transacción se procesa con el banco)
```

```
TERPA >
{"transactionId":"abc-123","amount":25.50,"chargedAmount":25.50,"status":"APPROVED",
"date":"...", "terminal":"SN001","itemId":""}
```

Flujo con itemId

```
PC > p(25.50, ITEM-01)\r\n
... (procesamiento)
```

```
TERPA >
{"transactionId":"abc-123","amount":25.50,"chargedAmount":25.50,"status":"APPROVED",
"date":"...", "terminal":"SN001","itemId":"ITEM-01"}
```

Flujo con control de eventos

```
PC > ping\r\n
TERPA > {"type":"pong","data":null}
```

```
PC > activePayment\r\n
TERPA > {"type":"activePayment","data":
{"command":"activePayment","enabled":true,"message":"payment events enabled"}}

PC > p25.50\r\n
... (procesamiento)

PC > status\r\n
TERPA > {"type":"status","data":
{"deviceState":"AVAILABLE","transactionState":"IDLE","paymentEventsEnabled":true,..
.}}
```

6. Manejo de Estados

Payment Events Toggle

`activePayment` y `desactivePayment` habilitan/deshabilitan el procesamiento de cobros.

Cuando están deshabilitados:

- `p{MONTO}` → responde NACK: "Payment events are disabled"
- `cancel` → responde NACK
- `void` → responde NACK

Esto permite bloquear operaciones durante ventanas de mantenimiento.

Device Busy

Si la terminal está ocupada (procesando un pago), cualquier nuevo `p{MONTO}`, `void`, o `cancel` puede recibir NACK con razón "Device busy".

7. Conexión por WiFi (alternativa)

Si se usa WiFi en lugar de serial:

Parámetro	Valor
Protocolo	TCP/IP
Puerto	8888
Modo	TerpaPay es servidor
Formato	Mismos comandos de texto plano

El servidor WiFi acepta un cliente a la vez. Callbacks: `onClientConnected`, `onClientDisconnected`, `onMessageReceived`.

8. Demo Rápido con Python

El archivo `TerpaPayDemoPy.py` es una herramienta interactiva de prueba:

```
# Instalar pyserial si no está
pip install pyserial

# Ejecutar demo interactivo
python3 TerpaPayDemoPy.py
```

Comandos disponibles en el demo:

```
> ping          → envía "ping"
> pay           → envía "p1" (cobro de $1)
> cancel        → envía "cancel"
> void          → envía "void"
> status        → envía "status"
> last_tx       → envía "last_transaction_status"
> list          → muestra lista de comandos
> raw <texto>   → envía texto sin resolver alias
```

El demo también incluye `TerpaPayStressDemo.py` para pruebas automatizadas de estrés:

```
python3 docs/TerpaPayStressDemo.py --port /dev/ttyACM0
```

Este script ejecuta ~27 comandos en secuencia probando flujos completos de pago, cancelación y void con diferentes combinaciones de estados.

9. Especificaciones Técnicas del Puerto Serial

Propiedad	Valor
Baudrate	9600
Bits	8
Paridad	Ninguna
Stop bits	1
Buffer TX	Sin control de flujo
Buffer RX	Polling cada ~50ms
Timeout de lectura recomendado	500ms
Timeout de espera de respuesta	2.5s

10. Resumen de Códigos de Estado

DeviceState

Valor	Significado
AVAILABLE	Listo para recibir comandos
BUSY	Ocupado, rechazar comandos entrantes
SEND_PAYMENT	Cobro enviado al lector de tarjetas
RECEIVE_PAYMENT_RESPONSE	Procesando respuesta del banco
START_CANCEL_FLOW	Cancelando operación en curso
START_REVERSAL_FLOW	Ejecutando reverso/anulación
ERROR_STATE	Estado de error

TransactionState

Valor	Significado
IDLE	Sin transacción activa
PAYMENT_IN_PROGRESS	Cobro en proceso
VOID_IN_PROGRESS	Reverso en proceso

LastTransactionStatus

Valor	Significado
NONE	No hay transacciones previas
PAYMENT_APPROVED	Última transacción aprobada
PAYMENT_DECLINED	Última transacción rechazada
PAYMENT_TIMEOUT	Timeout en última transacción
PAYMENT_CANCELLED	Última transacción cancelada

11. Solución de Problemas

Problema	Posible Causa	Solución
No hay respuesta	Puerto incorrecto	Verificar /dev/ttyACM0 o COMx
Respuesta ilegible	Baudrate incorrecto	Usar 9600 8N1

Problema	Posible Causa	Solución
Conexión rechazada	Terminal apagada	Verificar que la terminal esté encendida
Comando ignorado	Formato incorrecto	Verificar que termine con \r\n
NACK "JSON input"	Se envió JSON	Solo enviar comandos de texto plano
Sin respuesta al ping	Payments deshabilitados	El ping siempre responde; verificar conexión física