

# MANUAL TÉCNICO

## Sistema de Gestión de Cobros

Autor: Jairo De León  
Email: [jairodll48@gmail.com](mailto:jairodll48@gmail.com)

Fecha: Febrero 2026  
Versión: 1.0

# TABLA DE CONTENIDOS

## Contenido

TABLA DE CONTENIDOS .....	2
1. INTRODUCCIÓN .....	3
1.1 Propósito .....	3
1.2 Alcance .....	3
2. Requerimientos funcionales .....	4
2.1 Crear cliente .....	4
2.2 Registrar cobro .....	5
2.3 Procesar cobro .....	7
2.4 Procesar lote de cobros .....	9
2.5 Consulta de cobros por cliente .....	10
3. CONCLUSIONES .....	12
3.1 Resumen .....	12
3.2 Sugerencias .....	12

# 1. INTRODUCCIÓN

## 1.1 Propósito

Este documento describe los métodos implementados según el manual de la prueba técnica y cómo poder ejecutarlos.

## 1.2 Alcance

Este documento incluye:

- Ejecución de método 1 (Crear cliente)
- Ejecución de método 2 (Registrar cobro)
- Ejecución de método 3 (Procesar cobro)
- Ejecución de método 4 (Procesar lote de cobros)
- Ejecución de método 5 (Consulta de cobros por cliente)

## 2. Requerimientos funcionales

A continuación, se explica brevemente cada uno de los enunciados:

### 2.1 Crear cliente

Se registran los siguientes datos en la petición

usuario-controller

POST /usuarios/save

Parameters

No parameters

Request body **required** application/json

```
{  "idUsuario": 0,  "cul": "5288",  "nombre": "Usuario prueba",  "correo": "user@correo.com",  "estado": "Activo",  "fechaCreacion": "2026-02-14T19:44:23.051Z",  "fechaModificacion": "2026-02-14T19:44:23.051Z",  "password": "123",  "rol": 1}
```

NOTA: tomar en cuenta los siguientes roles para los usuarios

Resultados 1 x

select \* from gestion\_cobros.dbo.roles

	id_rol	nombre	fecha_creacion	fecha_modificacion
1	1	administrador	2024-11-19	[NULL]
2	2	cliente	2024-11-19	[NULL]
3	3	interno	2024-11-19	[NULL]

Validamos el resultado exitoso:

Server response

Code Details

200

Response body

```
{  "idUsuario": 20,  "cul": "5288",  "nombre": "Usuario prueba",  "correo": "user@correo.com",  "estado": "Activo",  "fechaCreacion": "2026-02-14T19:50:12.090+00:00",  "fechaModificacion": null,  "password": "$2a$10$H18gQEvRvtUgC8j.VCaVr08SR31Vok74U.nNErBAe8LFqReprnrVre"}
```

Response headers

```
cache-control: no-cache,no-store,max-age=0,must-revalidate
connection: keep-alive
content-type: application/json
date: Sat, 14 Feb 2026 19:50:12 GMT
expires: 0
keep-alive: timeout=60
pragma: no-cache
transfer-encoding: chunked
vary: Origin,Access-Control-Request-Method,Access-Control-Request-Headers
x-content-type-options: nosniff
x-frame-options: DENY
x-xss-protection: 0
```

	id_usuario	cui	nombre	correo	estado	fecha_creacion	fecha_modificacion	password
1	7	123456789	Juan	juan@gmail.com	Activo	2024-11-20	[NULL]	\$2a\$10\$RkE9Xr.Cat0dKuKAmKVBUuVnFQVGrsciaMAgbGGdmx8K5RvBCXUR6
2	12	1234565464646	Gustavo	gustavo@correo.com	Activo	2024-11-20	[NULL]	\$2a\$10\$G/Ncg/HZc4gYDrqWfVaLi.frZv.zfT9HhbPlqK3MkRo7ILUIQPesi
3	13	1564165465465	Evelyn	evelyn@correo.com	Activo	2024-11-20	[NULL]	\$2a\$10\$1Dlw0TqYcMP0.3hc8Gj.13w/DIYTVJhj/zjFwYduX1wu8dnUG
4	14	3033202650108	Jairo	jairodil48@gmail.com	Activo	2026-02-14	2026-02-14	\$2a\$10\$99wTn5zakOYNCCaO18ZOOg2pPEEEGG6cMe5p8h.M4YeRhcKiL
5	18	3033202650109	Jairo	jai0ro0@correo.com	Activo	2026-02-14	2026-02-14	\$2a\$10\$MkLAp70MQeDaUVEontZ7Oxku3eL2sVeEo8aleLJCdwSkMifZLb6
6	19	3033202650129	Jairo	jai0ro0@correo.com	Activo	2026-02-14	[NULL]	\$2a\$10\$MzViosEn.O.DCj/c4K3u3cgXK8rGjeoyCzC3ZrZz0x0JumMc
7	20	5288	Usuario prueba	user@correo.com	Activo	2026-02-14	[NULL]	\$2a\$10\$H1BgQEvRvtUgC8j.VCaVrO8SR31VoK74U.nNErBAeBLFqReprnvRe

En caso queramos volver a ingresar el mismo CUI, nos devolverá el siguiente error:

Curl

```

curl -X 'POST' \
  'http://localhost:8080/umg/usuarios/save' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
    "idUsuario": 0,
    "cui": "5288",
    "nombre": "Usuario prueba",
    "correo": "user@correo.com",
    "estado": "Activo",
    "fechaCreacion": "2026-02-14T19:44:23.051Z",
    "fechaModificacion": "2026-02-14T19:44:23.051Z",
    "password": "123",
    "rol": 1
  }'

```

Request URL

http://localhost:8080/umg/usuarios/save

Server response

Code
Details

400

Undocumented Error: response status is 400

Response body

```

{
  "success": false,
  "message": "Ya existe un usuario con el CUI/DPI proporcionado",
  "status": 400
}

```

## 2.2 Registrar cobro

El siguiente método se asocia al id del usuario creado anteriormente:

POST
/cobros/save

Parameters
Cancel
Reset

No parameters

Request body required
application/json

```

{
  "idUsuario": 20,
  "monto": 500,
  "moneda": "GTQ",
  "descripcion": "pago colegiatura cursos",
  "referenciaExterna": "REF-TEST-005"
}

```

Execute

Validamos el registro en base de datos con el estado pendiente

	id_cobro	id_usuario	moneda	moneda	estado	fecha_creacion	fecha_proceso	referencia_externa	descripcion
1	1	14	750.5	GTQ	PROCESADO	2026-02-14 10:53:17.063	2026-02-14 11:07:31.897	REF-A7B3C9D1	Cobro mensual
2	2	14	1,500.5	GTQ	FALLIDO	2026-02-14 11:06:15.610	2026-02-14 11:09:14.057	REF-A7B3C9D2	Cobro mensual
3	3	14	500	GTQ	PROCESADO	2026-02-14 11:22:19.357	2026-02-14 11:24:41.293	REF-TEST-001	Cobro de prueba
4	4	14	200	GTQ	PROCESADO	2026-02-14 11:22:30.827	2026-02-14 11:24:41.350	REF-TEST-001	Cobro de prueba
5	5	14	800	GTQ	PROCESADO	2026-02-14 11:22:35.100	2026-02-14 11:24:41.360	REF-TEST-001	Cobro de prueba
6	6	14	15,000	GTQ	FALLIDO	2026-02-14 11:22:40.140	2026-02-14 11:24:41.370	REF-TEST-001	Cobro de prueba
7	7	20	500	GTQ	PENDIENTE	2026-02-14 13:56:44.307	[NULL]	REF-TEST-005	pago colegiatura cursos

Se valida en bitácora las acciones

	id_auditoria	evento	resumen_payload	fecha	usuario_sistema	entidad	id_entidad
1	13	COBRO_REGISTRADO	Cobro registrado por 500.00 GTQ - Usuario: Usuario prueba	2026-02-14 13:56:44.327	SYSTEM	COBRO	7
2	12	LOTE_PROCESADO	Lote procesado: 4 total, 3 procesados, 1 fallidos, 0 pendientes	2026-02-14 11:26:41.217	SYSTEM	LOTE_COBROS	[NULL]
3	11	LOTE_PROCESADO	Lote procesado: 4 total, 3 procesados, 1 fallidos, 0 pendientes	2026-02-14 11:24:41.380	SYSTEM	LOTE_COBROS	[NULL]
4	10	COBRO_FALLIDO	Cobro rechazado: monto supera el límite permitido - Monto: 15000.00 GTQ	2026-02-14 11:24:41.377	SYSTEM	COBRO	6
5	9	COBRO_PROCESADO	Cobro procesado exitosamente - Monto: 800.00 GTQ	2026-02-14 11:24:41.363	SYSTEM	COBRO	5
6	8	COBRO_PROCESADO	Cobro procesado exitosamente - Monto: 200.00 GTQ	2026-02-14 11:24:41.353	SYSTEM	COBRO	4
7	7	COBRO_PROCESADO	Cobro procesado exitosamente - Monto: 500.00 GTQ	2026-02-14 11:24:41.327	SYSTEM	COBRO	3

En caso el id y el monto sean los incorrectos devolverá el siguiente error:

**POST** /cobros/save

Parameters

No parameters

Request body **required**

```
{  "idusuario": 200,  "monto": -500,  "moneda": "GTQ",  "descripcion": "pago colegiatura cursos",  "referenciaExterna": "REF-TEST-005"}
```

Curl

```
curl -X 'POST' \  'http://localhost:8080/umg/cobros/save' \  -H 'accept: */*' \  -H 'Content-Type: application/json' \  -d '{  "idusuario": 200,  "monto": -500,  "moneda": "GTQ",  "descripcion": "pago colegiatura cursos",  "referenciaExterna": "REF-TEST-005"  }'
```

Request URL

http://localhost:8080/umg/cobros/save

Server response

Code Details

400 Undocumented Error: response status is 400

Response body

```
{  "success": false,  "message": "El monto debe ser mayor a 0",  "status": 400}
```

## 2.3 Procesar cobro

Se hace continuación con el id registrado anteriormente

6	6	14	13000	GTQ	FALLIDO	2026-02-14 11:22:40.140	2026-02-14 11:24:41.370	REF-TEST-001	Cobro de prueba
7	7	20	500	GTQ	PENDIENTE	2026-02-14 13:56:44.307	[NULL]	REF-TEST-005	pago colegiatura cursos

### cobros-controller

**POST** /cobros/{id}/procesar

**Parameters**

Name

Description

**id** \* required

integer(\$int32)

(path)

Execute

**Responses**

Se valida la operación en base de datos

7	7	20	500	GTQ	PROCESADO	2026-02-14 13:56:44.307	2026-02-14 14:01:32.353	REF-TEST-005	pago colegiatura cursos
---	---	----	-----	-----	-----------	-------------------------	-------------------------	--------------	-------------------------

Verificamos que el campo de estado y la fecha hayan sido actualizados correctamente, si el cobro es mayor a 1500, no se podrá procesar

Name

Description

**id** \* required

integer(\$int32)

(path)

Execute

**Responses**

Curl

```
curl -X 'POST' \
'http://localhost:8080/umg/cobros/8/procesar' \
-H 'accept: */*' \
-d ''
```

Request URL

```
http://localhost:8080/umg/cobros/8/procesar
```

Server response

Code

Details

200

Response body

```
{
  "idCobro": 8,
  "idUsuario": 20,
  "monto": 18000,
  "moneda": "GTQ",
  "estado": "FALLIDO",
  "fechaCreacion": "2026-02-14T20:03:30.290+00:00",
  "fechaProceso": "2026-02-14T20:04:43.219+00:00",
  "referenciaExterna": "REF-TEST-006",
  "descripcion": "pago colegiatura cursos"
}
```

auditoria 1 X							
select * from gestion_cobros.dbo.auditoria a Enter a SQL expression to filter results (use Ctrl+Space)							
	id_auditoria	evento	resumen_payload	fecha	usuario_sistema	entidad	id_entidad
1	16	COBRO_FALLIDO	Cobro rechazado: monto supera el límite permitido - Monto: 18000.00 GTQ	2026-02-14 14:04:43.227	SYSTEM	COBRO	8
2	13	COBRO_REGISTRADO	Cobro registrado por 500.00 GTQ - Usuario: Usuario prueba	2026-02-14 14:05:50.300	SYSTEM	COBRO	6
3	14	COBRO_PROCESADO	Cobro procesado exitosamente - Monto: 500.00 GTQ	2026-02-14 14:01:32.373	SYSTEM	COBRO	7
4	13	COBRO_REGISTRADO	Cobro registrado por 500.00 GTQ - Usuario: Usuario prueba	2026-02-14 13:56:44.327	SYSTEM	COBRO	7
5	12	LOTE_PROCESADO	Lote procesado: 4 total, 3 procesados, 1 fallidos, 0 pendientes	2026-02-14 11:28:41.277	SYSTEM	LOTE_COBROS	[NULL]
6	11	LOTE_PROCESADO	Lote procesado: 4 total, 3 procesados, 1 fallidos, 0 pendientes	2026-02-14 11:24:41.380	SYSTEM	LOTE_COBROS	[NULL]
7	10	COBRO_FALLIDO	Cobro rechazado: monto supera el límite permitido - Monto: 15000.00 GTQ	2026-02-14 11:24:41.377	SYSTEM	COBRO	6
8	9	COBRO_PROCESADO	Cobro procesado exitosamente - Monto: 800.00 GTQ	2026-02-14 11:24:41.363	SYSTEM	COBRO	5
9	8	COBRO_PROCESADO	Cobro procesado exitosamente - Monto: 200.00 GTQ	2026-02-14 11:24:41.353	SYSTEM	COBRO	4



## 2.4 Procesar lote de cobros

Se procede a crear registros y a la vez se reutilizarán otros que ya fueron procesados

8	20	18.000	GTQ	FALLIDO	2026-02-14 14:03:30.290	2026-02-14 14:04:43.220	REF-TEST-006	pago colegiatura cursos
9	20	500	GTQ	PENDIENTE	2026-02-14 14:07:42.267	[NULL]	REF-TEST-007	pago colegiatura cursos
10	20	500	GTQ	PENDIENTE	2026-02-14 14:07:52.210	[NULL]	REF-TEST-008	pago colegiatura cursos
11	20	500	GTQ	PENDIENTE	2026-02-14 14:07:56.667	[NULL]	REF-TEST-009	pago colegiatura cursos

**POST** **/cobros/lotos/procesar**

**Parameters**

No parameters

**Request body** **required**

```
{  
  "idsCobros": [9,10,11.7]  
}
```

Se agregan a la lista los ids creados y 1 de prueba para validar que no fue procesado

```
{  
  "total": 3,  
  "procesados": 3,  
  "fallidos": 0,  
  "pendientes": 0,  
  "resultados": [  
    {  
      "idCobro": 9,  
      "estadoAnterior": "PENDIENTE",  
      "estadoNuevo": "PROCESADO",  
      "mensaje": "Procesado exitosamente"  
    },  
    {  
      "idCobro": 10,  
      "estadoAnterior": "PENDIENTE",  
      "estadoNuevo": "PROCESADO",  
      "mensaje": "Procesado exitosamente"  
    },  
    {  
      "idCobro": 11,  
      "estadoAnterior": "PENDIENTE",  
      "estadoNuevo": "PROCESADO",  
      "mensaje": "Procesado exitosamente"  
    }  
  ]  
}
```

Se logra verificar que fueron registrados únicamente 3 ya que son los únicos que cumplen con la validación, se verifica en bitácora

auditoria 1 X							
select * from gestion_cobros.dbo.auditoria a Enter a SQL expression to filter results (use Ctrl+Space)							
	id_auditoria	AZ evento	AZ resumen_payload	fecha	AZ usuario_sistema	AZ entidad	id_entidad
1	23	LOTE_PROCESADO	Lote procesado: 3 total, 3 procesados, 0 fallidos, 0 pendientes	2026-02-14 14:09:45.560	SYSTEM	LOTE_COBROS	[NULL]
2	22	COBRO_PROCESADO	Cobro procesado exitosamente - Monto: 500.00 GTQ	2026-02-14 14:09:45.557	SYSTEM	COBRO	11
3	21	COBRO_PROCESADO	Cobro procesado exitosamente - Monto: 500.00 GTQ	2026-02-14 14:09:45.553	SYSTEM	COBRO	10
4	20	COBRO_PROCESADO	Cobro procesado exitosamente - Monto: 500.00 GTQ	2026-02-14 14:09:45.543	SYSTEM	COBRO	9

## 2.5 Consulta de cobros por cliente

Utilizaremos el id del cliente creado en el primer método

GET

/cobros/usuario/{id}/cobros

Parameters

Name	Description
id <span>required</span>	
integer(\$int32)	20
(path)	
estado	
string	estado
(query)	
desde	
string(\$date-time)	desde
(query)	
hasta	
string(\$date-time)	hasta
(query)	

Execute

Clear

Responses

```
1 {
2   {
3     "idCobro": 11,
4     "idUsuario": 20,
5     "monto": 500,
6     "moneda": "GTQ",
7     "estado": "PROCESADO",
8     "fechaCreacion": "2026-02-14T20:07:56.667+00:00",
9     "fechaProceso": "2026-02-14T20:09:45.557+00:00",
10    "referenciaExterna": "REF-TEST-009",
11    "descripcion": "pago colegiatura cursos"
12  },
13  {
14    "idCobro": 10,
15    "idUsuario": 20,
16    "monto": 500,
17    "moneda": "GTQ",
18    "estado": "PROCESADO",
19    "fechaCreacion": "2026-02-14T20:07:52.210+00:00",
20    "fechaProceso": "2026-02-14T20:09:45.550+00:00",
21    "referenciaExterna": "REF-TEST-008",
22    "descripcion": "pago colegiatura cursos"
23  },
24  {
25    "idCobro": 9,
26    "idUsuario": 20,
27    "monto": 500,
28    "moneda": "GTQ",
29    "estado": "PROCESADO",
30    "fechaCreacion": "2026-02-14T20:07:42.267+00:00",
31    "fechaProceso": "2026-02-14T20:09:45.537+00:00",
32    "referenciaExterna": "REF-TEST-007",
33    "descripcion": "pago colegiatura cursos"
34  },
35  {
36    "idCobro": 8,
37    "idUsuario": 20,
38    "monto": 18000,
39    "moneda": "GTQ",
40    "estado": "FALLIDO",
41    "fechaCreacion": "2026-02-14T20:03:30.290+00:00",
42    "fechaProceso": "2026-02-14T20:04:43.220+00:00",
43    "referenciaExterna": "REF-TEST-006",
44    "descripcion": "pago colegiatura cursos"
45  },
46  {
47    "idCobro": 7,
48    "idUsuario": 20,
49    "monto": 500,
50    "moneda": "GTQ",
51    "estado": "PROCESADO",
52    "fechaCreacion": "2026-02-14T19:56:44.307+00:00",
53    "fechaProceso": "2026-02-14T20:01:32.353+00:00",
54    "referenciaExterna": "REF-TEST-005",
55    "descripcion": "pago colegiatura cursos"
56  }
57 }
```

*Ilustración 1: objeto retornado posterior a la ejecución por id de cliente, se copia json para una mejor visualización*

## **3. CONCLUSIONES**

### **3.1 Resumen**

Los cinco endpoints principales han sido desarrollados, probados y documentados, garantizando su correcto funcionamiento

### **3.2 Sugerencias**

Se pueden utilizar otras herramientas como postman para ejecutar las peticiones