

MANUAL TÉCNICO

Sistema de Gestión de Cobros

Autor: Jairo De León
Email: 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, "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 }
```

NOTA: tomar en cuenta los siguientes roles para los usuarios

Resultados 1 ×					
select * from gestion_cobros.dbo.roles r					
	123	id_rol	A-Z nombre	fecha_creacion	fecha_modificacion
Grilla	1	1	administrador	2024-11-19	[NULL]
Texto	2	2	cliente	2024-11-19	[NULL]
Texto	3	3	interno	2024-11-19	[NULL]

Validamos el resultado exitoso:

Server response

Code Details

200 Response body

```
{ "idUsuario": 20, "cui": "5288", "nombre": "Usuario prueba", "correo": "user@correo.com", "estado": "Activo", "fechaCreacion": "2026-02-14T19:50:12.090+00:00", "fechaModificacion": null, "password": "$2a$10$H1BgQEvRvtUgC8j.VcaVr08SR31VoK74U.nNErBAeBLFqRepnrvRe" }
```

Download

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
```

	<input type="checkbox"/>	id_usuario	cui	nombre	correo	estado	fecha_creacion	fecha_modificacion	password
1	<input checked="" type="checkbox"/>	7	123456789	Juan	juan@gmail.com	Activo	2024-11-20	[NULL]	\$2a\$10\$RkE9XrCat0dKuKAmVBUuvnfvGfrsiaMAgbGGdnx8K5RvBCXUR6
2	<input type="checkbox"/>	12	1235465464646	Gustavo	gustavo@correo.com	Activo	2024-11-20	[NULL]	\$2a\$10\$G/Ncg/HZc4gYDrqWFVaLi.frZv.ztT9HhbJqK3MkRo7JLUIQPesi
3	<input type="checkbox"/>	13	1564165465465	Evelyn	evelyn@correo.com	Activo	2024-11-20	[NULL]	\$2a\$10\$J1Dix0TqCMPO.3hcxBcj.3w/DIYTvljx/zjfwvYuXTwubdnUG
4	<input type="checkbox"/>	14	3033202650108	Jairo	jairodl48@gmail.com	Activo	2026-02-14	2026-02-14	\$2a\$10\$99wTn5zxak0YNCCa018ZOOG2pIEEEGG6cMe5p8n.M4YeRhCeK1L
5	<input type="checkbox"/>	18	3033202650109	Jairo	jairo0@correo.com	Activo	2026-02-14	2026-02-14	\$2a\$10\$MKlAp70MqeDaUVEonTz7Oxku3eL2sVeEo8aleJCdwsKmifrZLb6
6	<input type="checkbox"/>	19	3033202650129	Jairo	jairo00@correo.com	Activo	2026-02-14	[NULL]	\$2a\$10\$Wv08-en0.0.UC/c4kds3gaXsgjzOyCf5z2.v20xJ5B.mC
7	<input type="checkbox"/>	20	5288	Usuario prueba	user@correo.com	Activo	2026-02-14	[NULL]	\$2a\$10\$H1BgQEvrUgC8j.VCaVrO8SR31VoK74U.nNErBAeBLFqRepnrvRe

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

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	monto	moneda	estado	fecha_creacion	fecha_proceso	referencia_externa	descripcion
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	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	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	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	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	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	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	COBRO_REGISTRADO	Cobro registrado por 500.00 GTQ - Usuario: Usuario prueba	2026-02-14 13:56:44.327	SYSTEM	COBRO	7
2	LOTE_PROCESADO	Lote procesado: 4 total, 3 procesados, 1 fallidos, 0 pendientes	2026-02-14 11:24:41.271	SYSTEM	LOTE_COBROS	[NULL]
3	11	LOTE_PROCESADO	2026-02-14 11:24:41.380	SYSTEM	COBRO	6
4	10	COBRO_FALLIDO	2026-02-14 11:24:41.377	SYSTEM	COBRO	5
5	9	COBRO_PROCESADO	2026-02-14 11:24:41.363	SYSTEM	COBRO	4
6	8	COBRO_PROCESADO	2026-02-14 11:24:41.353	SYSTEM	COBRO	3
7	7	COBRO_PROCESADO	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

b 7	b 7	14	15.000	GTQ	FALLIDO	2020-02-14 11:24:40.140	2020-02-14 11:24:41.570	REF-TEST-001	[NULL]	REF-TEST-005	Cobro de prueba	pago colegiatura cursos
--------	--------	----	--------	-----	---------	-------------------------	-------------------------	--------------	--------	--------------	-----------------	-------------------------

cobros-controller

POST /cobros/{id}/procesar

Parameters

Name Description

id * required
integer(\$int32) 7
(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) 8
(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

select * from gestion_cobros.dbo.auditoria a | Enter a SQL expression to filter results (use Ctrl+Space)

Grilla	<input type="checkbox"/> id_auditoria	Az evento	Az resumen_payload	fecha	Az usuario_sistema	Az entidad	Iaz 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	15	COBRO_REGISTRADO	Cobro registrado por 500.00 GTQ - Usuario: Usuario prueba	2026-02-14 14:03:30.300	SYSTEM	COBRO	8
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

3	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	9	20	500	GTQ	PENDIENTE	2026-02-14 14:07:42.267	[NULL]	REF-TEST-007	pago colegiatura cursos
10	10	20	500	GTQ	PENDIENTE	2026-02-14 14:07:52.210	[NULL]	REF-TEST-008	pago colegiatura cursos
11	11	20	500	GTQ	PENDIENTE	2026-02-14 14:07:56.667	[NULL]	REF-TEST-009	pago colegiatura cursos

POST /cobros/lotes/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						
	id_auditoria	evento	resumen_payload	fecha	usuario_sistema	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 * required integer(\$int32) (path)	20
estado string (query)	estado
desde string(\$date-time) (query)	desde
hasta string(\$date-time) (query)	hasta

Execute **Clear**

Responses



```
1 | {
2 |     "idCobro": 11,
3 |     "idUsuario": 20,
4 |     "monto": 500,
5 |     "moneda": "GTQ",
6 |     "estado": "PROCESADO",
7 |     "fechaCreacion": "2024-02-14T20:07:56.467+00:00",
8 |     "fechaProceso": "2024-02-14T20:09:45.557+00:00",
9 |     "referenciaExterna": "REF-TEST-009",
10 |     "descripcion": "pago colegiatura cursos"
11 | },
12 | [
13 |     {
14 |         "idCobro": 10,
15 |         "idUsuario": 20,
16 |         "monto": 500,
17 |         "moneda": "GTQ",
18 |         "estado": "PROCESADO",
19 |         "fechaCreacion": "2024-02-14T20:07:52.310+00:00",
20 |         "fechaProceso": "2024-02-14T20:09:45.550+00:00",
21 |         "referenciaExterna": "REF-TEST-009",
22 |         "descripcion": "pago colegiatura cursos"
23 |     },
24 |     {
25 |         "idCobro": 9,
26 |         "idUsuario": 20,
27 |         "monto": 500,
28 |         "moneda": "GTQ",
29 |         "estado": "PROCESADO",
30 |         "fechaCreacion": "2024-02-14T20:07:42.287+00:00",
31 |         "fechaProceso": "2024-02-14T20:09:45.537+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": "FALSO POS",
41 |         "fechaCreacion": "2024-02-14T20:03:30.290+00:00",
42 |         "fechaProceso": "2024-02-14T20:04:43.220+00:00",
43 |         "referenciaExterna": "REF-TEST-006",
44 |         "descripcion": "pago colegiatura cursos"
45 |     },
46 |     [
47 |         {
48 |             "idCobro": 7,
49 |             "idUsuario": 20,
50 |             "monto": 500,
51 |             "moneda": "GTQ",
52 |             "estado": "PROCESADO",
53 |             "fechaCreacion": "2024-02-14T19:56:44.307+00:00",
54 |             "fechaProceso": "2024-02-14T20:01:32.353+00:00",
55 |             "referenciaExterna": "REF-TEST-005",
56 |             "descripcion": "pago colegiatura cursos"
57 |         }
58 |     ]
59 | }
```

Ilustración 1: objeto returned 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