

TEC

CURSO

BASES DE DATOS 1

CASO#1

Payment Assistant

Profesor

Rodrigo Núñez Núñez

Estudiantes

Carol Daniela Araya Conejo – 2024089174 |

Lindsay Nahome Marín Sánchez – 2024163904 |

Contenido

Listado de entidades	1
Investigación	6
Permisos, roles, accesos, grants, denys y similares	6
UserRoles:	8
Service	8
ServiceCategory	8
Modules	9
Restrictions	9
ContactInfoType	10
Transacciones y pagos	11
PaymentAttempt	11
Transaction	11
Subscripciones y planes	12
Subscriptions	13
Features	13
Bitácora del Sistema	14
LogTypes	14
LogSources	14
LogSeverity	15
Idiomas y monedas	15
Language	15
Currency	15
Manejo de APIs	16
Mastercard Send	16
Monedero Auto Product – BAC	19
Manejo de pagos agendados (Punto de vista: Tareas)	20
Envío de recordatorios	24
Referencias bibliográficas	26

Listado de entidades

A continuación, se presenta un listado de las entidades utilizadas en el diseño de base de datos correspondiente al caso 1, el cual cumple el rol de un Payment Assistant. Este se detalla de forma resumida, en la que partes importantes tales como mapeo de estas se abarcan en la parte de investigación.

1. User

2. UserRoles

Esta tabla maneja la relación entre los usuarios y sus respectivos roles.

3. Role

4. AuthSession

Las columnas con datos delicados como los tokens se guardan en un varbinary(500).

5. AuthPlatform

Las columnas con datos delicados como las keys se guardan en un varbinary(500).

6. Restriction

Al estar trabajando bajo una modalidad Allow All, se manejarán restricciones.

7. RoleRestriction

Esta tabla maneja la relación entre los roles de usuario y sus respectivas restricciones.

8. UserRestriction

Esta tabla maneja la relación entre los usuarios y sus respectivas restricciones.

9. Module

10. NotificationConfirmationReceived

11. NotificationMethod

12. Reminder

Este se refiere a todo tipo de mensaje SMS que recibe un usuario.

13. ReminderType

Se trabajará con recordatorios para confirmaciones, autorizaciones y aproximación de fechas de pago.

14. PaymentMethod

15. AvailablePaymentMethod

16. Service

Esta estructura se encarga de almacenar información útil sobre los servicios que estén registrados dentro de la aplicación. Estos facilitarán la navegación dentro de la aplicación, pues permiten a los usuarios reconocer servicios con los que están familiarizados.

17. ServiceCategory

Este indica el tipo de servicio del que se trata (Agua, Electricidad, Renta...).

18. Schedule

Name-> Se utiliza un varchar (50), se debe a que se utilizará descripciones tales como: "Cada 2 semanas" y "Cada primer lunes del mes".

19. ScheduleDetail

datePart -> al cumplir la utilidad de identificar el uso de la función, se define como varchar(10), ya que a nivel de cálculo de fechas se utilizará para programar pagos tales como "DAY" (Día) "WEEK" (Semana) "MONTH" (Mes) "YEAR" (Año)

20. AvailablePaymentMethodPerUser

21. AvailablePaymentMethodPerService

22. ContactUserInfo

23. ContactServiceInfo

24. ScheduledPayment

Amount: Decimal (10,2) -> Hasta 8 dígitos enteros y 2 decimales (Ejemplo: 99999999.99)

25. PlanPerUser

El presente caso, permite solamente a los usuarios obtener suscripciones las cuales representan planes con características funcionales dentro de la aplicación.

26. Log

El registro con el que se estará trabajando captura todas las acciones que ocurren en el programa.

27. LogTypes

28. Se refiere al tipo de registro de log (por ejemplo, "Login", "Logout", "Pago ejecutado").

Ayuda a categorizar el propósito del log.

29. LogSource

Indica el origen del registro de log (por ejemplo, una aplicación, un servidor, un componente del sistema). Permite identificar de dónde proviene el evento registrado.

30. LogSeverity

Define la gravedad del evento registrado. Ayuda a priorizar la respuesta a los eventos según su importancia.

31. Language

Se refiere al idioma en el que se muestra la información o contenido (por ejemplo, "Español", "Inglés", "Francés"). Define el lenguaje utilizado para la interfaz de usuario o los mensajes.

32. Translation

Representa la traducción de un texto o contenido en un idioma diferente. Permite ofrecer la misma información en varios idiomas según las necesidades del usuario.

Caption -> Se almacena como nvarchar(300) para tomar en cuenta todo tipo de caracteres alrededor del mundo.

33. ContactInfoType

Representa el tipo de contacto, este es utilizado de forma relevante en el medio de pago ya que formas de transacción como sinpe movil amerita un número telefónico de origen y destino. De igual forma un pago por medio de PayPal se necesita un correo electrónico para el destino (payee definido por PayPal). Este manejo es explicado a mayor detalle en el apartado de investigación.

34. AudioPerUser

Registra los audios que fueron generados por el usuario y no obstante no se relacionaron con un pago que haya sido agendado.

35. PaymentAudios

Registra los audios que modificaron o estuvieron relacionados con el proceso de generar, o editar un pago (como tarea).

36. AudioFiles

En este avance no se aplica análisis y registro de interacción con el usuario de forma detallada.

37. PaymentAttempt

La función principal de esta tabla es almacenar la respuesta de el intento de pago sea exitosa o fallida. así como los datos de pago de la cuenta origen y destino estos encriptados cuando sea necesario. (explicado con mayor detalle en la sección de investigación)

38. PlanLimit

Al ser una app controlada por la cantidad de pagos y transferencias según el tipo de suscripción adquirido por el usuario. Este limite es utilizado por algunas features, por ello esta tabla es la que posee un campo BIT llamado unlimited si este es false la feature tendrá un límite que será controlado por la aplicación, de lo contrario la feature no tiene un limite. Se brindan ejemplos de las features explicadas:

39. ExchangeRate

Se refiere a la tasa de cambio entre dos monedas. Es el valor de una moneda en términos de otra moneda. Un "ExchangeRate" define cuántas unidades de una moneda se necesitan para obtener una unidad de otra moneda.

Decimal(10,4)->Permite almacenar valores como 1.2345, 1234.5678 sin perder precisión

40. Currency

Almacenar la información de las monedas disponible. Además, puede incluir información como la tasa de cambio, el símbolo de la moneda, y si está activa o no.

41. TransType

Muestra los tipos de transacciones a nivel de formas de manipular el dinero.

idTransType	name	deleted
1	Débito	0
2	Crédito	0
3	Transferencia	0
4	Pago con tarjeta	0
5	Pago en efectivo	0
6	Billetera digital	0
7	Criptomoneda	0
8	Reembolso	0

42. TransSubType

Es una categoría más específica dentro de un tipo de transacción general (como débito, crédito o transferencia). Define la naturaleza o propósito de una transacción, indicando si es, por ejemplo, una suscripción, un cashback, un reembolso o una disputa de pago.

transSubTypeid	name	deleted
1	Suscripción	0
2	Cashback	0
3	Claim Approved	0
4	Claim Denied	0
5	Pago Recurrente	0
6	Contracargo	0
7	Reembolso Parcial	0
8	Reembolso Total	0

43. Transaction

Esta estructura ayuda a almacenar y gestionar eficientemente las transacciones dentro del sistema de pagos. La cual es explicada a mayor detalle en la investigación.

44. PlanPrice

Amount -> NULL, esto se debe ya que al tener un plan gratuito este no será requerido.

FK_currencyId -> NULL, esto se debe ya que al tener un plan gratuito este no será requerido.

45. Subscription

Se refiere al modelo de pago donde un usuario paga una tarifa recurrente para acceder a un servicio o producto durante un período específico. Este pago puede ser mensual, anual o por otro intervalo de tiempo definido.

46. FeaturePerPlan

Hace referencia a las características o funcionalidades que están disponibles en un plan de suscripción específico. Los diferentes planes pueden ofrecer diferentes conjuntos de características y límites.

47. PlanFeature

Se refiere a las características o funcionalidades específicas que están asociadas con un plan de suscripción. Cada plan de un servicio o producto puede tener un conjunto de características que varían según el nivel de suscripción o el tipo de plan.

DataName-> Representa el tipo de dato que define el valor del feature.

Description->Representa una descripción del feature correspondiente a uno o varios planes.

Enable-> Valor condicional, este define si la feature está habilitada.

Feature	Tipo	Descripción
Número de pagos procesados	Con límite	Solo permite hasta 500 pagos al mes.
Cuentas bancarias vinculadas	Con límite	Máximo de 3 cuentas por usuario.
Soporte de tarjetas	Con límite	Solo se aceptan tarjetas estándar de débito y crédito.
Automatización de pagos	Sin límite	Se pueden configurar reglas personalizadas para pagos.
Recordatorios de pago	Con límite	Solo 5 notificaciones de recordatorio al mes.
Seguridad y cifrado	Sin límite	Protección avanzada con encriptación de extremo a extremo.
Acceso a reportes financieros	Con límite	Solo reportes básicos, sin exportación de datos.
Integraciones con servicios	Sin límite	Se pueden conectar múltiples plataformas compatibles.

Investigación

En la presente sección se muestra datos, explicación y mapeos relevantes que fueron resultado de la investigación del diseño.

Permisos, roles, accesos, grants, denys y similares

El presente diseño de bases de dato es utilizado bajo la forma Allow All. Por lo que a continuación de muestran los permisos definidos para cada rol de usuario en el sistema.

Permisos predeterminados (Usuarios)

- **Módulo de gestión de cuentas**
 - Crear cuentas de usuario.
 - Editar cuentas de usuario personales.
 - Eliminar cuentas de usuario personales.
- **Módulo de voz**
 - Utilizar comando de voz
- **Módulo de pagos**
 - Registrar pago recurrente
 - Editar pago recurrente
 - Eliminar pago recurrente
 - Confirmar detalles del pago
 - Confirmar monto, cuenta de origen al pago a ejecutar.
 - Enviar SMS de confirmación de pago
 - Autorizar pago a ejecutar.
 - Reintentar pago fallido
 - Eliminar pago configurado
 - Consultar historial de transacciones
 - Filtrar historial de transacciones
 - Recibir notificaciones de pagos próximos y confirmaciones
- **Módulo de planes**
 - Ver planes de suscripción
 - Renovar suscripción
- **Módulo de cuentas bancarias**
 - Configurar cuenta bancaria
 - Editar cuenta bancaria

Eliminar cuenta bancaria

Permisos predeterminados (Administrador de Servicios)

- **Módulo de gestión de cuentas**
 - Crear cuentas de administrador de servicios.
 - Editar cuentas de administrador de servicios.
 - Eliminar cuentas de administrador de servicios.
- **Módulo de servicios:**
 - Editar perfil del servicio
- **Módulo de cuentas bancarias**
 - Configurar cuenta bancaria
 - Editar cuenta bancaria
 - Eliminar cuenta bancaria

Permisos predeterminados (Administrador de Plataforma)

- **Módulo de gestión de cuentas**
 - Crear cuentas de usuarios.
 - Editar cuentas de usuario.
 - Eliminar cuentas de usuario.
 - Crear cuentas de administrador.
 - editar cuentas de administrador.
 - eliminar cuentas de administrador.
- **Módulo de control de usuarios:**
 - Otorgar permisos a usuario.
 - Deshabilitar permisos a usuario.
 - Ver historial de pagos de usuario.
- **Modulo Configuración de la aplicación:**
 - Modificar las configuraciones generales de la aplicación
 - frecuencia de recordatorios
 - notificaciones
 - opciones de integración de APIs.
 - Gestionar planes de suscripción
 - Gestionar facturación de usuarios

UserRoles:

Los usuarios con el rol **User** son los principales clientes de la aplicación. Tienen restricciones sobre el módulo de control de usuarios, servicios y configuración general.

Los usuarios con el rol de **Service Administrator** tienen la capacidad de agregar un perfil de servicio y editarlo para que otros usuarios lo utilicen como cuenta destino de sus pagos. Este rol resulta conveniente para compañías que quieran registrar sus servicios dentro del sistema.

Los usuarios con el rol de **Platform Administrator** no tienen restricciones. Estas cuentas están reservadas para que miembros del equipo de desarrollo administren el funcionamiento de la aplicación según las necesidades de la plataforma.

roleId	name
00	User
01	Service Administrator
02	Platform Administrator

Service

Estos perfiles son controlados por un Service Administrator. A continuación, se realiza un mapeo de esta entidad.

ServiceCategory

Las siguientes son las categorías bajo las cuales se puede clasificar un tipo de servicio. Esta clasificación resulta conveniente para comunicarse por medio de comando de voz.

serviceCategoryId	name
01	Electricity
02	Water
03	Rent
04	Insurance
05	Loan
06	Health Care
07	Car Insurance
08	Telephone
09	Other

Modules

Tipos de módulos con los que trabaja el sistema.

moduleId	name
01	AccManagement
02	Voice
03	Payments
04	Plans
05	BankAcc
06	Services
07	UserManagement
08	GeneralConfig

Restrictions

A continuación, se listan algunas restricciones establecidas para el sistema.

restrictionId	description	code	moduleId
0001	Create User	AC_CRUSR	AccManagement
0002	Edit Own User	AC_EDOUSR	AccManagement
0003	Remove Own User	AC_RMVOUSR	AccManagement
0004	Create Service Admin	AC_CRSRV	AccManagement
0005	Edit Service Admin	AC_EDSRV	AccManagement
0006	Remove Service Admin	AC_RMVSRV	AccManagement
0007	Create Platform Admin	AC_CRADM	AccManagement
0008	Edit Platform Admin	AC_EDADM	AccManagement
0009	Remove Platform Admin	AC_RMVADM	AccManagement

ContactInfoType

A continuación, se brinda un mapeo de los tipos de contacto.

contactInfoTypeId	name
00	PhoneNumber
01	Email
02	Website

Transacciones y pagos

Las entidades relacionadas con pagos deben trabajarse con especial cuidado, dado que administran las carteras virtuales de los usuarios. Por este motivo, se utilizan estructuras más complejas y con mejores medidas de seguridad.

PaymentAttempt

La siguiente es una tabla que ejemplifica una entidad de PaymentAttempt. Omitiendo las IDs de llaves foráneas.

paymentAttemptId	amount	actualAmount	moduleId	result	chargeToken	error	date	checkSum	userAuthorized	status	authNumber	response
123456789	500.25	500.25	2	Success	0×5d41402abc4b2a76b	0	2025-03-14 15:30:00	0×1a2b3c4d5e6f7g8h9i	1	Completed	AUTH1234ABC	1
987654321	1000.00	995.00	3	Error	0×3a41402def7c1b98c	1	2025-03-14 16:45:00	0×2b3c4d5e6f7g8h9ij	0	Pending	AUTH5678XYZ	0

Transaction

Este es un mapeo de la tabla de Transactions, omitiendo algunas IDs de llaves foráneas y la columna de descripción.

transactionId	amount	transDateTime	postTime	paymentAttemptId	refNumber	checkSum
123456789	250.75	2025-03-14 10:45:00	2025-03-14 11:00:00	987654321	REF12345ABC	0×5d41402abc4b2a76b
987654321	1500.50	2025-03-14 16:30:00	2025-03-14 16:45:00	123456789	REF67890XYZ	0×3a41402def7c1b98c

Subscripciones y planes

El sistema de asistente de pagos contará con tres tipos de suscripciones y una versión gratuita. Las cuales están basadas en la cantidad de pagos y transferencias. Las cuales su diferencia radica dentro del intento de pago y el pago exitoso.

None (Gratis)

- Pagos limitados.
- 1 pago agendado al mes.
- Configuración de 1 cuenta bancaria.

Basic (3.99\$)

- Pagos limitados.
- 6 pagos agendados al mes.
- Configuración de 1 cuenta bancaria.

Pro (9.99\$)

- Pagos limitados.
- 10 pagos agendados al mes.
- Configuración de 3 cuentas bancarias.

Premium (19.99\$)

- Pagos ilimitados.
- Configuración de 8 cuentas bancarias.
- Los usuarios con este beneficio reciben ayuda más rápida en caso de problemas con pagos o configuración.

★ Distribución de los Extras por Plan				
Feature	None (Gratis)	Basic (\$3.99)	Pro (\$9.99)	Premium (\$19.99)
Pagos programados	2	8	20	Ilimitados
Cuentas bancarias	1	2	4	8
Transferencias	1	3	8	20
Ads (Anuncios)	✓ (Si)	✓ (Si)	✗ (No)	✗ (No)
Soporte prioritario	✗ (No)	✗ (No)	✓ (Si)	✓ (VIP)
Notificaciones avanzadas	✗ (No)	✗ (No)	✓ (Si)	✓ (Si, con IA)
Pagos automáticos avanzados	✗ (No)	✗ (No)	✗ (No)	✓ (Si)
Pagos limitados	✓ (Si)	✓ (Si)	✓ (Si)	✗ (No)

Plan	Precio	Pagos Agendados	Transferencias	Cuentas Bancarias	Extras
None	Gratis	2 pagos	1 transferencia	1 cuenta	Anuncios
Basic	3.99\$	8 pagos	3 transferencias	2 cuentas	Soporte estándar
Pro	9.99\$	20 pagos	8 transferencias	4 cuentas	Soporte prioritario, notificaciones avanzadas
Premium	19.99\$	Ilimitados	20 transferencias	8 cuentas	Sin anuncios, soporte VIP, pagos automáticos avanzados

Subscriptions

La siguiente tabla ilustra la entidad de suscripciones, la cual muestra los tipos de suscripciones que serán ofrecidas al usuario.

subscriptionId	description	logoURL
00	None	https://example.com/logos/none.png
01	Basic	https://example.com/logos/basic.png
02	Pro	https://example.com/logos/pro.png
03	Premium	https://example.com/logos/premium.png

Features

planFeatureId	description	enabled	dataType
09	Configure 3 bank accounts	1	int
10	Configure 8 bank accounts	1	int
11	Priority support	1	bit
12	Ads	1	bit
13	Unlimited scheduled payments	1	bit

Bitácora del Sistema

La bitácora será utilizada para registrar eventos que suceden en el sistema, principalmente sobre tablas que tienen la modalidad de actualizar. A continuación, se detalla un mapeo de los principales tipos de logs que serán parte del registro en la tabla.

LogTypes

logTypeId	name
01	Login
02	Logout
03	Login failed
04	Change password
05	Disable restriction
06	Record audio
07	Execute Payment
08	Set up a bank account
09	Send notification

LogSources

La siguiente tabla describe las posibles fuentes de un registro que puede insertarse dentro de la bitácora del sistema a realizar.

logSourceId	name
00	Database
01	Mobile App
02	Api Server
03	Notification System

LogSeverity

logSeverityId	name
01	Info
02	Warning
03	Error

Idiomas y monedas***Language***

Esta tabla maneja los lenguajes disponibles en la aplicación.

languageId	name	culture
01	English	en-US
02	Spanish	es-ES
03	French	fr-FR
04	Japanese	ja-JP
05	German	de-DE

Currency

Esta tabla maneja los tipos de monedas alrededor del mundo.

currencyId	name	acronym	symbol
01	US Dollar	USD	\$
02	Euro	EUR	€
03	Japanese Yen	JPY	¥
04	British Pound	GBP	£
05	Costa Rican Colón	CRC	₡

Manejo de APIs

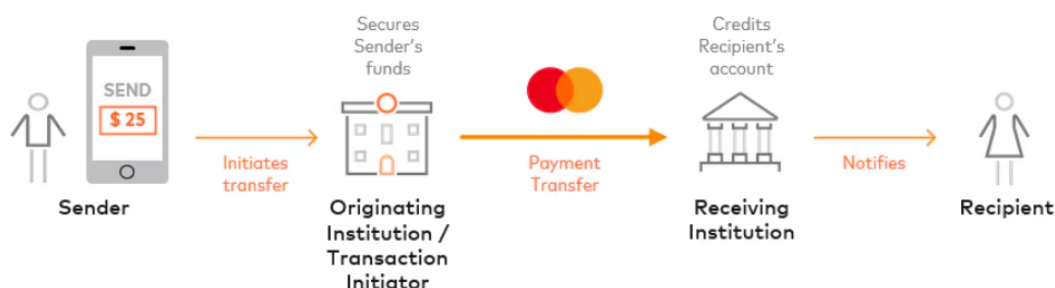
Mastercard Send

Mediante esta API, los usuarios pueden realizar pagos convenientemente a partir de información de contacto como números de teléfono o correos electrónicos.

Overview

The **Mastercard Send** Person-to-Person (P2P) services enable registered Program Participants (such as banks, issuers, and digital players) to provide P2P payment solutions that enable consumers/individuals (Senders) to transfer funds quickly, such as sending money to family and friends.

Acting as an Originating Institution or Transaction Initiator, the registered participant uses the services to initiate a Payment Transaction via the Mastercard Send platform to transfer funds to the Receiving Institution (Issuer) holding the Recipient's account.



Requests

Los datos relevantes para ejecutar una solicitud a la API de Mastercard Send serían los siguientes:

- transfer_reference: "DEF123456"
- payment_type": "P2P"
- amount: "5300"
- Currency: "USD"
- SenderAccountUri: "pan:5102589999999921;exp=2077-02;cvc=123"
- RecipientAccountUri: "pan:5102589999999913;exp=2077-02;cvc=123"
- status = "PENDING"

Request

Sandbox URL: <https://sandbox.api.mastercard.com/send/static/v1/partners/{partnerId}/transfers/payment>

MTF URL: <https://sandbox.api.mastercard.com/send/v1/partners/{partnerId}/transfers/payment>

Production URL: <https://api.mastercard.com/send/v1/partners/{partnerId}/transfers/payment>

```
{
  "payment_transfer": {
    "transfer_reference": "DEF123456",
    "payment_type": "P2P",
    "amount": "5300",
    "currency": "USD",
    "sender_account_uri": "pan:510258999999921;exp=2077-02;cvc=123",
    "sender": {
      "first_name": "John",
      "middle_name": "Tyler",
      "last_name": "Jones",
      "nationality": "USA"
    }
  }
}
```

Responses

La respuesta que regrese el API de Mastercard Send depende en el *Transfer Status*, esta es una notación que especifica el estado de la transferencia una vez realizada.

Transfer Status

Processed payment transfer requests will have a `transfer.status` response value that indicates whether the Payment Transaction was successful. This table shows the possible values for POST and GET calls.

Status	POST response	GET response	Meaning
APPROVED	Yes	Yes	The transaction was successful and has been approved.
DECLINED	Yes*	Yes	The transaction has been declined, either by the Receiving Institution or Mastercard.
UNKNOWN	Yes	Yes	The request has been accepted but processing has not completed within the allotted timeframe, possibly due to timeouts or network communication issues. For guidance, see Transaction Responses with an 'UNKNOWN' Status .
ERROR	—	Yes	Can be returned when a network or system error prevented processing of the transaction. This status is rarely seen in production.
REVERSED	—	Yes	The transaction has been reversed. This is unlikely to occur. Mastercard Send Payment Transactions are generally irrevocable and can only be reversed in exceptional circumstances.

Los *HTTP Response Codes* son otro tipo de dato que regresa la solicitud de pago. Este brinda una razón específica sobre los motivos por los que una tarjeta haya sido aceptada o rechazada según el caso.

HTTP Response Codes	Description
200 (OK)	The GET request to retrieve details for a transfer was successful.
201 (CREATED)	The POST request to create a transfer (Payment Transaction) was processed successfully. A response <code>status</code> value of 'APPROVED' indicates the transaction was successful. If the status is 'UNKNOWN', see Transaction Responses with an 'UNKNOWN' Status . If the POST request <code>decline_details</code> query parameter is set to 'true', a 201 response will also be returned when a transaction is declined (<code>status</code> value is 'DECLINED').
202 (ACCEPTED)	Can occur for some timeouts. The POST request was accepted but did not complete during the allotted timeframe, so the response <code>status</code> value is 'UNKNOWN'. Processing of the request will continue asynchronously. You can use a GET request to retrieve the latest status, see Transaction Responses with an 'UNKNOWN' Status .
400 (BAD REQUEST)	The request could not be fulfilled due to general errors such as validation errors or missing required data.
401 (UNAUTHORIZED)	Error code response for missing or invalid authentication token.
402 (REQUEST FAILED)	The default response when the POST request to create a transfer (Payment Transaction) was processed successfully but the transaction was declined. You can use a GET request to see the Network Response Code, which indicates the reason for declining the transaction, see Network Response Codes .

En caso de errores, la API regresa *códigos de error* que indican el problema que ocurrió durante el proceso de solicitud al API.

Error Codes

If your API request is unsuccessful, you should receive a service error response or a gateway error response.

Field	Description
RequestId	An application-specific request identifier. Example: rqst_73HB-5RO5-0OGS-53SG
Source	The unique identifier that attempts to define the field in error, when available. If an error is not associated with a specific field, System will be returned. If an error is produced because of required data, it would be presented with the field missing data. Examples: amount, recipient_account_uri
ReasonCode	The general cause of the error; see the table below. Examples: INVALID_INPUT_FORMAT, INVALID_INPUT_LENGTH, INVALID_INPUT_VALUE
Description	A textual description of the error. This is optional and will only be displayed if more information is available than is stored in the data identifier and reason code. Examples: Invalid Account URI, Destination Currency ISO code must be within 3 characters length

Monedero Auto Product – BAC

Dentro de las formas de pago se utilizará el monedero que permite utilizar la API phone to phone, el cual corresponde al envío directo de débito por teléfono. En este caso es necesario el URL POST.

POST

<https://rgw.6997-bd831c1c.us-south.ri1.apiconnect.appdomain.cloud/bac-development/development/payment-execution-s/payment-execution-procedure/v1.0.0/debit-and-transfer-funds/phone-to-phone/execute>

Este funciona por medio de Client Credentials, las cuales serán almacenadas en el método de pago. Al realizar una transacción vía teléfono se necesita un Phone Number Sender (usuario) y un Phone Numero Receiver (origen), los cuales serán obtenidos por medio de la tabla que posee un patrón de características variables. A continuación, se muestra el cuerpo de los parámetros a enviar.

Se obtiene los datos de contacto al relacionarlos con el medio de pago es decir un FK de **payment_contactServiceInfo** en y **payment_avaliablepaymentmethodperservice** es por ello la razón del campo **payment_avaliablepaymentmethodperservice** permitiendo NULL ya que puede que la información de pago no sea necesaria autenticarla por medio de un api como lo es un teléfono, email (en caso de pay pal). El mismo patrón aplica para el medio de pago de un usuario.

^
ResponseType

Ejemplo
Esquema

{
"Response" : true
"ResponseReason" : "0"
"AdditionalResponseInformation" : ""
}

^
TransactionReferencesPhoneToPhone

Ejemplo
Esquema

{
"AccountServiceReference" : "Sin Descripción"
"ClearingSystemReference" : "NoSeguro"
"PhoneNumberSender" : "88145325"
"PhoneNumberReceiver" : "83258555"
"CurrencyCode" : "Colones"
"CurrencyAndAmount" : "1367.00"
}

Manejo de pagos agendados (Punto de vista: Tareas)

Con la finalidad de almacenar los pagos agendados por el usuario se utiliza la tabla **payment_scheduledPayments** con ella se almacena un pago que se realizará según la calendarización que el usuario brindó, o de lo contrario un pago único. Estos pagos pueden ser deshabilitados por el usuario, por ello se utiliza el campo enable.

No obstante, al querer proceder con un pago, se genera un registro en la tabla **payment_paymentAttempts**, con ella se podrá realizar el intento de pago cuando es necesario, en caso de ser un pago exitoso se genera un registro en la tabla **payment_transactions**, es por ello por lo que se tiene una FK del intento de pago en la transacción referenciando al intento exitoso de no ser exitoso no se genera una transacción y se almacena el error.

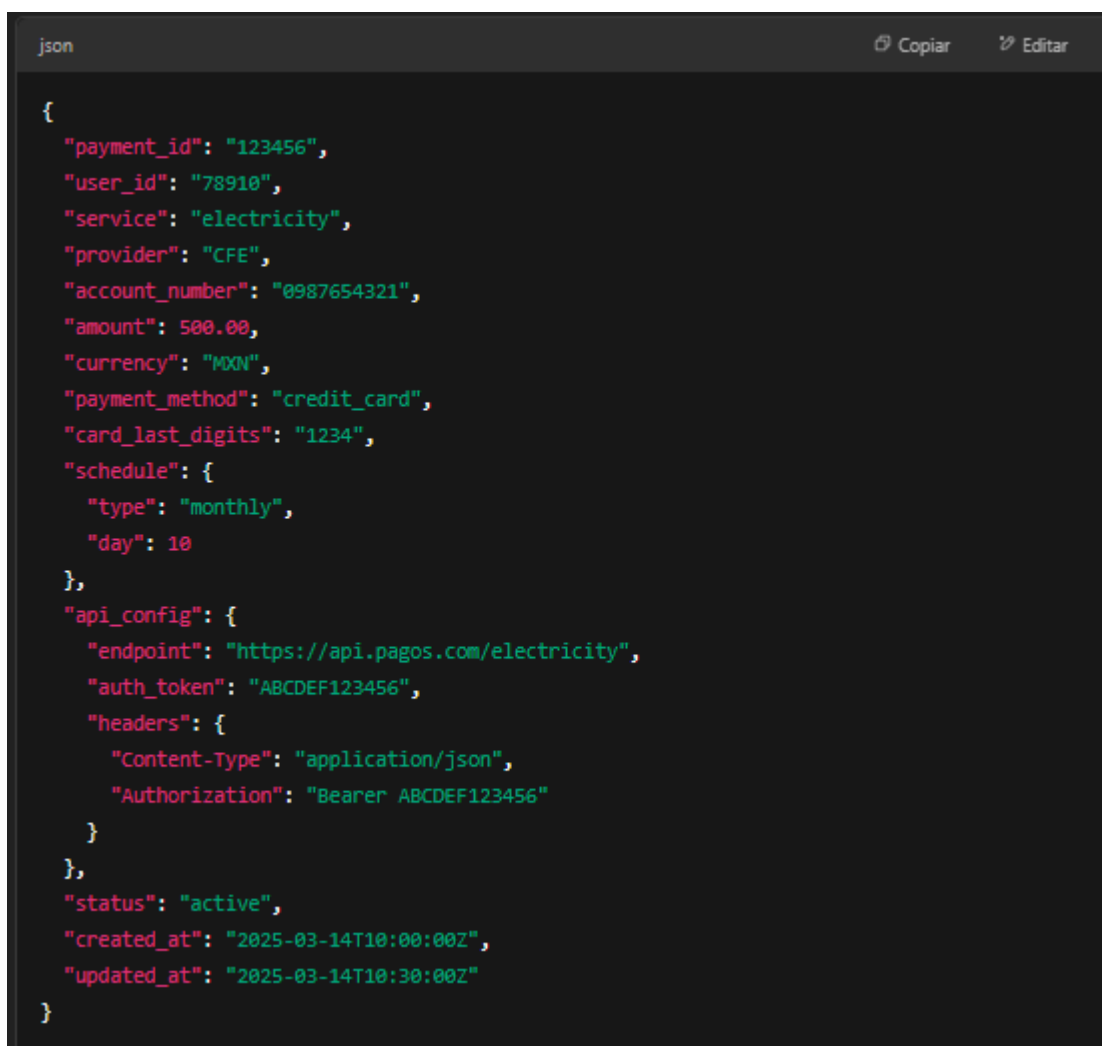
Dato relevante en payment_paymentAttempts: Al ser un pago hacia un servicio (Este puede ser formal (netflix) e informal (un sinpe movil)), se utiliza un FK llamada **avaliablepaymentmethodperserviceId** como cuenta destino y **avaliablepaymentmethodperuserId** como cuenta origen. Cabe recalcar que estos se encuentran en el pago agendado también ya que permite tener un acceso rápido. Estos aceptan nulos ya que los pagos no solo se pueden utilizar para esa funcionalidad, se podrían utilizar para un cobro de suscripción también. A continuación, se brinda un mapeo de la tabla analizada:

payment_scheduledPayments

scheduledPaymentId	scheduleId	userId	serviceId	createdDate	deleted	lastUpdated	enabled	amount	currencyId
1	1	1001	201	2025-03-12 10:00:00	0	2025-03-12 10:00:00	1	99.99	1
2	2	1002	202	2025-03-11 15:30:00	0	2025-03-11 15:30:00	1	49.5	2
3	3	1003	203	2025-03-10 09:00:00	0	2025-03-10 09:00:00	1	150.75	1

payment_availablepaymentmethods

Esta tabla contempla los casos en los que se amerite un API y así como configuración especial en cada tipo de medio de pago; la misma se guarda en un archivo JSON el campo es llamado configJson. A continuación, se brinda un ejemplo de un contenido que puede almacenar el JSON, en este caso se almacena los datos necesarios para realizar un pago de luz.



```

{
  "payment_id": "123456",
  "user_id": "78910",
  "service": "electricity",
  "provider": "CFE",
  "account_number": "0987654321",
  "amount": 500.00,
  "currency": "MXN",
  "payment_method": "credit_card",
  "card_last_digits": "1234",
  "schedule": {
    "type": "monthly",
    "day": 10
  },
  "api_config": {
    "endpoint": "https://api.pagos.com/electricity",
    "auth_token": "ABCDEF123456",
    "headers": {
      "Content-Type": "application/json",
      "Authorization": "Bearer ABCDEF123456"
    }
  },
  "status": "active",
  "created_at": "2025-03-14T10:00:00Z",
  "updated_at": "2025-03-14T10:30:00Z"
}

```

La siguiente tabla representa el mapeo con información que puede almacenar la tabla **payment_availablepaymentmethods**.

id	name	token	expTokenDate	maskAccount	callbackURLGET	callbackPost	callbackRedirect	payment methodsId	consumer Key	configJSON
1	Sinpe Movil - BAC	tC123abc456def789	2025-12-31 23:59:59	****1234	https://rgw.6997-bd831c1c.us-south.ri1.apiconnectt.appdomain.cloud/bac-development/development/payment-execution-s/payment-execution-procedure/v1.0.0/debit-and-transfer-funds/health-check/retrieve	https://rgw.6997-bd831c1c.us-south.ri1.apiconnectt.appdomain.cloud/bac-development/development/payment-execution-s/payment-execution-procedure/v1.0.0/debit-and-transfer-funds/health-check/retrieve	https://miapp.com/redirect	101	consumidorClave001	Mostrado en el ejemplo anterior
2	PayPal	pp987xyz654321	2026-01-15 23:59:59	****5678	https://api-m.sandbox.paypal.com/v1/invoicing/invoices?page=3&page_size=4&total_count_required=true\	https://api-m.sandbox.paypal.com/v2/checkout/orders	https://miapp.com/paypal/redirect	102	consumidorClave002	Mostrado en el ejemplo anterior
3	Account to Account – Master Card	txf789ghi456klm	2025-11-30 23:59:59	****9876	https://sandbox.api.mastercard.com/send/static/v1/partners/{partnerId}/transfers/{transferId}	https://sandbox.api.mastercard.com/send/static/v1/partners/{partnerId}/transfer s/payment	https://miapp.com/banco/redirect	103	consumidorClave003	Mostrado en el ejemplo anterior

payment_AvailablePaymentMethodsPerService

Nota: Esta tabla funciona como almacenamiento para datos de cuenta origen (usuario) y datos de cuenta destino (servicio).

Para explicar esta entidad, se puede utilizar un ejemplo en el que una cuenta destino, en el caso de la API de PayPal, puede utilizar información de contacto como el email_address de la otra cuenta (Payee según la nomenclatura que utiliza PayPal). Como es un correo electrónico, este dato no se tiene que encriptar en el API para que devuelva un token; no obstante, sigue siendo un medio de pago.

Otro ejemplo es el BAC que maneja el *phone to pone*, el cual requiere el dato "PhoneNumberSender":"88145325" y "PhoneNumberReceiver":"83258555". De igual forma, antes de eso maneja las clientCredentials.

Un último ejemplo es el número de abonado de un pago. Estos cuentan como medio de pago, no obstante están almacenados en un archivo JSON que corresponde a la configuración necesaria para ser enviada al API, explicado en la sección anterior.

Para ilustrar el funcionamiento de payment_AvailablePaymentMethodsPerService, se añade la tabla de contactServiceInfo, la cual maneja todos los contactos de un servicio que pueden llegar a funcionar como cuenta destino.

contactServiceInfoId	value	lastUpdate	enabled	contactInfoTypeId	serviceId
1	123-456-7890	2025-03-14 10:00:00	true	00	101
2	987-654-3210	2025-03-14 12:00:00	true	00	101
3	example@mail.com	2025-03-14 14:00:00	true	01	101
4	contact@mail.com	2025-03-14 16:00:00	true	01	101




Envío de recordatorios

SMS

Se utiliza la API Twilio. Cada intento de pago debe generar una notificación la cual tiene tipos de notificación de los cuales se mapean en la siguiente tabla correspondiente a la entidad `payment_reminderTypes`.

reminderTypeId	name	enable	answer_type
1	Recordatorio	1	N/A
2	Confirmación	1	Sí/No
3	Autorización	1	Código de Autorización

Para poder registrar el medio de notificación se utiliza la tabla **payment_NotificationMethods** la cual permite obtener información de acceso a las API'S que sean utilizadas en el sistema, así como los métodos que permiten realizar `callBack`, o solamente una notificación del sistema.

```
Send an SMS message  Node.js  
```

```
1 // Download the helper library from https://www.twilio.com/docs/node/install
2 const twilio = require("twilio"); // Or, for ESM: import twilio from "twilio";
3
4 // Find your Account SID and Auth Token at twilio.com/console
5 // and set the environment variables. See http://twil.io/secure
6 const accountSid = process.env.TWILIO_ACCOUNT_SID;
7 const authToken = process.env.TWILIO_AUTH_TOKEN;
8 const client = twilio(accountSid, authToken);
9
10 async function createMessage() {
11   const message = await client.messages.create({
12     body: "Hi there",
13     from: "+15557122661",
14     to: "+15558675310",
15   });
16
17   console.log(message.body);
18 }
19
20 createMessage();
```

Con la finalidad de guardar la interacción con el API, se tiene la tabla **payment_reminders** con ella se guarda los datos necesarios para enviar un mensaje. En referencia a formato INSERT se crea el campo lastReminder que indica si este fue el último registro agregado, es decir la última notificación enviada al usuario en relación con el pago agendado. Entre otros campos, se encuentran información necesaria para registrar una respuesta por parte de la API.

idreminder	requestId	message	status	createdDate	authorizedDate	lastReminder	error	ticketId	expectedResult
1	1001	[message]	1	2025-03-14 09:00:00	2025-03-14 09:05:00	0	NULL	TICKET001	success
2	1002	[message]	0	2025-03-14 10:30:00	2025-03-14 10:35:00	1	"Timeout error"	TICKET002	error
3	1003	[message]	1	2025-03-14 11:15:00	2025-03-14 11:20:00	0	NULL	TICKET003	success

Dentro de la tabla que permite almacenar la notificación se guarda los datos de resultado, en caso de ser success y error. Por un lado, en el caso de success se almacena dentro de un objeto JSON (el cual se muestra un ejemplo en la siguiente ilustración) por lo que la permite almacenar la respuesta. Por otro lado, en caso de ser error, se almacena la cadena de caracteres.

```

{
  "sid": "SM00000000000000000000000000000000",
  "date_created": "Thu, 25 Mar 2025 10:30:00 +0000",
  "date_updated": "Thu, 25 Mar 2025 10:30:10 +0000",
  "date_sent": "Thu, 25 Mar 2025 10:30:10 +0000",
  "account_sid": "AC000000000000000000000000000000",
  "to": "+1234567890",
  "from": "+0987654321",
  "body": "Hello, this is a test message from Twilio!",
  "status": "sent",
  "num_segments": 1,
  "num_media": 0,
  "direction": "outbound-api",
  "api_version": "2010-04-01",
  "price": "-0.00750",
  "price_unit": "USD",
  "error_code": null,
  "error_message": null,
  "uri": "/2010-04-01/Accounts/AC000000000000000000000000000000/Messages/SM000000000000000000000000000000"
}

```

Referencias bibliográficas

Mastercard. (2025). *Payment Transfer API*. <https://developer.mastercard.com/mastercard-send-person-to-person/documentation/api-reference/payment-transfer/>

Bac Credomatic. (s.f.). *Monedero 120 API*. Bac Credomatic Developer Portal. Recuperado el 14 de marzo de 2025, de https://developer-test.baccredomatic.com/outer/development/product/9740/api/9737#/Monedero_120/definitions

Twilio. (s.f.). *Phone number resource*. Twilio. Recuperado el 14 de marzo de 2025, de <https://www.twilio.com/docs/messaging/api/phonenumber-resource>

PayPal. (s.f.). *PayPal Checkout Standard*. PayPal Developer. Recuperado el 14 de mar. de 2025, de <https://developer.paypal.com/studio/checkout/standard>

Stripe. (s.f.). *Subscription management features explained and how to choose a software solution*. Stripe. Recuperado el 14 de mar. de 2025, de <https://stripe.com/es/resources/more/subscription-management-features-explained-and-how-to-choose-a-software-solution>

Adobe. (2024, diciembre 11). *Divulgaciones del usuario de la IA generativa de Adobe*. Adobe. <https://helpx.adobe.com/es/acrobat/using/gen-ai-user-guidelines.html>