Documentación técnica:<Notificaciones de colas>

**Versión <4.0>**

**CU\_0075**

**Historial de Revisiones**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** | **Autor** |
| 05/04/2025 | 1.0 | Documento inicial | Angel Zambrano |
| 14/04/2025 | 1.0 | Documento inicial | Luis Ramos |
| 17/04/2025 | 2.0 | Ajustes en la arquitectura propuesta | Angel Zambrano |
| 22-Abr-2025 | 3.0 | Ajustes en la documentacion | Angel Zambrano |
| 29-may-2025 | 4.0 | Ajustes en la documentacion | Angel Zambrano |

# Breve descripcion

Este proyecto tiene como objetivo el desarrollo de una solución integral para la gestión automatizada de notificaciones a clientes, tanto en tiempo real como en procesos masivos. La finalidad es asegurar el envío oportuno, confiable y trazable de mensajes informativos relacionados con operaciones de tarjetas de crédito, utilizando un ecosistema serverless y altamente disponible sobre la nube de AWS.

Para las notificaciones en línea, se expone una API segura que recibe eventos de diversas fuentes (procesador de pagos, microservicios internos) y los enruta hacia el servicio de notificaciones Latinia. En caso de indisponibilidad del servicio, los mensajes se encolan automáticamente en un sistema de colas FIFO con lógica de reintentos y fallback a una DLQ (Dead Letter Queue) para garantizar la persistencia y posterior reenvío.

En el caso de notificaciones batch, se automatiza la recolección de datos desde el ODS mediante procesos orquestados por AWS Glue y EventBridge. Estos datos se procesan para generar archivos .build y .csv categorizados por tipo de notificación (nemónico), los cuales son transferidos a una ruta compartida (EFS) consumida por Latinia para su procesamiento masivo.

La solución incorpora validaciones por tipo de mensaje (nemónico), control de calidad de datos, trazabilidad completa, monitoreo con CloudWatch y manejo de excepciones para asegurar un flujo robusto, escalable y tolerante a fallos.

# Flujo de eventos

# Flujo actual

Actualmente, Banco Bolivariano gestiona dos tipos de notificaciones dirigidas a sus clientes:

* Notificaciones en linea
* Notificaciones Batch

# Notificaciones en linea

En el flujo en línea, Credimatic envía una petición de trama 8587 hacia una interfaz expuesta a través del OSB (ReceptarNotificacionTC) del Banco Bolivariano. Este servicio de bus maneja las siguientes notificaciones:

* Diferimiento de consumo y avance (DIFCO)
* Diferimiento de consumo y avance (DIFC2)
* Consumo por internet (CONIN)
* Consumo por internet adicional (CONI2)
* Límite de cupo alcanzado (LIMCA)

Esta petición incluye una trama de datos con la información de la transacción, la cual es procesada por el sistema y posteriormente remitida al servicio de Latinia, responsable de gestionar y distribuir las notificaciones a los canales de comunicación habilitados por cada cliente (como SMS, correo electrónico, push).

En caso de mantenimiento o indisponibilidad del servicio de Latinia, se coordina con el área de monitoreo para detener temporalmente el envío desde Credimatic. Durante este período, las notificaciones generadas se acumulan y permanecen en espera hasta que Latinia restablezca su servicio, momento en el cual se procede con el envío masivo de los mensajes pendientes.

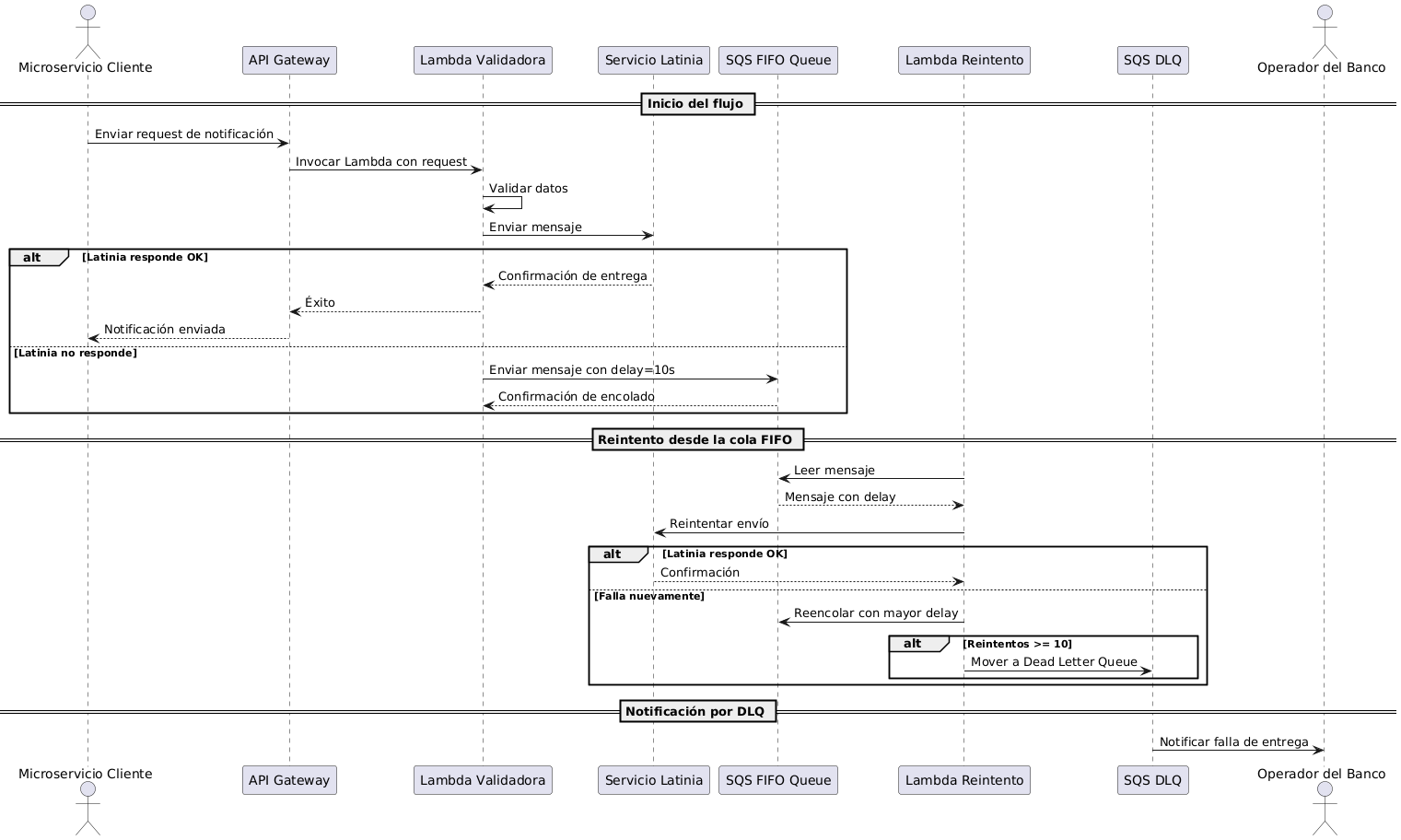
# Notificaciones Batch

Para las notificaciones en modo batch, el proceso actual consiste en extraer los datos de transacciones desde la base de datos operacional DB\_TC\_ODS alojada en On premise, generando posteriormente uno o varios archivos .build que contiene los registros correspondientes.

Este archivo es cargado por una tarea gaw a una fuente de datos accesible por Latinia, que se encarga de descargarlo y procesarlo para realizar el envío masivo de notificaciones a los clientes respectivos.

# Flujo propuesto

**Diagrama de secuencia Flujo notificaciones en linea-Batch**



Para las notificaciones en línea, se habilita un servicio en la nube encargado de recibir los eventos emitidos por la procesadora a través del subsistema SAT de notificaciones al cliente. Los eventos se reciben a través de un endpoint que obtiene los datos de la procesadora, obtiene datos de clientes y los envia hacia Latinia.

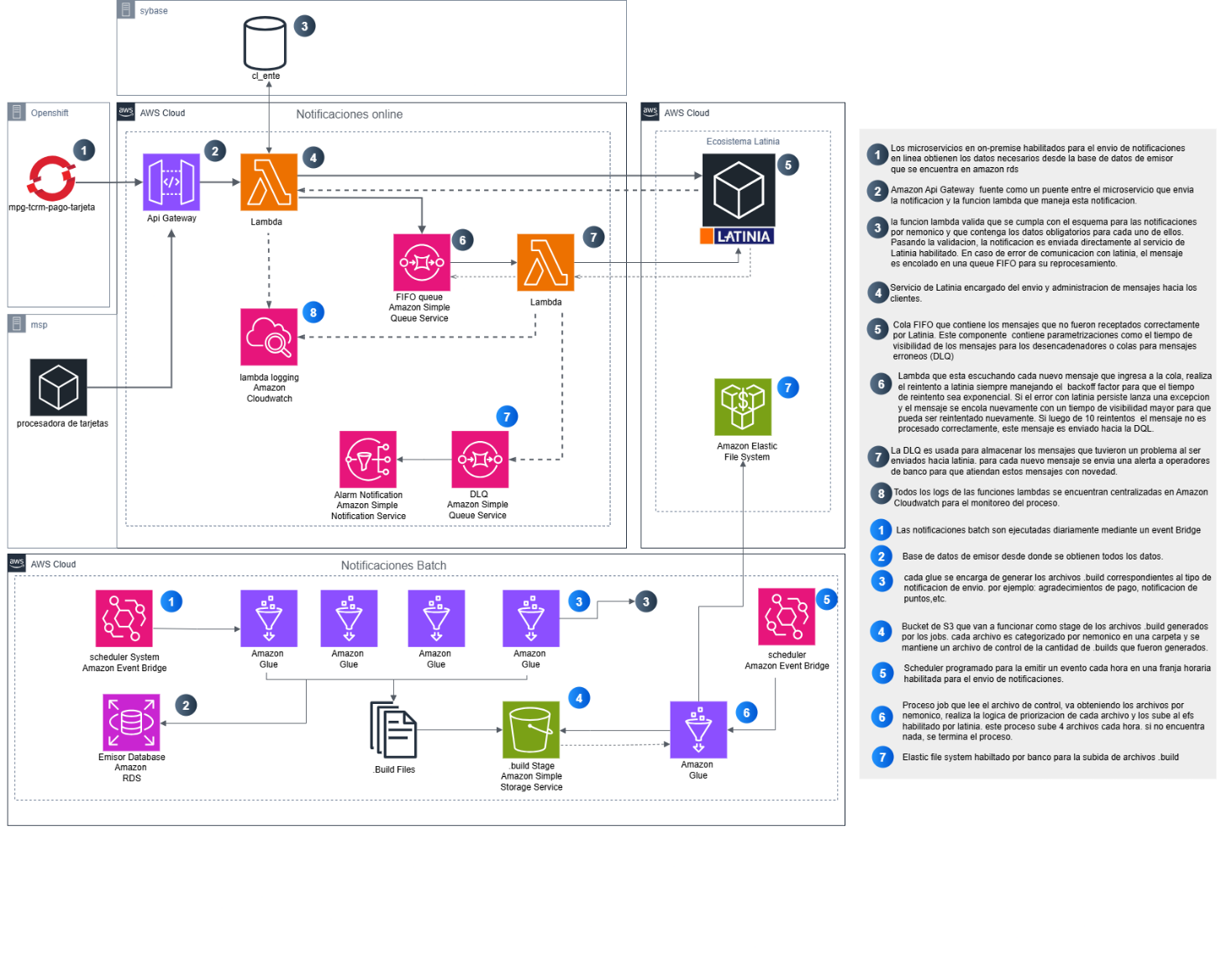
Ademas, se habilita un endpoint que permite a los microservicios enviar las notificaciones generadas hacia Latinia. El sistema implementa un flujo de reintentos utilizando mensajes encolados, asegurando el reenvío de notificaciones en caso de fallos en la entrega.

**Tabla

El contenido generado por IA puede ser incorrecto.**

En el caso de las notificaciones batch, se mantiene la generación diaria de archivos .build, los cuales se construyen a partir de datos extraídos de la base DB\_TC\_ODS y se almacenan en la ruta correspondiente del sistema de archivos EFS de Latinia. Posteriormente, una GAW se encarga de tomar el archivo generado y subirlo automáticamente a EFS para su posterior procesamiento.

# Arquitectura propuesta



# Descripcion del proceso

# Notificaciones en linea

# Notificaciones Batch

El proceso de notificaciones masivas está diseñado para enviar mensajes automáticos a los clientes asociados a los siguientes nemónicos:

* TCRMC: Notificación de tarjetas empresariales
* NPGTC: Notificación de puntos ganados
* NAPTC: Notificación de agradecimiento de pago
* RETCC: Aviso de renovación entregada por courier
* RETCO: Aviso de renovación con entrega en oficina

Como parte de la modernización del sistema, estos procesos han sido migrados desde paquetes DTSX hacia ETL desarrollados en AWS Glue con PySpark.

La lógica funcional se mantiene, continuando con la búsqueda de datos en las fuentes basadas en Sybase, pero ahora incorporando la conexión directa con el nuevo ODS operacional DB\_TC\_ODS para garantizar consistencia y trazabilidad en la generación de notificaciones, además, de obtener datos de cliente en Sybase, base de datos cl\_ente,cl\_direccion\_email.

A continuación, se detalla el funcionamiento específico de cada proceso ETL correspondiente a las notificaciones masivas. Cada flujo ha sido adaptado a la nueva arquitectura sobre AWS Glue con PySpark, conservando la lógica funcional existente, pero integrando las nuevas fuentes de datos del ODS operacional.

# Notificación de agradecimiento de pago

|  |  |
| --- | --- |
| Campo | Detalle |
| NEMÓNICO | NAPTC |
| Nombre del proceso | Agradecimiento de pago |
| Nombre del ETL | job\_emi\_noti\_agradec |
| Frecuencia de ejecución | diario |
| Tipo de archivo generado | .build / .ready |
| Nombre ejemplo de archivo | NAPTC202505271230010001.build |
| Notificación | **Datos** |
| Agradecimiento de pago - NAPTC | Nombre del plástico |
| Valor de la transacción |
| Fecha de proceso |
| Saldo pago de contado |

1. Se consulta la tabla adq\_tc\_m\_catalogos para obtener el valor correspondiente al catálogo 'NEMONICO\_NOTIFICACION' con código\_banco = 1, cuyo valor esperado es 'NAPTC'.
2. Se consulta el id de la fecha de proceso consultando la tabla emi\_ods\_fecha transformando la fecha de proceso pasada como parámetro con convert(datetime,@e\_fecha,101) filtrando con el valor de d\_fecha se guarda en una variable @i\_id\_fecha = i\_id\_fecha
3. Se recuperan los estados que excluyen el envío de esta notificación desde el catálogo 'TCR\_EXCL\_NOTI\_AGRA' en la misma tabla adq\_tc\_m\_catalogos. (Ver Anexo 6.11)
4. Se consulta el catálogo ‘TipoFactura’ en adq\_tc\_m\_catalogos donde el valor de s\_codigo\_banco sea igual a 67 (tipo factura de pagos) y el valor se guarda en la variable @v\_cod\_transac
5. Se consulta el catalogo ‘EstExcNotiAgra’ en adq\_tc\_m\_catalogos donde se listan los estados de cuenta que no se toman en cuenta para el envio de la notificación (Ver anexo 7.15)
6. Se consultan los datos de cuenta,monto\_trans i\_id\_fec\_proceso,i\_id\_fec\_tran agregando en la tabla temporal #tmp\_pagos de la tabla emi\_transac\_rotativas filtrando con el valor de i\_id\_fec\_proceso sea igual a @v\_id\_fecha\_proceso y el i\_id\_tip\_trans sea igual al valor de @v\_cod\_transac y el i\_num\_cuenta sea mayor a 0
7. se obtienen los siguientes datos cuenta,monto,id\_fec\_proc,id\_fec\_tran consultando la tabla temporal tmp\_pagos uniendo la tabla emi\_cuenta\_cliente tomando el valor de la cuenta sea igual al valor de p\_cuenta uniendo con la tabla emi\_ods\_cliente donde el valor de emi\_cuente\_cliente.i\_cod\_cliente sea igual al valor de emi\_ods\_cliente.i\_cod\_cliente uniendo con la tabla emi\_maestro\_cartera\_diaria donde el valor de tmp\_pagos.cuenta sea igual al valor de emi\_maestro\_cartera\_diaria.i\_num\_cuenta y el valor de emi\_maestro\_cartera\_diaria.i\_id\_fecha\_proceso sea igual @v\_id\_fecha\_proceso filtrando los valores de emi\_cuenta\_cliente.i\_num\_bin no sea igual a 422249 y el valor de emi\_ods\_cliente.s\_tipo\_identifica este entre (‘C’ y ‘P’) y el valor de ods\_cuenta\_cliente.i\_id\_estado no este entre los valores del catalogo EstExcNotiAgra y donde el valor de emi\_maestro\_cartera\_diaria.i\_num\_pago\_vcdos sea menor o igual al valor de adq\_tc\_m\_parametros valor CPAEX. El resultado del Query se almacena en la tabla temporal #tmp\_pago2
8. se consultan los siguientes datos cuenta, numero de pago que es un contador monto\_total que es la suma del campo monto, fec\_tran que se obtiene realizando una subquery en la tabla ods\_fecha donde el campo emi\_ods\_fecha.i\_id\_fecha sea igual al valor máximo de tmp\_pagos2.i\_id\_fec\_tran,fecha\_proceso igual a @v\_fecha\_proceso desde la tabla #tmp\_pagos2 agrupando por cuenta. El resultado se agrega en la tabla temporal #tmp\_dato\_agrp
9. se consultan los siguientes campos cuenta,identifica,tipo\_identifica desde la tabla #tmp\_dato\_agrp realizando un join emi\_cuenta\_cliente donde el valor de #tmp\_dato\_agrp.cuenta sea igual al valor de emi\_cuenta\_cliente.i\_num\_cuenta realizando un join con la tabla emi\_ods\_cliente.i\_cod\_cliente sea igual al valor de emi\_cuenta\_cliente.i\_cod\_cliente. El resultado de la Query se guarda en la tabla #tmp\_cliente
10. se consulta la cuenta, nombrePlastico (emi\_t\_medio\_pago\_tarjeta. mp\_nombenred) y descripcion del bin (adq\_tc\_c\_bin\_propio.bp\_descripcion) desde la tabla #tmp\_dato\_agrp realizando un join a la tabla emi\_tarjeta\_propia donde el valor de #tmp\_dato agrp.cuenta sea igual a emi\_tarjeta\_propia realizando un inner join con la tabla adq\_tc\_c\_bin\_propio donde el valor de adq\_tc\_c\_bin\_propio.bp\_s\_num\_bin sea igual a #tmp\_dato\_agrp.s\_num\_tarjeta(primeros 6 digitos BIN) validando que el valor de emi\_tarjeta\_propia.s\_tipo\_tarjeta sea igual a ‘P’ y el emi\_tarjeta\_propia.i\_cod\_cliente no sea igual a 0. El resultado de la consulta se guarda en la tabla #tmp\_tarjeta
11. Se consulta el saldo pago minimo,saldo actual rotativo,saldo actual diferido,saldo anterior rotativo, suma de créditos(pago mes + credito del mes mas devoluciones del mes), fecha tope de pago y fecha de corte de la tabla #tmp\_dato\_agrp realizando un JOIN con emi\_maestro\_cartera\_diaria donde la cuenta de #tmp\_dato\_agrp.agrp\_cuenta este en emi\_maestro\_cartera\_diaria.i\_num\_cuenta y emi\_maestro\_cartera\_diaria.i\_id\_fec\_proceso sea igual al valor de @v\_is\_fecha\_proceso, se realiza un join con la emi\_ods\_fecha comparando el valor de i\_id\_fec\_cor\_act sea igual al campo i\_id\_fecha, el resultado se guarda en la tabla temporal #tmp\_cartera.
12. Con las tablas #tmp\_dato\_agrp,#tmp\_cliente,#tmp\_tarjeta y #tmp\_cartera se realiza un insert en la tabla emi\_t\_noti\_agradec. El mapeo se realiza en las siguientes tablas (Ver anexo 7.16 )
13. Por cada registro en la tabla emi\_t\_noti\_agradec, se consulta con el valor s\_identifica la tabla cl\_ente que se encuentra en ambiente Sybase base de datos cobis usando la columna en\_ced\_ruc, de esta consulta se obtiene el valor de en\_ente. Se realiza un inner join con la tabla cl\_direccion\_email con el valor de\_ente con el valor en\_ente de\_descripcion, se obtiene el utlimo registro con la fecha de modificación ultima de\_fecha\_modificacion
14. Se registran los datos en la tabla emi\_t\_noti\_agradec con los datos de i\_codigo\_mis y s\_correo
15. Se tratan los registros que se tienen en la tabla y se genera el archivo .build (Ver anexo 6.13). se valida que el numero de registros en el archivo .build no supere los 20000 registros.en caso de que sobrepasen los 20000 registros, se debe crear otro archivo aparte agregando el numero secuencial del archivo al final con 2. El valor del limite de registros por archivos .build se encuentra parametrizado en el valor NOTI\_BATCH\_LIMI
16. Se sube el archivo en el s3 habilitado y una GAW mueve el archivo a la fuente de recepción de latinia

# Notificación de Puntos Ganados

|  |  |
| --- | --- |
| Campo | Detalle |
| NEMÓNICO | NPGTC |
| Nombre del proceso | Puntos ganados |
| Nombre del ETL | job\_emi\_noti\_ptos |
| Frecuencia de ejecución | diario |
| Parametros de ETL | Tipo\_proceso=B,dia |
| Tipo de archivo generado | .build / .ready |
| Nombre ejemplo de archivo | NPGTC202505271230010001.build |

1. Se consulta la tabla adq\_tc\_m\_catalogos para obtener el valor correspondiente al catálogo 'NEMONICO\_NOTIFICACION' con código\_banco = 2, cuyo valor esperado es 'NPGTC'.
2. Se genera la fecha de proceso del job en formato dd/mm/AAAA, parámetro tipoProceso B y días -7
3. Se ejecuta el stored procedure pa\_tcre\_cciclofact pasando como parámetros tipoProceso,fechaProceso y días (ver anexo 7.17)
4. Se valida que el valor devuelto por el pa\_tcre\_cciclofact, entonces se llama al stored procedure pa\_con\_ppuntosmcvs (Ver anexo 7.19) pasando como parámetro la fecha de proceso y si el segundo el ciclo es igual a 2, se manda el segundo parámetro como ‘VISA’. El stored procedure llena la tabla emi\_tc\_datos\_masivos\_puntos
5. Se consultan los datos de entrada de las tablas cobis..cl\_ente y cl\_direccion\_email tomando como valor de filtro el numero de identificacion para en\_ced\_ruc el numero de identificacion de la tabla emi\_tc\_datos\_masivos\_puntos.identificacion y donde el valor de en\_subtipo no sea igual a ‘P’
6. Se cargan los registros en la tabla emi\_tc\_datos\_masivos\_puntos con los datos de direcciones de cliente
7. Se tratan los registros que se tienen en la tabla emi\_tc\_datos\_masivos\_puntos y se genera el archivo .build (Ver anexo 7.13). se valida que el numero de registros en el archivo .build no supere los 20000 registros. En caso de que sobrepasen los 20000 registros, se debe crear otro archivo aparte agregando el numero secuencial del archivo al final con 2. El valor del limite de registros por archivos .build se encuentra parametrizado en el valor NOTI\_BATCH\_LIMI
8. Se sube el archivo en el s3 habilitado y una GAW mueve el archivo a la fuente de recepción de latinia

# Notificacion de tarjetas empresariales

|  |  |
| --- | --- |
| Campo | Detalle |
| NEMÓNICO | TCRMC |
| Nombre del proceso | Tarjetas empresariales |
| Nombre del ETL | job\_emi\_t\_noti\_tcempr |
| Frecuencia de ejecución | diario |
| Tipo de archivo generado | .build / .ready /.csv |
| Nombre ejemplo de archivo | TCRMC202505271230010001.build |

1. Se consulta la tabla adq\_tc\_m\_catalogos para obtener el valor correspondiente al catálogo 'NEMONICO\_NOTIFICACION' con código\_banco = 3, cuyo valor esperado es 'TCRMC'.
2. Se ejecuta el stored procedure pa\_con\_nnotiempresarial (Ver anexo 7.20) pasando como parámetro el valor de la fecha del proceso en formato AAAA/mm/dd
3. Se trata la salida del stored procedure categorizando por el numero de prefijo utlizado en la salida. Si es 1 es el nombre del archive que se va a generar; Si es 2 es el contenido del archivo .build; si es 3,son los metadatos para relacionar cada registro en el archive .build con el archive .csv que contiene los datos de los empleados por empresas. Se adjunta el contenido y se sube a la ruta habilitada por latinia para la recepcion de mensajes.

# Notificacion de renovaciones de tarjeta

# Afectaciones

# Requerimientos no funcionales

# Tecnologías usadas

* Lenguajes de programación: Python 3.10
* Librerias: requests, boto3, pymysql
* Aws Glue Job
* Aws lambda
* Aws Cron Schedule
* Amazon Api Gateway
* Cloudwatch

# Esquemas de seguridad

El proceso de Débitos Automáticos ha sido diseñado conforme a los estándares de seguridad de la información definidos por el Banco Bolivariano y en alineación con las mejores prácticas de arquitectura sobre Amazon Web Services (AWS). A continuación, se detallan los mecanismos de seguridad aplicados a cada componente de la solución, con el objetivo de garantizar la confidencialidad, integridad, disponibilidad y trazabilidad de la información procesada.

# Cifrado de información

**Datos en tránsito:** Toda comunicación entre los servicios desplegados sobre AWS se encuentra cifrada mediante el protocolo TLS 1.2 o superior, asegurando la integridad y confidencialidad de los datos transferidos entre los buckets de Amazon S3, funciones Lambda, AWS Glue, Aurora MySQL y otros servicios interconectados.

Datos en reposo: Los datos almacenados en Amazon S3 y Amazon Aurora MySQL están cifrados utilizando claves administradas por el banco a través de AWS Key Management Service (KMS). Este cifrado aplica a archivos operativos, tablas transaccionales (TB\_SOLICITUD\_DEBITO, TB\_RESULTADO\_DEBITO) y registros en bases de datos históricas y de control (TB\_AFILIACIONES, TB\_CUARENTENA).

# Gestión de identidades y accesos (IAM)

Se aplica el principio de mínimo privilegio mediante la implementación de roles IAM específicos para cada servicio:

* Las funciones Lambda poseen únicamente permisos estrictamente necesarios sobre recursos puntuales (por ejemplo, lectura de un bucket determinado, escritura en Aurora).
* Los procesos de transformación y catalogación (Glue Jobs) acceden exclusivamente a bases y tablas autorizadas dentro del AWS Glue Data Catalog. Se controla el acceso administrativo a la consola de AWS mediante autenticación multifactor (MFA) y políticas condicionales, registradas en AWS CloudTrail.

# Autenticación de servicios externos

Las interfaces expuestas para la recepción de archivos o integración con terceros (ej. instituciones externas como Minsait) deben incorporar mecanismos robustos de autenticación, incluyendo:

* Autenticación basada en OAuth 2.0 o tokens JWT firmados.
* Validación de origen mediante certificados digitales o restricciones de IP.
* Políticas de validación de estructura y consistencia antes de iniciar el procesamiento.
* Monitoreo, Registro y Auditoría

La arquitectura contempla una trazabilidad completa de las actividades mediante:

AWS CloudTrail, para el registro de acciones sobre los recursos AWS (accesos, modificaciones, ejecuciones).

AWS CloudWatch Logs, configurado para monitorear ejecuciones de funciones Lambda, Glue Jobs, accesos a S3 y conexiones a la base de datos Aurora.

Publicación de eventos críticos a través de Amazon SNS, permitiendo notificaciones automatizadas al área de operaciones en caso de fallas, rechazos de archivos o interrupciones en el flujo.

# Validación e integración de archivos

Los archivos procesados deben ser validados estructural y referencialmente. En caso de inconsistencias, los mismos son derivados a un bucket de cuarentena o almacenados en la tabla TB\_CUARENTENA, con mecanismos de seguridad activos (cifrado, control de acceso). Se recomienda el uso de hashes SHA-256 para validar integridad de los archivos durante su carga.

La validación se ejecuta en múltiples puntos del proceso para evitar el procesamiento de datos maliciosos o alterados.

# Disponibilidad y recuperación

La base de datos Amazon Aurora MySQL está desplegada en modo Multi-AZ, asegurando alta disponibilidad y conmutación automática ante fallos. Se encuentra habilitado el respaldo automático de base de datos con retención de hasta 14 días, permitiendo la restauración ante fallos operativos o lógicos.

Se recomienda mantener una política de replicación asincrónica para entornos de contingencia o análisis offline.

# Gobierno y ciclo de vida y trazabilidad

Los buckets de S3 tienen habilitado el versionado de objetos, facilitando la trazabilidad de archivos y la recuperación de versiones anteriores ante eventos no deseados. Las modificaciones de código fuente (Lambdas, Glue, definiciones de tablas) se gestionan mediante flujos CI/CD integrados al repositorio institucional, con control de cambios, revisiones y aprobaciones conforme a las políticas internas del banco.

# Restricciones de entorno

Los entornos de desarrollo, pruebas y producción están completamente segregados, asegurando la integridad de los datos productivos y evitando accesos cruzados indebidos. Los recursos productivos están incorporados al inventario de activos del Banco Bolivariano, bajo custodia del área de infraestructura y seguridad TI.

# Política de calidad de datos

# Consideraciones generales

Por tratarse de información batch procesada durante el día en la plataforma de MINSAIT bajo la operativa de varios subsistemas que interactúan para los diferentes procesos, y cuya información no está disponible en su totalidad para el Banco, se va a excluir validaciones de cálculos manteniéndose únicamente validaciones de atributos (integridad de la información).

Es indispensable asegurar el mantenimiento y actualización de estas estructuras a fin evitar el rechazo de registros por nuevos atributos y estructuras no actualizados oportunamente.

Se debe asegurar la incorporación de validaciones de datos en el diseño del ODS para detectar y manejar datos incompletos o incorrectos. Esto puede incluir chequeos de integridad, reglas de negocio, y validaciones de formato.

Establecer que todos los datos sensibles, como información personal identificable (PII), cumplan con las regulaciones de privacidad aplicables, como GDPR (Reglamento General de Protección de Datos), antes de ser almacenados en el ODS.

# Consideraciones especificas

Implementar un control de versiones de la estructura de los archivos para detectar cambios inesperados en los atributos durante el proceso de carga y garantizar que se está cargando la versión correcta del arch

# Acceso a la información

Sujeto a estándares, procedimientos y esquema de seguridad del Banco Bolivariano para procesos batch y manejo de información en la nube.

# Datos de auditoria

1. Registrar todas las operaciones de carga, incluyendo detalles sobre el origen de los datos, los usuarios que iniciaron el proceso, y cualquier modificación realizada.
2. Implementar un sistema que permita rastrear cambios en los datos, quién los realizó y cuándo.
3. Generar informe de auditoría que permita revisar las operaciones y detectar posibles anomalías o violaciones de políticas.
4. En base a la política definida por Banco, para cada uno de los eventos almacenados se registra la siguiente información: usuario, tipo de evento, fecha y hora, resultado (exitoso o fallido), origen del evento, datos o componentes afectados, IP de origen de la transacción.
5. El contenido del log de errores debe incluir la información correspondiente a la ejecución de cada proceso. Esto incluye, para cada proceso: el nombre del programa, fecha de inicio, fecha de finalización, descripción del error, detalle del campo afectado y el valor que se estaba procesando en el momento del error.
6. En cuanto al log general, su contenido dependerá del tipo de error reportado. Por ejemplo, si se trata de un error relacionado con cuotas, debe incluir el número de la cuota afectada. El log debe contener variables específicas para cada tipo de error y para cada interfaz, además de incluir datos básicos que permitan identificar y analizar los incidentes de manera efectiva.

# Casos de contingencia

# Consideraciones generales

Sujeto a estándares y procedimientos de manejo de contingencias del Banco, contemplando acuerdos de niveles de atención de acuerdo a la criticidad de la novedad:

* Novedad Leve: Esquema de respaldos totales e incrementales para contar con un punto de retorno en caso de errores en la data o en el procesamiento.
* Novedad Grave: Evaluar la inclusión del esquema de Medios de Pago en el Plan de Recuperación de Desastres del Banco.
* El método de contingencia por parte del MSP depende de si existen dependencias con otro archivo. En caso de que falte un archivo necesario, se debe escalar la situación enviando un correo a Credimatic informando la ausencia. Es fundamental identificar primero si el archivo tiene dependencias o si puede procesarse de manera independiente para determinar el curso de acción adecuado.

# Consideraciones especificas

1. Realizar copias de seguridad regulares de los datos tanto del fichero plano como del ODS. Esto es esencial para poder restaurar los datos en caso de fallos durante la carga.
2. Elaborar documento técnico que detalle cómo restaurar los datos y reanudar las operaciones en caso de una interrupción.
3. Implementar un sistema de versionado que permita identificar qué versión de las estructuras ha sido cargada en cada ejecución del proceso, facilitando la recuperación de versiones anteriores si es necesario.

# Manejo de excepciones

# Consideraciones generales

Diseñar procesos para manejar datos que no cumplen con los criterios de calidad, como rechazar registros, corregir datos automáticamente (cuando sea posible), o redirigir datos problemáticos a cuarentena para revisión manual.

# Consideraciones especificas

1. Implementar un sistema como cloudwatch o X-Ray que registre todos los errores que ocurren durante el proceso de carga, con suficiente detalle para facilitar la resolución de problemas.
2. Configurar alertas automáticas que notifiquen al equipo de TI o a los responsables del
3. En caso de errores temporales, por ejemplo, pérdida de conexión, implementar mecanismos que automaticen el reintento de la operación. Este proceso se podrá re ejecutar automáticamente hasta tres (3) veces, de existir un error por cuarta ocasión se deberá notificar al Operador de TI.

# Anexos

# Json de notificaciones de consumo por internet

|  |
| --- |
| **Consumo por Internet** |
| {  "header": {  "id": "CONIN20250527143000000001",  "refCompany": "BOLIVARIANO",  "refService": "CONIN",  "keyValue": "847722",  "channels": "PNK",  "refMsgLabel": "Avisos24"  },  "info": {  "loginEnterprise": "BOLIVARIANO",  "refContract": "CONIN"  },  "data": {  "tipotrj": "Visa",  "valor": "89.99",  "numtrj": "\*\*\*\* \*\*\*\* \*\*\*\* 0101",  "identi": "0987654321",  "des\_transaccion": "Compra en Amazon",  "estado\_pais": "Guayas",  "nom\_pais": "Ecuador",  "fecha": "2025-05-27",  "hora": "10:45:00",  "nombre\_cliente": "Luis Martínez",  "motivo": "Compra en línea",  "canal": "PNK"  },  "addresses": [  {  "className": "Email",  "type": "Personal",  "ref": "cliente@example.com"  }  ]  } |

# Json de notificaciones de consumo por internet Tc adicional

|  |
| --- |
| **Consumo por internet Tc adicional** |
| {  "header": {  "id": "CONI220250527143000000001",  "refCompany": "BOLIVARIANO",  "refService": " CONI2",  "keyValue": "847723",  "channels": "PNK",  "refMsgLabel": "Avisos24"  },  "info": {  "loginEnterprise": "BOLIVARIANO",  "refContract": "CONI2"  },  "data": {  "tipotrj": "Visa",  "valor": "89.99",  "numtrj": "\*\*\*\* \*\*\*\* \*\*\*\* 0101",  "identi": "0987654321",  "des\_transaccion": "Compra en Amazon",  "estado\_pais": "Guayas",  "nom\_pais": "Ecuador",  "fecha": "2025-05-27",  "hora": "10:45:00",  "nombre\_cliente": "Luis Martínez",  "motivo": "Compra en línea",  "canal": "PNK"  },  "addresses": [  {  "className": "Email",  "type": "Personal",  "ref": "cliente@example.com"  }  ]  } |

# Json de notificación de límite de cupo alcanzado

|  |
| --- |
| Notificación de límite de cupo alcanzado |
| {  "header": {  "id": "LIMCA20250527143000000001",  "refCompany": "BOLIVARIANO",  "refService": "LIMCA",  "keyValue": "847722",  "channels": "Web",  "refMsgLabel": "Avisos24"  },  "info": {  "loginEnterprise": "BOLIVARIANO",  "refContract": "LIMCA"  },  "data": {  "tipotrj": "Visa",  "numtrj": "\*\*\*\* \*\*\*\* \*\*\*\* 9999",  "identi": "0944455566",  "des\_comercio": "Farmacia SanaSana",  "motivo": "Límite de cupo alcanzado",  "nombre\_titular": "Ana Torres",  "fecha": "2025-05-27",  "hora": "13:05:00",  "nombre\_cliente": "Ana Torres"  },  "addresses": [  {  "className": "Email",  "type": "to",  "ref": "cliente@example.com"  }  ]  } |

# Json de notificación de Bloqueo definitivo de tarjeta de credito

|  |
| --- |
| **Bloqueo definitivo de la cuenta** |
| {  "header": {  "id": "TCBLQ20250527143000000001",  "refCompany": "BOLIVARIANO",  "refService": "TCBLQ",  "keyValue": "847722",  "channels": "WAP",  "refMsgLabel": "Avisos24"  },  "info": {  "loginEnterprise": "BOLIVARIANO",  "refContract": "TCACT"  },  "data": {  "tipotrj": "VSA",  "numtrj": "\*\*\*\* \*\*\*\* \*\*\*\* 3456",  “motivo”:”robo”,  "identi": "09\*\*\*\*\*\*55",  "nombre\_titular": "Pedro Sánchez",  "fecha": "2025-05-27",  "hora": "08:20:00",  "nombre\_cliente": "Pedro Sánchez",  "motivo": "Activación exitosa",  "canal": "WEB"  },  "addresses": [  {  "className": "email",  "type": "to",  "ref": "cliente@example.com"  }  ]  } |

# Json de notificación de bloqueo temporal de tarjeta de credito

|  |
| --- |
| **Notificacion de bloqueo temporal de tarjeta de credito** |
| {  "header": {  "id": "BQTCT20250527143000000001",  "refCompany": "BOLIVARIANO",  "refService": "BQTCT",  "keyValue": "847722",  "channels": "WAP",  "refMsgLabel": "Avisos24"  },  "info": {  "loginEnterprise": "BOLIVARIANO",  "refContract": "BQTCT"  },  "data": {  "tipotrj": "VSA",  "numtrj": "\*\*\*\* \*\*\*\* \*\*\*\* 3456",  “motivo”:”viaje”,  "identi": "09\*\*\*\*\*\*55",  "nombre\_titular": "Pedro Sánchez",  "fecha": "2025-05-27",  "hora": "08:20:00",  "nombre\_cliente": "Pedro Sánchez",  "motivo": "Activación exitosa",  "canal": "WAP"  },  "addresses": [  {  "className": "email",  "type": "to",  "ref": "cliente@example.com"  }  ]  } |

# Json de notificación de negación de bloqueo temporal

|  |
| --- |
| **Notificacion de negación de bloqueo temporal** |
| {  "header": {  "id": "ERRTC20250527143000000001",  "refCompany": "BOLIVARIANO",  "refService": "ERRTC",  "keyValue": "847722",  "channels": "WAP",  "refMsgLabel": "Avisos24"  },  "info": {  "loginEnterprise": "BOLIVARIANO",  "refContract": "ERRTC"  },  "data": {  "tipotrj": "VSA",  "numtrj": "\*\*\*\* \*\*\*\* \*\*\*\* 3456",  "identi": "09\*\*\*\*\*\*55",  "nombre\_titular": "Pedro Sánchez",  "fecha": "2025-05-27",  "hora": "08:20:00",  "nombre\_cliente": "Pedro Sánchez",  “des\_transaccion”:”bloqueo de tarjeta negado”  },  "addresses": [  {  "className": "email",  "type": "to",  "ref": "cliente@example.com"  }  ]  } |

# Desbloqueo temporal de tarjeta de credito

|  |
| --- |
| **Notificacion de debloqueo temporal de tarjeta de credito** |
| {  "header": {  "id": "DESTC20250527143000000001",  "refCompany": "BOLIVARIANO",  "refService": "DESTC",  "keyValue": "847722",  "channels": "WAP",  "refMsgLabel": "Avisos24"  },  "info": {  "loginEnterprise": "BOLIVARIANO",  "refContract": "BQTCT"  },  "data": {  "tipotrj": "VSA",  "numtrj": "\*\*\*\* \*\*\*\* \*\*\*\* 3456",  “motivo”:”viaje”,  "identi": "09\*\*\*\*\*\*55",  "nombre\_titular": "Pedro Sánchez",  "fecha": "2025-05-27",  "hora": "08:20:00",  "nombre\_cliente": "Pedro Sánchez",  "motivo": "desbloqueo de tarjeta",  "canal": "WAP"  },  "addresses": [  {  "className": "email",  "type": "to",  "ref": "cliente@example.com"  }  ]  } |

# Json de notificación de activación de tarjeta de credito

|  |
| --- |
| **Notificación de activación de tarjeta de credito** |
| {      "header": {        "id": "TACT20250527143000000001",        "refCompany": "BOLIVARIANO",        "refService": "TACT",        "keyValue": "847722",        "channels": "WAP",        "refMsgLabel": "Avisos24"      },      "info": {        "loginEnterprise": "BOLIVARIANO",        "refContract": "TACT"      },      "data": {        "tipotrj": "Visa",  "identi": "09\*\*\*\*\*\*33",        "numtrj": "\*\*\*\* \*\*\*\* \*\*\*\* 0101",        "canal": "VEN",  “plazo”:8,  “fecha”:”17-05-25”      },      "addresses": [        {          "className": "Email",          "type": "to",          "ref": "cliente@example.com"        }      ]    } |

# Json de notificación de Precancelacion de diferidos

|  |
| --- |
| **Notificacion de precancelacion de diferidos** |
| {      "header": {        "id": "TCPDD20250527143000000001",        "refCompany": "BOLIVARIANO",        "refService": "TCPDD",        "keyValue": "847722",        "channels": "WAP",        "refMsgLabel": "Avisos24"      },      "info": {        "loginEnterprise": "BOLIVARIANO",        "refContract": "TCPDD"      },      "data": {        "tipotrj": "Visa",        "valor": "89.99",  "identi": "09\*\*\*\*\*\*33",        "numtrj": "\*\*\*\* \*\*\*\* \*\*\*\* 0101",        "canal": "VEN",  “plazo”:8      },      "addresses": [        {          "className": "Email",          "type": "to",          "ref": "cliente@example.com"        }      ]    } |

# Json de notificación de pago de tarjeta de credito

|  |
| --- |
| **Notificacion de pago de tarjeta de credito** |
| {      "header": {        "id": "PAGTC20250527143000000001",        "refCompany": "BOLIVARIANO",        "refService": "PAGTC",        "keyValue": "847722",        "channels": "VEN",        "refMsgLabel": "Avisos24"      },      "info": {        "loginEnterprise": "BOLIVARIANO",        "refContract": "PAGTC"      },      "data": {        "tipotrj": "Visa",        "valor": "89.99",        "numtrj": "\*\*\*\* \*\*\*\* \*\*\*\* 0101",        "canal": "VEN"      },      "addresses": [        {          "className": "Email",          "type": "to",          "ref": "cliente@example.com"        }      ]    } |

# Estados excluyentes por cuentas de tarjeta

|  |  |
| --- | --- |
| Código | Descripción |
| 0000070000 | Castigado Reestructurado lega |
| 0000060000 | Castigado Reestructurado Norm |
| 0000030000 | Castigo Automático |
| 0000040000 | Castigo Automático en Legal |
| 0000010000 | Castigo de Cartera (3-B) |
| 0000020000 | Castigo de Cartera (Saldo 0) |
| 0600000000 | Cliente en Jurídico |
| 0010000000 | Cliente en Pre-Legal (3-8) |
| 0100000000 | Cuenta en Legal (3-A) |
| 0200000000 | Demandado |
| 0500000000 | Demandado (previa reestructuración) |
| 0020000000 | En Dpto. Legal sin documentación |
| 0400000000 | Legal (previa reestructuración) |
| 0040000000 | Pre-Legal previa restructuración |
| 0300000000 | Reestructuración deuda en legal |
| 0030000000 | Reestructuración deuda prelegal |
| 0000050000 | Venta de Cartera Castigada |
| 0000080000 | Castigado Refinanciado |
| 0000090000 | Castigo Refinanciado Demandado |
| 0700000000 | Refinanciado Demandado |

# Diccionario de datos tabla emi\_t\_noti\_agradec

|  |  |  |
| --- | --- | --- |
| Nombre del Campo | Tipo de Dato | Descripción |
| i\_cod\_mis | int | Código del envío o misión de la notificación. |
| s\_correo | varchar(100) | Correo electrónico del cliente. |
| i\_num\_cuenta | bigint | Número de cuenta asociada a la tarjeta o producto. |
| s\_tipo\_identifica | varchar(10) | Tipo de documento de identificación (ej. CI, RUC, PAS). |
| s\_identifica | varchar(20) | Número de identificación del cliente. |
| s\_nom\_plas | varchar(50) | Nombre del cliente impreso en el plástico de la tarjeta. |
| s\_descrip\_bi | varchar(50) | Descripción del BIN o producto de tarjeta (ej. VISA Platinum). |
| d\_fecha\_proceso | date | Fecha en que se procesó la notificación. |
| i\_num\_pago | int | Número o secuencia de pago asociado. |
| m\_monto | decimal(18,2) | Monto total pagado por el cliente. |
| m\_saldo\_pm | decimal(18,2) | Saldo del pago mínimo pendiente. |
| m\_saldo\_rotativo | decimal(18,2) | Saldo rotativo actual del cliente. |
| m\_saldo\_diferido | decimal(18,2) | Saldo diferido vigente (cuotas pendientes). |
| m\_pago\_contado | decimal(18,2) | Monto pagado al contado. |
| m\_sum\_credito | decimal(18,2) | Suma de créditos o abonos aplicados. |
| m\_saldo\_pc | decimal(18,2) | Saldo de pago al contado o total a pagar. |
| d\_fec\_top\_pag | date | Fecha tope de pago para evitar intereses. |
| d\_fec\_corte | date | Fecha de corte del ciclo de facturación. |
| d\_fecha\_transac | date | Fecha de la transacción o del movimiento procesado. |

# Ejemplos de generación de archivos .build

|  |  |
| --- | --- |
| **Notificaciones batch** | **Archivo de muestra** |
| Agradecimiento de pago |  |
| Notificación de puntos |  |
| Notificación tarjetas empresariales |  |
| Archivo adjunto notificación tarjeta empresarial |  |

# Parametros usados en el proyecto

|  |  |  |  |
| --- | --- | --- | --- |
| Parámetro | Tabla / Fuente | Tipo de Dato Estimado | Descripción |
| TIPOFAC\_PAGOTAR | adq\_tc\_m\_parametros | int | Código del tipo de factura para pago con tarjeta. Valor referencial: 67. |
| CPAEX | adq\_tc\_m\_parametros | int | Límite máximo permitido para ic\_numrecimp en contratos impagados. |
| BIN\_EXCLUIDO | adq\_tc\_m\_parametros | varchar(6) | BIN de tarjeta prepago excluido. Valor usado: '422249'. |
| NOTI\_BATCH\_LIMI | adq\_tc\_m\_parametros | int | Límite máximo de registros por archivo .build, e.g., 20000. |
| BINES\_VISA | adq\_tc\_m\_parametros | varchar | “01, 31, 30” |
| BINES\_MASTERCARD | Adq\_tc\_m\_parametros | varchar | “03” |

# Estados excluidos notificación agradecimiento de pago

|  |  |
| --- | --- |
| Estado | Descripción |
| Castigado | Castigo de Cartera (3-B) |
| Castigado | Castigo de Cartera con Saldo 0 |
| Castigado | Castigo Automático |
| Castigado | Castigo Automático con situación Legal |
| Castigado | Venta de Cartera Castigada |
| Castigado | Castigado Reestructurado Normativo |
| Castigado | Castigado Reestructurado Legal |
| Castigado | Castigado Refinanciado |
| Castigado | Castigo Refinanciado en Proceso Judicial (Demanda) |
| Prejuridico | Cliente en Prejurídico (3-8) |
| Prejuridico | En Departamento Legal sin Documentación |
| Prejuridico | Reestructuración de Deuda en Prejurídico |
| Prejuridico | Prejurídico con Reestructuración Previa |
| Legal | Cuenta en Legal (3-A) |
| Legal | Cliente Demandado |
| Legal | Reestructuración de Deuda en Legal |
| Legal | Legal con Reestructuración Previa |
| Legal | Demandado con Reestructuración Previa |
| Legal | Cliente en Jurídico |
| Legal | Refinanciado en Proceso Judicial (Demanda) |

# Mapeo de datos de tablas temporales hacia tabla emi\_p\_noti\_agradec

|  |  |  |  |
| --- | --- | --- | --- |
| Campo Destino (tcr\_tmp\_agradec) | Alias SQL | Campo Origen | Tabla Origen |
| i\_num\_cuenta | CUENTA\_TC | cuenta | #tmp\_dato\_agrp (alias agrp) |
| s\_tipo\_identifica | TI\_IDENT | tipo\_identifica | #tmp\_cliente (alias cl) |
| s\_identifica | CO\_IDENT | identifica | #tmp\_cliente (alias cl) |
| s\_nom\_plas | NOM\_PLAS | s\_nom\_plas1 | #tmp\_tarjeta (alias tp) |
| s\_descrip\_bi | DESC\_BIN | s\_descripcion | #tmp\_tarjeta (alias tp) |
| d\_fecha\_proceso | FCH\_PRO | fec\_proc | #tmp\_dato\_agrp (alias agrp) |
| i\_num\_pago | NUM\_PAGO | numero\_pago | #tmp\_dato\_agrp (alias agrp) |
| m\_monto | MONTO\_PAGO | monto\_total | #tmp\_dato\_agrp (alias agrp) |
| m\_saldo\_pm | SALDO\_PM | m\_pago\_min | #tmp\_cartera (alias cd) |
| m\_saldo\_rotativo | SALDO\_ACTUAL | m\_sald\_act\_rot | #tmp\_cartera (alias cd) |
| m\_saldo\_diferido | SALDO\_DIFERIDO | m\_sald\_act\_dif | #tmp\_cartera (alias cd) |
| m\_pago\_contado | PAGO\_CONTADO | m\_sald\_ant\_rot | #tmp\_cartera (alias cd) |
| m\_sum\_credito | SUM\_CREDITOS | m\_sum\_creditos | #tmp\_cartera (alias cd) |
| d\_fec\_top\_pag | FCH\_TOPE | d\_fec\_top\_pag | #tmp\_cartera (alias cd) |
| d\_fec\_corte | FCHA\_CORTE | d\_fecha | #tmp\_cartera (alias cd) |
| d\_fecha\_transac | FCH\_TRX | fec\_tran | #tmp\_dato\_agrp (alias agrp) |

# Parámetros de entrada y salida pa\_tcre\_cciclofact

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nombre | Tipo de Dato | Tipo | Obligatorio | Descripción |
| @tipo\_proceso | VARCHAR(2) | Entrada | Sí | Tipo de proceso: 'A' (por fecha de cierre) o 'B' (por fecha vencimiento) |
| @fechaproceso | DATE | Entrada | Sí | Fecha base del proceso |
| @dia | INT | Entrada | Solo si @tipo\_proceso = 'B' | Días a sumar a @fechaproceso para calcular la fecha de vencimiento |
| ciclo | VARCHAR | Salida | No aplica | Código del ciclo de facturación |
| fecha | VARCHAR(20) | Salida | No aplica | Fecha de cierre del ciclo (formato MM/DD/YYYY) |
| fechaAntes | VARCHAR(20) | Salida | No aplica | Fecha de cierre del mes anterior del mismo ciclo |
| i\_mes\_cierre | INT | Salida | Solo si @tipo\_proceso = 'B' | Mes de cierre del ciclo actual |

# Diccionario de datos tabla emi\_t\_saldo\_puntos

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Campo | Descripción | Tipo de Dato | Longitud | Decimales |
| CODENTID | Entidad | Alfanumérico | 8 | — |
| CODPROGR | Programa | Alfanumérico | 8 | — |
| DESCCPRO | Descripción del programa | Alfanumérico | 20 | — |
| TIPOPROG | Tipo de programa (1 = consumos, 2 = pagos de consumos) | Alfanumérico | 1 | — |
| FECPROCES | Fecha de proceso | Alfanumérico | 10 | — |
| SALDOANT | Saldo anterior de puntos | Numérico | 13 | 0 |
| SALDOACT | Saldo actual de puntos | Numérico | 13 | 0 |
| ACUMULABLES | Puntos acumulables | Numérico | 13 | 0 |
| LIBERADOS | Puntos liberados | Numérico | 13 | 0 |
| GENERADOS | Puntos generados | Numérico | 13 | 0 |
| GANADOS | Puntos ganados | Numérico | 13 | 0 |
| PERDIDOS | Puntos perdidos | Numérico | 13 | 0 |
| RECOMPENSADOS | Puntos recompensados | Numérico | 13 | 0 |
| REDIMIDOS | Puntos canjeados | Numérico | 13 | 0 |
| DEVUELTOS\_CANJE | Puntos devueltos por canje | Numérico | 13 | 0 |
| AJUSTES\_POSITIVOS | Ajustes positivos | Numérico | 13 | 0 |
| AJUSTES\_NEGATIVOS | Ajustes negativos | Numérico | 13 | 0 |
| PUNTOS\_EFECTIVIZAR | Puntos retenidos | Numérico | 13 | 0 |
| EXPIRADOS | Puntos caducados | Numérico | 13 | 0 |
| PROXIMOS\_CADUCAR | Puntos próximos a caducar | Numérico | 13 | 0 |
| FPROXCADU | Fecha de próximos puntos a caducar | Alfanumérico | 10 | — |
| PROXIMOS\_CADUCAR1 | Puntos próximos a caducar 1 | Numérico | 13 | 0 |
| FPROXCADU1 | Fecha de próximos puntos a caducar 1 | Alfanumérico | 10 | — |
| PROXIMOS\_CADUCAR2 | Puntos próximos a caducar 2 | Numérico | 13 | 0 |
| FPROXCADU2 | Fecha de próximos puntos a caducar 2 | Alfanumérico | 10 | — |
| DISPONIBLES | Puntos disponibles | Numérico | 13 | 0 |
| SALDOANT\_GENERADOS | Saldo anterior generados | Numérico | 13 | 0 |
| SALDOACT\_GENERADOS | Saldo actual generados | Numérico | 13 | 0 |
| FECHA\_ULT\_ACTUALIZ\_SALDOS | Fecha de última actualización de saldos | Alfanumérico | 10 | — |
| FILLER | Relleno (campo reservado sin uso funcional) | Alfanumérico | 110 | — |

# Parametros de entrada y salida pa\_con\_ppuntosmcvs

|  |  |  |
| --- | --- | --- |
| Nombre del Parámetro | Tipo de Dato | Requerido |
| fechaproceso | DATE | Sí |
| i\_marca | VARCHAR(3) | Sí |
| FechaProceso | DATE | No |
| RegistrosProcesados | INTEGER | No |
| Mensaje | VARCHAR | No |

# Parametros de entrada y salida del stored procedure pa\_con\_nnotiempresarial

|  |  |  |  |
| --- | --- | --- | --- |
| Campo | Tipo de Dato | Valor por Defecto | Descripción |
| e\_fechaProceso | DATETIME | (Requerido) | Fecha de proceso para la cual se generarán las notificaciones. |
| e\_filename | VARCHAR(50) | 'TCRMC' | Prefijo para el nombre del archivo principal de control. |
| e\_fileadd | VARCHAR(50) | 'MDPMC' | Prefijo para los nombres de archivos CSV individuales. |
| e\_secuencial | VARCHAR(3) | 'TCR' | Prefijo para generar números secuenciales. |
| e\_nemonico | VARCHAR(8) | 'TCRMC' | Código nemónico usado en metadatos y headers. |
| dato | VARCHAR(800) | (Salida) | Línea de datos formateada que puede contener:• Headers de archivos (tipo 1)• Contenido de datos (tipo 2)• Metadatos XML (tipo 3). |