documento de diseño

Cargue Masivo Reglas Contables V1.0

Universidad Libre

Gestión de proyectos

**Contenido**

[1. Introducción 3](#_Toc208576966)

[1.1 Propósito del Documento 3](#_Toc208576967)

[1.2 Alcance 3](#_Toc208576968)

[2. Arquitectura de la Solución 4](#_Toc208576969)

[2.1 Visión General de la Arquitectura 4](#_Toc208576970)

[2.2 Componentes de la Solución 5](#_Toc208576971)

[3. Diseño de la Base de Datos 6](#_Toc208576972)

[3.1 Diagrama de Entidad-Relación (ERD) 6](#_Toc208576973)

[3.2 Estructura de Tablas 6](#_Toc208576974)

[3.2.1 TABREC – Reglas contables 6](#_Toc208576975)

[3.2.2 TTABREC – Temporal de reglas contables 7](#_Toc208576976)

[3.2.3 TABCOD – Tipos de segmentaciones 7](#_Toc208576977)

[3.2.4 TABLCI – Tipos de segmentaciones 7](#_Toc208576978)

[4. Reglas de Negocio y Lógica de Proceso 8](#_Toc208576979)

[5. Diagrama de Flujo del Sistema 10](#_Toc208576980)

[5.1 Diagrama del Proceso de Carga 10](#_Toc208576981)

[6. Integraciones 11](#_Toc208576982)

[6.1 Interfaces de usuario 11](#_Toc208576983)

[6.2 Plantilla de cargue 11](#_Toc208576984)

[6.3 Reporte – Resultado del cargue masivo 12](#_Toc208576985)

[7. Glosario de Términos 13](#_Toc208576986)

**Historial de versiones**

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Versión** | **Descripción** |
| Cristian Puentes, Franco Fonseca, Sofia Barahona | Versión 1.0 | Creación inicial del documento |

**Aprobaciones**

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Cargo** | **Fecha** |
|  |  |  |

# Introducción

## Propósito del Documento

Este documento describe el diseño técnico y la arquitectura de la solución para la funcionalidad de carga masiva de reglas contables, basándose en los requerimientos funcionales y no funcionales ya definidos.

Su objetivo es servir como una guía funcional y técnica para el equipo del banco y el equipo de desarrollo, detallando los componentes de software, la estructura de la base de datos, las reglas de negocio, los diagramas de flujo y las consideraciones de integración.

## Alcance

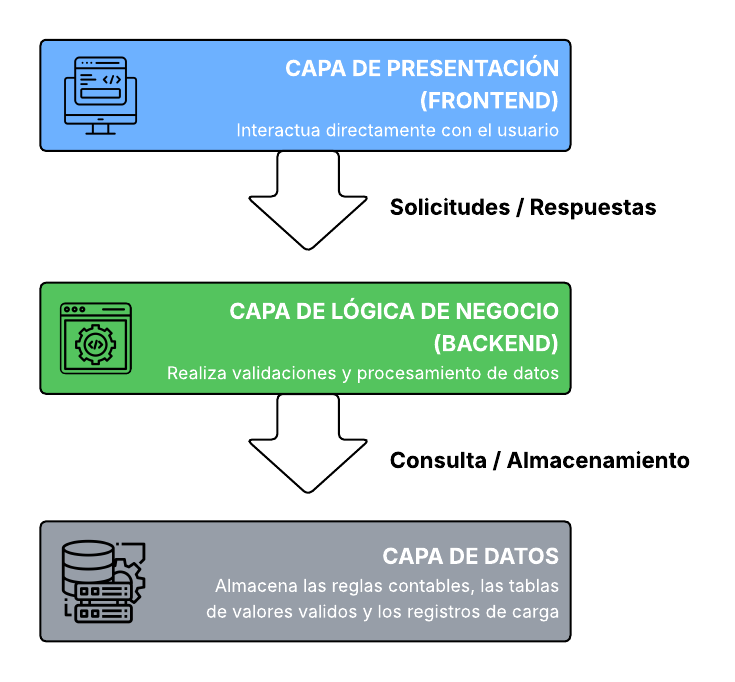
La solución se centrará en crear una interfaz de usuario web para la carga de archivos CSV. El procesamiento será asíncrono y la información se validará mediante integraciones con la base de datos DB2 for i. Esto permitirá la validación, inserción de registros y generación de reportes.

El alcance incluye la funcionalidad de seguimiento del estado de los trabajos de carga y la descarga de reportes, al igual que la consulta de registros. La generación de la plantilla del archivo de entrada y la edición o eliminación de registros ya cargados no están contempladas en esta fase, ya que se consideran fuera del objetivo principal del proyecto. Esta delimitación busca gestionar expectativas y brindar claridad sobre los objetivos de la solución.

# Arquitectura de la Solución

## Visión General de la Arquitectura

La arquitectura de la solución se basa en un diseño de tres capas para asegurar la separación de responsabilidades y facilitar el mantenimiento.



## Componentes de la Solución

**CAPA FRONT-END**

El frontend es la parte de la aplicación que el usuario puede ver y con la que interactúa directamente. Es la "pantalla" o la interfaz que se carga en el navegador del usuario.

**¿Qué hace?**

Es responsable de la presentación visual y la interacción. Aquí es donde el usuario arrastra y suelta el archivo CSV o hace clic en un botón para seleccionarlo.

El frontend realizar validación básica del archivo (Por ejemplo, que el formato sea CSV) antes de enviarlo. Su objetivo principal es recibir las acciones del usuario y mostrarle los resultados de una manera clara y legible.

**CAPA BACK-END**

El backend es la lógica detrás de la interfaz. Funciona en un servidor y no es visible para el usuario. Es el "cerebro" de la aplicación.

**¿Qué hace?**

El backend procesa las solicitudes que vienen del frontend. Cuando el frontend envía el archivo CSV, el backend es quien lo recibe. Es el encargado de leer los datos del archivo, realizar los cálculos o manipulaciones necesarias y decidir qué hacer con esa información, como guardarla en una base de datos y enviarle una respuesta al frontend.

**CAPA DE DATOS**

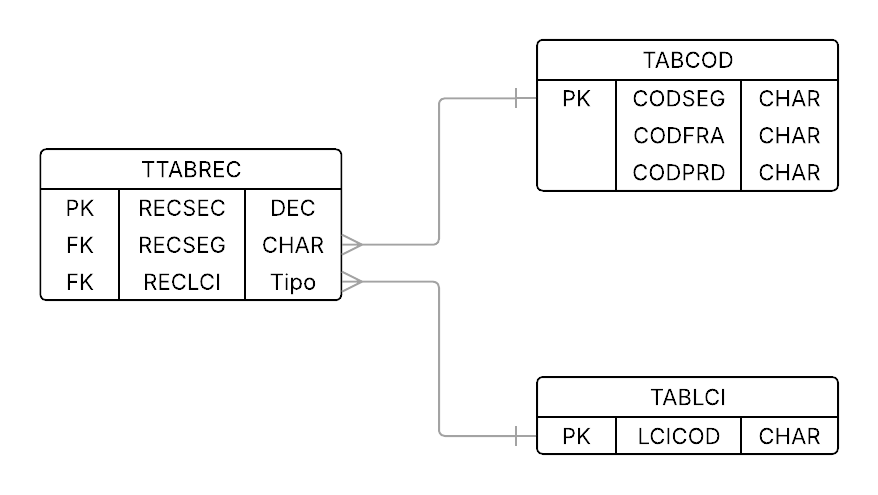
La base de datos es el componente donde se almacena y se organiza toda la información de la aplicación de manera persistente.

**¿Qué hace?**

Una vez que el backend ha procesado el archivo CSV, la base de datos es el lugar seguro donde se guardan esos datos para poder consultarlos más tarde, la base de datos almacena la información para que el usuario pueda ver un resumen histórico, resultado de su cargue o generar informes más adelante.

# Diseño de la Base de Datos

## Diagrama de Entidad-Relación (ERD)



**Ilustración 1 - ERD Integridad de datos**

## Estructura de Tablas

### TABREC – Reglas contables

Tabla que almacena toda la información de reglas contables del sistema, la cual está estructurada de la siguiente forma:

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre | Tipo | Precisión | Escala |
| RECPOP | CHARACTER | 5 | 0 |
| RECCTR | CHARACTER | 6 | 0 |
| RECCNL | CHARACTER | 3 | 0 |
| RECCRQ | CHARACTER | 5 | 0 |
| RECPRD | CHARACTER | 3 | 0 |
| RECTMN | CHARACTER | 3 | 0 |
| RECCPE | CHARACTER | 6 | 0 |
| RECTAQ | CHARACTER | 3 | 0 |
| RECTTR | CHARACTER | 3 | 0 |
| RECACC | CHARACTER | 6 | 0 |

### TTABREC – Temporal de reglas contables

Tabla que almacena temporalmente toda la información de reglas contables extraídas del archivo CSV subida por el usuario, la cual está estructurada de la siguiente forma:

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre | Tipo | Precisión | Escala |
| RECSEC | DECIMAL | 6 | 0 |
| RECCML | CHARACTER | 3 | 0 |
| RECCDP | CHARACTER | 50 | 0 |
| RECCTR | CHARACTER | 5 | 0 |
| RECDSC | CHARACTER | 150 | 0 |
| RECTMN | CHARACTER | 3 | 0 |
| RECSEG | CHARACTER | 3 | 0 |
| RECCDC | CHARACTER | 3 | 0 |
| RECACC | CHARACTER | 3 | 0 |
| RECTAQ | CHARACTER | 3 | 0 |
| RECPOP | CHARACTER | 3 | 0 |
| RECCPE | CHARACTER | 4 | 0 |
| RECTSM | TIMESTAMP | 26 | 6 |
| RECCSC | DECIMAL | 10 | 0 |
| RECCSS | CHARACTER | 5 | 0 |
| RECSTS | CHARACTER | 2 | 0 |

### TABCOD – Tipos de segmentaciones

Tabla que almacena las posibles segmentaciones junto con la codificación correspondiente de franquicia y producto, la cual está estructurada de la siguiente forma:

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre | Tipo | Precisión | Escala |
| CODSEG | CHARACTER | 3 | 0 |
| CODFRA | CHARACTER | 3 | 0 |
| CODPRD | CHARACTER | 3 | 0 |

### TABLCI – Tipos de segmentaciones

Tabla que almacena todos los posibles valores de canales de procesamiento, la cual está estructurada de la siguiente forma:

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre | Tipo | Precisión | Escala |
| LCICOD | CHARACTER | 3 | 0 |
| LCIDSC | CHARACTER | 50 | 0 |

# Reglas de Negocio y Lógica de Proceso

**Validaciones del Archivo de Entrada**

Estas validaciones deben ejecutarse en la capa de presentación (frontend) y ser replicadas en el servidor (backend) como una medida de seguridad y robustez.

* **Validación de Formato:** La funcionalidad de carga debe aceptar exclusivamente archivos con la extensión .csv. Cualquier otro tipo de archivo debe ser rechazado inmediatamente, mostrando un mensaje de error al usuario.
* **Validación de Estructura:** El sistema debe verificar que el archivo de entrada cumpla con la estructura de columnas predefinida en la plantilla. Esto incluye el número correcto de columnas y el orden esperado. Un archivo con una estructura incorrecta debe ser rechazado antes de iniciar el procesamiento.

**Validaciones de los Registros (Lógica de Negocio)**

Estas reglas de negocio son la columna vertebral del procesamiento y deben ser implementadas en el backend para garantizar la consistencia de los datos a nivel de la base de datos.

* **Detección de Duplicados en el archivo insumo:** El sistema debe identificar y descartar los registros que estén duplicados en el archivo CSV que se esté subiendo, teniendo en cuenta los siguientes campos:
  + RCOCNL
  + RCOCTR
  + RCOTMN
  + RCOSEG
  + RCOTDC
  + RCOTAQ
  + RCOPOP
  + RCOCRE
* **Detección de Duplicados en el archivo insumo:** El sistema debe identificar y descartar los registros que ya existen en la tabla principal de reglas contables (TABREC). Esta validación debe realizarse de forma eficiente, idealmente mediante una operación de comparación a nivel de base de datos antes de la inserción final.
* **Validación de la Integridad de los Datos:** Cada registro individual debe ser validado línea por línea. Esto implica:
  + **Campos Requeridos:** Verificar que ninguno de los siguientes campos venga vacío:
    - * RCOCNL
      * RCOCTR
      * RCOTMN
      * RCOSEG
      * RCOTDC
      * RCOTAQ
      * RCOPOP
      * RCOCRE
  + **Valores de Negocio:** Validar que los valores cargados tengan datos validos teniendo en cuenta que:
    - * RCOSEG: El campo debe existir en la tabla TABCOD.
      * RCOLCI: El campo debe existir en la tabla TABLCI.
* **Generación de respaldo:** La solución deberá generar una copia de seguridad de las reglas contables existentes en el sistema cada vez que se realice un cargue. Esta copia será única; por lo tanto, si ya existe una copia previa, deberá ser reemplazada en cada nueva ejecución del cargue.

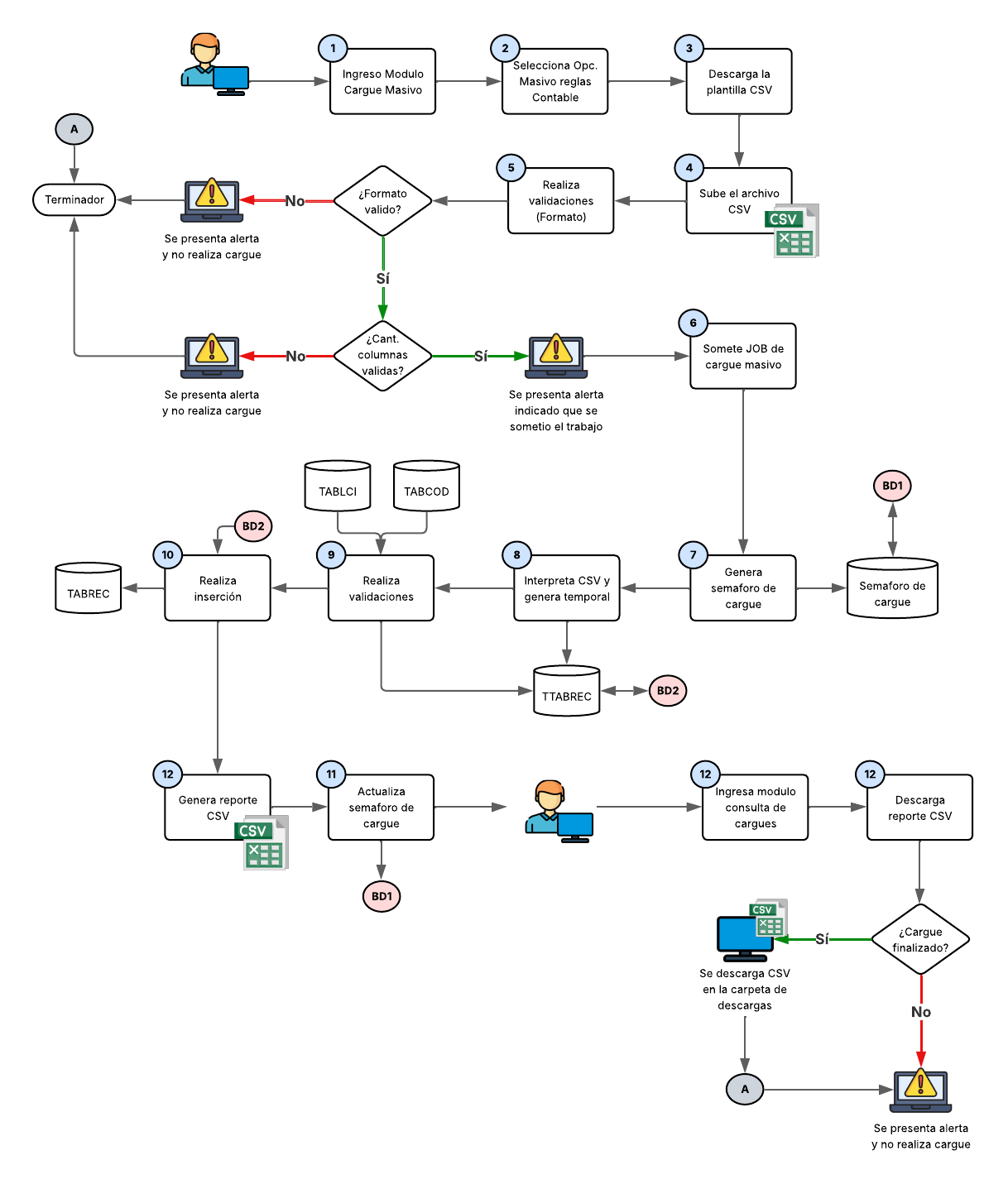
**Reglas de Comportamiento del Sistema**

Estas reglas definen el comportamiento general de la aplicación y la experiencia del usuario.

* **Procesamiento Asíncrono:** La carga masiva se ejecutará como un batch job en segundo plano. Esto asegura que el frontend no se bloquee, permitiendo al usuario continuar navegando mientras el proceso se completa.
* **Generación de Reporte Detallado:** Al finalizar el procesamiento, se debe generar un reporte (CSV) que documente el resultado de cada registro individual. El reporte debe incluir, para cada línea del archivo original, el estado (Éxito, Error, Duplicado, Etc…) y una descripción concisa del motivo en caso de falla.
* **Visualización y Descarga:** La aplicación debe proporcionar una pantalla de seguimiento que permita:
  + Consultar el estado de los trabajos de carga.
  + Descargar el reporte de resultados una vez que el proceso haya finalizado.
* **Inmutabilidad de los Datos:** El sistema no debe ofrecer ninguna funcionalidad para editar o eliminar los registros de reglas contables una vez que se han insertado exitosamente en la base de datos. Esta es una restricción funcional explícita.
* **Control de Concurrencia:** El alcance del proyecto no incluye control de concurrencia; por lo tanto, quien realice el cargue masivo de reglas contables debe asegurarse de ser el único usuario en el proceso, ya que esto podría afectar el correcto funcionamiento del sistema.
* **Descarga de resultados:** El sistema debe permitir descargar el resultado del cargue en formato CSV, siempre y cuando el cargue ya esté finalizado.

# Diagrama de Flujo del Sistema

## Diagrama del Proceso de Carga



**Ilustración 2 - Diagrama de flujo del proceso de cargue masivo de reglas contables**

# Integraciones

## Interfaces de usuario

A continuación, se presenta cada una de las interfaces por las cuales el usuario podrá realizar la el cargue de la navegación correspondiente y consulta el resultado del cargue:

Interfaz de usuario gráfica, Aplicación, Teams

El contenido generado por IA puede ser incorrecto.

**Ilustración 3 - Interfaz del usuario para cargue de la plantilla**

Tabla

El contenido generado por IA puede ser incorrecto.

**Ilustración 4 - Interfaz del usuario consulta de cargues masivos**

Interfaz de usuario gráfica, Aplicación, Word

El contenido generado por IA puede ser incorrecto.

**Ilustración 5 - Consulta de registros cargados**

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

**Ilustración 6 - Consulta detallada de un registro del cargue**

## Plantilla de cargue

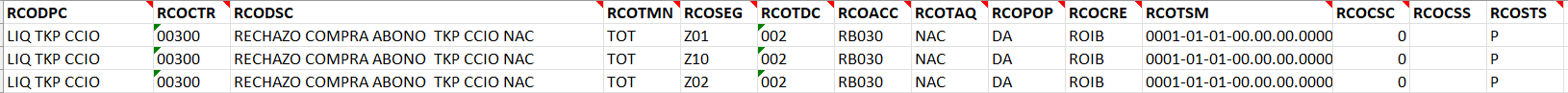
La plantilla de cargue tiene que respetar el siguiente formato:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nombre | Descripción | Tipo | Precisión | Escala |
| RECSEC | Numero único del registro | DECIMAL | 6 | 0 |
| RECCML | Canal de procesamiento | CHARACTER | 3 | 0 |
| RECCDP | Descripción del proceso que pertenece la regla contable | CHARACTER | 50 | 0 |
| RECCTR | Código ISO de la transacción | CHARACTER | 5 | 0 |
| RECDSC | Descripción de la regla contable | CHARACTER | 150 | 0 |
| RECTMN | Tipo de monto | CHARACTER | 3 | 0 |
| RECSEG | Segmentación | CHARACTER | 3 | 0 |
| RECCDC | Tipo de transacción (crédito o débito) | CHARACTER | 3 | 0 |
| RECACC | Acción o regla contable | CHARACTER | 3 | 0 |
| RECTAQ | Tipo de adquirencia | CHARACTER | 3 | 0 |
| RECPOP | Código de proceso | CHARACTER | 3 | 0 |
| RECCPE | Rechazo | CHARACTER | 4 | 0 |
| RECTSM | Timestamp (Fecha-hora-nanosegundo) | TIMESTAMP | 26 | 6 |
| RECCSC | Código de error SQLCODE | DECIMAL | 10 | 0 |
| RECCSS | Código de error SQLTATE | CHARACTER | 5 | 0 |
| RECCSTS | Estatus del registro | CHARACTER | 2 | 0 |

Se debe tener en cuenta que los siguientes campos deben tener siempre los siguientes valores:

* + **RECTSM:** 0001-01-01-00.00.00.000000
  + **RECCSC:** 0
  + **RECCSS:** Blanco
  + **RECCSTS:** P

Se presenta a continuación un ejemplo de la plantilla:



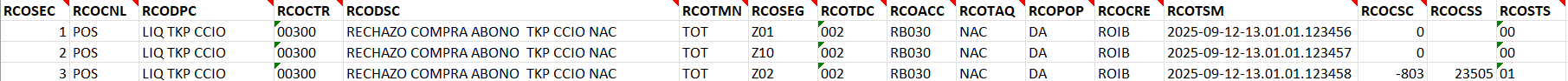
**Ilustración 4 - Ejemplo de cargue de la plantilla**

## Reporte – Resultado del cargue masivo

El reporte del resultado del cargue masivo se realiza bajo la siguiente estructura

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nombre | Descripción | Tipo | Precisión | Escala |
| RECSEC | Numero único del registro | DECIMAL | 6 | 0 |
| RECCML | Canal de procesamiento | CHARACTER | 3 | 0 |
| RECCDP | Descripción del proceso que pertenece la regla contable | CHARACTER | 50 | 0 |
| RECCTR | Código ISO de la transacción | CHARACTER | 5 | 0 |
| RECDSC | Descripción de la regla contable | CHARACTER | 150 | 0 |
| RECTMN | Tipo de monto | CHARACTER | 3 | 0 |
| RECSEG | Segmentación | CHARACTER | 3 | 0 |
| RECCDC | Tipo de transacción (crédito o débito) | CHARACTER | 3 | 0 |
| RECACC | Acción o regla contable | CHARACTER | 3 | 0 |
| RECTAQ | Tipo de adquirencia | CHARACTER | 3 | 0 |
| RECPOP | Código de proceso | CHARACTER | 3 | 0 |
| RECCPE | Rechazo | CHARACTER | 4 | 0 |
| RECTSM | Timestamp (Fecha-hora-nanosegundo) | TIMESTAMP | 26 | 6 |
| RECCSC | Código de error SQLCODE | DECIMAL | 10 | 0 |
| RECCSS | Código de error SQLTATE | CHARACTER | 5 | 0 |
| RECCSTS | Estatus del registro | CHARACTER | 2 | 0 |

Se presenta a continuación un ejemplo del reporte:



**Ilustración 5 - Resultado del cargue masivo**

**Nota:** Los posibles valores de RECSTS será entregados al cliente con el manual de usuario de la solución, teniendo en cuenta las reglas de negocio definidas en este documento.

# Glosario de Términos

* **Backend (Capa de lógica de negocio):** Es el "cerebro" de la aplicación. Se ejecuta en un servidor, no es visible para el usuario y se encarga de procesar las solicitudes del frontend, realizar cálculos, manipular datos y guardarlos en una base de datos.
* **Capa de Datos:** Es el componente donde la información de la aplicación se almacena y organiza de forma persistente, permitiendo su consulta posterior. En este caso, almacena las reglas contables, tablas de valores válidos y registros de carga.
* **CSV:** Formato de archivo de texto plano que utiliza comas (o algún otro delimitador) para separar valores. Es el formato exclusivo que la solución de carga masiva acepta para los archivos de entrada.
* **DB2 for i:** Una base de datos que se utiliza para validar y almacenar la información de las reglas contables, permitiendo la inserción de registros y la generación de reportes.
* **Frontend (Capa de Presentación):** Es la parte de la aplicación con la que el usuario interactúa directamente. Es la interfaz visual que se carga en el navegador y es responsable de la presentación y la validación básica de archivos.
* **Procesamiento Asíncrono:** Un método de procesamiento en el que la carga masiva se ejecuta en segundo plano como un batch job. Esto evita que la interfaz de usuario se bloquee, permitiendo al usuario continuar con otras actividades mientras el proceso finaliza.
* **Reglas Contables:** Conjunto de criterios y lógicas que la aplicación gestiona. La solución se centra en la carga masiva de estas reglas.
* **Reporte Detallado:** Un archivo CSV generado al final del proceso de carga masiva que documenta el resultado de cada registro individual. Incluye el estado (Éxito, Error, Duplicado) y una descripción del motivo en caso de falla.
* **TABCOD:** Tabla que almacena las posibles segmentaciones junto con la codificación de franquicia y producto.
* **TABLCI:** Tabla que almacena todos los posibles valores de canales de procesamiento.
* **TABREC:** Tabla principal del sistema que almacena de manera definitiva toda la información de las reglas contables.
* **TTABREC:** Tabla temporal que almacena los datos de las reglas contables extraídos del archivo CSV antes de su procesamiento y validación final.