

**Especificación de requerimientos – Funcionalidad cargue masivo de reglas contables**

**Bogota**

**2025**

# Historial de Versiones

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Fecha** | **Versión** | **Autor** | **Organización** | **Descripción** |
| 06/09/2025 | 1.0 | Cristian Puentes | Universidad Libre | Versión Inicial |

# Aprobaciones

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre y Apellido** | **Cargo** | **Departamento y Organización** | **Fecha** | **Firma** |
|  |  |  |  |  |

# Propósito

**REQ-0001:** Capacidad de cargar masivamente reglas contables basado en un insumo entregado en formato CSV

# Alcance del Producto / Software

**REQ-0001**

El sistema propuesto abarca la totalidad del flujo de trabajo para la importación de reglas contables. Esto incluye la interacción del usuario para la selección y carga de un archivo de valores separados por comas (CSV) en la interfaz web, la transmisión segura de dicho archivo al servidor, el procesamiento asincrónico en el servidor, la validación de cada registro con respecto a las reglas de negocio y los datos existentes, la inserción de los registros válidos en la base de datos DB2, y la generación de un reporte detallado con los resultados.

Además, se contempla una pantalla de seguimiento que permitirá la visualización en tiempo real del estado del proceso de carga y un resumen de los registros procesados. El sistema no incluye la funcionalidad de edición o eliminación de registros de reglas contables una vez que han sido cargados.

# Entorno operativo

La solución se ejecuta en plataforma IBM Power®, en las instalaciones del Banco, cuyo sistema operativo es IBM i para soportar los ambientes de Desarrollo, calidad y Producción.

La solución se ejecuta desde la web mediante una URL para la capa de presentación.

# Requerimientos funcionales

## 4.1 *BBU-0001 Cargue masivo de reglas contables*

|  |  |
| --- | --- |
| Descripción de Funcionalidad | Se requiere cargar reglas contables desde un archivo CSV. El backend en RPGLE valida cada registro, evitando duplicados, y genera un reporte detallado de los resultados. La interfaz web con JavaScript muestra el progreso y el resumen de la carga. Finalmente, el usuario puede descargar el reporte para su revisión. |
| Prioridad | Media |
| Acciones y Comportamiento esperado | |
| 1. **Carga del archivo:**  * **Acción del usuario:** Accede a la pantalla de carga masiva y selecciona un archivo CSV. * **Comportamiento del sistema (Frontend):** Inicia la subida del archivo al servidor y muestra una barra de progreso visual para el usuario.  1. **Inicio del procesamiento:**  * **Acción del sistema (Backend):** Una vez recibido, el servidor almacena el archivo en el IFS (Integrated File System) de IBM i. El sistema responde inmediatamente con un identificador único para el trabajo, lo que permite que el procesamiento ocurra de manera asíncrona. * **Comportamiento esperado:** El usuario puede cerrar la pantalla o continuar navegando, ya que la carga del archivo ha finalizado en el frontend.  1. **Ejecución de la lógica de negocio:**  * **Acción del sistema (Backend):** Un trabajo por lotes (batch job) se activa en segundo plano para procesar el archivo CSV. Este trabajo lee los datos línea por línea y los carga en una tabla temporal de *staging* en la base de datos DB2. * **Comportamiento esperado:** El sistema valida la integridad de los datos, detecta los registros que ya existen en la tabla principal de reglas contables y los marca como duplicados. Solo inserta los registros nuevos y válidos en la tabla final, todo esto a nivel de base de datos para una mayor eficiencia.  1. **Generación del reporte:**  * **Acción del sistema (Backend):** El proceso por lotes finalizado genera un reporte detallado. Este archivo (CSV o Excel) incluye una línea por cada registro del archivo original, indicando si se cargó con éxito, si fue descartado por duplicado o si contenía algún error de validación, junto con el motivo del fallo. * **Comportamiento esperado:** El reporte es guardado en el IFS del servidor, listo para ser descargado.  1. **Visualización de resultados:**  * **Acción del usuario:** El usuario navega a la pantalla de seguimiento del cargue y selecciona el trabajo recién finalizado. * **Comportamiento esperado:** La pantalla muestra un resumen del proceso y, al hacer clic, presenta una tabla con los resultados detallados de cada registro. El sistema utiliza paginación para garantizar un rendimiento óptimo, incluso con archivos de gran tamaño. Un botón de descarga le permite al usuario obtener el reporte de resultados | |
| Lista de requerimientos funcionales | * F-01: Proceso de Carga de Archivos * F-02: Lógica de Procesamiento y Validación * F-03: Generación de Reporte * F-04: Pantalla de Seguimiento y Visualización |
| Supuestos y Restricciones | **Supuestos**   * **Formato del archivo de entrada:** El archivo de entrada será un CSV con un formato y una estructura de columnas predefinidos y consistentes, que se ajustarán a una plantilla específica. Se asume que el usuario cliente ha generado este archivo siguiendo dicha plantilla. * **Permisos de usuario:** El usuario que inicie la carga masiva tendrá los permisos necesarios para realizar la subida de archivos al servidor y las operaciones de escritura en las tablas de la base de datos DB2 for i.   **Restricciones**   * **Tecnologías de implementación:** El desarrollo del backend debe realizarse exclusivamente en lenguaje RPGLE en la plataforma IBM i, mientras que el frontend se limitará al uso de JavaScript. * **Gestión de la lógica de negocio:** La validación de duplicados y la inserción de datos se realizarán en el backend. El frontend no contendrá lógica de negocio compleja, y su comunicación con el backend será a través de APIs REST. * **Procesamiento asíncrono:** La carga masiva se ejecutará como un proceso por lotes (batch job) en segundo plano para evitar fallas por tiempo de espera del lado del cliente y para no comprometer el rendimiento de otras aplicaciones en línea. * **Origen del insumo:** La solución no incluye la generación de la plantilla o el archivo CSV. El archivo de entrada debe ser entregado por el cliente, ya que este es el insumo para el proceso de carga masiva. |
| Criterios de aceptación | * El sistema permite a un usuario subir un archivo CSV. * El backend valida que el archivo de entrada sea un CSV y tenga el formato y la estructura correctos. * El procesamiento de los registros en el backend se realiza de forma asíncrona, sin bloquear al usuario. * El sistema carga masivamente los registros válidos a la tabla principal de reglas contables. * El sistema identifica y descarta registros duplicados de manera eficiente durante el proceso de carga, basándose en la validación en la base de datos. * Para cada proceso de carga, el sistema genera un reporte detallado en formato CSV o Excel, incluyendo el estado de cada registro (éxito, error, duplicado) y el motivo del fallo. * El reporte de resultados puede ser descargado por el usuario desde la interfaz de seguimiento. * Una pantalla dedicada muestra un listado de todos los trabajos de carga masiva, con información relevante como el estado, la fecha y el usuario. * Al seleccionar un trabajo, el sistema presenta una vista detallada de los registros procesados con paginación optimizada para manejar grandes volúmenes de datos. |

## 4.1.1 *Requerimientos detallados*

**REQ 0001-1:** Pantalla de cargue masivo

|  |  |
| --- | --- |
| Alcance de requerimiento | La solución debe contar con la capacidad de configurar las listas desplegables requeridas durante los pasos de creación de afiliados comerciales y posventa. |
| Descripción de datos | * Plantilla Excel en formato CSV |
| Operaciones soportadas | * Consulta y adición de registros sobre la tabla de acciones contables |
| Flujo de Trabajo Principal | |
| 1. El usuario prepara un archivo en formato **CSV** con todas las reglas contables a cargar. 2. El usuario accede a la pantalla de carga del sistema y selecciona el archivo. 3. El sistema revisa de forma rápida que el archivo tenga la extensión .csv y que no sea demasiado grande. 4. Cuando el archivo se transfiere al IFS se genera una alerta notificando al usuario que se cargó exitosamente. 5. El sistema recibe el archivo y lo guarda en un área temporal. 6. En lugar de procesarlo inmediatamente, el sistema inicia un **proceso por lotes** en segundo plano. 7. El sistema responde al usuario con un mensaje de confirmación y un número de trabajo único para que pueda hacerle seguimiento. 8. El proceso por lotes lee cada registro del archivo. 9. Todos los datos se cargan primero en una **tabla de preparación temporal** en la base de datos. 10. El sistema ejecuta una única operación que compara los datos temporales con los que ya existen en la base de datos. Solo los registros completamente nuevos son insertados en la tabla final. 11. Durante este proceso, se registran los errores (por ejemplo, datos con formato incorrecto, campos vacíos) y los registros duplicados. 12. Una vez finalizado el procesamiento, el sistema genera un **reporte detallado** con los resultados de la carga y lo guarda. 13. El estado del trabajo se actualiza a COMPLETADO o CON\_ERRORES. | |
| Reportes y Salidas | * Reporte en csv en una ruta IFS |
| Integraciones | * Para este requerimiento no aplica integraciones. |

**REQ 0001-2:** Pantalla de visualización del resultado del cargue masivo

|  |  |
| --- | --- |
| Alcance de requerimiento | Esta funcionalidad permite a los usuarios visualizar y descargar los resultados del cargue de reglas contables |
| Descripción de datos | * Reporte ubicado en una ruta IFS |
| Operaciones soportadas | * Consulta del reporte en el IFS |
| Flujo de Trabajo Principal | |
| 1. El usuario navega a la **pantalla de seguimiento** del sistema. 2. En esta pantalla, ve una lista con todos los trabajos de carga masiva que se han ejecutado. 3. Por cada trabajo, se muestran detalles como el nombre del archivo, el usuario que lo subió y el estado final (COMPLETADO o CON\_ERRORES). 4. Desde aquí, el usuario puede seleccionar un trabajo finalizado y descargar el   **reporte de resultados** en un archivo CSV o Excel para un análisis fuera del sistema.   1. Al seleccionar un trabajo específico, la pantalla de seguimiento muestra una tabla con los resultados detallados de cada registro. 2. La interfaz utiliza una técnica llamada **"procesamiento del lado del servidor"** para cargar solo la información necesaria en la pantalla, lo que permite manejar grandes volúmenes de datos sin que la aplicación se vuelva lenta o se bloquee. 3. En esta tabla, el usuario puede ver claramente los registros que no se cargaron, junto con un **mensaje de error específico** que explica la razón de la falla. 4. Esto permite al usuario identificar rápidamente los problemas y tomar las medidas correctivas necesarias, como modificar el archivo de origen y volver a intentar la carga de los registros fallidos. | |
| Reportes y Salidas | * Archivo CSV ubicado en la carpeta de descargas |
| Integraciones | * No aplican integraciones para este requerimiento |

# 5. Casos de Uso

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nº** | **Caso de Uso** | **Actor Principal** | **Descripción** | **Resultado Esperado** |
| 1 | Cargar archivo CSV | Usuario final | El usuario accede a la pantalla de carga masiva y selecciona un archivo CSV. | El archivo se sube al servidor y se almacena en el IFS. Se genera un ID de trabajo. |
| 2 | Procesamiento asíncrono del archivo | Sistema | Se activa un batch job que procesa el archivo línea por línea. | Los datos válidos se insertan en la tabla final. Los duplicados se descartan. |
| 3 | Generación del reporte | Sistema | Se genera un reporte con el estado de cada registro (éxito, error, duplicado). | El reporte se guarda en el IFS y queda disponible para descarga. |
| 4 | Visualización del seguimiento | Usuario | El usuario accede a la pantalla de seguimiento y selecciona el trabajo. | Se muestra un resumen del proceso y una tabla paginada con los resultados. |
| 5 | Validación del formato del archivo | Sistema) | Al recibir el archivo, el backend verifica que sea un CSV y que cumpla con la estructura esperada. | Si el archivo no cumple, se rechaza y se notifica al usuario con un mensaje de error. |
| 6 | Notificación de estado del trabajo | Sistema | El sistema actualiza el estado del trabajo (pendiente, en proceso, finalizado, con errores). | El usuario puede consultar el estado en tiempo real desde la pantalla de seguimiento. |
| 7 | Reintento de carga | Usuario | El usuario puede volver a cargar un archivo corregido si el anterior fue rechazado. | Se crea un nuevo trabajo con el archivo corregido y se inicia el procesamiento. |
| 8 | Consulta de historial de cargas | Usuario | El usuario accede a un listado de trabajos anteriores con filtros por fecha, estado. | Se muestra el historial completo con opción de ver detalles y descargar reportes. |
| 9 | Eliminación de trabajos fallidos | Usuario | El usuario puede eliminar trabajos que hayan fallado o que ya no sean relevantes. | El sistema elimina el trabajo y sus archivos asociados del IFS. |
| 10 | Control de concurrencia | Sistema | El sistema evita que un mismo usuario cargue múltiples archivos simultáneamente. | Se muestra un mensaje de advertencia si ya hay un trabajo en curso. |
| 11 | Gestión de errores críticos | Sistema | Si ocurre un error inesperado durante el procesamiento, se registra y se notifica al usuario. | El trabajo se marca como fallido y se genera un reporte de error técnico. |

# 6. Glosario

* **RPGLE:** Lenguaje de programación de alto nivel utilizado en la plataforma IBM i.
* **IBM i:** Sistema operativo y plataforma de hardware de IBM, conocida anteriormente como AS/400.
* **DB2 for i:** Sistema de gestión de bases de datos relacionales integrado en la plataforma IBM i.
* **IFS (Integrated File System):** Sistema de archivos de IBM i que proporciona acceso a datos en formato de *stream files*, similar a los sistemas de archivos de PC y UNIX.
* **REST API:** Estilo de arquitectura para el diseño de interfaces de programación de aplicaciones (APIs) que utiliza solicitudes HTTP para acceder y usar datos.
* **JSON:** (JavaScript Object Notation) Formato ligero de intercambio de datos, ideal para la comunicación entre el frontend y el backend.3
* **CSV:** (Comma-Separated Values) Formato de archivo de texto plano para almacenar datos tabulares.
* **Frontend:** La parte de la aplicación con la que el usuario interactúa directamente en el navegador web, desarrollada con JavaScript.6
* **Backend:** La lógica de negocio que se ejecuta en el servidor (en este caso, en IBM i) y gestiona el acceso a la base de datos y otras operaciones del sistema.
* **Asíncrono:** Tipo de operación que permite que un proceso se inicie sin bloquear al usuario o al sistema, de modo que el sistema pueda realizar otras tareas mientras se completa.