****

MÓDULO DE CARGUE MASIVO DE REGLAS CONTABLES

Docente: Mg. Marcela Cifuentes Velásquez

Ingeniería de Sistemas

PRESENTADO POR

SOFIA YURANY BARAHONA SANCHEZ

CRISTIAN ALEXANDER PUENTES ROZO

FRANCO STIVEN FONSECA MONTAÑEZ

Bogotá - Colombia

6 septiembre 2025

Universidad Libre – Sede el Bosque

Gestión de Proyectos

**TABLA DE CONTENIDO**

[**TABLA DE ILUSTRACIONES** 3](#_Toc211964812)

[**INDICE DE TABLAS** 4](#_Toc211964813)

[**TABLA DE ANEXOS** 5](#_Toc211964814)

[**RESUMEN EJECUTIVO** 6](#_Toc211964815)

[**INTRODUCCION** 6](#_Toc211964816)

[**OBJETIVO GENERAL** 7](#_Toc211964817)

[**OBJETIVOS ESPECÍFICOS** 7](#_Toc211964818)

[**ALCANCE** 7](#_Toc211964819)

[**LÍMITES Y DELIMITACIONES** 7](#_Toc211964820)

[**METODOLOGIA** 8](#_Toc211964821)

[**ENFOQUES Y PROCESOS** 8](#_Toc211964822)

[**APLICACION DE LAS ETAPAS** 8](#_Toc211964823)

[**1.** **ETAPA DE INICIO** 8](#_Toc211964824)

[**2.** **ETAPA DE PLANFICACIÓN** 8](#_Toc211964825)

[**2.1. Definición de Objetivos** 8](#_Toc211964826)

[**2.2. Identificación del público objetivo** 8](#_Toc211964827)

[**2.3. Análisis de la competencia** 9](#_Toc211964828)

[**3.** **REQUISITOS FUNCIONALES Y TÉCNICOS.** 10](#_Toc211964829)

[**3.1. Requisitos Funcionales** 10](#_Toc211964830)

[**3.2. Requisitos Técnicos** 11](#_Toc211964831)

[**4.** **CREACIÓN DE UN PLAN DE PROYECTO** 12](#_Toc211964832)

[**4.1. Estrategia de Ejecución** 12](#_Toc211964833)

[**4.2. Fases y Actividades** 12](#_Toc211964834)

[**4.3. Herramientas Y Recursos** 13](#_Toc211964835)

[**4.3.1. Recursos Humanos** 13](#_Toc211964836)

[**4.3.2. Recursos Tecnológicos** 13](#_Toc211964837)

[**4.3.3. Herramientas de Pruebas y Calidad** 14](#_Toc211964838)

[**4.3.4. Recursos de Gestión del Cambio y Capacitación** 14](#_Toc211964839)

[**4.3.5. Recursos Financieros** 14](#_Toc211964840)

[**5.** **PROPUESTA AL CLIENTE** 14](#_Toc211964841)

[**6.** **RESULTADOS DE LA PLANEACIÓN** 15](#_Toc211964842)

[**6.1. Documentación de Requerimientos** 15](#_Toc211964843)

[**6.2. Definición del Alcance** 15](#_Toc211964844)

[**6.3. Plan de Proyecto** 15](#_Toc211964845)

[**6.4. Gestión de Riesgos** 15](#_Toc211964846)

[**6.5. Estimación de Recursos y Presupuesto** 16](#_Toc211964847)

[**7.** **ANÁLISIS DE REQUERIMIENTOS** 16](#_Toc211964848)

[**7.1. Propósito** 16](#_Toc211964849)

[**7.2. Alcance del Producto / Software** 16](#_Toc211964850)

[**7.3. Entorno operativo** 16](#_Toc211964851)

[**7.4. Requerimientos funcionales** 17](#_Toc211964852)

[**7.4.1. *BBU-0001 Cargue masivo de reglas contables*** 17](#_Toc211964853)

[**7.5. *Requerimientos detallados*** 19](#_Toc211964854)

[**7.6. Casos de Uso** 21](#_Toc211964855)

[**8.** **DISEÑO** 23](#_Toc211964856)

[**8.1.** **Alcance** 23](#_Toc211964857)

[**8.2.1.** **Visión General de la Arquitectura** 23](#_Toc211964858)

[**8.2.2.** **Componentes de la Solución** 24](#_Toc211964859)

[**8.3.** **Diseño de la Base de Datos** 25](#_Toc211964860)

[**8.3.1.** **Diagrama de Entidad-Relación (ERD)** 25](#_Toc211964861)

[**8.3.2.** **Estructura de Tablas** 25](#_Toc211964862)

[**8.3.2.1.** **TABREC – Reglas contables** 25](#_Toc211964863)

[**8.3.2.2.** **TTABREC – Temporal de reglas contables** 26](#_Toc211964864)

[**8.3.2.3.** **TABCOD – Tipos de segmentaciones** 26](#_Toc211964865)

[**8.3.2.4.** **TABLCI – Tipos de segmentaciones** 27](#_Toc211964866)

[**9.** **REGLAS DE NEGOCIO Y LÓGICA DE PROCESO** 27](#_Toc211964867)

[**9.1. Validaciones del Archivo de Entrada** 27](#_Toc211964868)

[**9.2. Validaciones de los Registros (Lógica de Negocio)** 27](#_Toc211964869)

[**9.3.** **Reglas de Comportamiento del Sistema** 28](#_Toc211964870)

[**10.** **DIAGRAMA DE FLUJO DEL SISTEMA** 30](#_Toc211964871)

[**9.1.** **Diagrama de actividades** 30](#_Toc211964872)

[**9.2.** **Diagrama de clases** 31](#_Toc211964873)

[**9.3.** **Diagrama de componentes** 32](#_Toc211964874)

[**9.4.** **Diagrama de despliegue** 33](#_Toc211964875)

[**9.5.** **Diagrama de Secuencia** 34](#_Toc211964876)

[**9.6.** **Diagrama de estado** 35](#_Toc211964877)

[**10.** **INTEGRACIONES** 36](#_Toc211964878)

[**10.1.** **Interfaces de usuario** 36](#_Toc211964879)

[**10.2.** **Plantilla de cargue** 38](#_Toc211964880)

[**10.3.** **Reporte – Resultado del cargue masivo** 39](#_Toc211964881)

[**REFERENCIAS** 41](#_Toc211964882)

# **TABLA DE ILUSTRACIONES**

[Ilustración 1. Arquitectura de la solución 24](#_Toc211964893)

[Ilustración 2. ERD Integridad de datos 26](#_Toc211964894)

[Ilustración 3. Diagrama de flujo del proceso de cargue masivo de reglas contables 31](#_Toc211964895)

[Ilustración 4. Diagrama de clases 32](#_Toc211964896)

[Ilustración 5. Diagrama de componentes 33](#_Toc211964897)

[Ilustración 6. Diagrama de despliegue 34](#_Toc211964898)

[Ilustración 7. Diagrama de secuencia 35](#_Toc211964899)

[Ilustración 8. Diagrama de estado 36](#_Toc211964900)

[Ilustración 9. Interfaz del usuario para cargue de la plantilla 37](#_Toc211964901)

[Ilustración 10. Interfaz del usuario consulta de cargues masivos 38](#_Toc211964902)

[Ilustración 11. Consulta de registros cargados 38](#_Toc211964903)

[Ilustración 12. Consulta detallada de un registro del cargue 39](#_Toc211964904)

[Ilustración 13. Ejemplo de cargue de la plantilla 40](#_Toc211964905)

[Ilustración 14. Resultado del cargue masivo 41](#_Toc211964906)

# **INDICE DE TABLAS**

[Tabla 1. Cargue Masivo de Reglas Contables 17](#_Toc211964883)

[Tabla 2. Requerimiento de pantalla de cargue masivo 19](#_Toc211964884)

[Tabla 3. Requerimiento de Pantalla de Visualización de resultado del cargue masivo 20](#_Toc211964885)

[Tabla 4. Casos de uso 21](#_Toc211964886)

[Tabla 5. Reglas contables TABREC 25](#_Toc211964887)

[Tabla 6. Temporal de reglas contables - TTABREC 26](#_Toc211964888)

[Tabla 7. Tipos de segmentaciones - TABCOD 26](#_Toc211964889)

[Tabla 8. Tipos de segmentación - TABLCI 27](#_Toc211964890)

[Tabla 9. Tabla de la plantilla de cargue 38](#_Toc211964891)

[Tabla 10. Tabla de reporte de cargue masivo 39](#_Toc211964892)

# **TABLA DE ANEXOS**

Documentos adicionales como gráficos, tablas, encuestas, imágenes.

[Anexo 1. Acta Aceptación. 8](#_Toc187068569)

[Anexo 2. Archivo del Cronograma y presupuesto. 10](#_Toc187068570)

Anexo 3. Solicitud de Cambios.

# **RESUMEN EJECUTIVO**

El presente proyecto tiene como objetivo desarrollar un módulo de software para el cargue de reglas contables, que permita que el Banco Davivienda optimice la generación de asientos contables para el seguimiento de los movimientos financieros registrados en sus cuentas. El desarrollo del proyecto se sustenta debido a que actualmente el proceso de cargue de dichas reglas contables se realiza de manera manual, lo que genera el uso de personal en una tarea repetitiva, dado que diariamente el flujo de registros se cuenta por millones. Esta situación incrementa el riesgo ya existente de comisión de errores, retrasos y costos operacionales.

Para el correcto desarrollo del proyecto se estableció una rigurosa gestión, completando cada etapa del ciclo de vida del proyecto, desde la concepción de la idea hasta la entrega de documentación y capacitación en el uso del módulo.

Se estima que la duración del proyecto sea de 77 días, y el presupuesto aproximado sea de $172.112.000 millones, que cubrirá recursos humanos, tecnológicos y de gestión del cambio.

# **INTRODUCCION**

El sector bancario afronta un sin fin de desafíos en la actualidad, uno de ellos es modernizar y automatizar procesos internos que den respuesta a las crecientes exigencias regulatorias, operativas y tecnológicas del sector. En este contexto, los procesos manuales en la industria bancaria son insostenibles y requieren de la automatización para mejorar la eficiencia y lograr mantenerse competitivas en un entorno dinámico, ya que los bancos deben controlar los costos para poder competir con el mercado cada vez más creciente de las Startups Fintech. En este orden de ideas, la automatización de procesos internos brinda al sector financiero una herramienta que permite sustituir tareas lentas y propensas a errores por flujos de trabajo optimizados y tecnológicos, mejorando la escalabilidad y garantizando el cumplimiento normativo. [1]

El Banco Davivienda ha identificado en su proceso contable una tarea que implica la destinación de recurso humano para su ejecución, tarea que se convierte en critica debido a en ella se concentran millones de registros diarios, convirtiéndola en un cuello de botella en el flujo de la información. Esta situación genera la necesidad de implementar un módulo de software que permita el cargue masivo de reglas contables, con el fin de facilitar y optimizar la generación de asientos contables, minimizando el tiempo invertido en esta tarea y optimizando la trazabilidad de los registros financieros.

El presente documento describe la planeación del proyecto, incluyendo la definición de objetivos, alcance, actividades a realizar, estimación de tiempos y presupuesto requerido. Asimismo, se presenta la metodología de ejecución y los entregables del proyecto, información fundamental para asegurar el éxito en la implementación del módulo.

Con el desarrollo del proyecto, el Banco Davivienda busca dar un paso estratégico hacia la transformación digital de sus procesos contables, fortaleciendo la capacidad operativa, el cumplimiento normativo y la competitividad en el sector financiero.

**OBJETIVOS**

## **OBJETIVO GENERAL**

Desarrollar un módulo que permita el cargue de reglas contables necesarias para la generación de asientos contables usados por el Banco Davivienda para llevar el seguimiento monetario en sus cuentas**.**

## **OBJETIVOS ESPECÍFICOS**

1. Analizar los requerimientos funcionales y no funcionales del módulo de cargue de reglas contables, en colaboración con las áreas operativas y tecnológicas del banco.

2. Desarrollar un módulo robusto que permita el ingreso, validación y almacenamiento de reglas contables, asegurando su integración con el sistema de generación de asientos contables.

3. Implementar pruebas funcionales e integrales del módulo, así como su despliegue en el entorno productivo, asegurando su mantenimiento evolutivo y correctivo posterior.

# **ALCANCE**

## **LÍMITES Y DELIMITACIONES**

Para le desarrollo del proyecto se plantearon las siguientes disposiciones que permiten comprender el alcance que va a tener el módulo y las restricciones que se preveen.

1. El módulo solo permite que el cargue del documento de las reglas contables se realice en formato CSV
2. Solo se puede cargar el documento de las reglas contables desde este modulo
3. El archivo de respuesta que se genera tras el cargue masivo de las reglas contables solo se genera en formato CSV
4. Se permite la búsqueda de registros como funcionalidad adicional del modulo

Las siguientes disposiciones marcan una hoja de ruta de las restricciones que se plantean para el modulo

1. El módulo de cargue no acepta registros en formato XLS, TXT
2. La respuesta del cargue no se ve en pantalla, si no en el documento generado en formato CSV

# **METODOLOGIA**

## **ENFOQUES Y PROCESOS**

Para el desarrollo de este proyecto se eligió la metodología denominada Cascada o también conocida como Waterfall, esto debido a que se caracteriza por su enfoque secuencial y estructurado, dos características que decantaron la elección. Esta metodología contempla que cada fase del ciclo de vida del desarrollo del software debe completarse antes de continuar con la siguiente, si bien existen algunas excepciones para este caso, como lo es la revisión del documento de requerimientos, proceso que implica una retroalimentación, la mayoría de las actividades están claramente definidas y debido a que este proyecto se desarrolla en el sector financiero se requiere un alto nivel de detalle, control y trazabilidad.

# **APLICACION DE LAS ETAPAS**

1. **ETAPA DE INICIO**

Esta etapa es el punto de partida del proyecto debido a que en ella se formaliza la idea del proyecto, se establece la conformación del equipo de trabajo y la definición de responsabilidades. En esta etapa también se oficializa el primer documento entregable de todo el proyecto, este es el Acta de Constitución, documento oficial que establece los objetivos generales, el alcance preliminar y la designación del líder de proyecto.

1. **ETAPA DE PLANFICACIÓN**

### **2.1. Definición de Objetivos**

El objetivo principal del proyecto es desarrollar un módulo de software que permita el cargue masivo de reglas contables, con el fin de generar asientos contables que cumplan con la normativa del Banco Davivienda y que permita reducir los riesgos asociados a errores manuales que actualmente están presentes en el desarrollo de esta tarea. Para alcanzar este propósito, se busca diseñar e implementar una solución modular que facilite este proceso y que permita la integración con los demás sistemas de la entidad, también se contempla la entrega de documentación técnica y manuales de usuario, junto con un plan de capacitación que permita la correcta adopción y aprovechamiento de esta herramienta por parte de los equipos operativos del banco.

### **2.2. Identificación del público objetivo**

El público objetivo para este proyecto está conformado principalmente por el equipo operativo del Banco Davivienda, que es el responsable de la gestión y control de la información contable; estos usuarios harán uso del módulo de cargue de las reglas contables y generación de asientos contables. También se pueden establecer personal que indirectamente se va a ver beneficiado con la implementación de dicho modulo, como lo es el área de auditoría, debido a la mayor trazabilidad y transparencia en los registros contables.

### **2.3. Análisis de la competencia**

Actualmente las entidades financieras requieren optimizar la mayoría de los procesos con el fin de ofrecer un mejor servicio y mantenerse competitivos en un mercado cada vez más dinámico. Bajo esta premisa, el sector bancario ha decido apoyarse en compañías que brinden soluciones de software personalizadas que respondan a las necesidades específicas de cada área de la empresa, esto brinda a la compañía la certeza de que cada solución se ajusta a la perfección con los requerimientos establecidos, esto se conoce como desarrollo de microservicios, es decir que es una metodologías que permiten una construcción modular, facilitando a los equipos el desarrollo, mantenimiento y operación de sistemas complejos de manera más eficiente. [2]

Dentro de este mercado tan competitivo se pueden identificar tres grandes grupos de empresas. El primero está conformado por las grandes consultoras internacionales que tienen presencia en nuestro país. Algunos de los ejemplos más relevantes de estas empresas son: Accenture, TCS, Globant e Indra. Lo que caracteriza a estas compañías es su vasta experiencia en proyectos bancarios a nivel latinoamericano y su capacidad de conformar grandes y talentosos equipos de trabajo. No obstante, esto implica un elevado costo en su contratación y un burocrático manejo interno de sus procesos, situación que limitan la fluida y flexible contratación.

Por otra parte, están las empresas de software nacionales o de presencia regional. Algunas de las compañías más relevantes en este sector son: Ceiba Software, Pragma, Turing, o Bamboo, entre otras. Sus principales ventajas radican en que ofrecen un balance entre calidad y costo, adicional a esto cuentan con el conocimiento del marco normativo que rige el sector financiero en Colombia. La principal desventaja de estas compañías es que debido a que no cuentan con estructuras organizacionales tan robustas pueden surgir ciertas limitaciones a la hora de asumir proyectos de escala masiva.

Por último, se encuentra el mercado emergente y actualmente en auge y expansión de las FinTech, estas compañías suelen destacarse por su innovación, rapidez en la entrega y costos competitivos. No obstante, pueden presentar menor capacidad para ofrecer el soporte integral y más si es en ambientes que son altamente regulados como el sector financiero.

Actualmente el sector de desarrollo de software ha visto un cambio en cuanto a la percepción que tiene el Venture Capital, según el informe denominado “Colombia TechReport 2024”, se explica que si bien históricamente este sector ha sido visto con muy buenos ojos por el capital de riesgo, en los últimos años debido al aumento de soluciones low-code/no-code y la estandarización de productos SaaS han reducido el diferenciador competitivo, lo cual genera que se presiones las valuaciones a la baja en el futuro. [3]

## **REQUISITOS FUNCIONALES Y TÉCNICOS.**

### **3.1. Requisitos Funcionales**

1. **Cargue de reglas contables**
   1. El sistema debe permitir cargar un archivo en formato **CSV** que contenga las reglas contables.
   2. El cargue debe realizarse únicamente desde el módulo diseñado.
   3. No se aceptarán archivos en otros formatos (XLS, TXT, etc.).
2. **Validación de reglas**
   1. El módulo debe validar la estructura del archivo cargado (columnas, tipo de datos, obligatoriedad).
   2. Debe identificar y reportar errores de consistencia (ej. duplicados, campos vacíos, formatos inválidos).
   3. En caso de errores, debe generar un archivo de retroalimentación en formato **CSV**.
3. **Generación de respuesta**
   1. El sistema debe producir un archivo en formato **CSV** con los resultados del cargue (registros válidos, rechazados, causas de error).
   2. La respuesta se visualizará en pantalla y en el documento descargable.
4. **Integración con el sistema contable**
   1. El módulo debe integrarse con el sistema de generación de asientos contables del banco.
   2. Debe enviar las reglas aprobadas para su procesamiento contable.
5. **Funcionalidad de búsqueda**
   1. Permitir a los usuarios buscar reglas cargadas en el histórico mediante filtros (fecha de cargue, tipo de asiento, código de regla, etc.).
6. **Gestión de usuarios**
   1. El acceso al módulo debe estar restringido a perfiles autorizados.
   2. Debe existir trazabilidad de quién carga, aprueba o rechaza los archivos.

### **3.2. Requisitos Técnicos**

1. **Arquitectura y Plataforma**
   1. Desarrollo bajo un enfoque modular que permita integración futura con otros sistemas bancarios.
   2. Compatible con la infraestructura tecnológica de Davivienda.
   3. Soporte para grandes volúmenes de datos (mínimo 1 millón de registros por archivo).
2. **Base de Datos**
   1. Almacenamiento en base de datos relacional (ej. Oracle, SQL Server o PostgreSQL, según lineamientos del banco).
   2. Tablas optimizadas para búsquedas y consultas rápidas.
3. **Seguridad**
   1. Cumplimiento de normativas bancarias (ej. Sarbanes-Oxley, Basilea, normas locales de la Superfinanciera).
   2. Cifrado en tránsito y en reposo de la información (TLS/SSL).
   3. Control de acceso basado en roles y autenticación con el sistema corporativo (LDAP/AD).
4. **Rendimiento**
   1. Procesamiento en paralelo para manejar cargas masivas de datos.
   2. Tiempo máximo de validación y respuesta por cargue: **< 10 minutos para 1 millón de registros**.
5. **Interfaz de usuario**
   1. Interfaz web intuitiva, accesible desde la intranet del banco.
   2. Compatibilidad con navegadores estándar (Chrome, Edge, Firefox).
6. **Disponibilidad y soporte**
   1. El módulo debe estar disponible en un entorno de **alta disponibilidad (24/7)**.
   2. Recuperación ante desastres (backups automáticos, plan de contingencia).
7. **Pruebas y calidad**
   1. Pruebas unitarias, de integración y de rendimiento.
   2. Asegurar que la solución cumpla con los niveles de servicio (SLA) definidos.
8. **CREACIÓN DE UN PLAN DE PROYECTO**

## **4.1. Estrategia de Ejecución**

El proyecto seguirá la metodología **Cascada (Waterfall)**, asegurando que cada fase se cierre antes de pasar a la siguiente. Esto garantiza trazabilidad y control en un entorno financiero regulado.

## **4.2. Fases y Actividades**

**Fase 1. Inicio (Semana 1)**

* **Actividades:**
  + Elaboración y aprobación del Acta de inicio.
  + Conformación del equipo de trabajo.
  + Definición de roles y responsabilidades.

**Fase 2. Planificación (Semanas 2-4)**

* **Actividades:**
  + Levantamiento de información con usuarios clave.
  + Análisis y documentación de requerimientos funcionales y técnicos.
  + Definición del plan de pruebas.
  + Validación de requerimientos con las áreas operativas y tecnológicas.

**Fase 3. Diseño y Desarrollo (Semanas 5-10)**

* **Actividades:**
  + Diseño de la arquitectura del módulo.
  + Creación de base de datos para almacenar reglas contables.
  + Desarrollo del cargue de archivos CSV.
  + Desarrollo de validaciones de datos.
  + Integración con el sistema de asientos contables.
  + Implementación de logs y generación de archivos de respuesta CSV.

**Fase 4. Pruebas (Semanas 11-12)**

* **Actividades:**
  + Ejecución de pruebas unitarias.
  + Pruebas integrales con datos reales.
  + Pruebas de rendimiento (mínimo 1 millón de registros).
  + Corrección de errores detectados.

**Fase 5. Implementación (Semana 13)**

* **Actividades:**
  + Instalación en entorno productivo.
  + Configuración de accesos y seguridad.
  + Pruebas finales en productivo.

**Fase 6. Capacitación y Cierre (Semana 14)**

* **Actividades:**
  + Elaboración de manual de usuario.
  + Capacitación al equipo operativo.
  + Entrega formal de documentación y cierre del proyecto.

**3. Mecanismos de Control**

* Reuniones diarias de seguimiento con el equipo de trabajo.
* Reportes de avance quincenales al patrocinador.

## **4.3. Herramientas Y Recursos**

### **4.3.1. Recursos Humanos**

* **Gerente de Proyecto** → Coordinación, seguimiento y gestión del plan.
* **Desarrolladores (2)** → Construcción del módulo, integración y pruebas.
* **Equipo de Tecnología del Banco** → Soporte en infraestructura y despliegue.
* **Usuarios Clave (Davivienda)** → Validación funcional, pruebas de aceptación.
* **Equipo de QA/Testers** → Ejecución de pruebas unitarias, integrales y de rendimiento.
* **Especialista en Seguridad** → Validación de cumplimiento normativo y control de accesos.

### **4.3.2. Recursos Tecnológicos**

* **Infraestructura de Servidores**
  + Power BI.
* **Bases de Datos**
  + Base de Datos DB2
* **Lenguajes y Frameworks de Desarrollo**
  + Backend: IBM i
  + Frontend: Javascript, Node Js, HTML Y CSS.

**4.3.3. Herramientas de Pruebas y Calidad**

* Para el desarrollo de esta actividad se establece las pruebas unitarias de cada funcionalidad y rendimiento

### **4.3.4. Recursos de Gestión del Cambio y Capacitación**

* Manual de Usuario (digital e impreso).
* Guías rápidas en formato PDF.
* Plataforma de e-learning (ej. Moodle, Blackboard).
* Sesiones presenciales o virtuales de capacitación.

### **4.3.5. Recursos Financieros**

* Presupuesto aprobado: **$172.112.000 COP**
  + Recurso humano (Inicio de Proyecto, Planeación, Ejecución, Monitoreo): $167.888.000
  + Infraestructura tecnológica (Equipos y Licencias): $4.224.000

1. **PROPUESTA AL CLIENTE**

El Banco Davivienda enfrenta actualmente un proceso manual para el cargue de reglas contables, lo que representa una tarea repetitiva, con alta probabilidad de error y un consumo elevado de tiempo y recursos, en especial considerando que diariamente se manejan más de **1 millón de registros**.

Para superar esta problemática, proponemos el desarrollo de un **módulo de software especializado** que permita el **cargue masivo de reglas contables en formato CSV**, asegurando la **integración con el sistema de generación de asientos contables** y la optimización del proceso de gestión financiera.

El módulo validará los datos cargados, generará archivos de retroalimentación con los resultados y permitirá la trazabilidad de cada cargue realizado, cumpliendo con los estándares de seguridad y regulaciones aplicables al sector financiero.

**Beneficios para Davivienda**

* Reducción de los tiempos de procesamiento en más del **90%**.
* Disminución de errores manuales y de costos asociados.
* Mayor trazabilidad y control de la información contable.
* Fortalecimiento de la **transformación digital** y cumplimiento normativo.

**Entregables Principales**

* Documento de requerimientos.
* Módulo de cargue masivo en ambiente productivo.
* Documento de pruebas y resultados de validación.
* Manual de usuario y capacitación a personal operativo.

1. **RESULTADOS DE LA PLANEACIÓN**

## **6.1. Documentación de Requerimientos**

* Documento formal con los **requerimientos funcionales y técnicos** aprobados.
* Definición de **requerimientos no funcionales** (rendimiento, seguridad, disponibilidad, etc.).
* Validación y firma de conformidad por parte del cliente.

**6.2. Definición del Alcance**

* Establecimiento de lo que está **incluido** y lo que está **fuera de alcance**.
* Lista de funcionalidades priorizadas (cargue CSV, validación, integración, reportes).
* Identificación de restricciones.

## **6.3. Plan de Proyecto**

* Cronograma detallado con fases, actividades y tiempos.
* Asignación de responsables para cada actividad.
* Identificación de hitos clave (documento de requerimientos, pruebas, capacitación).

**6.4. Gestión de Riesgos**

* Identificación de riesgos potenciales
* Evaluación de nivel de impacto y probabilidad.
* Planes de mitigación y contingencia documentados.

### **6.5. Estimación de Recursos y Presupuesto**

* Definición de los recursos humanos (equipo técnico, usuarios clave, gerente de proyecto).
* Recursos tecnológicos necesarios (servidores, bases de datos, herramientas de pruebas).
* Ajuste y validación del presupuesto total.

1. **ANÁLISIS DE REQUERIMIENTOS**

## **7.1. Propósito**

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

## **7.2. 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.

## **7.3. 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.

## **7.4. Requerimientos funcionales**

### **7.4.1. *BBU-0001 Cargue masivo de reglas contables***

Tabla 1. 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. |

### **7.5. *Requerimientos detallados***

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

Tabla 2. Requerimiento de 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

Tabla 3. Requerimiento de Pantalla de Visualización de 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 |

## **7.6. Casos de Uso**

Tabla 4. 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. |

# **DISEÑO**

## **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.

* 1. **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.

Ilustración 1. Arquitectura de la solución



* + 1. **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.

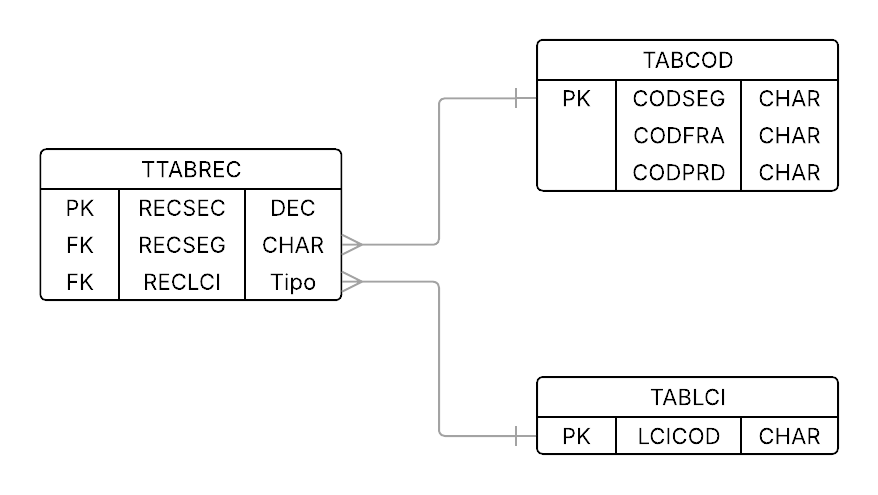
**¿Qué hace?**

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.

* 1. **Diseño de la Base de Datos**

### **Diagrama de Entidad-Relación (ERD)**

Ilustración 2. 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:

Tabla 5. Reglas contables TABREC

|  |  |  |  |
| --- | --- | --- | --- |
| 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:

Tabla 6. Temporal de reglas contables - TTABREC

|  |  |  |  |
| --- | --- | --- | --- |
| 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:

Tabla 7. Tipos de segmentaciones - TABCOD

|  |  |  |  |
| --- | --- | --- | --- |
| 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:

Tabla 8. Tipos de segmentación - TABLCI

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

# **REGLAS DE NEGOCIO Y LÓGICA DE PROCESO**

## **9.1. 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.

## **9.2. 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.
  1. **Reglas de Comportamiento del Sistema**

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

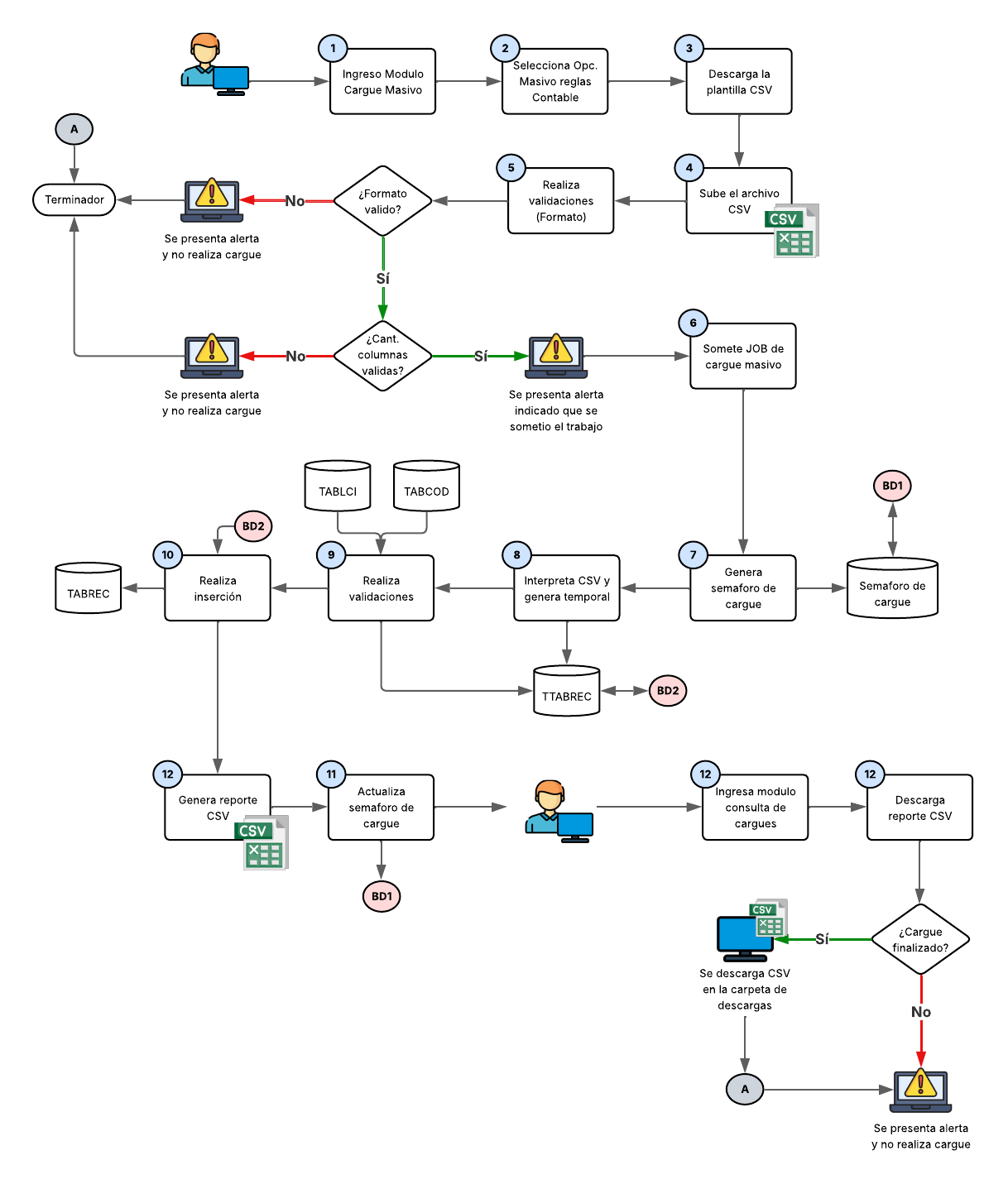
* **Procesamiento Asíncrono:** La carga masiva se ejecutará como un batch job en segundo plano. Esto asegura que el frontend no se bloquee, permitiendo al usuario continuar navegando mientras el proceso se completa.
* **Generación de Reporte Detallado:** Al finalizar el procesamiento, se debe generar 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 de actividades**

Es un diagrama que representa flujos de control entre varias actividades o acciones, enfatizando la secuencia, las condiciones y la concurrencia de dichos flujos

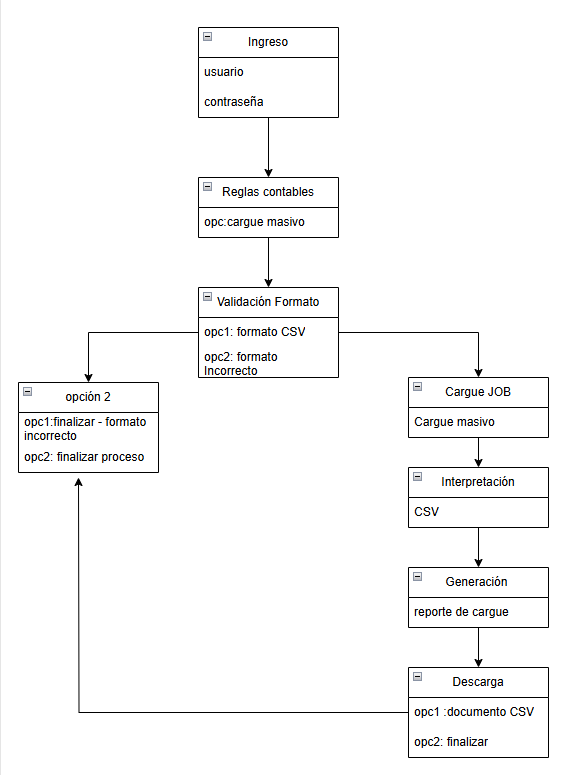
Ilustración 3. Diagrama de flujo del proceso de cargue masivo de reglas contables



## **Diagrama de clases**

Es un diagrama que describe la estructura estática del sistema, mostrando clases, sus atributos, operaciones y las relaciones entre clases.

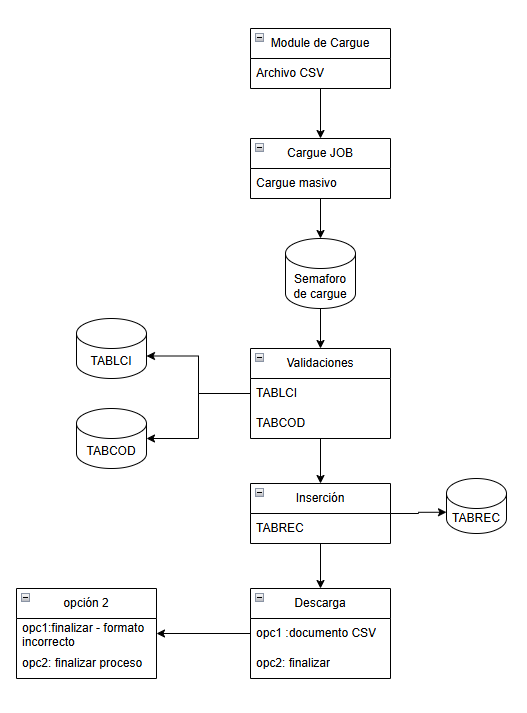
Ilustración 4. Diagrama de clases

****

* 1. **Diagrama de componentes**

Es un diagrama que representa los componentes del sistema, sus interfaces, y cómo están conectados y sus dependencias

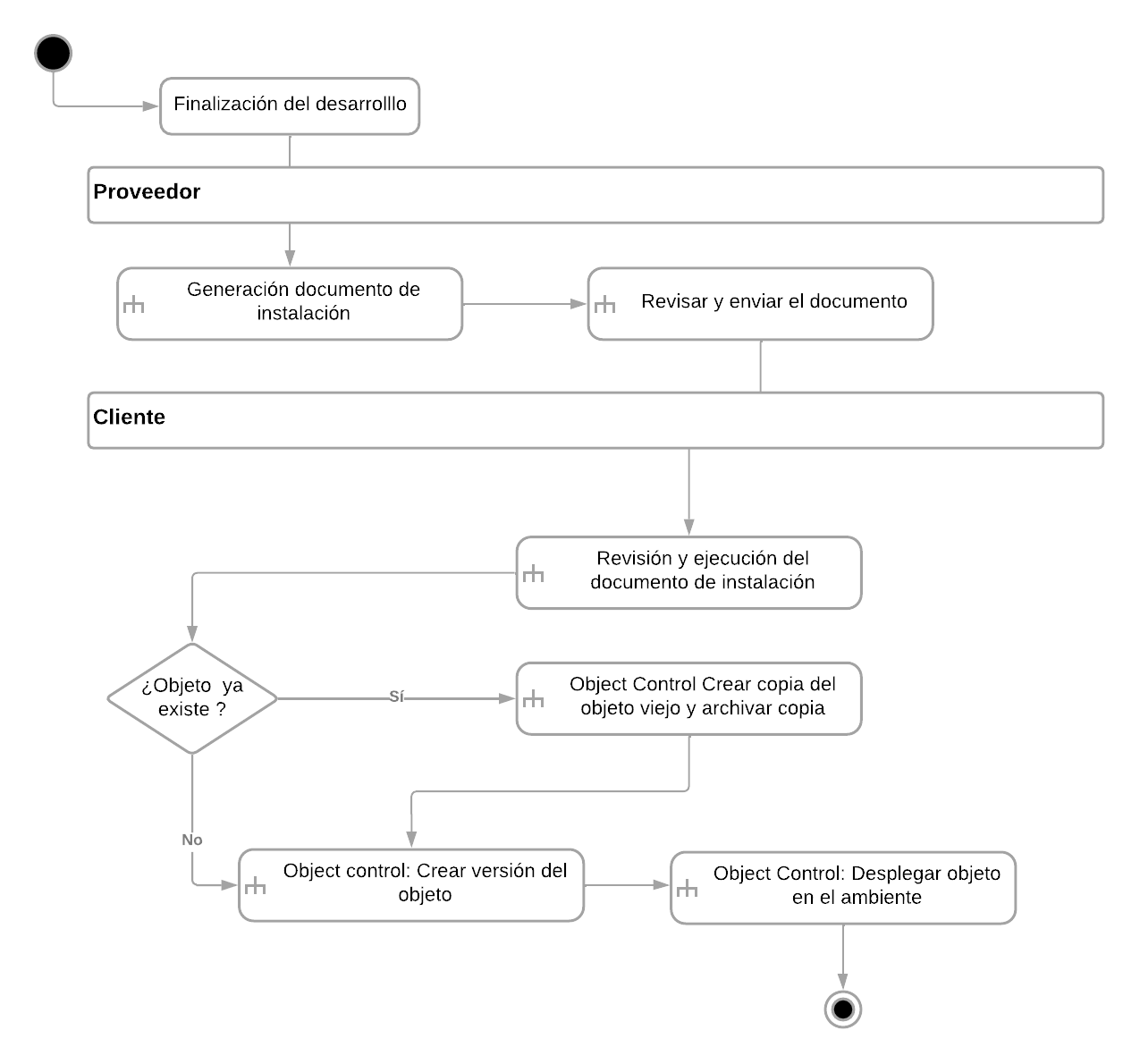
Ilustración 5. Diagrama de componentes



* 1. **Diagrama de despliegue**

Es un diagrama de estructura que muestra la arquitectura de ejecución del sistema: nodos físicos o de ejecución, artefactos de software desplegados en esos nodos, y las rutas de comunicación entre ellos

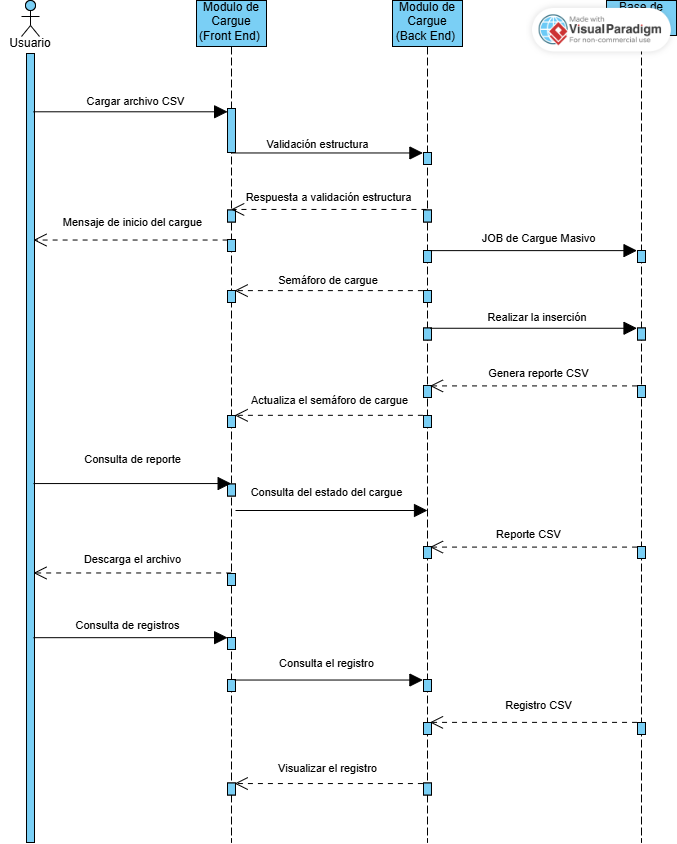
Ilustración 6. Diagrama de despliegue



* 1. **Diagrama de Secuencia**

Es un diagrama que muestra cómo los objetos intercambian mensajes a lo largo del tiempo, representando la secuencia temporal de dichos mensajes en un escenario particular del sistema

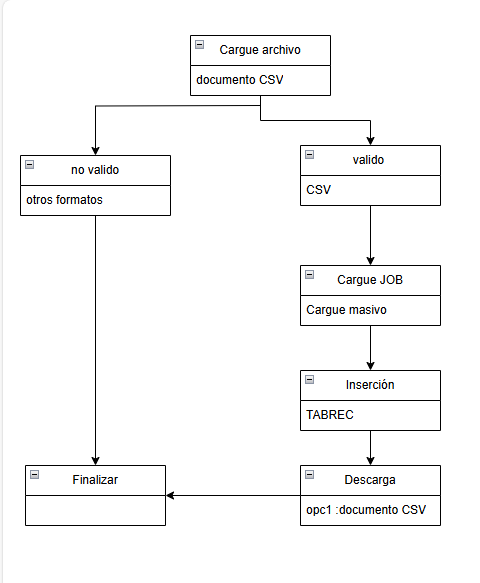
Ilustración 7. Diagrama de secuencia



* 1. **Diagrama de estado**

Es un diagrama de comportamiento que muestra los distintos estados por los que puede pasar una entidad en respuesta a eventos externos, junto con las acciones asociadas a estos cambios de estado.

Ilustración 8. Diagrama de estado

****

# **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 9. Interfaz del usuario para cargue de la plantilla

Tabla

El contenido generado por IA puede ser incorrecto.Ilustración 10. Interfaz del usuario consulta de cargues masivos

Ilustración 11. Consulta de registros cargados

Interfaz de usuario gráfica, Aplicación, Word

El contenido generado por IA puede ser incorrecto.

Ilustración 12. Consulta detallada de un registro del cargue

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

## **Plantilla de cargue**

La plantilla de cargue tiene que respetar el siguiente formato:

Tabla 9. Tabla de la plantilla de cargue

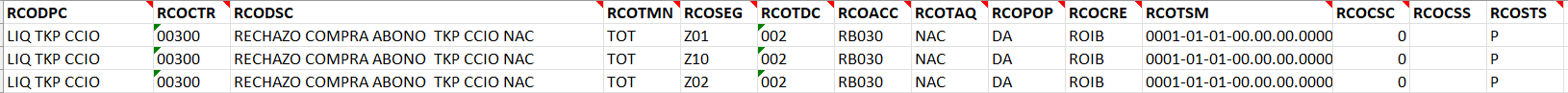
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nombre | Descripción | Tipo | Precisión | Escala |
| RECSEC | Numero único del registro | DECIMAL | 6 | 0 |
| RECCML | Canal de procesamiento | CHARACTER | 3 | 0 |
| RECCDP | Descripción del proceso que pertenece la regla contable | CHARACTER | 50 | 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 13. Ejemplo de cargue de la plantilla



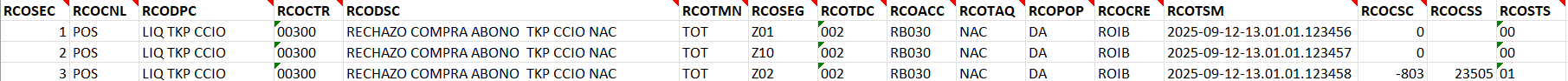
## **Reporte – Resultado del cargue masivo**

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

Tabla 10. Tabla de reporte de cargue masivo

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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 14. Resultado del cargue masivo

# **REFERENCIAS**

[1] <https://newo.ai/insights/banking-business-process-automation-efficiency-accuracy-and-innovation/?utm_source=chatgpt.com>

[2] <https://recursos.exportemos.pe/informe-de-mercado-para-software.pdf>

[3]https://assets.kpmg.com/content/dam/kpmg/co/pdf/2025/04/Colombia-Tech-Report-2024-vf.pdf