
Especificación de requisitos de software

Proyecto: EUREKA BANK SOAP JAVA
Revisión 2.0



Ficha del documento

Fecha	Revisión	Autor	Verificado dep. calidad.
10/12/2025	1.0	Añasco Silvia Enriquez Sheylee Salazar Augusto	Msg. Eduardo Campaña
28/01/2026	2.0	Añasco Silvia Enriquez Sheylee Salazar Augusto	Msg. Eduardo Campaña

Contenido

1 Introducción	3
1.1 Propósito	4
1.2 Alcance	4
1.3 Personal involucrado	4
1.4 Definiciones, acrónimos y abreviaturas	5
1.5 Referencias	5
1.6 Resumen	6
2 Descripción general	6
2.1 Perspectiva del producto	6
2.2 Funcionalidad del producto	6
2.3 Características de los usuarios	7
2.4 Restricciones	7
2.5 Suposiciones y dependencias	7
2.6 Evolución previsible del sistema	8
3 Requisitos específicos	8
3.1 Requisitos comunes de los interfaces	8
3.1.1 Interfaces de usuario	8
3.1.2 Interfaces de hardware	9
3.1.3 Interfaces de software	9
3.1.4 Interfaces de comunicación	9
3.2 Requisitos funcionales	10
3.2.1 RF-01: Autenticación de Usuario	10
3.2.2 RF-02: Realizar Depósito	10
3.2.3 RF-03: Realizar Retiro	10
3.2.4 RF-04: Realizar Transferencia	11
3.2.5 RF-05: Consultar Movimientos	11
3.3 Requisitos no funcionales	11
3.3.1 Requisitos de rendimiento	11
3.3.2 Seguridad	12
3.3.3 Fiabilidad	12
3.3.4 Disponibilidad	12
3.3.5 Mantenibilidad	13
3.3.6 Portabilidad	13
3.4 Otros requisitos	14

1 Introducción

El presente documento constituye la Especificación de Requisitos de Software (ERS) para el sistema EUREKA BANK SOAP JAVA. Este documento describe en detalle las necesidades funcionales y no funcionales identificadas para el desarrollo de la plataforma bancaria, sirviendo como medio de comunicación entre el equipo de desarrollo y los interesados en el proyecto. Su contenido abarca desde la descripción general del producto hasta los requisitos específicos de las interfaces y operaciones de la lógica de negocio.

1.1 Propósito

El propósito de este documento es definir de manera detallada y exhaustiva los requisitos funcionales y no funcionales para la versión 1.0 del sistema EUREKA BANK SOAP JAVA. Este documento establece la base técnica y funcional para el desarrollo de una solución bancaria multiplataforma basada en una arquitectura orientada a servicios (SOA) utilizando Java y SOAP.

El documento servirá como guía de referencia durante las fases de diseño, implementación, pruebas y validación del sistema, asegurando que el producto final cumpla con las expectativas de gestión de cuentas de ahorro y operaciones financieras.

1.2 Alcance

El alcance de este ERS se centra en la definición y desarrollo del sistema EUREKA BANK SOAP JAVA, un prototipo de banca electrónica diseñado para la gestión de operaciones financieras primarias

El sistema implementa las funcionalidades esenciales de la lógica de negocio, incluyendo el registro atómico de depósitos, retiros y transferencias, y la consulta detallada de movimientos históricos. No incluye la gestión de préstamos, inversiones, o transferencias interbancarias (ACH/SWIFT). El foco es la funcionalidad básica de cuenta.

Se desarrollará un Servicio Web basado en el protocolo SOAP (Simple Object Access Protocol) utilizando el framework JAX-WS en Java. Este servicio actúa como la capa de negocio central..

Es necesaria la utilización de MySQL Server como un servicio de Base de Datos Gestionada en la Nube (DBaaS), asegurando alta disponibilidad y encriptación en reposo. La gestión de la infraestructura subyacente de la nube (servidores físicos, balanceo de carga) es responsabilidad del proveedor de servicios de nube Google Cloude, no del equipo de desarrollo

Finalmente, se deberá desarrollar múltiples clientes (Escritorio, Web y Móvil - Android) para demostrar la interoperabilidad del servicio SOAP, a la vez que se toma en cuenta la implementación de interfaces de usuario final amigables.

1.3 Personal involucrado

Tabla 1

Nombre	Silvia Añasco
Rol	Líder del equipo
Categoría profesional	Estudiante de Ingeniería en Software

Responsabilidades	Implementación de cliente Android y Pruebas Unitarias.
Información de contacto	sianasco@espe.edu.ec
Aprobación	

Tabla 2

Nombre	Sheylee Enriquez
Rol	Miembro del equipo de desarrollo
Categoría profesional	Estudiante de Ingeniería en Software
Responsabilidades	Desarrollo de Interfaces (Consola/Escritorio) y consumo de servicios.
Información de contacto	saenriquez2@espe.edu.ec
Aprobación	

Tabla 3

Nombre	Augusto Salazar
Rol	Miembro del equipo de desarrollo
Categoría profesional	Estudiante de Ingeniería en Software
Responsabilidades	Diseño de Base de Datos MySQL y Lógica del Servicio Web SOAP.
Información de contacto	casalazar@espe.edu.ec
Aprobación	

1.4 Definiciones, acrónimos y abreviaturas

A continuación, se presentan las definiciones, acrónimos y abreviaturas necesarios para interpretar este documento de manera apropiada:

Definiciones:

- **SOAP:** Simple Object Access Protocol. Protocolo de mensajería para servicios web.
- **MVC:** Modelo-Vista-Controlador. Patrón de arquitectura de software.
- **JAX-WS:** Java API for XML Web Services. Facilita el desarrollo de servicios SOAP.

Acrónimos y abreviaturas:

- **SOA:** Service Oriented Architecture - Arquitectura Orientada a Servicios
- **DBMS:** Database Management System - Sistema Gestor de Bases de Datos (MySQL Server en este caso)
- **ERS:** Especificación de Requisitos de Software.
- **WSDL:** Web Services Description Language.
- **VPC:** Virtual Private Connection - Red Privada Virtual

1.5 Referencias

Tabla 4

Referencia	Título	Ruta	Fecha	Autor
REF-001	Formato IEEE 830	Formato ieee830 PDF	01/09/2018	Guaqueta Daniel

1.6 Resumen

El presente documento define las especificaciones para el sistema EurekaBank, una solución de banca electrónica basada en una arquitectura orientada a servicios (SOA) que permite la gestión centralizada de cuentas de ahorro y transacciones financieras.

El documento se organiza en las siguientes secciones:

- **Introducción:** Se describen los antecedentes, propósitos y el alcance del proyecto, identificando a las partes interesadas, además de establecer el contexto técnico (Java SOAP) y los objetivos del sistema propuesto.
- **Descripción General:** Se proporciona una visión general de los objetivos y características del producto, incluyendo su adaptación a los procesos del negocio bancario, las restricciones operativas, las suposiciones y las dependencias tecnológicas.
- **Requisitos específicos:** Se detallan los requisitos funcionales, específicamente la autenticación de usuarios (Login), la gestión de depósitos y retiros, la realización de transferencias y la consulta de historial de movimientos. Asimismo, se especifican los requisitos no funcionales relacionados con el rendimiento de los servicios web, la seguridad de las transacciones y la usabilidad de las interfaces cliente.
- **Apéndices:** Se incluye información adicional complementaria, como el glosario de términos técnicos y referencias bibliográficas esenciales para el desarrollo.

2 Descripción general

2.1 Perspectiva del producto

El sistema EUREKA BANK SOAP JAVA se posiciona como un producto central e independiente dentro de un ecosistema bancario, adhiriéndose a una Arquitectura Orientada a Servicios (SOA).

- **Componente Servidor:** Se erige como el eje principal. Es una aplicación Java que aloja el servicio web **SOAP (WSEureka)**. Su función es encapsular la lógica de negocio (Patrón MVC) y actuar como el único punto de contacto entre las aplicaciones cliente y el subsistema de persistencia.
- **Componente de Persistencia (Cloud):** La base de datos MySQL ha sido migrada a la nube (DBaaS) de Google Cloude, lo que la convierte en una dependencia crítica de red. La capa de servicio se conecta a este *endpoint* remoto para ejecutar todas las transacciones.
- **Componentes Clientes:** Múltiples aplicaciones (Escritorio, Web, Móvil) consumen el WSDL expuesto por el servidor, actuando como la capa de presentación que orquesta las solicitudes de los usuarios a través del protocolo SOAP.

El sistema debe operar como un *middleware* transaccional, garantizando la consistencia de los datos en un entorno de red distribuido.

2.2 Funcionalidad del producto

El sistema EUREKA BANK SOAP JAVA es una solución multiplataforma que centraliza la lógica de negocio en un Servidor de Aplicaciones Java y expone sus funciones mediante servicios web SOAP.

Las funcionalidades principales son:

- **Autenticación (Login):** Verificación de credenciales de usuario contra la base de datos utilizando encriptación SHA.
- **Gestión de Depósitos:** Permite ingresar dinero a una cuenta específica, actualizando el saldo y registrando el movimiento con fecha y hora.
- **Gestión de Retiros:** Permite la salida de dinero de una cuenta, validando la disponibilidad de fondos.
- **Transferencias:** Permite mover fondos entre una cuenta origen y una cuenta destino de manera atómica (transaccional).
- **Consulta de Movimientos:** Permite visualizar el historial de transacciones (depósitos, retiros, transferencias) y el saldo actual de una cuenta específica.

2.3 Características de los usuarios

Tabla 5

Tipo de usuario	Administrador
Formación	Educación de tercer nivel
Habilidades	Conocimiento en protocolos de gestión bancaria y operativa del sistema (front-end).
Actividades	Realizar depósitos/retiros/transferencias a cuentas de clientes y consultar el historial de movimientos de las cuentas bajo su responsabilidad.

2.4 Restricciones

Las restricciones definen el entorno tecnológico y operacional ineludible para el proyecto:

- **Tecnología del Servicio:** La comunicación entre cliente y servidor debe ser implementada con el protocolo SOAP, limitando el uso de alternativas como REST.
- **Lenguaje y Frameworks:** La totalidad del código de negocio y servicio debe ser desarrollado en el lenguaje Java (JDK 17 o superior, como se sugiere por las prácticas modernas de Java) y utilizando JAX-WS para los servicios web.
- **Persistencia:** Se utilizará MySQL Server como sistema gestor de bases de datos.
- **Restricción de Infraestructura (Nube):** La Base de Datos debe estar implementada como un servicio gestionado en la nube (DBaaS). Esto impone la restricción de que toda conexión debe ser encriptada (TLS/SSL), y el acceso debe estar restringido mediante políticas de firewall de la nube.
- **Patrón Arquitectónico:** El servidor debe seguir el patrón Modelo-Vista-Controlador (MVC), con separación estricta de responsabilidades entre las capas.

2.5 Suposiciones y dependencias

La implementación del proyecto está basada en una serie de suposiciones y dependencias que podrían afectar el cumplimiento de los requisitos de software. Si alguno de estos factores cambia, es posible que se necesiten modificaciones en los requisitos y en la planificación del proyecto.

Suposiciones:

- El endpoint de la base de datos en la nube está operativo, accesible y protegido por credenciales.

- El servidor de aplicaciones Java y la instancia de Base de Datos en la nube residen dentro de una VPC o que tienen rutas de red seguras configuradas para la comunicación *back-end*.
- Los clientes finales tienen conectividad de red (Internet/Intranet) al servidor de aplicaciones (Payara Server).

Dependencias:

- Se depende de la resolución de nombres de dominio y la configuración correcta de los Grupos de Seguridad (Security Groups) de la nube.
- Las definiciones de esquema de la base de datos (tablas Cuenta, Movimiento, TipoMovimiento, etc.) son precisas y han sido cargadas correctamente.

Si cualquiera de estas suposiciones o dependencias cambia, se deberá reevaluar el alcance y los requisitos del sistema para asegurar que siga cumpliendo con las expectativas de sus usuarios.

2.6 Evolución previsible del sistema

- Migración de la infraestructura a la nube para mejorar la escalabilidad.
- Implementación de autenticación mediante Tokens (JWT) para mayor seguridad en los clientes móviles.
- Módulo de reportes en PDF para los estados de cuenta mensuales.

3 Requisitos específicos

3.1 Requisitos comunes de los interfaces

Este apartado define las interfaces que el sistema EUREKA BANK SOAP JAVA utilizará para interactuar tanto con los usuarios como con otros componentes de hardware y software necesarios para su correcto funcionamiento.

3.1.1 Interfaces de usuario

El sistema presenta una interfaz gráfica estandarizada en sus versiones Web y Escritorio, utilizando formularios claros para la entrada de datos.

- **Pantalla de Login:** Presenta campos para "Usuario" y "Contraseña" con un botón de "Ingresar". Incluye el logo de Eureka Savings Bank.
- **Menú Principal:** Barra de navegación o botones centrales que permiten seleccionar: Depósitos, Retiros, Transferencias o Consultas.
- **Pantalla de Operaciones (Depósito/Retiro/Transferencia):** Formularios que solicitan el "Código de Cuenta" (y cuenta destino si aplica) y el "Importe". Muestran mensajes de confirmación en verde (éxito) o rojo (error).
- **Pantalla de Consulta:** Muestra el saldo actual y una tabla con los movimientos.

3.1.2 Interfaces de hardware

Los requisitos de hardware se dividen en dos categorías, siendo la más crítica la del servidor que aloja el servicio:

- Servidor de Aplicaciones: El hardware que aloja el Web Service (Servidor) debe tener la capacidad de ejecutar un sistema operativo compatible con Java (ej. Linux o Windows Server), y proveer los recursos (CPU, RAM) necesarios para sostener el servidor de aplicaciones (Payara Server) bajo una carga de trabajo concurrente.
- Dispositivos Cliente: Se requiere un dispositivo final (PC, laptop, smartphone o tablet) con la capacidad de ejecutar un navegador web o el sistema operativo Android, así como conectividad de red para consumir el servicio.

3.1.3 Interfaces de software

La implementación funcional del sistema EUREKA BANK SOAP JAVA establece una dependencia directa con una pila tecnológica específica en sus componentes principales.

El módulo Servidor se basa en el lenguaje Java y debe ser desplegado en un contenedor de aplicaciones robusto como Payara Server. La definición del servicio web se gestiona mediante el framework JAX-WS (Java API for XML Web Services), esencial para la correcta serialización y exposición del protocolo SOAP.

A nivel de arquitectura interna, el código del servidor se rige por el patrón Modelo-Vista-Controlador (MVC), garantizando la separación de la lógica de negocio, la presentación y la capa de persistencia. Esta última capa se comunica con MySQL Server, que funge como un Servicio Gestionado de Base de Datos en la Nube (DBaaS), requiriendo el uso de un driver de conectividad JDBC para establecer la comunicación segura y transaccional. Por el lado del Cliente Móvil, la capacidad de consumir las operaciones SOAP requiere la integración de librerías especializadas como KSoap2, adaptando el protocolo a las capacidades del entorno Android.

3.1.4 Interfaces de comunicación

El sistema se basa en una arquitectura orientada a servicios, lo que requiere la definición clara de los protocolos de comunicación:

- **Protocolo de Servicios (Cliente-Servidor): SOAP (Simple Object Access Protocol)**, versión 1.1. Este protocolo es mandatorio y define el formato de los mensajes XML intercambiados.
- **Definición de Servicios: WSDL (Web Services Description Language)**. El servidor debe exponer su WSDL para que los clientes puedan generar los stubs de conexión y entender las operaciones disponibles.
- **Protocolo de Transporte (Servidor/Cliente): HTTP/HTTPS**. Se debe privilegiar el uso de HTTPS en un entorno de producción para la encriptación del tráfico de mensajes SOAP.
- **Protocolo de Transporte (Servidor/BD en Nube): JDBC sobre TCP/IP**. La conexión a la base de datos en la nube debe realizarse a través de un endpoint FQDN (Fully Qualified Domain Name) y, obligatoriamente, debe utilizar **SSL/TLS** para la encriptación de datos sensibles en tránsito.

3.2 Requisitos funcionales

3.2.1 RF-01: Autenticación de Usuario

Tabla 6

Número de requisito	RF-01		
Nombre de requisito	Autenticación de Usuario		
Tipo	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
Fuente del requisito	Ing. Mauricio Campaña		
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/ Opcional

Descripción: El sistema debe permitir el acceso solo a usuarios registrados y activos.

Entrada: Usuario, contraseña.

Proceso: El sistema envía las credenciales al servicio web WSEureka.login. El servicio verifica en la tabla Usuario comparando el hash SHA de la contraseña.

Salida: Acceso al menú principal o mensaje de "Credenciales incorrectas".

3.2.2 RF-02: Realizar Depósito

Tabla 7

Número de requisito	RF-02		
Nombre de requisito	Realizar Depósito		
Tipo	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
Fuente del requisito	Ing. Mauricio Campaña		
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/ Opcional

Descripción: Permitir el incremento de saldo en una cuenta.

Entrada: Código de Cuenta, Importe.

Proceso: Se invoca el método registrarDeposito. El sistema verifica que la cuenta exista y esté 'ACTIVA', suma el importe al saldo actual, incrementa el contador de movimientos y registra el movimiento en la tabla Movimiento con tipo '003' (Depósito).

Salida: Mensaje de éxito con el nuevo saldo o error si la cuenta no existe.

3.2.3 RF-03: Realizar Retiro

Tabla 8

Número de requisito	RF-03		
Nombre de requisito	Realizar Retiro		
Tipo	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
Fuente del requisito	Ing. Mauricio Campaña		
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/ Opcional

Descripción: Permitir la disminución de saldo por retiro de efectivo.

Entrada: Código de Cuenta, Importe.

Proceso: El sistema verifica que el saldo sea mayor o igual al importe solicitado. Si es válido, resta el importe, actualiza el saldo y registra el movimiento con tipo '004' (Retiro).

Salida: Confirmación de retiro o mensaje "Fondos Insuficientes".

3.2.4 RF-04: Realizar Transferencia

Tabla 9

Número de requisito	RF-04		
Nombre de requisito	Realizar Transferencia		
Tipo	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
Fuente del requisito	Ing. Mauricio Campaña		
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/ Opcional

Descripción: Transferir fondos de una cuenta origen a una destino.

Entrada: Cuenta Origen, Cuenta Destino, Importe.

Proceso: Debe ejecutarse como una transacción atómica (ACID). Se debita de la cuenta origen y se acredita en la cuenta destino. Si falla alguna de las dos, se hace *rollback* de toda la operación.

Salida: Mensaje "Transferencia realizada con éxito".

3.2.5 RF-05: Consultar Movimientos

Tabla 10

Número de requisito	RF-05		
Nombre de requisito	Consultar Movimientos		
Tipo	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
Fuente del requisito	Ing. Mauricio Campaña		
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/ Opcional

Descripción: Ver el historial de una cuenta.

Entrada: Código de Cuenta.

Proceso: El servicio traerMovimientos realiza un SELECT con INNER JOIN entre las tablas Movimiento y TipoMovimiento filtrando por el código de cuenta.

Salida: Lista de objetos Movimiento contenido fecha, tipo, acción e importe, visualizados en una tabla.

3.3 Requisitos no funcionales

3.3.1 Requisitos de rendimiento

Tabla 11

Número de requisito	RNF-01		
Nombre de requisito	Tiempo de respuesta de transacciones		
Tipo	<input type="checkbox"/> Requisito	<input checked="" type="checkbox"/> Restricción	
Fuente del requisito	Equipo de desarrollo		
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/ Opcional

Descripción:

El servicio web SOAP (WSEureka) debe procesar las peticiones de operaciones financieras (depósitos, retiros y transferencias) y devolver una respuesta al cliente en un tiempo no mayor a 3 segundos bajo condiciones normales de carga, garantizando una experiencia fluida para el usuario final.

3.3.2 Seguridad

Tabla 12

Número de requisito	RNF-02
Nombre de requisito	Contraseñas seguras
Tipo	<input type="checkbox"/> Requisito <input checked="" type="checkbox"/> Restricción
Fuente del requisito	Equipo de desarrollo
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Descripción:

El sistema debe utilizar la función SHA para el almacenamiento de contraseñas de usuario

Tabla 13

Número de requisito	RNF-03
Nombre de requisito	Control de Acceso de Red
Tipo	<input type="checkbox"/> Requisito <input checked="" type="checkbox"/> Restricción
Fuente del requisito	Equipo de desarrollo
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Descripción:

El acceso a la instancia de la BD en la nube debe estar estrictamente restringido mediante reglas de firewall para permitir la conexión únicamente desde la dirección IP privada del servidor de aplicaciones de Java.

3.3.3 Fiabilidad

Tabla 14

Número de requisito	RNF-04
Nombre de requisito	Atomicidad de las operaciones
Tipo	<input type="checkbox"/> Requisito <input checked="" type="checkbox"/> Restricción
Fuente del requisito	Equipo de desarrollo
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Descripción:

Se debe garantizar la atomicidad de las operaciones de transferencia, depósito y retiro, mediante el manejo de transacciones (uso de cn.setAutoCommit(false), cn.commit(), y cn.rollback()) para asegurar que la operación se complete por completo o no se realice en absoluto.

3.3.4 Disponibilidad

Tabla 15

Número de requisito	RNF-05
Nombre de requisito	Disponibilidad del servidor

Tipo	<input type="checkbox"/> Requisito	<input checked="" type="checkbox"/> Restricción
Fuente del requisito	Equipo de desarrollo	
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Descripción:

El servicio web debe estar disponible para las aplicaciones cliente (Escritorio, Web, Móvil) de forma continua durante las horas de operación.

Tabla 16

Número de requisito	RNF-06
Nombre de requisito	Disponibilidad de la base de datos
Tipo	<input type="checkbox"/> Requisito <input checked="" type="checkbox"/> Restricción
Fuente del requisito	Equipo de desarrollo
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Descripción:

La base de datos debe estar desplegada en una configuración Multi-AZ (Múltiples Zonas de Disponibilidad) para asegurar una disponibilidad mínima (99.95% o superior, mientras prendida la instancia).

3.3.5 Mantenibilidad

Tabla 17

Número de requisito	RNF-07
Nombre de requisito	Mantenibilidad
Tipo	<input type="checkbox"/> Requisito <input checked="" type="checkbox"/> Restricción
Fuente del requisito	Equipo de desarrollo
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Descripción:

El código debe seguir una arquitectura modular (MVC) y utilizar paquetes bien definidos para facilitar la localización y modificación del código fuente

3.3.6 Portabilidad

Tabla 18

Número de requisito	RNF-08
Nombre de requisito	Portabilidad
Tipo	<input type="checkbox"/> Requisito <input checked="" type="checkbox"/> Restricción
Fuente del requisito	Equipo de desarrollo

Descripción:

La lógica de los clientes debe ser compatible con múltiples plataformas (Escritorio, Web, Consola, Móvil - Android).

3.4 Otros requisitos

Tabla 19

Número de requisito	RL-01
Nombre de requisito	Integridad y Auditoría de Movimientos

Tipo	<input type="checkbox"/> Requisito	<input checked="" type="checkbox"/> Restricción
Fuente del requisito	Equipo de desarrollo	
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Descripción:

Para cumplir con normas de auditoría financiera, el sistema no debe permitir la eliminación o modificación de registros en la tabla Movimiento una vez insertados. Toda corrección debe realizarse mediante un contra-movimiento.

Tabla 20

Número de requisito	RC-01	
Nombre de requisito	Localización e Idioma	
Tipo	<input type="checkbox"/> Requisito	<input checked="" type="checkbox"/> Restricción
Fuente del requisito	Equipo de desarrollo	
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Descripción:

Todas las interfaces gráficas (Web, Consola, Escritorio y Móvil) y los mensajes de error del sistema deben presentarse en idioma Español. Los formatos de moneda deben respetar la configuración regional, mostrando el símbolo de moneda (\$) y dos decimales para los importes.