



Universidad Tecnológica de **Tijuana**

# PLANTILLA DE ESPECIFICACIÓN Y VALIDACIÓN DE REQUERIMIENTOS

## **INTEGRANTES:**

Cárdenas Zalapa Jesús Antonio

Medina Palacios Fernando

Rodríguez Rosales Jesús Antonio

Ramírez Aispuro Juan Jose

Arguelles Galvez Antonio

## **ASIGNATURA:**

Desarrollo Móvil Integral

## **GRUPO:**

10-B

## **PARCIAL:**

Primero

## **FECHA DE CREACIÓN:**

Miércoles 22 de enero de 2025

## ***ESTANDAR IEEE-830***

---

## ***ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE***

Proyecto:

Modulo: [Nombre del Módulo]

**FICHA DEL DOCUMENTO**

Fecha	Revisión	Autor	Verificó

*Insertar una fila por cada revisión, en la tabla anterior. La columna Revisión, se refiere al número de revisión.*

**VALIDACIONES DEL DOCUMENTO**

Documento validado por las partes en fecha:

Por el cliente	Por la empresa suministradora

# INDICE DE CONTENIDO

<b>FICHA DEL DOCUMENTO</b>	<b>3</b>
<b>INDICE DE CONTENIDO</b>	<b>4</b>
<b>1 INTRODUCCIÓN</b>	<b>5</b>
1.1 PROPÓSITO	5
1.2 ALCANCE	5
1.3 PERSONAL INVOLUCRADO	5
1.4 DEFICIONES, ACRÓNIMOS Y ABREVIATURAS	5
1.5 REFERENCIAS	6
1.6 RESUMEN	6
<b>2 DESCRIPCIÓN GENERAL</b>	<b>6</b>
2.1 PERSPECTIVA DEL PRODUCTO	6
2.2 FUNCIONALIDAD DEL PRODUCTO	6
2.3 CARACTERISTICAS DE LOS USUARIOS	6
2.4 RESTRICCIONES	7
2.5 SUPOSICIONES Y DEPENDENCIAS	7
2.6 EVOLUCIÓN PREVISIBLE DEL SISTEMA	7
<b>3 REQUISITOS ESPECIFICOS</b>	<b>7</b>
<b>3.1 REQUISITOS COMUNES DE LAS INTERFACES</b>	<b>8</b>
3.1.1 INTERFACES DE USUARIO	8
3.1.2 INTERFACES DE HARDWARE	8
3.1.3 INTERFACES DE SOFTWARE	9
3.1.4 INTERFACES DE COMUNICACIÓN	9
<b>3.2 REQUISITOS NO FUNCIONALES</b>	<b>9</b>
3.2.1 REQUISITOS DE RENDIMIENTO	9
3.2.2 REQUISITOS DE SEGURIDAD	9
3.2.3 REQUISITOS DE FIABILIDAD	9
3.2.4 REQUISITOS DE DISPONIBILIDAD	10
3.2.5 REQUISITOS DE MANTENIBILIDAD	10
3.2.6 PORTABILIDAD	10
<b>3.4 OTROS REQUISITOS</b>	<b>10</b>

# 1 INTRODUCCIÓN

*A lo largo de esta especificación de requisitos de software se observará el planteamiento de un objetivo claro que llevará a la resolución de un problema que si bien no se considera atenuante en el día a día de las personas, dentro del área industrial se podría considerar como una herramienta o solución beneficiosa para los usuarios de la misma. El objetivo es esclarecer el camino, definiciones y perspectivas que puede tomar el cumplimiento de la vista general u el alcance dentro de lo esperado para la implementación de esta herramienta.*

## 1.1 PROPÓSITO

*A lo largo del documento se abordará la problemática o área de oportunidad que se identifique dentro del rubro de contraloría como puede ser la realización de las auditorías y el compuesto de presupuestos con su respectiva factura y su seguimientos, desde que es lo que se está generando, por quién, para quién y los detalles necesarios para en caso de cualquier auditoría tener los registros de movimientos realizados disponibles con las regulaciones necesarias.*

### ▪ 1.1.2 Audiencia a la que va dirigido:

*La audiencia a la que va dirigido el documento es para los profesionales dentro de áreas dentro de la industria financiera como por ejemplo un contralor, dpto de finanzas, recursos humanos entre muchos otros departamentos, así como los docentes calificados para evaluar el sistema solución y el planteamiento del proyecto en cuestión.*

## 1.2 ALCANCE

Contralyze considera abarcar las funciones relacionadas con las finanzas dentro de una organización como puede ser el área de contraloría ya que permitirá realizar las funciones de transacciones manteniendo la realización y gestionar reportes así como presupuestos vinculados a proveedores y clientes dentro de la organización, así mismo se permitirá realizar facturas y recibos asociados tanto a clientes como a proveedores, manteniendo así un sistema de historial eficiente y accesible.

Presentará un sistema de historial donde permitirá manejar las diferentes interacciones y transacciones que mejoren la trazabilidad y la toma de decisiones.

## 1.3 PERSONAL

### INVOLUCRADO

<b>Nombre</b>	Cárdenas Zalapa Jesús Antonio
<b>Rol</b>	Desarrollador full stack
<b>Categoría profesional</b>	Ingeniería en desarrollo de software
<b>Responsabilidades</b>	Implementación de interfaces de usuario y lógica del servidor.
<b>Información de contacto</b>	cardenaszalapajesus@gmail.com
<b>Aprobación</b>	

<b>Nombre</b>	Medina Palacios Fernando
<b>Rol</b>	Desarrollador full stack
<b>Categoría profesional</b>	Ingeniería en desarrollo de software
<b>Responsabilidades</b>	Desarrollo y mantenimiento de componentes tanto front-end como back-end, integración de APIs.
<b>Información de contacto</b>	
<b>Aprobación</b>	

<b>Nombre</b>	Rodríguez Rosales Jesús Antonio
<b>Rol</b>	Desarrollador full stack
<b>Categoría profesional</b>	Ingeniería en desarrollo de software
<b>Responsabilidades</b>	Diseño e implementación de bases de datos, creación de microservicios y seguridad de la aplicación.
<b>Información de contacto</b>	0321101313@ut-tijuana.edu.mx
<b>Aprobación</b>	

<b>Nombre</b>	Ramirez Aispuro Juan Jose
<b>Rol</b>	Desarrollador full stack
<b>Categoría profesional</b>	Desarrollo de componentes React/Next.js para la PWA y lógica de negocio en Node.js/Express.

<b>Responsabilidades</b>	Responsable de desarrollar e implementar software backend y frontend, así como realizar y fortalecer el área de base de datos.
<b>Información de contacto</b>	0321101713@ut-tijuana.edu.mx
<b>Aprobación</b>	

<b>Nombre</b>	Arguelles Galvez Antonio
<b>Rol</b>	Desarrollador full stack
<b>Categoría profesional</b>	Ingeniería en desarrollo de software
<b>Responsabilidades</b>	Integración y despliegue continuo (CI/CD), mantenimiento de entornos y gestión de versiones.
<b>Información de contacto</b>	0320127747@ut-tijuana.edu.mx
<b>Aprobación</b>	

*Relación de personas involucradas en el desarrollo del sistema, con información de contacto.*

*Esta información es útil para que el gestor del proyecto pueda localizar a todos los participantes y recabar la información necesaria para la obtención de requisitos, validaciones de seguimiento, etc.*

## 1.4 DEFINICIONES, ACRÓNIMOS Y ABREVIATURAS

<b>NOMBRE</b>	<b>DESCRIPCIÓN</b>
<i>Api</i>	<i>Conjunto de reglas y protocolos que permiten la comunicación entre diferentes aplicaciones o sistemas.</i>
<i>Framework</i>	<i>Estructura de software que facilita el desarrollo de aplicaciones al proporcionar herramientas y librerías predefinidas.</i>
<i>Base de datos</i>	<i>Sistema que almacena, organiza y gestiona datos de manera estructurada para su fácil acceso y manipulación.</i>
<i>Contralyzer</i>	<i>Nombre de la aplicación relacionado con finanzas específicamente contraloría.</i>
<i>Node</i>	<i>Entorno de ejecución de JavaScript que permite desarrollar aplicaciones del lado del servidor.</i>
<i>PWA</i>	<i>Aplicaciones web que ofrecen experiencia similar a las apps nativas, funcionando offline y con notificaciones push.</i>
<i>Software</i>	<i>Conjunto de programas y datos que permiten ejecutar tareas en un dispositivo electrónico.</i>
<i>BD</i>	<i>Abreviación de base de datos, sistema de almacenamiento y gestión de información.</i>

<b>Auditoría</b>	<i>Proceso de revisión y análisis de sistemas, procesos o cuentas para evaluar su cumplimiento y eficiencia.</i>
<b>Regulación</b>	<i>Conjunto de normas y leyes que rigen una industria o actividad para garantizar su correcto funcionamiento.</i>
<b>Entidad</b>	<i>Hace referencia a quienes regulan y mantienen actualizadas las reglas y pautas para mantener todo en orden.</i>
<b>React Native</b>	<i>Framework de desarrollo para crear aplicaciones móviles nativas usando JavaScript y React.</i>
<b>Expo</b>	<i>Plataforma y conjunto de herramientas que facilita el desarrollo y prueba de aplicaciones en React Native sin necesidad de configuración compleja.</i>
<b>JavaScript</b>	<i>Lenguaje de programación de alto nivel utilizado principalmente para el desarrollo web, tanto en frontend como en backend.</i>
<b>Service Worker</b>	<i>Script que actúa como un proxy entre la web y el navegador, permitiendo funcionalidades como el almacenamiento en caché y el funcionamiento offline.</i>
<b>Manifest</b>	<i>Archivo JSON que define la configuración y comportamiento de una Progressive Web App (PWA), incluyendo iconos, colores y pantalla de inicio.</i>
<b>JS</b>	<i>Abreviatura de JavaScript.</i>
<b>MVC</b>	<i>(Model-View-Controller) Patrón de arquitectura de software que separa la lógica de negocio (Modelo), la interfaz de usuario (Vista) y la gestión de datos (Controlador) para mejorar la organización y mantenimiento del código.</i>

## 1.5

### REFERENCIAS

Referencia	Título	Fecha	Autor u organización
<a href="#">Tarjeta de Crédito Empresarial y Plataforma de Gestión de Gastos   Clara</a>	Clara	20 de enero de 2025	CF TECH, S.A. DE C.V., SOFOM ENR

## 2 DESCRIPCIÓN GENERAL

### 2.1 PERSPECTIVA DEL PRODUCTO

El producto es una **aplicación móvil para la gestión financiera empresarial**, diseñada para proporcionar al departamento de finanzas una herramienta eficiente, centralizada y accesible. Permitirá registrar y clasificar transacciones financieras, gestionar presupuestos, configurar pagos recurrentes y almacenar facturas y recibos, vinculándolos a clientes y proveedores. Además, proporcionará reportes visuales y análisis interactivos para monitorear el desempeño financiero y tomar decisiones estratégicas basadas en datos.

En el ámbito de facturación, la aplicación ofrecerá la capacidad de registrar y almacenar facturas y recibos asociados tanto a clientes como a proveedores. Estas facturas podrán gestionarse en términos de estado, facilitando el seguimiento de pagos pendientes, realizados o vencidos. La integración de información de clientes y proveedores permitirá mantener un historial de interacciones y transacciones que mejorará la trazabilidad y la toma de decisiones.

### 2.2 FUNCIONALIDAD DEL PRODUCTO

#### 1. Gestión de Usuarios

- Permite registrar, autenticar y administrar usuarios del sistema.
- Define roles y permisos específicos para controlar el acceso a las funcionalidades de la aplicación.

#### 2. Gestión de Transacciones

- Registra ingresos y egresos con clasificación por categorías.
- Visualiza el historial de transacciones mediante reportes detallados.
- Ofrece filtros avanzados por fecha, tipo, categoría y usuario.

#### 3. Control de Presupuestos

- Crea y administra presupuestos para períodos específicos.
- Realiza seguimiento del gasto y compara contra los límites establecidos.
- Genera alertas si el presupuesto se excede o se aproxima al límite.

#### 4. Clasificación por Categorías



- Organiza transacciones en categorías de ingresos y egresos.
- Proporciona estadísticas por categoría para identificar tendencias de gasto o ingreso.

#### **5. Manejo de Pagos Recurrentes**

- Configura pagos automáticos con fechas de inicio y fin.
- Asigna categorías y frecuencias como diaria, semanal, mensual o anual.
- Genera notificaciones para los pagos próximos o vencidos.

#### **6. Gestión de Facturas y Recibos**

- Permite crear, almacenar y rastrear facturas y recibos emitidos o recibidos.
- Asocia cada factura a un cliente, proveedor o transacción específica.
- Muestra el estado de las facturas: "Pendiente", "Pagada" o "Vencida".

#### **7. Relación con Clientes y Proveedores**

- Registra información detallada de clientes y proveedores, incluyendo contactos, historial de transacciones y facturas.
- Facilita la generación de reportes para analizar relaciones comerciales.

#### **8. Reportes y Análisis Financieros**

- Ofrece reportes gráficos interactivos sobre ingresos, egresos, presupuestos y pagos recurrentes.
- Presenta resúmenes financieros diarios, mensuales y anuales.
- Ayuda a identificar patrones de comportamiento financiero y oportunidades de optimización.

## **2.3 CARACTERÍSTICAS DE LOS USUARIOS**

<b>Tipo de usuario</b>	Gerente Financiero
<b>Formación</b>	Licenciatura o maestría en Finanzas, Contabilidad, Administración de Empresas o Economía.
<b>Habilidades</b>	Toma de decisiones estratégicas, análisis financiero avanzado, interpretación de reportes, liderazgo y gestión de equipos.
<b>Actividades</b>	Supervisión de presupuestos, análisis de reportes financieros, aprobación de transacciones importantes, definición de estrategias financieras y coordinación con otros departamentos.

<b>Tipo de usuario</b>	Analista Financiero
<b>Formación</b>	Licenciatura en Finanzas, Contabilidad, Administración o afines.
<b>Habilidades</b>	Manejo de herramientas financieras, elaboración de informes, análisis de datos, conocimientos en normativas contables y capacidad de síntesis.
<b>Actividades</b>	Registro y seguimiento de transacciones, generación de reportes financieros, análisis de desviaciones presupuestarias y evaluación de tendencias económicas.

<b>Tipo de usuario</b>	Contador
<b>Formación</b>	Licenciatura en Contaduría Pública o Finanzas.
<b>Habilidades</b>	Gestión de cuentas, auditoría, conciliaciones bancarias, manejo de impuestos y normativas fiscales.
<b>Actividades</b>	Registro contable, validación de facturas, gestión de pagos y cobros, auditoría de transacciones y cumplimiento de regulaciones fiscales.

<b>Tipo de usuario</b>	Ejecutivo de Cuentas / Tesorería
<b>Formación</b>	Licenciatura en Finanzas, Administración o Contabilidad.
<b>Habilidades</b>	Manejo de flujo de caja, control de pagos y cobros, negociación con proveedores y clientes, y gestión de riesgos financieros.
<b>Actividades</b>	Programación de pagos, conciliación bancaria, seguimiento de cuentas por cobrar y por pagar, y gestión de relaciones con proveedores y clientes.

<b>Tipo de usuario</b>	Administrador del Sistema (TI/Soporte Técnico)
<b>Formación</b>	Ingeniería en Sistemas, Informática o afines.

<b>Habilidades</b>	Gestión de bases de datos, seguridad de la información, integración de sistemas, mantenimiento y soporte técnico.
<b>Actividades</b>	Administración de usuarios y permisos, monitoreo de seguridad, resolución de incidencias técnicas y actualización del sistema.

*Descripción de los usuarios del producto, incluyendo nivel educacional, experiencia y experiencia técnica. Se debe agregar una tabla por cada tipo de usuario.*

## 2.4 RESTRICCIONES

- **Metodologías de Desarrollo:** Se utilizará una metodología ágil como Scrum para garantizar iteraciones rápidas y retroalimentación constante.
- **Lenguajes de Programación:** El sistema estará desarrollado en **React Native** para la aplicación móvil y **Node.js** para el backend, con base de datos en **PostgreSQL**.
- **Normativas y Estándares:** Se seguirán buenas prácticas de seguridad en el desarrollo de software y normativas contables aplicables al sector financiero.
- **Restricciones de Hardware:** La aplicación debe poder ejecutarse en dispositivos móviles Android y iOS con al menos 2 GB de RAM y versiones actualizadas de sus sistemas operativos.
- **Sistema Operativo:** El backend será compatible con servidores Linux y Windows, pero se recomienda su implementación en entornos basados en Linux para mejor estabilidad y rendimiento.
- **Seguridad y Accesibilidad:** El sistema debe garantizar la encriptación de datos sensibles y autenticación de usuarios con roles y permisos específicos.

*Descripción de aquellas limitaciones a tener en cuenta a la hora de diseñar y desarrollar el sistema, tales como el empleo de determinadas metodologías de desarrollo, lenguajes de programación, normas particulares, restricciones de hardware, de sistema operativo etc.*

## 2.5 SUPOSICIONES Y DEPENDENCIAS

- **Disponibilidad del Sistema Operativo:** Se asume que Android e iOS continuarán brindando soporte y compatibilidad con React Native en futuras versiones.
- **Infraestructura en la Nube:** Se asume que los servicios en la nube utilizados (como AWS o Google Cloud) mantendrán su disponibilidad y costos dentro del rango previsto.
- **Conectividad a Internet:** Se espera que los usuarios dispongan de una conexión estable a Internet para sincronización de datos en tiempo real.
- **Regulaciones Financieras:** Cualquier cambio en las normativas de facturación y contabilidad podría requerir modificaciones en el sistema.

## 2.6 EVOLUCIÓN PREVISIBLE DEL SISTEMA

- **Automatización Avanzada:** Implementación de automatización en procesos de auditoría, generación de reportes automáticos y categorización inteligente de transacciones.
- **Soporte Multiplataforma Ampliado:** Expansión para incluir una versión web completa que permita la administración desde cualquier dispositivo con acceso a Internet.
- **Seguridad Avanzada:** Implementación de autenticación biométrica (huella dactilar y reconocimiento facial) y doble factor de autenticación para mejorar la protección de datos.
- **Gestión Multiempresa:** Capacidad para gestionar múltiples empresas dentro de una misma cuenta, permitiendo el control financiero de diversas organizaciones desde una única plataforma.
- **Compatibilidad con Normativas Internacionales:** Adaptación del sistema para cumplir con regulaciones fiscales y contables en diferentes países, facilitando su expansión global.
- **Interfaz Personalizable:** Desarrollo de un sistema de personalización de dashboards y reportes según las necesidades específicas de cada usuario o empresa.

- **Módulo de Inteligencia Financiera:** Creación de herramientas de análisis y proyección financiera con gráficos interactivos y recomendaciones basadas en datos históricos.
- **Soporte y Asistencia Automatizada:** Implementación de chatbots con IA para resolver dudas y ofrecer asistencia en tiempo real dentro de la plataforma.

### 3 REQUISITOS ESPECÍFICOS

#### REQUERIMIENTOS FUNCIONALES

Identificador del Requerimiento	RF-A01
Nombre del Requerimiento	Gestion de usuarios
Tipo <b>X</b>	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Descripción del requerimiento:	El sistema debe permitir el registro, autenticación y administración de usuarios, así como la definición de roles y permisos para controlar el acceso a las funcionalidades de la aplicación.
Características del Requerimiento	<ul style="list-style-type: none"> <li>•Registro de usuarios.</li> <li>•Autenticación de usuarios.</li> <li>•Asignación de roles y permisos.</li> </ul>
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional

#### REQUERIMIENTOS FUNCIONALES

Identificador del Requerimiento	RF-B01
Nombre del Requerimiento	Gestión de transacciones
Tipo <b>X</b>	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Descripción del requerimiento:	El sistema debe permitir registrar ingresos y egresos, clasificarlos por categorías, visualizar el historial de transacciones mediante reportes detallados y aplicar filtros avanzados por fecha, tipo, categoría y usuario.
Características del Requerimiento	<ul style="list-style-type: none"> <li>•Registro de transacciones (ingresos y egresos).</li> <li>•Clasificación por categorías.</li> <li>•Generación de reportes detallados.</li> <li>•Filtros avanzados.</li> </ul>

Prioridad del requisito	<input type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional
-------------------------	--	--	--

REQUERIMIENTOS FUNCIONALES	
Identificador del Requerimiento	RF-B02
Nombre del Requerimiento	Control de presupuestos
Tipo <b>X</b>	<input type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Descripción del requerimiento:	El sistema debe permitir crear y administrar presupuestos para períodos específicos, realizar seguimiento del gasto, comparar contra los límites establecidos y generar alertas si el presupuesto se excede o se aproxima al límite.
Características del Requerimiento	•Creación y administración de presupuestos. •Seguimiento de gastos. •Alertas de exceso o aproximación al límite.
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional

REQUERIMIENTOS FUNCIONALES	
Identificador del Requerimiento	RF-C01
Nombre del Requerimiento	Clasificación por categorías
Tipo <b>X</b>	<input type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Descripción del requerimiento:	El sistema debe organizar transacciones en categorías de ingresos y egresos, y proporcionar estadísticas por categoría para identificar tendencias de gasto o ingreso.
Características del Requerimiento	•Organización de transacciones por categorías. •Generación de estadísticas por categoría.
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional

REQUERIMIENTOS FUNCIONALES	
Identificador del Requerimiento	RF-B03
Nombre del Requerimiento	Manejo de pagos recurrentes

<b>Tipo</b> <b>X</b>	<input type="checkbox"/> <b>Requisito</b> <input type="checkbox"/> <b>Restricción</b>
<b>Descripción del requerimiento:</b>	El sistema debe permitir configurar pagos automáticos con fechas de inicio y fin, asignar categorías y frecuencias (diaria, semanal, mensual, anual), y generar notificaciones para pagos próximos o vencidos.
<b>Características del Requerimiento</b>	•Configuración de pagos recurrentes. •Asignación de categorías y frecuencias. •Notificaciones de pagos próximos o vencidos.
<b>Prioridad del requisito</b>	<input type="checkbox"/> <b>Alta/Esencial</b> <input type="checkbox"/> <b>Media/Deseado</b> <input type="checkbox"/> <b>Baja/Opcional</b>

### REQUERIMIENTOS FUNCIONALES

<b>Identificador del Requerimiento</b>	RF-B04
<b>Nombre del Requerimiento</b>	Gestión de Facturas y Recibidos
<b>Tipo</b> <b>X</b>	<input type="checkbox"/> <b>Requisito</b> <input type="checkbox"/> <b>Restricción</b>
<b>Descripción del requerimiento:</b>	El sistema debe permitir crear, almacenar y rastrear facturas y recibos emitidos o recibidos, asociarlos a clientes, proveedores o transacciones específicas, y mostrar su estado ("Pendiente", "Pagada" o "Vencida").
<b>Características del Requerimiento</b>	•Creación y almacenamiento de facturas y recibos. •Asociación a clientes, proveedores o transacciones. •Visualización del estado de las facturas.
<b>Prioridad del requisito</b>	<input type="checkbox"/> <b>Alta/Esencial</b> <input type="checkbox"/> <b>Media/Deseado</b> <input type="checkbox"/> <b>Baja/Opcional</b>

### REQUERIMIENTOS FUNCIONALES

<b>Identificador del Requerimiento</b>	RF-C02
<b>Nombre del Requerimiento</b>	Relación con Clientes y Proveedores
<b>Tipo</b> <b>X</b>	<input type="checkbox"/> <b>Requisito</b> <input type="checkbox"/> <b>Restricción</b>
<b>Descripción del requerimiento:</b>	El sistema debe permitir registrar información detallada de clientes y proveedores, incluyendo contactos, historial de transacciones y facturas, y facilitar la generación de reportes para analizar relaciones comerciales.
<b>Características del Requerimiento</b>	•Registro de información de clientes y proveedores. •Generación de reportes de relaciones comerciales.
<b>Prioridad del requisito</b>	<input type="checkbox"/> <b>Alta/Esencial</b> <input type="checkbox"/> <b>Media/Deseado</b> <input type="checkbox"/> <b>Baja/Opcional</b>

REQUERIMIENTOS FUNCIONALES	
Identificador del Requerimiento	RF-C03
Nombre del Requerimiento	Reportes y Análisis Financieros
Tipo <b>X</b>	<input type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Descripción del requerimiento:	El sistema debe ofrecer reportes gráficos interactivos sobre ingresos, egresos, presupuestos y pagos recurrentes, presentar resúmenes financieros diarios, mensuales y anuales, y ayudar a identificar patrones de comportamiento financiero y oportunidades de optimización.
Características del Requerimiento	<ul style="list-style-type: none"> <li>• Generación de reportes gráficos interactivos.</li> <li>• Resúmenes financieros diarios, mensuales y anuales.</li> <li>• Identificación de patrones y oportunidades de optimización.</li> </ul>
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional

*y realizar la descripción del requisito*

*Definición de acciones fundamentales que debe realizar el software al recibir información, procesarla y producir resultados.*

*En ellas se incluye:*

- *Comprobación de validez de las entradas*
- *Secuencia exacta de operaciones*
- *Respuesta a situaciones anormales (desbordamientos, comunicaciones, recuperación de errores)*
- *Parámetros*
- *Generación de salidas*
- *Relaciones entre entradas y salidas (secuencias de entradas y salidas, formulas para la conversión de información)*
- *Especificación de los requisitos lógicos para la información que será almacenada en base de datos (tipo de información, requerido)*

*La distribución de los párrafos que forman este punto puede diferir del propuesto en esta plantilla, si las características del sistema aconsejan otra distribución para ofrecer mayor claridad en la exposición.*



**3.1****REQUISITOS COMUNES DE LAS INTERFACES**

La aplicación contará con una interfaz intuitiva y adaptable para dispositivos móviles con sistema operativo Android e iOS. Las principales características incluyen:

**3.1.1 INTERFACES DE USUARIO**

- Diseño basado en Material Design para una experiencia de usuario coherente y fluida.
- Esquema de colores con una paleta base en tonos azul y blanco para transmitir profesionalismo y confianza.
- Uso de iconografía y elementos gráficos que faciliten la navegación y comprensión de datos financieros.
- Dashboard principal con acceso rápido a reportes financieros, ingresos, egresos y presupuestos.
- Secciones organizadas por pestañas para gestión de transacciones, clientes, proveedores y reportes.
- Notificaciones emergentes para alertas de presupuesto, vencimiento de facturas y pagos recurrentes.
- Accesibilidad optimizada con opciones de texto en alto contraste y soporte para lectores de pantalla.

**3.1.2 INTERFACES DE HARDWARE**

- Compatible con dispositivos móviles Android (versión 9.0 o superior) y iOS (versión 14.0 o superior).
- Requisitos mínimos: 2GB de RAM y 200MB de almacenamiento disponible.
- Uso de la cámara del dispositivo para escaneo de facturas y recibos.
- Soporte para autenticación biométrica (huella dactilar o reconocimiento facial) en dispositivos compatibles.

**3.1.3 INTERFACES DE SOFTWARE**

- Integración con sistemas contables externos a través de API REST.
- Interfaz con pasarelas de pago para gestionar transacciones en línea.
- Definición del formato de intercambio de datos: JSON para comunicación entre frontend y backend.
- Integración con PostgreSQL como base de datos para almacenamiento seguro de la información.

### 3.1.4 INTERFACES DE COMUNICACIÓN

- Comunicación entre el frontend (React Native) y el backend (Node.js [express] ) a través de API REST.
- Cifrado de datos mediante HTTPS y uso de JWT para autenticación de usuarios.
- Sincronización en tiempo real con WebSockets para actualizaciones inmediatas de transacciones y estados de facturas. ( **pendiente** )
- Protocolos de seguridad basados en OAuth 2.0 para integración con servicios de terceros.

### 3.2.1

#### REQUISITOS DE RENDIMIENTO

- Capacidad para soportar una alta gama de usuarios concurrentes y sus transacciones.
- Latencia máxima de respuesta de API: +500 ms en operaciones estándar.

### 3.2.2 REQUISITOS DE SEGURIDAD

- Control de acceso basado en roles (RBAC) para restringir permisos.
- Registro de logs de actividad con auditoría de accesos y modificaciones.
- Verificación de integridad en archivos de facturas y recibos.

### 3.3.3 REQUISITOS DE FIABILIDAD

- Tiempo entre fallos mínimo de 6 meses
- Soporte de recuperación y respaldo en base de datos sobre su información.

### 3.3.4 REQUISITOS DE DISPONIBILIDAD

- Disponibilidad del sistema del 99.5% anual.
- Implementación de balanceo de carga para mejorar estabilidad.

### 3.3.5 REQUISITOS DE MANTENIBILIDAD

- Código modular y documentado para facilitar actualizaciones y correcciones.
- Mantenimiento correctivo y evolutivo cada 6 meses.
- Generación de reportes automáticos de uso y estadísticas de acceso mensuales y trimestrales.

### 3.3.6 PORTABILIDAD

- Código fuente estructurado para facilitar su despliegue en otros entornos móviles y web.
- Uso de tecnologías multiplataforma como React Native para minimizar dependencias específicas del sistema operativo.
- Compatible con servidores Linux y Windows para el backend.

## 3.4 OTROS REQUISITOS

- Cumplimiento con normativas contables aplicables en el sector financiero.
- Adaptación a normativas de protección de datos (GDPR y Ley de Protección de Datos Personales).

## 4. METODOLOGÍA

### 4.1 Enfoque Ágil

El desarrollo del sistema se basará en metodologías ágiles, permitiendo iteraciones cortas (sprints) y revisiones periódicas. Este enfoque favorece la adaptación a cambios y la incorporación temprana del feedback del usuario. Se llevarán a cabo reuniones diarias (daily stand-ups) para coordinar el progreso del equipo, así como revisiones al final de cada sprint para evaluar el cumplimiento de los objetivos y planificar mejoras o nuevas funcionalidades.

### 4.2 Integración Continua y Entrega Continua (CI/CD)

El proceso de desarrollo incorporará prácticas de integración y entrega continua para garantizar la calidad y estabilidad del software en cada iteración. Esto incluye la ejecución automatizada de pruebas unitarias y de integración, así como despliegues incrementales en entornos de prueba y producción. El objetivo es minimizar errores y permitir una rápida respuesta ante cambios o incidencias.

## 5. SELECCIÓN DE TECNOLOGÍAS

### 5.1 Tecnologías Frontend

- **Aplicación Móvil:** Se utilizará **React Native** para el desarrollo de la aplicación móvil, permitiendo compartir gran parte del código entre plataformas (iOS y Android) y asegurar una experiencia de usuario fluida.
- **Aplicación Web Progresiva (PWA):** Se desarrollará con **Next.js**, lo que aportará ventajas como el renderizado del lado del servidor (SSR), generación estática (SSG) y optimización automática, garantizando una interfaz responsiva y un rendimiento excelente en navegadores modernos.

## 5.2 Tecnologías Backend y de Integración

- **Backend:** Se desarrollará en **Node.js** utilizando el framework **Express**, proporcionando una API REST que gestione la lógica de negocio y la interacción con la base de datos.
- **Base de Datos:** Se empleará **PostgreSQL** por su robustez y capacidad para manejar transacciones complejas, asegurando la integridad y seguridad de la información.
- **Integraciones Externas:** Se prevé el uso de servicios externos para la autenticación (OAuth 2.0, autenticación multifactor), pasarelas de pago (Stripe o PayPal) y servicios de facturación electrónica, facilitando la interoperabilidad con otros sistemas.

## 6. ARQUITECTURA

### 6.1 Arquitectura de Software

El sistema se estructurará en una arquitectura por capas, adoptando el patrón **MVC (Modelo - Vista - Controlador)** para organizar de forma clara y separada la lógica del negocio, la interfaz de usuario y el flujo de datos. Esta estructura se desglosa de la siguiente manera:

- **Modelo (Model):**  
Contendrá la lógica de negocio y el acceso a datos. Aquí se definirán las entidades principales, como usuarios, transacciones, presupuestos, categorías, pagos recurrentes, facturas, proveedores y clientes, y se implementarán las operaciones CRUD sobre la base de datos PostgreSQL. Este componente se encargará de validar la información y garantizar la integridad de los datos.
- **Vista (View):**  
Representa la interfaz de usuario, implementada tanto en la aplicación móvil (usando React Native) como en la aplicación web progresiva. En la PWA, se utilizará **Next.js**, lo que permite aprovechar el renderizado del lado del servidor para mejorar la carga inicial, la indexación SEO y la optimización

del rendimiento. La Vista se encargará de presentar los datos de manera intuitiva y responsiva, permitiendo a los usuarios interactuar con el sistema mediante menús, formularios y notificaciones en tiempo real.

- **Controlador (Controller):**

Actúa como intermediario entre el Modelo y la Vista. Los Controladores, implementados en el backend con Node.js y Express, recibirán las peticiones provenientes de la interfaz de usuario, invocarán la lógica del Modelo para procesar dichas peticiones y devolverán las respuestas adecuadas a la Vista. Además, se encargará de gestionar aspectos como la autenticación, el enrutamiento y la coordinación de servicios.

Esta arquitectura MVC permite:

- Una **separación clara de responsabilidades**, facilitando el mantenimiento y la evolución del sistema.
- **Escalabilidad**, ya que se pueden agregar nuevos módulos o funcionalidades sin afectar la integridad de otros componentes.
- **Colaboración eficiente**, permitiendo que equipos diferentes trabajen simultáneamente en la interfaz, la lógica del negocio y la integración con la base de datos.
- **Reutilización y modularidad** de componentes, lo que reduce redundancias y acelera el desarrollo de nuevas funcionalidades.

## 6.2 Arquitectura de Despliegue


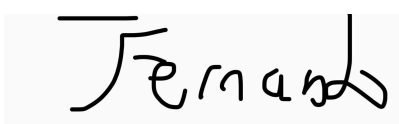



- **Contenedores y Orquestación:**

Se desplegará el sistema en contenedores Docker, facilitando la portabilidad y la consistencia entre los entornos de desarrollo, prueba y producción.

- **Infraestructura en la Nube:**

Se utilizarán servicios en la nube para alojar el backend, la base de datos y las integraciones, garantizando alta disponibilidad, escalabilidad dinámica y un rápido tiempo de recuperación ante fallos.

**FIRMAS**

<b>INTEGRANTES</b>	
<b>Nombre</b>	<b>Firma</b>
<b>Cárdenas Zalapa Jesús Antonio</b>	
<b>Medina Palacios Fernando</b>	
<b>Rodríguez Rosales Jesús Antonio</b>	
<b>Ramirez Aispuro Juan Jose</b>	
<b>Arguelles Galvez Antonio</b>	

<b>MAESTROS</b>	
<b>Nombre</b>	<b>Firma</b>
<b>Ray Brunett Parra Galaviz</b>	
<b>Daniel Enrique Torres Aldana</b>	