

Requerimientos Funcionales (Historias de Usuario)

ID	QUIERO	PARA	CRITERIOS DE ACEPTACIÓN
HU01	Visualizar mi saldo	Saber cuanto tengo en mi cartera digital	<ul style="list-style-type: none"> • Debe mostrar el código de la cartera, nombre de la cartera y el saldo actual.
HU02	Iniciar en mi sesión personal	Tener privacidad al momento de realizar mis actividades financieras	<ul style="list-style-type: none"> • El “usuario” y “contraseña” deben ser únicos para cada cliente.
HU03	Registrarme	Llevar control de mis finanzas	<ul style="list-style-type: none"> • El sistema debe proporcionar un formulario de registro con la siguiente información del cliente (Nombre, Apellido, Cedula, Telefono, Correo Electronico, Dirección). • El sistema debe validar la información ingresada por el usuario para garantizar que se cumplan los requisitos de formato y longitud para cada campo. • Se deben mostrar mensajes de error claros y descriptivos si se ingresan datos incorrectos o faltantes. • El sistema debe generar un identificador único (un número de cliente) para cada cliente registrado.
HU04	Añadir un registro en mis transacciones	Tener un control de mi flujo financiero	<ul style="list-style-type: none"> • El sistema debe proporcionar una interfaz que permita al usuario agregar una nueva transacción a su registro de transacciones. • Debe haber un formulario o una pantalla dedicada para ingresar los detalles de la transacción (cantidad, la fecha, la categoría, una descripción opcional). • El sistema debe permitir al usuario seleccionar el tipo de transacción que desea agregar (categoría) (por ejemplo, ingreso, gasto, transferencia, etc.).

			<ul style="list-style-type: none"> • Debe haber opciones predefinidas para los diferentes tipos de transacciones, y el usuario también debe tener la opción de agregar un tipo personalizado si es necesario.
HU05	Generar un reporte de mis transacciones	Poder realizar un análisis de mis ingresos, egresos, entre otros.	<ul style="list-style-type: none"> • El formato del reporte debe incluir columnas claras que muestren los detalles de las transacciones, como la fecha, el tipo de transacción, la cantidad, la descripción, etc. • El usuario debe poder elegir la forma en que se agrupan y ordenan los datos en el reporte, por ejemplo, por fecha, por tipo de transacción, por categoría, etc. • El sistema debe mostrar los datos agrupados y ordenados de manera coherente y fácil de entender en el reporte generado.
HU06	Caracterizar por categoría mi reporte de transacciones	Poder realizar un análisis específico y más detallado de cada categoría	<ul style="list-style-type: none"> • El sistema debe permitir al usuario seleccionar las categorías de transacciones que desea incluir en el reporte. • Deben estar disponibles opciones predefinidas para las categorías más comunes, así como la opción de categorías personalizadas si es necesario. • El sistema debe agrupar las transacciones en el reporte según las categorías seleccionadas por el usuario. • Cada grupo debe mostrar el total de transacciones y el total de montos asociados a esa categoría.

1. **Registro de Usuario:** Permitir que los usuarios se registren en la aplicación proporcionando información básica como nombre, dirección de correo electrónico y contraseña.
2. **Inicio de sesión:** Permitir a los usuarios iniciar sesión en la aplicación con sus credenciales previamente registradas.

3. **Registro de Transacciones:** Permite a los usuarios registrar transacciones financieras, como ingresos, gastos, transferencias entre cuentas, pagos de facturas, etc.
4. **Categorización de Transacciones:** Permitir a los usuarios categorizar las transacciones registradas para un mejor seguimiento y análisis de los gastos y los ingresos.
5. **Visualización de Saldo:** Mostrar a los usuarios el saldo actual de sus cuentas financieras y su saldo disponible después de tener en cuenta las transacciones registradas.
6. **Generación de Reportes:** Permitir a los usuarios generar reportes detallados sobre sus transacciones financieras, incluyendo resúmenes de ingresos y gastos por período de tiempo, gráficos de tendencias, etc.

Requerimientos No Funcionales

1. **Usabilidad:** La aplicación debe ser intuitiva y fácil de usar, con una interfaz de usuario clara y amigable.
 - a. Métrica: Porcentaje de usuarios que completan el proceso de registro/inicio de sesión correctamente en el primer intento.
 - b. Meta: Más del 90% de los usuarios deben completar el proceso de registro/inicio de sesión correctamente en el primer intento.
2. **Rendimiento:** La aplicación debe ser rápida y eficiente, con tiempos de carga mínimos y capacidad para manejar múltiples usuarios simultáneamente.
 - a. Métrica: Tiempo medio de carga de la página principal de la aplicación.
 - b. Meta: Menos de 3 segundos de tiempo medio de carga de la página principal.
3. **Disponibilidad:** La aplicación debe estar disponible las 24 horas del día, los 7 días de la semana, con un tiempo de inactividad mínimo programado para mantenimiento.
 - a. Métrica: Tiempo de inactividad planificado y no planificado durante un período de tiempo determinado (por ejemplo, un mes).
 - b. Meta: Menos de 1 hora de tiempo de inactividad planificado y no planificado por mes.
4. **Escalabilidad:** La aplicación debe ser capaz de escalar horizontalmente para manejar un aumento en el número de usuarios y transacciones sin degradación del rendimiento.
 - a. Métrica: Tiempo de respuesta del servidor bajo diferentes cargas de usuarios (por ejemplo, 100, 500, 1000 usuarios simultáneos).
 - b. Meta: El tiempo de respuesta del servidor debe mantenerse por debajo de 500 ms para todas las cargas de usuarios simuladas.
5. **Seguridad:** La aplicación debe proteger los datos financieros de los usuarios y garantizar su confidencialidad, integridad y disponibilidad.
 - a. Métrica: Número de intentos de acceso no autorizados bloqueados por el sistema de seguridad.
 - b. Meta: Todos los intentos de acceso no autorizados deben ser bloqueados y registrados, con una tasa de detección del 100%.
6. **Compatibilidad:** La aplicación debe ser compatible con una variedad de dispositivos y navegadores web para garantizar una experiencia consistente para todos los usuarios.
 - a. Métrica: Porcentaje de usuarios que acceden a la aplicación desde diferentes dispositivos y navegadores.
 - b. Meta: La aplicación debe ser compatible con al menos el 95% de los dispositivos y navegadores más utilizados según las estadísticas de uso.
7. **Cumplimiento Normativo:** La aplicación debe cumplir con las regulaciones financieras y de privacidad pertinentes, como GDPR, PCI-DSS, etc.

- a. Métrica: Resultados de las auditorías de cumplimiento normativo (por ejemplo, auditorías de seguridad, cumplimiento de GDPR, cumplimiento de PCI-DSS, etc.).
 - b. Meta: La aplicación debe pasar todas las auditorías de cumplimiento normativo sin hallazgos significativos.
- 8. **Respuesta ante errores:** La aplicación debe manejar errores de manera adecuada, proporcionando mensajes de error claros y procesando transacciones de manera segura incluso en caso de fallo del sistema.
 - a. Métrica: Tiempo medio de respuesta del sistema ante errores críticos (por ejemplo, errores de servidor, errores de base de datos).
 - b. Meta: El tiempo medio de respuesta del sistema ante errores críticos debe ser inferior a 1 segundo.

ARQUITECTURA

Se tomó la decisión de usar los microservicios como la arquitectura a implementar, por lo siguientes motivos:

- **Escalabilidad:** Los microservicios permiten escalar partes específicas de la aplicación de manera independiente, lo que puede ser beneficioso en una aplicación financiera donde ciertas funcionalidades pueden experimentar picos de carga mientras que otras no.
- **Despliegue ágil:** Al dividir la aplicación en microservicios más pequeños y manejables, los equipos pueden trabajar de manera más autónoma y rápida en el desarrollo, pruebas y despliegue de nuevas funcionalidades.
- **Resiliencia:** Los microservicios pueden aislar fallos, lo que significa que un fallo en un microservicio no afectará a otros, lo que mejora la resiliencia del sistema en su conjunto.
- **Flexibilidad tecnológica:** Cada microservicio puede estar desarrollado y desplegado utilizando la tecnología más adecuada para su propósito específico, lo que permite una mayor flexibilidad y adaptabilidad tecnológica.

La arquitectura en el modelo c4 se encuentra en el documento adjunto.

PLAN DE PRUEBAS

GENERAL

FinanceMe sirve como una solución a la gestión de las finanzas personales que no están incluidas en productos financieros como cuentas de ahorros en bancos u otras entidades financieras. Sirve para tener una mejor comprensión del flujo económico (se incluyen los ingresos y egresos) y darle el poder a cada usuario de analizar cómo se mueve su capital y gestionar mejor sus recursos.

LÍMITES (Es dinámico, nunca es fijo. Aprendizaje continuo)

En esta sección se describen qué actividades se llevarán a cabo en el proyecto.

ALCANCE

Cubierto	Descripción	No cubierto	
Pruebas funcionales	<ul style="list-style-type: none">● Módulos: Registro de usuario, inicio de sesión, registro de transacciones, categorización de transacciones, visualización de saldo y generación de reportes.● Técnicas: Pruebas de unidad, integración y sistema.● Prioridad: Alta.		
Pruebas no funcionales	<ul style="list-style-type: none">● Módulos: Todos los módulos de la aplicación.● Técnicas: Pruebas de carga y rendimiento.● Prioridad: Alta.		
Pruebas de usabilidad	<ul style="list-style-type: none">● Módulos: Todos los módulos de la aplicación.● Técnicas: Recolección de comentarios de usuarios y evaluación de la satisfacción de estos.		
Criterios de Entrada y Salida	Se establece cuando todas las pruebas funcionales y no funcionales pasan satisfactoriamente.		
Calidad de datos	Garantizar que los datos de prueba reflejan escenarios reales.		

Pruebas unitarias	Se realizarán pruebas unitarias en cada microservicio.		
		Instalación y migración	No se evaluará la instalación y migración a otro entorno de desarrollo
		Pruebas de caja blanca	No se realizarán evaluaciones detalladas de la estructura interna. Las pruebas se centran en la funcionalidad general, la usabilidad, el rendimiento y otros aspectos externos de la aplicación.

Sí no se hacen alguno de los cubiertos, estos automáticamente se considerarán como no cubiertos y se entenderá que se contempló en un principio pero se decidió que no se haría dicha prueba.

CONFIGURACIÓN DEL EQUIPO

- Flujo de trabajo:
 1. Product Owner entrega la documentación como los requerimientos, datos de prueba, etc.
 2. El equipo de Devs proporciona construcción y entregables de software.
 3. Los DevOps brindan soporte al ambiente de pruebas ejecutando los casos de prueba y si se encuentra una falla se le notifica al probador para que la analice.
 4. Por último, las actividades del equipo de pruebas son; crear el plan de pruebas, crear los casos de prueba, actualizar los casos de prueba si se hacen cambios en los requerimientos.

Después de acabar el flujo, el equipo de pruebas le entrega al Product Owner un reporte de los casos de prueba, reporte de defectos y fallos en el software al equipo de Devs y un reporte detallado sobre los resultados de pruebas a los DevOps que permitan hacer ajustes en el ambiente de pruebas.

RIESGOS DE CALIDAD

Riesgos informados por el PO o el equipo de Devs ya identificados, por lo que el equipo de pruebas deberá tener esto en cuenta a la hora de desarrollar su trabajo para mitigar estos riesgos. Estos riesgos se califican de 1 a 5, siendo 5 un riesgo muy alto y 1 un riesgo muy bajo.

En este caso, sería:

Riesgo	Mitigación
Fallas en la integración de microservicios. Prioridad de Riesgo: 5	Realizar pruebas exhaustivas de integración.
Inconsistencia en la categorización de transacciones por interpretaciones diferentes del usuario. Prioridad de Riesgo: 3	Ofrecer una interfaz clara y ejemplos para ayudar a los usuarios a comprender y seguir un enfoque coherente en la categorización.
Desviación en los tiempos de carga de la página principal que superen los límites establecidos durante picos de usuarios o debido a la carga de transacciones al mismo tiempo. Prioridad de Riesgo: 5	Realizar pruebas exhaustivas de carga y rendimiento para identificar y abordar posibles cuellos de botella.

CALENDARIO DE HITOS

Se proponen momentos para hablar sobre las actividades en el equipo de pruebas. El Product Owner o líder del equipo de Devs son los que planifican este calendario y se documentan las reuniones desde el momento que se conceptualiza el producto hasta que está listo para los usuarios finales. Reuniones con fecha de inicio y fecha de fin de algún refuerzo y/o evaluación continua de un proceso.

TRANSICIONES

Qué criterios se van a tomar en cuenta para pasar de un nivel de prueba a otro.

CRITERIOS DE ENTRADA

- Todas las pruebas funcionales y no funcionales deben haber sido pasadas.
- El equipo de DevOps debe haber confirmado que el entregable está funcional en el ambiente de pruebas.
- Se debe realizar una evaluación exhaustiva de la cobertura de pruebas, asegurándose de que se hayan realizado pruebas suficientes en todos los aspectos de la aplicación.
- Antes de avanzar a la siguiente fase, se requiere la aprobación formal por parte del equipo de Devs indicando que han abordado satisfactoriamente todas las cuestiones y defectos identificados durante las pruebas.

CRITERIOS DE SALIDA

- Informe de los casos de prueba, reporte de defectos y resultados detallados entregados a los respectivos responsables por parte del equipo de pruebas.
- No hay fallas, paradas o procesos culminados de forma inesperada (superior al 5% de las pruebas).
- El Product Owner aprueba que el producto definido por las últimas pruebas de sistema satisface razonablemente las expectativas del usuario.
- El equipo de pruebas ha ejecutado las pruebas planeadas para cada ambiente.

CONFIGURACIÓN DEL AMBIENTE DE PRUEBAS

El sistema operará en varios navegadores de internet y combinaciones de sistemas operativos. Se deja constancia de que los responsables de la configuración de estos entornos serán los del equipo de DevOps pero la configuración del entorno permanece bajo el control del jefe del equipo de pruebas.

Internet Browser	OS
Chrome	Windows 10
Brave	Windows 10
Opera	Windows 11
Chrome	Windows 11
Brave	Windows 11