**Informe Proyecto**

**Capstone**

**Sección-003D**

**App de Finanzas Personales o Pymes**

***Integrantes:***

*Cristopher Arredondo*

*Angel Cea*

**Página de Actualizaciones del Proyecto**

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre del Integrante | Fecha | Actualización / Avance | Observaciones |
| Angel Cea | 01-09-2025 |  |  |
| Cristopher Arredondo | 02-09-2025 |  |  |

Notas:

• Nombre del Integrante: Persona que realiza la actualización.

• Fecha: Día en que se registra el avance o modificación.

• Actualización / Avance: Descripción breve y precisa del trabajo realizado.

• Observaciones: Comentarios adicionales, dificultades, próximos pasos o retroalimentación.

Contenido

[**Problemática** 5](#_heading=h.871mf4mo0ir)

[**Contexto** 5](#_heading=h.l1y17bkqxprb)

[**Problemas concretos (personas naturales y pymes)** 6](#_heading=h.4chwvjfrpoiv)

[**Solución** 8](#_heading=h.x5rvw43j1ee1)

[**Propuesta general** 8](#_heading=h.mnhw8bfjr3mi)

[**Características principales** 8](#_heading=h.6zsbrnfu1h1g)

[**Beneficios frente a soluciones actuales** 9](#_heading=h.xers3oi7l06p)

[**Valor agregado** 10](#_heading=h.qulp0tiarjmo)

[**Beneficiarios** 11](#_heading=h.n8sczxd6j7yq)

[**Personas naturales** 11](#_heading=h.fulwyk4xk9tt)

[**Microempresas (pymes)** 12](#_heading=h.lxaidetra6kt)

[**El mercado** 13](#_heading=h.j7f51fgqdqba)

[**Alcance** 14](#_heading=h.f6u9kvj27zil)

[**Alcance general** 14](#_heading=h.iv5mjhwbulx7)

[**Funcionalidades incluidas en la primera versión** 14](#_heading=h.wuv1yglnk7n2)

[**Alcance para usuarios** 15](#_heading=h.2197f6x2mbko)

[**Metodología** 16](#_heading=h.d37bqgdfsxd4)

[**Enfoque general** 16](#_heading=h.q3nthnwqc4yr)

[**Etapas del desarrollo** 16](#_heading=h.qheli6r3a8en)

[**Requerimientos** 18](#_heading=h.t65q8ndkfzos)

[**No Funcionales** 18](#_heading=h.lf186zfgfftu)

[**Funcionales** 20](#_heading=h.m8vcfaec5x72)

[**Tecnologías** 22](#_heading=h.ojcj7ubm2o5)

[**Cronograma** 23](#_heading=h.bohimwsj59du)

[**Costos** 24](#_heading=h.umpbum4zaaay)

[**Riesgos** 25](#_heading=h.duv8r9vknf35)

[**Riesgos técnicos** 25](#_heading=h.9mo3g96cur2o)

[**Riesgos de gestión del proyecto** 26](#_heading=h.2h43d9j3wfrx)

[**Riesgos de calidad** 27](#_heading=h.d9h6thi7g84w)

[**Pruebas** 28](#_heading=h.mteyodw0feee)

[**Alcance de pruebas** 28](#_heading=h.m775exnkt4o0)

[**Datos de prueba (base mínima)** 28](#_heading=h.s7vqgzmqwmca)

[**Matriz de casos de prueba funcionales** 29](#_heading=h.rtw5i0m417q8)

[**Pruebas negativas y de validación** 32](#_heading=h.bpe17922gsyi)

[**Pruebas de rendimiento percibido (en la app)** 33](#_heading=h.geoo19xs44ts)

[**Pruebas de usabilidad básica (dentro de la app)** 33](#_heading=h.qwkoefrowfzf)

[**Lista de verificación UAT (aceptación de usuario)** 33](#_heading=h.egquflrhwxi8)

# **Problemática**

## **Contexto**

Las finanzas, tanto personales como de microempresas (pymes), continúan siendo dispersas y propensas a errores.  
En el caso de las **personas naturales**, la mayoría lleva un control limitado a través de resúmenes bancarios o de tarjetas de crédito, que solo muestran consumos retrospectivos sin clasificaciones detalladas. Esto dificulta identificar los llamados “gastos hormiga”, proyectar ahorros o diseñar planes de inversión. Las aplicaciones de los bancos ofrecen gráficos básicos, pero suelen estar restringidas al ecosistema de la propia entidad, sin integrar otras fuentes como billeteras digitales, compras compartidas o deudas familiares. En la práctica, muchas personas terminan organizándose en planillas manuales o simplemente no tienen una visión clara de su situación financiera real.

En el caso de las **pymes**, aunque la mayoría de las boletas o facturas ya son electrónicas y la mayoría de las ventas se procesan con tarjetas o transferencias digitales, la información financiera queda distribuida en múltiples plataformas: portales bancarios, Transbank, SII, Mercado Pago, registros internos, entre otros. Muchos emprendedores dependen de las cartolas bancarias o de reportes parciales, los cuales muestran solo movimientos crudos (abonos/cargos), a veces con retraso, sin clasificaciones ni proyecciones de flujo. Revisar varias fuentes en paralelo es lento, confuso y no entrega una visión consolidada.

En ambos casos —personas naturales y pymes— se observa que las herramientas actuales no consideran aspectos clave como **deudas compartidas, cuotas, objetivos de ahorro ni planes de inversión accesibles**.

En contraste, la solución propuesta consiste en una **app móvil integrada**, independiente de bancos y con visión multi-fuente, capaz de consolidar ingresos, gastos, deudas e inversiones en un solo lugar. Además, ofrece clasificación flexible, alertas de vencimientos, división de deudas compartidas, tanto para el usuario individual que busca organizar su economía diaria como para el emprendedor que necesita mayor control de su negocio.

## **Problemas concretos (personas naturales y pymes)**

Barreras de entrada a Excel y planillas

* No todas las personas dominan Excel; estructurar una planilla exige tiempo y conocimientos.
* Cada paso adicional eleva la probabilidad de omisión o registro tardío.
* Resultado: baja adherencia, datos incompletos y decisiones a ciegas.

Fragmentación y duplicidad de información

* Registros dispersos entre bancos, SII, Transbank, Mercado Pago, softwares de facturación y registros internos.
* Dificultad para consolidar ingresos/gastos por categoría, mes, proyecto o persona.
* Falta de visión fortalecida y duplicación de movimientos.

Deudas y gastos compartidos sin trazabilidad

* Dividir cuentas o compras grandes aún requiere cálculos manuales o acuerdos informales.
* No existen mecanismos sincronizados entre participantes, lo que genera disputas.

Costos ocultos y riesgo operativo

* Horas perdidas revisando plataformas y cartolas.
* Errores en cálculos manuales que derivan en sobreestimación o subestimación de recursos.
* En pymes, la falta de estandarización compromete la caja y la relación con proveedores.

Brecha de mercado

Las soluciones actuales (cartolas, resúmenes bancarios y apps interactivas de entidades financieras) presentan limitaciones críticas:

* Están restringidas a un banco o institución.
* No integran flujos completos de deudas, cuotas ni inversiones.
* No entregan recomendaciones accionables.

El proyecto propuesto llena este vacío con una app móvil independiente de bancos, multi-fuente, simple y prescriptiva. Su valor diferencial es transformar datos dispersos en decisiones de gestión financiera que aumentan control, liquidez y capacidad de planificación, tanto en pymes como en personas naturales.

# **Solución**

## **Propuesta general**

Nuestra solución consiste en el desarrollo de una aplicación móvil de finanzas, creada en base a **Angular e Ionic**, que funciona de manera independiente a los bancos y se apoya en **Firebase/Firestore** para el almacenamiento seguro de datos. El diseño está orientado tanto a **personas naturales como a microempresas**, que necesitan fortalecer, organizar y proyectar sus finanzas de manera sencilla, sin depender de planillas complejas ni de información fragmentada proveniente de cartolas bancarias.

A diferencia de estas cartolas o de las apps interactivas que ofrecen algunos bancos, nuestra aplicación no se limita a mostrar movimientos pasados de forma aislada. Por el contrario, ofrece una **visión multi-fuente y personalizada**, donde el usuario puede registrar y clasificar sus ingresos, gastos, deudas, inversiones y objetivos financieros, obteniendo reportes claros y proyecciones útiles para la toma de decisiones.

Como elementos diferenciales, la app incorpora la **división de gastos y deudas compartidas** —ideal para salidas sociales o compras en conjunto—, así como un sistema de **objetivos financieros** que permite establecer metas y medir el progreso hacia ellas.

## **Características principales**

Unificación de información

* Centraliza en un solo lugar los datos financieros que hoy están dispersos entre bancos, Transbank, SII, Mercado Pago y registros manuales.
* Permite al usuario ingresar manualmente sus operaciones y ver el impacto inmediato en su flujo de caja y en sus proyecciones.
* Evita la necesidad de revisar varias cartolas en paralelo.

Clasificación flexible

* El usuario puede etiquetar movimientos según categorías personalizadas (ventas, gastos fijos, inversión, deudas, ocio, etc.).
* Supera las categorizaciones genéricas de los bancos, que suelen ser limitadas e inexactas.

Deudas y gastos compartidos

* Incluye un módulo para dividir deudas o gastos compartidos (por ejemplo, una compra grande pagada a cuotas o una cena en grupo).
* Calcula automáticamente la parte correspondiente de cada persona, evitando disputas y cálculos manuales.

Proyecciones financieras

* A partir de la información registrada, la app muestra proyecciones de flujo de caja, capacidad de ahorro y margen para inversión.
* Ofrece una visión anticipada que las cartolas bancarias no entregan, ya que estas se limitan a mostrar el pasado.

## **Beneficios frente a soluciones actuales**

* Versus cartolas bancarias: la app no se limita a mostrar abonos y cargos sin clasificar; consolida y organiza la información de manera personalizada.
* Versus apps de bancos: no depende de un único ecosistema, integra múltiples fuentes y permite gestionar deudas compartidas y cuotas.
* Versus Excel: elimina la necesidad de conocimientos técnicos avanzados y ofrece reportes automáticos, con menos pasos y mayor claridad.

## **Valor agregado**

* Para pymes: un control financiero integral que mejora la planificación de caja, fortalece la relación con proveedores y reduce el riesgo de morosidad.
* Para personas naturales: una herramienta que visibiliza los “gastos hormiga”, fomenta el ahorro y facilita la toma de decisiones sobre inversiones y consumo.
* Para el mercado: una solución freemium/premium con potencial de escalabilidad, capaz de generar ingresos sostenibles y de posicionarse como referente en gestión financiera simple y accesible.

# **Beneficiarios**

## **Personas naturales**

Las personas naturales, en su mayoría, no cuentan con una formación financiera ni el hábito de registrar de manera sistemática los ingresos y gastos. Es por esto que dependen de resúmenes bancarios o del simple *“saldo disponible”* en sus cuentas, lo que no refleja un compromiso futuro ni ayuda a identificar cosas como los *“gastos hormigas”*.

Con la aplicación, estas personas podrán:

* **Visibilizar gastos invisibles**: al registrar consumos pequeños y recurrentes, tendrán la visualización del impacto real de los gastos hormigas
* **Planificar ahorros**: gracias a las proyecciones, podrán estimar cuanto pueden destinar al ahorro sin comprometer su liquidez
* **Clasificar sus finanzas**: cada usuario podrá organizar sus finanzas según sus propias prioridades (hogar, ocio, transporte, educación, etc.)
* **Evitar errores y omisiones**: al tener todo en un solo lugar, reducirán la probabilidad de olvidar deudas o confundir gastos con ingresos.

En definitiva, la app entrega a las personas naturales una herramienta clara, sencilla y práctica para mejorar el control de sus recursos, fomentar el ahorro y tomar decisiones más informadas sobre consumo e inversión.

## **Microempresas (pymes)**

Las pymes enfrentan grandes dificultades en el control de sus flujos de caja, la gestión de deudas y la planificación de inversiones. Algunas dependen de Excel, cartolas bancarias o los registros dispersos, lo que generan desorden y una mayor pérdida de tiempo

Con la aplicación, las pymes se beneficiarán de:

* **Unificación de información financiera:** consolidación en un solo lugar de datos provenientes de bancos, Transbank, SII y registros internos.
* **Control de caja fortalecido:** con proyecciones y categorizaciones personalizadas, podrán anticipar periodos de déficit o superávit.
* **Gestión de deudas y pagos**: la app facilita el registro de compromisos financieros, evitando olvidos y reduciendo el riesgo de morosidad.
* **Mayor confianza con proveedores:** al contar con un control más riguroso, podrán cumplir puntualmente sus compromisos, fortaleciendo relaciones comerciales y mejorando su reputación.
* **Ahorro de tiempo y reducción de errores:** en lugar de revisar varias plataformas y cartolas, tendrán información clara y organizada de inmediato.

En resumen, las pymes obtendrán un sistema de gestión financiera integral que mejora su eficiencia operativa y les permite concentrarse en lo que realmente importa: hacer crecer su negocio.

## **El mercado**

A nivel de mercado, la solución también genera beneficios:

* **Accesibilidad:** al ser una app independiente de bancos y pensada en un esquema freemium/premium, puede llegar tanto a usuarios básicos como a quienes demanden mayores funcionalidades.
* **Escalabilidad:** el modelo permite crecer en número de usuarios sin necesidad de depender de convenios con instituciones financieras específicas.
* **Formalización financiera:** al facilitar que personas y pymes registren y organicen mejor sus finanzas, se promueve un ecosistema económico más ordenado y con menos riesgos de morosidad.
* **Atracción para inversionistas:** el mercado de apps financieras en Chile y Latinoamérica está en crecimiento, y soluciones simples, accesibles y multi-fuente tienen alto potencial de adopción masiva.

# **Alcance**

## **Alcance general**

El proyecto considera el desarrollo de una aplicación móvil de finanzas, independiente de bancos, diseñada en Angular e Ionic y soportada en Firebase/Firestore para almacenamiento seguro. Su alcance principal es brindar a personas naturales y microempresas (pymes) una herramienta de gestión financiera simple, clara y accesible, que permita unificar, clasificar y proyectar su información económica sin depender de planillas complejas o de la fragmentación de cartolas bancarias.

## **Funcionalidades incluidas en la primera versión**

La versión inicial contempla las siguientes capacidades:

* **Ingreso manual de datos financieros**: el usuario podrá registrar ingresos, gastos, deudas, cuotas e inversiones.
* **Unificación en un solo panel**: los datos registrados se centralizarán, evitando la dispersión entre cartolas y plataformas.
* **Clasificación flexible de movimientos**: cada operación podrá ser etiquetada según categorías personalizadas, adaptándose a las necesidades del usuario.
* **Gestión de deudas y gastos compartidos**: módulo para dividir cuentas o compras entre varias personas, con cálculos automáticos de los montos correspondientes.
* **Proyecciones financieras**: la aplicación mostrará estimaciones de flujo de caja, capacidad de ahorro y margen de inversión a partir de la información ingresada.
* **Reportes claros**: generación de resúmenes visuales y fáciles de interpretar, diferenciando entre ingresos, gastos, deudas e inversiones.

## **Alcance para usuarios**

* **Personas naturales:** tendrán acceso a una herramienta sencilla que les permitirá visibilizar sus gastos, planificar su ahorro y organizar mejor sus finanzas personales.
* **Pymes:** podrán mejorar el control de caja, gestionar deudas y optimizar la relación con proveedores gracias a un registro más confiable y organizado.

# **Metodología**

## **Enfoque general**

El proyecto se desarrollará bajo una metodología de iteración incremental tradicional, lo que significa que la aplicación no se construirá en una sola entrega final, sino en etapas sucesivas, donde cada versión incorpora nuevas funcionalidades sobre una base estable.

Este enfoque permite avanzar de manera controlada, entregando resultados utilizables desde etapas tempranas, reduciendo riesgos y validando continuamente que la solución responda a las necesidades de personas naturales y pymes.

## **Etapas del desarrollo**

1. **Análisis y levantamiento de requerimientos**

* Definición de las funcionalidades prioritarias de la aplicación.
* Identificación de las necesidades específicas de los dos tipos de beneficiarios principales: personas naturales y pymes.
* Revisión de los sistemas actuales que los usuarios emplean (cartolas bancarias, Excel, apps bancarias) para asegurar que la solución entregue un valor diferencial.

1. **Diseño de la arquitectura y experiencia de usuario (UX/UI)**

* Diseño de la arquitectura técnica en base a Angular, Ionic y Firebase/Firestore.
* Definición de la experiencia de usuario con prototipos de interfaces simples, claras y adaptadas a móviles.
* Creación de diagramas funcionales y de flujo de la información.

1. **Desarrollo incremental**

* Construcción progresiva de módulos clave:
* Ingreso manual de datos financieros.
* Clasificación flexible de operaciones.
* Módulo de deudas y gastos compartidos.
* Proyecciones financieras básicas.
* Reportes y visualizaciones.
* Cada módulo se integra con los anteriores, asegurando un sistema estable y coherente.

1. **Pruebas y validación**

* Pruebas unitarias y funcionales de cada módulo desarrollado.
* Validación con casos de uso de personas naturales y pymes para asegurar la utilidad real.
* Corrección de errores y mejoras en base a la retroalimentación de usuarios de prueba.

1. **Implementación y despliegue**

* Preparación del entorno para publicación en tiendas móviles.
* Entrega de la aplicación lista para su uso por los primeros usuarios.

# **Requerimientos**

## **No Funcionales**

|  |  |  |
| --- | --- | --- |
| **ID** | **Requerimiento No Funcional** | **Detalle / Justificación** |
| RNF-01 | Tecnología base: la aplicación debe desarrollarse en Angular e Ionic. | Permite construir una app híbrida, compatible con Android e iOS, con un solo código. |
| RNF-02 | Base de datos: debe usar Firebase/Firestore. | Asegura almacenamiento en la nube, sincronización y seguridad en el manejo de datos. |
| RNF-03 | Seguridad de datos: toda la información debe transmitirse con cifrado. | Protege la información sensible de usuarios frente a ataques o intercepciones. |
| RNF-04 | Confidencialidad: los datos registrados por el usuario no podrán compartirse sin autorización explícita. | Refuerza la confianza de usuarios y cumple con buenas prácticas de privacidad. |
| RNF-05 | Disponibilidad: la aplicación debe estar disponible al menos el 99% del tiempo. | Para garantizar continuidad de uso, especialmente en pymes que dependen de ella a diario. |
| RNF-06 | Rendimiento: las consultas y registros deben responder en un máximo de 3 segundos. | Asegura fluidez en la experiencia del usuario. |
| RNF-07 | Usabilidad: la interfaz debe ser intuitiva, clara y simple, orientada a usuarios sin conocimientos técnicos. | Permite adopción masiva tanto en personas naturales como pymes. |
| RNF-08 | Soporte offline: la app debe permitir registrar operaciones sin conexión a internet. | Garantiza continuidad en contextos de baja conectividad, sincronizando luego los datos. |
| RNF-09 | Escalabilidad: la arquitectura debe permitir la incorporación de futuras funcionalidades. | Ejemplo: integraciones bancarias, módulos de notificaciones o IA. |
| RNF-10 | Mantenibilidad: el código debe estar documentado y estructurado para facilitar actualizaciones. | Reduce costos de mantenimiento y asegura continuidad del proyecto. |

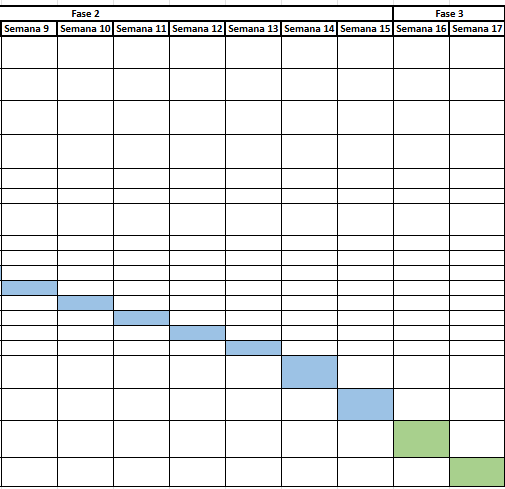
## **Funcionales**

|  |  |  |
| --- | --- | --- |
| **ID** | **Requerimiento Funcional** | **Detalle / Criterio de Aceptación** |
| RF-01 | Registro de usuario y autenticación | Permitir crear cuenta, iniciar/cerrar sesión y recuperar contraseña usando Firebase Authentication. |
| RF-02 | Perfil de usuario | Guardar nombre, tipo de usuario (persona/pymes), moneda (CLP por defecto) y preferencias básicas. |
| RF-03 | Ingreso manual de ingresos | Crear ingresos con fecha, monto, categoría, descripción y medio de pago. |
| RF-04 | Ingreso manual de gastos | Crear gastos con fecha, monto, categoría, descripción y medio de pago. |
| RF-05 | Registro de deudas | Crear deudas con acreedor/deudor, monto total, número de cuotas, tasa (opcional) y fechas de vencimiento. |
| RF-06 | Registro de inversiones | Registrar inversiones con tipo, monto, fecha y horizonte (sin cálculos automáticos externos). |
| RF-07 | Edición y eliminación de registros | Editar/eliminar ingresos, gastos, deudas e inversiones con confirmación. |
| RF-08 | Clasificación flexible | Crear/editar categorías personalizadas y asignarlas a cualquier movimiento. |
| RF-09 | Búsqueda y filtros | Filtrar por fecha, tipo (ingreso/gasto/deuda/inversión), categoría y rango de montos; búsqueda por texto. |
| RF-10 | Panel unificado | Mostrar en una vista consolidada saldos, totales de ingresos/gastos, deudas activas e inversiones registradas. |
| RF-11 | Proyecciones financieras | Generar proyecciones básicas (mensuales/trimestrales) de flujo, capacidad de ahorro y margen para inversión con datos ingresados. |
| RF-12 | Gestión de gastos/deudas compartidas | Dividir un gasto/deuda entre varias personas, asignar porcentajes o partes iguales y calcular automáticamente los montos. |
| RF-13 | Adjuntos (comprobantes) | Adjuntar imágenes/PDF de boletas o facturas a un movimiento; visualizar/descargar desde el detalle. |
| RF-14 | Moneda y formato local | Usar CLP por defecto y formatos de fecha/monto locales; permitir cambiar moneda (sin tipos de cambio automáticos). |
| RF-15 | Persistencia en Firestore | Guardar y recuperar todos los datos del usuario en colecciones seguras por UID. |
| RF-16 | Validaciones | Validar campos obligatorios, tipos de dato, montos positivos y fechas válidas; mensajes claros ante errores. |
| RF-17 | Gestión de categorías | CRUD de categorías (crear, renombrar, eliminar) y reasignación masiva al eliminar una categoría. |
| RF-18 | Copia de seguridad/exportación | Exportar todos los datos del usuario a CSV/JSON desde la app. |
| RF-19 | Eliminar cuenta (Right to Delete) | Permitir al usuario borrar su cuenta y todos sus datos asociados. |

|  |  |  |
| --- | --- | --- |
| **Capa / Área** | **Tecnología / Servicio** | **Rol en el proyecto** |
| Frontend móvil (UI) | Angular 20.1.5 | Base del front: componentes, enrutamiento, estados y lógica de presentación. |
| UI Móvil / Bridge nativo | Ionic 7.2.1 | Componentes UI móviles y empaquetado híbrido para Android/iOS. |
| Acceso a hardware/dispositivo | Capacitor (Ionic) | Plugins nativos (filesystem, network, device info) si se requieren. |
| Lenguaje | TypeScript | Tipado estático para reducir errores y mejorar mantenibilidad. |
| Entorno runtime | Node.js 22.14.0 + npm 10.9.2 | Gestión de dependencias, ejecución de scripts de build y entorno de desarrollo. |
| Backend-as-a-Service | Firebase | Plataforma administrada: autenticación, base de datos, reglas de seguridad, almacenamiento. |
| Autenticación | Firebase Authentication | Registro, login, recuperación de contraseña. |
| Base de datos | Cloud Firestore | Persistencia de ingresos, gastos, deudas, inversiones, categorías, perfiles, etc. |
| Reglas de seguridad | Firestore Security Rules | Control de acceso a documentos/colecciones por UID. |
| Almacenamiento de archivos | Firebase Storage | Respaldo de comprobantes (boletas/facturas) en imágenes/PDF. |
| Control de versiones | Git + GitHub | Versionado, PRs, manejo de ramas y revisiones de código. |
| Lenguaje adicional | Python 3.13.2 | Uso en análisis de datos, generación de reportes y posibles módulos de procesamiento offline. |

# **Tecnologías**

# **Cronograma**



# **Costos**

El presente diagrama de costos refleja los gastos reales asociados al desarrollo del proyecto Capstone.

# **Tabla de Costos Estimados**

|  |  |  |
| --- | --- | --- |
| **Categoría** | **Concepto** | **Costo estimado (CLP)** |
| **Tecnología** | Firebase (plan gratuito) | $0 |
| **Operación** | Internet (2 × $14.990 × 4 meses) | $119.920 |
| Electricidad (2 casas × 4 meses) | $840.000 |
| Pruebas en celulares personales | $0 |
| **Académico** | Impresiones | $10.000 |
| Transporte (práctica + instituto) | $80.000 |
| **Recurso Humano** | Horas Hombre (320 h × $5.000) | $1.600.000 |
| **Software** | Visual Studio Code | $0 |
| Node.js, Angular, Ionic | $0 |
| **Sistema** | PCs personales de los integrantes | $1.600.000 |
| Sistema operativo Windows 10 pro | $0 |
| **Total estimado** | | **$4.249.920** |

# **Distribución de Costos**

La mayor proporción de costos corresponde a las **horas hombre (71%)**, que reflejan el esfuerzo invertido por ambos integrantes en el desarrollo del proyecto y la práctica profesional. La **electricidad (23%)** e **internet (3%)** representan gastos operativos indispensables para sostener el trabajo remoto. Finalmente, el **transporte (2%)** y las **impresiones (<1%)** constituyen costos académicos menores pero necesarios para cumplir con entregas y actividades presenciales.

# 

# **Riesgos**

## **Riesgos técnicos**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Riesgo** | **Probabilidad** | **Impacto** | **Mitigación** |
| RT-01 | Versiones nuevas de Angular/Ionic: posibles cambios y deprecaciones frente a lo que dominamos. | Media | Media | Capacitación autodidacta con documentación oficial, tutoriales y foros; pruebas de concepto antes de avanzar. |
| RT-02 | Uso limitado de Firestore: la experiencia previa es solo con Authentication. | Alta | Alta | Estudiar estructura de colecciones/subcolecciones, diseñar modelo de datos antes de implementar. |
| RT-03 | Dependencia de librerías externas desconocidas. | Media | Media | Fijar versiones estables, evaluar alternativas, documentar el uso de cada librería. |
| RT-04 | Errores de integración Angular ↔ Firebase/Firestore. | Media | Alta | Validar integración paso a paso, usar entornos de prueba, revisar reglas de seguridad. |
| RT-05 | Consumo excesivo de recursos (RAM/batería) por inexperiencia en optimización. | Media | Media | Revisar documentación de Ionic y buenas prácticas de rendimiento; pruebas de carga básica. |
| RT-06 | Errores de modelado de datos en Firestore que obliguen a rehacer colecciones. | Media | Alta | Diseñar esquema inicial revisado entre los dos, validar antes de poblar datos reales. |
| RT-07 | Limitaciones en almacenamiento gratuito de Firebase (cuotas superadas). | Baja | Media | Monitorizar consumo, usar plan gratuito solo para pruebas, considerar costos mínimos si se escala. |

## **Riesgos de gestión del proyecto**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Riesgo** | **Probabilidad** | **Impacto** | **Mitigación** |
| RG-01 | Retrasos por curva de aprendizaje en nuevas versiones y Firestore. | Alta | Alta | Estimar tiempos con holgura, priorizar funcionalidades críticas. |
| RG-02 | Sobrecarga académica por práctica y capstone. | Alta | Media | Planificación semanal realista, dividir tareas equitativamente. |
| RG-03 | Enfermedades o imprevistos personales. | Baja | Alta | Mantener entregas parciales, respaldo de código en GitHub, comunicación constante. |
| RG-04 | Descoordinación en el uso de Git/GitHub (conflictos de merge, pérdida de cambios). | Media | Media | Definir flujo de commits claro, ramas separadas y revisiones conjuntas. |
| RG-05 | Dependencia de internet para documentarse. | Media | Baja | Descargar documentación offline, mantener ejemplos locales como guía. |
| RG-06 | Subestimación del esfuerzo de pruebas. | Media | Media | Dedicar tiempo a validar con datos de prueba antes de considerar el desarrollo como “listo”. |
| RG-07 | Comunicación insuficiente entre los dos integrantes. | Baja | Alta | Reuniones cortas de seguimiento (diarias o cada dos días) para revisar avances y problemas. |
| RG-08 | Falta de experiencia en gestión de proyectos grandes. | Media | Media | Apoyarse en metodologías básicas de iteración incremental, mantener documentación simple pero clara. |

## **Riesgos de calidad**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Riesgo** | **Probabilidad** | **Impacto** | **Mitigación** |
| RQ-01 | Código no optimizado o difícil de mantener. | Media | Media | Seguir guías de estilo oficiales, usar linters ( ; ) básicos y comentar adecuadamente. |
| RQ-02 | Interfaz poco intuitiva para usuarios sin experiencia en apps financieras. | Media | Alta | Crear prototipos simples, pedir retroalimentación a compañeros antes de programar. |
| RQ-03 | Errores de usabilidad (formularios poco claros, inputs sin validaciones). | Media | Alta | Incluir validaciones de campos y mensajes de error amigables. |
| RQ-04 | Falta de pruebas en diferentes dispositivos. | Alta | Media | Probar en emuladores Android/iOS y al menos en un par de celulares físicos. |
| RQ-05 | Documentación insuficiente del proyecto. | Media | Media | Documentar instalación, dependencias y flujo de uso en README y comentarios de código. |
| RQ-06 | Dependencia excesiva del almacenamiento manual de datos (error humano de usuario). | Alta | Media | Minimizar pasos de ingreso, permitir edición fácil, mensajes claros al registrar. |

# **Pruebas**

## **Alcance de pruebas**

* **Funcional**: Ingresos, gastos, deudas, inversiones, clasificación, filtros/búsquedas, panel unificado, proyecciones, reportes, adjuntos, conciliación manual, calendario de vencimientos, operación offline con posterior sincronización.
* **No funcional**: Usabilidad básica (claridad de formularios), rendimiento percibido (≤3 s en operaciones comunes), soporte offline, seguridad de datos (no exponer información entre usuarios).

## **Datos de prueba (base mínima)**

* **Personas naturales**
* Ingresos: sueldo $900.000 (mensual), ingreso extra $120.000 (una vez)
* Gastos: alimentación $180.000, transporte $60.000, ocio $70.000, servicios $80.000
* Deuda: celular $360.000 en 6 cuotas
* Inversión: depósito $100.000 (registro simple, sin cálculo externo)
* **Pyme**
* Ingresos (ventas): 10 registros entre $25.000 y $180.000
* Gastos: proveedor A $250.000, proveedor B $180.000, servicios $90.000
* Deuda: compra insumos $600.000 en 4 cuotas.
* Inversión: $300.000 (registro simple)

Nota: todos los montos en CLP; fechas dentro del mes corriente y el siguiente para observar proyecciones.

## **Matriz de casos de prueba funcionales**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Módulo** | **Caso** | **Pasos (en la app)** | **Resultado esperado (Criterio de aceptación)** |
| PF-01 | Registro/Acceso | Crear cuenta de usuario | Abrir app → “Crear cuenta” → ingresar datos → Guardar | Se crea la cuenta en Firebase Auth; el usuario queda autenticado y redirigido al panel inicial. |
| PF-02 | Autenticación | Iniciar sesión | Abrir app → ingresar correo/contraseña → Ingresar | Acceso concedido; se muestra el panel con datos del usuario. |
| PF-03 | Perfil de usuario | Configurar perfil | Menú → “Perfil” → nombre, tipo (persona/pyme), moneda (CLP) → Guardar | Datos almacenados en Firestore y visibles al reabrir la app. |
| PF-04 | Ingresos | Crear ingreso manual (persona) | Abrir “Nuevo ingreso” → completar (monto, fecha, categoría, medio, descripción) → Guardar | El ingreso aparece en la lista; el panel unificado actualiza totales sin recargar la pantalla. |
| PF-05 | Gastos | Crear gasto manual (persona) | “Nuevo gasto” → completar campos → Guardar | Lista y panel actualizan totales; se refleja el nuevo saldo. |
| PF-06 | Deudas | Registrar deuda con cuotas (persona) | “Nueva deuda” → monto total, número de cuotas, fechas de vencimiento → Guardar | Se registran las cuotas programadas; el panel refleja deuda activa y futuras obligaciones. |
| PF-07 | Inversiones | Registrar inversión (persona) | “Nueva inversión” → tipo, monto, fecha → Guardar | La inversión se lista y el panel la distingue de ingresos/gastos. |
| PF-08 | Clasificación | Crear categoría personalizada | “Categorías” → “Nueva categoría” → nombre/color → Guardar | Disponible en selectores; al reclasificar movimientos, reportes reflejan la nueva categoría. |
| PF-09 | Filtros/Búsqueda | Filtrar por rango de fechas y categoría | Vista de movimientos → configurar filtros → aplicar | Se muestran solo los movimientos que cumplen; Exportar CSV respeta el filtro activo. |
| PF-10 | Panel unificado | Ver totales consolidados | Abrir panel principal | Ver totales de ingresos/gastos/deudas/inversiones coherentes con lo ingresado. |
| PF-11 | Deudas compartidas | Dividir gasto entre 3 personas | “Nuevo gasto compartido” → agregar 3 participantes → partes iguales → Guardar | Monto por persona correcto; total = suma de partes; si se edita un valor, el cálculo se actualiza correctamente. |
| PF-12 | Edición | Editar gasto (monto/fecha) | Abrir gasto → Editar → Guardar | Totales se recalculan; el cambio queda persistido en Firestore. |
| PF-13 | Eliminación | Eliminar ingreso | Abrir ingreso → Eliminar → Confirmar | Se elimina el registro; panel/reportes se actualizan acorde. |
| PF-14 | Exportación | Exportar CSV del mes | Reportes → seleccionar período → “Exportar CSV” | Archivo CSV con columnas y formato esperado; cantidad de filas coincide con el filtro aplicado. |
| PF-15 | Persistencia | Cerrar sesión y volver a ingresar | Registrar algunos movimientos → Cerrar sesión → Iniciar sesión | Todos los datos permanecen disponibles para el usuario autenticado. |
| PF-16 | Categorías | Eliminar categoría usada (con reasignación) | Menú “Categorías” → eliminar categoría → seleccionar categoría destino para movimientos afectados | La categoría se elimina y los movimientos quedan correctamente reasignados. |
| PF-17 | Respaldo | Exportar respaldo completo | Menú → “Exportar” → “Exportar todo (CSV/JSON)” | Archivo con todos los registros del usuario, legible y con consistencia de conteos. |
| PF-18 | Cuenta | Eliminar cuenta y datos | Menú → “Configuración” → “Eliminar cuenta” → Confirmar | Se elimina la cuenta de Firebase Auth y todos los datos del usuario en Firestore. |

## **Pruebas negativas y de validación**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Caso** | **Entrada inválida** | **Resultado esperado** |
| PN-01 | Monto vacío | Monto = “” | Mensaje claro: “El monto es obligatorio”; no guarda el registro. |
| PN-02 | Monto negativo | Monto = -1000 | Mensaje: “El monto debe ser positivo”; no guarda. |
| PN-03 | Fecha inválida | Vencimiento anterior a la fecha de origen | Mensaje: “La fecha de vencimiento no es válida”; no guarda. |
| PN-04 | Categoría inexistente | Eliminar categoría que está en uso | Forzar reasignación a otra categoría antes de eliminar; impedir eliminación directa. |
| PN-05 | Duplicado accidental | Crear 2 veces el mismo gasto en 1 minuto | Mostrar alerta no bloqueante: “¿Seguro? Parece duplicado”. |

## **Pruebas de rendimiento percibido (en la app)**

* **Acciones objetivo:** crear/editar/eliminar movimiento; abrir panel; filtrar/buscar; abrir reportes.
* **Criterio:** tiempo de respuesta ≤ 3 s (RNF-06) con los datasets base definidos en el informe.
* **Medición:** cronómetro manual y logs de tiempo en consola (solo desarrollo), sin herramientas externas.

## **Pruebas de usabilidad básica (dentro de la app)**

**Objetivo:** que una persona sin experiencia registre 5 movimientos y vea un reporte sin pedir ayuda.

**Criterios:**

* Completa el flujo en ≤ 8 minutos.
* No comete más de 2 errores corregidos por validación.
* Comprende dónde ver totales en el panel y cómo exportar un reporte.

## **Funciones mínimas para el lanzamiento inicial**

Marcar **✅** si se cumple en la app:

* Puedo registrar ingresos, gastos, deudas e inversiones y verlos unificados.
* Puedo clasificar y luego filtrar/buscar por categoría/fecha.
* Puedo dividir un gasto/deuda con otras personas y el cálculo es correcto.
* Puedo exportar un reporte a CSV (y respaldo completo CSV/JSON).
* La app funciona offline y sincroniza al volver la conexión.
* Las acciones comunes responden en ≤ 3 s.
* Ningún dato mío aparece en otra cuenta (aislamiento por usuario).