

Documento de Ejecución del Software ARKA

Concha Castro, Ana Cecilia Cornejo Castro, José Gabriel
Sardon Medina, Ivan Matthias Silva Murillo Rodrigo Antonio
Valdivia Castillo, José Miguel Villar Ticono, Pamela Allison

17 de Septiembre de 2025

Contents

0.1	RESUMEN	2
1	Introducción	3
2	Ciclo de vida	4
2.1	Análisis	4
2.1.1	Requisitos	5
2.1.1.1	Funcionales:	5
2.1.1.2	No Funcionales:	7
2.1.2	Modelo de requisitos	9
2.1.2.1	Comportamiento	9
2.1.2.1.1	Diagrama de Casos de Uso	9
2.1.2.1.2	Especificaciones de caso de uso	11
2.2	C. Modelo de Datos	11
2.2.1	Modelo Entidad Relación	11
2.2.1.1	Entidades del Sistema	11
2.2.1.2	Relationships (Relaciones)	13
2.2.1.3	Restricciones de Integridad	13
2.2.1.4	Índices Recomendados	14

0.1. RESUMEN

El presente documento describe el proceso de desarrollo del sistema software de gestión de economía familiar Arka, una plataforma orientada a facilitar la organización y control de los ingresos, egresos y presupuestos en el ámbito doméstico. Se detallan los requisitos funcionales y no funcionales, así como los modelos de análisis, incluyendo casos de uso, mockups y diagramas de clases.

A nivel de diseño, se presentan los diagramas de secuencia y clases de diseño que definen la arquitectura técnica del sistema. Asimismo, se incorpora una sección administrativa que documenta las interfaces de usuario y administrador, junto con el diccionario de base de datos.

Este documento proporciona una base integral para la comprensión, validación y futura implementación de Arka.

Palabras clave: Arka, gestión de economía familiar, requisitos de software, análisis de sistemas, diseño de software, UML, finanzas personales.

1. Introducción

El presente documento tiene como objetivo detallar el proceso completo de desarrollo del sistema Arka, una plataforma orientada a la gestión de economía familiar. Esta herramienta busca facilitar la administración de los ingresos, egresos, presupuestos y metas financieras del hogar, promoviendo un control eficiente y organizado de las finanzas personales.

A lo largo de este informe se abordan de manera estructurada todas las etapas fundamentales del ciclo de vida del software, incluyendo el análisis, diseño y construcción.

En la fase de análisis, se identifican los requisitos funcionales y no funcionales, se modelan los casos de uso, y se presentan los modelos de comportamiento, presentación e información del sistema. Posteriormente, se define la arquitectura técnica a través de diagramas de diseño como los de secuencia y clases, con el fin de representar cómo se interconectan de forma eficiente los distintos componentes del sistema.

Además, se incluye una sección administrativa que documenta las interfaces gráficas tanto para usuarios como para administradores, junto con el diccionario de base de datos que respalda las operaciones del sistema. Todo ello con el propósito de ofrecer una visión integral y coherente del desarrollo del sistema Arka, facilitando su validación, mantenimiento y futura implementación.

2. Ciclo de vida

2.1. Análisis

La presente sección tiene como objetivo analizar en profundidad el sistema Arka, una plataforma orientada a la gestión de economía familiar, que busca brindar a los usuarios herramientas para el control y monitoreo de sus finanzas personales, incluyendo el registro de ingresos, egresos, presupuestos y metas económicas. El análisis permite identificar los principales actores del sistema, los requerimientos funcionales y no funcionales, así como los casos de uso que guiarán el desarrollo del software.

En esta sección se presenta un análisis detallado del sistema Arka, abordando los distintos aspectos que conforman su funcionamiento. Este análisis se divide en tres apartados clave:

- **Modelo de Comportamiento**, que detalla la lógica de interacción entre los componentes del sistema y los usuarios mediante diagramas de secuencia o actividades.
- **Modelo de Presentación**, que describe las interfaces visuales a través de las cuales el usuario interactuará con la plataforma.
- **Modelo de Información**, donde se detalla la estructura de los datos del sistema mediante el diccionario de tablas y otras representaciones del modelo de datos.

El propósito de esta sección es establecer una base sólida para el diseño y construcción del sistema, asegurando una comprensión clara de cómo debe comportarse la aplicación frente a distintos escenarios de uso vinculados a la economía del hogar.

2.1.1. Requisitos

A continuación se presentarán los requisitos para el funcionamiento de Arka, estos serán clasificados en funcionales y no funcionales, se presentan en distintas tablas numeradas para su mayor comprensión:

2.1.1.1. Funcionales:

TABLA I: REQUISITO DE USUARIO PARA EL REGISTRO DE USUARIOS

Campo	Descripción
Requisito:	RU-01
Tipo de Requisito:	Usuario
Caso de Uso:	CU-017
Descripción:	Realizar registro de usuarios (familiares) por parte del administrador.
Justificación:	Permite al administrador incorporar nuevos miembros a la plataforma familiar.
Originador:	Administrador
Criterio de Aceptación:	El administrador puede registrar usuarios y asignar límites de gasto.
Prioridad:	Crítica

TABLA II: REQUISITO DE USUARIO PARA INGRESAR AL SISTEMA

Campo	Descripción
Requisito:	RU-02
Tipo de Requisito:	Usuario
Caso de Uso:	CU-001
Descripción:	Realizar inicio de sesión (login).
Justificación:	Es necesario para ingresar al sistema de manera segura.
Originador:	Usuario final (familiar).
Criterio de Aceptación:	El sistema permite validar las credenciales e ingresar al sistema.
Prioridad:	Crítica

TABLA III: REQUISITO DE USUARIO PARA CERRAR SESIÓN

Campo	Descripción
Requisito:	RU-03
Tipo de Requisito:	Usuario
Caso de Uso:	CU-002
Descripción:	Cerrar sesión.
Justificación:	Permite al usuario salir del sistema y proteger su información.
Originador:	Usuario final (familiar).
Criterio de Aceptación:	El sistema cierra sesión y redirige al inicio de sesión.
Prioridad:	Importante

TABLA IV: REQUISITO DE USUARIO PARA VISUALIZAR BALANCE

Campo	Descripción
Requisito:	RU-04
Tipo de Requisito:	Usuario
Caso de Uso:	CU-004
Descripción:	Visualizar balance personal con ingresos y egresos.
Justificación:	Permite al usuario analizar su situación financiera personal.
Originador:	Usuario final (familiar).
Criterio de Aceptación:	El sistema muestra ingresos, egresos y balance filtrados.
Prioridad:	Importante

TABLA V: REQUISITO PARA LA VISUALIZACIÓN DE GRÁFICOS DE BALANCE PERSONAL

Campo	Descripción
Requisito:	RU-05
Tipo de Requisito:	Usuario
Caso de Uso:	CU-005
Descripción:	Visualizar gráficos de balance personal.
Justificación:	Ofrece una visualización intuitiva del estado financiero.
Originador:	Usuario final (familiar).
Criterio de Aceptación:	El usuario accede a gráficos personalizados de ingresos y egresos.
Prioridad:	Importante

2.1.1.2. No Funcionales:**TABLA VI: REQUISITO DE RENDIMIENTO**

Campo	Descripción
Requisito:	RNF-01
Tipo de Requisito:	Rendimiento
Descripción:	El sistema debe responder en menos de 2 segundos para operaciones básicas.
Justificación:	Garantizar una experiencia de usuario fluida.
Criterio de Aceptación:	Tiempo de respuesta menor o igual a 2 segundos para el 95% de las operaciones.
Prioridad:	Importante

TABLA VII: REQUISITO DE SEGURIDAD

Campo	Descripción
Requisito:	RNF-02
Tipo de Requisito:	Seguridad
Descripción:	El sistema debe encriptar toda la información financiera sensible.
Justificación:	Proteger la información financiera de los usuarios.

Campo	Descripción
Criterio de Aceptación:	Uso de encriptación AES-256 para datos sensibles.
Prioridad:	Crítica

TABLA VIII: REQUISITO DE USABILIDAD

Campo	Descripción
Requisito:	RNF-03
Tipo de Requisito:	Usabilidad
Descripción:	La interfaz debe ser intuitiva para usuarios no técnicos.
Justificación:	Facilitar la adopción por parte de familias.
Criterio de Aceptación:	90% de usuarios completan tareas básicas sin ayuda.
Prioridad:	Importante

2.1.2. Modelo de requisitos

Aquí se desarrollarán las especificaciones necesarias para el correcto desarrollo y funcionamiento del sistema. Este modelo se divide en 3 componentes principales: el modelo de comportamiento, el modelo de presentación y modelo de información

2.1.2.1. Comportamiento

El modelo de comportamiento describe la lógica de interacción entre los componentes del sistema y los usuarios a través de diagramas de casos de uso.

2.1.2.1.1. Diagrama de Casos de Uso

El sistema Arka cuenta con los siguientes actores principales:

- **Administrador Familiar:** Usuario con privilegios para gestionar la configuración familiar
- **Usuario Familiar:** Miembro de la familia que gestiona sus finanzas personales

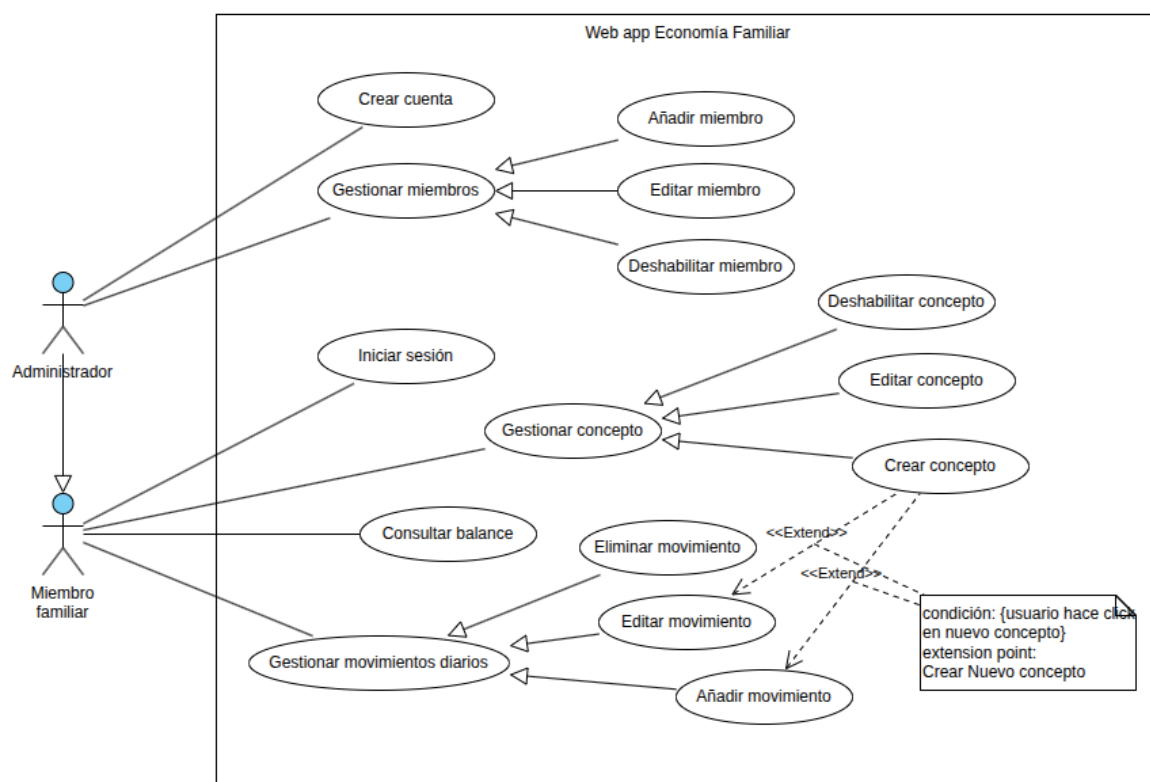


Figure 1: Diagrama de casos de uso del sistema Arka

Para visualizar el diagrama debe acceder al siguiente [repositorio](#) y seleccionar la rama main. Tendrá un listado de varias herramientas. El diagrama de casos de uso se encuentra

en el documento **VisualParadigm.md**, entrar y seleccionar el diagrama de casos de uso.

2.1.2.1.2. Especificaciones de caso de uso

2.2. C. Modelo de Datos

2.2.1. Modelo Entidad Relación

El modelo entidad-relación (E-R) representa la estructura conceptual de la base de datos del sistema Arka, definiendo las entidades principales, sus atributos y las relaciones entre ellas.

2.2.1.1. Entidades del Sistema

FAMILIA

- **Atributos:**
 - id_familia (PK): Identificador único de la familia
 - nombre: Nombre de la familia
 - fecha_creacion: Fecha de creación de la familia
 - presupuesto_general: Presupuesto total de la familia
 - estado: Estado activo/inactivo de la familia

USUARIO

- **Atributos:**
 - id_usuario (PK): Identificador único del usuario
 - id_familia (FK): Referencia a la familia
 - nombre: Nombre completo del usuario
 - email: Correo electrónico único
 - password: Contraseña encriptada
 - rol: Tipo de usuario (administrador/familiar)
 - limite_gasto: Límite de gasto mensual asignado
 - fecha_creacion: Fecha de registro
 - ultimo_acceso: Fecha y hora del último acceso
 - estado: Estado activo/inactivo del usuario

CATEGORIA

- **Atributos:**
 - id_categoria (PK): Identificador único de la categoría
 - id_familia (FK): Referencia a la familia
 - nombre: Nombre de la categoría
 - tipo: Tipo de categoría (ingreso/egreso)
 - presupuesto_asignado: Presupuesto mensual para la categoría

- color: Color identificativo en hexadecimal
- descripcion: Descripción de la categoría
- estado: Estado activo/inactivo

TRANSACCION

- **Atributos:**

- id_transaccion (PK): Identificador único de la transacción
- id_usuario (FK): Referencia al usuario que realizó la transacción
- id_categoria (FK): Referencia a la categoría
- tipo: Tipo de transacción (ingreso/egreso)
- monto: Cantidad monetaria
- descripcion: Descripción detallada
- fecha_transaccion: Fecha de la transacción
- fecha_registro: Fecha de registro en el sistema
- metodo_pago: Efectivo, tarjeta, transferencia, etc.
- estado: Pendiente, confirmada, cancelada

META_FINANCIERA

- **Atributos:**

- id_meta (PK): Identificador único de la meta
- id_usuario (FK): Referencia al usuario propietario
- nombre: Nombre de la meta financiera
- descripcion: Descripción detallada de la meta
- monto_objetivo: Cantidad objetivo a alcanzar
- monto_actual: Cantidad acumulada actualmente
- fecha_inicio: Fecha de inicio de la meta
- fecha_limite: Fecha límite para alcanzar la meta
- estado: Activa, completada, cancelada
- prioridad: Alta, media, baja

PRESUPUESTO

- **Atributos:**

- id_presupuesto (PK): Identificador único del presupuesto
- id_familia (FK): Referencia a la familia
- id_categoria (FK): Referencia a la categoría
- mes: Mes del presupuesto
- año: Año del presupuesto
- monto_asignado: Monto presupuestado
- monto_gastado: Monto utilizado

- fecha_creacion: Fecha de creación del presupuesto

2.2.1.2. Relationships (Relaciones)

1. FAMILIA - USUARIO

- Relación: 1:N (Una familia tiene múltiples usuarios)
- Descripción: Cada usuario pertenece a una única familia

2. FAMILIA - CATEGORIA

- Relación: 1:N (Una familia tiene múltiples categorías)
- Descripción: Las categorías son específicas por familia

3. USUARIO - TRANSACCION

- Relación: 1:N (Un usuario realiza múltiples transacciones)
- Descripción: Cada transacción es registrada por un usuario específico

4. CATEGORIA - TRANSACCION

- Relación: 1:N (Una categoría clasifica múltiples transacciones)
- Descripción: Cada transacción debe estar asociada a una categoría

5. USUARIO - META_FINANCIERA

- Relación: 1:N (Un usuario puede tener múltiples metas)
- Descripción: Las metas financieras son individuales por usuario

6. FAMILIA - PRESUPUESTO

- Relación: 1:N (Una familia maneja múltiples presupuestos)
- Descripción: Los presupuestos se organizan por familia

7. CATEGORIA - PRESUPUESTO

- Relación: 1:N (Una categoría puede tener presupuestos en diferentes períodos)
- Descripción: Cada presupuesto está asociado a una categoría específica

2.2.1.3. Restricciones de Integridad

Restricciones de Entidad:

- Todos los identificadores primarios (PK) deben ser únicos y no nulos
- Los campos email en USUARIO deben ser únicos
- Los nombres de categoría deben ser únicos por familia

Restricciones Referenciales:

- Todas las llaves foráneas (FK) deben referenciar registros existentes
- No se permite eliminar familias con usuarios activos
- No se permite eliminar usuarios con transacciones asociadas
- No se permite eliminar categorías con transacciones asociadas

Restricciones de Dominio:

- Los montos deben ser valores positivos
- Las fechas de límite deben ser posteriores a las fechas de inicio
- Los roles de usuario deben ser 'administrador' o 'familiar'
- Los tipos de transacción deben ser 'ingreso' o 'egreso'
- Los estados deben estar dentro de los valores permitidos

Restricciones Semánticas:

- El monto actual de una meta no puede exceder el monto objetivo
- El monto gastado en un presupuesto no debería exceder el monto asignado
- Un usuario no puede tener transacciones en categorías inactivas
- Las fechas de transacción no pueden ser futuras

2.2.1.4. Índices Recomendados

Para optimizar el rendimiento de las consultas más frecuentes:

- **USUARIO:** Índice en (email), (id_familia, estado)
- **TRANSACCION:** Índice en (id_usuario, fecha_transaccion), (id_categoria, tipo)
- **META_FINANCIERA:** Índice en (id_usuario, estado)
- **PRESUPUESTO:** Índice en (id_familia, mes, año)