****

**UNIVERSIDAD PRIVADA DE TACNA**

**FACULTAD DE INGENIERÍA**

**Escuela Profesional de Ingeniería de Sistemas**

***“App de Gestión Financiera para el Registro y Análisis de Gastos Personales”***

Curso: Patrones de Software

Docente: Ing. Patrick Cuadros

Integrantes:

***Ayma Choque, Erick Yoel (2021072616)***

***Poma Machicado, Fabiola Estefani (2021070030)***

***Tapia Vargas, Dylan Yariet (2021072630)***

**Tacna – Perú**

**2025**

| CONTROL DE VERSIONES | | | | | |
| --- | --- | --- | --- | --- | --- |
| Versión | Hecha por | Revisada por | Aprobada por | Fecha | Motivo |
| 1.0 | MPV | ELV | ARV | 10/10/2020 | Versión Original |

App de Gestión Financiera para el Registro y Análisis de Gastos Personales

Documento de Arquitectura de Software

Versión *{1.0}*

| CONTROL DE VERSIONES | | | | | |
| --- | --- | --- | --- | --- | --- |
| Versión | Hecha por | Revisada por | Aprobada por | Fecha | Motivo |
| 1.0 | MPV | ELV | ARV | 10/10/2020 | Versión Original |

**Contenido**

[***1.***](#_heading=h.m8sr9rfqsbf2) ***INTRODUCCIÓN 5***

[**1.1.**](#_heading=h.a6dfovpdt9wy) **Propósito (Diagrama 4+1) 5**

[**1.2.**](#_heading=h.fettwz3liw6h) **Alcance 5**

[**1.3.**](#_heading=h.e3lstvrx4p7m) **Definición, siglas y abreviaturas 5**

[**1.4.**](#_heading=h.m76q3rdfpy9s) **Organización del documento 5**

[***2.***](#_heading=h.cymktu6205aq) ***OBJETIVOS Y RESTRICCIONES ARQUITECTONICAS 5***

[2.1.1.](#_heading=h.o8aku3lc21f0) Requerimientos Funcionales 5

[2.1.2.](#_heading=h.yvaok1xj5y7v) Requerimientos No Funcionales – Atributos de Calidad 5

[***3.***](#_heading=h.cztg4pwhk6wy) ***REPRESENTACIÓN DE LA ARQUITECTURA DEL SISTEMA 6***

[**3.1.**](#_heading=h.mayl1aajd6ps) **Vista de Caso de uso 6**

[3.1.1.](#_heading=h.ke75jt2n1ojd) Diagramas de Casos de uso 6

[**3.2.**](#_heading=h.ks2npx14sje4) **Vista Lógica 6**

[3.2.1.](#_heading=h.avrxr8lhi9y) Diagrama de Subsistemas (paquetes) 7

[3.2.2.](#_heading=h.5gg147onsb5e) Diagrama de Secuencia (vista de diseño) 7

[3.2.3.](#_heading=h.f1t6p5qxytq5) Diagrama de Colaboración (vista de diseño) 7

[3.2.4.](#_heading=h.z9x27bvzok4r) Diagrama de Objetos 7

[3.2.5.](#_heading=h.4d9izp7wkegf) Diagrama de Clases 7

[3.2.6.](#_heading=h.h9jc3qsbw404) Diagrama de Base de datos (relacional o no relacional) 7

[**3.3.**](#_heading=h.eft4bbn43o35) **Vista de Implementación (vista de desarrollo) 7**

[3.3.1.](#_heading=h.5oihdocvcacl) Diagrama de arquitectura software (paquetes) 7

[3.3.2.](#_heading=h.6lj23az3a35) Diagrama de arquitectura del sistema (Diagrama de componentes) 7

[**3.4.**](#_heading=h.3enzs9ozhqzi) **Vista de procesos 7**

[3.4.1.](#_heading=h.s85nu27iwnsd) Diagrama de Procesos del sistema (diagrama de actividad) 8

[**3.5.**](#_heading=h.s4mf1l7zcxex) **Vista de Despliegue (vista física) 8**

[3.5.1.](#_heading=h.2yukfg279xp5) Diagrama de despliegue 8

[***4.***](#_heading=h.f6gcj1xnpyc) ***ATRIBUTOS DE CALIDAD DEL SOFTWARE 8***

[**Escenario de Funcionalidad 8**](#_heading=h.nxgt94knt9ie)

[**Escenario de Usabilidad 8**](#_heading=h.zi0n09eh5xg)

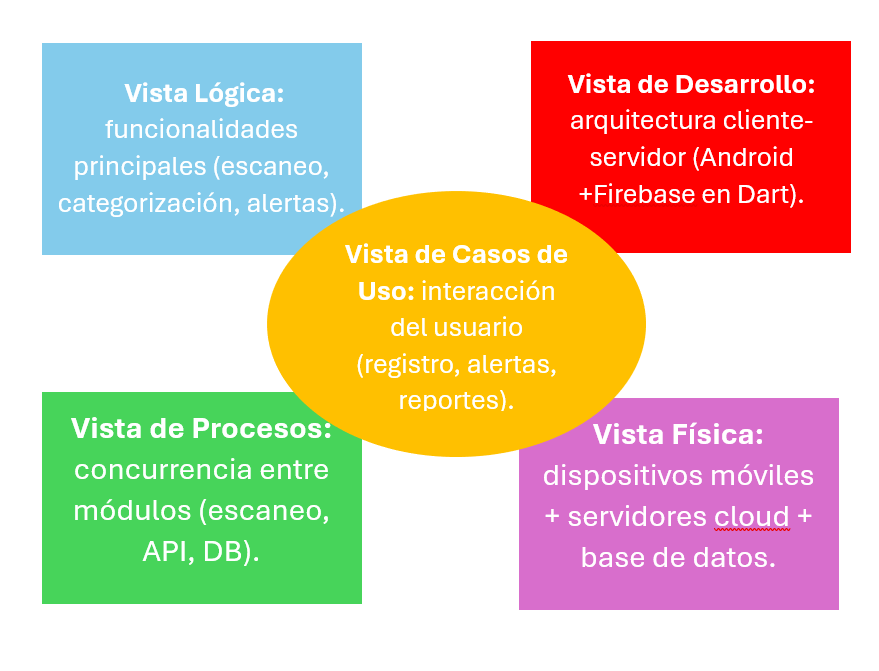
[**Escenario de confiabilidad 9**](#_heading=h.66zj6l1mwi3y)

[**Escenario de rendimiento 9**](#_heading=h.hgkhjzzfiin)

[**Escenario de mantenibilidad 9**](#_heading=h.muu36r1ba1k1)

[**Otros Escenarios 9**](#_heading=h.fcvgwaxe4un0)

1. INTRODUCCIÓN
   1. Propósito (Diagrama 4+1)



* 1. Alcance

Este proyecto tiene como objetivo desarrollar y mejorar una plataforma digital que permita a los usuarios gestionar sus finanzas personales de forma segura y eficiente. Incluye el desarrollo de aplicaciones móviles y web con las siguientes funcionalidades clave:

1. Gestión de Finanzas Personales:

* Registro y gestión de ingresos y egresos.
* Creación de categorías personalizadas para transacciones.
* Generación de gráficos y reportes financieros.

1. Seguridad y Accesibilidad:

* Protección de datos mediante encriptación y autenticación multifactor.
* Interfaces fáciles de usar en dispositivos móviles y web.

1. Escalabilidad y Rendimiento:

* El sistema debe manejar un creciente número de usuarios y transacciones sin perder rendimiento.

Límites del Alcance:

No incluye el desarrollo de productos financieros adicionales (préstamos, inversiones, etc.).

No se incluirán servicios físicos (atención en sucursales o cajeros automáticos).

* 1. Definición, siglas y abreviaturas

#### Definición de Términos Clave

* Usuario: Persona que interactúa con la plataforma digital para gestionar sus finanzas personales.
* Plataforma Digital: Conjunto de aplicaciones móviles y web diseñadas para la gestión de finanzas personales.
* Transacción: Operación financiera registrada por el usuario, ya sea de ingreso o egreso.
* Presupuesto: Límite establecido por el usuario para controlar sus gastos en diferentes categorías.
* Reportes Financieros: Informes generados por la plataforma que muestran un resumen de las finanzas del usuario, como ingresos, egresos y balance general.
* Notificaciones Push: Alertas enviadas a los usuarios en tiempo real sobre eventos relevantes, como gastos excesivos o pagos recurrentes.

#### Siglas y Abreviaturas

* RF: Requerimiento Funcional.
* RNF: Requerimiento No Funcional.
* MFA: Autenticación Multifactor (Multi-Factor Authentication).
* AES: Advanced Encryption Standard (Estándar de Encriptación Avanzado).
* CSV: Comma Separated Values (Valores Separados por Comas).
* PDF: Portable Document Format (Formato de Documento Portátil).
* API: Interfaz de Programación de Aplicaciones (Application Programming Interface).
* UX/UI: Experiencia de Usuario / Interfaz de Usuario (User Experience / User Interface).

# **OBJETIVOS Y RESTRICCIONES ARQUITECTONICAS**

[Establezca las prioridades de los requerimientos y las restricciones del proyecto)

* 1. Priorización de requerimientos

## 

| RF | Requerimiento | Descripción | Prioridad |
| --- | --- | --- | --- |
| RF01 | Registrar usuario | Registro de usuarios con email y contraseña. |  |
| RF02 | Iniciar sesión | Inicio de sesión con validación de credenciales y autenticación de usuario. |  |
| RF03 | Recuperar contraseña | Recuperación de contraseña mediante enlace de recuperación. |  |
| RF04 | Gestionar perfil | Gestión de perfil de usuario (editar, ver información). |  |
| RF05 | Registrar ingresos | Registro de ingresos (monto, categoría, fecha, descripción). |  |
| RF06 | Registrar egresos | Registro de egresos (monto, categoría, fecha, descripción, ubicación opcional). |  |
| RF07 | Editar transacciones | Edición de transacciones registradas. |  |
| RF08 | Crear categorías | Creación de categorías personalizables para ingresos y egresos. |  |
| RF09 | Generar gráficos de gastos | Generación de gráficos de gastos por categoría (barras, pastel). |  |
| RF10 | Visualizar histórico financiero | Visualización de histórico mensual/anual de ingresos y egresos. |  |
| RF11 | Comparar gastos | Comparación de gastos entre diferentes periodos. |  |
| RF12 | Generar balance general | Generación de balance general (ingresos vs egresos). |  |
| RF13 | Clasificar gastos desde imagen | El sistema escanea fotos de boletas y extrae montos, y los categoriza automáticamente. |  |
| RF14 | Clasificar gastos desde audio | El sistema convierte las grabaciones de voz a texto, identifica los datos clave del gasto mencionado y los clasifica en la categoría adecuada de forma automatizada. |  |
| RF15 | Notificar gastos excesivos | Alertar cuando los gastos superen un porcentaje de los ingresos. |  |
| RF16 | Alertar sobre presupuesto | Avisar cuando se alcance el límite presupuestado en una categoría. |  |
| RF17 | Recomendar ahorros | Sugerir pautas de ahorro según patrones de gasto del usuario. |  |
| RF18 | Recordar pagos recurrentes | Notificar sobre vencimientos de servicios o suscripciones. |  |
| RF19 | Generar reportes en PDF | Crear informes financieros en formato PDF. |  |
| RF20 | Exportar datos | Permitir la exportación de datos a formatos CSV o Excel. |  |

### Requerimientos Funcionales

*[Definir la prioridad de los requerimientos funcionales.]*

| *ID* | *Descripcion* | *Prioridad* |
| --- | --- | --- |
|  |  |  |

### 

### Requerimientos No Funcionales – Atributos de Calidad

*[Definir la prioridad de los requerimientos NO funcionales.]*

| *ID* | *Descripcion* | *Prioridad* |
| --- | --- | --- |
|  |  |  |
|  |  |  |

*[Los Atributos de Calidad (QAs) son propiedades medibles y evaluables de un sistema, estas propiedades son usadas para indicar el grado en que el sistema satisface las necesidades de los stakeholders [Wojcik 2013].*

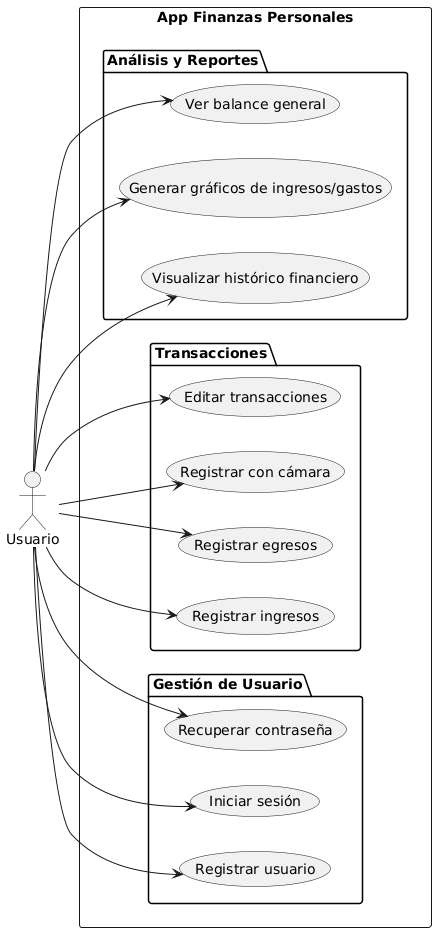
*Los QAs además son concebidos como aquellos requerimientos que no son funcionales. De hecho, la funcionalidad es mayormente ortogonal a los QAs; un diseño puede cumplir con la funcionalidad deseada y fallar a la hora de satisfacer sus requerimientos de calidad. De esta manera, se entiende a la funcionalidad como la capacidad del sistema para hacer el trabajo para el cual fue pensado, independientemente de la estructura. Existen QAs mayormente usados que se suelen identificar en numerosos sistemas y se tienen que describir, aunque la lista no es fina ya que muy a menudo hay situaciones en que podrían identificarse y proponerse nuevas propiedades para las diversas necesidades de stakeholders.]*

* 1. Restricciones

*[Aquí van las restricciones del proyecto]*

# **REPRESENTACIÓN DE LA ARQUITECTURA DEL SISTEMA**

* 1. Vista de Caso de uso



### Diagramas de Casos de uso

*La descripción de la estructura se ilustra utilizando un conjunto de casos de uso escenarios lo que genera una nueva vista. Los escenarios describen secuencia de iteraciones entre objetos y entre procesos. Se utilizan para identificar y validar el diseño de arquitectura.*

* 1. Vista Lógica

*[La vista lógica se encarga de representar los requerimientos funcionales del sistema. Esta sección describe las partes del diseño del modelo significativas para la arquitectura, tales como subsistemas y paquetes.]*

### 

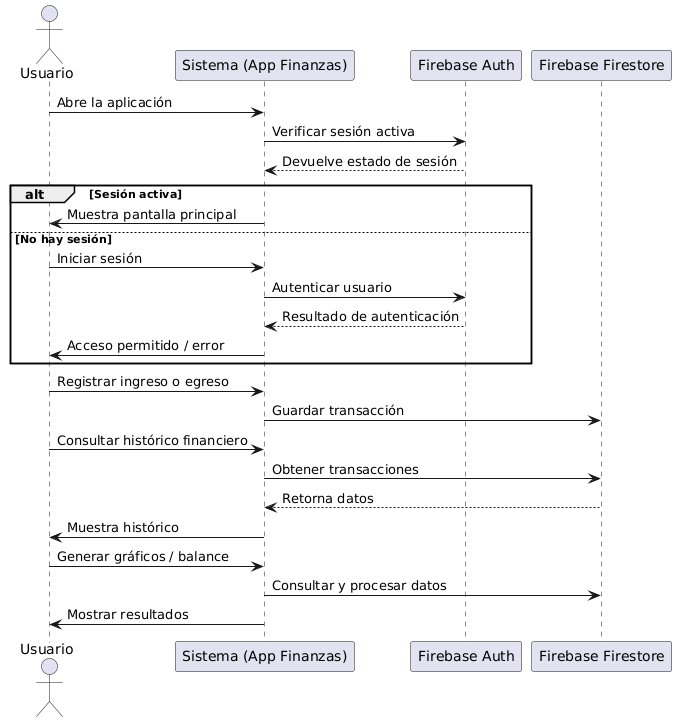
### Diagrama de Subsistemas (paquetes)

*[Diagrama que define los límites entre el sistema, o parte del sistema, y su ambiente, mostrando las entidades que interactúan con él. ​ Este diagrama es una vista de alto nivel de un sistema.*

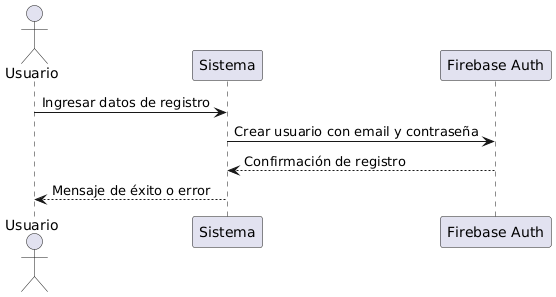
*Asimismo, se debe desplegar las partes arquitectónicamente significativas del modelo de diseño, como ser la descomposición en capas, subsistemas o paquetes. Una vez presentadas estas unidades lógicas principales, se profundiza en ellas hasta el nivel que se considere adecuado.]*

### Diagrama de Secuencia (vista de diseño)

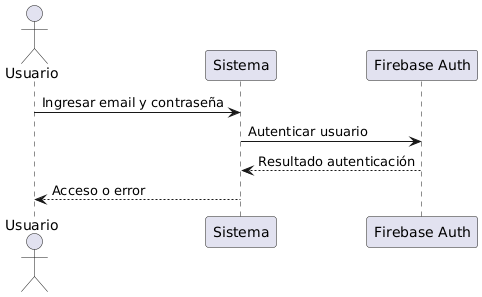
Diagrama de Secuencia General



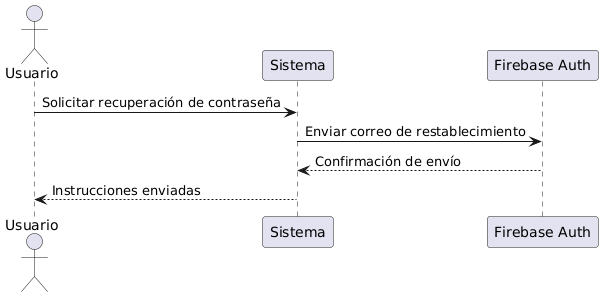
**1. Diagrama de Secuencia: Registrar usuario**



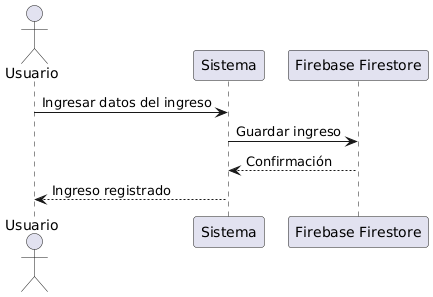
**2. Diagrama de Secuencia: Iniciar sesión**

****

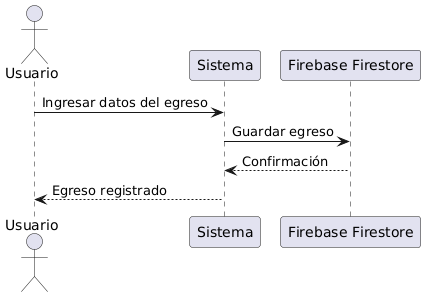
**3. Diagrama de Secuencia: Recuperar contraseña**

****

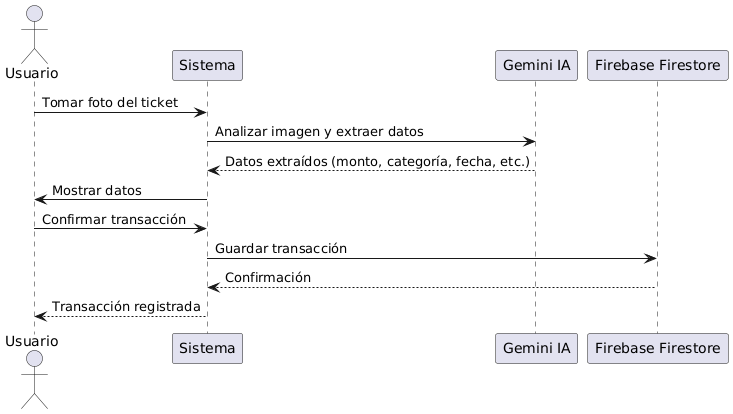
**4. Diagrama de Secuencia: Registrar ingresos**

****

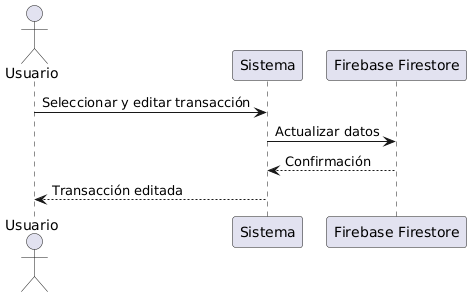
**5. Diagrama de Secuencia: Registrar egresos**

****

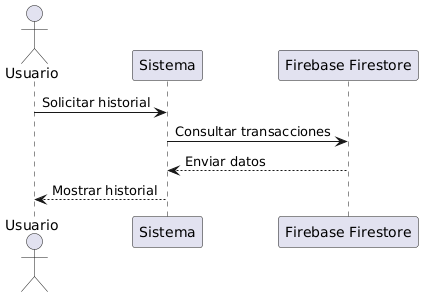
**6. Diagrama de Secuencia: Registrar con cámara**

****

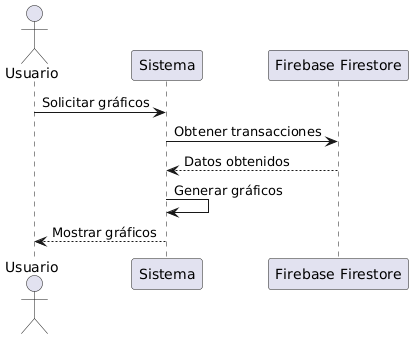
**7. Diagrama de Secuencia: Editar transacciones**

****

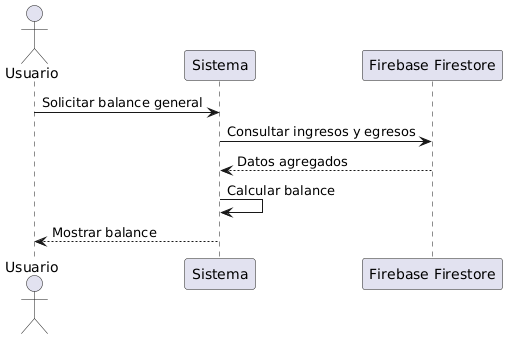
**8. Diagrama de Secuencia: Visualizar histórico financiero**

****

**9. Diagrama de Secuencia: Generar gráficos de ingresos/gastos**

****

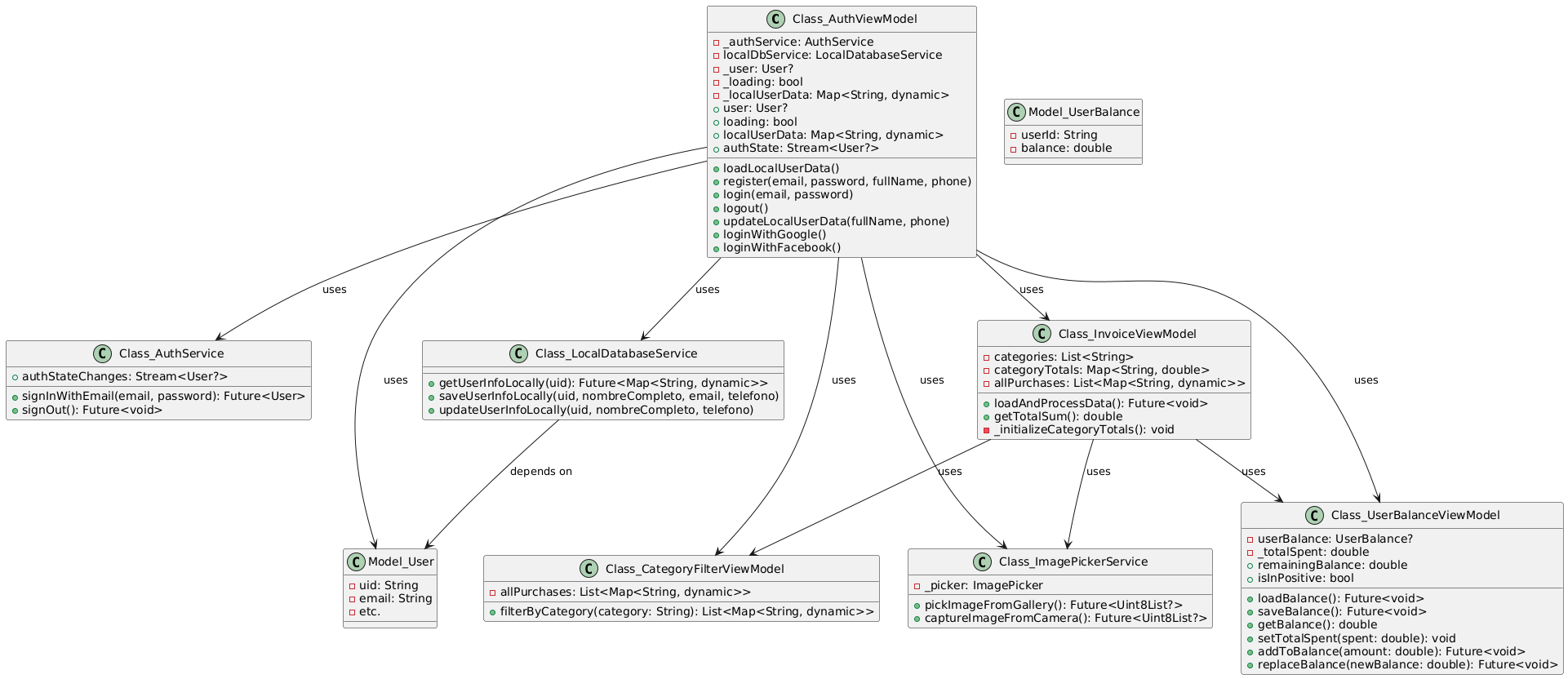
**10. Diagrama de Secuencia: Ver balance general**

****

### Diagrama de Colaboración (vista de diseño)

### Diagrama de Objetos

### Diagrama de Clases



### Diagrama de Base de datos (relacional o no relacional)

* 1. Vista de Implementación (vista de desarrollo)

*[Se detalla la estructura general del Modelo de Implementación y el mapeo de los subsistemas, paquetes y clases de la Vista Lógica a subsistemas y componentes de implementación de manera más detallada]*

### Diagrama de arquitectura software (paquetes)

*[Se detalla la manera como fue implementado el sistema propuesto, se describe visualmente las capas que tiene el sistema, como están distribuidas y sus principales funciones]*

### Diagrama de arquitectura del sistema (Diagrama de componentes)

*[Se detalla la manera como fue implementado el sistema propuesto, se describe visualmente las capas que tiene el sistema, como están distribuidas y sus principales funciones]*

* 1. Vista de procesos

*[Describe la descomposición del sistema procesos pesados. Indica que procesos o grupos de procesos se comunican o interactúan entre sí y los modos en que estos se comunican.]*

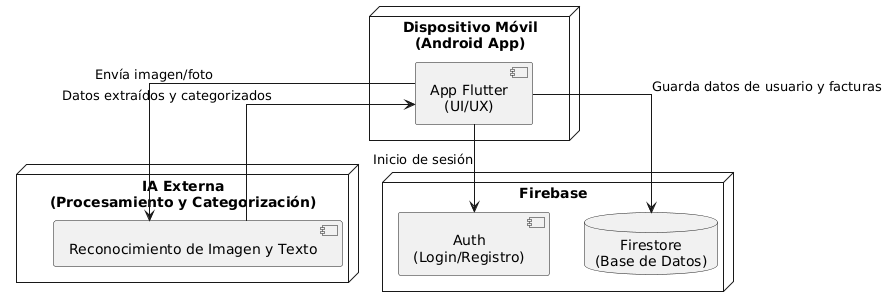
### Diagrama de Procesos del sistema (diagrama de actividad)

*[Se realizará un diagrama del o los procesos del sistema donde se exponga las actividades donde interviene el sistema propuesto, adicionando diagramas que definan el detalle la descomposición del sistema en procesos pesados. Indica que procesos o grupos de procesos se comunican o interactúan entre sí y los modos en que estos se comunican]*

* 1. Vista de Despliegue (vista física)

*[Se despliega uno o más escenarios de distribución física del sistema sobre los cuales se ejecutará y hará el despliegue del mismo. Muestra la comunicación entre los diferentes nodos que componen los escenarios antes mencionados, así como el mapeo de los elementos de la Vista de Procesos en dichos nodos]*

### Diagrama de despliegue

**

# **ATRIBUTOS DE CALIDAD DEL SOFTWARE**

*[Los Atributos de Calidad (QAs) son propiedades medibles y evaluables de un sistema, estas propiedades son usadas para indicar el grado en que el sistema satisface las necesidades de los stakeholders [Wojcik 2013].*

*Los QAs además son concebidos como aquellos requerimientos que no son funcionales. De hecho, la funcionalidad es mayormente ortogonal a los QAs; un diseño puede cumplir con la funcionalidad deseada y fallar a la hora de satisfacer sus requerimientos de calidad. De esta manera, se entiende a la funcionalidad como la capacidad del sistema para hacer el trabajo para el cual fue pensado, independientemente de la estructura. Existen QAs mayormente usados que se suelen identificar en numerosos sistemas y se tienen que describir, aunque la lista no es fina ya que muy a menudo hay situaciones en que podrían identificarse y proponerse nuevas propiedades para las diversas necesidades de stakeholders.]*

*.*

**Escenario de Funcionalidad**

*[se califica de acuerdo con el conjunto de características y capacidades del programa, la generalidad de las funciones que se entregan y la seguridad general del sistema.]*

**Escenario de Usabilidad**

*[Este atributo de calidad se refiere a la facilidad con la que un usuario puede aprender a utilizar e interpretar los resultados producidos por un sistema [Barbacci 1995]. Para este atributo de calidad, se suelen considerar diversos aspectos de la interacción humano computadora, tales como: aprendizaje del sistema, utilización eficiente del sistema, minimización del impacto de errores, adaptación del sistema a las necesidades del usuario, confianza y satisfacción, entre otros.]*

**Escenario de confiabilidad**

*[Es el equilibrio entre la confidencialidad, la integridad, la irrefutabilidad y la disponibilidad de la información y datos manipulados por el sistema. Se trata del estado de un sistema, el cual puede ser transitorio y volátil. La seguridad de un sistema se caracteriza por mecanismos y técnicas empleados para intentar reducir los más posible el impacto provocado por un ataque, y las amenazas (entendidas como los caminos mediante los cuales se pueden provocar un ataque).*

*Abarca los planos de observación físico, lógico y humanos. Posee tres tipos de enfoque: prevención, precaución y reacción.]*

**Escenario de rendimiento**

*[Se mide con base en la velocidad de procesamiento, el tiempo de respuesta, el uso de recursos, el conjunto y la eficiencia.] (Pressman 2010, pág. 187)*

**Escenario de mantenibilidad**

*[Combina la capacidad del programa para ser ampliable (extensibilidad), adaptable y servicial. (Pressman 2010, pág. 187)*

**Otros Escenarios**

*[“Otros escenarios como por ejemplo: Performance”*

***Performance****: El atributo de calidad Performance se refiere a la capacidad de responder, ya sea el tiempo requerido para responder a eventos determinados, o bien, la cantidad de eventos procesados en un intervalo de tiempo dado. La Performance caracteriza la proyección en el tiempo de los servicios entregados por el sistema.]*