****

**UNIVERSIDAD PRIVADA DE TACNA**

**FACULTAD DE INGENIERIA**

**Escuela Profesional de Ingeniería de Sistemas**

**Proyecto *“DetectGets”***

Curso: *Inteligencia Artifical*

Docente: Dr. Oscar J. Jimenez-Flores

Integrantes:

***Cano Sucso, Anthony Alexander 2020067573***

***Jose Luis Jarro Cachi 2020067148***

***Valverde Zamora, Jean Pier Elias 2020066920***

**Tacna – Perú**

***2024***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CONTROL DE VERSIONES | | | | | |
| Versión | Hecha por | Revisada por | Aprobada por | Fecha | Motivo |
| 1.0 | TICM | ACS | JPV | 26/06/2024 | Versión Original |

Sistema *DetectGets*

Documento de Arquitectura de Software

Versión *1.0*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CONTROL DE VERSIONES | | | | | |
| Versión | Hecha por | Revisada por | Aprobada por | Fecha | Motivo |
| 1.0 | JPEVZ | ACS y TICM | ACS y TICM | 18/05/2024 | Versión Original |

INDICE GENERAL

**Contenido**

[***1.***](#_heading=h.30j0zll) ***INTRODUCCIÓN 5***

[**1.1.**](#_heading=h.1fob9te) **Propósito (Diagrama 4+1) 5**

[**1.2.**](#_heading=h.3znysh7) **Alcance 5**

[**1.3.**](#_heading=h.2et92p0) **Definición, siglas y abreviaturas 5**

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

[***2.***](#_heading=h.3dy6vkm) ***OBJETIVOS Y RESTRICCIONES ARQUITECTONICAS 5***

[2.1.1.](#_heading=h.1ci93xb) Requerimientos Funcionales 5

[2.1.2.](#_heading=h.3whwml4) Requerimientos No Funcionales – Atributos de Calidad 5

[***3.***](#_heading=h.1t3h5sf) ***REPRESENTACIÓN DE LA ARQUITECTURA DEL SISTEMA 6***

[**3.1.**](#_heading=h.2s8eyo1) **Vista de Caso de uso 6**

[3.1.1.](#_heading=h.17dp8vu) Diagramas de Casos de uso 6

[**3.2.**](#_heading=h.2bn6wsx) **Vista Lógica 6**

[3.2.1.](#_heading=h.26in1rg) Diagrama de Subsistemas (paquetes) 7

[3.2.2.](#_heading=h.qsh70q) Diagrama de Secuencia (vista de diseño) 7

[3.2.3.](#_heading=h.3as4poj) Diagrama de Colaboración (vista de diseño) 7

[3.2.4.](#_heading=h.1pxezwc) Diagrama de Objetos 7

[3.2.5.](#_heading=h.49x2ik5) Diagrama de Clases 7

[3.2.6.](#_heading=h.2p2csry) Diagrama de Base de datos (relacional o no relacional) 7

[**3.3.**](#_heading=h.lnxbz9) **Vista de Implementación (vista de desarrollo) 7**

[3.3.1.](#_heading=h.35nkun2) Diagrama de arquitectura software (paquetes) 7

[3.3.2.](#_heading=h.1ksv4uv) Diagrama de arquitectura del sistema (Diagrama de componentes) 7

[**3.4.**](#_heading=h.147n2zr) **Vista de procesos 7**

[3.4.1.](#_heading=h.3o7alnk) Diagrama de Procesos del sistema (diagrama de actividad) 8

[**3.5.**](#_heading=h.23ckvvd) **Vista de Despliegue (vista física) 8**

[3.5.1.](#_heading=h.ihv636) Diagrama de despliegue 8

[***4.***](#_heading=h.44sinio) ***ATRIBUTOS DE CALIDAD DEL SOFTWARE 8***

[**Escenario de Funcionalidad 8**](#_heading=h.2jxsxqh)

[**Escenario de Usabilidad 8**](#_heading=h.z337ya)

[**Escenario de confiabilidad 9**](#_heading=h.3j2qqm3)

[**Escenario de rendimiento 9**](#_heading=h.1y810tw)

[**Escenario de mantenibilidad 9**](#_heading=h.4i7ojhp)

[**Otros Escenarios 9**](#_heading=h.2xcytpi)

1. INTRODUCCIÓN
   1. Propósito

El propósito es definir la arquitectura del sistema DetectGets, una aplicación móvil para verificar la originalidad de cargadores y accesorios de smartphones. Se utilizará el modelo de vistas 4+1 para representar la arquitectura.

* 1. Alcance:

El proyecto "DetectGets" comprende el desarrollo e implementación de una aplicación móvil innovadora que permita a los usuarios identificar de manera confiable si los cargadores y accesorios para sus dispositivos móviles son originales, autorizados o piratas. El alcance del proyecto incluye:

•Desarrollo de una aplicación móvil compatible con los sistemas operativos Android e iOS, brindando cobertura a la mayoría de dispositivos móviles en el mercado.

•Implementación de funcionalidades para analizar cargadores y accesorios mediante la medición de parámetros eléctricos y la comparación con valores de referencia de productos genuinos.

•Integración de tecnologías de inteligencia artificial y procesamiento de imágenes para el análisis de las especificaciones técnicas de los cargadores y accesorios.

•Creación y mantenimiento de una base de datos con información detallada de cargadores y accesorios originales y autorizados, que servirá como referencia para las comparaciones.

•Desarrollo de un sistema de autenticación de usuarios y gestión de perfiles para permitir el acceso personalizado y el seguimiento de análisis realizados.

•Implementación de medidas de seguridad y privacidad para proteger los datos de los usuarios y garantizar la confidencialidad de la información.

• Integración con servicios en la nube, como Amazon Web Services (AWS), para el procesamiento de datos, el almacenamiento y el aprovisionamiento de recursos computacionales según sea necesario.

* 1. Definición, siglas y abreviaturas:

IA: Inteligencia Artificial

Carga: Proceso de recarga de la batería de un dispositivo

Corriente eléctrica: Flujo de electricidad a través del cable de carga

* 1. Organización del documento

Diagrama

Descripción generada automáticamente

# **OBJETIVOS Y RESTRICCIONES ARQUITECTONICAS**

* 1. Priorización de requerimientos

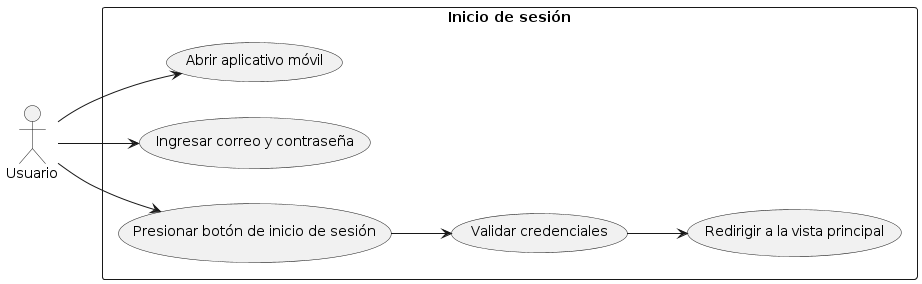
|  |  |  |
| --- | --- | --- |
| **ID** | **Requerimiento Funcional** | **Descripción** |
| **RF01** | Autenticación de usuarios | La aplicación debe permitir a los usuarios registrarse y autenticarse. |
| **RF02** | Gestión de perfiles de usuario | Los usuarios deben poder crear, editar y eliminar su perfil. |
| **RF03** | Conexión de dispositivos móviles | La aplicación debe permitir a los usuarios conectar y gestionar sus dispositivos móviles o escritorio. |
| **RF04** | Análisis de cargadores y accesorios | La aplicación debe analizar los cargadores y accesorios conectados mediante parámetros eléctricos y especificaciones técnicas. |
| **RF05** | Comparación de parámetros con valores de referencia | La aplicación debe comparar los parámetros medidos con una base de datos de valores de referencia de productos genuinos. |
| **RF06** | Generación de reportes de análisis | La aplicación debe generar reportes detallados de los análisis realizados, indicando si el producto es original, autorizado o pirata. |
| **RF07** | Almacenamiento y gestión de reportes | Los usuarios deben poder almacenar, ver y eliminar los reportes de análisis realizados. |
| **RF08** | Recomendaciones de productos compatibles | La aplicación debe proporcionar recomendaciones de productos compatibles y seguros para el dispositivo del usuario. |
| **RF09** | Configuración de notificaciones | Los usuarios deben poder configurar y recibir notificaciones relacionadas con la aplicación. |
| **RF10** | Modo oscuro | La aplicación debe ofrecer la opción de activar un modo oscuro para una mejor experiencia visual. |
| **RF11** | Cerrar sesión | Los usuarios deben poder cerrar sesión de manera segura en la aplicación. |

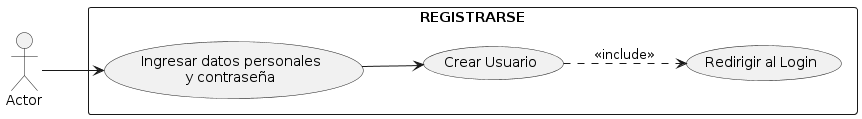
|  |  |  |
| --- | --- | --- |
| **ID** | **Requerimiento No Funcional** | **Descripción** |
| **RNF01** | Interfaz de usuario intuitiva y atractiva | La interfaz de usuario debe ser intuitiva, fácil de usar y visualmente atractiva. |
| **RNF02** | Rendimiento y eficiencia | La aplicación debe ser rápida y eficiente en el uso de recursos del dispositivo. |
| **RNF03** | Seguridad y privacidad de datos | La aplicación debe garantizar la seguridad y privacidad de los datos de los usuarios. |
| **RNF04** | Escalabilidad | La arquitectura de la aplicación debe ser escalable para soportar crecimiento futuro. |
| **RNF05** | Integración con servicios en la nube | La aplicación debe integrarse con servicios en la nube como AWS. |
| **RNF06** | Actualizaciones remotas | La aplicación debe permitir actualizaciones remotas de funciones y seguridad. |

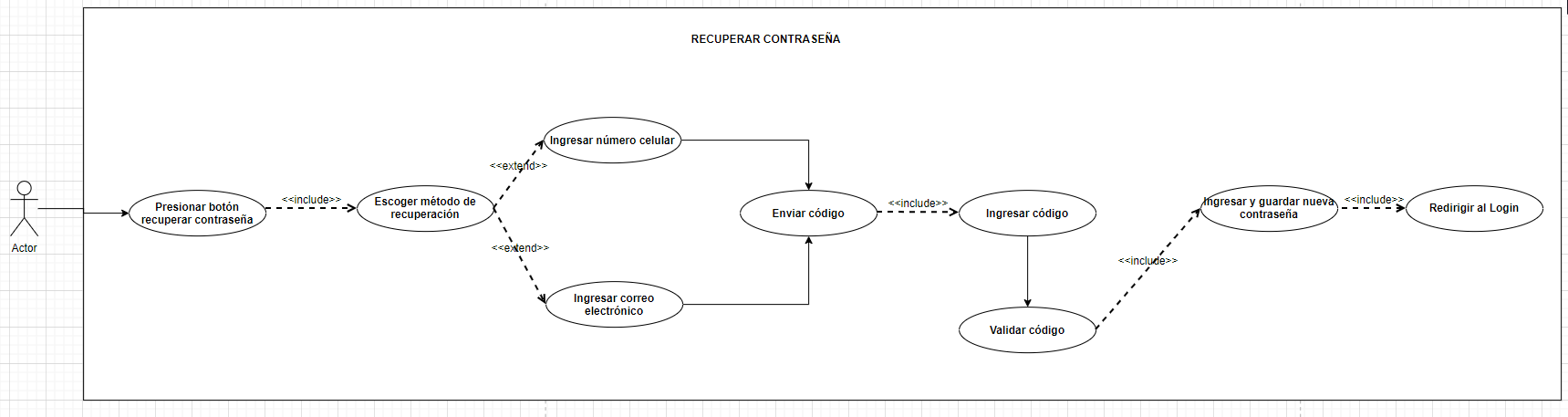
# **REPRESENTACIÓN DE LA ARQUITECTURA DEL SISTEMA**

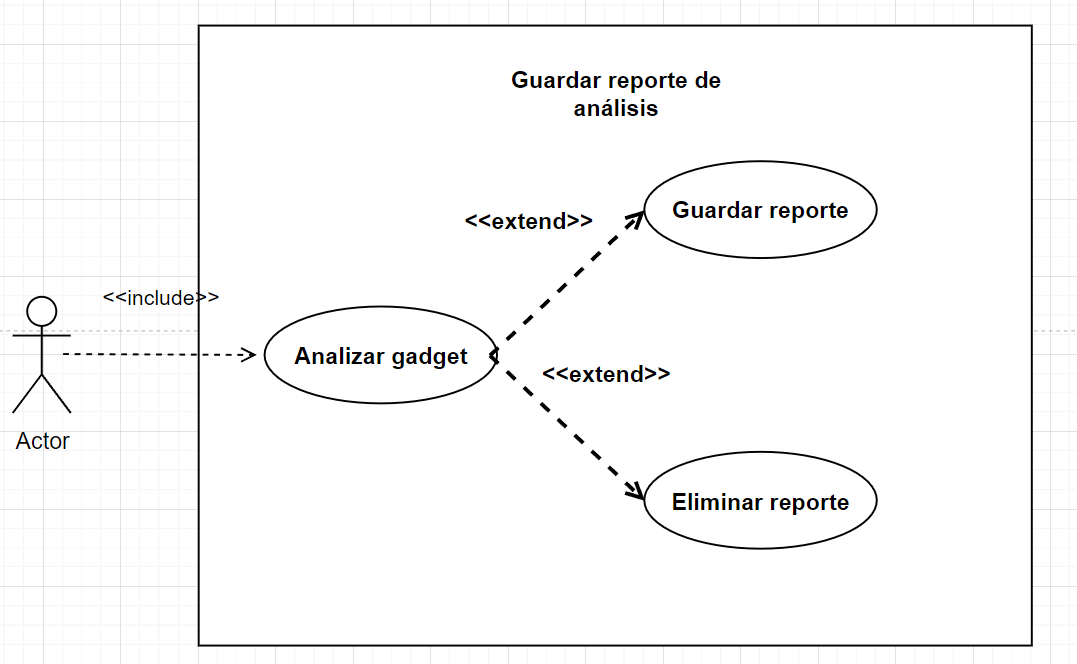
* 1. Vista de Caso de uso

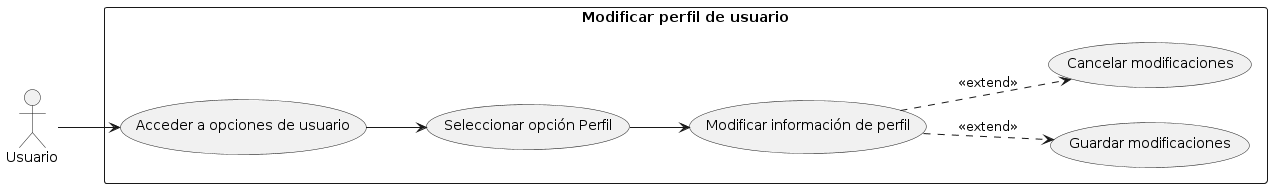
### Diagramas de Casos de uso

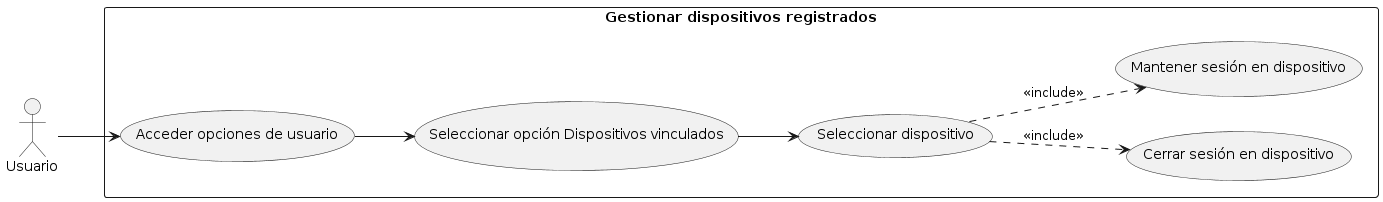


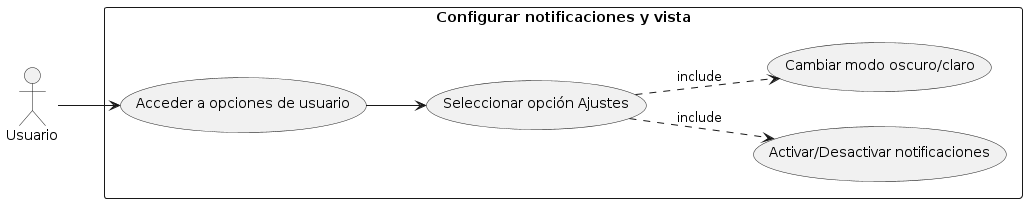


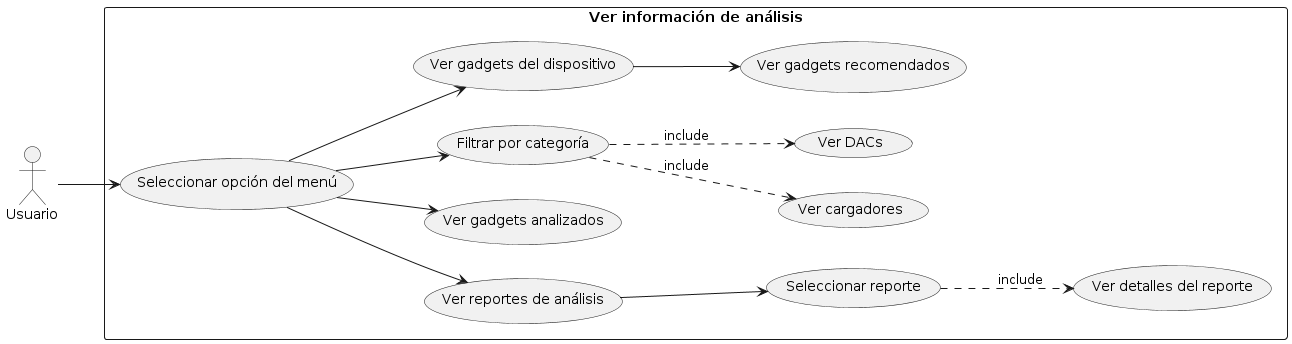










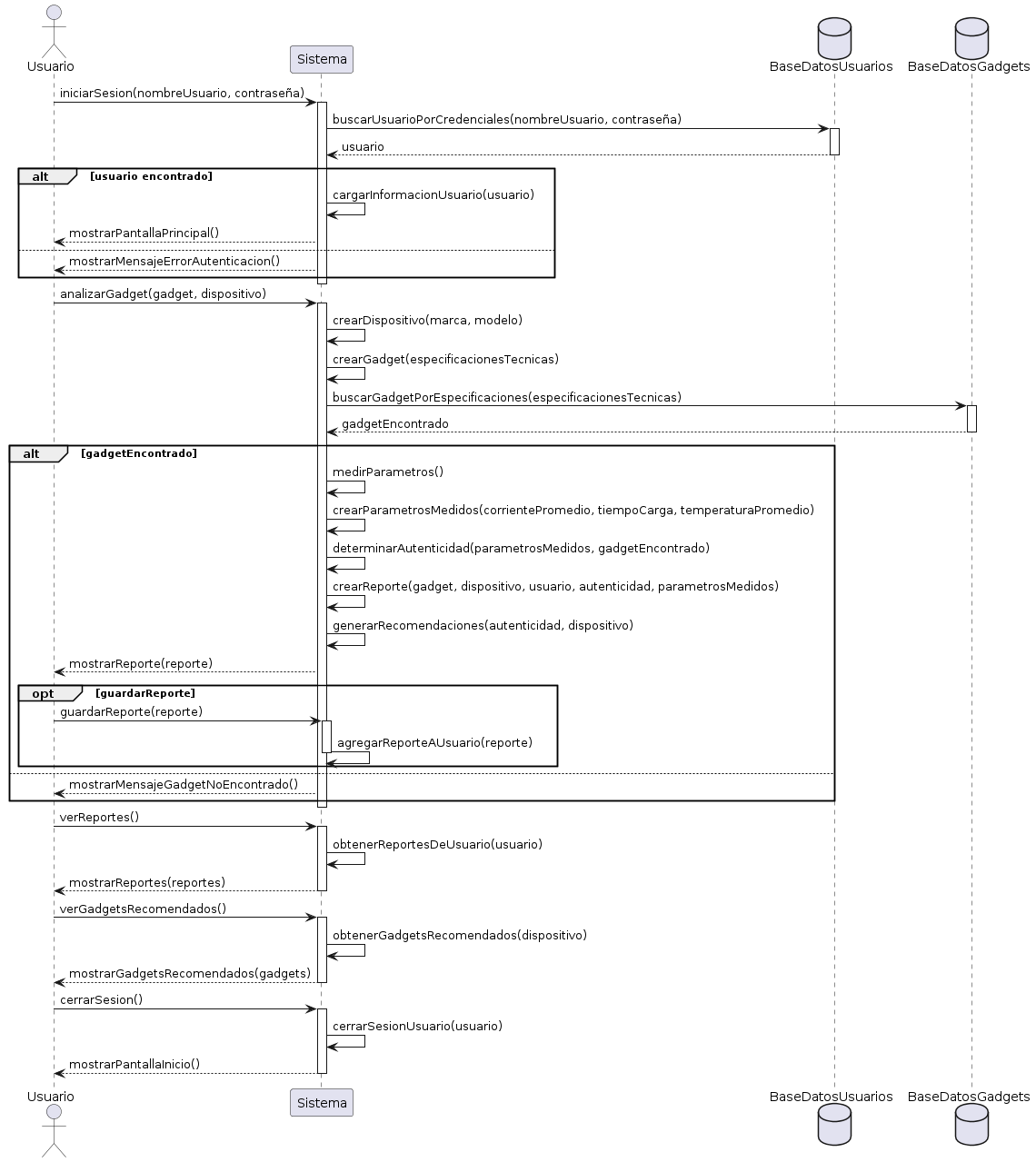


### 

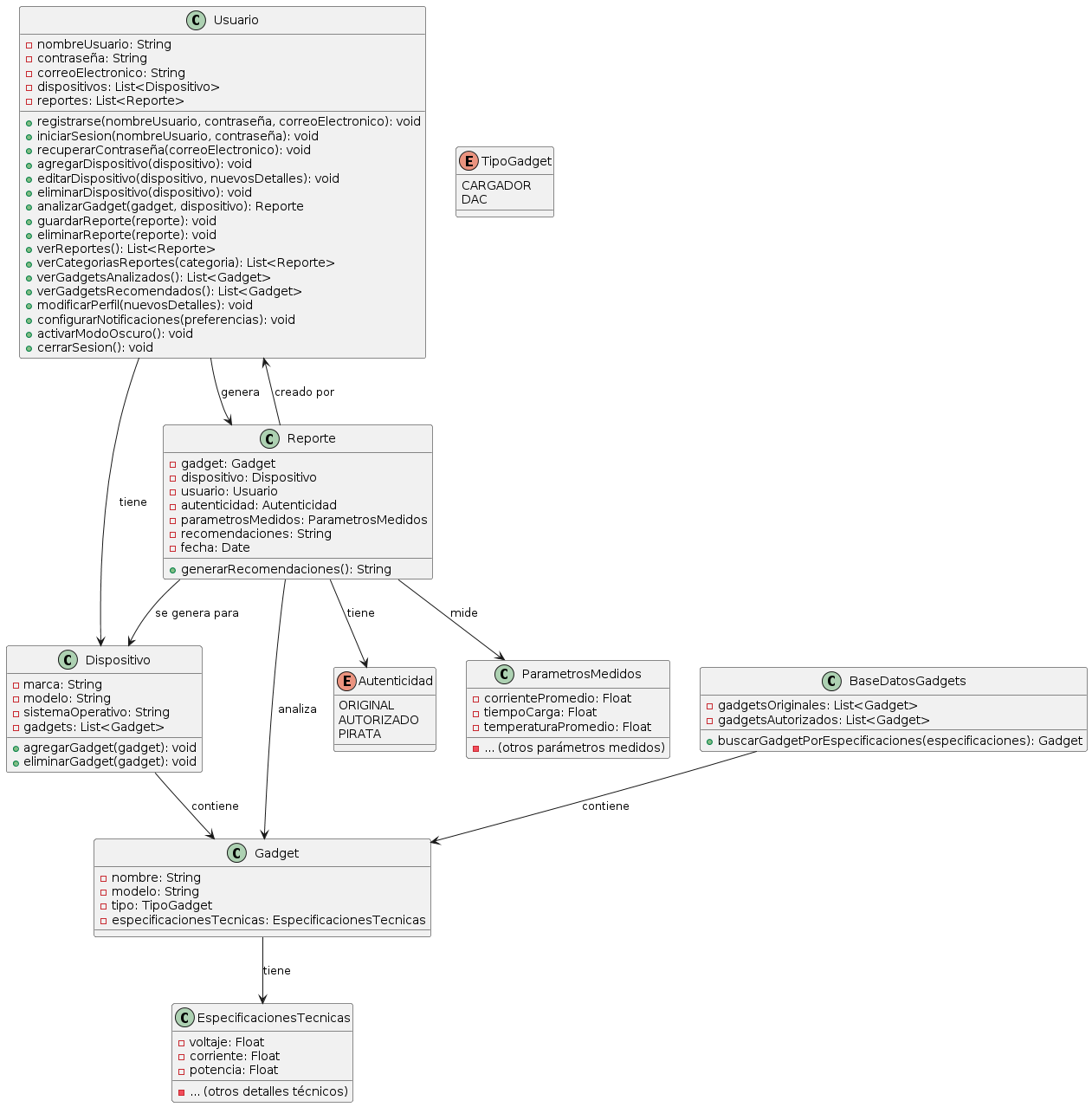
### Diagrama de Subsistemas (paquetes)

* 1. Vista de Implementación

### Diagrama de secuencia:



### Diagrama de clases:



* + 1. Diagrama de base de datos relacional:

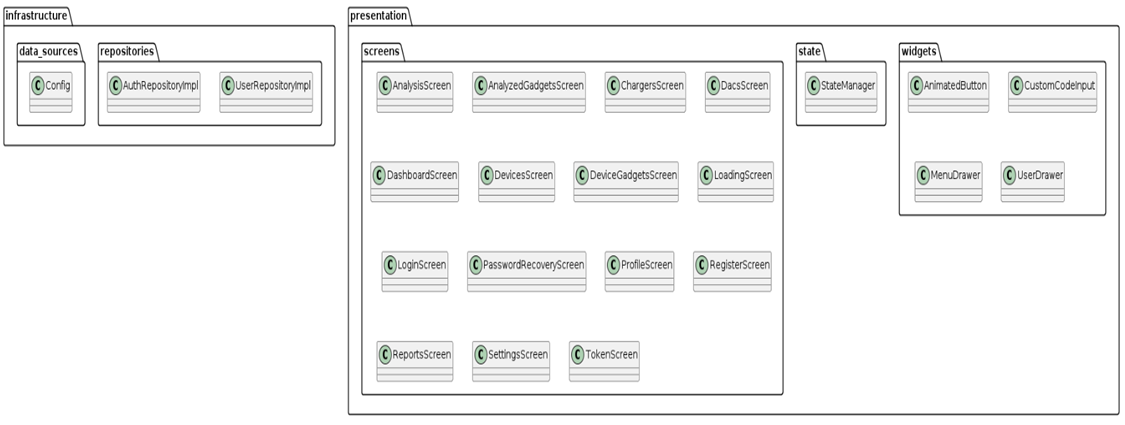
Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

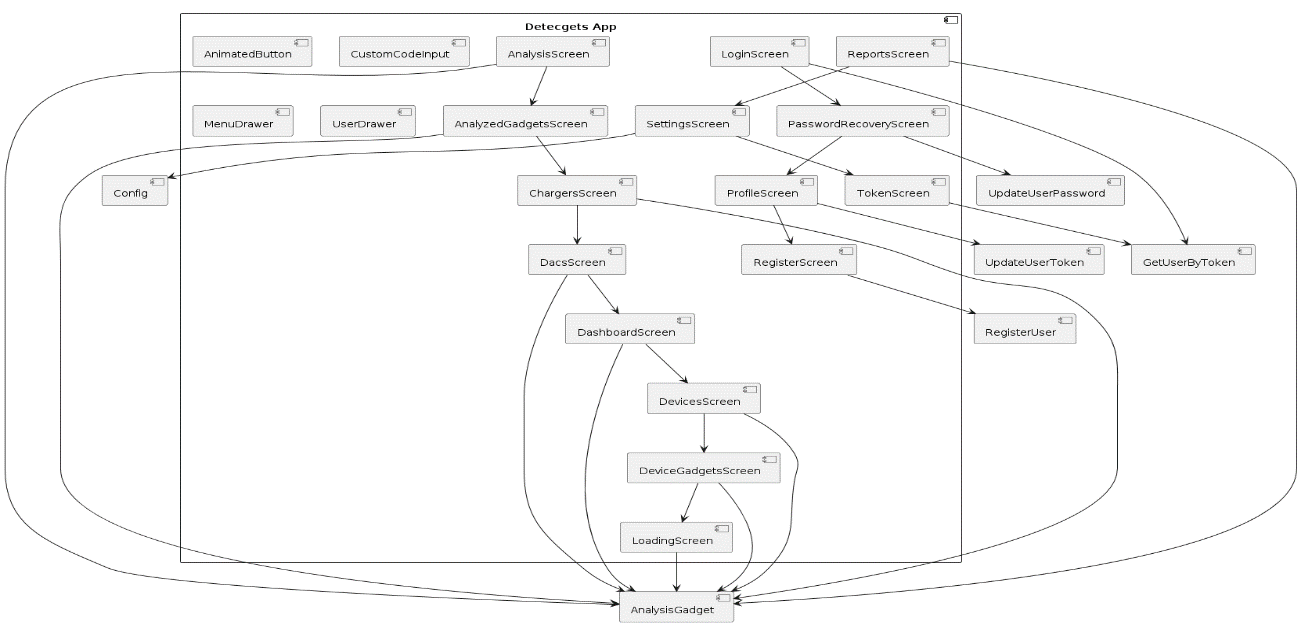
* 1. Vista de implementación:
     1. Diagrama de arquitectura de software:Interfaz de usuario gráfica, Aplicación

        Descripción generada automáticamente





* + 1. Diagrama de arquitectura del sistema:
    2. Diagrama de componentesDiagrama

       Descripción generada automáticamente

Lógica de negocio y repositorios:  
Diagrama

Descripción generada automáticamente

Servicios externos e integracions:  
Diagrama

Descripción generada automáticamente

Nodos principales:

* Dispositivo Móvil del Usuario
  + Aplicación DetectGets
  + Base de Datos Local
* Servidor en la Nube
  + Base de Datos Central
  + Servicio de Autenticación
* Servicio de Notificaciones Push

# **ATRIBUTOS DE CALIDAD DEL SOFTWARE**

**Escenario de Funcionalidad**

DetectGets debe ser capaz de detectar y clasificar correctamente al menos el 95% de los accesorios conectados.

La aplicación debe soportar una base de datos de al menos 1000 modelos de accesorios diferentes

**Escenario de Usabilidad**

Los usuarios deben ser capaces de realizar una verificación de accesorio en menos de 30 segundos desde la conexión.

La interfaz debe ser intuitiva, permitiendo a nuevos usuarios entender los resultados sin necesidad de tutorial.

**Escenario de confiabilidad**

La aplicación debe funcionar sin errores críticos durante al menos 99.9% del tiempo de uso.

Los datos de usuario deben estar encriptados tanto en almacenamiento local como en tránsito.

**Escenario de rendimiento**

El proceso de verificación no debe tomar más de 5 segundos desde la conexión del accesorio hasta mostrar el resultado.

La aplicación no debe consumir más del 5% de la batería del dispositivo en un día de uso normal.

**Escenario de mantenibilidad**

La base de datos de accesorios debe poder actualizarse remotamente sin necesidad de actualizar la aplicación completa.

El código debe estar modularizado para permitir la fácil adición de nuevos tipos de accesorios a verificar en el futuro

**Otros Escenarios**

La aplicación debe ser capaz de manejar al menos 100 verificaciones de accesorios por día sin degradación del rendimiento.

El tiempo de inicio de la aplicación no debe exceder los 2 segundos en dispositivos de gama media.