**UNIVERSIDAD PRIVADA DE TACNA**

**FACULTAD DE INGENIERÍA**

**Escuela Profesional de Ingeniería de Sistemas**

****

**“Proyecto AccessGuard - Sistema de control de acceso a las instalaciones físicas de la UPT aplicando tecnología de respuesta de validación por voz”**

Curso: ***Soluciones Móviles I***

Docente: ***Ing. Elard Rodríguez Marca***

Integrantes:

**Arce Bracamonte, Sebastian Rodrigo (2019092986)**

**Chávez Linares, César Fabián (2019063854)**

**Hernández Cruz, Angel Gadiel (2021070017)**

**Tacna – Perú**

***2025***

| CONTROL DE VERSIONES | | | | | |
| --- | --- | --- | --- | --- | --- |
| Versión | Hecha por | Revisada por | Aprobada por | Fecha | Motivo |
| 1.0 | CCL | SAB | AHC | 16/06/2025 | Versión Original |

**Sistema móvil de control de seguridad para la entrada y salida de alumnos**

**Documento de Arquitectura de Software**

**Versión 1.0**

| CONTROL DE VERSIONES | | | | | |
| --- | --- | --- | --- | --- | --- |
| Versión | Hecha por | Revisada por | Aprobada por | Fecha | Motivo |
| 1.0 | CCL | SAB | AHC | 16/06/2025 | Versión Original |

ÍNDICE GENERAL

**Contenido**

[***1. INTRODUCCIÓN 3***](#_heading=h.ftkn8h5x02wj)

[**1.1. Propósito (Diagrama 4+1) 3**](#_heading=h.qgq33uelk45c)

[**1.2. Alcance 4**](#_heading=h.fv7cf2bwmm44)

[**1.3. Definición, siglas y abreviaturas 4**](#_heading=h.j9q2wxjksdvk)

[**1.4. Organización del documento 4**](#_heading=h.65jp8r4hj5ae)

[***2. OBJETIVOS Y RESTRICCIONES***](#_heading=h.x2h0ku1z3f38) ***ARQUITECTÓNICAS 4***

[2.1.1. Requerimientos Funcionales 5](#_heading=h.cor6fut6a8hr)

[2.1.2. Requerimientos No Funcionales – Atributos de Calidad 5](#_heading=h.x8ofwnkh3eof)

[***3. REPRESENTACIÓN DE LA ARQUITECTURA DEL SISTEMA 6***](#_heading=h.wl5fabegp2z9)

[**3.1. Vista de Caso de uso 6**](#_heading=h.ootsd8oe79hb)

[3.1.1. Diagramas de Casos de uso 6](#_heading=h.v8f5hpj4bh5d)

[**3.2. Vista Lógica 6**](#_heading=h.lxkmynww86dg)

[3.2.1. Diagrama de Subsistemas (paquetes) 6](#_heading=h.e10jfc8rhmsi)

[3.2.2. Diagrama de Secuencia (vista de diseño) 6](#_heading=h.ymteuum9s10s)

[3.2.3. Diagrama de Colaboración (vista de diseño) 6](#_heading=h.2jhgklhkc0kd)

[3.2.4. Diagrama de Objetos 6](#_heading=h.zbq57jitywb1)

[3.2.5. Diagrama de Clases 7](#_heading=h.qrwpfg5oe1k1)

[3.2.6. Diagrama de Base de datos (relacional o no relacional) 7](#_heading=h.66xltkxrh32n)

[**3.3. Vista de Implementación (vista de desarrollo) 7**](#_heading=h.m5om2xprzqm8)

[3.3.1. Diagrama de arquitectura software (paquetes) 7](#_heading=h.eg9p8mzh1n9v)

[3.3.2. Diagrama de arquitectura del sistema (Diagrama de componentes) 7](#_heading=h.w3za8g516e4o)

[**3.4. Vista de procesos 7**](#_heading=h.55tcl3ncp087)

[3.4.1. Diagrama de Procesos del sistema (diagrama de actividad) 7](#_heading=h.s2loyausio3d)

[**3.5. Vista de Despliegue (vista física) 7**](#_heading=h.t60xxhu710k3)

[3.5.1. Diagrama de despliegue 7](#_heading=h.hfw3dblkb80t)

[***4. ATRIBUTOS DE CALIDAD DEL SOFTWARE 8***](#_heading=h.v8zpwuhqv2yr)

[**Escenario de Funcionalidad 8**](#_heading=h.ex9l1r42pply)

[**Escenario de Usabilidad 8**](#_heading=h.wewulw52wt5c)

[**Escenario de confiabilidad 8**](#_heading=h.9ymcizuznpzu)

[**Escenario de rendimiento 8**](#_heading=h.t8j72r1dgy7t)

[**Escenario de mantenibilidad 8**](#_heading=h.8k8lqlix0bh1)

[**Otros Escenarios 8**](#_heading=h.or6dynbs60zt)

* INTRODUCCIÓN

En la actualidad, el control del acceso de estudiantes, docentes y visitantes a las instalaciones universitarias suele gestionarse de manera manual o con sistemas poco integrados, lo que genera ineficiencias, riesgos de seguridad y dificultades para mantener registros precisos. En un entorno dinámico como el de la Universidad Privada de Tacna, la falta de un sistema automatizado de acceso no solo afecta la seguridad del campus, sino que también limita la capacidad de la institución para gestionar y supervisar el flujo de personas en tiempo real.

Este proyecto propone la implementación de un Sistema de Control y Validación de Accesos, diseñado para digitalizar y centralizar la gestión de ingresos y salidas de todos los usuarios del campus universitario, incluyendo estudiantes, docentes, personal administrativo y visitantes. El sistema permitirá registrar, verificar y supervisar cada acceso, utilizando tecnologías como códigos de barras y una aplicación móvil para mejorar la eficiencia y seguridad en el proceso.

El sistema estará integrado con una base de datos centralizada que garantizará la trazabilidad completa de cada acceso, permitiendo tanto el monitoreo en tiempo real como la generación de reportes detallados para la administración universitaria. Se incorporarán funcionalidades clave como la validación automática de credenciales, notificaciones instantáneas y un módulo para consultas históricas de accesos.

Con esta solución, se espera fortalecer considerablemente la seguridad del campus, optimizar los recursos humanos encargados de la vigilancia y brindar a la comunidad universitaria una experiencia más ágil y confiable al momento de ingresar a las instalaciones. En última instancia, el sistema contribuirá a la creación de un entorno más seguro y mejor organizado para todos los miembros de la universidad.

* + Propósito (Diagrama 4+1)

El propósito del Diagrama 4+1 en el Sistema de Control de Acceso Universitario mediante Aplicativo Móvil es proporcionar una visión integral de la arquitectura del sistema desde múltiples perspectivas, que permitan comprender su estructura, comportamiento y lógica de operación. Este enfoque aborda tanto los requisitos funcionales (como el registro de accesos, validación mediante códigos de barras o DNI, y notificaciones en tiempo real) como los no funcionales (seguridad, escalabilidad, usabilidad y disponibilidad).

La arquitectura modular propuesta facilita el mantenimiento y evolución del sistema, permitiendo gestionar distintos tipos de usuarios (estudiantes, personal de seguridad, administrativos) y adaptarse a nuevas normativas de control. Además, se prioriza una experiencia de usuario fluida a través de una interfaz móvil intuitiva y rápida. Las decisiones de diseño consideran la eficiencia operativa, la integración con servicios en la nube y el cumplimiento de estándares de seguridad, garantizando un acceso seguro, confiable y en tiempo real a las instalaciones universitarias.

* + Alcance

La aplicación web diseñada será una herramienta integral para manejar amplios segmentos del proceso de desarrollo de software, proporcionando un recurso vital para los profesionales en la gestión de configuración de software. Su objetivo es proporcionar un conjunto de herramientas para mejorar la coordinación y la administración de las actividades de gestión de configuración de software. Dentro del alcance del proyecto, se incluirán los siguientes objetivos:

* Desarrollo de app móvil (Flutter) para estudiantes, docentes, visitantes y personal de seguridad.
* Panel web administrativo con dashboards e informes de accesos.
* Validación mediante código de barras, con alertas y notificaciones.
* Registro histórico y reglas de control por horarios, rutas y roles.
* Soporte para roles diferenciados (administrador general, administrador de facultad, guardias, visitantes).
  + Definición, siglas y abreviaturas
  + Organización del documento

El documento se ha estructurado empleando la representación de la arquitectura de acuerdo con la arquitectura de 4 + “1” vistas propuestas por IBM Rational. La representación se realizará a fin de mostrar diferentes perspectivas del producto software, empleando las vistas siguientes:

**PERSPECTIVA FUNCIONAL**

Vista de Casos de Usos:

Presenta la arquitectura desde la perspectiva del usuario final. Esta vista se desarrolla a través del Modelo de Casos de Usos (usando Diagramas de Casos de Uso de UML).

**PERSPECTIVA ESTRUCTURAL**

Vista Lógica:

Presenta la arquitectura desde la perspectiva del desarrollador. Permite mostrar la organización de las piezas fundamentales de la arquitectura, organizando los elementos de diseño (clases, tablas, etc.).

**PERSPECTIVA DE CONSTRUCCIÓN**

Vista de Implementación:

Presenta la arquitectura desde la perspectiva del programador, definiendo los componentes software a ser desarrollados, la distribución de las clases, tablas y demás.

**PERSPECTIVA DINÁMICA**

Vista de los Procesos / Tareas:

Presenta la arquitectura desde la perspectiva del desarrollador a fin de definir aspectos de concurrencia, comunicación interprocesos, sincronizaciones, etc.

**PERSPECTIVA DE LOS DATOS**

Vista de Datos

Presenta la arquitectura de datos que soportará los requerimientos de información del sistema software. Se emplea el Modelo Entidad Relación.

**PERSPECTIVA DEL DESPLIEGUE**

Vista de Despliegue:

Presenta la arquitectura desde la perspectiva del implantador de la solución. Define como los componentes de la arquitectura serán desplegados sobre la infraestructura de TI definida.

* **OBJETIVOS Y RESTRICCIONES ARQUITECTÓNICAS**
  + Priorización de requerimientos
    1. Requerimientos Funcionales

| **ID** | **Descripción** | **Prioridad** |
| --- | --- | --- |
| RF-01 | Inicio de sesión por rol | Alta |
| RF-02 | Redirección de vistas por tipo de usuario | Alta |
| RF-03 | Dashboard de registros de ingreso/salida | Alta |
| RF-04 | Filtro de registros por fecha | Media |
| RF-05 | Visualización de ingresos de externos | Alta |
| RF-06 | Gestión de usuarios (guardias) | Alta |
| RF-07 | Asignación de facultades a usuarios | Media |
| RF-08 | Escaneo de código de barras | Alta |
| RF-09 | Reproducción de voz con información del estudiante | Alta |
| RF-10 | Registro manual de visitantes externos | Alta |
| RF-11 | Vista a facultad receptora de externos | Alta |
| RF-12 | Registro de todos los accesos | Alta |
| RF-13 | Vista de permanencia luego del horario | Alta |
| RF-14 | Registro de entrada y salida del estacionamiento | Alta |
| RF-15 | Validación al ingreso del estacionamiento | Alta |

* + 1. Requerimientos No Funcionales – Atributos de Calidad

| **ID** | **Requerimiento** | **Prioridad** |
| --- | --- | --- |
| RNF-01 | Plataforma móvil exclusiva para guardias | Alta |
| RNF-02 | Registro local optimizado | Alta |
| RNF-03 | Disponibilidad continua | Alta |
| RNF-04 | Interfaz rápida e intuitiva | Alta |
| RNF-05 | Soporte para escaneo y reproducción de voz | Alta |

* + Restricciones

1. Alcance:

El SGCS se centrará exclusivamente en la gestión de la configuración de software, excluyendo cualquier funcionalidad no directamente relacionada con este propósito.

1. Tiempo:

El desarrollo e implementación del SGCS deberán completarse dentro de un plazo máximo de 4 meses, limitando la capacidad de incorporar características adicionales fuera del alcance inicial.

1. Costo:

La inversión total para el desarrollo del SGCS no debe exceder los S/. 15,000, restringiendo posibles expansiones o mejoras que sobrepasen este presupuesto.

1. Calidad:

El SGCS debe cumplir con todos los estándares de calidad establecidos por el equipo de desarrollo y los interesados, incluyendo pruebas exhaustivas y revisiones de código.

1. Recursos:

El equipo de desarrollo tendrá un máximo de 5 integrantes, limitando el trabajo a la vez.

1. Riesgos:

Factores como el aumento de costos, retrasos en las entregas, mala planificación y errores de cálculo en los recursos pueden impactar negativamente el proyecto.

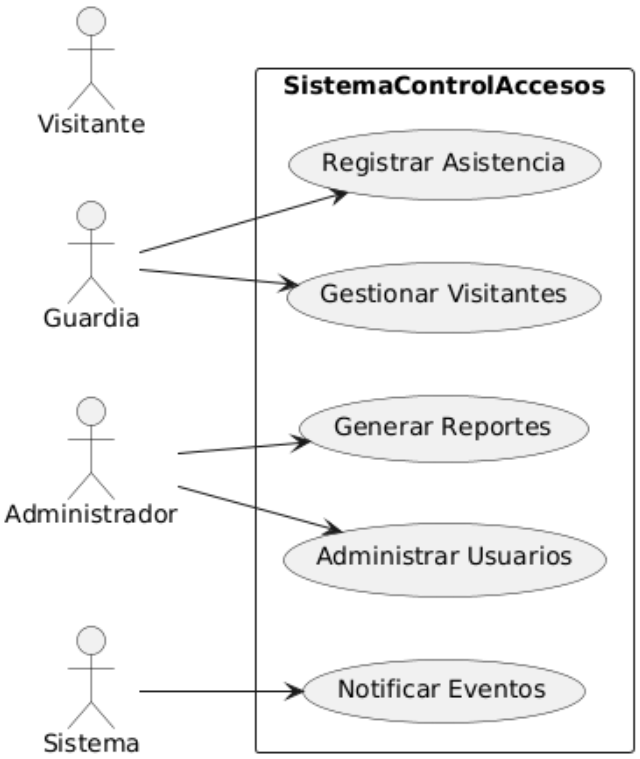
1. Satisfacción al cliente:

El éxito del proyecto se mide por la satisfacción del cliente, siendo imperativo cumplir con todas las expectativas establecidas en la fase inicial de requerimientos.

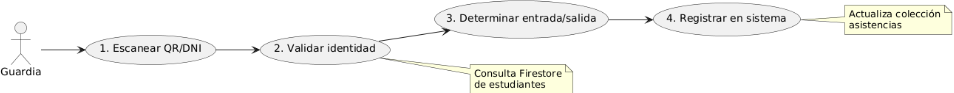
* **REPRESENTACIÓN DE LA ARQUITECTURA DEL SISTEMA**

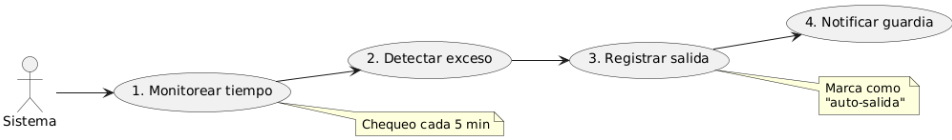
* + Vista de Caso de uso

Diagramas de Casos de uso

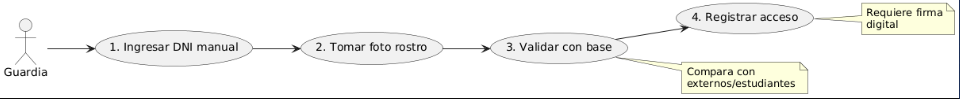


1. Registro de Accesos

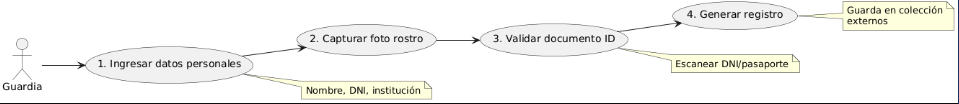
1.1 Registrar asistencia (Estudiante/Guardia)  


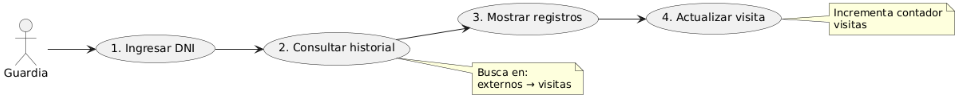
1.2 Registrar salida automática  


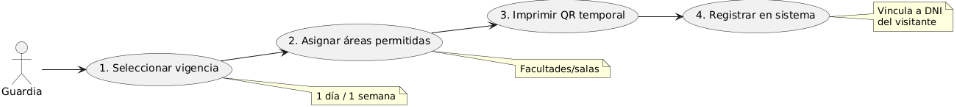
1.3 Forzar registro manual



2. Gestión de Visitantes

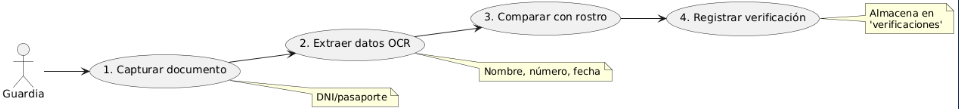
2.1 Registrar nuevo visitante  


2.2 Buscar visitante recurrente  


2.3 Generar pase temporal  


3. Verificación

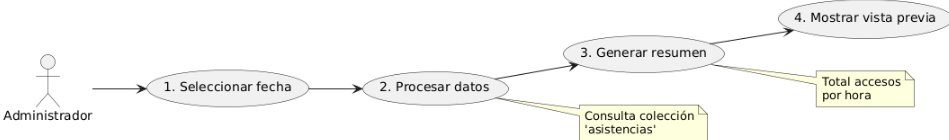
3.1 Validar credencial estudiante  


3.2 Escanear documento identidad  


3.3 Verificar estado de cuenta

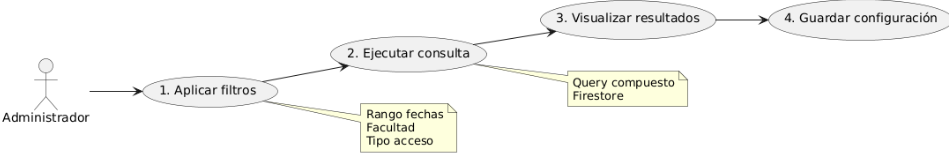


4. Reportes

4.1 Generar reporte diario  


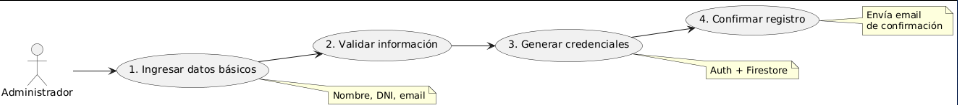
4.2 Exportar datos estadísticos  


4.3 Filtrar por facultad/tiempo

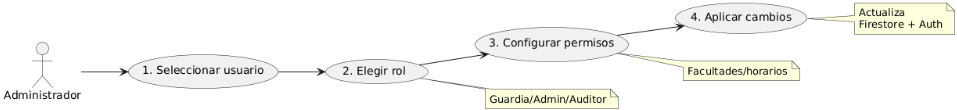


5. Administración

5.1 Crear/modificar usuario



5.2 Asignar roles

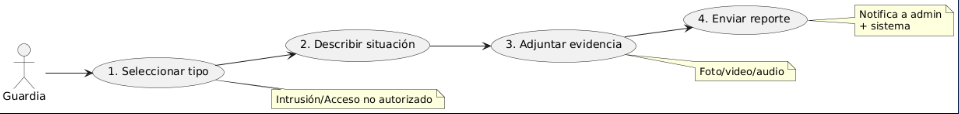


5.3 Desactivar cuenta

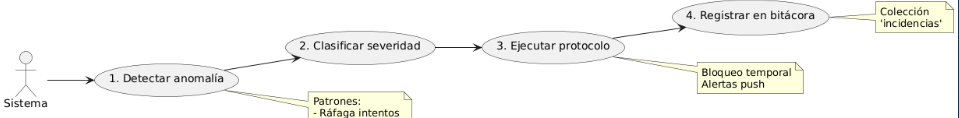


6. Seguridad

6.1 Registrar incidencia manual

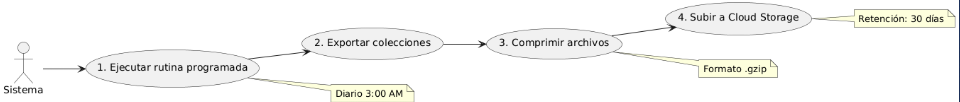


6.2 Notificar intento fallido

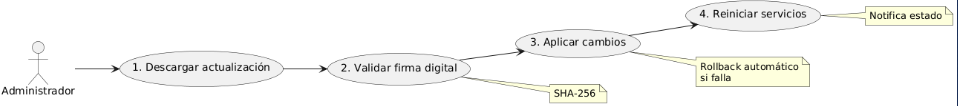


7. Sistema

7.1 Respaldar datos



7.2 Actualizar configuración



* + Vista Lógica

* + - Diagrama de Subsistemas (paquetes)

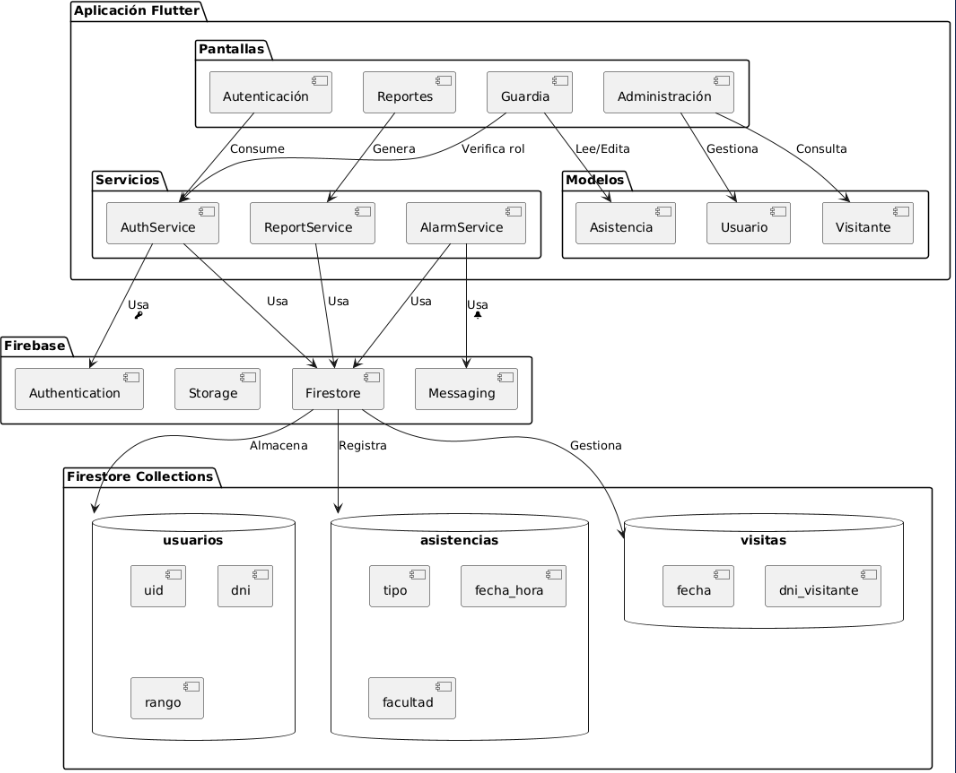
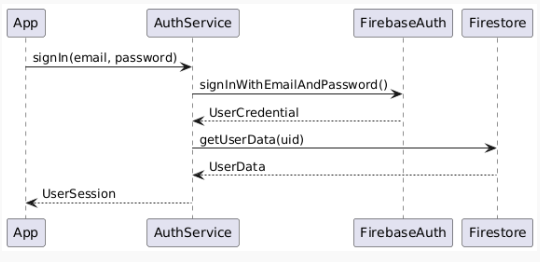
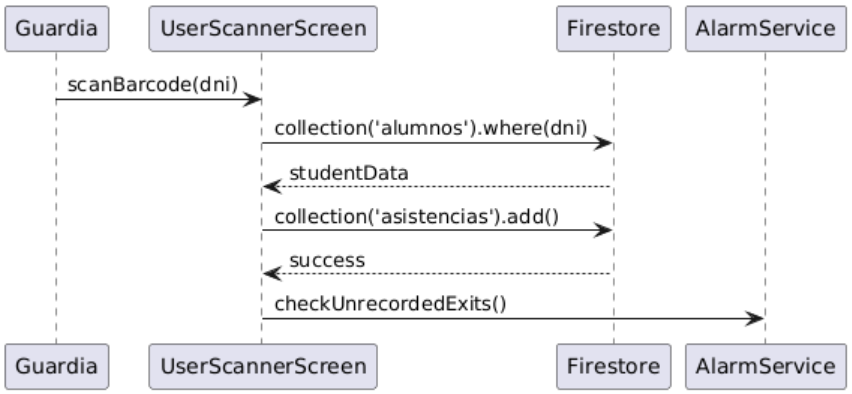


Diagrama de Secuencia (vista de diseño)

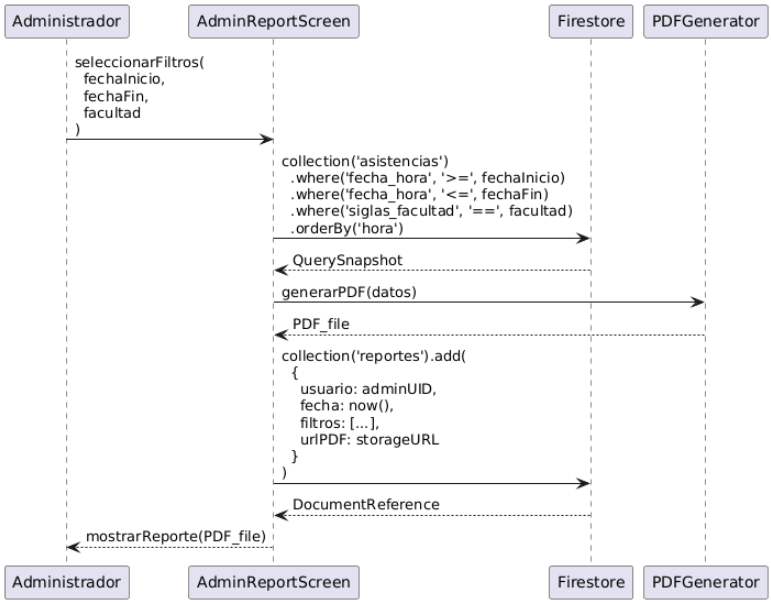
Secuencia\_Autenticacion



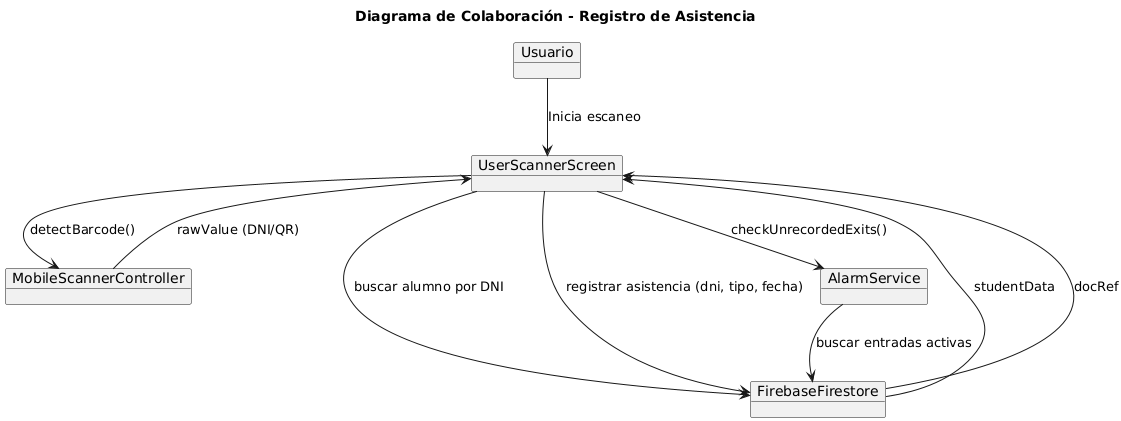
Secuencia\_RegistroAsistencia



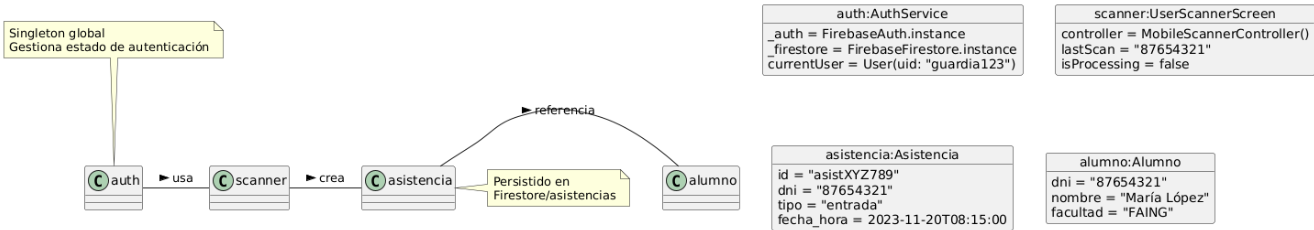
Secuencia\_Reportes



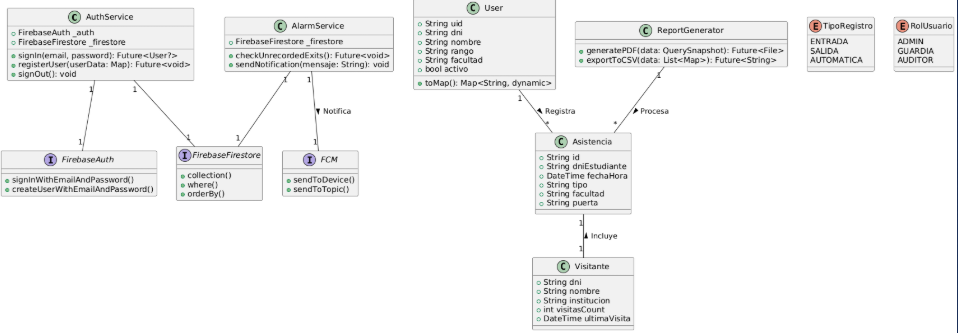
* + - Diagrama de Colaboración (vista de diseño)



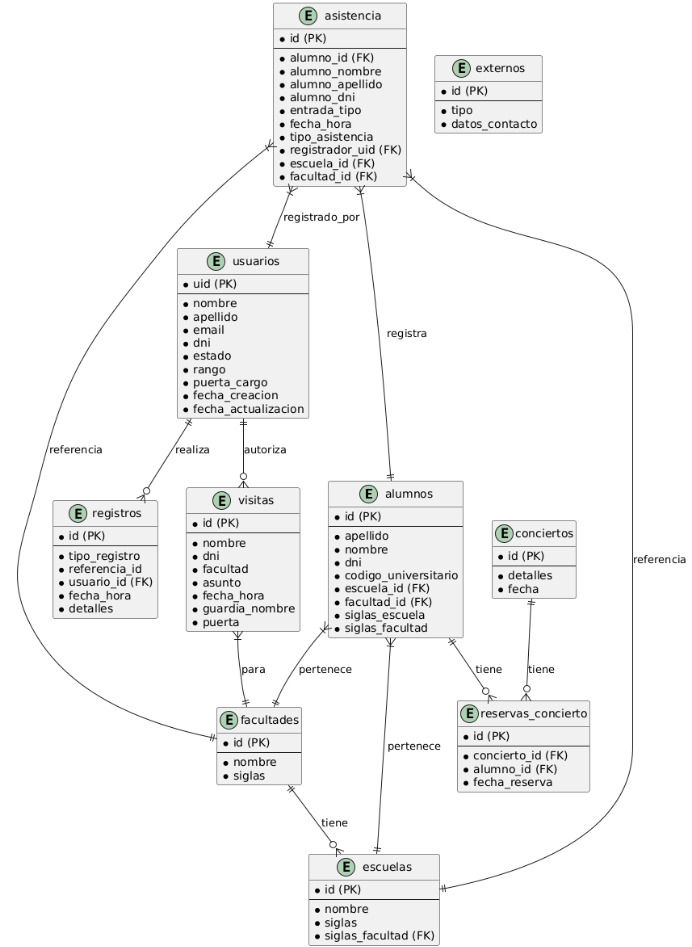
* + - Diagrama de Objetos



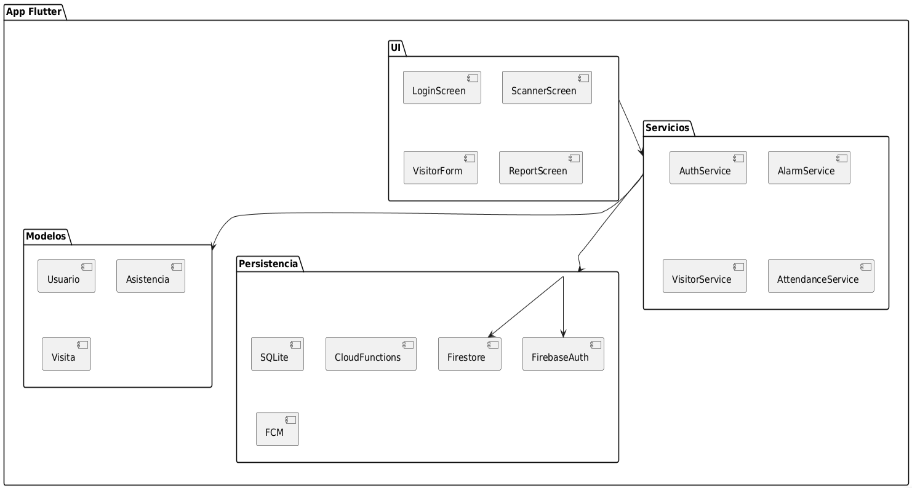
* + - Diagrama de Clases



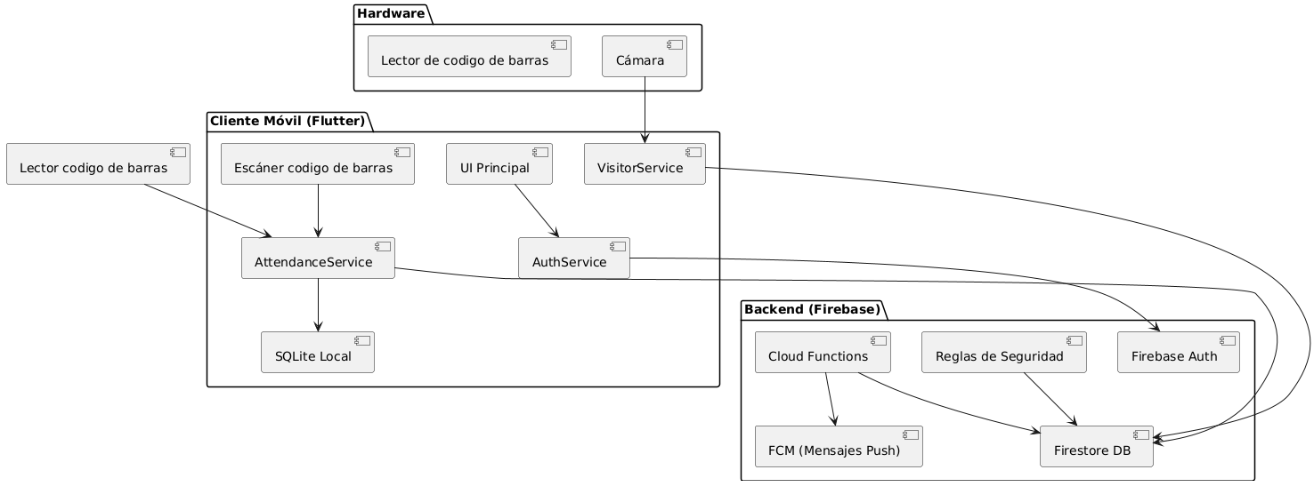
* + - Diagrama de Base de datos



* + Vista de Implementación (vista de desarrollo)
    - Diagrama de arquitectura software (paquetes)



* + - Diagrama de arquitectura del sistema (Diagrama de componentes)



El sistema AccessGuard implementa una arquitectura basada en el patrón MVC tradicional, enriquecida con capas de servicios especializados y repositorios para gestionar la complejidad del control de accesos universitarios. Esta estructura organiza los componentes en niveles bien definidos:

**1. Capa de Presentación (Vistas)**

* **App Móvil (Flutter):**
  + Interfaz para guardias con escáner de código de barras integrado
  + Formularios dinámicos para registro de visitantes
  + Visualización de alertas en tiempo real
* **Panel Web Administrativo:**
  + Dashboards interactivos con filtros temporales
  + Módulos de gestión de usuarios y facultades
  + Generador de reportes exportables

**2. Controladores (Gestión de Peticiones)**

* **Controladores Especializados:**
  + **AccessController:** Gestiona validación de credenciales (DNI/CODIGO UNIVERSITARIO) y flujo de accesos
  + **VisitorController:** Administra el registro manual de visitantes externos
  + **AlertController:** Coordina las notificaciones a facultades y alertas horarias
  + **ReportController:** Procesa solicitudes de generación de reportes

**3. Servicios de Dominio (Lógica de Negocio)**

* **Módulos Clave:**
  + **ValidationService:** Implementa la validación de códigos de barras y generación de respuestas vocales
  + **AccessLogService:** Registra y audita todos los movimientos en el sistema
  + **NotificationService:** Gestiona el envío de alertas push y notificaciones **cruzadas**
  + **UserManagementService:** Administra la asignación de roles y facultades

**4. Capa de Persistencia**

* **Repositorios Especializados:**
  + **UserRepository:** Operaciones CRUD para usuarios y guardias
  + **AccessRepository:** Consultas de registros históricos de accesos
  + **VisitorRepository:** Gestión de datos de visitantes externos
* **Modelos de Datos:**
  + Entidades principales: Usuario, Estudiante, Visitante, Acceso, Facultad
  + Relaciones bien definidas (1-N, N-M) para la trazabilidad

**5. Componentes de Soporte**

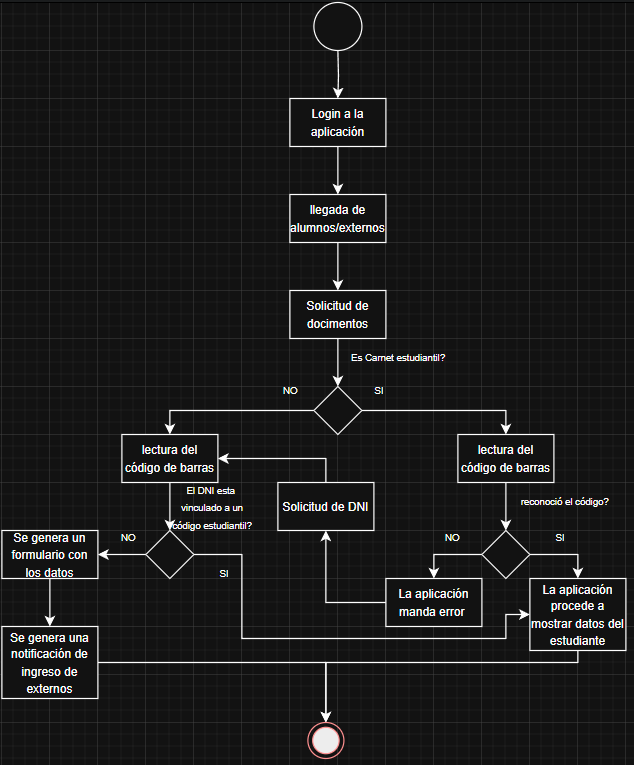
* **Motor de Notificaciones: Sistema independiente para alertas push y correos**
* **API Gateway: Punto único de entrada para todas las solicitudes**
* **Servicio de Voz: Conversión texto-voz para anuncios de validación**

**Flujo de Interacción Típico:**

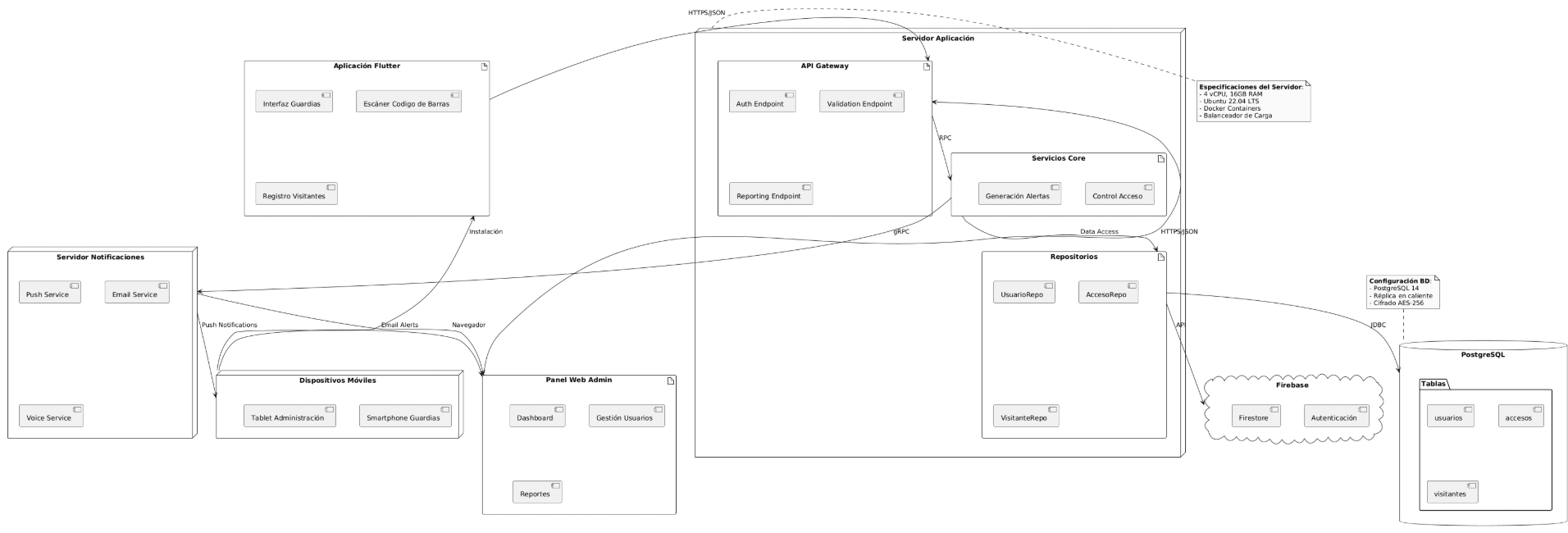
1. Un guardia escanea un código de barras mediante la app móvil
2. La vista envía la solicitud al AccessController
3. El controlador invoca el ValidationService
4. El servicio consulta el UserRepository para validar credenciales
5. El repositorio accede a la base de datos centralizada
6. El servicio genera una respuesta vocal con los datos del estudiante
7. El controlador registra el acceso mediante el AccessLogService
8. La vista muestra el resultado y reproduce el anuncio vocal

**Ventajas de la Arquitectura:**

* Separación clara de responsabilidades entre componentes
* Servicios reutilizables para lógica compleja (validaciones, notificaciones)
* Persistencia aislada en repositorios especializados
* Escalabilidad mediante microservicios potenciales (ej: módulo de voz independiente)
* Consistencia en la aplicación de reglas de negocio (ej: políticas de acceso)
  + Vista de procesos
    - Diagrama de Procesos del sistema (diagrama de actividad)



* + Vista de Despliegue (vista física)
    - Diagrama de despliegue



Los diagramas de Distribución, y su objetivo es mostrar cómo están dispuestas las particiones físicas del sistema en construcción y cómo se asignan los componentes de software a estas particiones. Representan las relaciones físicas entre los componentes de software y hardware en el sistema que se integrará.

Un diagrama de Despliegue ilustra cómo y dónde se implementará el sistema. Las máquinas físicas y los procesadores se representan como nodos, y la estructura interna puede representarse mediante nodos o artefactos incrustados. A medida que los artefactos se colocan en los nodos para modelar el despliegue del sistema, la ubicación se guía por las especificaciones de despliegue.

* **ATRIBUTOS DE CALIDAD DEL SOFTWARE**

**Escenario de Funcionalidad:**

* Fuente del estímulo: Guardia de seguridad
* Estímulo: Escanear el código de barras del carnet universitario del estudiante
* Entorno: Entrada principal de la universidad
* Artefacto: Módulo de autenticación y validación de acceso
* Respuesta: El sistema verifica el código y muestra los datos del estudiante con confirmación de acceso
* Medida de la respuesta: Validación y despliegue de datos en menos de 2 segundos
* Atributo de calidad afectado: Funcionalidad

**Escenario de Usabilidad:**

* Fuente del estímulo: Guardia nuevo o con poca experiencia
* Estímulo: Registrar manualmente a un visitante sin código de barras
* Entorno: Turno de vigilancia en facultades
* Artefacto: Interfaz móvil del módulo de registro manual
* Respuesta: El guardia completa el formulario de registro sin ayuda externa
* Medida de la respuesta: Registro completado en menos de 1 minuto
* Atributo de calidad afectado: Usabilidad

**Escenario de Rendimiento:**

* Fuente del estímulo: Usuario administrador
* Estímulo: Consultar los ingresos y salidas de toda una semana académica
* Entorno: Panel administrativo en horario pico
* Artefacto: Módulo web de reportes y dashboard
* Respuesta: Los registros se cargan y visualizan correctamente
* Medida de la respuesta: No debe superar los 5 segundos de carga
* Atributo de calidad afectado: Rendimiento

**Escenario de Mantenibilidad:**

* Fuente del estímulo: Desarrollador del sistema
* Estímulo: Actualizar la lógica de validación para nuevos formatos de código de barras
* Entorno: Entorno de desarrollo controlado
* Artefacto: Código fuente del módulo de escaneo
* Respuesta: Se aplica el cambio sin afectar las demás funcionalidades
* Medida de la respuesta: El cambio se realiza y se prueba en menos de 2 horas
* Atributo de calidad afectado: Mantenibilidad

**Escenario de Seguridad:**

* Fuente del estímulo: Usuario no autorizado
* Estímulo: Intentar ingresar al sistema como guardia sin credenciales válidas
* Entorno: Aplicación móvil en un dispositivo externo
* Artefacto: Módulo de inicio de sesión
* Respuesta: El acceso es denegado y se registra el intento
* Medida de la respuesta: Bloqueo inmediato con registro del evento en el sistema
* Atributo de calidad afectado: Seguridad