



UNIVERSIDAD PRIVADA DE TACNA

FACULTAD DE INGENIERÍA

Escuela Profesional de Ingeniería de Sistemas

**Plataforma Móvil de Exámenes de Simulacro de Exámenes
de Certificación de Inglés “Simulapp”**

Curso: Patrones de Software

Docente: Mag. Patrick Cuadros Quiroga

Integrantes:

<i>Cabrera Catari, Camila Fernanda</i>	<i>(2021069824)</i>
<i>Melendez Huarachi, Gabriel Fari</i>	<i>(2021070311)</i>

Tacna – Perú
2025



CONTROL DE VERSIONES					
Versión	Hecha por	Revisada por	Aprobada por	Fecha	Motivo
1.0	CCC, GMH	CCC, GMH	CCC, GMH	25/08/2025	Versión Original

Plataforma Móvil de Exámenes de Simulacro de Exámenes de Certificación de Inglés “Simulapp”

Documento de Arquitectura de Software

Versión {1.0}

CONTROL DE VERSIONES					
Versión	Hecha por	Revisada por	Aprobada por	Fecha	Motivo
1.0	CCC, GMH	CCC, GMH	CCC, GMH	25/08/2025	Versión Original

ÍNDICE GENERAL

1. INTRODUCCIÓN	5
1.1. Propósito (Diagrama 4+1)	5
1.2. Alcance	5
1.3. Definición, siglas y abreviaturas	5
1.4. Organización del documento	5
2. OBJETIVOS Y RESTRICCIONES ARQUITECTÓNICAS	5
2.1.1. Requerimientos Funcionales	5
2.1.2. Requerimientos No Funcionales – Atributos de Calidad	5
3. REPRESENTACIÓN DE LA ARQUITECTURA DEL SISTEMA	6
3.1. Vista de Caso de uso	6
3.1.1. Diagramas de Casos de uso	6
3.2. Vista Lógica	6
3.2.1. Diagrama de Subsistemas (paquetes)	7
3.2.2. Diagrama de Secuencia (vista de diseño)	7
3.2.3. Diagrama de Colaboración (vista de diseño)	7
3.2.4. Diagrama de Objetos	7
3.2.5. Diagrama de Clases	7
3.2.6. Diagrama de Base de datos (relacional o no relacional)	7
3.3. Vista de Implementación (vista de desarrollo)	7
3.3.1. Diagrama de arquitectura software (paquetes)	7
3.3.2. Diagrama de arquitectura del sistema (Diagrama de componentes)	7
3.4. Vista de procesos	7
3.4.1. Diagrama de Procesos del sistema (diagrama de actividad)	8
3.5. Vista de Despliegue (vista física)	8
3.5.1. Diagrama de despliegue	8
4. ATRIBUTOS DE CALIDAD DEL SOFTWARE	8
Escenario de Funcionalidad	8
Escenario de Usabilidad	8
Escenario de confiabilidad	9
Escenario de rendimiento	9
Escenario de mantenibilidad	9

1. INTRODUCCIÓN

1.1. Propósito (Diagrama 4+1)

El propósito del sistema "Simulapp" es ofrecer a los usuarios una plataforma intuitiva y accesible que les permita prepararse de manera efectiva para los exámenes de certificación de inglés, además de facilitarles la búsqueda de instituciones certificadoras, comparar precios y visualizar fechas de exámenes. El sistema se desarrollará utilizando el marco de trabajo "4+1" para la visualización de la arquitectura, que incluye las siguientes vistas: vista lógica, vista de desarrollo, vista de procesos, vista de despliegue y vista de casos de uso. Esta metodología permite entender y comunicar las diferentes perspectivas del sistema, desde su diseño hasta su implementación, garantizando su efectividad y facilidad de mantenimiento.

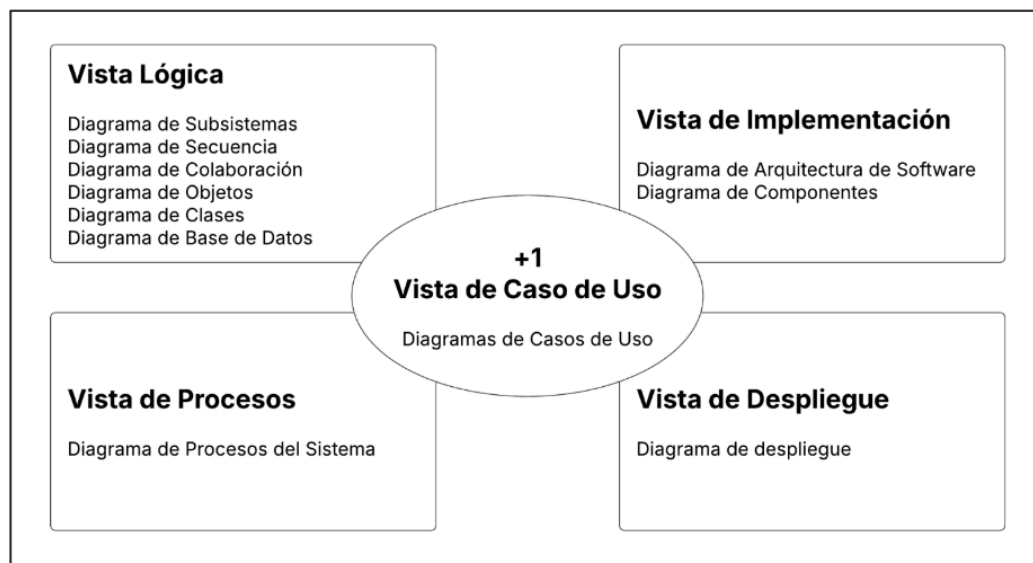


Figura 01. Diagrama 4+1 de la aplicación "simulapp".

Fuente: Elaboración propia.

1.2. Alcance

El proyecto se centra en el desarrollo de una aplicación móvil para dispositivos Android e iOS, orientada a personas interesadas en obtener una certificación de inglés. La aplicación permitirá a los usuarios:

- Practicar secciones específicas de exámenes de certificación, como listening y reading.
- Consultar una lista de institutos de certificación con sus ubicaciones y características.
- Visualizar un calendario con las fechas de exámenes.
- Comparar precios de exámenes entre diversos institutos.

Esta aplicación será utilizada principalmente por individuos que desean mejorar sus habilidades y medir su nivel de inglés en preparación para obtener certificaciones reconocidas.

1.3. Definición, siglas y abreviaturas

- UI (User Interface): Interfaz de Usuario, el espacio donde el usuario interactúa con el sistema.
- Flutter: Framework de desarrollo para crear aplicaciones móviles multiplataforma.
- DB (Database): Base de Datos, sistema estructurado para almacenar y gestionar información.
- SAD (Software Architecture Document): Documento que describe la arquitectura del software, sus componentes y cómo interactúan entre sí.
- API (Application Programming Interface): Interfaz de Programación de Aplicaciones, un conjunto de definiciones y protocolos que permiten que diferentes aplicaciones se comuniquen entre sí.

1.4. Organización del documento

Este documento está estructurado en varias secciones que cubren de manera integral todos los aspectos esenciales relacionados con el desarrollo y el diseño de la arquitectura del sistema. Cada sección ha sido elaborada con el propósito de garantizar que el sistema propuesto no solo cumpla con los requerimientos funcionales, sino que también asegure calidad, sostenibilidad y escalabilidad en el tiempo.

● Introducción

En esta primera sección se establece el contexto del proyecto, definiendo el propósito general del sistema, su alcance y las definiciones clave necesarias para comprender la propuesta arquitectónica. Se busca sentar una base sólida que permita al lector entender de manera clara los objetivos del sistema y su relevancia en el entorno donde será implementado. Además, se presentan los términos técnicos y conceptos fundamentales, con el fin de unificar criterios y evitar ambigüedades en la interpretación del documento.

● Objetivos y Restricciones Arquitectónicas

En esta parte se detallan los objetivos principales que guían el desarrollo de la solución tecnológica, alineados con las necesidades de los usuarios y los requerimientos del negocio. Asimismo, se explican las restricciones que condicionan las decisiones arquitectónicas, tales como limitaciones de hardware, software, presupuesto o tiempo de desarrollo. Se establecen también las prioridades estratégicas, como la necesidad de escalabilidad, seguridad de los datos o la interoperabilidad con otros sistemas, que servirán como criterios clave al momento de validar las decisiones de diseño.

● Representación de la Arquitectura del Sistema

Esta sección constituye el núcleo del documento, ya que describe cómo se estructura el sistema desde diferentes perspectivas. Se incluyen representaciones gráficas y textuales que abarcan la vista de casos de uso (interacción de los actores con el sistema), la vista lógica (organización de los componentes y su relación), la vista de implementación (cómo se materializa en el código y los módulos), la vista de procesos (flujo de operaciones y

conurrencia), y la vista de despliegue (infraestructura física o virtual donde el sistema será alojado). Cada una de estas vistas se explica en detalle con el objetivo de proporcionar una comprensión holística del funcionamiento del sistema.

- **Atributos de Calidad del Software**

Finalmente, se abordan los atributos de calidad que garantizarán el éxito y la aceptación del sistema a largo plazo. Se plantean escenarios específicos que evalúan dimensiones como la funcionalidad (capacidad del sistema para cumplir con los requerimientos establecidos), la usabilidad (facilidad de uso e interacción por parte de los usuarios finales), la confiabilidad (capacidad de operar sin fallas en diferentes contextos), el rendimiento (eficiencia en el manejo de recursos y tiempos de respuesta), y la mantenibilidad (facilidad de adaptación y mejora a futuro). Estos atributos no solo definen estándares de calidad, sino que también sirven como métricas para medir la efectividad del diseño arquitectónico.

2. OBJETIVOS Y RESTRICCIONES ARQUITECTÓNICAS

2.1. Priorización de requerimientos

A continuación, se presentan los requerimientos funcionales y no funcionales priorizados, lo que permitirá establecer un orden lógico de implementación de las funcionalidades y atributos de calidad del sistema.

1.1.1. Requerimientos Funcionales

Código	Requerimiento	Descripción
RF-01	Registrar usuario	Permitir a los usuarios crear una cuenta mediante un formulario de registro.
RF-02	Iniciar sesión	Facilitar el acceso a los usuarios mediante un sistema de inicio de sesión.
RF-03	Practicar exámenes	Permitir a los usuarios practicar secciones específicas de los exámenes (listening, reading).
RF-04	Consultar institutos	Mostrar una lista de institutos de certificación, incluyendo ubicación y características.
RF-05	Visualizar fechas de exámenes	Proporcionar un calendario con las fechas de los exámenes disponibles.
RF-06	Comparar precios	Permitir a los usuarios comparar precios de exámenes entre diferentes institutos.
RF-07	Acceder a recursos de preparación	Proveer materiales y recursos para la preparación de los exámenes.

1.1.2. Requerimientos No Funcionales – Atributos de Calidad

Código	Requerimiento	Descripción
RNF-01	Interfaz amigable	La aplicación debe tener una UI intuitiva y fácil de navegar para mejorar la experiencia del usuario.
RNF-02	Seguridad	Implementar medidas de seguridad para proteger la información personal y las credenciales de los usuarios.
RNF-03	Rendimiento	La aplicación debe cargar rápidamente y ser responsiva para una buena experiencia de uso.
RNF-04	Compatibilidad	Debe ser compatible con dispositivos Android.
RNF-05	Escalabilidad	La arquitectura debe permitir la adición de nuevas funcionalidades sin afectar el rendimiento.

2.2. Restricciones

- Presupuesto limitado de S/.3960
- Tiempo de desarrollo limitado a 4 meses.
- Necesidad de cumplir con las regulaciones de privacidad de datos.

3. REPRESENTACIÓN DE LA ARQUITECTURA DEL SISTEMA

3.1. Vista de Caso de uso

1.1.3. Diagramas de Casos de uso

El diagrama de casos de uso muestra las interacciones entre el actor y las funcionalidades del sistema.

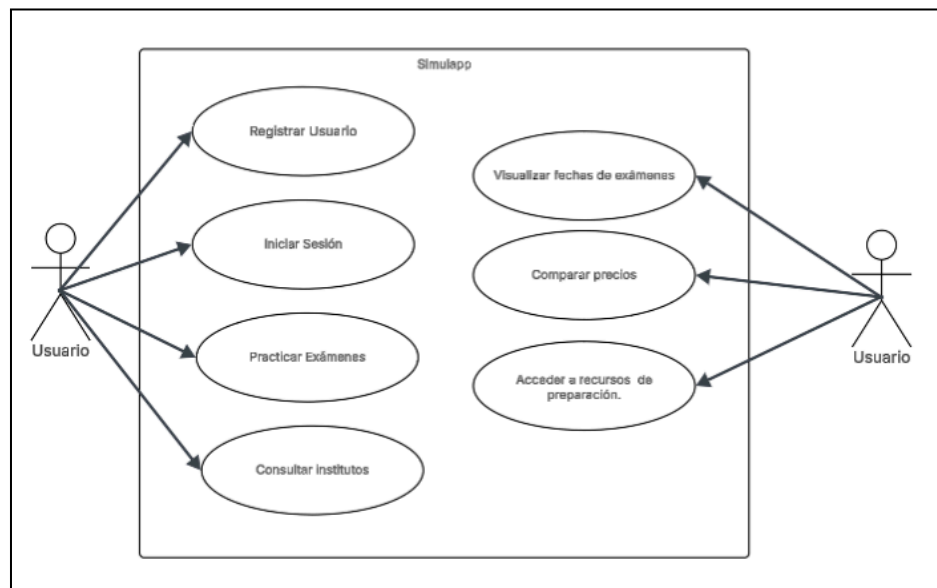


Figura 02. Diagrama de casos de uso de la aplicación "simulapp".

Fuente: Elaboración propia.

3.2. Vista Lógica

3.2.1. Diagrama de Subsistemas (paquetes)

El diagrama de paquetes muestra la organización modular de la aplicación SimulApp, representando los subsistemas principales y su relación entre ellos. Cada paquete agrupa un conjunto de componentes con responsabilidades similares, facilitando la comprensión de la arquitectura general y la gestión del código.

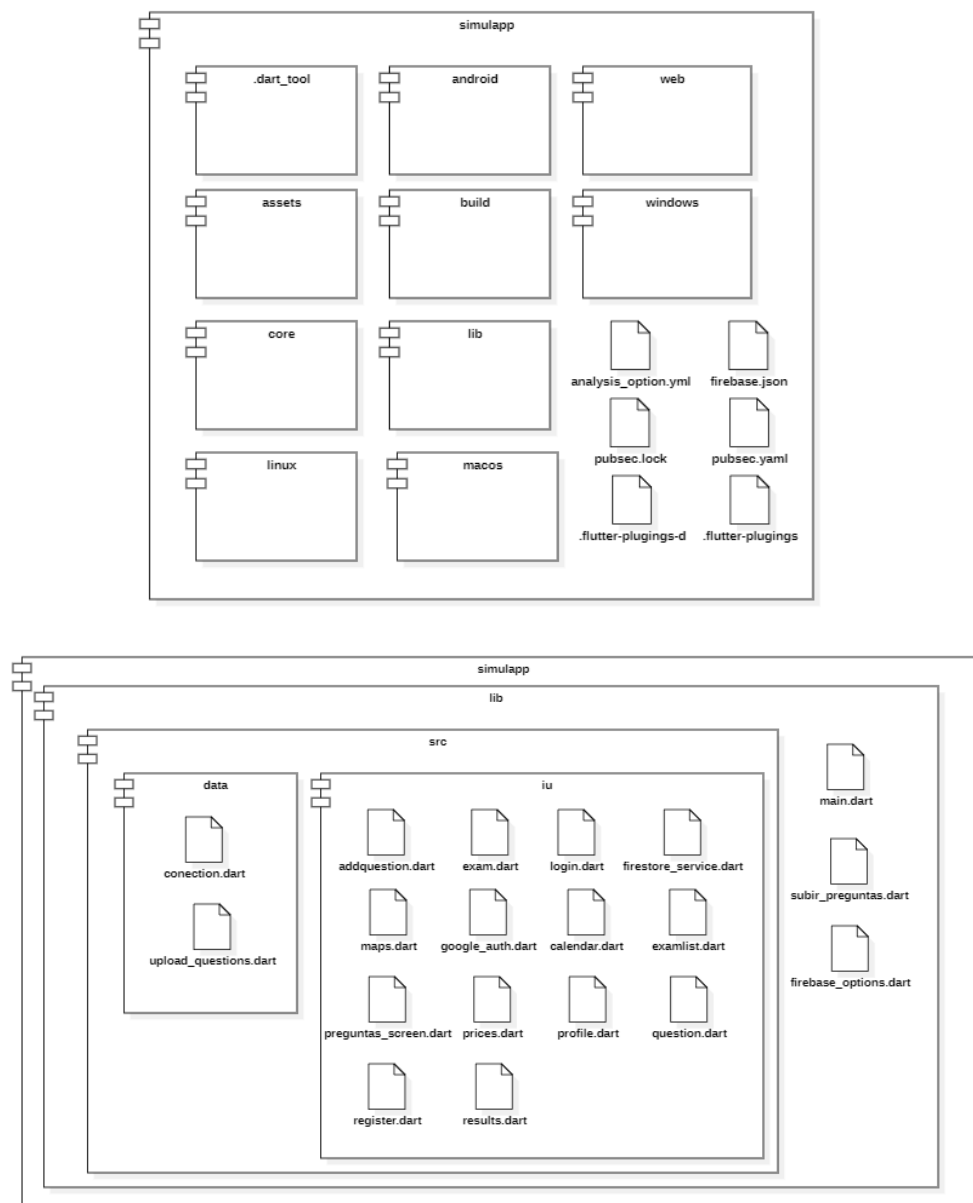


Figura 03. Diagrama de paquetes de la aplicación "simulapp".

Fuente: Elaboración propia.

3.2.2. Diagrama de Secuencia (vista de diseño).

Diagrama de Secuencia CU-01 “Registrar usuario”

El diagrama de secuencia del caso de uso “Registrar usuario” ilustra la interacción temporal entre el usuario, la aplicación y los servicios de backend durante el proceso de registro.

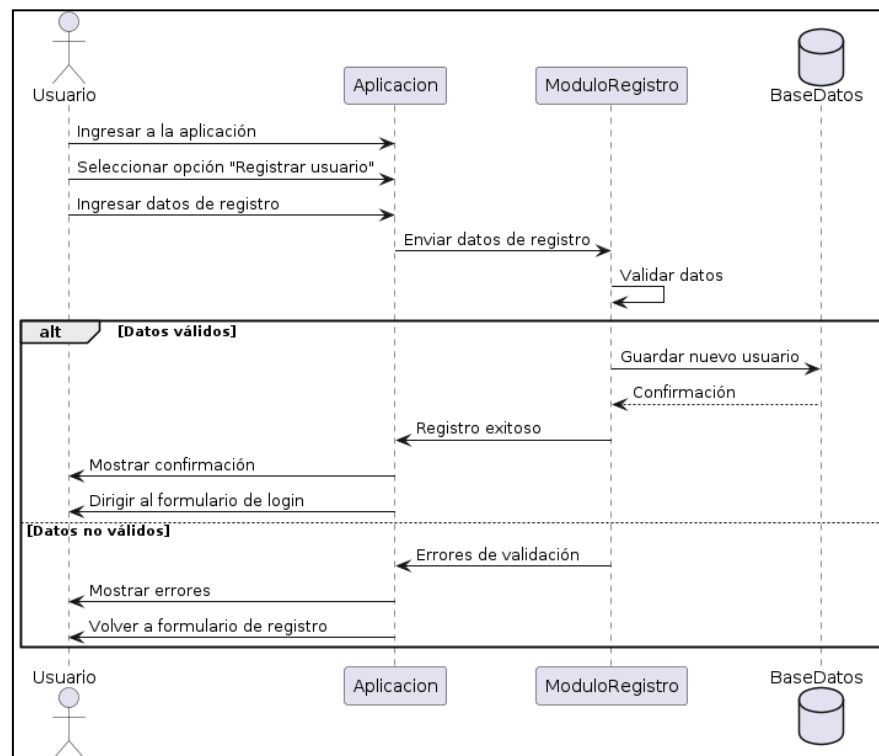


Figura 04. Diagrama de secuencia del caso de uso 01.

Fuente: Elaboración propia.

Diagrama de Secuencia CU-02 “Iniciar sesión”

El diagrama de secuencia “Iniciar sesión” muestra la interacción entre el usuario, la aplicación y los servicios de Firebase durante el proceso de autenticación.

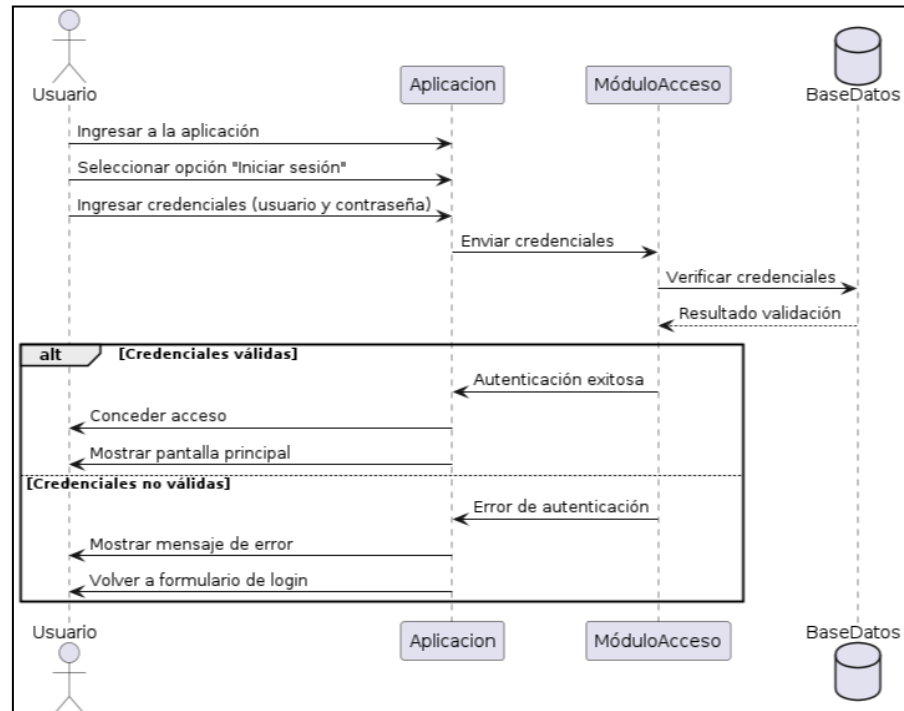


Figura 05. Diagrama de secuencia del caso de uso 02.

Fuente: Elaboración propia.

Diagrama de Secuencia CU-03 “Practicar exámenes”

El diagrama de secuencia “Practicar exámenes” representa cómo un usuario interactúa con la aplicación para realizar simulaciones de exámenes. La app solicita las preguntas desde Firestore, muestra cada pregunta al usuario y registra sus respuestas.

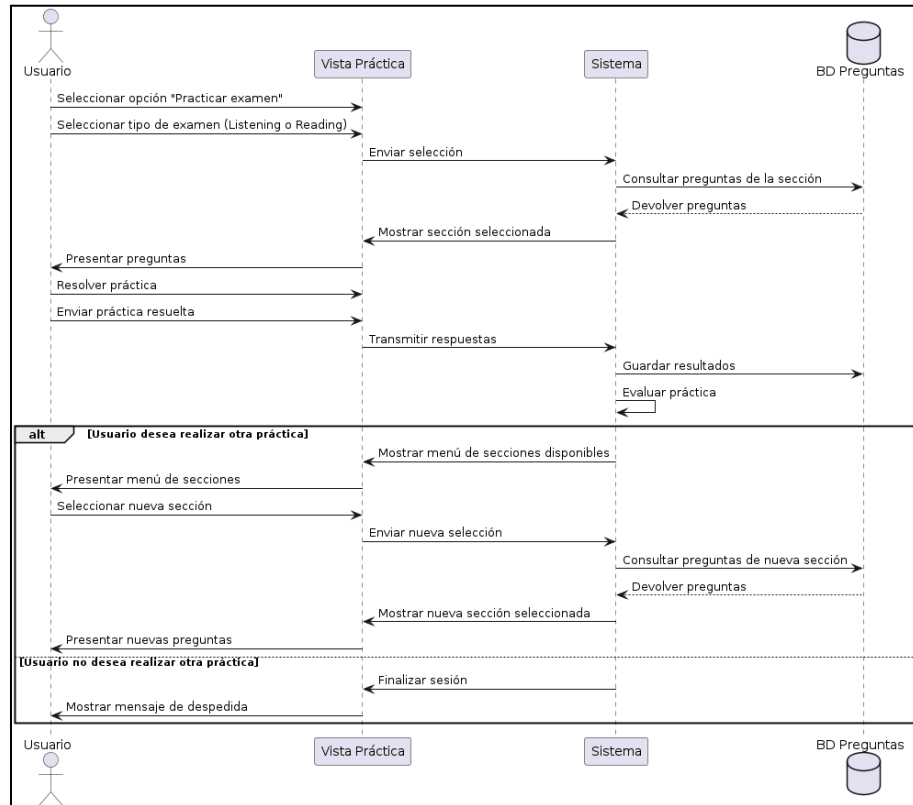


Figura 06. Diagrama de secuencia del caso de uso 03.

Fuente: Elaboración propia.

Diagrama de Secuencia CU-04 “Consultar institutos”

El diagrama de secuencia “Consultar institutos” ilustra cómo el usuario solicita información sobre los institutos disponibles dentro de la aplicación. La app envía la petición a la base de datos de Firebase, recupera los registros de los institutos y los muestra en la interfaz del usuario. Este flujo garantiza que el usuario pueda acceder a información actualizada sobre los institutos de manera rápida y eficiente.

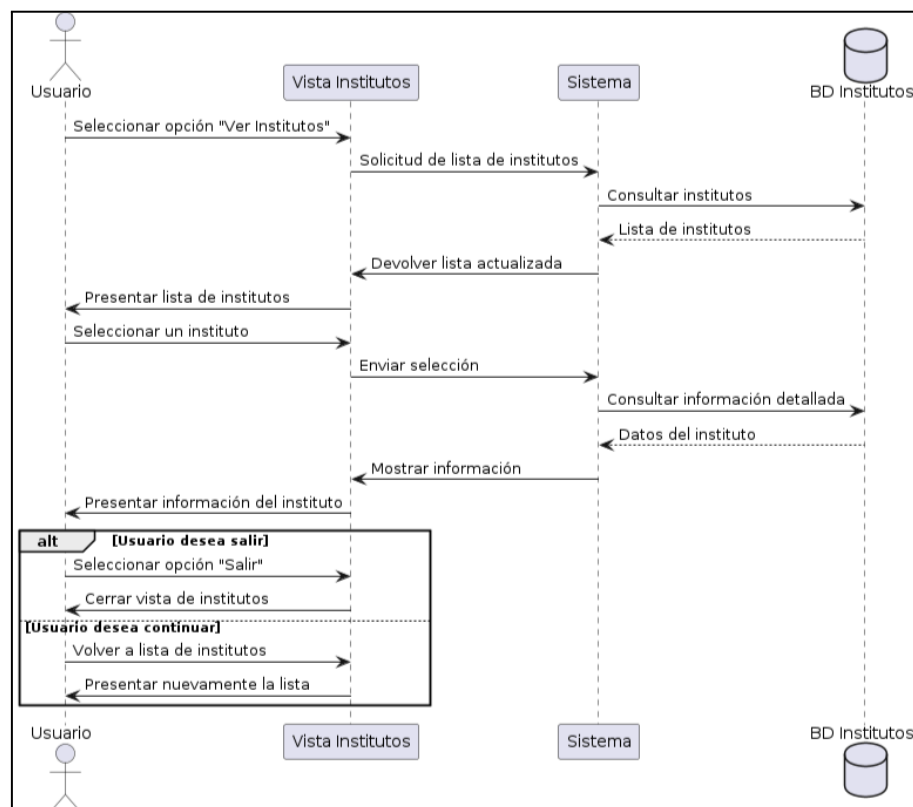


Figura 07. Diagrama de secuencia del caso de uso 04.

Fuente: Elaboración propia.

Diagrama de Secuencia CU-05 “Visualizar fecha de exámenes”

El diagrama “Visualizar fecha de exámenes” representa la interacción entre el usuario y la aplicación para consultar las fechas programadas de los exámenes de certificación. La app solicita los datos a Firestore, recibe la información y la presenta en la interfaz de manera organizada. Este diagrama permite evidenciar cómo se gestiona la consulta y presentación de información crítica para la planificación del usuario.

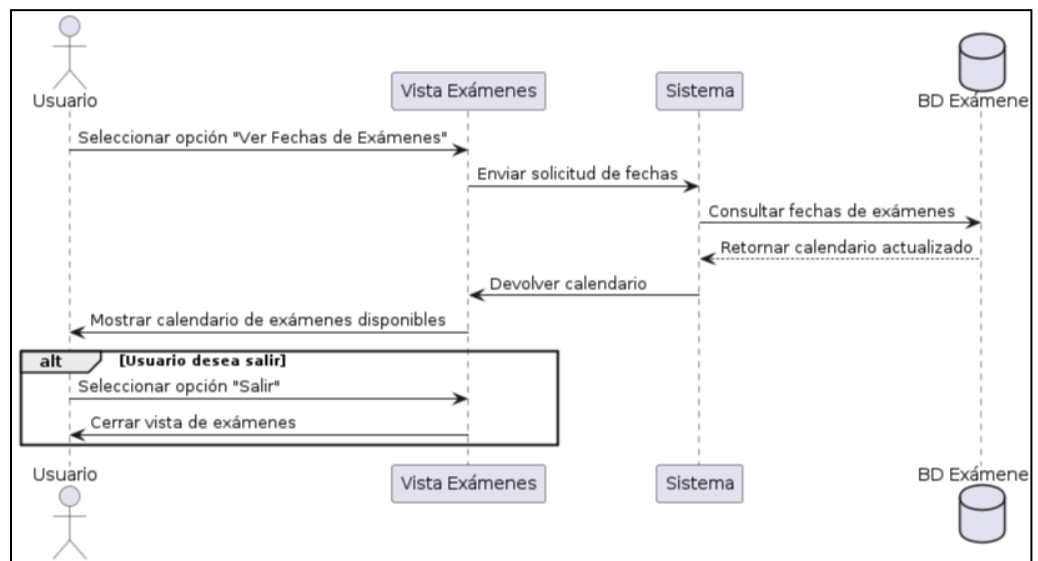


Figura 08. Diagrama de secuencia del caso de uso 05.

Fuente: Elaboración propia.

Diagrama de Secuencia CU-06 “Comparar precios”

El diagrama “Comparar precios” muestra cómo el usuario puede revisar y comparar los costos de los exámenes o cursos ofrecidos por diferentes institutos. La aplicación recupera la información de la base de datos y presenta los precios de manera comparativa, permitiendo al usuario tomar decisiones informadas. Este flujo destaca la secuencia de comunicación necesaria para consolidar datos y mostrarlos de forma clara.

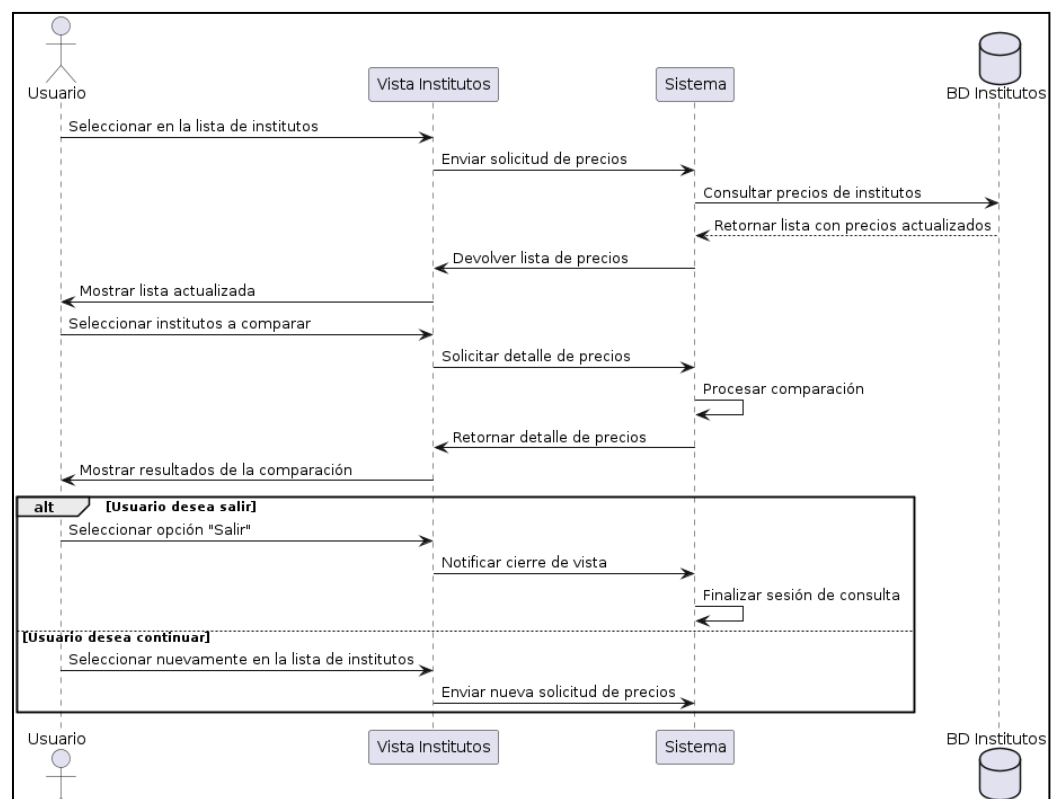


Figura 09. Diagrama de secuencia del caso de uso 06.

Fuente: Elaboración propia.

Diagrama de Secuencia CU-07 “Acceder a recursos de preparación”

El diagrama “Acceder a recursos de preparación” describe cómo el usuario interactúa con la aplicación para obtener materiales de estudio, como guías, simulaciones y ejercicios. La app solicita los recursos a Firestore o Firebase Storage, los recupera y los muestra al usuario en la interfaz. Este diagrama evidencia la secuencia de mensajes necesarios para garantizar que los usuarios tengan acceso rápido y organizado a los recursos educativos.

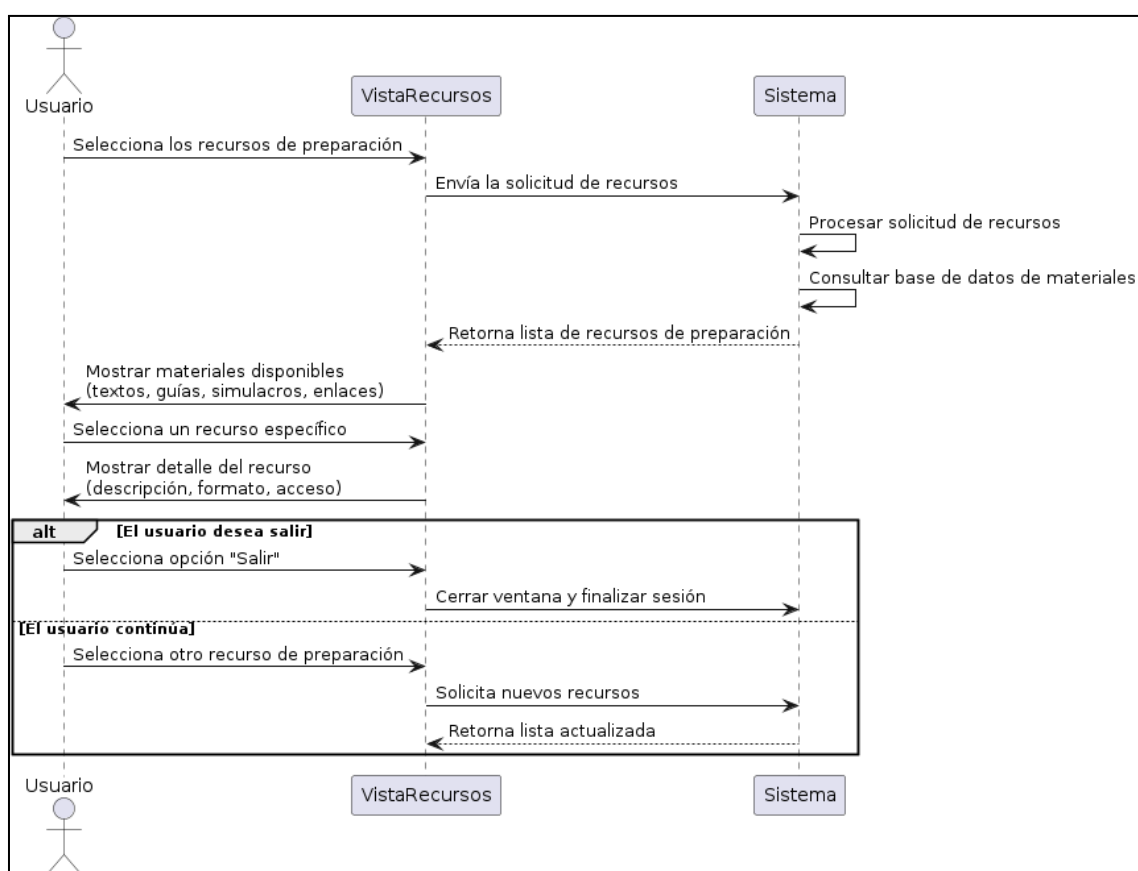


Figura 10. Diagrama de secuencia del caso de uso 07.

Fuente: Elaboración propia.

3.2.3. Diagrama de Colaboración (vista de diseño)

El diagrama de colaboración de SimulApp representa cómo los diferentes objetos y componentes del sistema interactúan entre sí para cumplir un caso de uso específico, destacando las relaciones y los mensajes enviados.

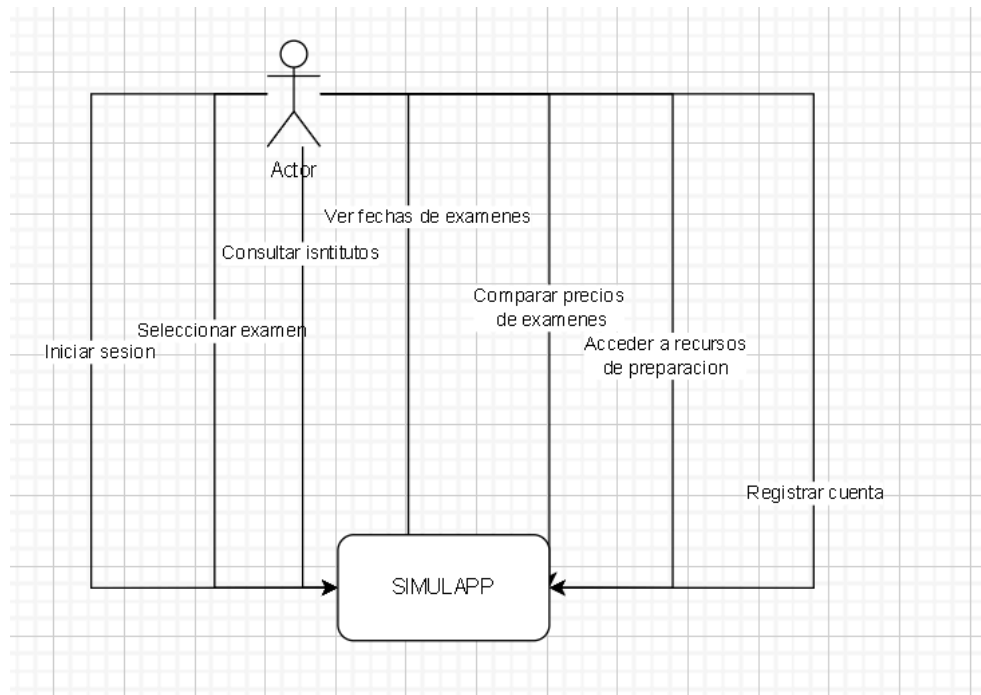


Figura 11. Diagrama de colaboración de simulapp.

Fuente: *Elaboración propia.*

3.2.4. Diagrama de Objetos

Diagrama de Actividades CU-01 “Registrar usuario”

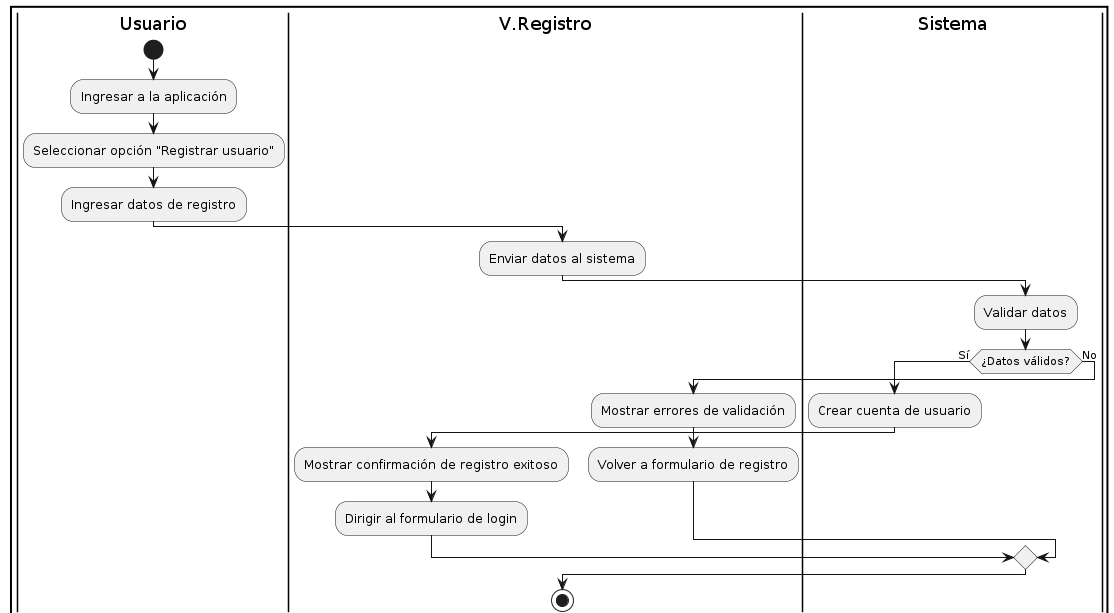


Figura 12. Diagrama de actividades con objetos del caso de uso 01.

Fuente: Elaboración propia.

Diagrama de Actividades CU-02 “Iniciar sesión”

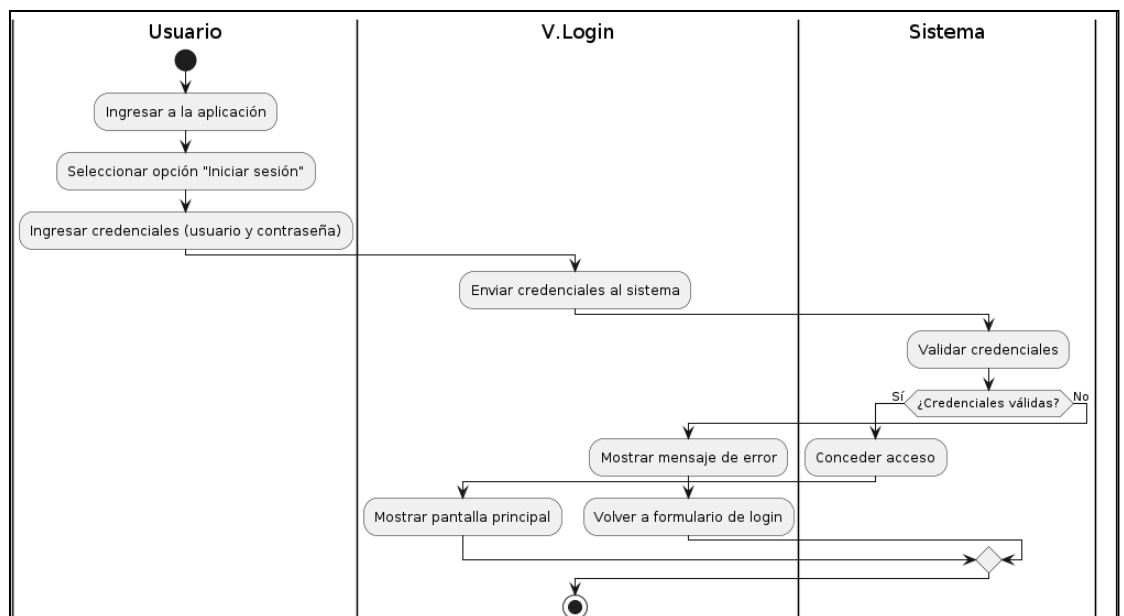


Figura 13. Diagrama de actividades con objetos del caso de uso 02.

Fuente: Elaboración propia.

Diagrama de Actividades CU-03 “Practicar exámenes”

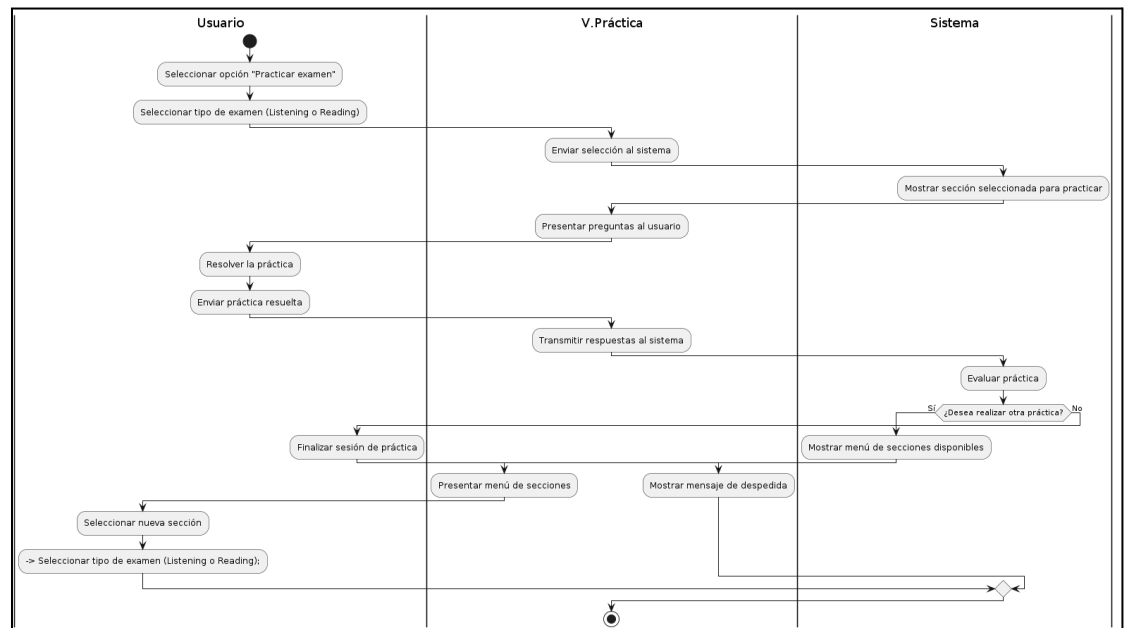


Figura 14. Diagrama de actividades con objetos del caso de uso 03.

Fuente: Elaboración propia.

Diagrama de Actividades CU-04 “Consultar institutos”

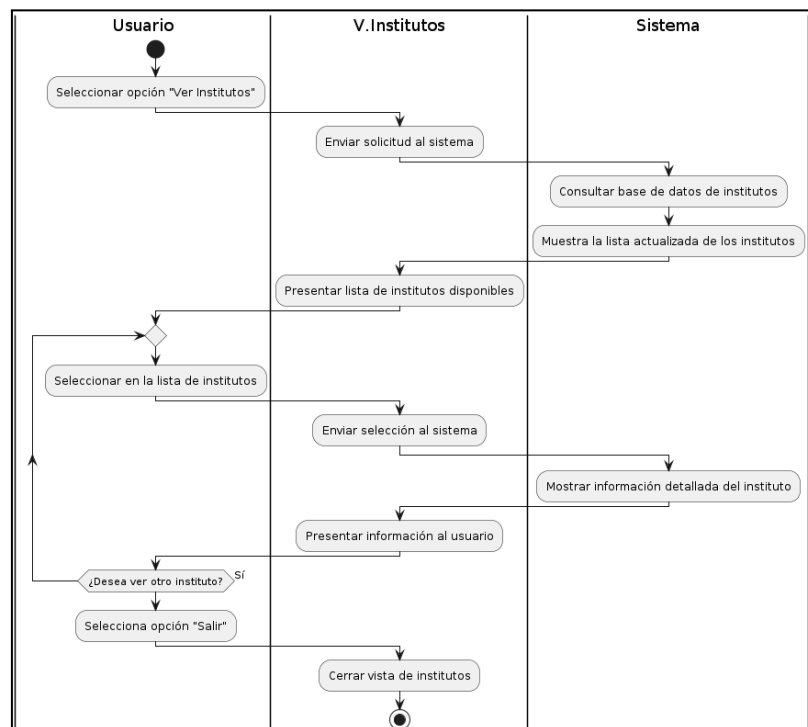


Figura 15. Diagrama de actividades con objetos del caso de uso 04.

Fuente: Elaboración propia.

Diagrama de Actividades CU-05 “Visualizar fecha de exámenes”

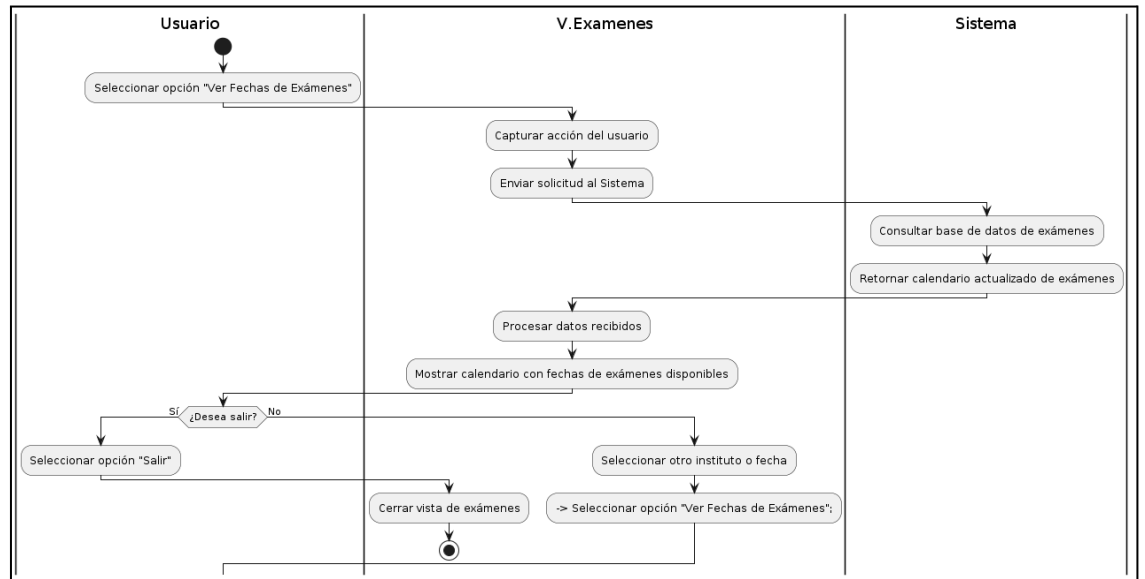


Figura 16. Diagrama de actividades con objetos del caso de uso 05.

Fuente: Elaboración propia.

Diagrama de Actividades CU-06 “Comparar precios”

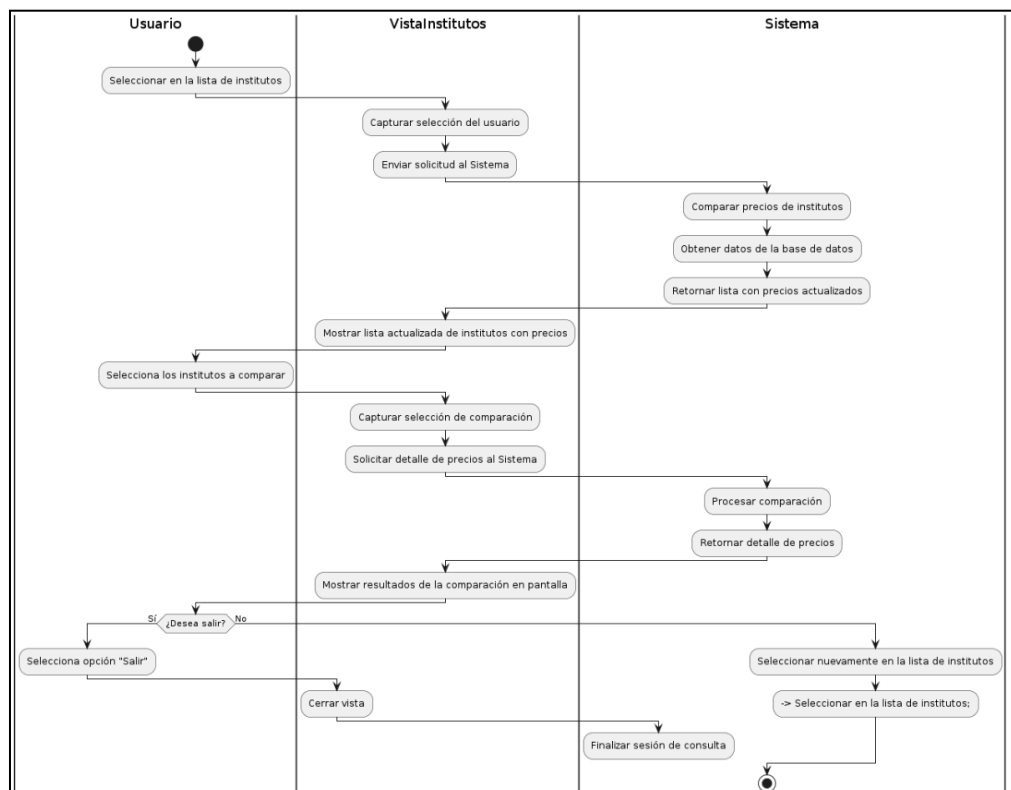


Figura 17. Diagrama de actividades con objetos del caso de uso 06.

Fuente: Elaboración propia.

Diagrama de Actividades CU-07 “Acceder a recursos de preparación”

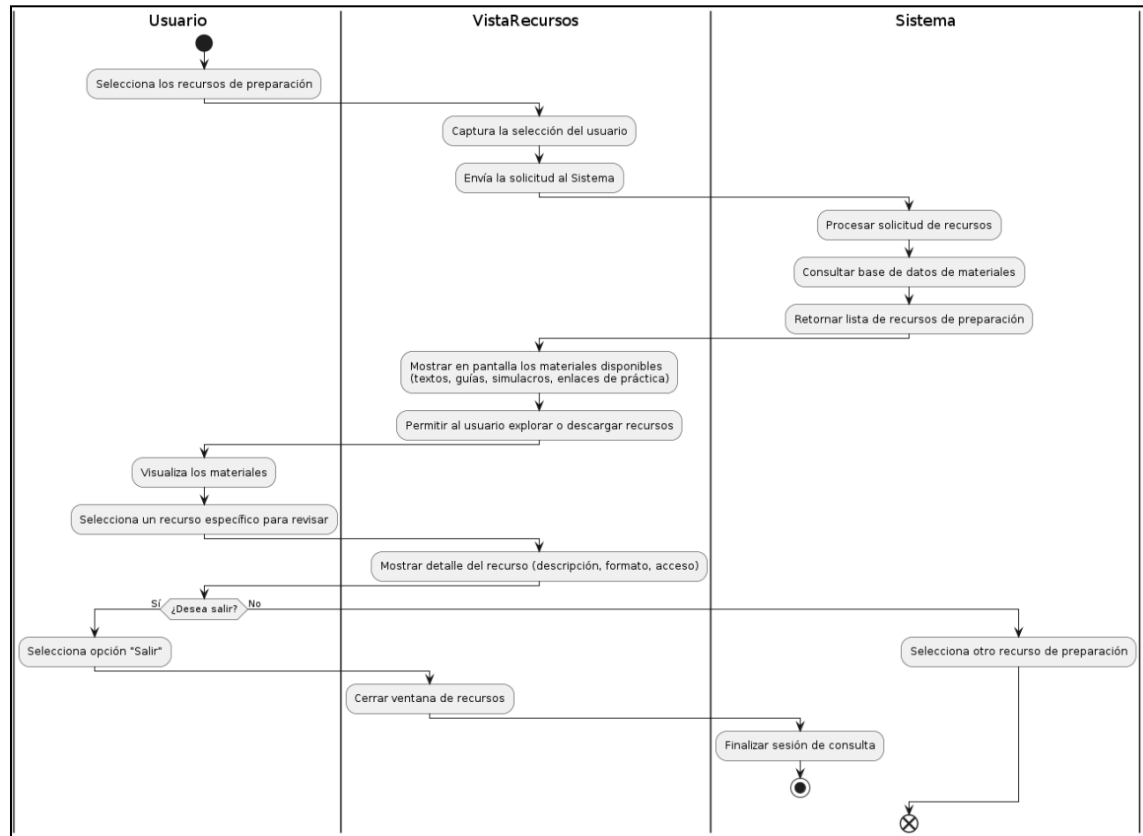
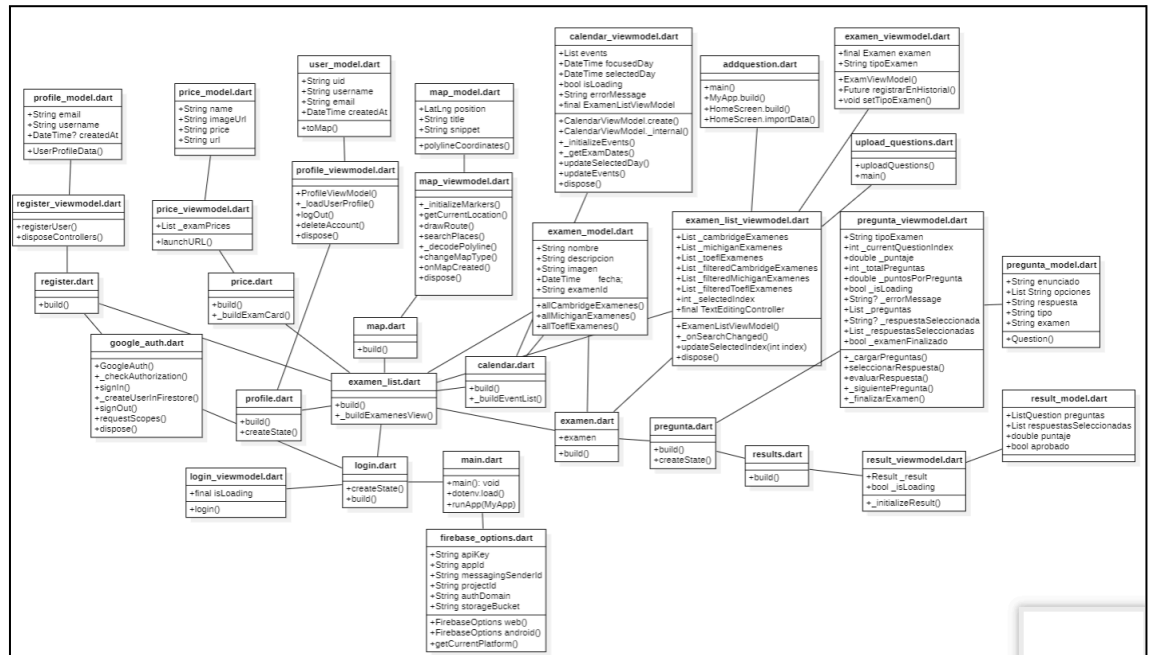


Figura 18. Diagrama de actividades con objetos del caso de uso 07.

Fuente: *Elaboración propia.*

3.2.5. Diagrama de Clases



[Link](#)

Figura 19. Diagrama de clases de simulapp.

Fuente: Elaboración propia.

3.2.6. Diagrama de Base de datos (relacional o no relacional)

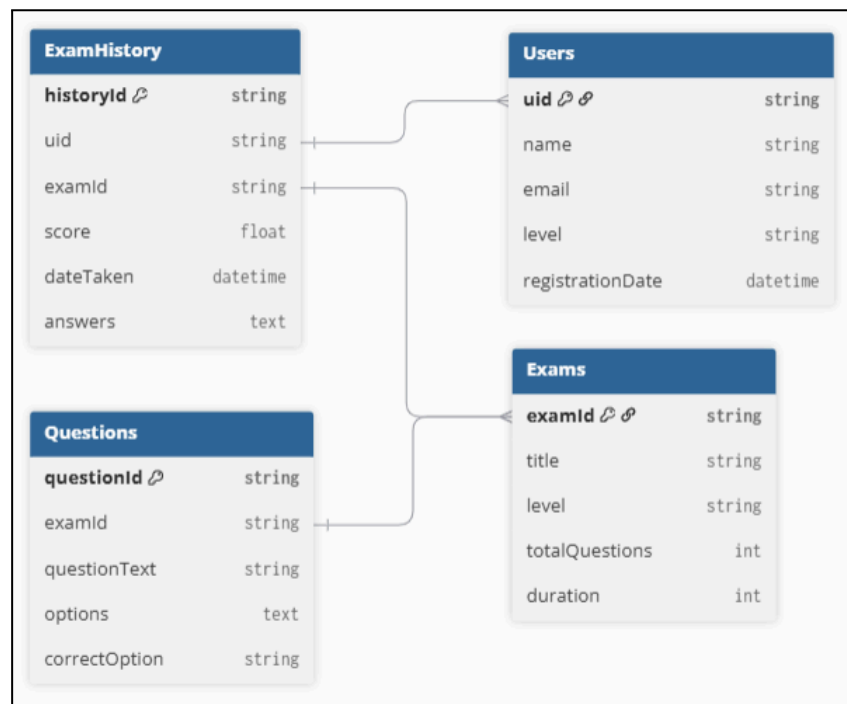


Figura 20. Diagrama entidad - relación de simulapp.

Fuente: Elaboración propia.

3.3. Vista de Implementación (vista de desarrollo)

3.3.1. Diagrama de arquitectura software (paquetes)

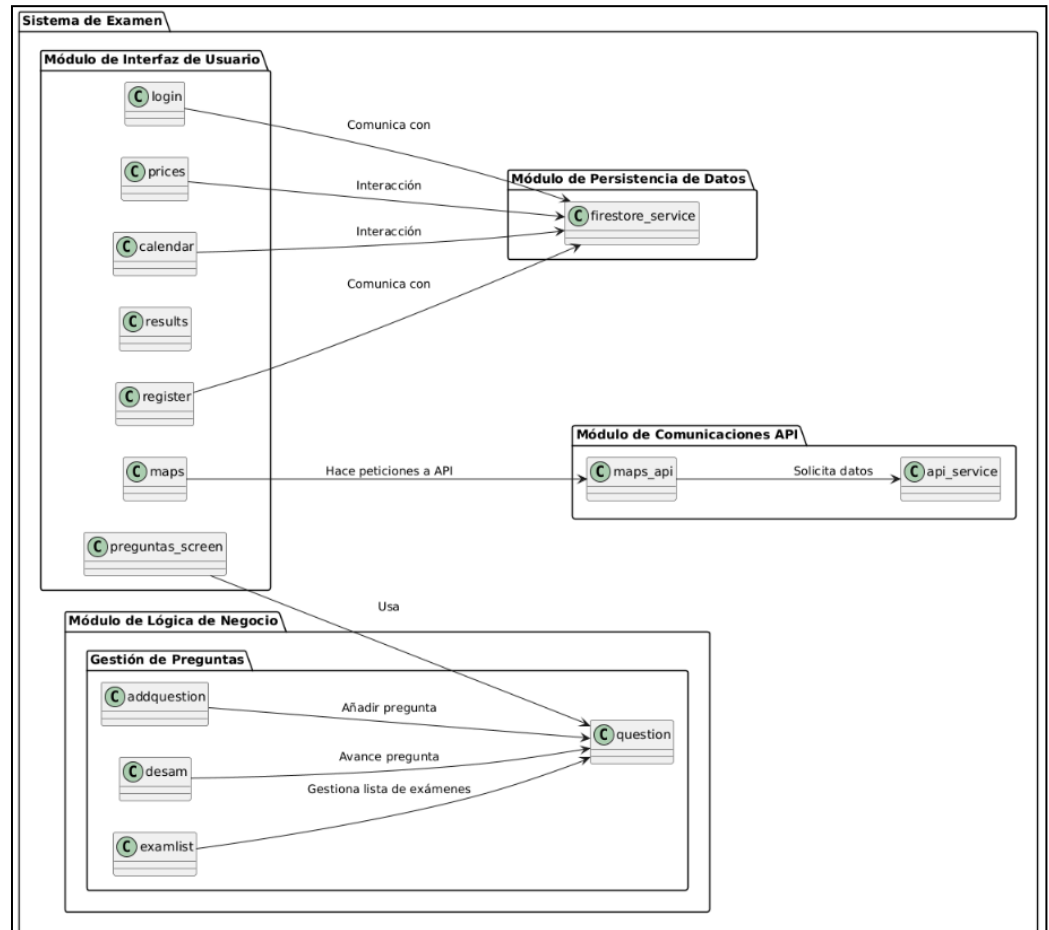


Figura 21. Diagrama de arquitectura (paquetes) de simulapp.

Fuente: Elaboración propia.

3.3.2. Diagrama de arquitectura del sistema (Diagrama de componentes)

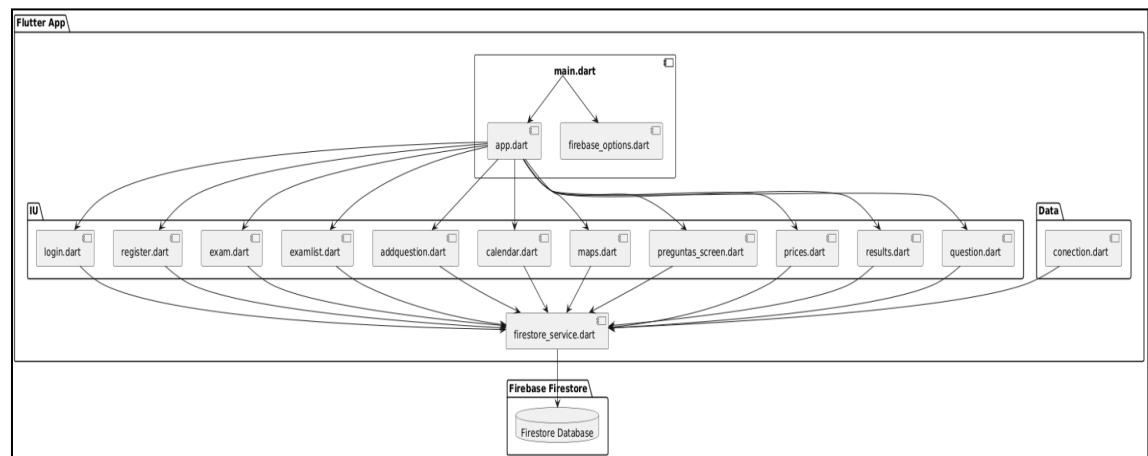
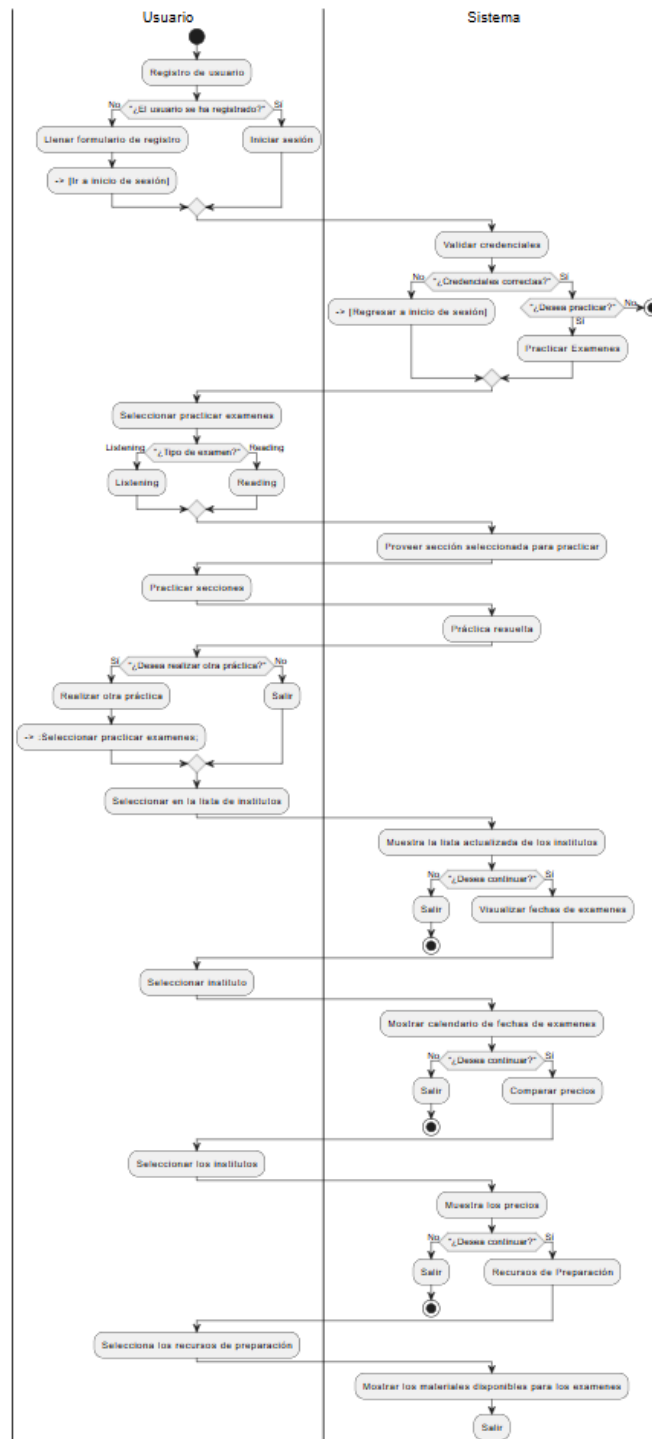


Figura 22. Diagrama de arquitectura (componentes) de simulapp.

Fuente: Elaboración propia.

3.4. Vista de procesos

3.4.1. Diagrama de Procesos del sistema (diagrama de actividad)



Link

Figura 23. El diagrama representa el flujo de simulapp.

Fuente: Elaboración propia.

3.5. Vista de Despliegue (vista física)

3.5.1. Diagrama de despliegue

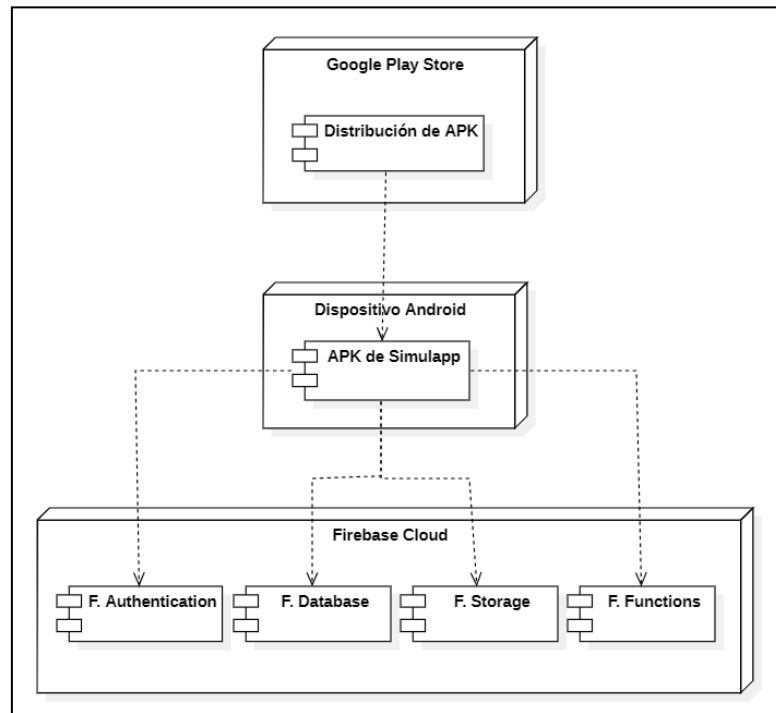


Figura 24. Diagrama de despliegue de simulapp.

Fuente: Elaboración propia.

4. ATRIBUTOS DE CALIDAD DEL SOFTWARE

Escenario de Funcionalidad

- **Descripción de la Situación:** Los usuarios necesitan acceder a simulacros de exámenes de certificación de inglés y consultar información sobre los institutos, precios y fechas de los exámenes. La plataforma debe garantizar que estas funcionalidades están siempre disponibles y operativas.
- **Actores:** Desarrolladores, Usuarios.
- **Eventos Disparadores:** El usuario selecciona una sección de examen, consulta institutos, visualiza fechas o compara precios.
- **Respuesta del Sistema:**
 - El sistema debe cargar las simulaciones de examen correctamente y permitir la práctica de las secciones seleccionadas.
 - Debe mostrar información actualizada de los institutos, precios y fechas de los exámenes en todo momento.
 - Los usuarios deben poder navegar sin interrupciones entre las funcionalidades.
- **Métrica de Éxito:**
 - Todos los simulacros de exámenes deben estar completos y accesibles sin errores.
 - La información debe estar actualizada con una frecuencia mínima de 24 horas para garantizar precisión.
 - Tiempo de respuesta para cargar los simulacros y las consultas de información no debe superar los 2 segundos.

Escenario de Usabilidad

- **Descripción de la Situación:** La aplicación debe ser fácil de usar, especialmente para usuarios con diversos niveles de habilidad tecnológica. La navegación debe ser intuitiva y los elementos clave (simulacros, consultas de institutos, calendario) deben ser fáciles de localizar.
- **Actores:** Desarrolladores, Usuarios.
- **Eventos Disparadores:** El usuario accede a la aplicación y comienza a interactuar con ella.
- **Respuesta del Sistema:**
 - La interfaz debe ser limpia y clara, con botones y opciones bien organizados.
 - La experiencia debe estar optimizada para ser accesible tanto en dispositivos Android como iOS.
 - Se deben realizar pruebas periódicas de usabilidad con usuarios para asegurar que la interfaz sea comprendida fácilmente.
- **Métrica de Éxito:**
 - La puntuación en las encuestas de usabilidad debe superar un 85% de satisfacción.
 - El tiempo medio de aprendizaje para un nuevo usuario debe ser de menos de 5 minutos.
 - Tasa de retención de usuarios debe ser superior al 70% después de 30 días de uso.

Escenario de confiabilidad

- **Descripción de la Situación:** La plataforma debe ser confiable, especialmente en el acceso a los simulacros y la consulta de información relevante. El sistema debe manejar posibles fallos sin interrumpir la experiencia del usuario.
- **Actores:** Desarrolladores, Sistema, Usuarios.
- **Eventos Disparadores:** Acceso a simulacros o información en cualquier momento.
- **Respuesta del Sistema:**
 - El sistema debe tener mecanismos de backup y recuperación para evitar pérdidas de datos o caídas del servicio.
 - Debe poder continuar funcionando sin errores si ocurre una interrupción temporal.
 - Los usuarios deben ser notificados de cualquier mantenimiento o error en tiempo real.
- **Métrica de Éxito:**
 - Tiempo de inactividad no debe superar el 1% de tiempo total de operación.
 - Los usuarios deben experimentar una tasa de fallos inferior al 0.5% en sus interacciones con la plataforma.
 - El sistema debe ser capaz de manejar hasta 1,000 usuarios concurrentes sin degradación de rendimiento.

Escenario de rendimiento

- **Descripción de la Situación:** Los tiempos de carga para los simulacros y la información deben ser rápidos para evitar frustraciones en los usuarios. El sistema debe manejar eficientemente las consultas de información y la carga de simulacros, sin retrasos significativos.
- **Actores:** Desarrolladores, Sistema.
- **Eventos Disparadores:** El usuario selecciona un simulacro o consulta una opción en la aplicación.
- **Respuesta del Sistema:**
 - Los simulacros deben cargarse en menos de 2 segundos desde la selección.
 - Las consultas de institutos, precios y fechas deben mostrarse al usuario en un máximo de 3 segundos.
 - El sistema debe poder manejar grandes volúmenes de consultas sin afectar la velocidad.
- **Métrica de Éxito:**
 - Tiempo de respuesta promedio por acción debe ser de menos de 2 segundos.
 - Los tiempos de carga deben mantenerse por debajo de los 3 segundos incluso bajo alta carga de usuarios.

Escenario de mantenibilidad

- **Descripción de la Situación:** El sistema debe ser fácilmente actualizable para incorporar nuevas funcionalidades, corregir errores y mejorar el rendimiento. La mantenibilidad del código es clave para asegurar la viabilidad del sistema a largo plazo.
- **Actores:** Desarrolladores, Administradores del sistema.
- **Eventos Disparadores:** Incorporación de nuevas funciones, corrección de errores, optimización del sistema.
- **Respuesta del Sistema:**
 - El sistema debe contar con un código modular y documentado que permita a los desarrolladores realizar cambios sin afectar la estabilidad.
 - Las actualizaciones deben ser fáciles de implementar sin interrumpir el servicio.
 - Las pruebas de regresión deben realizarse con cada actualización para asegurar que no se introduzcan nuevos problemas.
- **Métrica de Éxito:**
 - El tiempo de implementación de nuevas características debe ser inferior a 2 semanas.
 - Las actualizaciones deben completarse con una tasa de éxito del 95% o superior en las pruebas de integración.
 - El tiempo medio para resolver un incidente debe ser inferior a 24 horas.