Documento de Arquitectura de Software
Grupo 11 - 2021

Taller de Sistemas Empresariales
Taller de Sistemas de Información Java EE

Joaquín Cabrera

Agustina Corvo

Bruno Fernández

Gabriel Tulumello

Marcos Pulido

Marzo 2021

Tabla de Contenido

1. Introducción

Este documento presenta la arquitectura de la plataforma vacunas.uy, la cual fue planteada como trabajo laboratorio de la edición 2021 de los cursos Taller de Sistemas Empresariales (Ingeniería y Licenciatura en Computación) y Taller de Sistemas de Información Java EE (Tecnólogo en Informática) de la Facultad de Ingeniería, Universidad de la República.

Esta plataforma apunta a brindar soporte a procesos de vacunación en Uruguay, aprovechando tanto recursos como iniciativas de los distintos actores involucrados.

1.1. Objetivo del Documento

El objetivo de este Documento de Arquitectura de Software (Software Architecture Document, SAD) es brindar una visión comprensible de la arquitectura general de la plataforma vacunas.uy.

1.2. Representación de la Arquitectura

La arquitectura de la plataforma vacunas.uy está representada por diferentes vistas que permiten visualizar, entender y razonar sobre los elementos significativos de la arquitectura e identificar áreas de riesgo que requieran mayor detalle de elaboración [1]. En particular, las vistas utilizadas para representar la arquitectura de la plataforma vacunas.uy son:

1. **Vista de Casos de Uso:** Se detalla el modelo de dominio y el proceso de negocio junto con los casos de uso considerados como críticos para el sistema, así como también los actores que participan en los mismos.

2. **Vista de Restricciones:** Se exponen las distintas restricciones con las que tiene que convivir el sistema final y cómo fueron consideradas en el proceso de desarrollo. Algunas de ellas pueden ser relacionadas a normativas legales, aspectos tecnológicos, seguimiento de estándares, etc.

3. **Vista de Atributos de Calidad:** Se describen los requerimientos no funcionales del sistema.

4. **Vista Lógica:** Se presenta la arquitectura del sistema indicando los módulos lógicos principales, sus responsabilidades y dependencias, así como también los distintos refinamientos.

5. **Vista de Distribución:** Describe la infraestructura necesaria para desplegar el sistema.

6. **Vista de Implementación:** Se detallan los componentes de despliegue que conforman el sistema, así como también sus dependencias.

7. **Vista de Decisiones de Arquitectura:** La Vista de Decisiones de Arquitectura presenta y describe las principales decisiones de arquitectura tomadas.

1.3. Partes Interesadas (i.e. *stakeholders*)

Las partes interesadas en la plataforma vacunas.uy son:

1. **Usuarios:** Interesados en utilizar la aplicación. No necesariante son una persona física, puede ser un sistema que consuma servicios.
2. **Administradores de infraestructura:** Encargados de conocer el ambiente donde funcionará el sistema.
3. **Arquitecto:** Equipo que se encarga de pensar todos los aspectos de la arquitectura del sistema.
4. **Desarrolladores:** Equipo de desarrollo del sistema.
5. **Desarrolladores nodos Periféricos:** Equipo de desarrollo de sistemas periféricos que consumen recursos del sistema.

La Tabla 1 indica a qué parte interesada está orientada cada una de las vistas de la arquitectura.

Tabla 1: Partes Interesadas y Vistas

	Vista de Casos de Uso	Vista de Restricciones	Vista de Atributos de Calidad	Vista Lógica	Vista de Distribución	Vista de Implementación	Vista de Decisiones de Arquitectura
Usuarios	✓						
Administradores de infraestructura			✓		✓	✓	✓
Arquitecto	✓	✓	✓	✓	✓	✓	✓
Desarrolladores	✓	✓	✓	✓	✓	✓	✓
Desarrolladores nodos Periféricos	✓	✓					

1.4. Organización del Documento

El resto del documento se organiza en ocho secciones, cada una de las cuales describe una de las vistas que representan la arquitectura:

- Sección 2: Vista Conceptual
- Sección 3: Vista de Casos de Uso
- Sección 4: Vista de Restricciones
- Sección 5: Vista de Atributos de Calidad
- Sección 6: Vista Lógica
- Sección 7: Vista de Distribución
- Sección 8: Vista de Implementación
- Sección 9: Vista de Decisiones de Arquitectura

2. Vista Conceptual

La Vista Conceptual brinda una descripción general de la plataforma y presenta los principales conceptos asociados a la misma,

2.1. Descripción General de la Plataforma

En este apartado se describe a nivel global la plataforma vacunas.uy, detallando sus componentes y funcionalidades más importantes.

La plataforma vacunas.uy surge en el contexto actual, a nivel mundial, que está atravesando la sociedad por la pandemia de Coronavirus (COVID-19). Si bien ya se cuenta con vacunas desarrolladas para combatir esta enfermedad, es necesario, a su vez, contar con un sistema informático que sea capaz de gestionar todo el ciclo de vacunación, desde la recepción de las unidades, envío a los centros de vacunación, asignación del personal de la salud para realizar las vacunaciones, así como también permitir a los ciudadanos inscribirse en las distintas agendas para recibir las vacunas, entre otras funcionalidades. Además, no solo se espera que esta plataforma pueda gestionar el contexto actual, sino que pueda utilizarse a futuro para gestionar cualquier plan de vacunación.

El sistema está integrado por un componente central, el cual va a estar desplegado en un servidor de aplicaciones Wildfly en la plataforma de Elastic Cloud, éste se comunica con diversos sub-componentes que brindarán servicios a los distintos actores de esta realidad. Por un lado, los ciudadanos tendrán acceso al sub-componente denominado "Frontoffice", el cual será una aplicación web donde pueden ver información primaria acerca del avance del plan de vacunación, como por ejemplo, cantidad de actos vacunales en el día. Si se autentican, tendrán la posibilidad de gestionar su agenda y consultar los distintos vacunatorios disponibles. A su vez, tendrán acceso al sub-componente "Móvil" el cual consta de una aplicación para dispositivos Android, en la que pueden recibir notificaciones y solicitar certificados de vacunación.

Los vacunadores también tendrán acceso al sub-componente de "Frontoffice", a través del cual podrán consultar información acerca de su agenda de vacunación, incluyendo vacunatorios y puesto asignado. Además podrán interactuar con otros vacunadores mediante el uso de un chat.

Las autoridades tendrán acceso al sub-componente "Backoffice", el cual será un módulo web desplegado en el mismo contenedor Wildfly que el componente central. En él, una vez autenticados, podrán gestionar información central de la plataforma como ser enfermedades, vacunas, proveedores, etc. A su vez, podrán crear distintos planes de vacunación pudiendo personalizar el público objetivo y las agendas. Por último, podrán generar distintos tipos de reportes.

Los administradores también podrán acceder al sub-componente de "Backoffice", donde una vez autenticados podrán gestionar usuarios y roles, así como también los distintos nodos periféricos.

Finalmente, en cuanto a la comunicación, la plataforma vacunas.uy se intercomunicará con distintos servicios de recopilación y procesamiento de datos, como ser Salud.uy o la Plataforma de Interoperabilidad por parte de AGESIC. Asimismo interactuará con nodos denominados periféricos los cuales serán: Vacunatorio (el cual brinda información periódicamente al componente central en cuanto al suministro de dosis y asignación de vacunadores a puestos de

vacunación) y Socio Logístico (brinda información periódicamente acerca del estado del envío de las dosis hacia los vacunatorios).

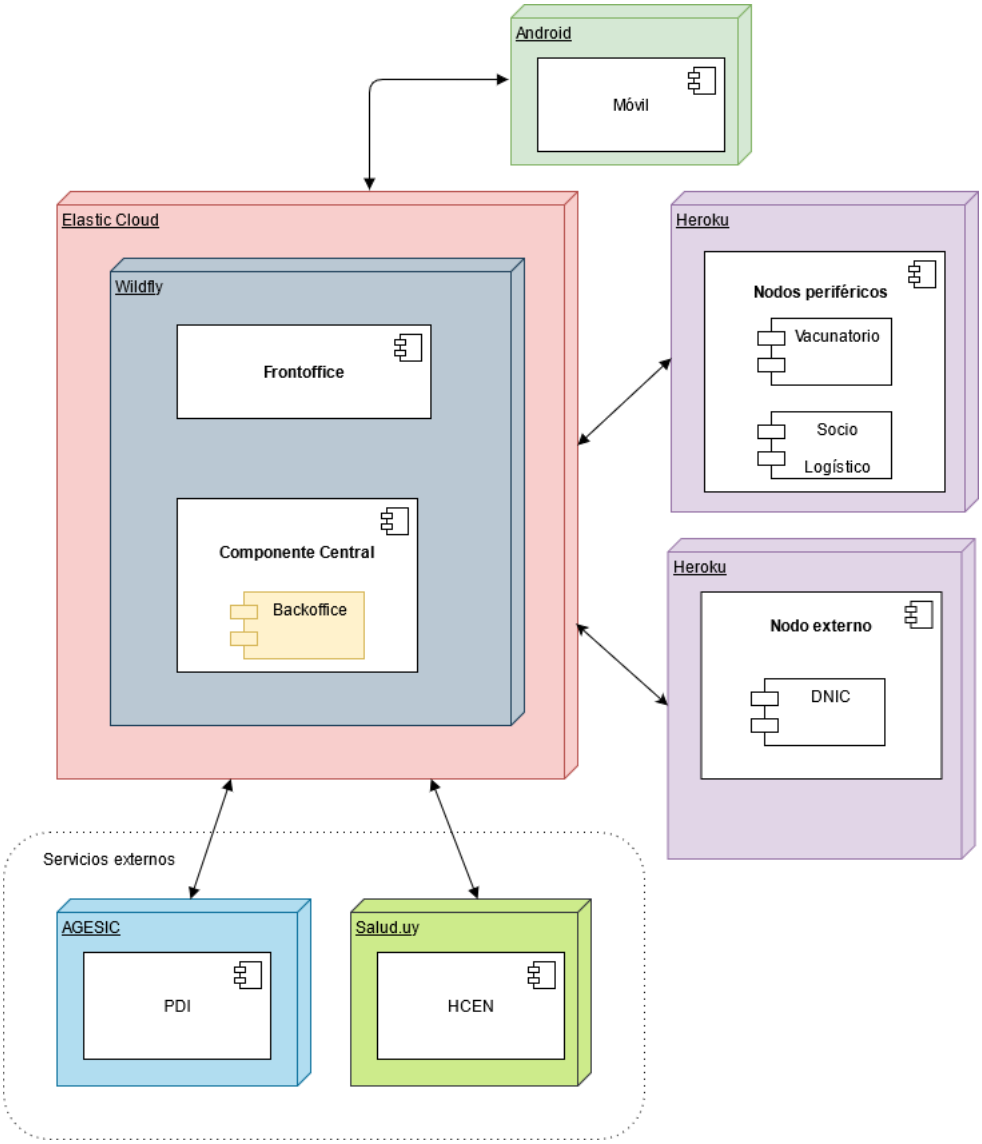

Figura 1: Descripción General de la Plataforma

2.2. Modelo Conceptual

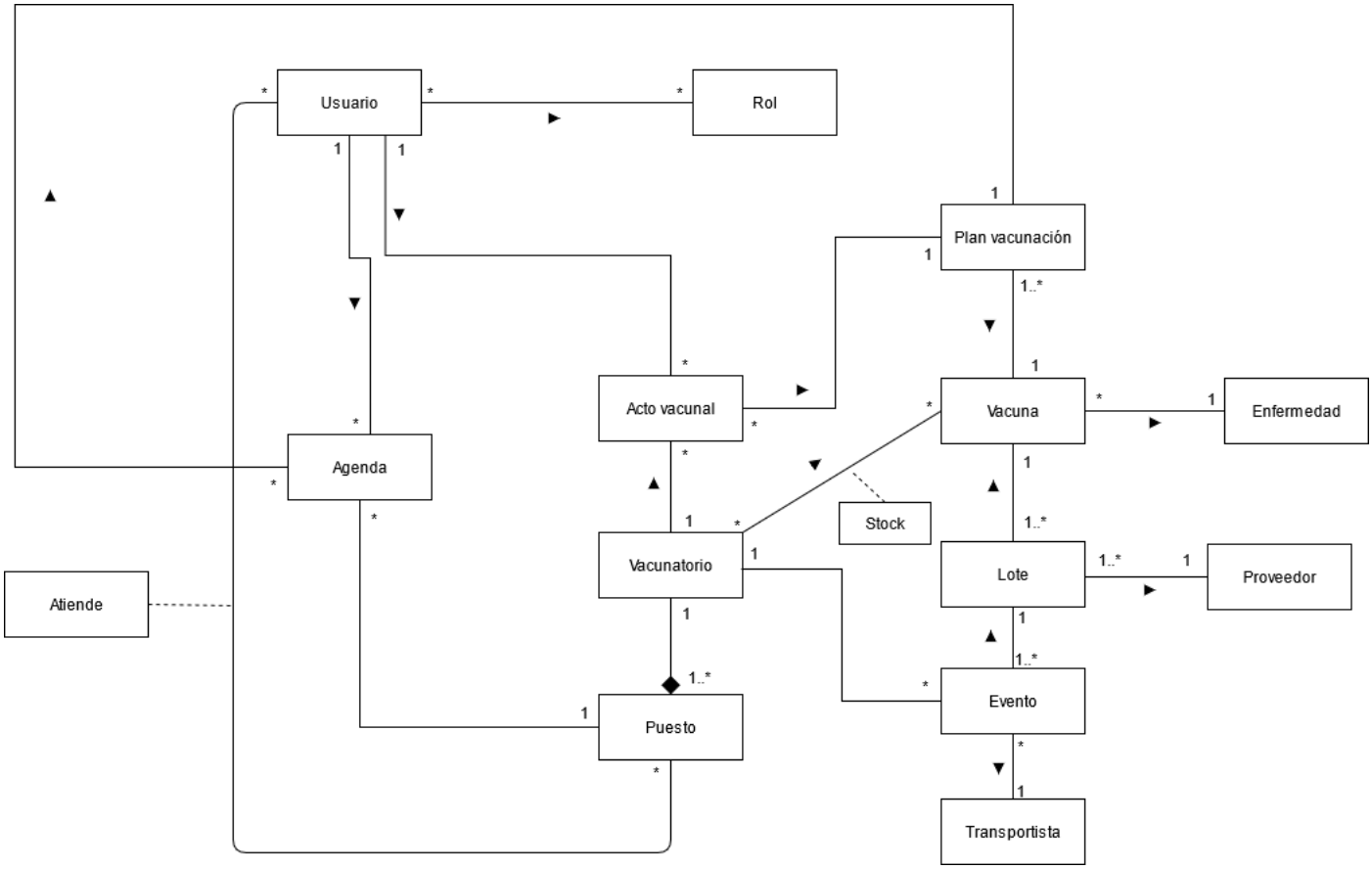

Figura 2: Modelo de Dominio

3. Vista de Casos de Uso

La Vista de Casos de Uso se centra en los aspectos funcionales de la plataforma. En esta vista se presentan los actores así como los casos de uso de la plataforma, y se detallan los casos de uso que se consideran críticos para la arquitectura.

3.1. Actores

En esta sección se presentan los actores que interactúan con la plataforma vacunas.uy.

3.2. Diagrama de Casos de Uso

En esta sección se utilizan Diagramas de Casos de Uso UML para presentar los casos de uso de la plataforma.

Casos de uso para Autoridad

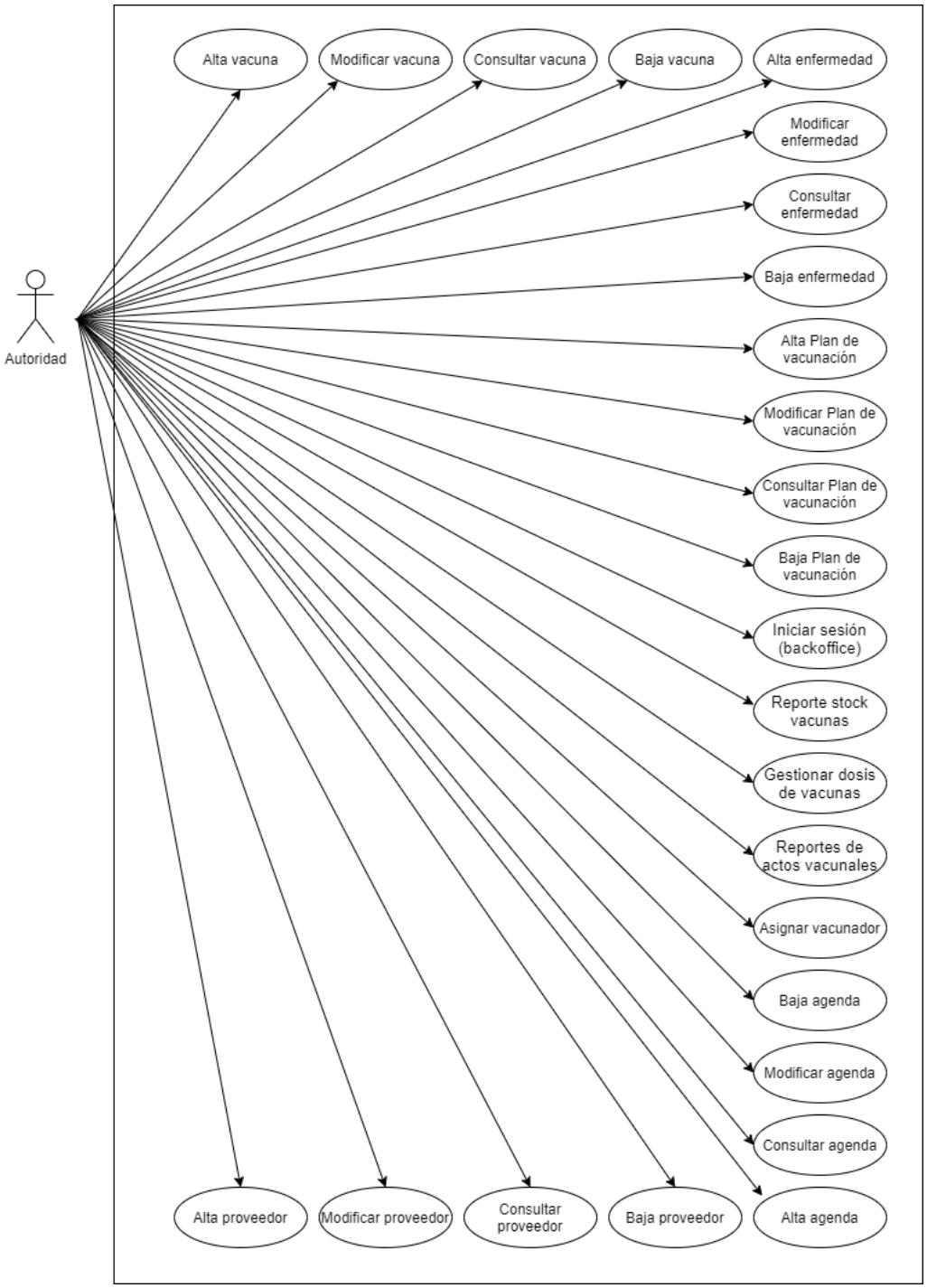

Figura 3: Caso de Uso - Autoridad

Casos de uso para Ciudadano - Vacunador - Invitado

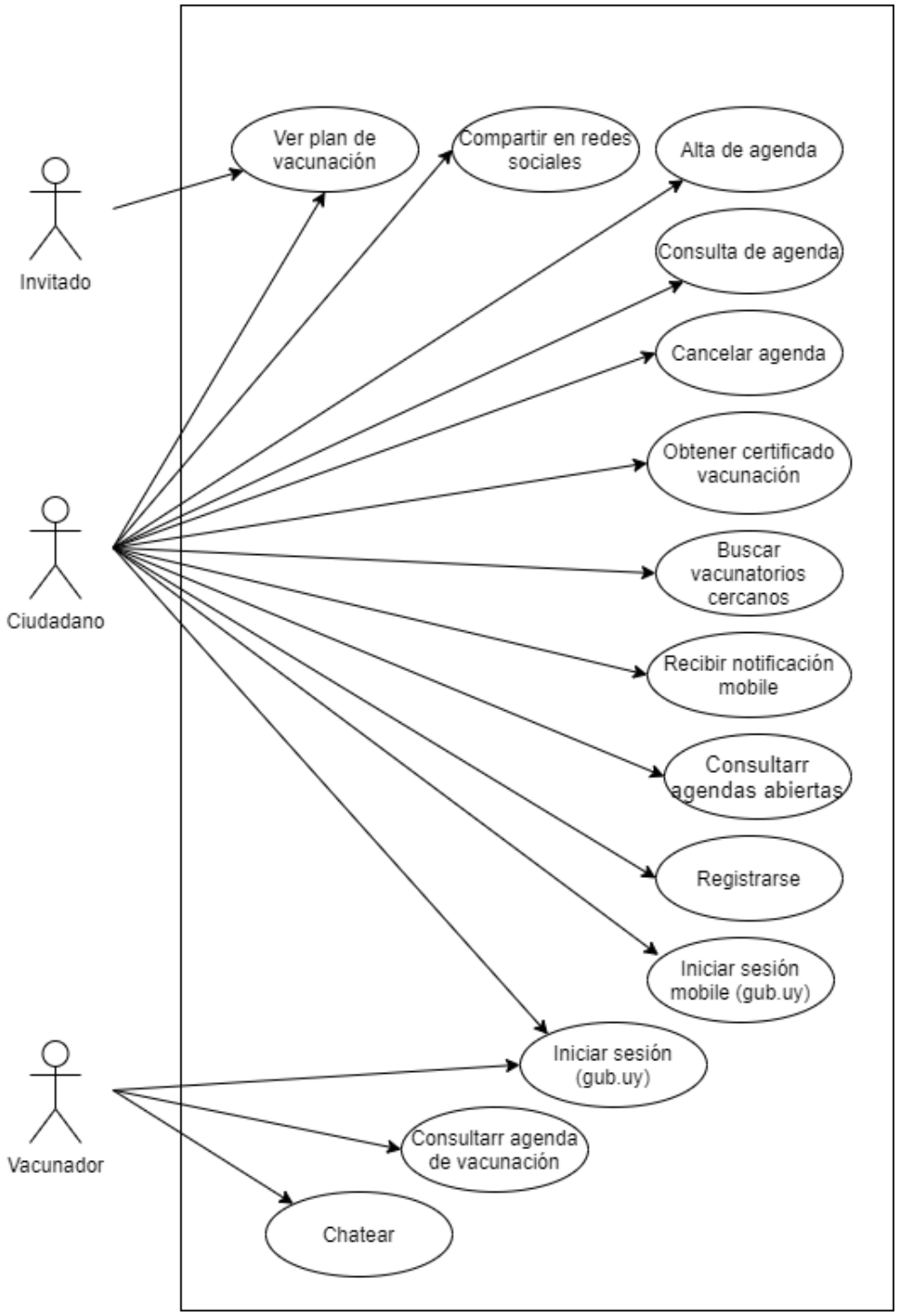

Figura 4: Caso de Uso - Ciudadano - Vacunador - Invitado

Casos de uso para Vacunatorio - Transportista

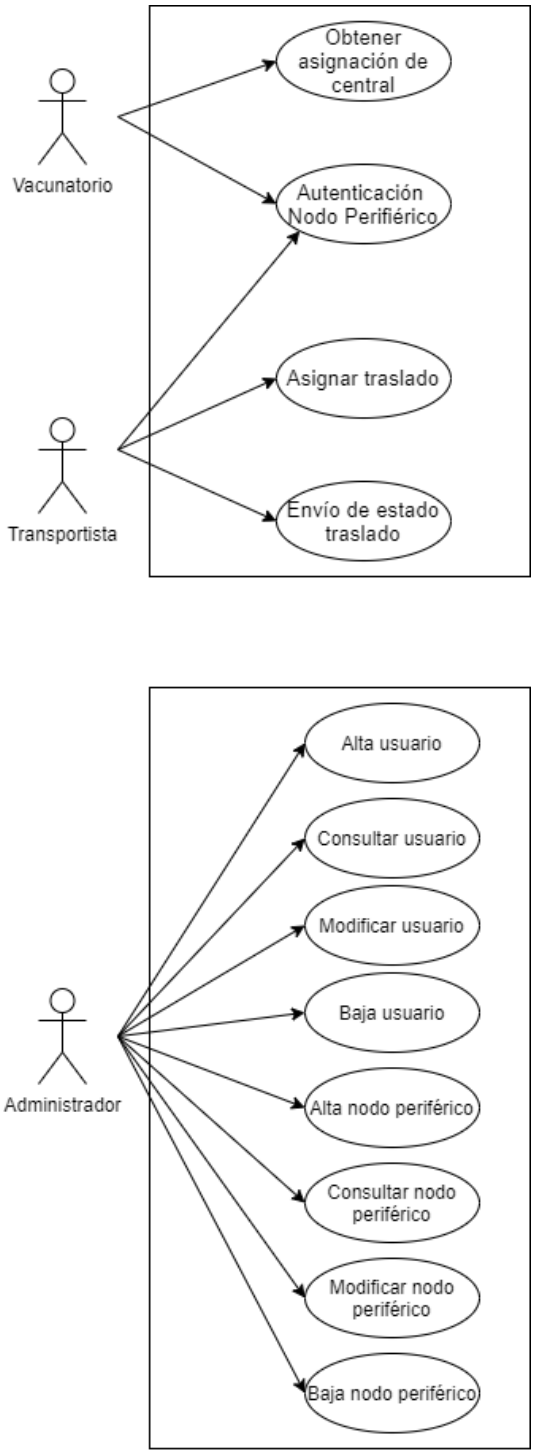

Figura 5: Caso de Uso - Vacunatorio - Transportista

3.3. Caso de Uso crítico 1

Iniciar sesión

Actores

Ciudadano, vacunador

Descripción

Identifica al actor como usuario registrado.

Pre-condiciones

El usuario debe estar registrado en gub.uy.

Flujo de Eventos

1. El ciudadano o vacunador proporciona la cédula y contraseña.
2. El sistema autentica al cliente mediante gub.uy.
3. El sistema localiza el perfil del usuario.
4. El sistema otorga el ingreso al usuario en el perfil asignado.

Post-condiciones

El usuario queda con una sesión activa en el sistema.

3.4. Caso de Uso crítico 2

Alta agenda

Actores

Ciudadano.

Descripción

Se realiza una reserva en vacunas.uy.

Pre-condiciones

- Ciudadano con sesión activa en el sistema.
- Existen planes de vacunas vigentes para el perfil de usuario.
- Existen vacunatorios con planes de vacunas vigentes para el perfil de usuario.
- Existe disponibilidad de agenda en los vacunatorios con planes de vacunas vigentes para el perfil de usuario.

Flujo de Eventos

1. El usuario solicita los planes de vacunas vigentes.
2. Se muestran los planes de vacunas vigentes para su perfil.
3. Se selecciona un plan de vacunación.
4. Se muestran los vacunatorios que poseen dichos planes.
5. Se selecciona un vacunatorio por parte del usuario.
6. Se muestra la agenda disponible para el vacunatorio seleccionado.
7. Se selecciona fecha y hora por parte del usuario dentro de los disponibles.
8. El sistema ingresa la reserva.
9. Se notifica al usuario.

Post-condiciones

El ciudadano queda con un plan de vacunación establecido para una fecha hora determinada.

3.5. Caso de Uso crítico 3

Obtener asignación de central

Actores

Vacunatorio

Descripción

El vacunatorio desea obtener las asignaciones de los vacunadores para determinada fecha.

Pre-condiciones

El vacunatorio está dado de alta en el sistema.

Flujo de Eventos

1. El vacunatorio inicia la solicitud enviando las credenciales para autenticación y la fecha para la que desea obtener las asignaciones.
2. El sistema autentica al vacuntorio y envia los datos solicitados.

Post-condiciones

El vacunatorio obtiene las asignaciones.

3.6. Caso de Uso crítico 4

Envío estado agenda a central

Actores

Vacunatorio

Descripción

El vacunatorio desea enviar el reporte de dósis suministradas al componente central.

Pre-condiciones

El vacunatorio está dado de alta en el sistema.

Flujo de Eventos

1. El vacunatorio inicia la solicitud enviando las credenciales para autenticación y el reporte con las dosis suministradas.
2. El sistema procesa y actualiza la información.

Post-condiciones

El sistema actualiza la información sobre las dósis suministradas.

3.7. Caso de Uso crítico 5

Buscar vacunatorios cercanos

Actores

Ciudadano

Descripción

Muestra en el dispositivo móvil un mapa con los vacunatorios en un radio determinado por el usuario con centro en la ubicación del equipo móvil.

Pre-condiciones

- Dispositivo móvil con Sistema Operativo Android.
- Dispositivo móvil con función de GPS activa.
- Dispositivo móvil con conectividad de datos.
- Ciudadano con sesión activa en el sistema.

Flujo de Eventos

1. El usuario selecciona el radio de búsqueda.
2. El sistema solicita a Android la ubicación del dispositivo.
3. El sistema envía la petición de búsqueda al Sistema Central.
4. El Sistema Central retorna el resultado de la búsqueda.

Post-condiciones

Se despliega en un mapa dentro de la aplicación el resultado devuelto.

4. Vista de Restricciones

En esta vista se muestran las restricciones que hay tanto en el desarrollo como en el producto terminado de este software las cuales pueden ser normativas, de estándares y tecnologías.

4.1. Normativas

Licenciamiento

Necesario para el acceso a el sistema de gub.uy para el enlace de los datos internos de vacunasUY con sus respectivas entidades en gub.uy.

4.2. Estándares

UML

Todas las representaciones gráficas de el flujo del sistema, de acceso a funcionalidades del sistema por usuario y de los modelos del diseño de relaciones entre datos tanto para los desarrolladores como para los clientes y usuarios están realizados utilizando el estándar de Unified Modeling Language (UML).

Interfaz Mobile

El usuario será capaz de acceder al sistema utilizando cualquier dispositivo Android que poseea como mínimo la versión 6.0, Marshmallow, en adelante.

Interfaz Web

El usuario teniendo la posibilidad de utilizar el sistema desde la web deberá de ser capaz de hacerlo utilizando cualquiera de los navegadores más utilizados respetando las versiones, Google Chrome (Versión 84.0+), Firefox (Versión 78.10+), Edge (Versión 90.0+) y Opera (76.0.4+).

4.3. Tecnológica

El desarrollo de la aplicación debe de estar hecho utilizando

A. BackEnd

   I Componente central

      1) Java EE
      2) Wildfly
      3) Elastic Cloud

   II Servicios Externos

      1) AGESIC - Salud.uy
      2) gub.uy

B. FrontEnd

   I Web

      1) JSF
      2) JSP
      3) JavaScript

   II Mobile

      1) Android Nativo

4.4. Módulos Existentes

A. AGESIC - Salud.uy
   Es el nodo externo de la Plataforma de Interoperabilidad para obtener acceso a los distintos servicios ofrecidos por los organismos públicos y actores del área de la salud.

B. GUB.UY
   Es el nodo externo para el inicio de sesión de los usuarios y que así se encuentren identificados por una persona real.

5. Vista de Atributos de Calidad

Describe los requerimientos no funcionales del sistema en las categorías de seguridad, confiabilidad y performance.

5.1. Seguridad

- La contraseña de los usuarios se protege mediante la utilización de funciones hash con salt en su almacenamiento.
- Se utiliza HTTPS para asegurar las interacciones del componente central con los componentes periféricos, plataformas y componente móvil.
- Todos los casos de uso provistos por vacunas.uy respetan la normativa vigente referente al área de salud, seguridad de la información y privacidad, y al intercambio de información. [2]

5.2. Confiabilidad

- El uptime de Mi Nube, plataforma Elastic Cloud que utilizaremos, es de un 99.5 %.
- El sistema garantiza que ante el fallo de algún proceso crítico se mantiene un estado de consistencia.
- Se cubrirá aproximadamente un 80 % de la lógica de negocio y servicios, con pruebas unitarias, garantizando así el funcionamiento de todos los casos de uso. Se puede consultar la información detallada en el documento Calidad de código fuente, adjunto en anexos.

5.3. Performance

- Se analizarán posibles cuellos de botella y se realizarán testeos de performance para mejorar posibles puntos de quiebre del sistema. El detalle de las pruebas realizadas puede consultarse en el documento Pruebas de performance, presente en los anexos.

6. Vista Lógica

La Vista Lógica describe la arquitectura lógica de la plataforma, utilizando varios niveles de refinamiento. En particular, se presentan y describen los principales componentes lógicos de la plataforma así como sus responsabilidades, dependencias e interacciones.

6.1. Arquitectura General del Sistema

En la Figura 6 se presenta un diagrama con la arquitectura general del sistema VacunasUY.

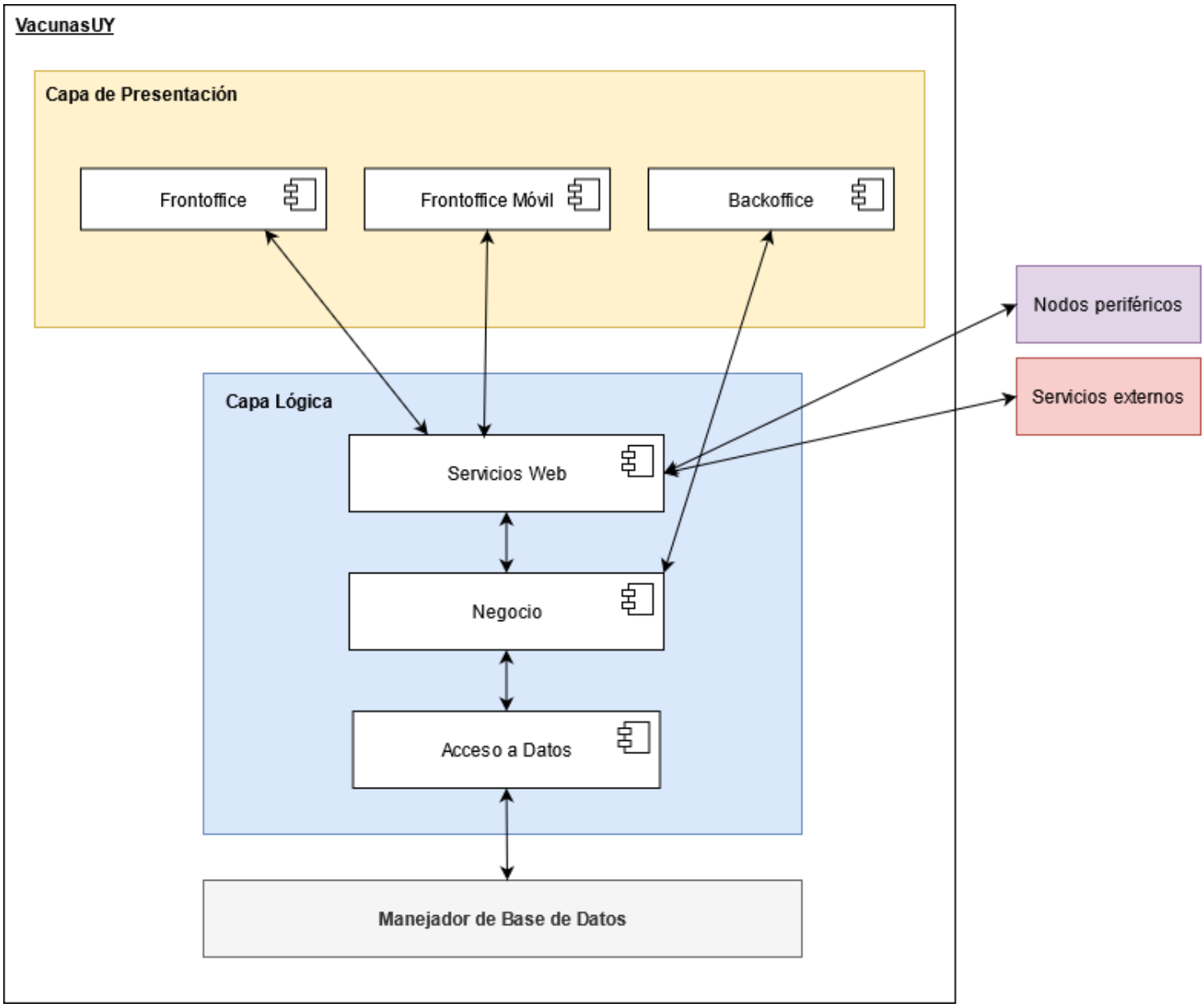

Figura 6: Arquitectura general en capas del sistema VacunasUY.

A grandes rasgos, el sistema se divide en tres capas principales, siendo Lógica la encargada de definir la lógica de negocio y brindar los servicios necesarios para interactuar con ella, así como también gestionar el acceso a la persistencia de los datos. La capa de Presentación, contiene a los componentes que realizarán la interacción con los usuarios finales. Finalmente, la capa de persistencia, es la encargada de almacenar los datos de la aplicación.

6.2. Componentes del sistema

En la Figura 7 se presentan los distintos componentes que integran las capas de la arquitectura general, y el flujo de interacción entre los mismos.

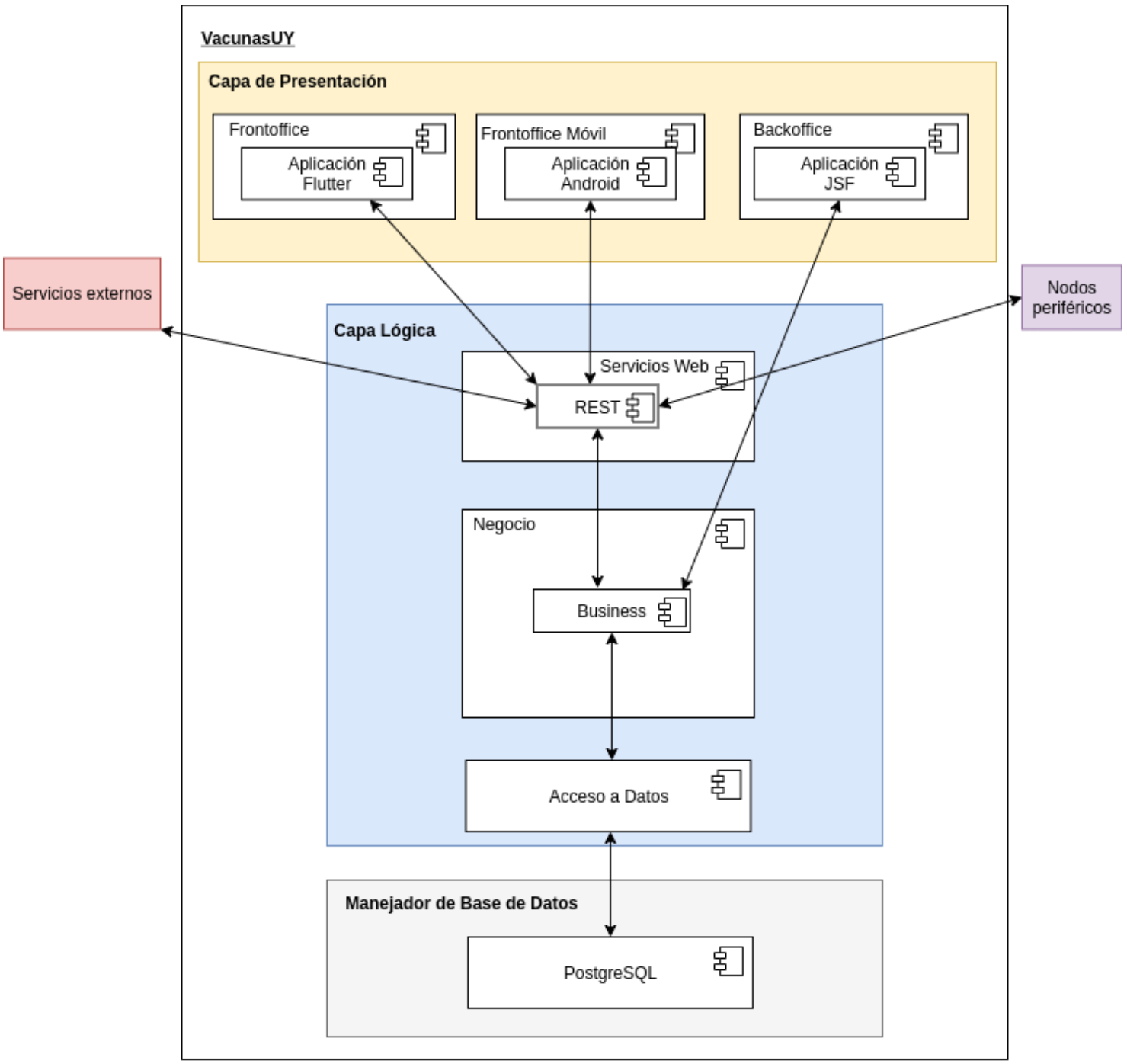

Figura 7: Arquitectura de componentes del sistema VacunasUY.

Aplicación Flutter

Este componente servirá como interfaz web para la interacción con los ciudadanos y vacunadores. Consume servicios prestados por el componente API REST.

Aplicación Android

Componente que sirve como interfaz para dispositivos móviles con sistema operativo Android para el uso de los ciudadanos. Consume servicios prestados por el componente REST.

Aplicación JSF

Este componente servirá como interfaz web para la interacción con las autoridades y administradores del sistema. Consume servicios prestados por el componente Servicios de Negocio.

REST

Ofrece servicios en formato de Web Services REST para que puedan ser consumidos por los componentes de Frontoffice tanto web como móvil.

Business

Implementa la lógica de todas las operaciones ofrecidas por esta capa. Se comunica directamente con el componente de acceso a datos para persistir la información..

Acceso a Datos

Encargado de gestionar el acceso a los datos con la capa de persistencia, ofreciendo operaciones para facilitar la interacción.

Base de datos PostgreSQL

Base de datos en la cual se guarda toda la información que gestiona el sistema.

6.3. Diagramas de secuencia

Iniciar sesión

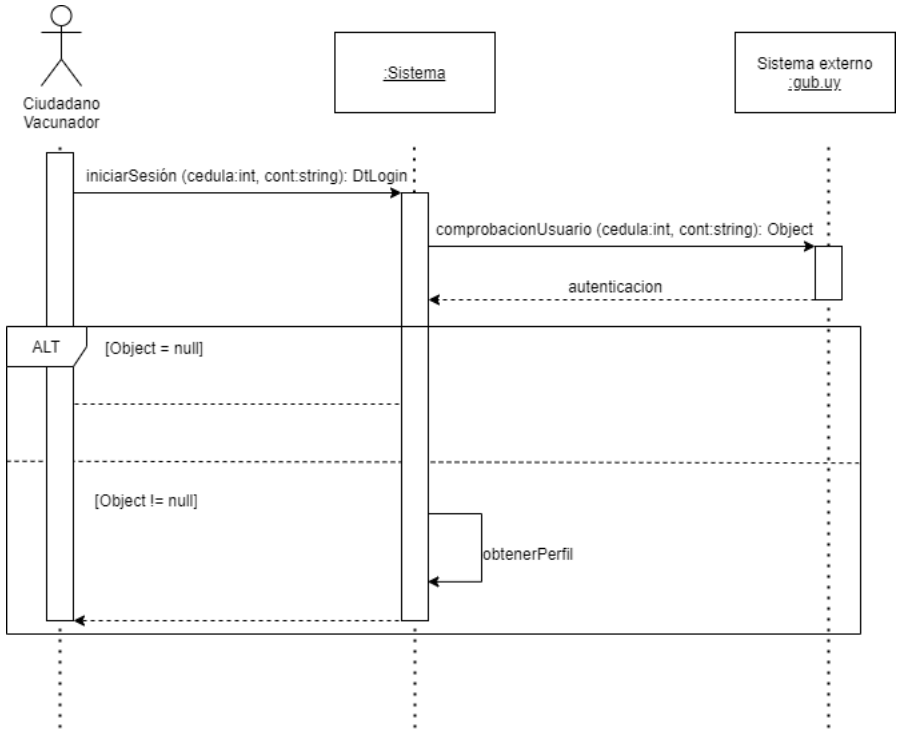

Figura 8: Diagrama de secuencia - Inicar sesión.

Alta Agenda

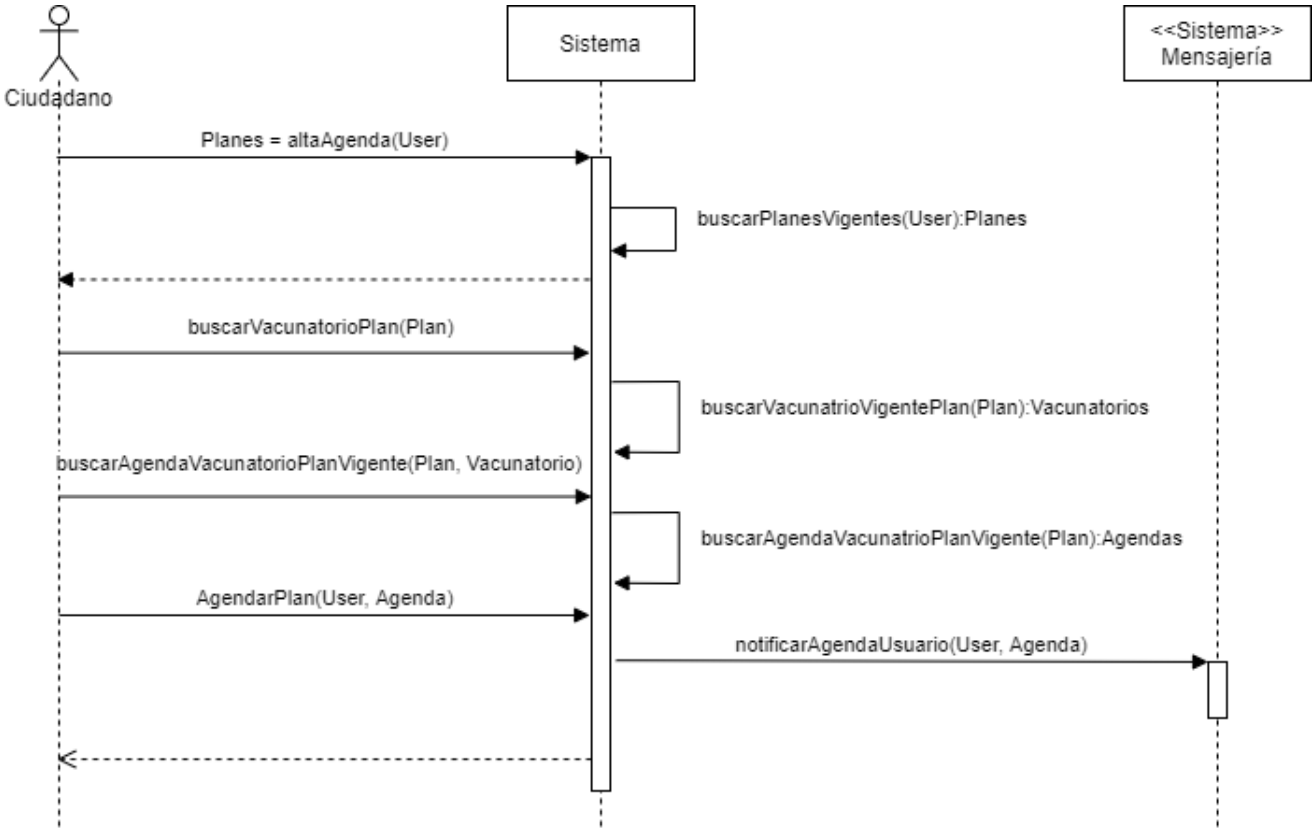

Figura 9: Diagrama de secuencia - Buscar vacunatorios cercanos.

Obtener asignación de central

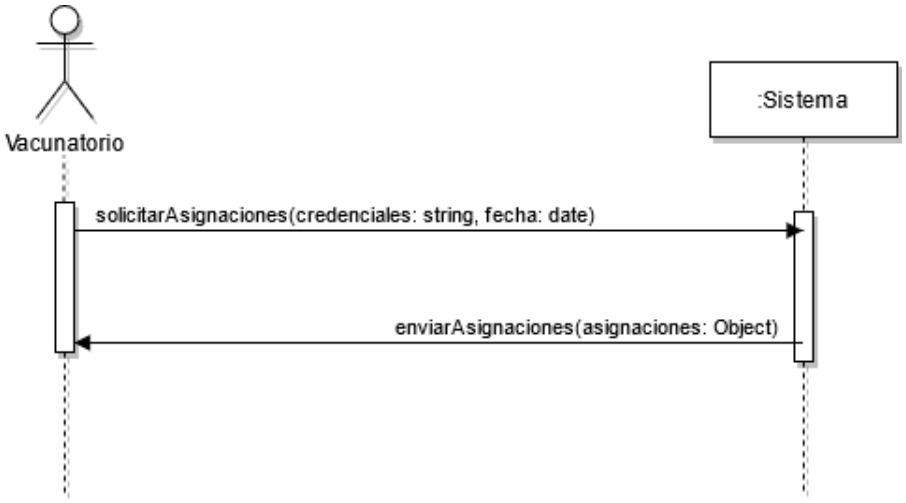

Figura 10: Diagrama de secuencia - Obtener asignación de central.

Envío estado agenda a central

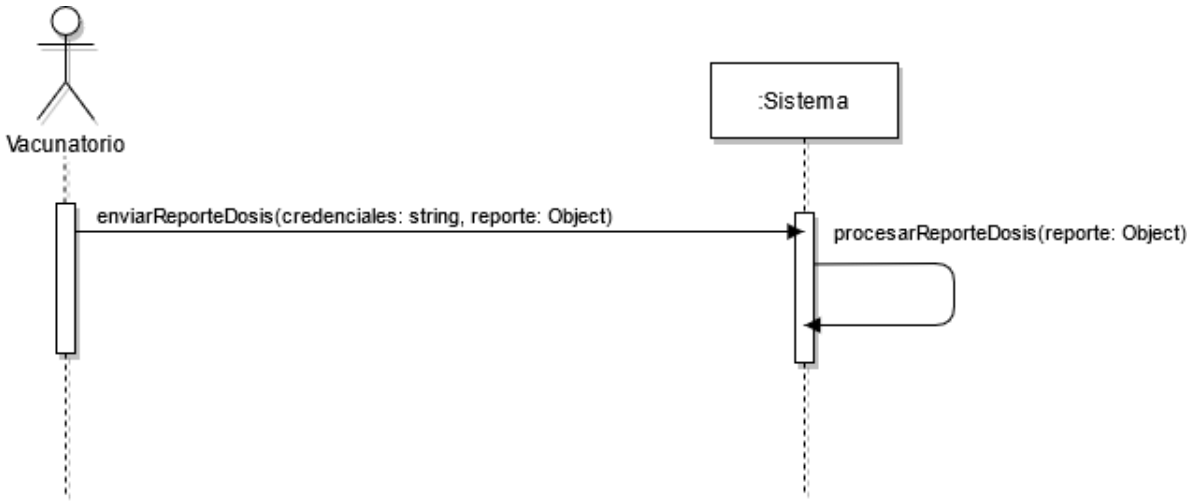

Figura 11: Diagrama de secuencia - Envío estado agenda a central.

Buscar vacunatorios cercanos

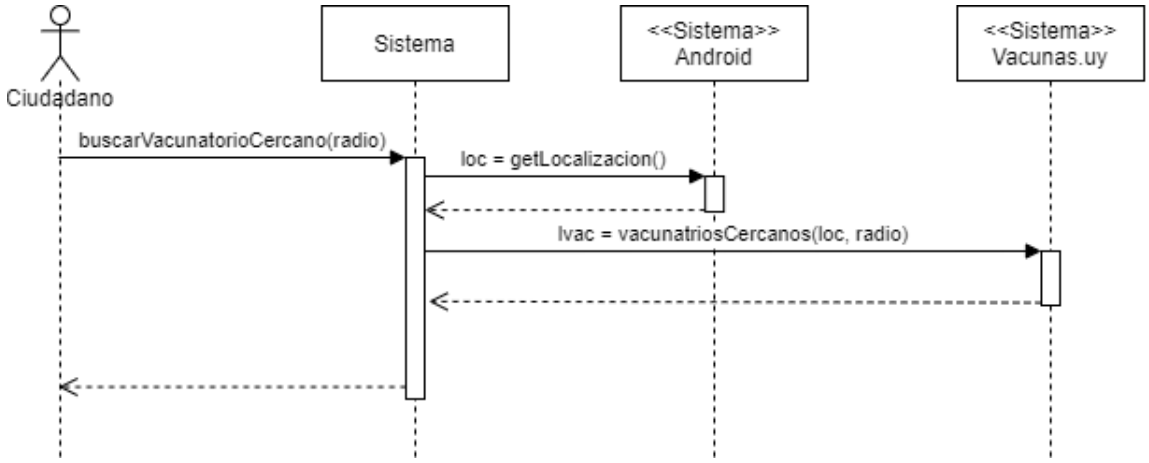

Figura 12: Diagrama de secuencia - Buscar vacunatorios cercanos.

7. Vista de Distribución

La vista de distribución muestra la arquitectura del sistema desde el punto de vista de distribución de los componentes del software, así como la relación entre ellos. Dichos componentes pueden ser tanto nodos físicos como nodos de software

La Figura 13 presenta el diagrama de despliegue para la plataforma VacunasUY.

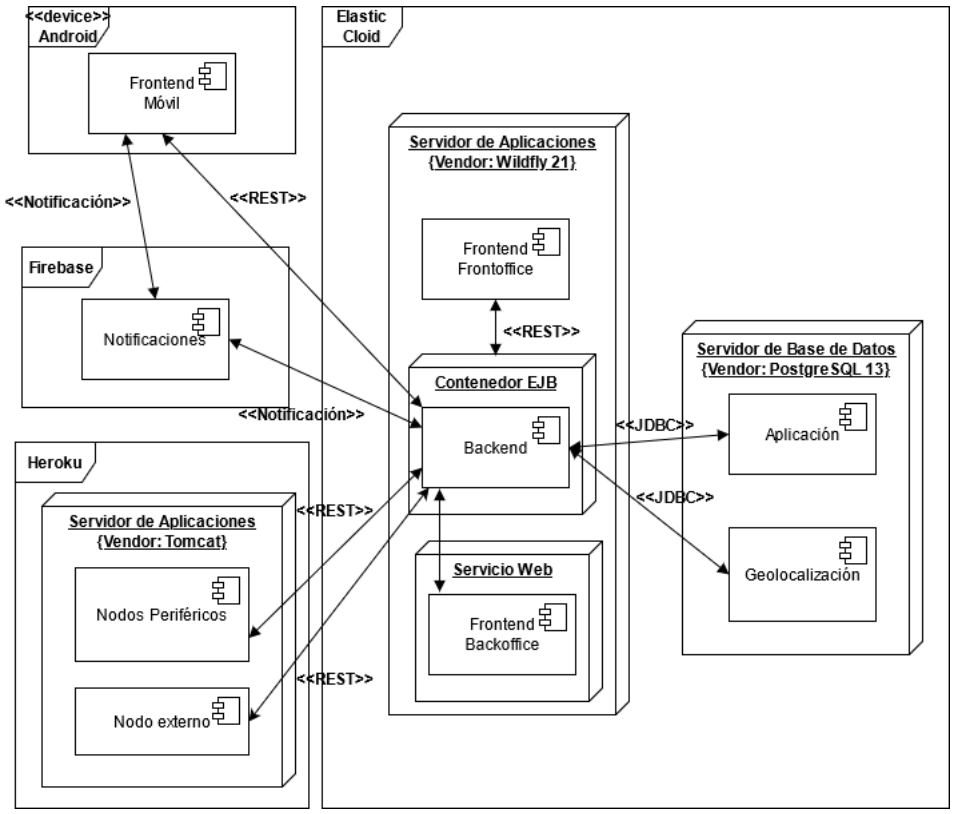

Figura 13: Diagrama de despliegue VacunasUY.

7.1. Arquitectura de despliegue

Para realizar el despliegue de la aplicación, se identificaron tres nodos que contendrán los componentes del sistema, listados en la sección 6. Vista Lógica. El primero de ellos corresponde a la plataforma Elastic Cloud de Antel, la cual ofrece servicios de despliegue de aplicaciones en distintas tecnologías. Esta será utilizada para desplegar los contenedores de aplicaciones en Wildfly 21, así como el servidor de bases de datos en PostgreSQL 13. En este servidor también se realizará el deploy del Frontoffice desarrollado en Flutter.

El segundo nodo corresponde a la plataforma Firebase ofrecida por Google. Se utiliza el servicio de notificaciones de esta plataforma para enviar notificaciones a los dispositivos Android que utilicen el sistema.

El tercer nodo corresponde a la plataforma Heroku, la cual fue utilizada para el despliegue de las aplicaciones que simularán los nodos periféricos y externos. Fueron implementados utilizando Spring Boot y corren en un servidor de aplicaciones Tomcat embebido.

Finalmente, el cuarto nodo, corresponde a dispositivos móviles que utilicen sistema operativo Android, donde se podrá instalar la aplicación desarrollada para este sistema.

En cuanto a comunicaciones, los componentes de Frontend se comunicarán con el Backend mediante servicios REST. Los nodos periféricos y externos, también utilizarán servicios REST para mantener una comunicación asincrónica con el Backend, donde dependiendo del código de estado de la solicitud se determina si se procesó correctamente la información o se necesita reintentar la operación. Finalmente, para comunicar la lógica de negocio con la persistencia, se utilizará la API JDBC.

8. Vista de Implementación

La vista de implementación se centra en los componentes en tiempo de ejecución que forman el sistema VacunasUY.

8.1. Estructura de la Aplicación

El sistema VacunasUY se desarrolla como una aplicación JavaEE. La capa de cliente consta de dos interfaces, una aplicación Web con páginas dinámicas y una aplicación móvil para Android OS.
La capa de negocio cuenta con tres JAR diferenciados, vacunasUY.jar, vacunasPDI.jar y vacunasMDB.jar.
El componete vacunasPDI.jar depende de Web Service y y el componente vacunasMDB.jar de servicios de mensagería.
A nivel de Base de Datos se debe destacar la integración con PostGIS.

8.2. Arquitectura de la Implementación

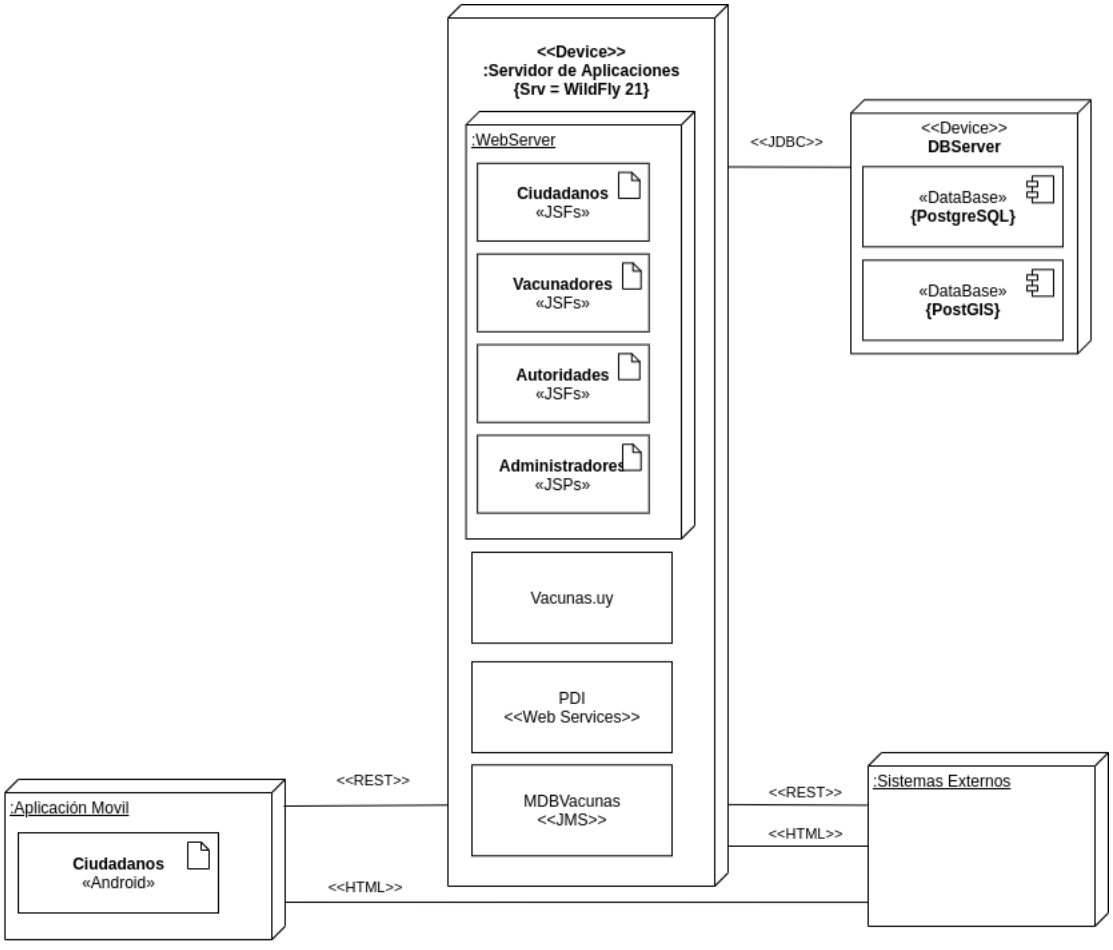

Figura 14: Diagrama de implementación VacunasUY.

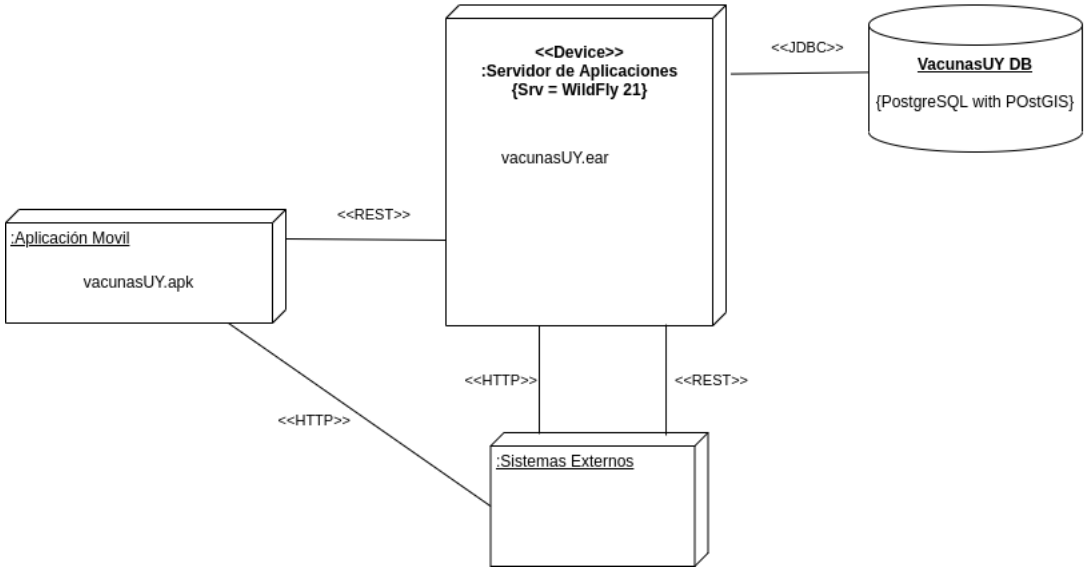

Figura 15: Diagrama de implementación resumido VacunasUY.

9. Vista de Decisiones de Arquitectura

La Vista de Decisiones de Arquitectura presenta y describe las principales decisiones de arquitectura tomadas.

Tabla 2: Decisión de Arquitectura 1

Identificador	D01
Nombre	Plataforma
Categorías	Desarrollo
Problema	La solución vacunas.uy cuenta con un componente central y un componente móvil, e interactúa con nodos periféricos y externos. Estos elementos permiten brindar funcionalidades a cuatro tipos de usuarios: ciudadanos, vacunadores, autoridades y administradores.
Alternativas	– – –
Decisión	Se implementará una solución con JEE y Android.
Justificación	Requerimiento del cliente.
Decisiones Relacionadas	D02

Tabla 3: Decisión de Arquitectura 2

Identificador	D02
Nombre	Software desarrollo
Categorías	Plataforma
Problema	Se debe definir el software que satisfaga los requerimientos del sistema a desarrollar.
Alternativas	JakartaEE PostgreSQL PostGIS Java (para Android) Kotlin (para Android)
Decisión	JakartaEE PostgreSQL PostGIS Java (para Android)
Justificación	Requerimiento del cliente. Sistema empresarial de software libre. El equipo de desarrollo conoce la herramienta.
Decisiones Relacionadas	D03, D04, D05

Tabla 4: Decisión de Arquitectura 3

Identificador	D03
Nombre	Distribución de sistema
Categorías	Plataforma
Problema	Ventajas y desventajas
Alternativas	Distribuir o no distribuir
Decisión	Se implementará un sistema en capas tanto físicas como lógicas.
Justificación	Se evalúa que un sistema distribuido favorece la tolerancia a fallos.
Decisiones Relacionadas	– – –

Tabla 5: Decisión de Arquitectura 4

Identificador	D04
Nombre	Sistema geográfico
Categorías	**Plataforma**
Problema	Se deben ubicar los vacunatorios en un mapa
Alternativas	– – –
Decisión	Se implementa un módulo en la BD que gestione Sistema de información Geográfica, PostGis.
Justificación	Se evalúa que PostGis es una buena solución libre y el equipo de desarrollo cuenta con experiencia en la herramienta.
Decisiones Relacionadas	– – –

Tabla 6: Decisión de Arquitectura 5

Identificador	D05
Nombre	Servidor de aplicaciones
Categorías	Plataforma
Problema	Determinar servidor de aplicaciones
Alternativas	GlassFish WebSphere WebLogic WildFly
Decisión	WildFly
Justificación	Requerimiento del cliente
Decisiones Relacionadas	– – –

Tabla 7: Decisión de Arquitectura 6

Identificador	D06
Nombre	Autenticación de ciudadano y vacunador
Categorías	Restricción
Problema	Se debe autenticar el ciudadano y el vacunador mediante el sistema gub.uy
Alternativas	– – –
Decisión	Se integra el sistema a la API de autenticación de gub.uy.
Justificación	Requerimiento del cliente
Decisiones Relacionadas	– – –

Tabla 8: Decisión de Arquitectura 7

Identificador	D07
Nombre	Framework Frontend
Categorías	Desarrollo
Problema	Se debe decidir el entorno de trabajo para desarrollar el frontend para el sistema.
Alternativas	Flutter Angular React
Decisión	Se decide utilizar Flutter.
Justificación	Experiencia de un desarrollador utilizando este entorno
Decisiones Relacionadas	– – –

Referencias

[1] Philippe B Kruchten. «The 4+ 1 view model of architecture». En: *IEEE software* 12.6 (1995), págs. 42-50.

[2] «Marco Jurídico - agesic». En: (). URL: https://centroderecursos.agesic.gub.uy/web/arquitectura-salud.uy/inicio/-/wiki/Arquitectura+para+Salud/Marco+Jur%5C%C3%5C%ADdico+de+la+Arquitectura+Referencia+para+Salud.

10. Anexos

10.1. Documento de Pruebas de performance

En este documento se detallan las pruebas de performance realizadas sobre la plataforma VacunasUYs y se documentan sus resultados.

10.2. Documento de Calidad de código fuente

Este documento especifica el proceso de calidad estudiado para el código fuente de la aplicación.