

# Documento de Arquitectura de Software

## Grupo 09 - 2021

Taller de Sistemas Empresariales  
Taller de Sistemas de Información Java EE

Nicolás San Martín 5.055.041-9

Matías Borggio 4.903.997-1

Ignacio Otero 5.266.715-7

Agustín Ruíz Díaz 5.042.744-4

Bruno Pardiñas 5.205.075-6

Marzo 2021

# Tabla de Contenido

<b>1. Introducción</b>	<b>3</b>
1.1. Objetivo del Documento . . . . .	3
1.2. Representación de la Arquitectura . . . . .	3
1.3. Partes Interesadas (i.e. <i>stakeholders</i> ) . . . . .	3
1.4. Organización del Documento . . . . .	5
<b>2. Vista Conceptual</b>	<b>6</b>
2.1. Descripción General de la Plataforma . . . . .	6
2.2. Modelo Conceptual . . . . .	7
<b>3. Vista de Casos de Uso</b>	<b>8</b>
3.1. Diagrama de Casos de Uso . . . . .	8
<b>4. Vista de Restricciones</b>	<b>21</b>
4.1. Normativas . . . . .	21
4.2. Estándares . . . . .	21
4.3. Tecnología . . . . .	21
4.4. Sistemas existentes . . . . .	22
4.5. Otras . . . . .	22
<b>5. Vista de Atributos de Calidad</b>	<b>23</b>
5.1. Usabilidad . . . . .	23
5.2. Confiabilidad . . . . .	23
5.3. Performance . . . . .	23
5.4. Seguridad . . . . .	23
<b>6. Vista Lógica</b>	<b>24</b>
6.1. Arquitectura General del Sistema . . . . .	24
6.2. Refinamiento - Capa de Negocio . . . . .	24
6.3. Refinamiento - Capa de acceso a datos . . . . .	26
6.4. Refinamiento - Capa de servicios . . . . .	26
6.5. Refinamiento - Capa de Presentación . . . . .	28
6.6. Diagramas de Interacción . . . . .	30
<b>7. Vista de Distribución</b>	<b>35</b>
7.1. Escenario 1 . . . . .	35
7.2. Escenario 2 . . . . .	36
<b>8. Vista de Implementación</b>	<b>37</b>
8.1. Arquitectura de Implementacion . . . . .	37
8.2. Front-End . . . . .	37
<b>9. Vista de Decisiones de Arquitectura</b>	<b>38</b>

# 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][2]. En particular, las vistas utilizadas para representar la arquitectura de la plataforma vacunas.uy son:

1. **Vista de Casos de Uso:** Presenta el modelo del dominio y describe los procesos de negocio más significativos. Presenta los actores y los casos de uso para el sistema.
2. **Vista de Restricciones:** Describe restricciones tecnológicas, normativas, uso de estándares, entre otros, las cuales deben ser respetadas tanto por el proceso de desarrollo como por el producto desarrollado.
3. **Vista de Atributos de Calidad:** Incluye aspectos de calidad, y describe los requerimientos no-funcionales del sistema.
4. **Vista Lógica:** Describe la arquitectura del sistema presentando varios niveles de refinamiento. Indica los módulos lógicos principales, sus responsabilidades y dependencias.
5. **Vista de Distribución:** Presenta aspectos físicos como topología, infraestructura informática, e instalación de ejecutables. Incluye plataformas y software de base.
6. **Vista de Implementación:** Describe los componentes de deployment contruidos y 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. **Clientes** Ciudadanos Uruguayos que deseen vacunarse.
2. **Autoridades** Gestionar las agendas y las dosis, además de obtener información cuantitativa de los registros a vacunas.
3. **Administradores** Gestionar usuarios, roles y nodos periféricos.
4. **Vacunadores** Trabajadores de los vacunatorios que desean registrar u obtener información sobre la vacunación del vacunatorio.

5. **Socios Logísticos** Instituciones asociadas que ofrecen servicios de envío y almacenaje para las vacunas.
6. **Desarrolladores** Encargados de crear y distribuir el software.
7. **Arquitecto** Encargados de manejar la estructura, funcionamiento e interacción entre las partes del software.
8. **Mantenimiento** Encargados de la modificación de un producto de software después de la entrega, para corregir errores, mejorar el rendimiento, u otros atributos.

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 Calidad de Servicio	Vista Lógica	Vista de Implementación	Vista de Distribución	Vista de Decisiones de Arquitectura	Vista Conceptual
Ciudadanos	✓		✓					✓
Autoridades	✓		✓					✓
Administradores	✓		✓					✓
Vacunadores	✓		✓					✓
Desarrolladores		✓	✓	✓	✓	✓	✓	✓
Arquitectos	✓	✓	✓	✓	✓	✓	✓	✓
Mantenimiento			✓	✓	✓	✓		✓

## 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

La solución vacunas.uy cuenta con un componente central y un componente móvil. Interactúa con nodos externos (PDI y Salud.uy) y periféricos (Vacunatorios y Socios logísticos). La plataforma brindará servicios a los ciudadanos, autoridades y vacunadores. Además contará con administradores para asegurar el funcionamiento esperado.

Los ciudadanos interactuarán con la plataforma a través de un frontoffice web y de un componente móvil. Se podrá obtener información de la ejecución del plan de vacunación, gestionar su agenda de vacunación y consultar los vacunatorios disponibles. La aplicación móvil notifica al ciudadano y permite obtener certificados de vacunación.

Las autoridades podrán tener acceso a las cuestiones administrativas de los planes de vacunaciones, sus etapas, las agendas disponibles, las dosis, coordinación de envío de estas, así como asignación de los vacunadores. Además tendrán disponible reportes de datos recabados por la plataforma sobre las agendas e interacciones de los ciudadanos.

Los administradores podrán gestionar los usuarios, roles y nodos periféricos, como por ejemplo vacunatorios o socios logísticos.

Los vacunadores obtendrán información de su agenda de vacunación, como los vacunatorios y puesto de vacunación asignados.

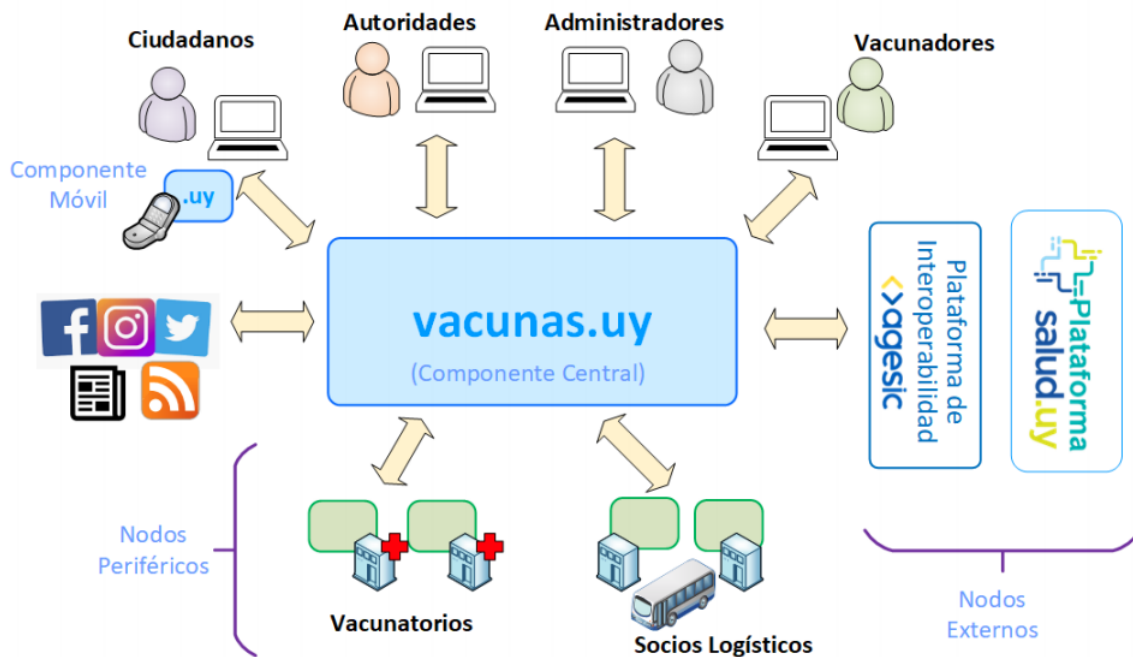


Figura 1: Descripción General de la Plataforma

## 2.2. Modelo Conceptual

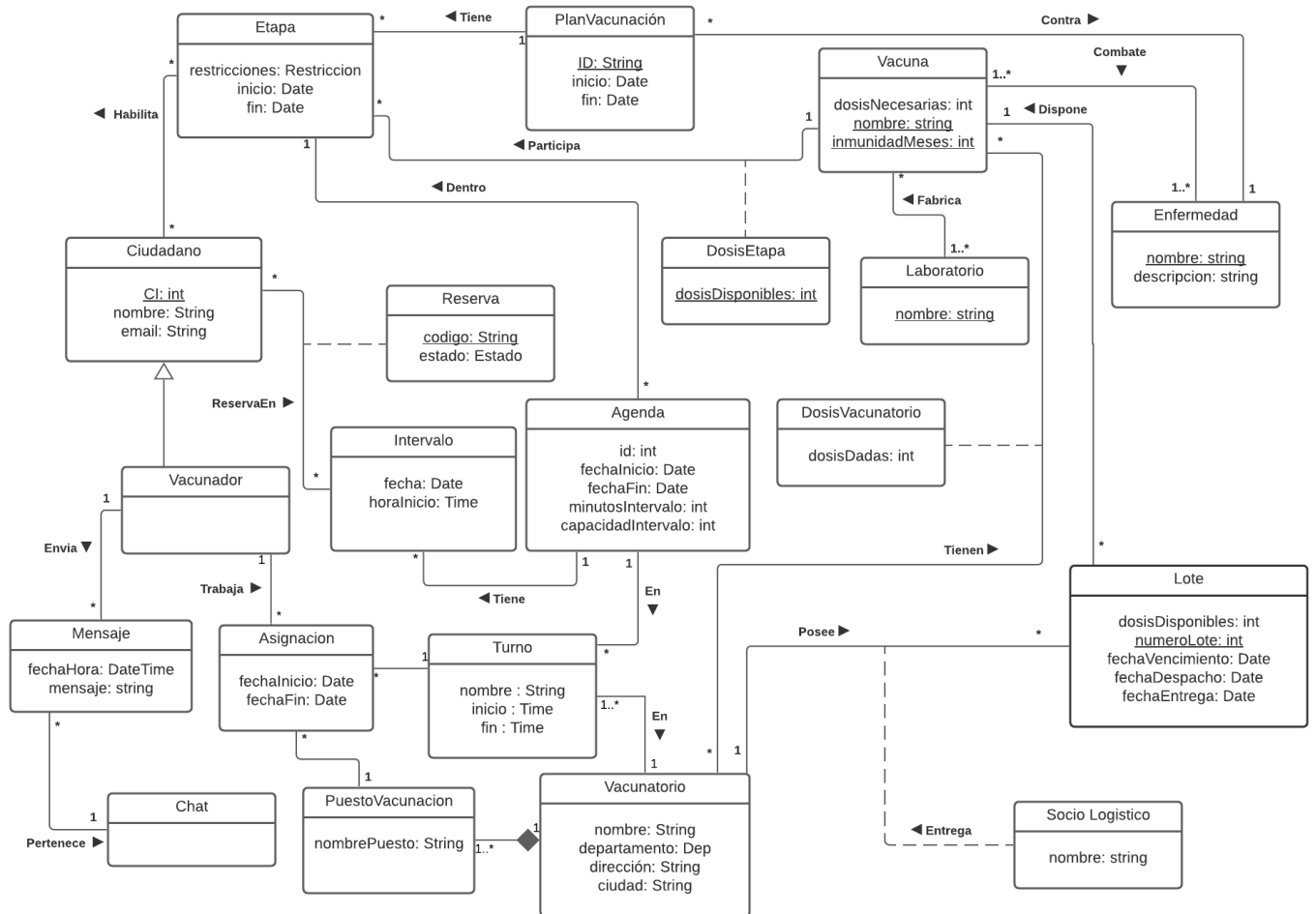


Figura 2: Modelo de dominio

### 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. 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, indicando cuáles se consideran críticos para la arquitectura.

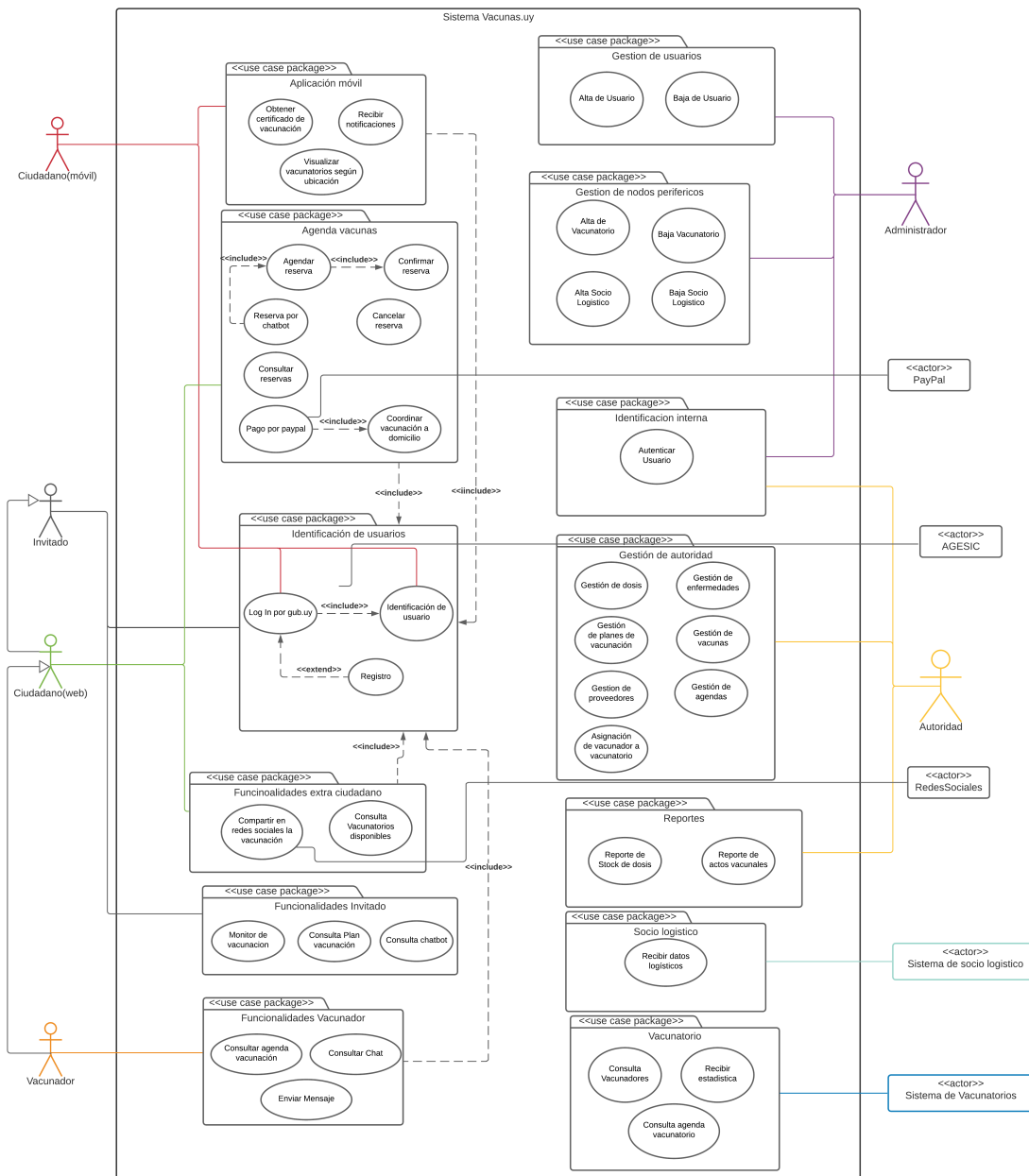


Figura 3: Diagrama de casos de uso



<b>Nombre</b>	Inicio de sesión por gub.uy
<b>Actores</b>	Invitado
<b>Pre-condiciones</b>	<ol style="list-style-type: none"> <li>1. El invitado no debe estar identificado.</li> </ol>
<b>Descripción</b>	Este caso de uso describe la interacción entre un invitado y el sistema, de modo que este se pueda reconocer como 'Ciudadano' o como 'Vacunador'. Las tareas que este usuario pueda realizar dentro del sistema dependerán de dicha autenticación, la cual se realiza mediante gub.uy.
<b>Curso típico de eventos</b>	<ol style="list-style-type: none"> <li>1. El caso de uso comienza cuando el actor invitado desea iniciar sesión en el sistema.</li> <li>2. El invitado ingresa sus datos (cédula y contraseña).</li> <li>3. Los datos ingresados se validan mediante gub.uy.</li> <li>4. La aplicación identifica al usuario e inicia con las funciones disponibles de acuerdo a los privilegios del invitado que inició sesión. Si se trata de un vacunador tiene disponible mediante una pestaña donde se exponen las funcionalidades de vacunador, en caso de un ciudadano la pestaña no se muestra.</li> <li>5. Fin del caso de uso.</li> </ol>
<b>Flujos alternativos</b>	<p>3A Los datos ingresados son incorrectos.</p> <ol style="list-style-type: none"> <li>1. Retorna 2</li> </ol> <p>3B Es la primera vez que es invitado inicia sesión en el sistema, por lo cual debe registrarse.</p> <ol style="list-style-type: none"> <li>1. Extiende al caso de uso Registro".</li> <li>2. Retorna 4.</li> </ol>
<b>Post-condiciones</b>	<ol style="list-style-type: none"> <li>1. El usuario queda autenticado en el sistema. La aplicación brinda distintas funcionalidades al usuario dependiendo de sus privilegios, es decir, dependiendo de si es vacunador o no.</li> </ol>

<b>Nombre</b>	Registro
<b>Actores</b>	Invitado
<b>Pre-condiciones</b>	<ol style="list-style-type: none"> <li>1. Es la primera vez que el actor Invitado inicia sesión en el sistema.</li> <li>2. El ciudadano hizo un login válido en gub.uy.</li> </ol>
<b>Descripción</b>	Este caso de uso describe la interacción entre un invitado y el sistema, de modo que este pueda ingresar información personal para su perfil de usuario.
<b>Curso típico de eventos</b>	<ol style="list-style-type: none"> <li>1. El caso de uso comienza cuando el actor invitado inicia sesión por primera vez en el sistema.</li> <li>2. El invitado ingresa y confirma su número de email.</li> <li>3. Los datos ingresados se validan.</li> <li>4. Fin del caso de uso.</li> </ol>
<b>Flujos alternativos</b>	<p>3A Los datos ingresados son incorrectos/inválidos.</p> <ol style="list-style-type: none"> <li>1. Retorna 2</li> </ol>
<b>Post-condiciones</b>	<ol style="list-style-type: none"> <li>1. Los datos ingresados por el usuario quedan registrados en su perfil. No es necesario volver a registrarse en el sistema al momento de iniciar sesión.</li> </ol>

<b>Nombre</b>	Agendar Reserva
<b>Actores</b>	Ciudadano
<b>Pre-condiciones</b>	El usuario debe haber iniciado sesión
<b>Descripción</b>	El usuario verifica su habilitación y agenda fecha y hora para vacunarse
<b>Curso típico de eventos</b>	<ol style="list-style-type: none"> <li>1. El usuario ingresa a la funcionalidad agendar reserva.</li> <li>2. El sistema lista las enfermedades registradas.</li> <li>3. El usuario selecciona una.</li> <li>4. El sistema valida la habilitación del usuario verificando que exista alguna etapa de algún plan de vacunación para el que cumpla las restricciones.</li> <li>5. El sistema lista los departamentos disponibles para la vacunación.</li> <li>6. El usuario selecciona un departamento.</li> <li>7. El sistema despliega una lista de las agendas (agrupadas por vacunatorio) dentro de ese departamento con cupo disponible para la fecha seleccionada de ese departamento.</li> <li>8. El sistema lista las fechas disponibles dentro las etapas de los planes de vacunación a los que esta habilitado, en un calendario.</li> <li>9. El usuario selecciona una fecha de las listadas.</li> <li>10. El usuario confirma su selección.</li> <li>11. El sistema muestra el intervalo asignado a la reserva realizada.</li> <li>12. Fin del CU.</li> </ol>
<b>Flujos alternativos</b>	<p>2A El usuario que no esta habilitado para ninguna etapa de algún plan de vacunación.</p> <ol style="list-style-type: none"> <li>1 El sistema notifica al usuario que no existe plan para el cual este habilitado.</li> <li>2 Fin del CU.</li> </ol> <p>5A La fecha seleccionada ya no esta disponible.</p> <ol style="list-style-type: none"> <li>1 El sistema notifica que la fecha ya no esta disponible.</li> <li>2 Retorna 3.</li> </ol> <p>G1 El usuario cancela la operación.</p> <ol style="list-style-type: none"> <li>1 Fin del CU.</li> </ol>

<b>Nombre</b>	Alta usuario
<b>Actores</b>	Administrador
<b>Pre-condiciones</b>	
<b>Descripción</b>	El administrador desea dar de alta un usuario nuevo
<b>Curso típico de eventos</b>	<ol style="list-style-type: none"> <li>1. El caso de uso comienza cuando el Administrador desea dar de alta a un usuario</li> <li>2. El administrador indica si el usuario a crear es un administrador o autoridad</li> <li>3. El administrador indica los datos del usuario(Email,Nombre, Contraseña)</li> <li>4. El administrador confirma el alta</li> <li>5. El sistema marca al usuario como creado</li> </ol>
<b>Flujos alternativos</b>	<p>3A Ya existe un usuario con el mismo nombre.</p> <ol style="list-style-type: none"> <li>1. Retorna 2</li> </ol> <p>3B Ya existe un usuario con el mismo email.</p> <ol style="list-style-type: none"> <li>1. Retorna 2</li> </ol> <p>G1 El administrador cancela la operación.</p> <ol style="list-style-type: none"> <li>1. Fin CU</li> </ol>

<b>Nombre</b>	Autenticar Usuario
<b>Actores</b>	Administrador, Autoridad
<b>Descripción</b>	Este caso de uso describe la interacción entre un usuario y el sistema, de modo que este se pueda reconocer como 'Autoridad' o como 'Administrador'. Las tareas que este usuario pueda realizar dentro del sistema dependerán de dicha autenticación
<b>Curso típico de eventos</b>	<ol style="list-style-type: none"> <li>1. El caso de uso comienza cuando un Administrador o Autoridad desea iniciar sesión en el sistema.</li> <li>2. El usuario ingresa sus datos (email y contraseña).</li> <li>3. El sistema valida los datos ingresados.</li> <li>4. Se identifica el usuario.</li> <li>5. La aplicación inicia con las funciones disponibles de acuerdo a los privilegios del usuario que inició sesión.</li> <li>6. Fin del caso de uso.</li> </ol>
<b>Flujos alternativos</b>	<p>3A Los datos ingresados son incorrectos.</p> <ol style="list-style-type: none"> <li>1. Retorna 2</li> </ol>

<b>Nombre</b>	Alta de Vacunatorio
<b>Actores</b>	Administrador
<b>Pre-condiciones</b>	
<b>Descripción</b>	El Administrador desea dar de alta un nuevo Vacunatorio
<b>Curso típico de eventos</b>	<ol style="list-style-type: none"> <li>1. El caso de uso comienza cuando el Administrador desea dar de alta un vacunatorio</li> <li>2. El Administrador ingresa los datos del vacunatorio (Nombre, Departamento, Ciudad, Dirección)</li> <li>3. Sistema checkea disponibilidad de nombre de vacunatorio.</li> <li>4. El sistema persiste los datos del vacunatorio creado</li> </ol>
<b>Flujos Alternativos</b>	<p>3A Ya existe un vacunatorio con el mismo Nombre:</p> <ol style="list-style-type: none"> <li>1 Retorna 2</li> </ol> <p>G1 El administrador cancela el alta:</p> <ol style="list-style-type: none"> <li>1 Fin de caso de uso</li> </ol>

<b>Nombre</b>	Alta de Socio Logístico
<b>Actores</b>	Administrador
<b>Pre-condiciones</b>	
<b>Descripción</b>	El Administrador da de de alta un nuevo Socio Logístico
<b>Curso típico de eventos</b>	<ol style="list-style-type: none"> <li>1. El caso de uso comienza cuando el Administrador desea dar de alta un Socio logístico</li> <li>2. El Administrador ingresa los datos del Socio Logístico (Nombre)</li> <li>3. Sistema checkea disponibilidad de nombre de Socio logístico.</li> <li>4. El sistema persiste los datos del Socio logístico creado.</li> </ol>
<b>Flujos Alternativos</b>	<p>3a Ya existe un Socio logístico con el mismo Nombre:</p> <ol style="list-style-type: none"> <li>1 Retorna 2</li> </ol> <p>G1 El administrador cancela el alta:</p> <ol style="list-style-type: none"> <li>1 Fin de caso de uso</li> </ol>

<b>Nombre</b>	Consultar chat
<b>Actores</b>	Vacunador
<b>Pre-condiciones</b>	El usuario debe haber iniciado sesión
<b>Descripción</b>	El usuario entra a la sala de chat de vacunadores
<b>Curso típico de eventos</b>	<ol style="list-style-type: none"> <li>1. El usuario ingresa al chat de vacunadores.</li> <li>2. El sistema lista los mensajes enviados al chat ordenados por fecha y hora.</li> <li>3. Fin del CU.</li> </ol>
<b>Flujos alternativos</b>	<p>G1 El usuario abandona la sala chat</p> <ol style="list-style-type: none"> <li>1 Fin del CU.</li> </ol>
<b>Post-condiciones</b>	El mensaje queda registrado en el sistema

<b>Nombre</b>	Enviar mensaje
<b>Actores</b>	Vacunador
<b>Pre-condiciones</b>	El usuario debe haber iniciado sesión y estar en la sala de chat
<b>Descripción</b>	El usuario envía un mensaje a la sala de chat de vacunadores
<b>Curso típico de eventos</b>	<ol style="list-style-type: none"> <li>1. El usuario escribe un mensaje en el campo de texto.</li> <li>2. El usuario confirma el envío del mensaje.</li> <li>3. Fin del CU.</li> </ol>
<b>Flujos alternativos</b>	<p><i>G1</i> El usuario abandona la sala chat</p> <ol style="list-style-type: none"> <li>1 Fin del CU.</li> </ol>
<b>Post-condiciones</b>	-



<b>Nombre</b>	Asignación de vacunador a vacunatorio
<b>Actores</b>	Autoridad
<b>Pre-condiciones</b>	El usuario autoridad debe haber iniciado sesión
<b>Descripción</b>	El usuario asigna una jornada laboral de un vacunador a un vacunatorio del sistema
<b>Curso típico de eventos</b>	<ol style="list-style-type: none"> <li>1. El usuario ingresa a la funcionalidad asignar vacunador.</li> <li>2. El sistema lista los vacunatorios con vacantes a ser ocupadas.</li> <li>3. El usuario selecciona un vacunatorio</li> <li>4. El sistema lista los vacunadores disponibles para el vacunatorio seleccionado.</li> <li>5. El usuario selecciona un vacunador.</li> <li>6. El sistema muestra los detalles de la asignación.</li> <li>7. El usuario confirma la asignación.</li> <li>8. El sistema persiste la asignación del vacunador al vacunatorio.</li> <li>9. Fin del CU.</li> </ol>
<b>Flujos alternativos</b>	<p>3A El sistema notifica que no hay vacunadores disponibles.</p> <p>1 Retorna 2.</p> <p>G1 El usuario cancela la operación</p> <p>1 Fin del CU.</p>
<b>Post-condiciones</b>	Se registra la asignación en el sistema

<b>Nombre</b>	Enviar información de dosis suministradas
<b>Actores</b>	Sistema de vacunatorios
<b>Pre-condiciones</b>	
<b>Descripción</b>	Se desean enviar los datos de dosis de vacunas suministradas desde el nodo de los vacunatorios hacia el componente central
<b>Curso típico de eventos</b>	<ol style="list-style-type: none"> <li>1. El sistema de vacunatorios informa que desea enviar la información.</li> <li>2. El sistema envía los datos sobre las dosis suministradas (Ciudadanos vacunados, dosis utilizadas) a una cola de mensajes.</li> <li>3. El componente central consume el mensaje de la cola.</li> <li>4. El componente central procesa los datos.</li> <li>5. Fin del CU.</li> </ol>
<b>Flujos alternativos</b>	<p><i>G1</i> El usuario cancela la operación</p> <ol style="list-style-type: none"> <li>1 Fin del CU.</li> </ol>
<b>Post-condiciones</b>	

<b>Nombre</b>	Recibir agenda de vacunación
<b>Actores</b>	Vacunatorio
<b>Pre-condiciones</b>	El vacunatorio debe estar registrado en el sistema
<b>Descripción</b>	El nodo Vacunatorio solicita su agenda de vacunación
<b>Curso típico de eventos</b>	<ol style="list-style-type: none"> <li>1. El vacunatorio envía una solicitud de consulta de agenda al sistema central con la ultima fecha de modificación.</li> <li>2. El sistema central envía la agenda para el vacunatorio solicitante si la fecha de modificación es posterior a la enviada por el vacunatorio.</li> <li>3. Fin del CU.</li> </ol>
<b>Flujos alternativos</b>	<p>2A La fecha de modificación enviada por el vacunatorio es la misma que la que posee el sistema central</p> <ol style="list-style-type: none"> <li>1 El sistema central notifica que no hubo cambios en la agenda.</li> <li>2 Fin del CU.</li> </ol> <p>2B El sistema central no se encuentra disponible.</p> <ol style="list-style-type: none"> <li>1 El vacunatorio no actualiza su base de datos local.</li> <li>2 Fin del CU.</li> </ol>
<b>Post-condiciones</b>	El vacunatorio actualiza su agenda y su fecha de modificación

<b>Nombre</b>	Enviar datos logísticos de socio
<b>Actores</b>	Socio logístico
<b>Pre-condiciones</b>	
<b>Descripción</b>	Ocurre un evento a comunicar (p. ej. recepción por parte del transportista, entrega de un lote de dosis a un vacunatorio) el sistema del socio logístico envía el evento al sistema. El evento es procesado por el sistema almacenando los datos correspondientes a este.
<b>Curso típico de eventos</b>	<ol style="list-style-type: none"> <li>1. El sistema del socio logístico envía un evento al sistema.</li> <li>2. El sistema recibe el evento.</li> <li>3. El sistema procesa el evento.</li> <li>4. Fin del CU.</li> </ol>
<b>Post-condiciones</b>	

## 4. Vista de Restricciones

La Vista de Restricciones describe restricciones que deben respetarse tanto en el proceso de desarrollo como en el producto [2].

### 4.1. Normativas

El sistema se debe adecuar a la ley N° 19.179 Ley de Software Libre y Formatos Abiertos rigiente.

El sistema se debe adecuar a la ley N° 18.331 Ley de Protección de Datos Personal.

Los mecanismos a proveer por vacunas.uy deben respetar la normativa vigente de privacidad de datos, en particular, en el área de salud.

### 4.2. Estándares

La comunicación entre el componente central y los nodos externos debe realizarse mediante Web Services SOAP.

La comunicación entre el componente móvil y el componente central debe realizarse mediante Web Services REST.

La interfaz de usuario debe estar orientada tanto a web como móvil. Debe ser posible visualizar el contenido en cualquiera de los browsers mas conocidos, por ejemplo: Google Chrome, Mozilla Firefox, Microsoft Edge, Opera, Safari, etc.

Las interacciones del componente central con los componentes periféricos, plataformas y componente móvil deben utilizando HTTPS.

Proteger la contraseña de usuarios utilizando funciones hash con salt en su almacenamiento.

### 4.3. Tecnología

El desarrollo completo de la aplicación debe estar realizado en la plataforma Java EE 8 o 7. El lenguaje de programación debe ser Java.

Se deberá utilizar Maven 3.6.3 para la gestión y configuración de proyectos.

El sistema de pagos a través de la web debe ser realizado utilizando Paypal.

El componente central debe ejecutarse en la solución Elastic Cloud de ANTEL.

Utilización de una base de datos NoSQL para la persistencia de una parte de los datos del sistema.

El control de versiones debera ser abordado con la herramienta gitlab

El Inicio de sesión por parte de los usuarios se realizará mediante gub.uy

#### **4.4. Sistemas existentes**

El sistema debe comunicarse con sistemas externos de los socios logísticos a través del envío de eventos al sistema central.

#### **4.5. Otras**

La fecha de entrega final del proyecto es el 21 de Junio de 2021.

Se apunta a tener un cubrimiento de aproximadamente 80 % de la lógica del sistema con pruebas automatizadas y al menos una prueba por cada servicio.

## **5. Vista de Atributos de Calidad**

La Vista de Atributos de Calidad se centra en los requerimientos de atributos de calidad de la plataforma.

### **5.1. Usabilidad**

El sistema debe ser intuitivo para que los ciudadanos mayores, o con menor experiencia en uso de dispositivos electrónicos, puedan utilizarla.

### **5.2. Confiabilidad**

La comunicación entre el componente central y los componentes de los nodos periféricos puede verse interrumpida de manera temporal sin que esto implique pérdida de funcionalidad a dichos nodos.

### **5.3. Performance**

Los tiempos de carga de las páginas, tanto web como móvil, no sobrepasaran los 5s (tiempo promedio de carga según Pingdom).

La aplicación debe ser escalable tanto vertical como horizontalmente.

### **5.4. Seguridad**

Proteger la contraseña de usuarios utilizando funciones hash con salt en su almacenamiento.

Las interacciones del componente central con los componentes periféricos, plataformas y componente móvil se realizan a través de HTTPS.

## 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

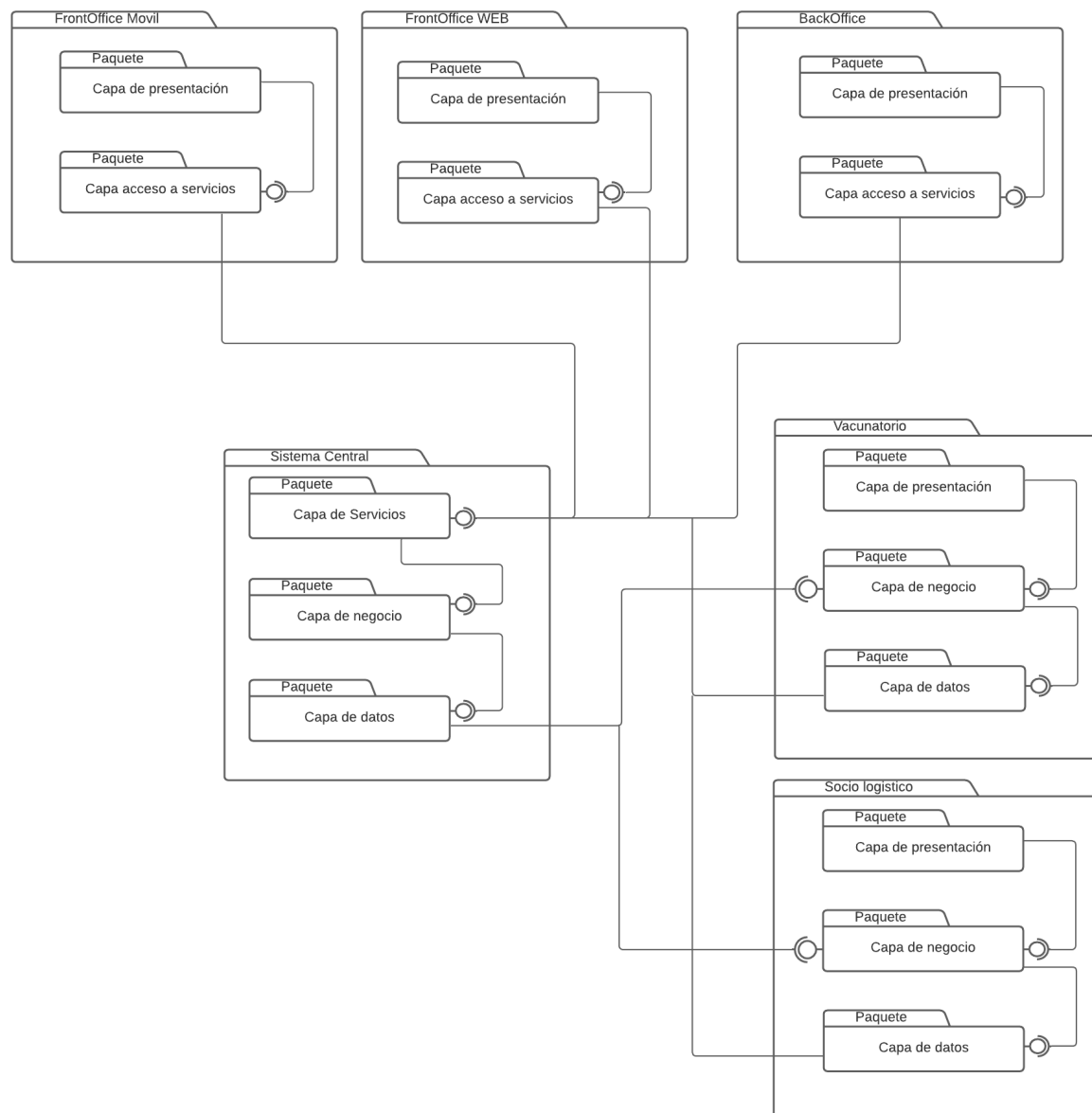


Figura 4: Refinamiento de la arquitectura general

### 6.2. Refinamiento - Capa de Negocio



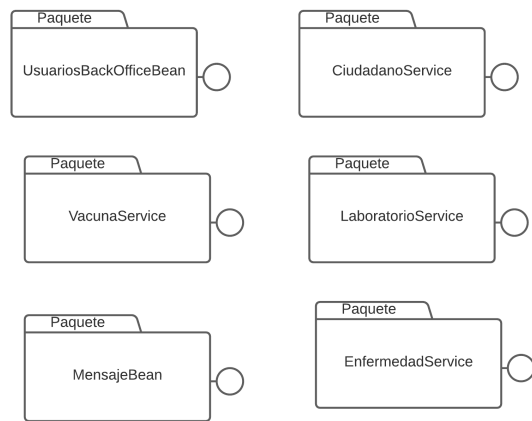


Figura 5: Interfaces a la lógica de negocio

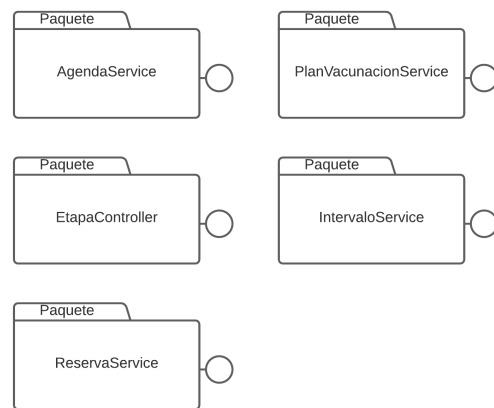


Figura 6: Interfaces a la lógica de negocio

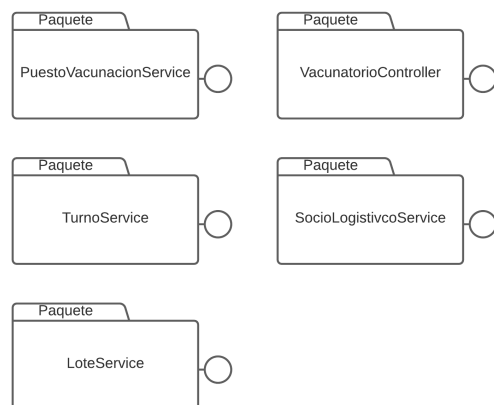


Figura 7: Interfaces a la lógica de negocio

### 6.3. Refinamiento - Capa de acceso a datos

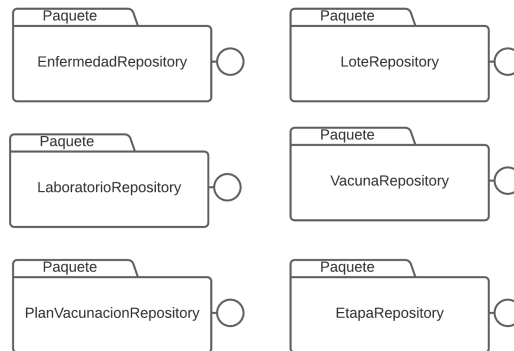


Figura 8: Repositorios(acceso a datos)

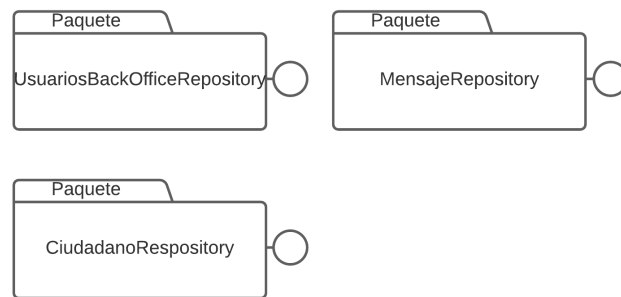


Figura 9: Repositorios(acceso a datos)

### 6.4. Refinamiento - Capa de servicios

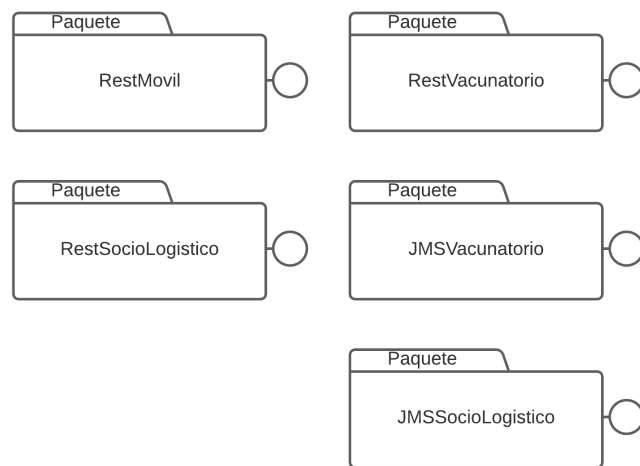


Figura 10: Servicios

## 6.5. Refinamiento - Capa de Presentación

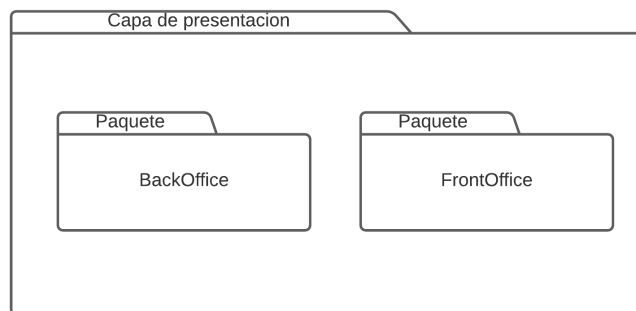


Figura 11: Capa de presentación

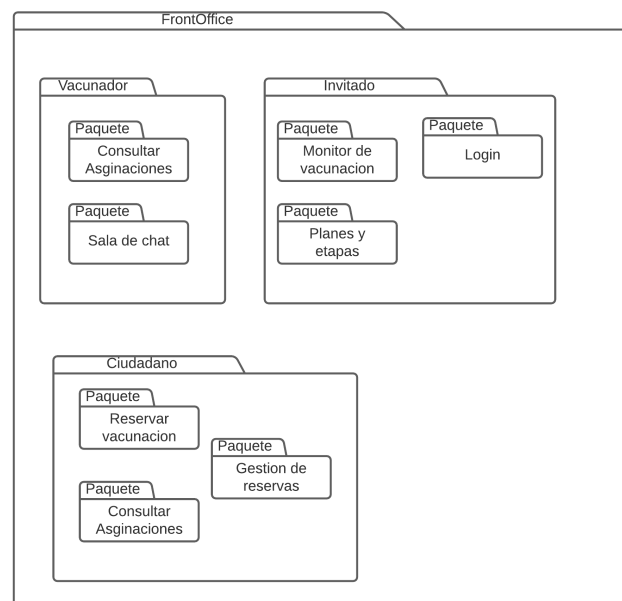


Figura 12: FrontOffice(presentación)

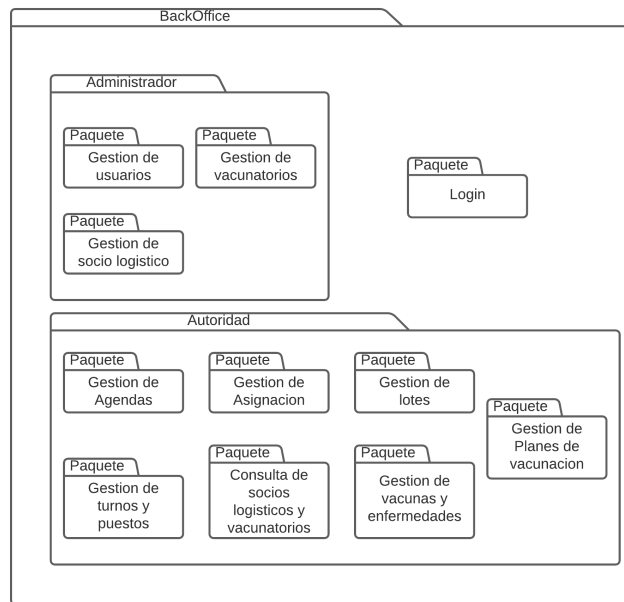


Figura 13: BackOffice(presentación)

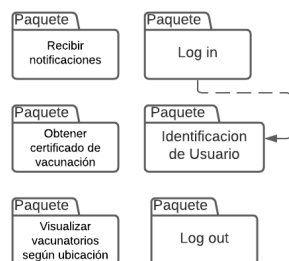


Figura 14: BackOffice(presentación)

## 6.6. Diagramas de Interacción

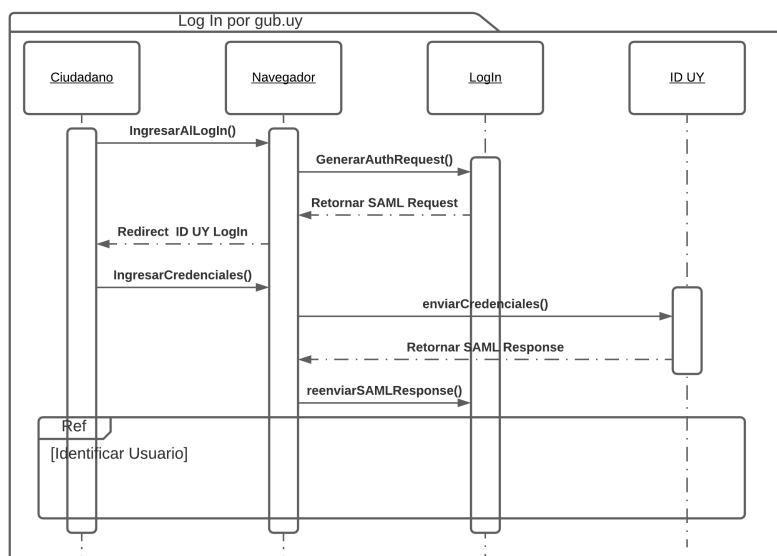


Figura 15: DSS Log In

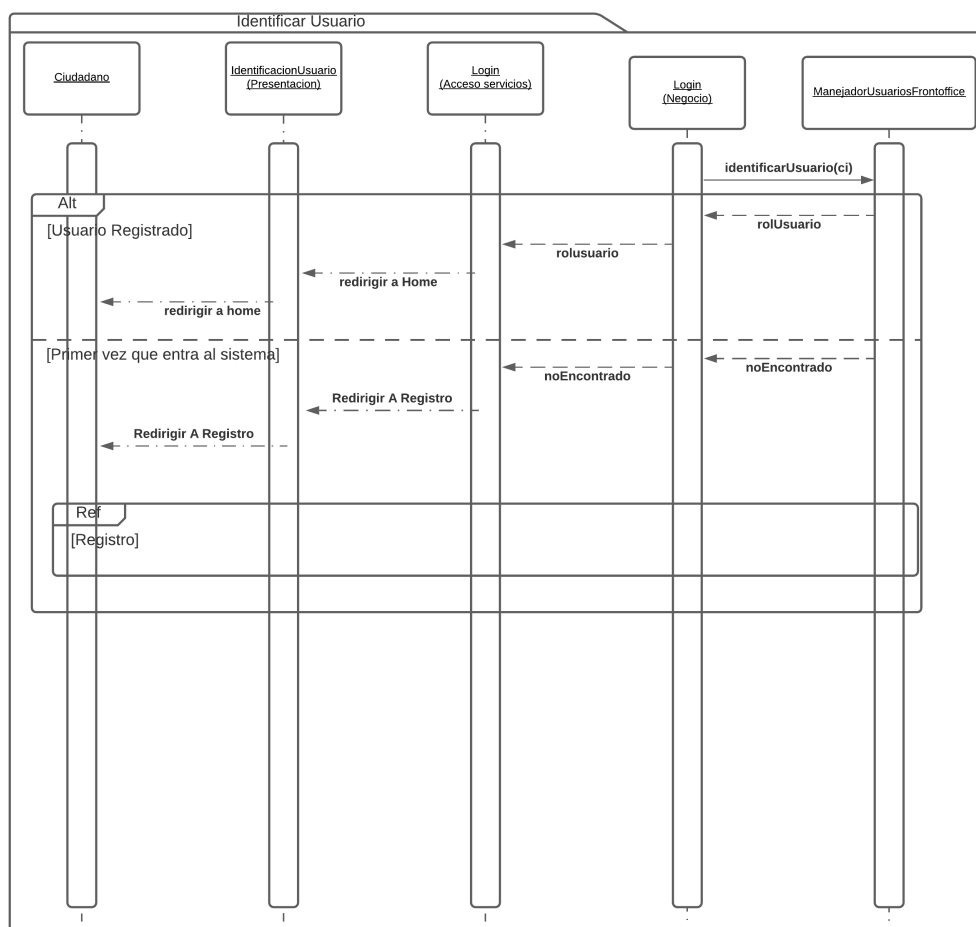


Figura 16: DSS Identificar usuario

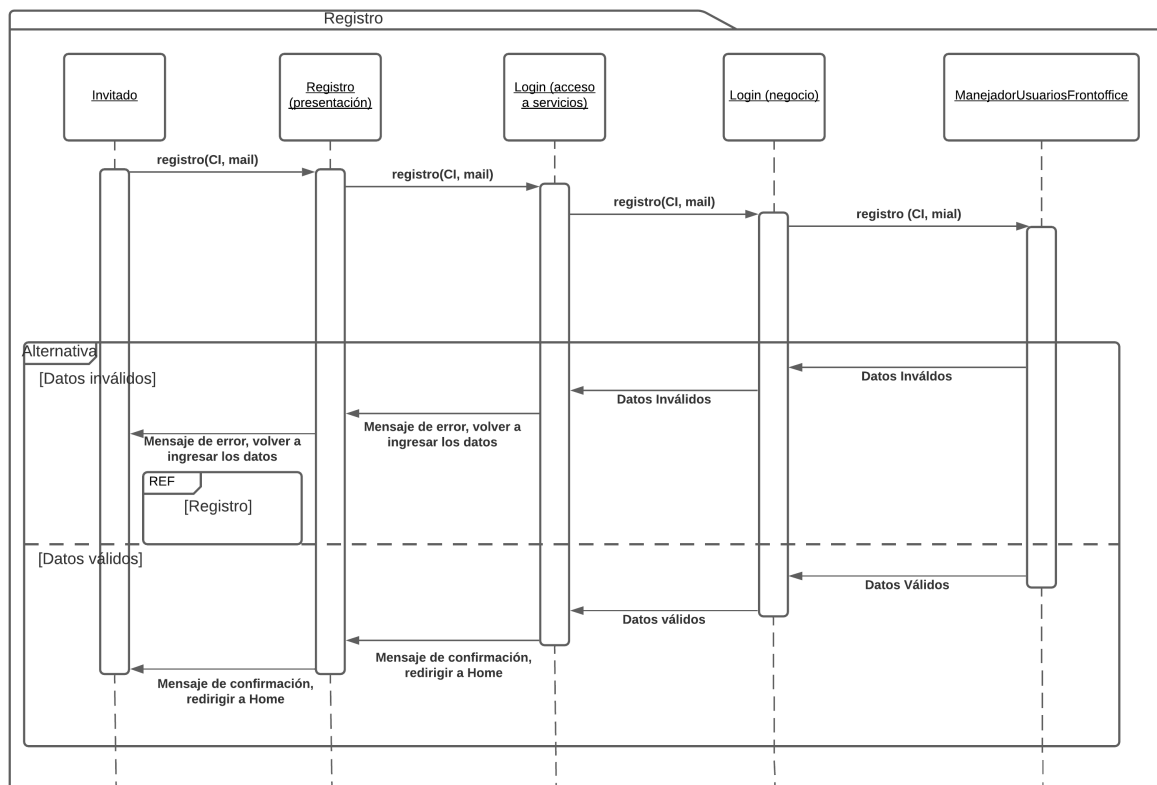


Figura 17: DSS Registro

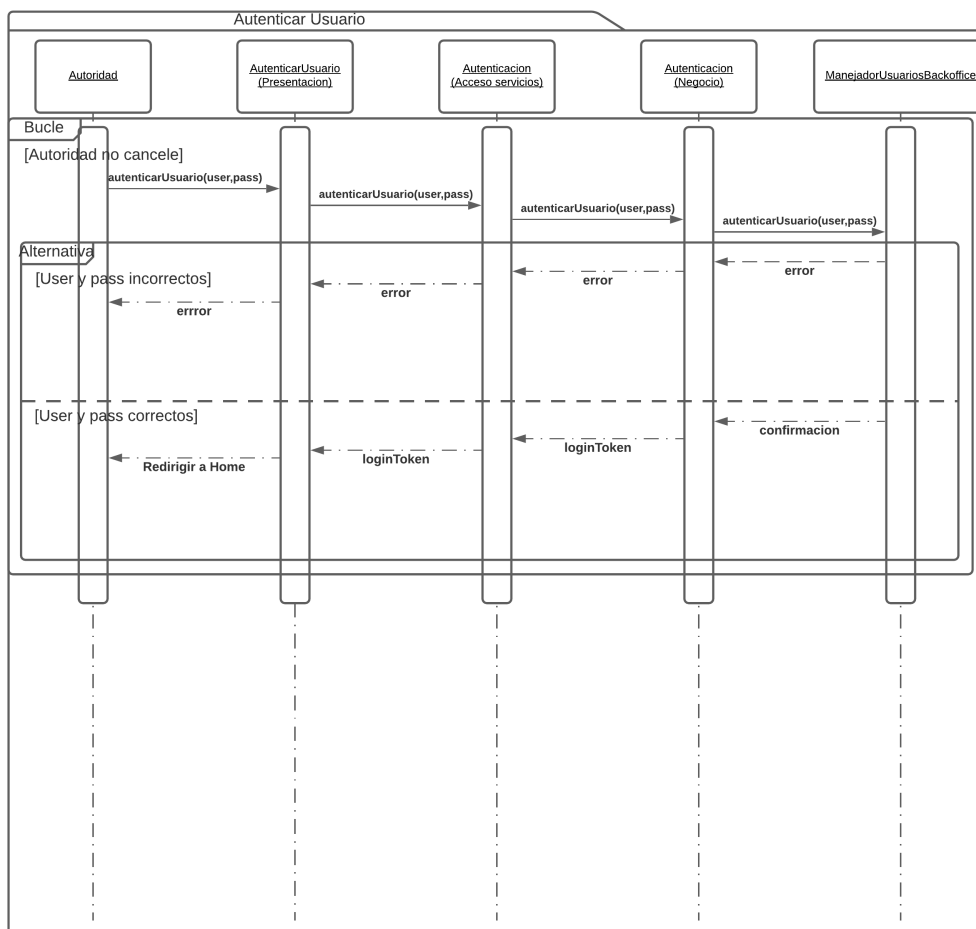


Figura 18: DSS Autenticar usuario



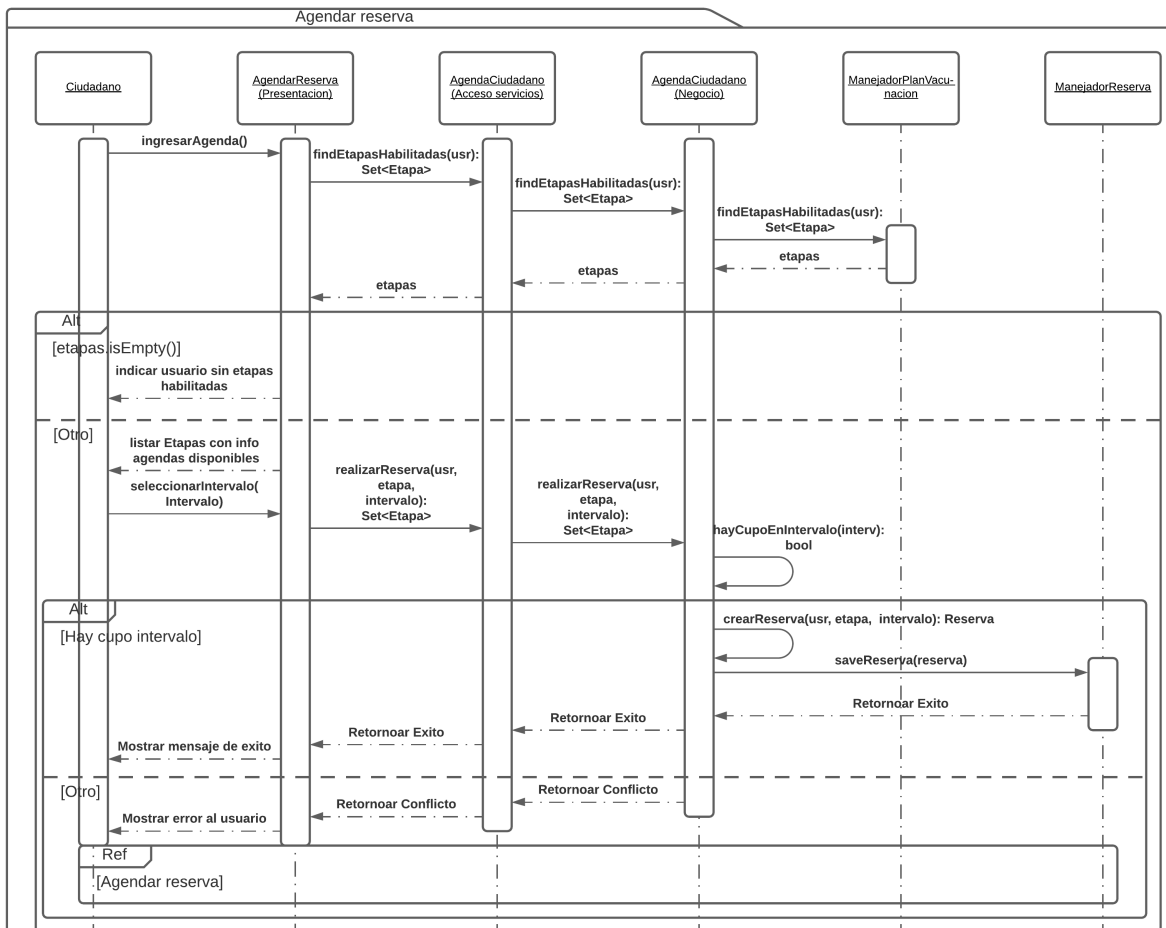


Figura 19: DSS Agendar Reserva

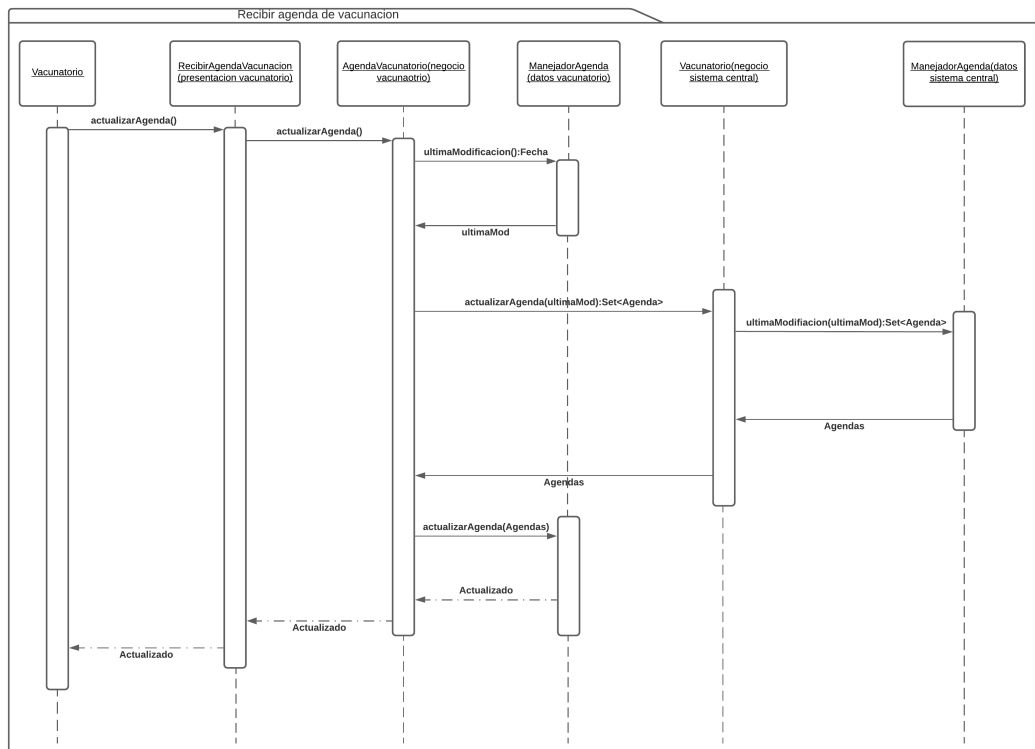


Figura 20: DSS Recibir agenda de vacunacion

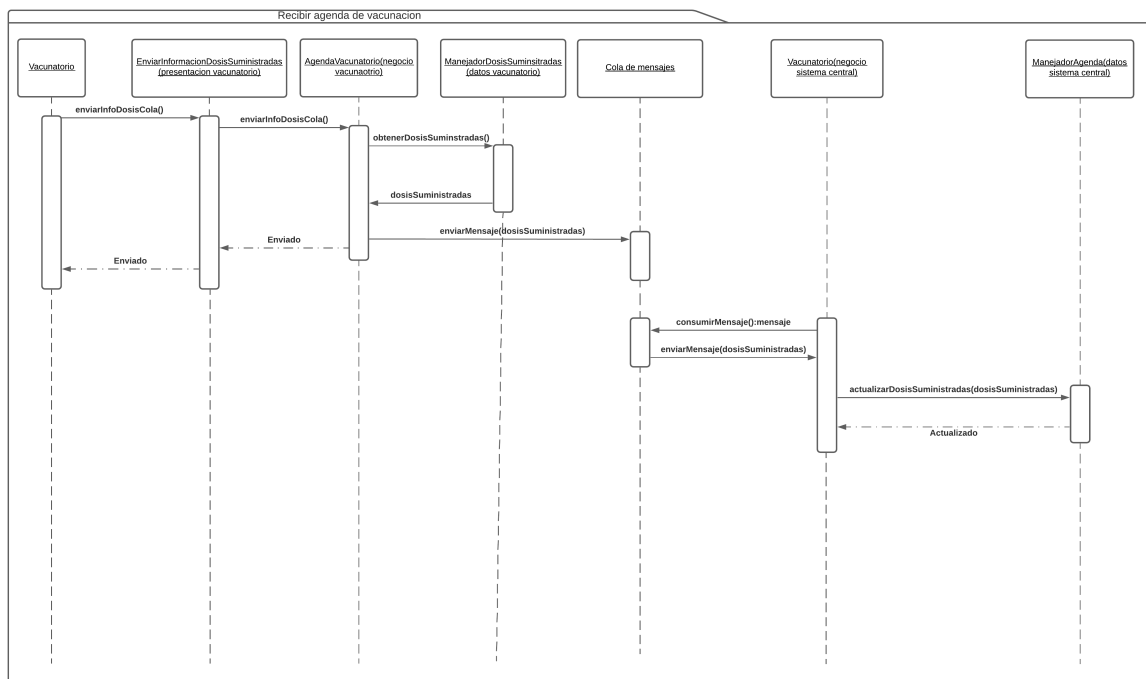


Figura 21: DSS enviar informacion de dosis suministradas

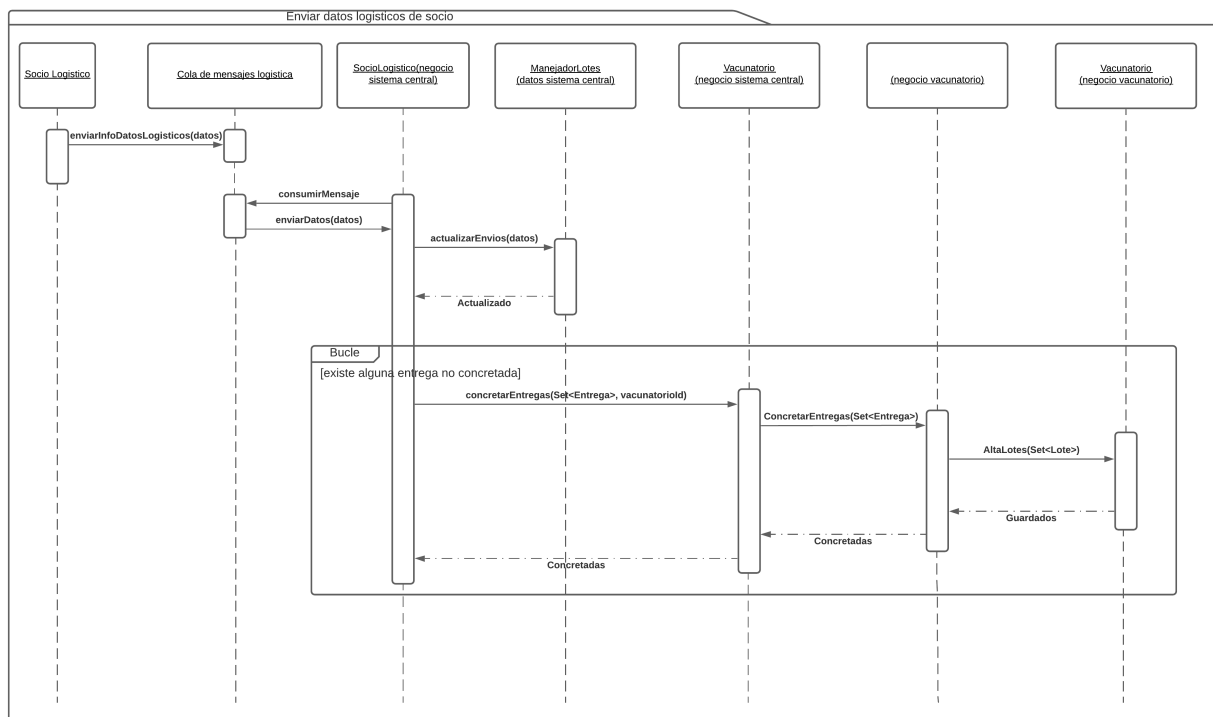
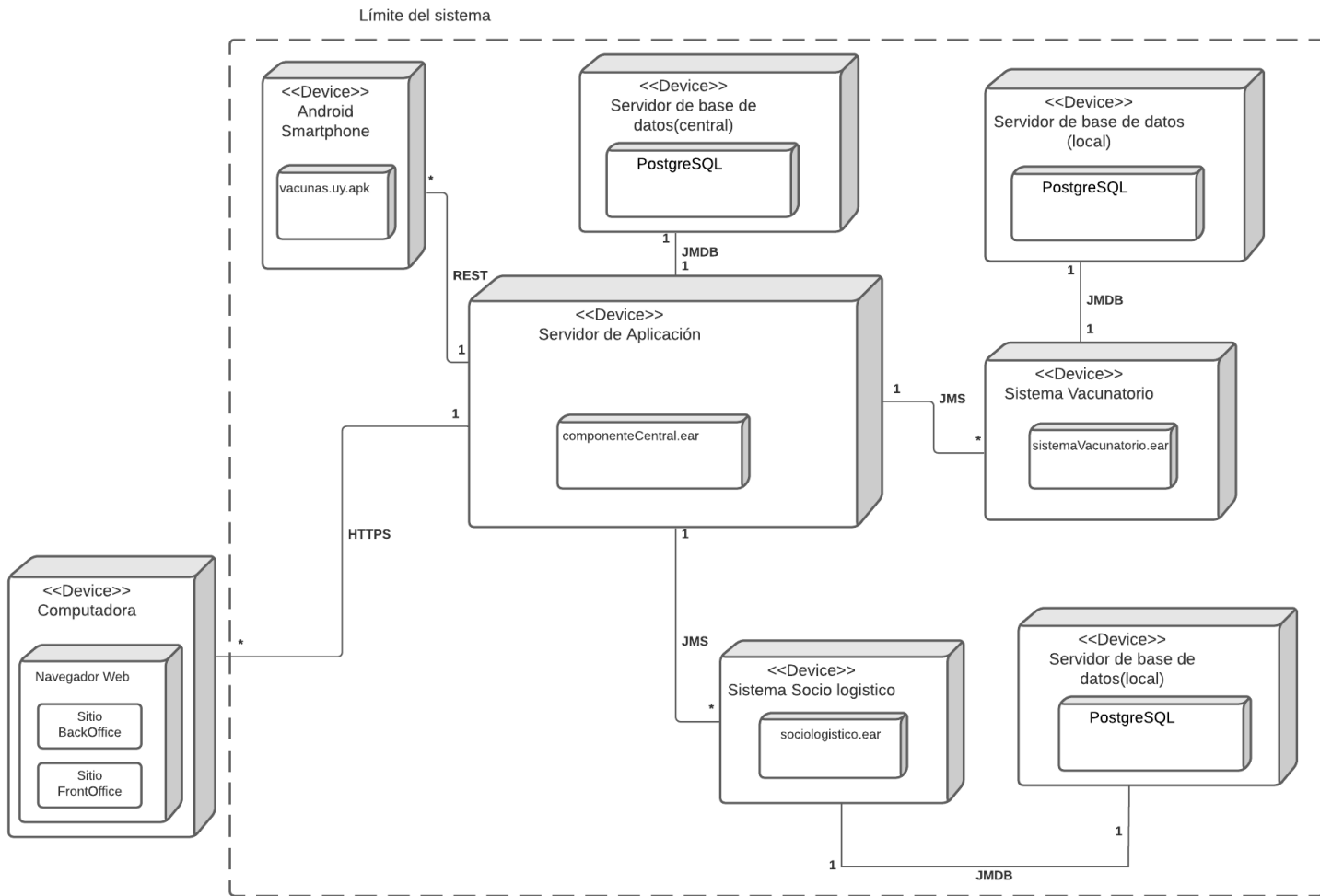


Figura 22: DSS Enviar datos logísticos de socio

## 7. Vista de Distribución

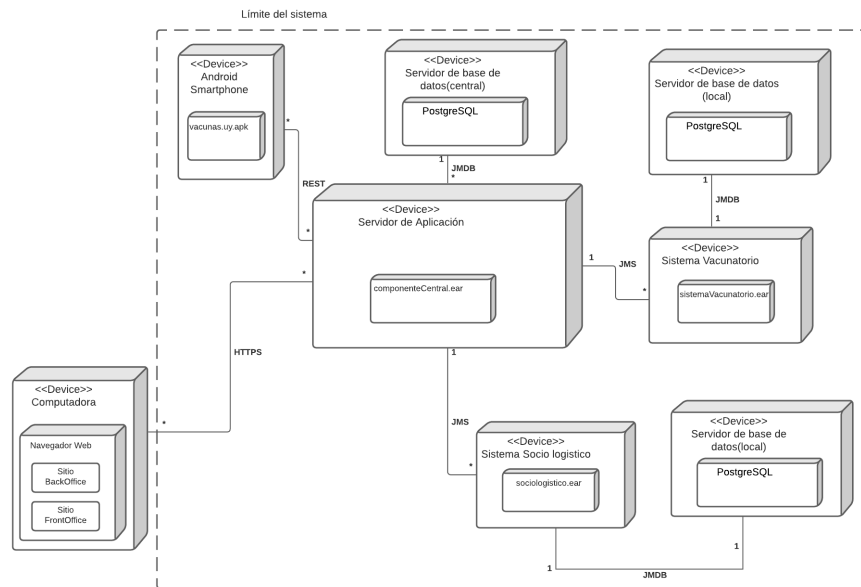
En la vista de distribución o *deployment* se plantean escenarios de distribución de los componentes en *tiers* o nodos. Se presentarán 2 diferentes escenarios.

### 7.1. Escenario 1



Este es un modelo de la distribución con un único servidor de aplicación. Al ejecutar un único contenedor de la aplicación, esta no está escalada horizontalmente y soportará menos requests, pero será más barato.

## 7.2. Escenario 2

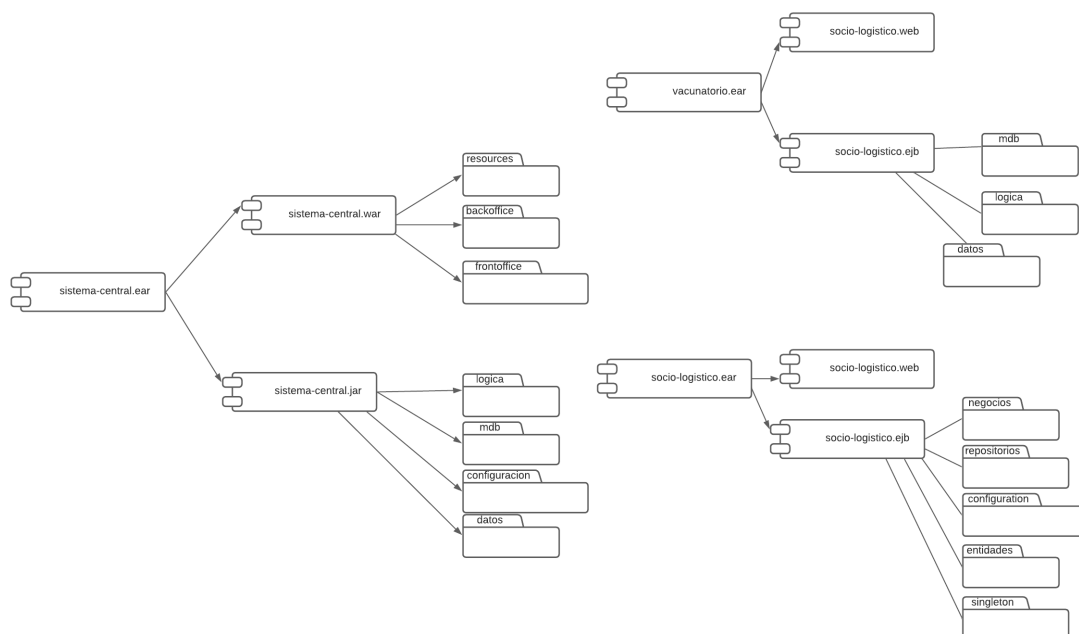


Este modelo de distribución tiene varios servidores de aplicación. Es ideal para poder soportar más usuarios web, con una muy buena escalabilidad. Tiene como desventaja potenciales mayores costos.

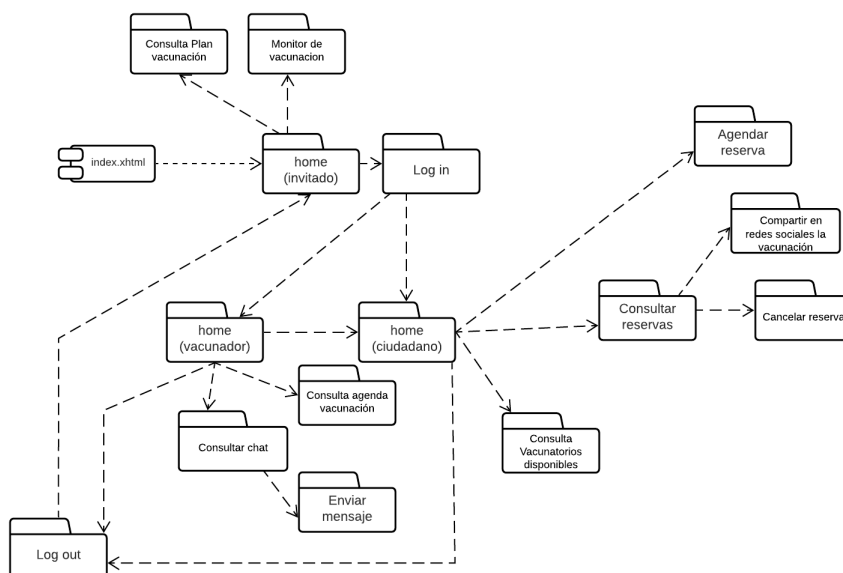
## 8. Vista de Implementación

Se presentan a continuación los componentes de deployment construidos para el sistema.

### 8.1. Arquitectura de Implementacion



### 8.2. Front-End



## 9. Vista de Decisiones de Arquitectura

La Vista de Decisiones de Arquitectura presenta y describe las principales decisiones de arquitectura tomadas [3][4].

Tabla 2: Decisión de Arquitectura

<b>Identificador</b>	1
<b>Nombre</b>	Separación por capas de las arquitecturas de los subsistemas
<b>Categorías</b>	Subsistemas
<b>Problema</b>	La aplicación debe poder ser fácilmente mantenible y escalable de manera horizontal
<b>Alternativas</b>	Otros patrones de diseño, como por ejemplo microservicios
<b>Decisión</b>	Separar por capas los subsistemas
<b>Justificación</b>	Permite distribuir en diferentes configuraciones y mantiene la implementación de cada capa separada de las otras, permitiendo sustituir una capa, por otra tecnología, siempre y cuando provea con los mismos servicios a la capa superior y consuma servicios sólo de capas inferiores.
<b>Decisiones Relacionadas</b>	Se contempla la posibilidad de distribuir la aplicación separando físicamente la base de datos, el componente central, frontoffice y backoffice de modo que facilite la escalabilidad horizontal

Tabla 3: Decisión de Arquitectura

<b>Identificador</b>	2
<b>Nombre</b>	Utilización de message bus para comunicar servidor central y vacunatorios, al igual que para los socios logísticos.
<b>Categorías</b>	Subsistemas, integración
<b>Problema</b>	El nodo periférico vacunatorio no debe perder funcionalidades si la comunicación se ve interrumpida
<b>Alternativas</b>	Otros tipos de comunicación, como por ejemplo que el sistema central comunique periódicamente el estado, intentando comunicarse con los nodos periféricos.
<b>Decisión</b>	Utilizar un message bus.
<b>Justificación</b>	Permite abstraer el concepto de la comunicación y permite a ambos componentes comunicarse asincrónicamente.
<b>Decisiones Relacionadas</b>	

Tabla 4: Decisión de Arquitectura

<b>Identificador</b>	3
<b>Nombre</b>	Acceso a las funcionalidades de 'ciudadano' siendo un vacunador.
<b>Categorías</b>	Funcionalidades, identificación
<b>Problema</b>	Un usuario identificado como 'vacunador' tiene funcionalidades extra, pero debe poder acceder también a todas las funcionalidades que tiene un 'ciudadano'.
<b>Alternativas</b>	Para una misma persona que estuviese identificada como 'vacunador' en el sistema, se podría tener dos cuentas por separado, una en la cual se le identifique como 'ciudadano' y tenga únicamente los privilegios asociados al mismo y otra cuenta en la que se le identifique como 'vacunador' en donde tenga exclusivamente las funcionalidades de 'vacunador'.
<b>Decisión</b>	Permitir a los usuarios identificados como 'vacunador' acceder a todas las funcionalidades de 'ciudadano', además de las de las funcionalidades extra que tiene 'vacunador', en una única cuenta.
<b>Justificación</b>	Más simple para el usuario, ya que no debe cambiar de cuenta para acceder a funcionalidades de distintos roles. A su vez, no será necesario dividir la interfaz en dos partes distintas, cada unas con las funcionalidades de cada rol.
<b>Decisiones Relacionadas</b>	

Tabla 5: Decisión de Arquitectura

<b>Identificador</b>	5
<b>Nombre</b>	Uso de tópicos con donde se enviarán mensajes encriptados para avisarle a los socios logísticos sobre lotes a entregar
<b>Categorías</b>	Comunicación,
<b>Problema</b>	La comunicación desde el sistema central hacia los socios logísticos debería funcionar independientemente de si los socios logísticos se encuentran disponibles o no
<b>Alternativas</b>	Utilizar otro mecanismo de mensajería, quizás una cola para cada socio logístico en el sistema.
<b>Decisión</b>	Usar un tópico .
<b>Justificación</b>	Se realizó de esta forma de modo para así no tener que configurar una nueva cola cada vez que se quiera agregar un socio logístico.
<b>Decisiones Relacionadas</b>	

Tabla 6: Decisión de Arquitectura

<b>Identificador</b>	4
<b>Nombre</b>	Uso de subscripciones permanentes al topico de mensajes
<b>Categorías</b>	Comunicación, Mensajería
<b>Problema</b>	Los socios logísticos pueden no estar disponibles cuando el sistema central manda un mensaje al topico.
<b>Alternativas</b>	Utilizar otro mecanismo de mensajería, quizás no utilizar pub/sub.
<b>Decisión</b>	Usar subscripciones permanentes.
<b>Justificación</b>	Manera directa de representar lo que se quiere en el esquema pub/sub. Aunque no estén activos los consumidores, debe esperar que lo consuman para quitarlo del topico.
<b>Decisiones Relacionadas</b>	

Tabla 7: Decisión de Arquitectura

<b>Identificador</b>	6
<b>Nombre</b>	Independencia del vacunatorio con el sistema central.
<b>Categorías</b>	Comunicación, Mensajería
<b>Problema</b>	El vacunatorio debe mantenerse funcional cuando el sistema central este deshabilitado.
<b>Alternativas</b>	Base de datos compartida.
<b>Decisión</b>	Mandar periódicamente desde el sistema central hacia los vacunatorios una copia de su base de datos que se almacena en los vacunatorios, de modo de que si el sistema central se cae, el vacunatorio tendrá datos recientes para poder seguir su funcionamiento.
<b>Justificación</b>	Como el vacunatorio tiene su propia base de datos, se mantiene más independencia con el sistema central, además, no resulta de suma necesidad tener los datos actualizados en cada segundo. Tener una única base de datos también podría ser inconveniente, ya que sería mucho mas exigida y se convierte en un punto de falla muy grande (si falla la base, tanto el sistema central como los vacunatorios se verán afectados)
<b>Decisiones Relacionadas</b>	



## Referencias

- [1] Philippe B Kruchten. «The 4+ 1 view model of architecture». En: *IEEE software* 12.6 (1995), págs. 42-50.
- [2] D Perovich y A Vignaga. *SAD del Subsistema de Reservas del Sistema de Gestión Hotelera*. Inf. téc. Thechnical Report RT03-15, InCo Pedeciba, Montevideo, Uruguay, 2003.
- [3] Uwe Van Heesch, Paris Avgeriou y Rich Hilliard. «A documentation framework for architecture decisions». En: *Journal of Systems and Software* 85.4 (2012), págs. 795-820.
- [4] Philippe Kruchten, Rafael Capilla y Juan Carlos Dueñas. «The decision view's role in software architecture practice». En: *IEEE software* 26.2 (2009), págs. 36-42.