

Documento de Arquitectura de Software

Grupo 16 - 2021

Taller de Sistemas Empresariales
Taller de Sistemas de Información Java EE

Federico Bentancor

Facundo Laborde

Michael Rodríguez

Gonzalo Santa María

Federico Sierra

Abril 2021

Tabla de contenido

1.	Introducción.....	4
1.1.	Objetivo del Documento	4
1.2.	Representación de la Arquitectura.....	4
1.3.	Partes Interesadas (i.e. <i>stakeholders</i>).....	5
1.4.	Organización del Documento.....	7
2.	Vista Conceptual	8
2.1.	Descripción General de la Plataforma	8
2.2.	Modelo Conceptual	10
3.	Vista de Casos de Uso.....	11
3.1.	Actores	11
3.2.	Diagrama de Casos de Uso	12
3.2.1.	Diagrama de casos de uso por actor	12
3.2.1.1.	Diagrama de casos de uso del rol “Ciudadano”	12
3.2.1.2.	Diagrama de casos de uso del rol “Vacunador”	13
3.2.1.3.	Diagrama de casos de uso del rol “Autoridad”	14
3.2.1.4.	Diagrama de casos de uso del rol “Administrador”	18
3.3.	Login de Usuario	19
3.4.	Agendar ciudadano	20
3.5.	Envío y recepción de agenda.....	21
4.	Vista de Restricciones.....	22
4.1.	Normativa y seguridad	22
4.2.	Comunicación.....	22
4.3.	Escalabilidad y Performance.....	22
5.	Vista de Atributos de Calidad	23
5.1.	Performance	23
5.2.	Seguridad.....	23
5.3.	Usabilidad.....	23
6.	Vista Lógica.....	24
6.1.	Arquitectura General del Sistema	24
6.2.	Refinamiento 1	25
6.3.	Refinamiento 2	26
6.4.	Refinamiento 3	27
6.5.	Refinamiento 4	28
7.	Vista de Distribución.....	29

7.1.	Arquitectura técnica	29
7.2.	Tecnología Requerida	30
7.3.	Despliegue	30
8.	Vista de Implementación.....	31
8.1.	Estructura de la aplicación	31
8.2.	Arquitectura de la implementación.....	32
9.	Vista de Decisiones de Arquitectura	33
9.1	Disponibilidad y escalabilidad	33
9.2	Interoperabilidad.....	33
9.3	Usabilidad	34
	Referencias	35

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:** Describe las funcionalidades proporcionadas por el sistema centrándose desde el punto de vista del actor.
2. **Vista de Restricciones:** Detalla las restricciones que el producto final debe respetar. Estas restricciones pueden ser tecnológicas y normativas, entre otras.
3. **Vista de Atributos de Calidad:** Enuncia aspectos de calidad y requerimientos no-funcionales del sistema.
4. **Vista Lógica:** Expone los requerimientos funcionales. Presenta las abstracciones lógicas fundamentales, sus responsabilidades y dependencias, utilizando patrones o estilos arquitectónicos.
5. **Vista de Distribución:** Se focaliza en describir la organización del sistema donde va a ser instalado, detallando los escenarios de distribución de los diferentes componentes.
6. **Vista de Implementación:** Detalla los módulos en el que está organizado el ambiente de producción, sumándole restricciones especiales de las plataformas y/o herramientas que se estén utilizando.
7. **Vista de Decisiones de Arquitectura:** Formula las decisiones llevadas a cabo respecto a la arquitectura del sistema.

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

Las partes interesadas en la plataforma vacunas.uy son:

1. **MSP:** Ministerio de Salud Pública de Uruguay. Institución integrante del gabinete del poder ejecutivo encargado de establecer las políticas y las estrategias en materia de salud pública.
2. **AGESIC:** Agencia de Gobierno Electrónico y Sociedad de la Información y del Conocimiento de Uruguay. Su finalidad es liderar la estrategia de Gobierno Electrónico y su implementación en el país.
3. **Laboratorios:** Son los proveedores de vacunas con los que la plataforma Vacunas.uy trabajará.
4. **Ciudadanos:** Usuario de la plataforma Vacunas.uy que puede: obtener información de la ejecución del plan de vacunación; gestionar su agenda de vacunación; obtener el Certificado de Vacunación; utilizar el “monitor de vacunación”; listar agendas abiertas actualmente y las próximas a abrirse; visualizar vacunatorios más cercanos de acuerdo a su ubicación; y compartir en redes sociales una vez se haya vacunado.
5. **Autoridad:** Usuario de la plataforma Vacunas.uy que puede: gestionar las enfermedades; gestionar las vacunas; gestionar los proveedores de vacunas; coordinar el envío de dosis a los vacunatorios; realizar asignación de vacunadores a dichos vacunatorios; definir planes de vacunación; definir agendas disponibles para dichos planes; crear restricciones para el público objetivo de dicho plan de vacunación; y emitir distintos reportes acerca de la vacunación, stock de dosis, etc.
6. **Vacunador:** Usuario de la plataforma Vacunas.uy que puede: obtener información sobre su agenda de vacunación; utilizar sala de chat para comunicarse con otros vacunadores; e ingresar actos vacunales.
7. **Administrador:** Usuario de la plataforma Vacunas.uy que puede: gestionar usuarios; gestionar roles; y gestionar nodos periféricos.
8. **OMS:** Organización Mundial de la Salud. Organismo de Organización de las Naciones Unidas especializado en gestionar políticas de prevención, promoción e intervención a nivel mundial en la salud.
9. **Socios logísticos:** Son los encargados del transporte de las vacunas a los diferentes vacunatorios. Deben enviar información a vacunas.uy sobre el estado del envío.
10. **Vacunatorios:** Entidades de la salud que disponen del espacio físico para realizar los actos vacunatorios en diferentes puestos de vacunación distribuidos en el mismo.

La Tabla 1 indica a que 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 Procesos	Vista de Implementación	Vista de Servicios	Vista de Distribución	Vista de Decisiones de Arquitectura
MSP		✓	✓						
AGESIC		✓		✓	✓	✓	✓	✓	✓
Laboratorios	✓		✓	✓	✓	✓	✓	✓	✓
Ciudadanos	✓		✓	✓		✓	✓		
Autoridad	✓		✓	✓		✓	✓		
Vacunador	✓		✓	✓		✓	✓		
Administrador	✓		✓	✓		✓	✓		
OMS		✓	✓						
Socios logísticos	✓		✓	✓		✓	✓		
Vacunatorios	✓		✓	✓		✓	✓		

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: brinda una descripción general de la plataforma y presenta los principales conceptos asociados a la misma,
- Sección 3: 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.
- Sección 4: Vista de Restricciones: Detalla las restricciones que el producto final debe respetar. Estas restricciones pueden ser tecnológicas y normativas, entre otras.
- Sección 5: Vista de Atributos de Calidad: Enuncia aspectos de calidad y requerimientos no-funcionales del sistema.
- Sección 6: Vista Lógica: Expone los requerimientos funcionales. Presenta las abstracciones lógicas fundamentales, sus responsabilidades y dependencias, utilizando patrones o estilos arquitectónicos.
- Sección 7: Vista de Distribución: Se focaliza en describir la organización del sistema donde va a ser instalado, detallando los escenarios de distribución de los diferentes componentes.
- Sección 8: Vista de Implementación: Detalla los módulos en el que está organizado el ambiente de producción, sumándole restricciones especiales de las plataformas y/o herramientas que se estén utilizando.
- Sección 9: Vista de Decisiones de Arquitectura: Formula las decisiones llevadas a cabo respecto a la arquitectura del sistema.

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 esta sección se presentará una descripción de la plataforma vacunas.uy, detallando sus principales funcionalidades, la estructura sobre la cual fue diseñada y los conceptos necesarios para lograr comprender el funcionamiento del sistema.

El sistema cuenta con un componente central que contendrá las capas de negocio, presentación y persistencia. La capa de negocio contendrá la lógica necesaria para llevar a cabo las funcionalidades del sistema. La capa de presentación será la encargada de generar todas las interfaces requeridas para el uso de la plataforma. Por último, la capa de persistencia será la encargada de persistir la información en las diferentes bases de datos. El componente central permitirá además la interacción entre el componente central los nodos periféricos y externos.

Los nodos externos brindarán información sobre los usuarios, obteniendo datos pertinentes de los mismos a la hora de usar el sistema y, además, se podrá obtener su historial médico, con lo cual, el sistema, estará informado de a qué planes de vacunación podrá o no acceder.

El sistema contará con dos tipos de nodos periféricos: vacunatorios y socios logísticos; los cuales brindaran información al componente central, de forma periódica o bajo demanda. El nodo vacunatorio otorgará información sobre las dosis que serán asignadas a los vacunatorios y también datos sobre los vacunadores que se asignarán a los diferentes puestos de vacunación. Estos nodos podrán realizar sus tareas de forma ininterrumpida gracias a que contarán con un componente que permitirá su independencia frente a cualquier tipo de problema existente en el componente central. En el caso de los nodos logísticos, informarán al sistema datos relacionados al transporte de las dosis. Este nodo enviará al componente central diferentes eventos críticos sobre los envíos, como podrán ser, datos de recepción y entrega de todas las dosis que transportarán hacia los centros de vacunación.

Los usuarios que utilizan el sistema están divididos en cuatro roles diferentes, siendo estos los siguientes:

- Ciudadano
- Autoridades
- Administradores
- Vacunadores

Se logra con esta división una manera simple y eficaz de repartir los permisos y posibles acciones dentro del sistema.

Cabe destacar que los usuarios con rol “Ciudadano” también podrán hacer uso, si así lo desean, de una funcionalidad que les permitirá compartir ciertas acciones en sus redes sociales. Estos, también, accederán a las funcionalidades por medio de un frontoffice web al componente central o también a través de una aplicación al componente móvil. Si el usuario elige ingresar a través del frontoffice, el mismo podrá obtener información sobre la ejecución de los planes de

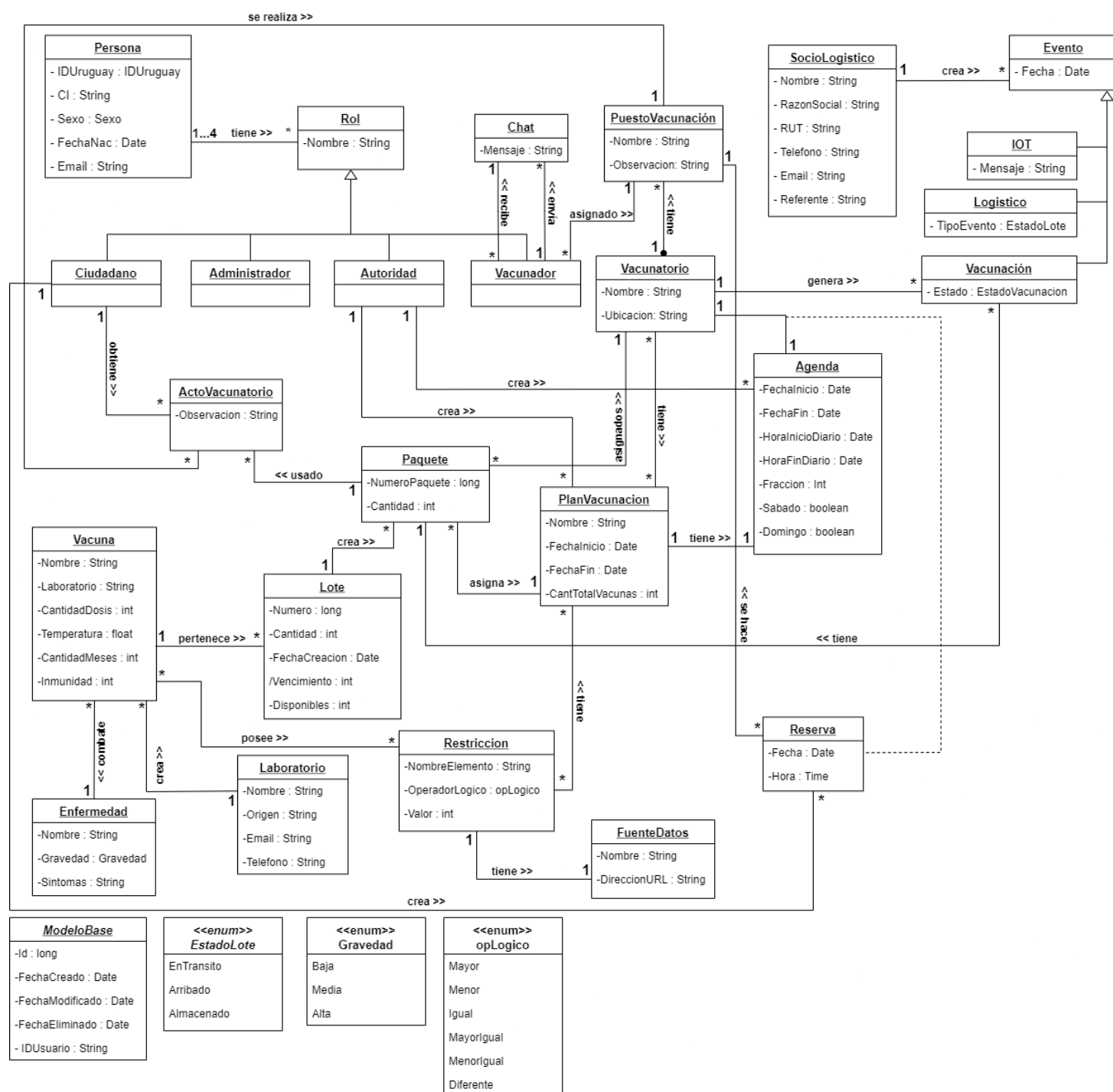
vacunación, hacer reservas o cancelar una reserva previamente realizada en alguno de estos planes; tendrá la opción de consultar los centros vacunatorios disponibles y también gestionará su agenda de vacunación. Si en cambio el usuario accede desde la aplicación móvil, se le brindarán notificaciones y un certificado de vacunación, en el cual se incluye toda la información pertinente sobre sus actos vacunatorios, dosis y períodos de inmunidad.

Los usuarios con rol “Vacunador”, accederán a las funcionalidades que serán brindadas a través del frontoffice web. Obtendrán, una vez ingresado al sitio, datos sobre su agenda de vacunación, en la cual podrán verificar el centro y el puesto al cual han sido asignados para cumplir los actos vacunatorios.

Los usuarios “Autoridades” estarán conectadas al componente central mediante un backoffice web, pudiendo así realizar todas las tareas del rol como, por ejemplo, coordinar envíos de dosis a los vacunatorios y asignación de vacunadores. Otra tarea importante de este rol será crear planes de vacunación y sus tareas relacionadas, como son: agenda del plan, establecer las etapas que formarán parte de este y sus restricciones; obtener reportes con información sobre el stock de las dosis en los diferentes vacunatorios, con lo que, de ser necesario, podrá hacer la toma de decisiones de una forma rápida y eficaz.

Por último, el rol “Administrador”, el cual realizará tareas de gestión de usuarios, roles y nodos periféricos. Esto se hace a través de un backoffice web.

2.2. Modelo Conceptual



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.

- Ciudadano: Este usuario podrá utilizar la plataforma vacunas.uy tanto en su entorno web como móvil.

Algunas de las principales funcionalidades que podrá realizar son:

- Autenticarse mediante usuario gub.uy
 - Acceder al monitor de vacunación
 - Registrarse en el sistema
 - Gestionar agenda de vacunación
 - Compartir en redes sociales
 - Obtener certificado de vacunación
 - Recibir recordatorios de reservas
 - Visualizar vacunatorios más cercanos de acuerdo a la ubicación en que se encuentre
- Vacunador: Este usuario podrá utilizar la plataforma vacunas.uy en su entorno web.

Algunas de las principales funcionalidades que podrá realizar son:

- Autenticarse mediante usuario gub.uy
 - Consulta de agenda de vacunación incluyendo información de vacunatorios y puesto de vacunación
 - Acceder a una sala de chat para el intercambio y consulta con otros vacunadores
- Autoridades: Usuario que podrá utilizar el backoffice de la plataforma vacunas.uy. El rol principal de este usuario es la gestión de los recursos a utilizar por “Ciudadanos” y “Vacunadores”.

Alguna de las principales tareas que realizará son:

- Autenticación mediante mecanismo interno
- Gestionar enfermedades, vacunas y proveedores de vacunas.
- Gestión de dosis de vacunas (recepción y envío a vacunatorios)
- Gestionar los planes de vacunación y agendas
- Generación de reportes vinculados al stock de dosis de vacunas disponibles

- Administradores: Usuario administrador de la plataforma. Utilizará el backoffice de vacunas.uy.

Tendrá habilitadas las siguientes funcionalidades:

- Autenticación mediante mecanismo interno
- Gestión de usuarios y roles
- Gestión de nodos periféricos

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, indicando cuáles se consideran críticos para la arquitectura.

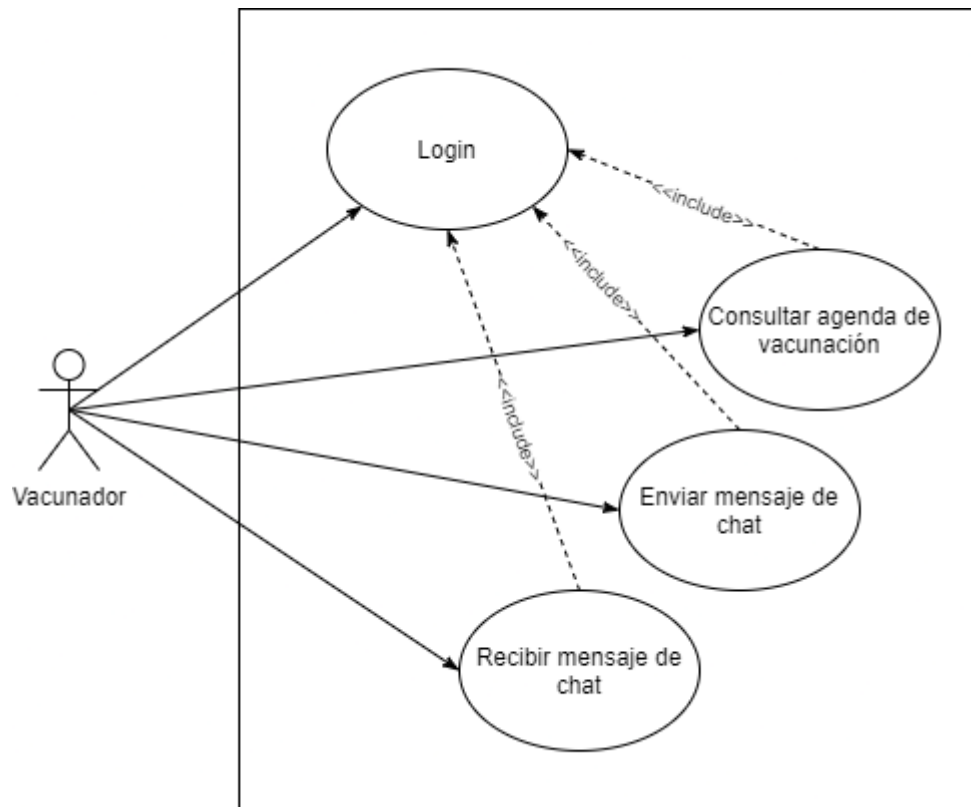
3.2.1. Diagrama de casos de uso por actor

A continuación, se detallará todos los casos de uso del sistema subdividiéndolo por actor.

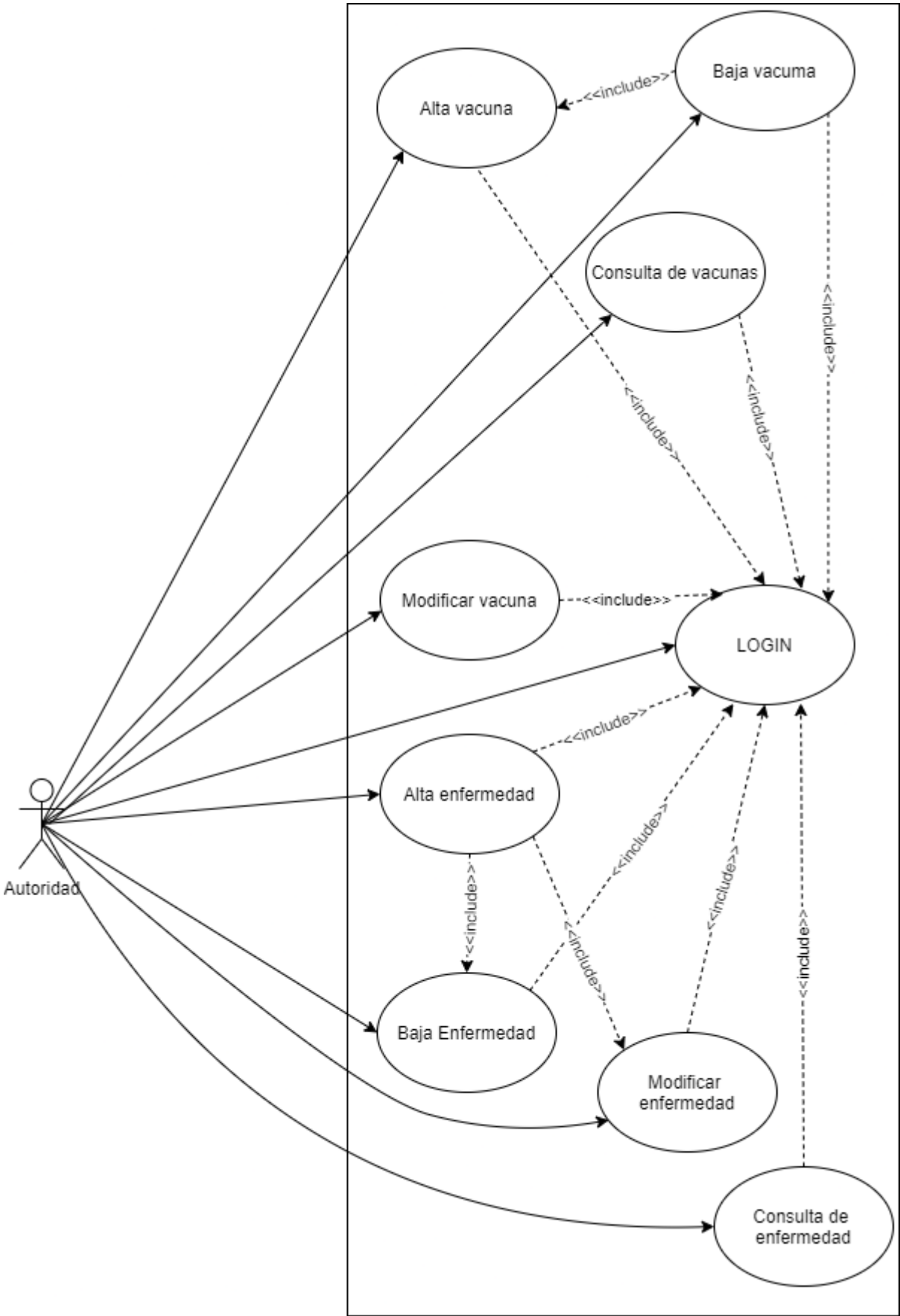
3.2.1.1. Diagrama de casos de uso del rol “Ciudadano”

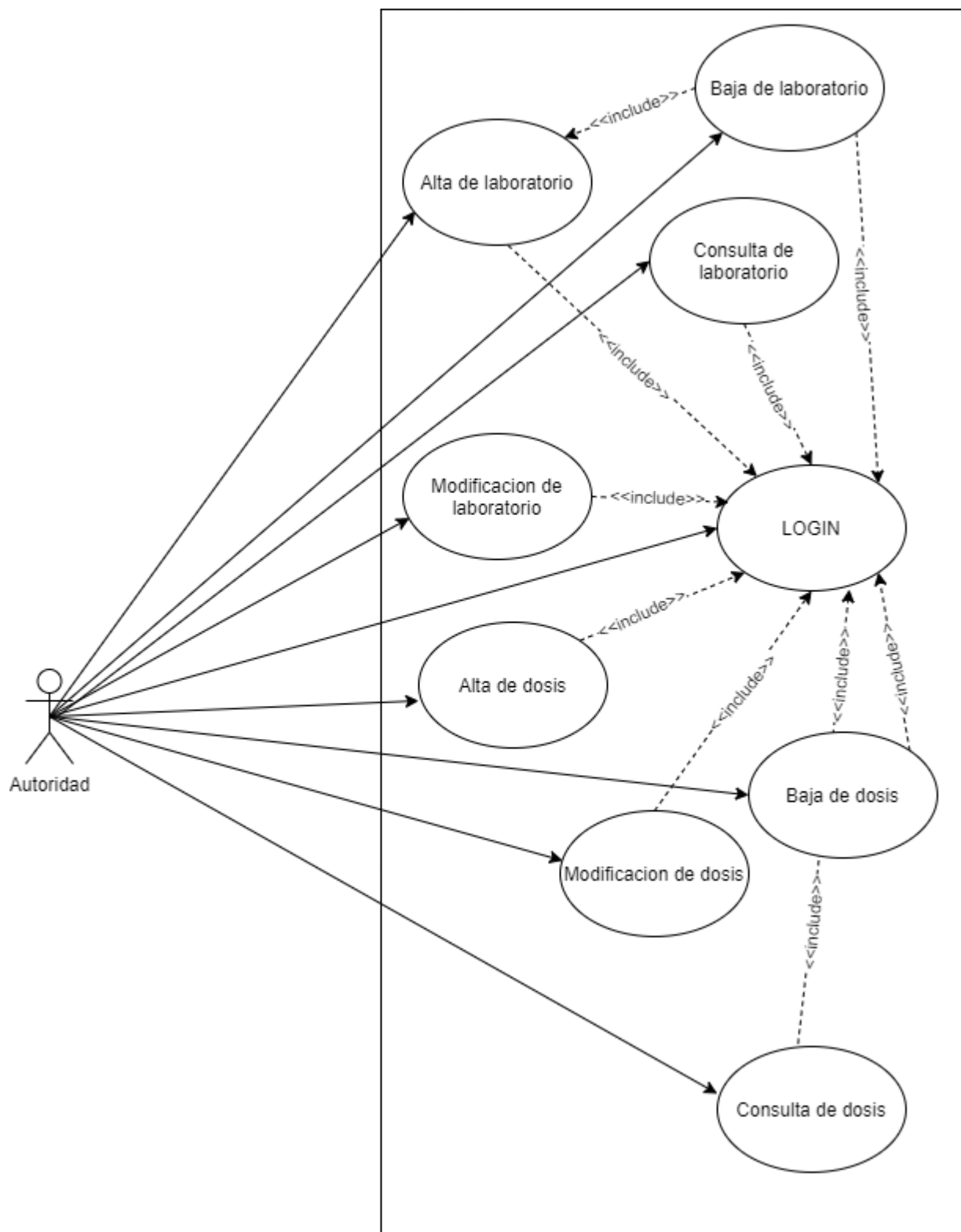


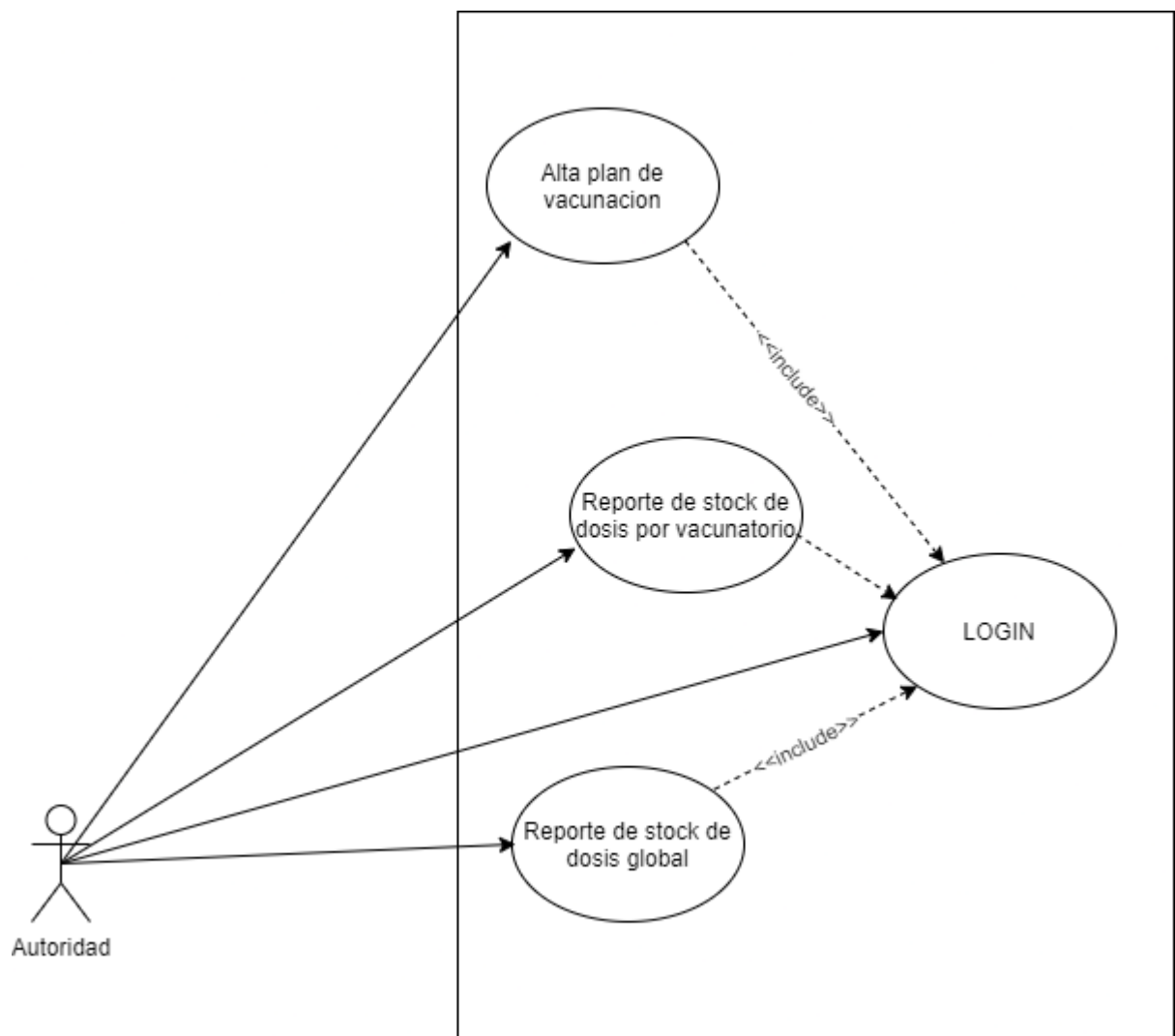
3.2.1.2. Diagrama de casos de uso del rol “Vacunador”

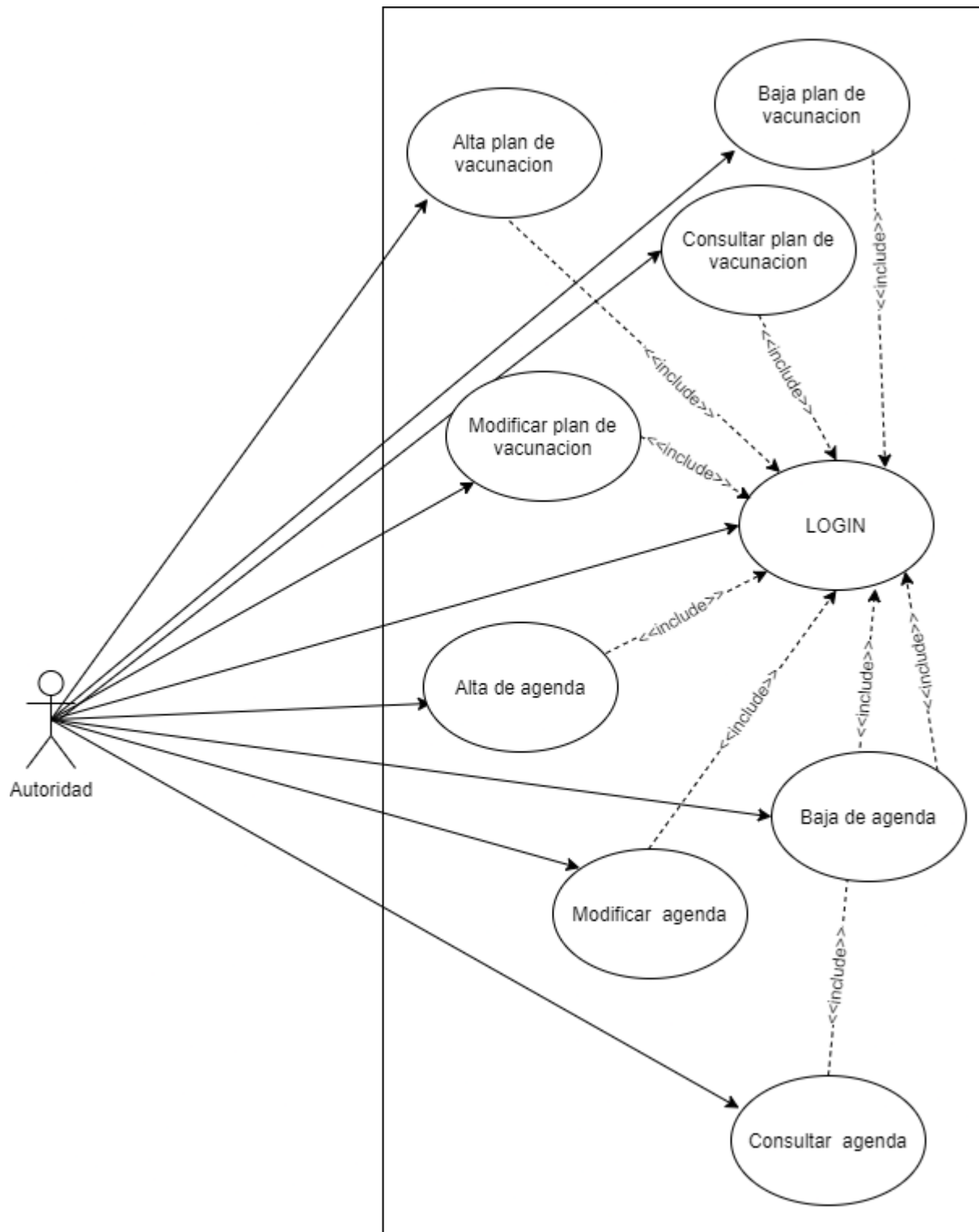


3.2.1.3. Diagrama de casos de uso del rol “Autoridad”









3.2.1.4. Diagrama de casos de uso del rol “Administrador”



3.3. Login de Usuario

Descripción

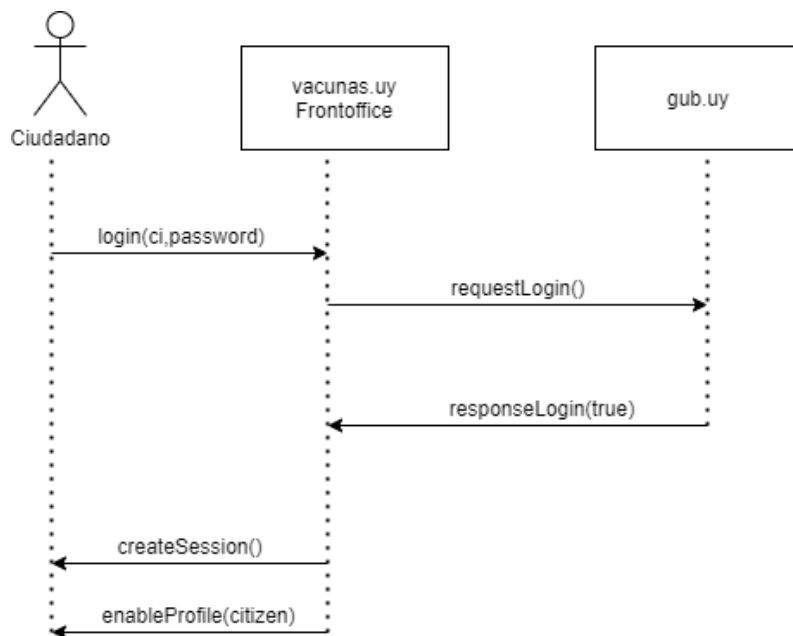
Caso de uso en el que el usuario de rol “Ciudadano” se autentica en la plataforma desde el frontoffice para acceder a las funcionalidades principales del sistema.

Pre-condiciones

El usuario debe existir en la base de datos con el rol “Ciudadano”.

Flujo de Eventos

El “Ciudadano” ingresa al portal vacunas.uy, presiona en “Ingresar a la plataforma” e introduce documento de identidad y contraseña. Desde el frontoffice se envía un *request* a gub.uy para autenticar el usuario y este envía un response validando la transacción. Finalmente se informa al usuario el ingreso a la plataforma y se crea la sesión.



Post-condiciones

Se crea sesión de usuario a nivel de frontend.

3.4. Agendar ciudadano

Descripción

Caso de uso en el que un usuario de rol “Ciudadano” crea una reserva en un vacunatorio para una fecha y hora determinada disponible.

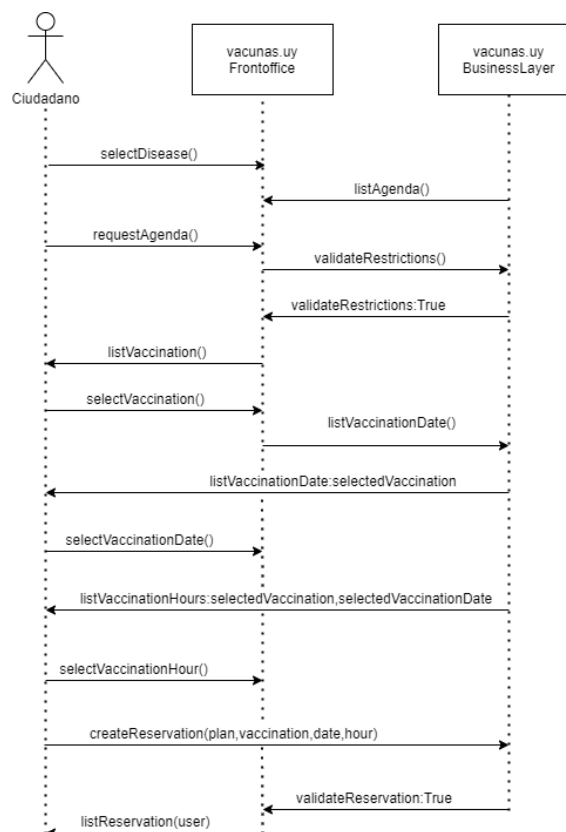
Pre-condiciones

El usuario debe existir en la base de datos con el rol “Ciudadano” y debe haber iniciado sesión previamente. Además tiene que existir una agenda, vacuna, vacunatorio y enfermedad asociada.

Flujo de Eventos

El “Ciudadano” desde el portal vacunas.uy, selecciona la opción “Agendarse”.

El usuario elige la enfermedad para la cual desea vacunarse. El sistema lista los vacunatorios disponibles teniendo en cuenta las restricciones del usuario (edad, sexo, etc.) y selecciona uno. A continuación, se listan las fechas disponibles para agenda en ese vacunatorio seleccionado. El usuario selecciona la fecha deseada y se listan los horarios disponibles. Luego de seleccionado el horario se envía a la capa de negocios la solicitud para agendar al usuario en el vacunatorio, fecha y horario seleccionado. Finalmente, se le informa en pantalla los datos de su reserva.



Post-condiciones

Se crea una entrada en la tabla “Reserva” con fecha y hora y los datos del “Ciudadano” y el puesto de vacunación asignado.

3.5. Envío y recepción de agenda

Descripción

Caso de uso en el que un usuario de rol “Autoridad” envía una agenda y es recibida en el vacunatorio determinado.

Pre-condiciones

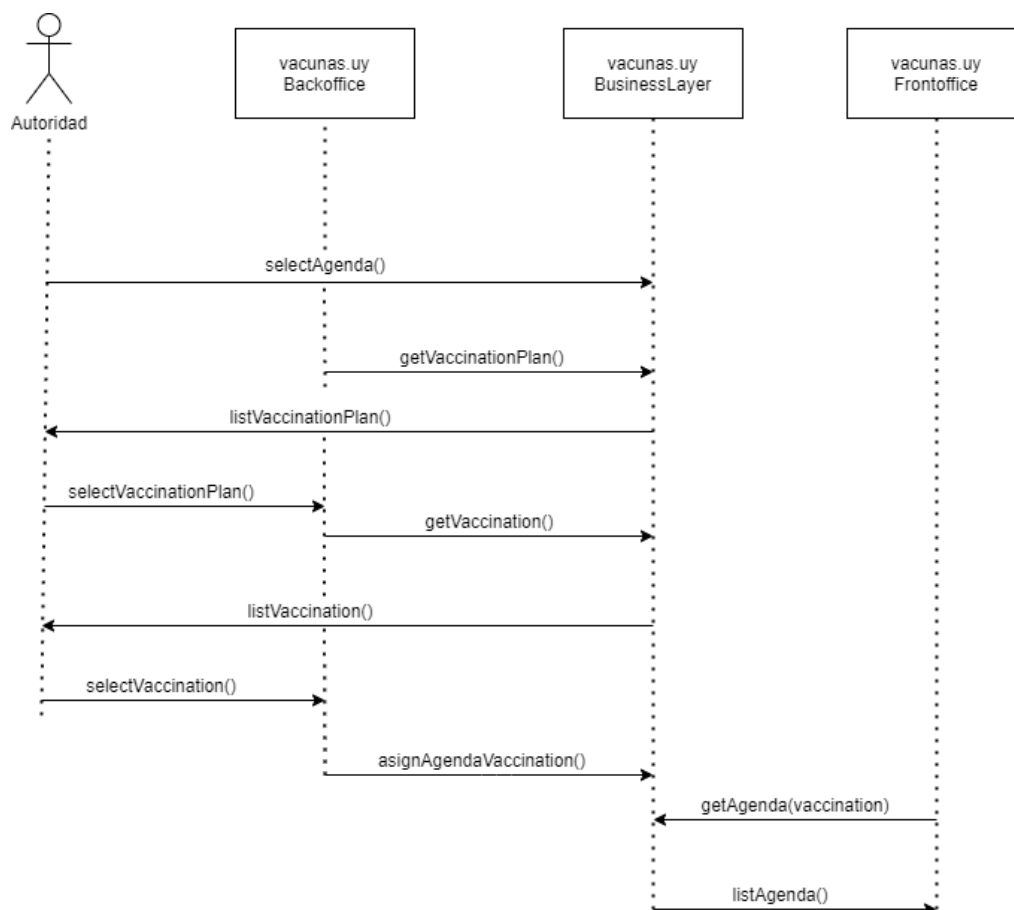
El usuario debe existir en la base de datos con el rol “Autoridad” y debe haber iniciado sesión previamente. Deben existir tanto el vacunatorio como el plan de vacunación al que se asigna la agenda.

Flujo de Eventos

La “Autoridad” desde el portal vacunas.uy, selecciona la opción “Enviar Agenda”.

El usuario selecciona la agenda y se solicita al sistema los planes de vacunación disponibles. El usuario selecciona un plan de vacunación y se listan los vacunatorios asociados al mismo. “Autoridad” selecciona el vacunatorio deseado y el sistema envía la agenda al vacunatorio y plan de vacunación previamente seleccionados.

Finalmente, el sistema recibe un pedido de las agendas y se reciben las agendas correspondientes en el vacunatorio.



Post-condiciones

Se crea una entrada en la tabla que asocia a la agenda con el plan de vacunación correspondiente.

Se listan las agendas en el frontend.

4. Vista de Restricciones

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

Esta sección está destinada a plasmar las restricciones que se aplican al proyecto. Las mismas, deben ser aplicadas tanto en el diseño, en el desarrollo y en el mantenimiento del mismo.

4.1. Normativa y seguridad

La información que manejará el sistema es delicada y de gran importancia, por eso es necesario proteger las contraseñas de usuario utilizando funciones de hash con salt en su almacenamiento. La comunicación de estos datos entre los diferentes componentes no es ajena a las restricciones de seguridad, estas se deben de realizar mediante el protocolo HTTPS.

Desde el punto de vista normativo, respetando los estatutos vigentes, el sistema respetará las normas que refieren a Historia Clínica Electrónica Nacional, Seguridad de la Información y Privacidad e Intercambio de Información.

4.2. Comunicación

El núcleo del sistema se desarrolla con la plataforma Java Enterprise Edition, la misma se comunica con el resto de los componentes mediante servicios web. En el caso de la aplicación Android, desarrollada en Java, se comunica con el núcleo mediante servicios REST, en cambio para los nodos externos y periféricos se utilizarán servicios SOAP.

4.3. Escalabilidad y Performance

Dado la magnitud de solicitudes que se espera tener en el sistema en momentos pico, es necesario que el mismo cuente con pruebas de estrés para asegurar el funcionamiento en el “peor escenario”.

Es importante destacar que toda la información que el sistema maneje será persistida en la base de datos correspondiente, ningún tipo de información será almacenada en los servidores web o de aplicación. Esto permitirá que el sistema esté preparado para ser escalable horizontalmente.

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. Performance

Este es un punto por demás importante ya que, debido a la criticidad de las funcionalidades que provee el sistema, se deberá contar con aplicaciones performantes.

Por ello, se realizarán, con especial cuidado, los controles necesarios para que las operaciones críticas se lleven a cabo en el menor tiempo posible. Para la medición y monitoreo de los servicios se utilizarán las herramientas de “Elastic Cloud” de minubeantel.uy. En base a *thresholds* definidos con las partes interesadas, lanzaremos alertas de performance y/o escalado horizontal automático.

Por otra parte, debido a la necesidad de almacenar gran cantidad de datos de geolocalización y eventos en tiempo real, se decidió optar por una base de datos no relacional (MongoDB), ya que una de las ventajas de estas tecnologías es su capacidad para escribir gran cantidad de datos en poco tiempo.

5.2. Seguridad

A nivel de arquitectura de software se hará hincapié en mantener reglas claras de seguridad para todos los usuarios registrados. Para ingresar al sistema un usuario deberá ser registrado, otorgándole las credenciales pertinentes; los datos personales de un usuario sólo podrán ser accedidos por él mismo y por los usuarios con rol “Administrador” que gestionarán dichos datos.

A nivel de despliegue, el sistema trabajará sobre el protocolo HTTPS, el cual brindará un cifrado a los datos intercambiados con el componente central y los diferentes nodos que el sistema establece comunicación.

5.3. Usabilidad

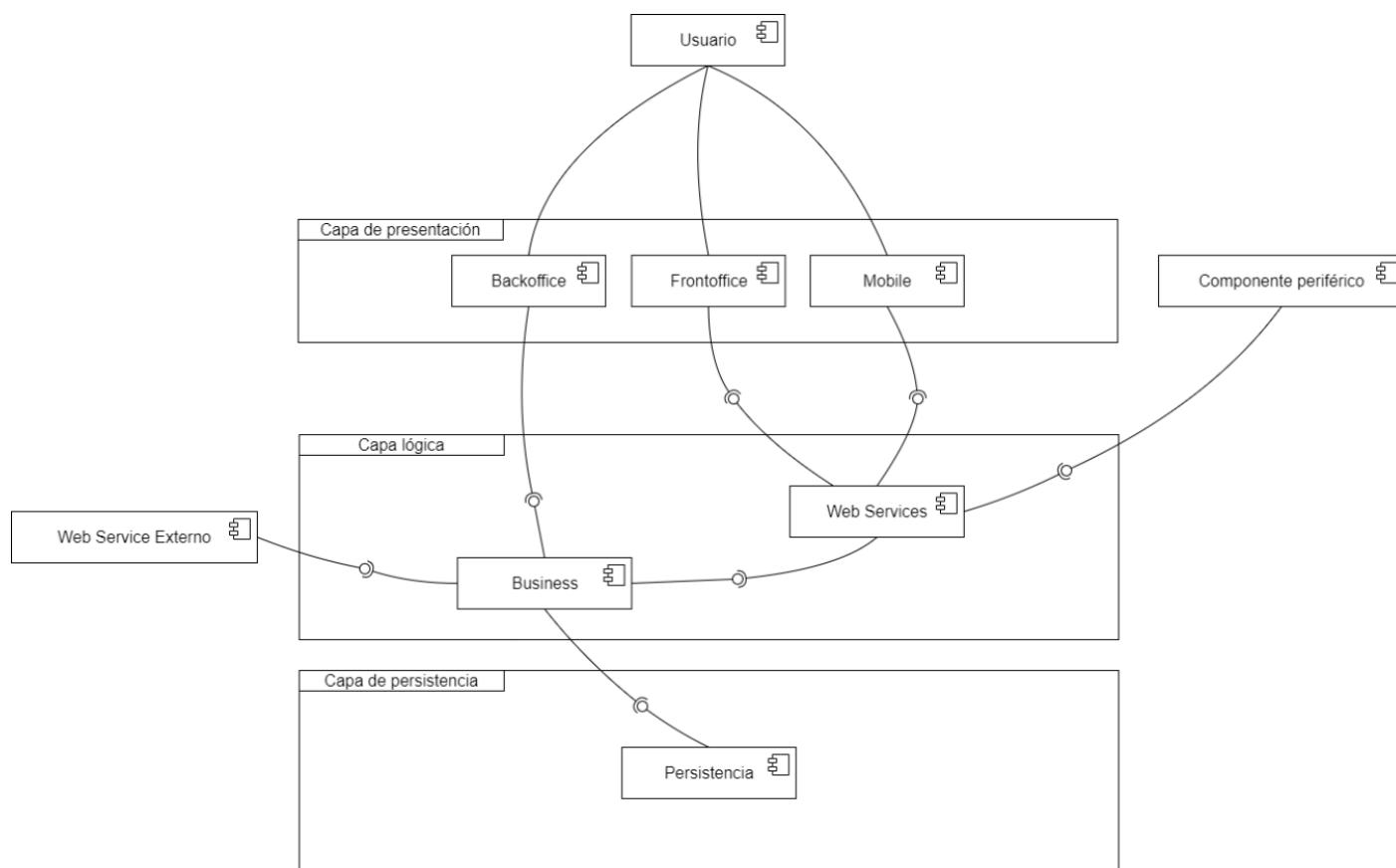
El sistema estará desarrollado para que su uso pueda hacerse por cualquier persona debido a que la plataforma web cuenta con un desarrollo simple e intuitivo para el usuario permitiendo así que quién use vacunas.uy podrá ser un usuario novato en el uso de aplicaciones web.

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 el siguiente diagrama se presenta la arquitectura lógica de la plataforma vacunas.uy.

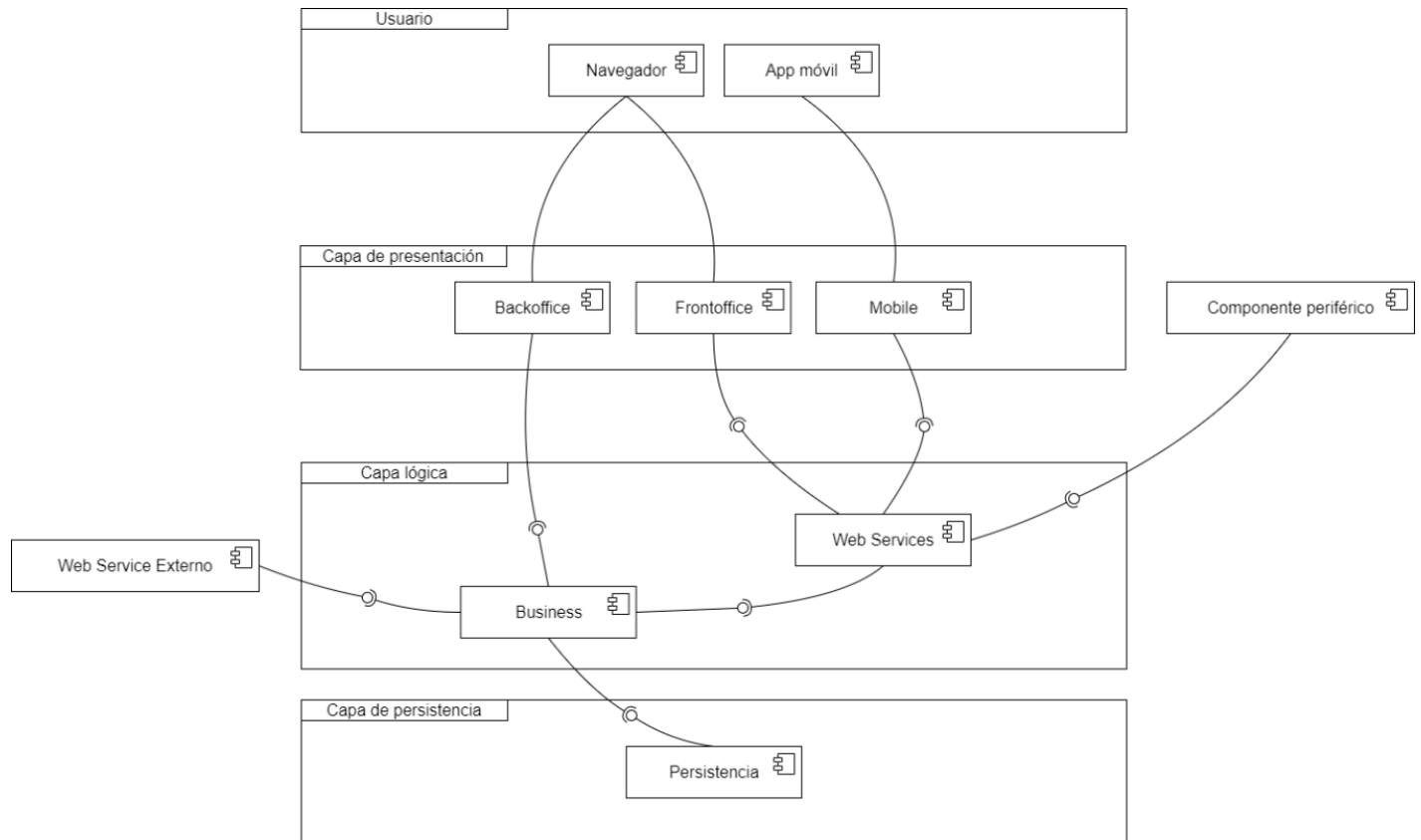


6.2. Refinamiento 1

En este primer refinamiento se detallan los diferentes tipos de accesos por los cuales los diferentes usuarios harán ingreso a la plataforma.

A través de la aplicación móvil (Componente “App móvil”), los usuarios con rol “Ciudadano” ingresarán al sistema desde cualquier dispositivo con sistema operativo Android accediendo a funcionalidades propias.

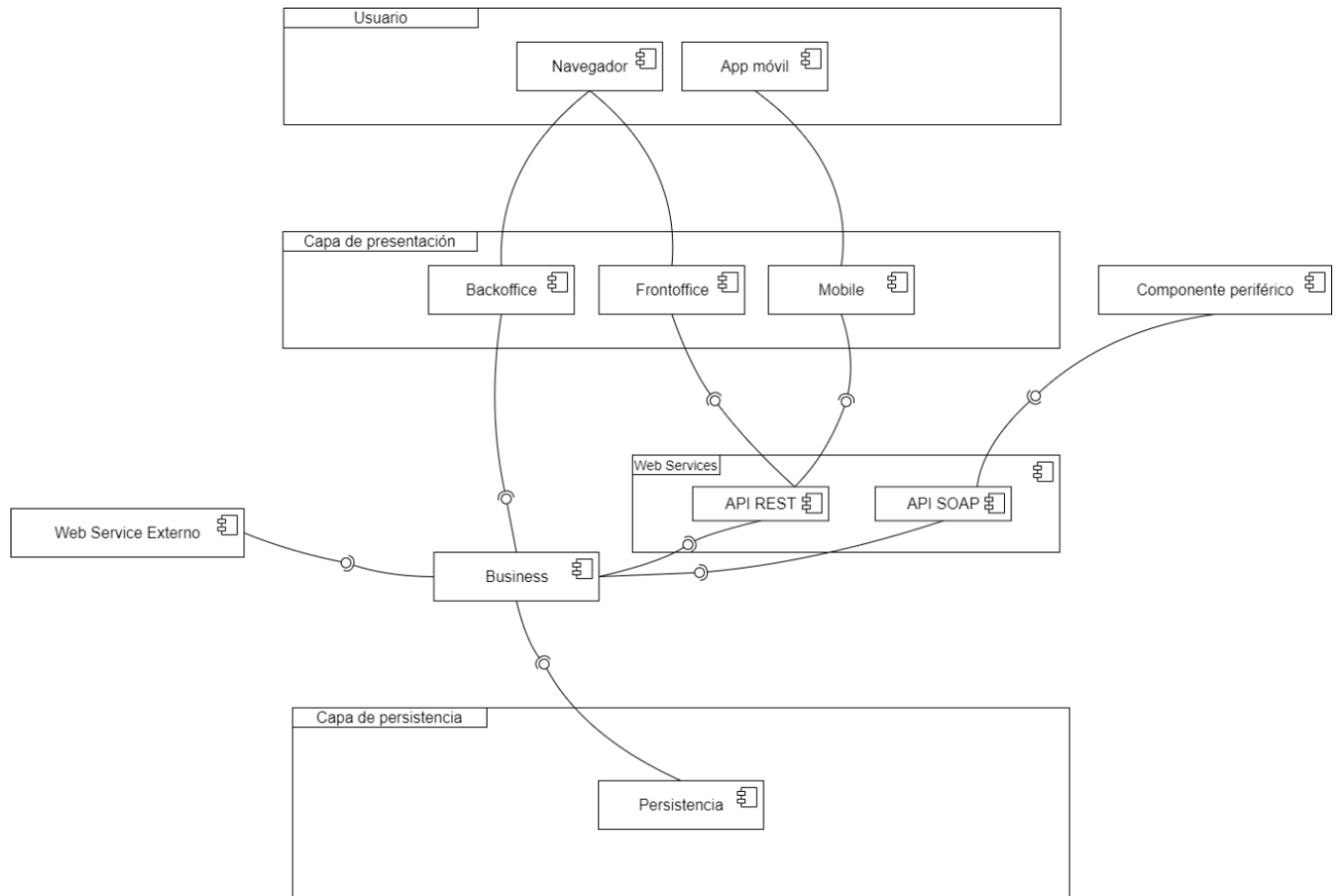
Por medio del navegador ingresarán al sistema los usuarios con cualquiera de los roles.



6.3. Refinamiento 2

En este segundo refinamiento se hizo desglose de los servicios que se expondrán para que los componentes Frontoffice, Mobile y los periféricos, actúen dentro del sistema.

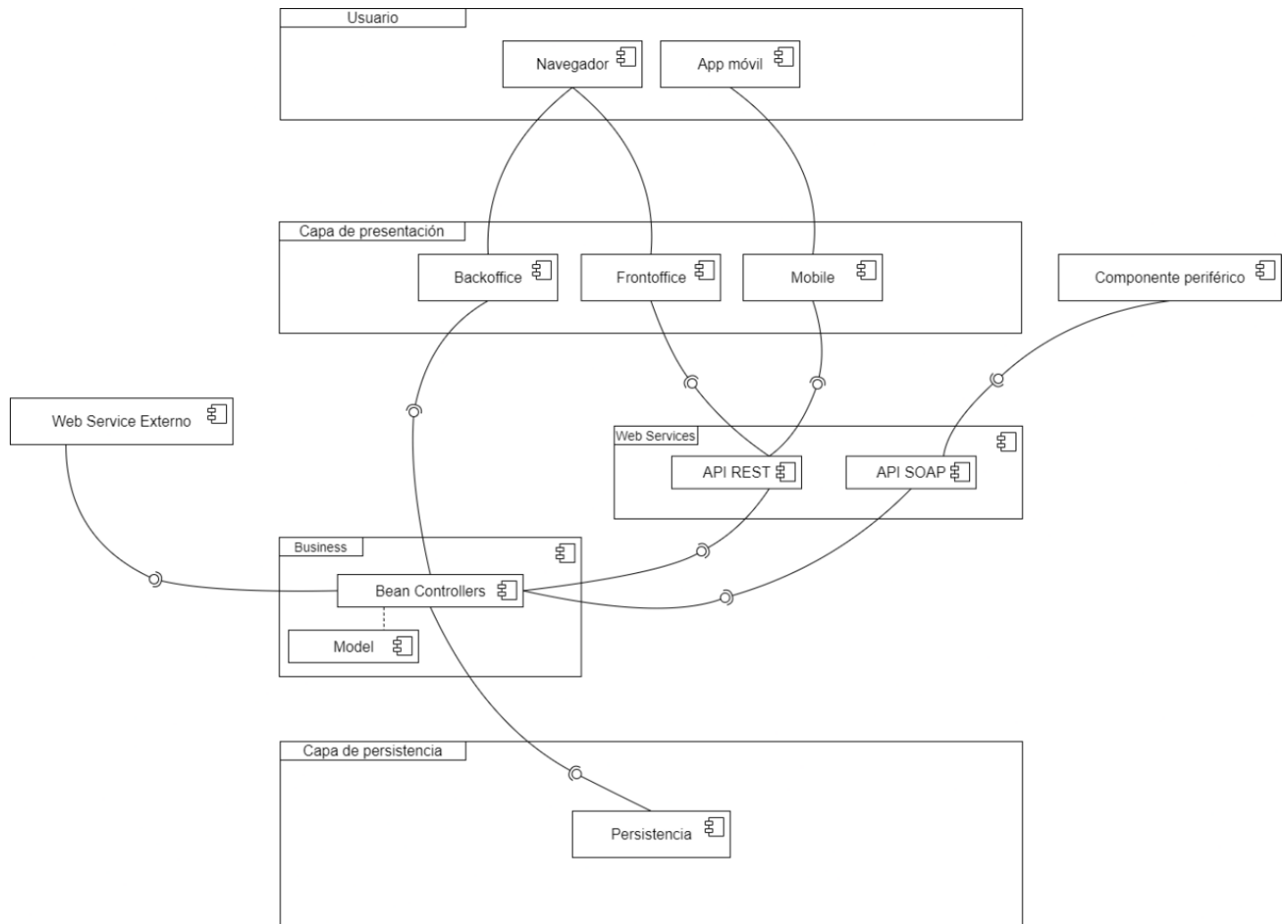
Para el caso del Frontoffice y Mobile utilizarán los servicios REST. En cambio, el componente periférico, accederá a los servicios SOAP.



6.4. Refinamiento 3

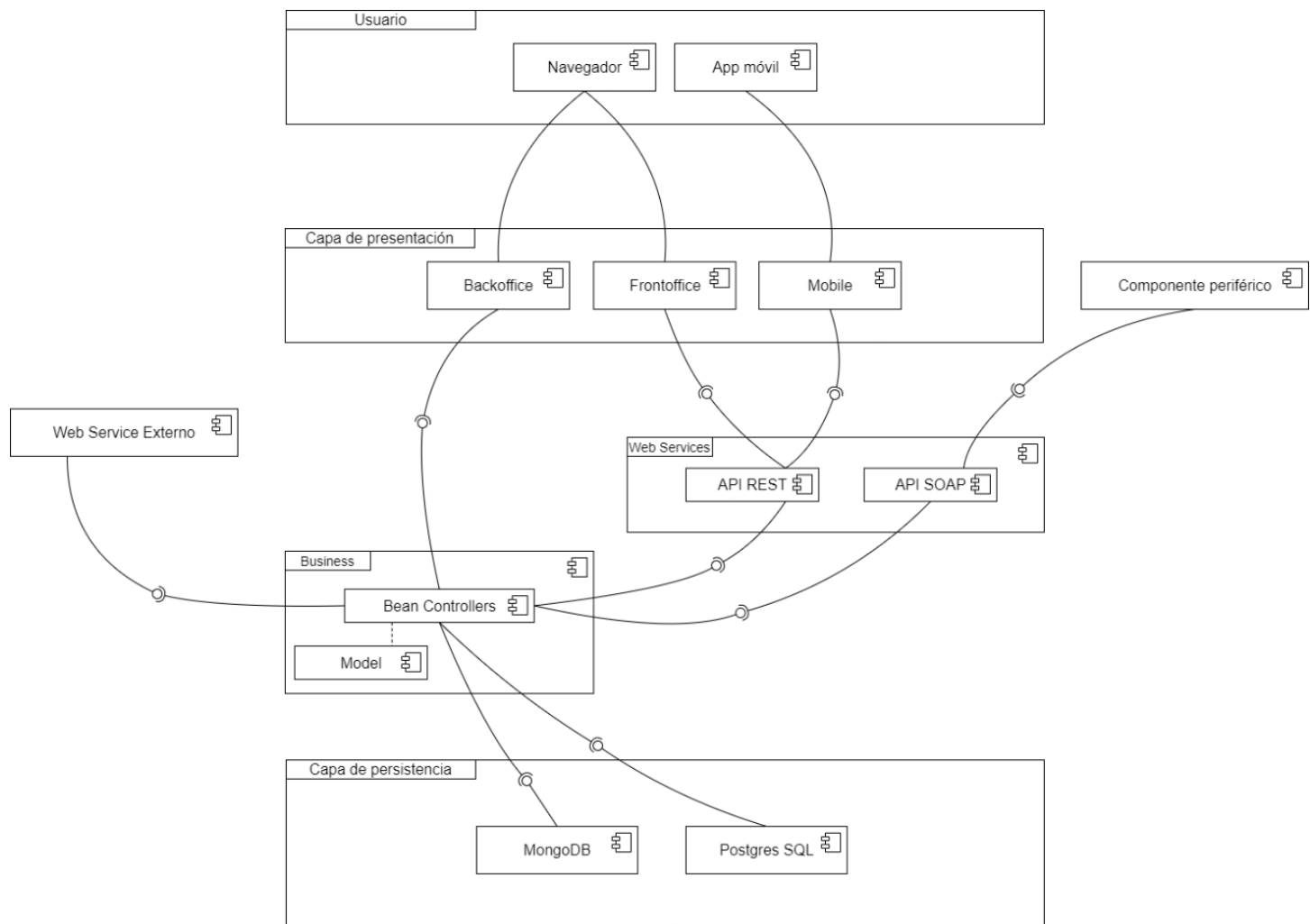
En el tercer refinamiento, se detalla el componente Business. En el mismo se encontrarán los “Bean Controllers” encargados de contener los métodos de la inteligencia del negocio, los cuales brindarán operaciones a los componentes que requieran, por medio de interfaces.

A su vez, este “Bean Controller” apoyará su lógica en el “Model” que cuenta con la especificación del modelo conceptual.



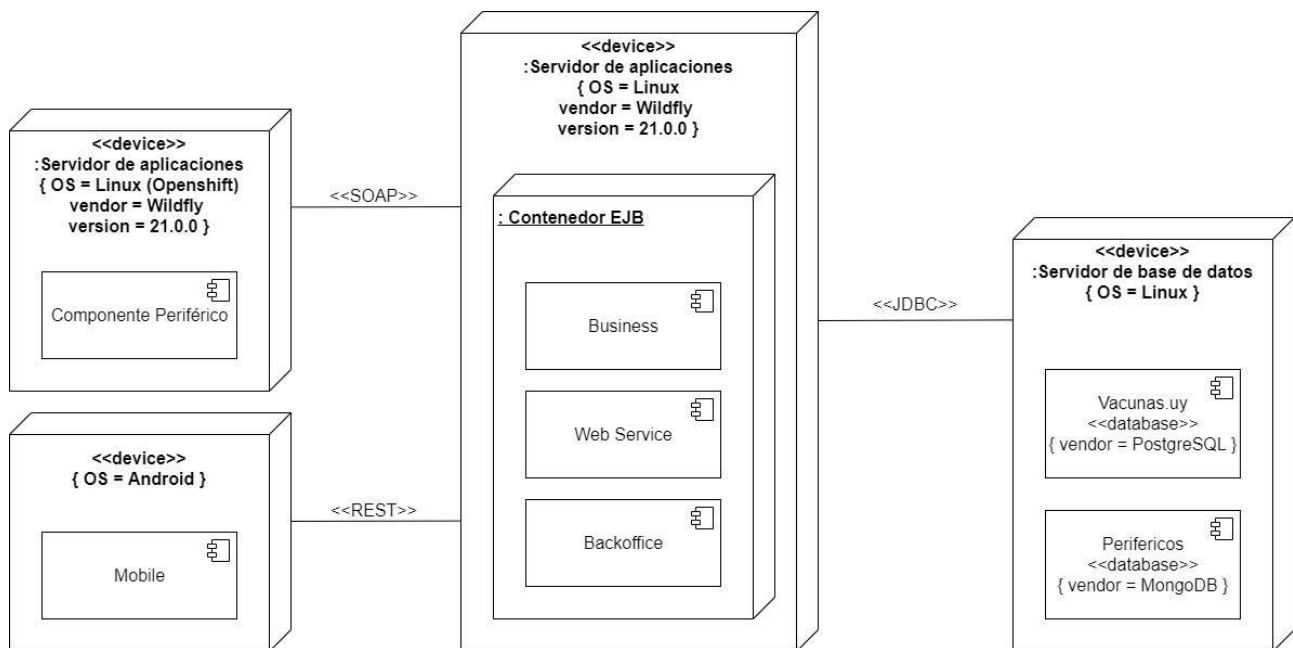
6.5. Refinamiento 4

En este último refinamiento, se pormenoriza la capa de persistencia, observándose que el “Bean Controller” hará la gestión de los datos en bases de datos MongoDB y PostgreSQL.



7. Vista de Distribución

En la siguiente vista haremos detalle de los nodos de la arquitectura técnica de la plataforma a desplegar, determinando también la relación entre ellos.



7.1. Arquitectura técnica

Del anterior diagrama se identifican cuatro nodos:

- **Servidor de aplicaciones:** dicho nodo contiene el contenedor EJB, en el cual se encuentran todas las estructuras lógicas que resolverán la inteligencia del negocio, así como también expondrá los servicios web para la interconectividad. Este dispositivo es un servidor con sistema operativo Linux, en el cual se ejecuta un servidor de aplicaciones Wildfly.
- **Servidor de base de datos:** en el mismo encontramos las dos bases de datos en las cuales la plataforma persiste la información. Este, a su vez, tiene dos motores independientes, que se encargan de guardar diferente información. Estos motores son, PostgreSQL y MongoDB, ambos se encuentran en un servidor con sistema operativo Linux.
- **Servidor de aplicaciones periférico:** en este se desplegarán los componentes periféricos que interactúan con el sistema (servicio logístico y vacunatorios). Estos están en un servidor Linux que ejecuta un servidor de aplicaciones Wildfly.
- **Mobile:** nodo que contiene al componente móvil. El mismo se ejecuta desde un sistema operativo Android.

7.2. Tecnología Requerida

Los usuarios que se conectarán al frontoffice o al backoffice, deberán hacer uso de un navegador (Google Chrome, Internet Explorer, Mozilla Firefox, etc.) con la habilitación de ejecución de JavaScript.

Los usuarios con rol “Ciudadano”, que accederán a la plataforma mediante dispositivos móviles, deberán haber instalado previamente el paquete de aplicación Android (APK) y contar con una versión igual o superior al Android 6.0 (Marshmallow).

Para la persistencia de la información utilizaremos PostgreSQL y MongoDB, en sus versiones 13.2 y 4.4 respectivamente.

Los servidores de aplicaciones utilizarán Wildfly en su versión 21.0.0 Final.

7.3. Despliegue

El servidor de aplicaciones que contiene el componente central se comunicará con el nodo de servidor de base de datos por medio de una API JDBC.

El nodo móvil se comunicará con el nodo central mediante API REST.

Los nodos periféricos establecerán comunicación con el servidor de aplicaciones central, utilizando API SOAP.

8. Vista de Implementación

En la vista de implementación presentamos el conjunto de componentes y su despliegue, que conforman el sistema desarrollado para este sistema.

Para el desarrollo de cada componente fueron utilizadas diversas tecnologías. Para el componente central se utilizará Java Enterprise Edition; el backoffice se implementará en Java Server Faces; el frontoffice se desarrollará en React JS; y, por último, para la aplicación móvil se utilizará Java Android.

8.1. Estructura de la aplicación

Cada parte de las anteriores se encuentra en un proyecto independiente, implementando estructuras adecuadas a la tecnología que se utiliza para cada una de ellas. Cada uno de estos módulos tiene un conjunto de paquetes asociados a la funcionalidad que ofrece.

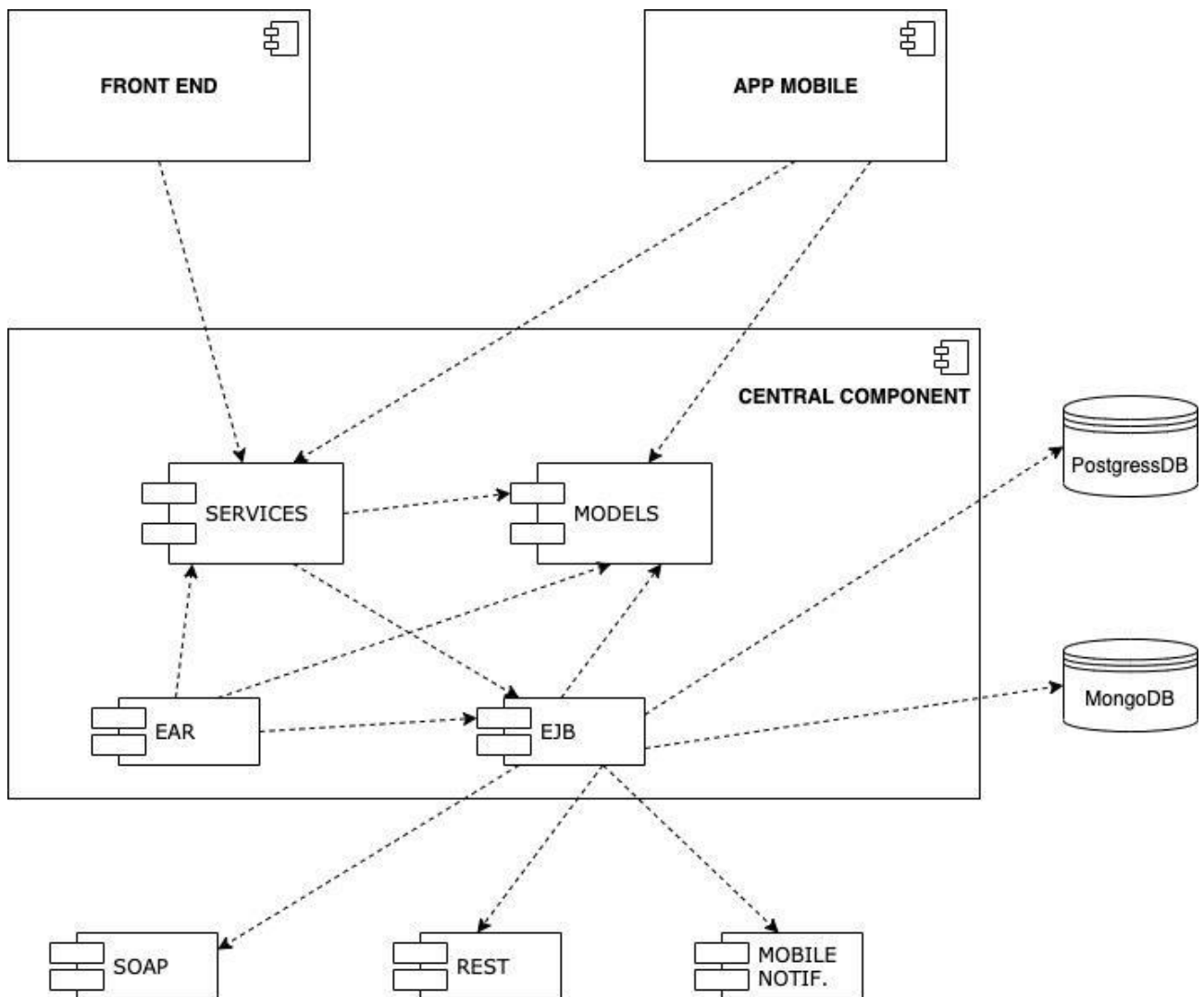
El componente central utiliza el estándar Model View Control (MVC) con los siguientes módulos:

- Servicios: Es el módulo que expone los servicios al frontoffice, a la aplicación Android, a los módulos externos y a los nodos periféricos.
- Models: Es el módulo que contendrá el modelado de métodos y clases del sistema.
- EAR: Es el módulo que almacena los archivos desplegables.
- EJB: Es el módulo que contiene toda la lógica de negocio.

Tanto el frontoffice como el backoffice, dependen exclusivamente del componente central para su funcionamiento. Dicho componente persistirá la información en una base de datos PostgreSQL y en una base de datos no relacional MongoDB. Además, se cuenta con un servicio de notificaciones para la aplicación móvil. Las comunicaciones de la capa lógica con otros componentes externos a ella se harán en un conjunto de servicios web REST y SOAP.

8.2. Arquitectura de la implementación

Se adjunta diagrama de Arquitectura de implementación que demuestra cuales son las dependencias entre los componentes y módulos de las capas que componen el sistema.



9. Vista de Decisiones de Arquitectura

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

9.1 Disponibilidad y escalabilidad

Problema:

Dado que el sistema a desarrollar es un sistema de misión crítica, es necesario que se contemplen algunos escenarios extremos como, por ejemplo:

- Fallas por saturación
- Problemas en los servidores de aplicación

Decisión:

Se despliega la aplicación en la nube de Antel que utiliza una plataforma Elastic Cloud. La misma permite un escalamiento tanto vertical como horizontal; también, cuenta con funciones de balanceo de carga dinámico y fail over.

Con este tipo de despliegue mantenemos los tiempos de respuesta en momentos “pico” y contamos con una alta tolerancia a fallos.

9.2 Interoperabilidad

Problema:

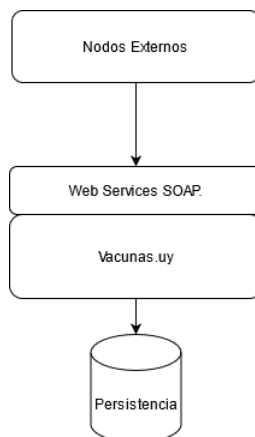
Conexión con nodos externos para actualizar información.

Solución:

Se dispondrá de un servicio web del tipo SOAP para comunicarse con los nodos externos.

Justificación:

Los servicios web SOAP permiten conexiones entre diferentes plataformas desarrolladas en diferentes lenguajes, esto es posible debido al uso de XML que permite invocar procedimientos remotos.



Problema:

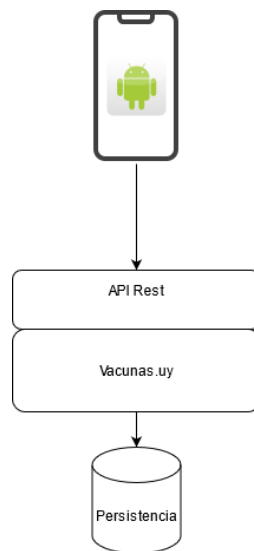
Conexión con aplicación Android para dispositivos móviles.

Solución:

Se dispondrá de un servicio web del tipo REST para obtener y enviar información.

Justificación:

Dado el alto uso de aplicaciones móviles hoy en día, es necesaria una solución que ayude a disminuir la carga de los servidores. REST permite disponibilizar recursos de forma fácil, sin mantener sesiones para no comprometer el rendimiento y mantener la interoperabilidad.



9.3 Usabilidad

Problema:

El frontoffice tiene que ser intuitivo y ágil ya que será utilizado por muchas personas de diferentes franjas etarias y conocimiento técnico.

Solución:

Se utilizará React JS para el desarrollo del frontoffice.

Justificación:

React JS permite desarrollo ágil y modular para facilitar los cambios. Se pueden desarrollar webs flexibles y de rápida respuesta dado que se basa en Javascript.

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. t'ec. Technical Report RT03-15, InCo Pedeciba, Montevideo, Uruguay, 2003.
- [3] Paulo Merson. *Data model as an architectural view*. Inf. t'ec. CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST, 2009.
- [4] Salud.uy. *Documento de Arquitectura HCEN*. Inf. t'ec. Ver. 1.1.6. 2017.
- [5] LINS. *Letra Trabajo Laboratorio 2021 - Taller de Sistemas Empresariales*. Inf. t'ec. 2021.
- [6] Humberto Cervantes y Rick Kazman. *Designing software architectures: a practical approach*. Addison-Wesley Professional, 2016.
- [7] Len Bass, Paul Clements y Rick Kazman. *Software Architecture in Practice (3rd Edition) (SEI Series in Software Engineering)*. Addison-Wesley Professional, 2012.
- [8] Ambysoft. *UML 2 Component Diagrams: An Agile Introduction*. Inf. t'ec. 2018. url: <http://www.agilemodeling.com/artifacts/componentDiagram.htm>.
- [9] Ambysoft. *UML 2 Deployment Diagrams: An Agile Introduction*. Inf. t'ec. 2018. url: <http://www.agilemodeling.com/artifacts/deploymentDiagram.htm>.
- [10] 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.
- [11] Philippe Kruchten, Rafael Capilla y Juan Carlos Duen~as. ((The decision view's role in software architecture practice)). En: *IEEE software* 26.2 (2009), págs. 36-42.