

Documento de Arquitectura de Software

Grupo 18 - 2022

Taller de Sistemas de Información Java EE

Mauricio Iglesias

Manuel Biurrun

Mathias Fernández

Julio 2022

Tabla de Contenido

1. Introducción	3
1.1. Objetivo del Documento	3
1.2. Representación de la Arquitectura	3
1.3. Partes Interesadas (i.e. <i>stakeholders</i>)	4
1.4. Organización del Documento	5
2. Vista Conceptual	6
2.1. Descripción General de la Plataforma	6
2.2. Modelo Conceptual	8
3. Vista de Casos de Uso	9
3.1. Actores	9
3.2. Diagrama de Casos de Uso	9
3.3. Iniciar Sesión - Critico	10
3.4. Proponer Iniciativa - Critico	10
3.5. Adherirse a Iniciativa - Critico	11
3.6. Reportar Proceso Participativo - Critico	11
3.7. Obtener Certificado de Participación - Critico	12
4. Vista de Restricciones	13
5. Vista de Atributos de Calidad	15
6. Vista Lógica	16
6.1. Arquitectura General del Sistema	16
6.2. Refinamiento 1	17
6.3. Refinamiento 2	17
6.4. Diagramas de Interacción	18
7. Vista de Distribución	23
7.1. Escenario 1	23
7.2. Escenario 2	23
7.3. Escenario 3	23
8. Vista de Implementación	24
9. Vista de Decisiones de Arquitectura	25

1. Introducción

Este documento presenta la arquitectura de la plataforma «eparticipation.uy», la cual fue planteada como trabajo laboratorio de la edición 2022 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 espacios de participación ciudadana digital en Uruguay.

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 «eparticipation.uy».

1.2. Representación de la Arquitectura

La arquitectura de la plataforma «eparticipation.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 «eparticipation.uy» son:

1. **Vista de Casos de Uso:** Representa las funcionalidades mas importantes del sistema a modelar desde el punto de vista de los distintos usuarios del mismo. Esta vista describe los comportamientos esperados del sistema.
2. **Vista de Restricciones:** Describe las diversas restricciones (e.g. tecnológicas, regulatorias, estándares) que deben ser respetadas por el sistema.
3. **Vista de Atributos de Calidad:** La vista de atributos de calidad lista las propiedades relacionadas a requerimientos no funcionales (e.g. escalabilidad, seguridad, mantenibilidad, modularidad) que deben ser cumplidas por el sistema.
4. **Vista Lógica:** Presenta las estructuras conceptuales (componentes/paquetes/módulos) que conforman el sistema y sus relaciones.
5. **Vista de Despliegue (Distribución/Deployment):** Presenta cómo va a funcionar el sistema en el entorno productivo. Muestra cómo están conectados los diferentes componentes de infraestructura del sistema (e.g. firewall, balanceadores de carga, servidores, base de datos). La vista de despliegue representa la disposición física del sistema.
6. **Vista de Implementación:** Representa las opciones arquitectónicas y tecnológicas. Aquí se incluyen las particularidades del estilo arquitectónico seleccionado para implementar la solución (e.g. las capas de una aplicación, basada en eventos) junto con el razonamiento que la sustenta. También se detallan las opciones tecnológicas (e.g. si se utiliza Java o Node.js) junto con los pros y los contras de su uso. En la vista de implementación se deberían justificar los recursos y las habilidades necesarias para ejecutar el proyecto.
7. **Vista de Decisiones de Arquitectura:** La vista de decisiones de arquitectura presenta y describe las principales decisiones de arquitectura tomadas sopesando las diferentes alternativas que se presentaban al momento de la decision.

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

Las partes interesadas en la plataforma *eparticipation.uy* son:

1. **Ciudadano:** Usuario de la plataforma *eparticipation.uy* que puede proponer iniciativas e integrarse en procesos participativos de interés, colaborando a través de distintos instrumentos como votaciones, encuestas, entre otros; e interactuando con dichas propuestas.
2. **Funcionarios:** Usuario de la plataforma *eparticipation.uy* que deberá crear procesos participativos y gestionar los aquellos propuestos por distintos organismos.
3. **Autoridades:** Usuario de la plataforma *eparticipation.uy* que se encargará de procesar las iniciativas propuestas por los diferentes ciudadanos de la plataforma, teniendo en cuenta las distintas evaluaciones que solicitará y en base a eso decidir si una iniciativa es válida para llevar a cabo. Generará reportes con métricas referentes a todos los elementos de la plataforma.
4. **Administradores:** Usuario de la plataforma *eparticipation.uy* que podrá gestionar y modificar todos los elementos relacionados a la plataforma.
5. **AGESIC:** Es la Agencia de Gobierno electrónico y Sociedad de la Información de Uruguay, unidad ejecutora dependiente de la Presidencia de la República, con autonomía técnica y que proveerá datos específicos de los usuarios de la plataforma.
6. **Gub.uy:** Servicio encargado de ejecutar el inicio de sesión de ciudadanos y funcionarios en nuestra plataforma *eparticipation.uy*.
7. **Evaladores:** Encargados de evaluar las iniciativas a pedido de las autoridades. Su evaluación servirá de ayuda para decidir si una iniciativa es viable.
8. **Organizaciones:** Tienen la potestad de proponer iniciativas en la plataforma *eparticipation.uy*, ver su estado actual y ser notificados ante eventuales cambios y/o aprobación.

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 Procesos	Vista de Implementación	Vista de Datos	Vista de Servicios	Vista de Distribución	Vista de Decisiones de Arquitectura
Ciudadanos	✓		✓	✓		✓	✓	✓		
Funcionarios	✓		✓	✓		✓	✓	✓		
Autoridades	✓		✓	✓		✓	✓	✓		
Administradores	✓	✓	✓	✓		✓	✓	✓		
AGESIC		✓		✓	✓	✓	✓	✓	✓	✓
Gub.uy		✓		✓	✓	✓	✓	✓	✓	✓
Evaluadores	✓		✓	✓	✓	✓	✓	✓	✓	✓
Organizaciones	✓		✓	✓	✓	✓	✓	✓	✓	✓

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 punto describiremos la plataforma `eparticipation.uy`, detallando los aspectos más relevantes. Nombraremos los subsistemas que la componen y la relevancia de los mismos.

En los últimos años han surgido varias plataformas que dan soporte a la e-participation ya que ha surgido la necesidad de que todas las personas puedan adquirir conocimientos y competencias para ejercer sus derechos y obligaciones en el entorno digital. `eparticipation.uy` dispone facilitar la participación ciudadana digital, fortaleciendo y ampliando los espacios y procesos participativos existentes, a nivel local y nacional.

El sistema se comunicará con la plataforma de Interoperabilidad (PDI) que brinda acceso a distintos servicios ofrecidos por organismos, con el fin de obtener datos de los ciudadanos.

El sistema se comunicará con nodos periféricos, estos serán: Evaluadores (que se da cuando una autoridad solicita la evaluación de una iniciativa) y Organizaciones (proponen iniciativas). Los Ciudadanos tendrán acceso a dos subcomponentes del componente central. Uno es el front-office, en el cual podrán ingresar sin autenticarse, y visualizar información básica de iniciativas anteriormente propuestas. Si se autentican, podrán proponer iniciativas o adherirse a ellas (comentarlas, compartirlas y seguirlas) y también podrá configurar el tipo de notificaciones que desea recibir, como alertas ante fechas límites o cambios de estado de iniciativas en las que participa, sigue o propone.

El segundo subcomponente se trata de uno móvil, que también permitirá al ciudadano autenticarse en el sistema, brindándole la posibilidad de recibir notificaciones de acuerdo a la configuración realizada en el front-office.

Los funcionarios también dispondrán de un subcomponente front-office. Tendrán la posibilidad de autenticarse en el sistema para tener acceso a la gestión de procesos participativos. Podrán crear procesos participativos, definir sus fases e instrumentos, supervisar su ejecución y tomar decisiones en base a la participación ciudadana en el marco del proceso.

Las autoridades dispondrán de un subcomponente de back-office que les dará la posibilidad de autenticarse en el sistema para tener acceso al procesamiento de iniciativas, esto les permitirá definir qué tipo de evaluaciones necesita, decidir si es publicada, supervisar su ejecución pudiendo exportar las adhesiones que tiene hasta el momento, si la iniciativa se llevará finalmente a cabo y, finalmente, notificar su decisión.

Finalmente, los administradores dispondrán de un subcomponente de back-office que les permitirá luego de autenticarse, la gestión de usuarios (alta, baja, modificación y consulta) de usuarios y roles. Así mismo también podrán gestionar nodos periféricos.[3].

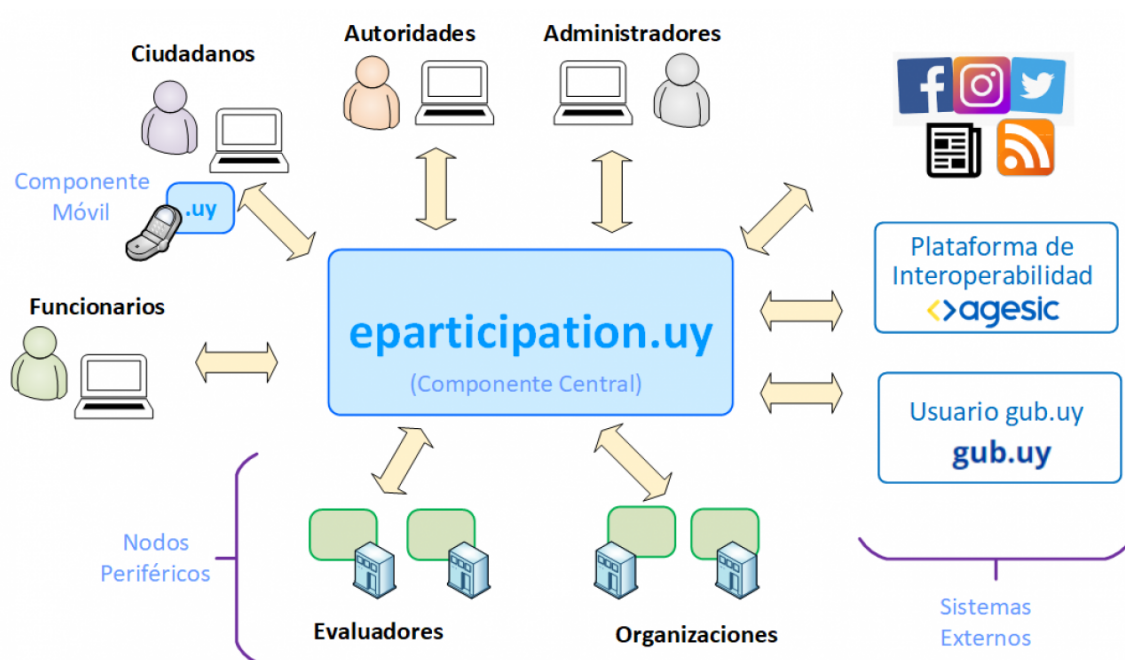


Figura 1: Descripción General de la Plataforma

2.2. Modelo Conceptual

El modelo conceptual abstrae los aspectos más importantes desde el punto de vista arquitectónico.

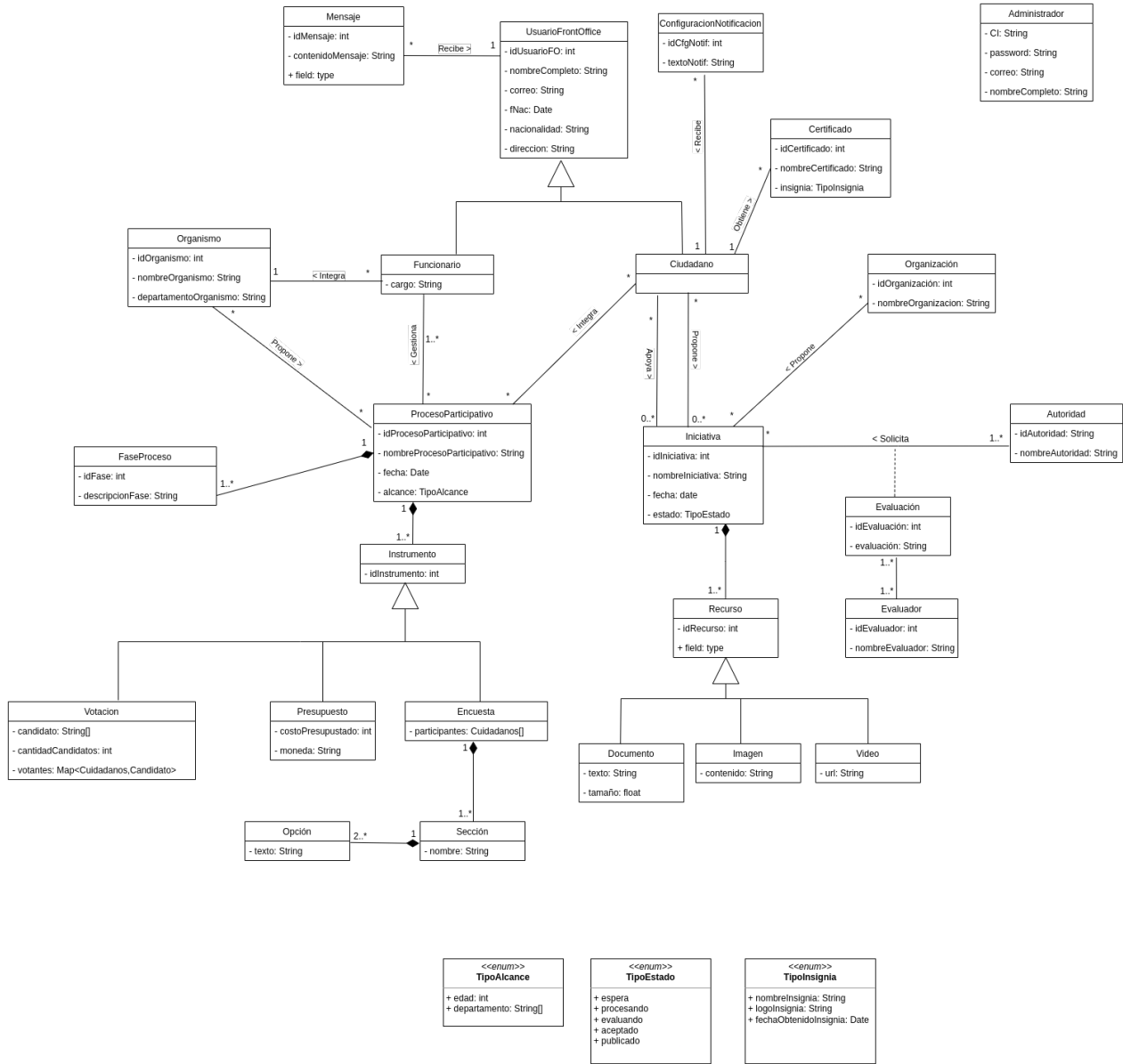


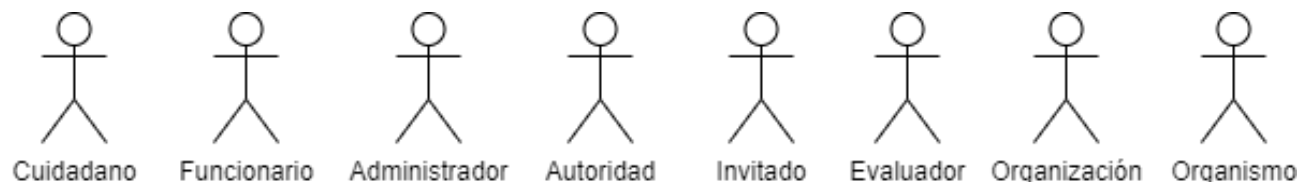
Figura 2: Modelo Conceptual

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 *eparticipation.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, indicando cuáles se consideran críticos para la arquitectura.

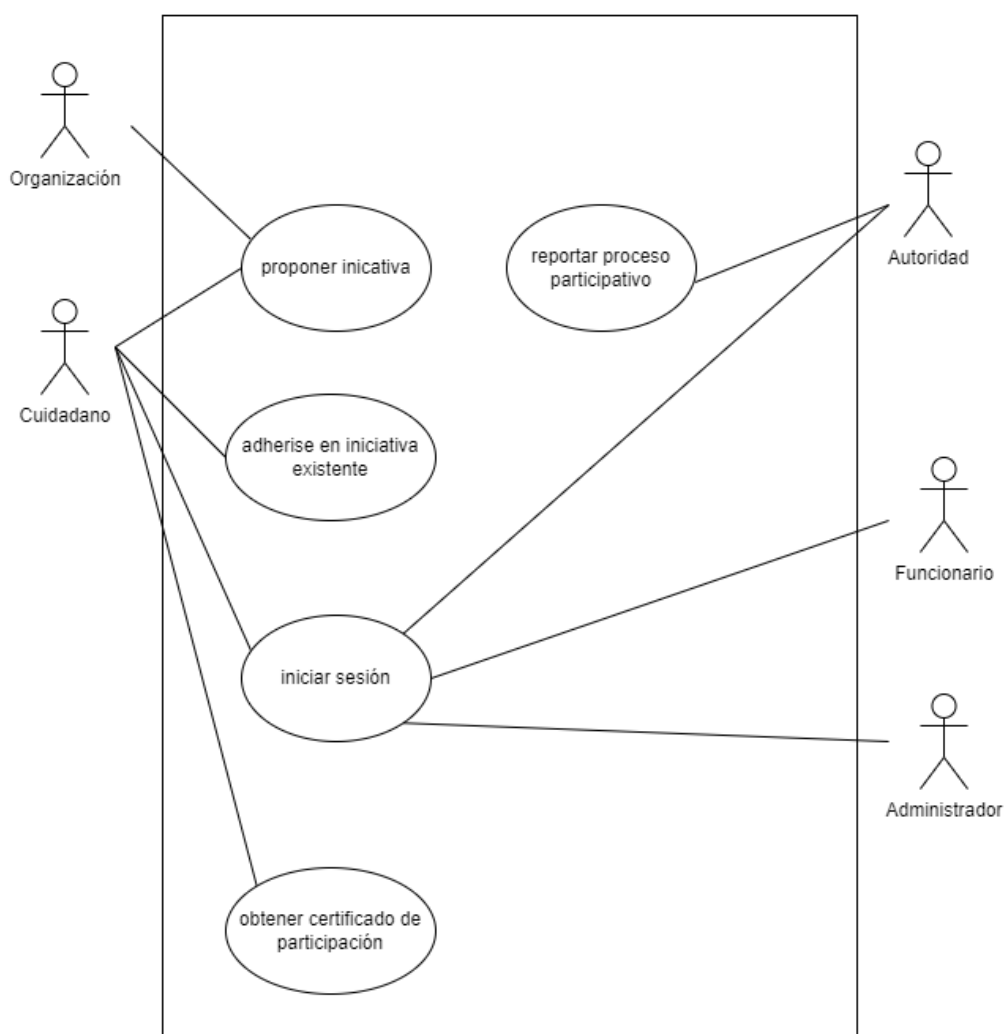


Figura 3: Vista de Casos de Uso

3.3. Iniciar Sesión - Crítico

Descripción

El caso de uso comienza cuando un usuario decide loguearse en la plataforma eparticipation.uy, debe ingresar sus credenciales.

Actores

Ciudadano, Funcionario, Autoridad, Administrador

Pre-condiciones

El usuario debe estar registrado en el sistema.

Flujo de Eventos

1. El usuario brinda sus credenciales
2. El sistema valida las credenciales
 - 2.1. El sistema autoriza al usuario
 - 2.1.1. Se redirecciona al usuario a su página principal
 - 2.2. El sistema no autoriza al usuario
 - 2.2.1. Se solicita al usuario sus credenciales nuevamente

Post-condiciones

Se crea una sesión para el usuario

3.4. Proponer Iniciativa - Crítico

Descripción

El caso de uso comienza cuando un actor decide proponer una nueva iniciativa, para ello se debe ingresar un nombre, una descripción de la iniciativa y opcionalmente una imagen o video.

Actores

Ciudadano, Organización

Pre-condiciones

El actor debe estar logueado en el sistema en caso de ser un ciudadano.

Flujo de Eventos

1. El usuario indica que desea proponer una iniciativa
2. El sistema le solicita al usuario que ingrese los datos requeridos.
3. El usuario ingresa los datos y hace clic en Enviar
 - 3.1. No falta ningún dato
 - 3.1.1. La iniciativa ya existe

- 3.1.1.1. Se le indica al usuario que ya existe una iniciativa con ese nombre.
- 3.1.2. La iniciativa no existe
 - 3.1.2.1. Se realiza el alta de una iniciativa con estado PENDIENTE
- 3.2. Faltan datos
 - 3.2.1. Faltan datos obligatorios por ingresar.

Post-condiciones

Se crea una nueva iniciativa con estado espera.

3.5. Adherirse a Iniciativa - Critico

Descripción

El caso de uso comienza cuando un actor decide adherirse a una iniciativa, para ello se debe confirmar la decisión especificando si el usuario quiere o no seguir a la iniciativa, luego se agrega al usuario a una lista publica de ciudadanos adheridos.

Actores

Ciudadano

Pre-condiciones

- El actor debe estar logueado en el sistema.
- El actor no debe estar adherido a la iniciativa.

Flujo de Eventos

1. El usuario indica que desea adherirse a una iniciativa
2. El sistema agrega al usuario a una lista publica (accesible por otros usuarios) con otros ciudadanos que tambien estan adheridos a la iniciativa.
3. El sistema agrega la iniciativa a las iniciativas que esta adherido dicho ciudadano.
4. El sistema configura las notificaciones de nuevas noticias de la iniciativa en cuestion para el usuario.

Post-condiciones

- Se agrega la iniciativa a la lista de adhecciones del ciudadano.
- Se agrega al ciudadano a la lista de adheridos de la iniciativa.

3.6. Reportar Proceso Participativo - Critico

Descripción

El caso de uso comienza cuando un actor decide reportar un proceso participativo, para esto se le muestra un dashboard con información acerca de los elementos de la plataforma, se le ofrece la opción de aplicar filtros y de obtener un reporte del proceso participativo en formato pdf o csv.

Actores

Autoridad

Pre-condiciones

El actor debe estar logueado en el sistema.

Flujo de Eventos

1. El usuario indica que desea reportar un proceso participativo.
2. El sistema muestra un dashboard con la información de los elementos de la plataforma.
3. El sistema ofrece la funcionalidad de filtrar la información ofrecida así como descargar un reporte de los procesos en formato pdf o csv.
4. El actor decide descargar un reporte de los procesos participativos.
5. El sistema pregunta al actor si prefiere formato pdf o csv.
6. Se descarga el reporte de procesos en el formato solicitado.

3.7. Obtener Certificado de Participacion - Critico

Descripcion

Actores

Ciudadano Movil

Pre-condiciones

El actor debe estar logueado en el sistema.

Flujo de Eventos

1. El sistema encuentra un nuevo logro conseguido por parte de un ciudadano.
2. El sistema asigna ese logro con el nivel correspondiente (por ejemplo: oro, plata, bronce)
3. El ciudadano podra ver en su perfil una nueva insignia correspondiente al logro alcanzado (por ejemplo: X cantidad de iniciativas adheridas)

Post-condiciones

Se asigna un nuevo logro al ciudadano (visible desde ciudadano movil)

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 eparticipation.uy 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 Unify Modeling Language (UML)

Interfaz Mobile

El usuario será capaz de acceder al sistema utilizando cualquier dispositivo Android que posea 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 (90.0), Firefox (Versión 78.10), Edge (Versión 90.0) y Opera (76.0.4).

4.3. Tecnológicas

El desarrollo de la aplicación debe de estar hecho utilizando

■ BackEnd

- **Componente central**

- Jakarta EE
- Wildfly
- Elastic Cloud

- **Servicios Externos**

- AGESIC - eparticipation.uy
- gub.uy

■ FrontEnd

- **Web**

- JSF
- ReactJS

- **Mobile**

- Kotlin
- Android Nativo

4.3. Módulos Existentes

- **AGESIC - eparticipation.uy**

Es el nodo externo de la Plataforma de Interoperabilidad para obtener acceso a los distintos servicios ofrecidos por los organismos públicos.

- **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

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

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

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 eparticipation.uy respetan la normativa vigente referente al área de salud, seguridad de la información y privacidad, y al intercambio de información

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 85 % de la lógica de negocio y servicios con pruebas unitarias, garantizando así el funcionamiento de todos los casos de uso.

3. Performance

- Se analizarán posibles cuellos de botella y se realizarán testeos de performance para mejorar posibles puntos de quiebre del sistema.

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

Vista de alto nivel de la arquitectura lógica del sistema.

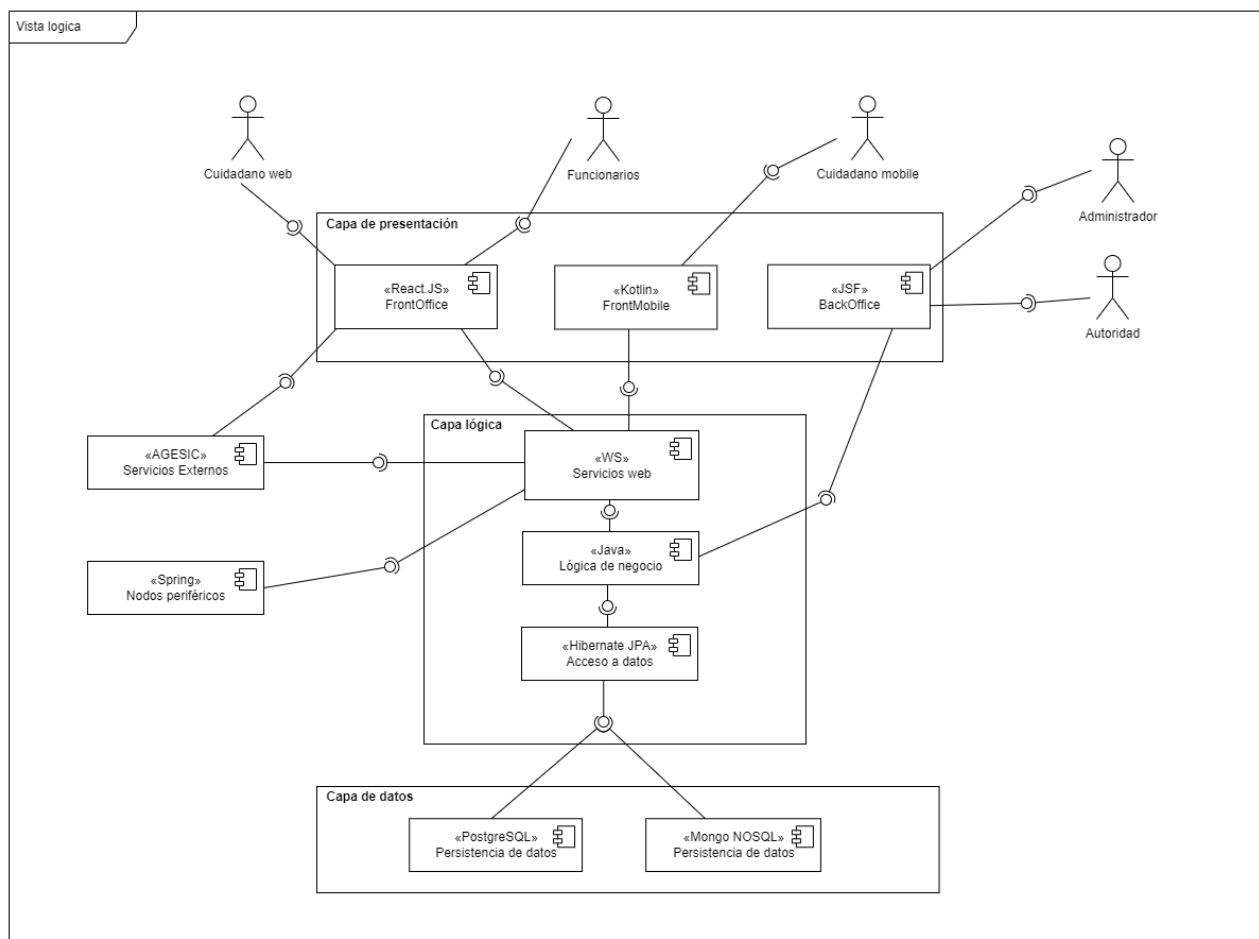


Figura 4: Arquitectura general en capas del sistema eparticipation.uy.

6.2. Refinamiento 1

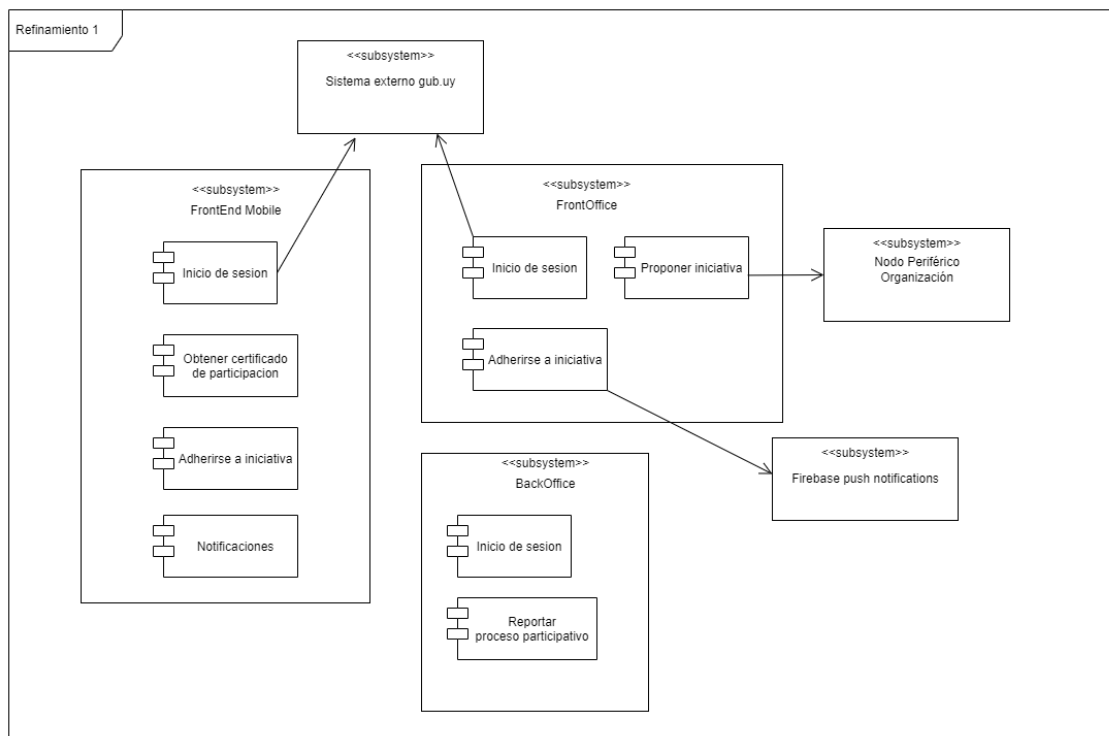


Figura 5: Vista Lógica Refinamiento 1

6.3. Refinamiento 2

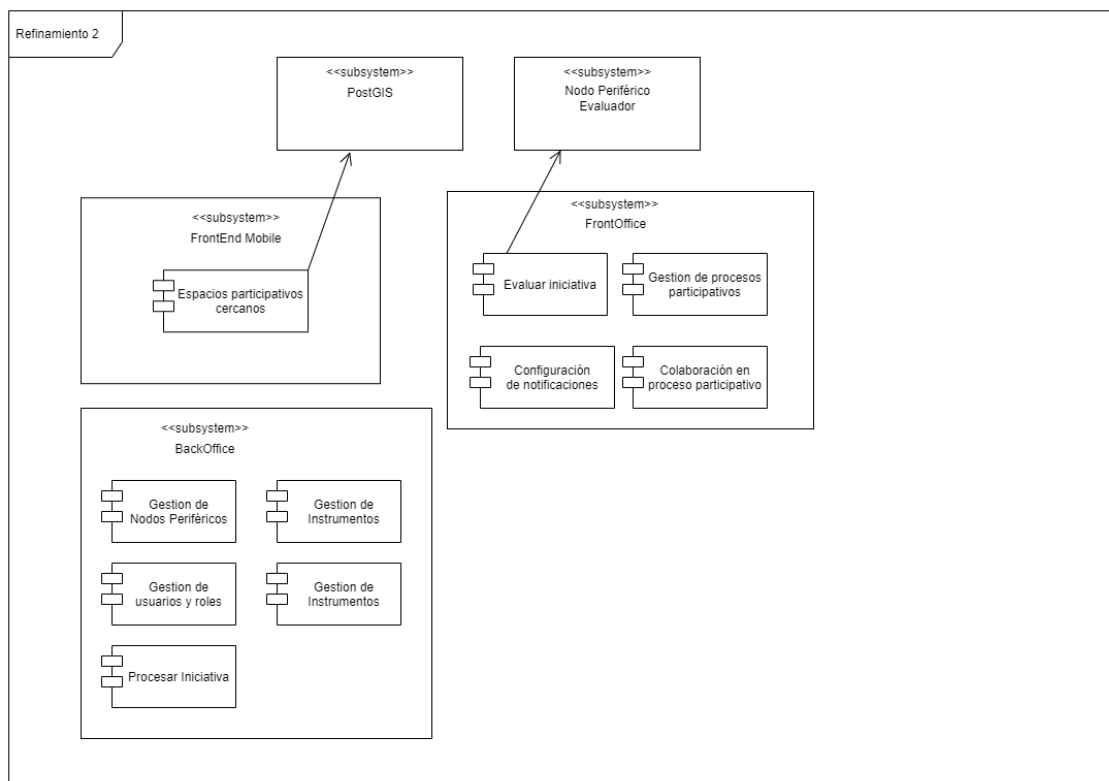


Figura 6: Vista Lógica Refinamiento 2

6.4. Diagramas de Interacción

En esta sección se incluyen diagramas de interacción, vinculados a los casos de uso críticos, que se considere apropiado especificar con mayor detalle.

Inicio de sesión

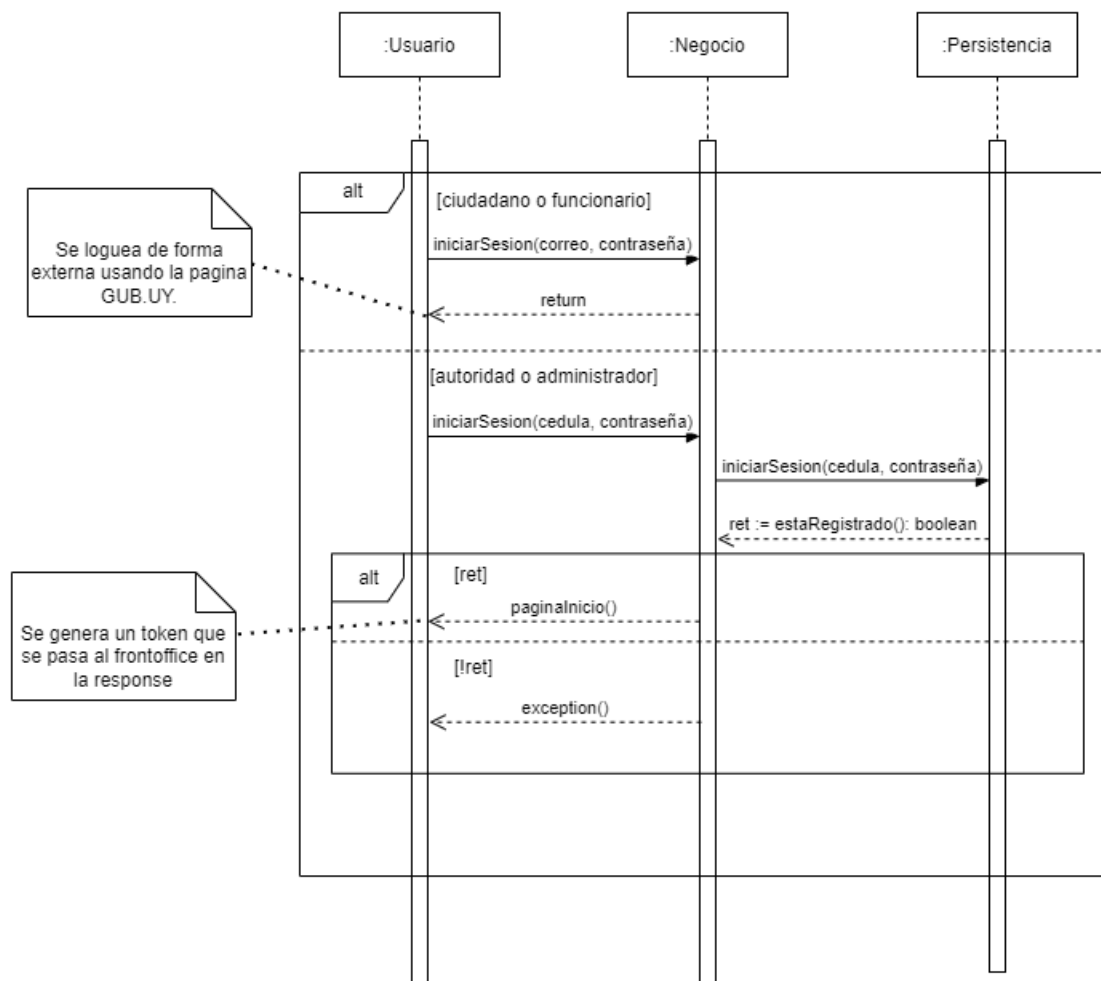


Figura 7: Diagrama de caso de uso crítico 1: Iniciar Sesión

Proponer iniciativa

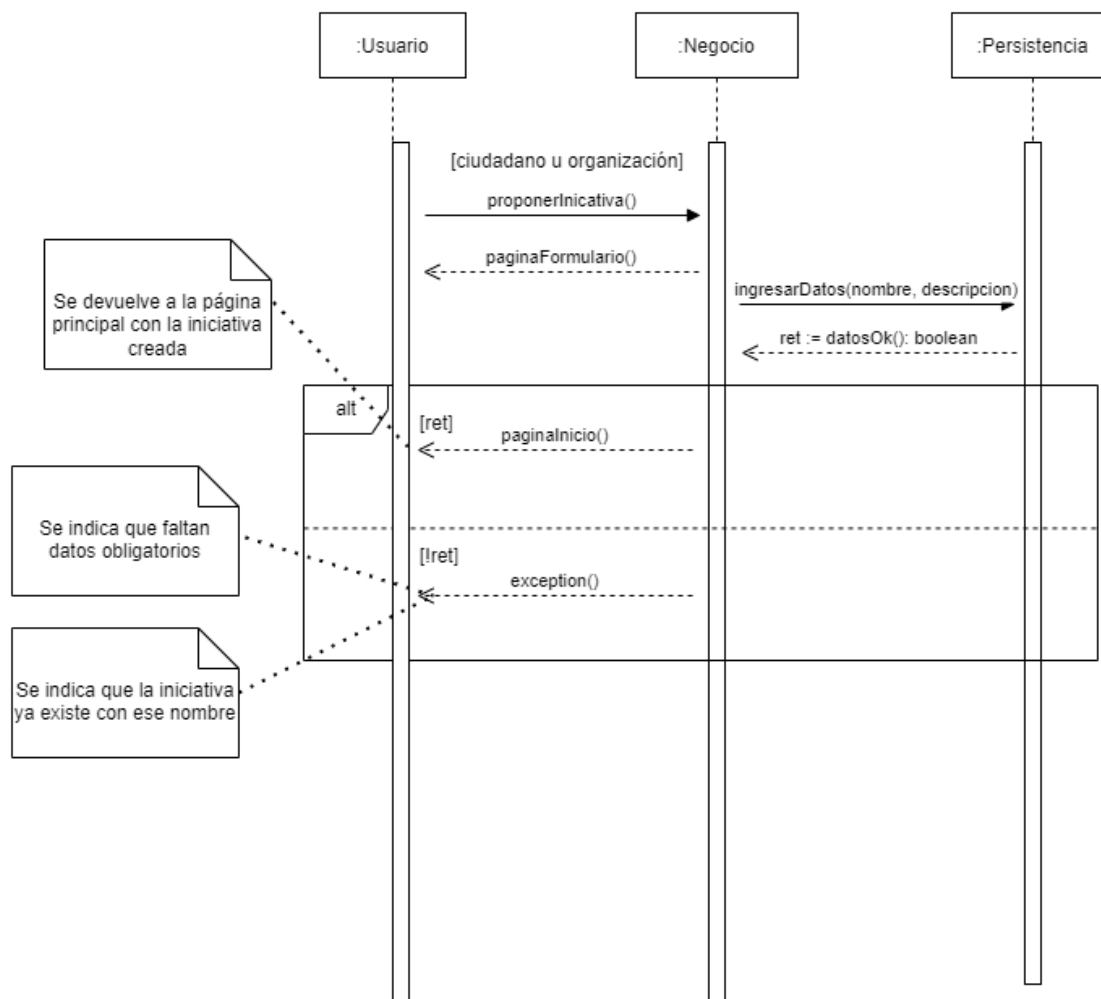


Figura 8: Diagrama de caso de uso crítico 2: Proponer Iniciativa

Adherirse a iniciativa

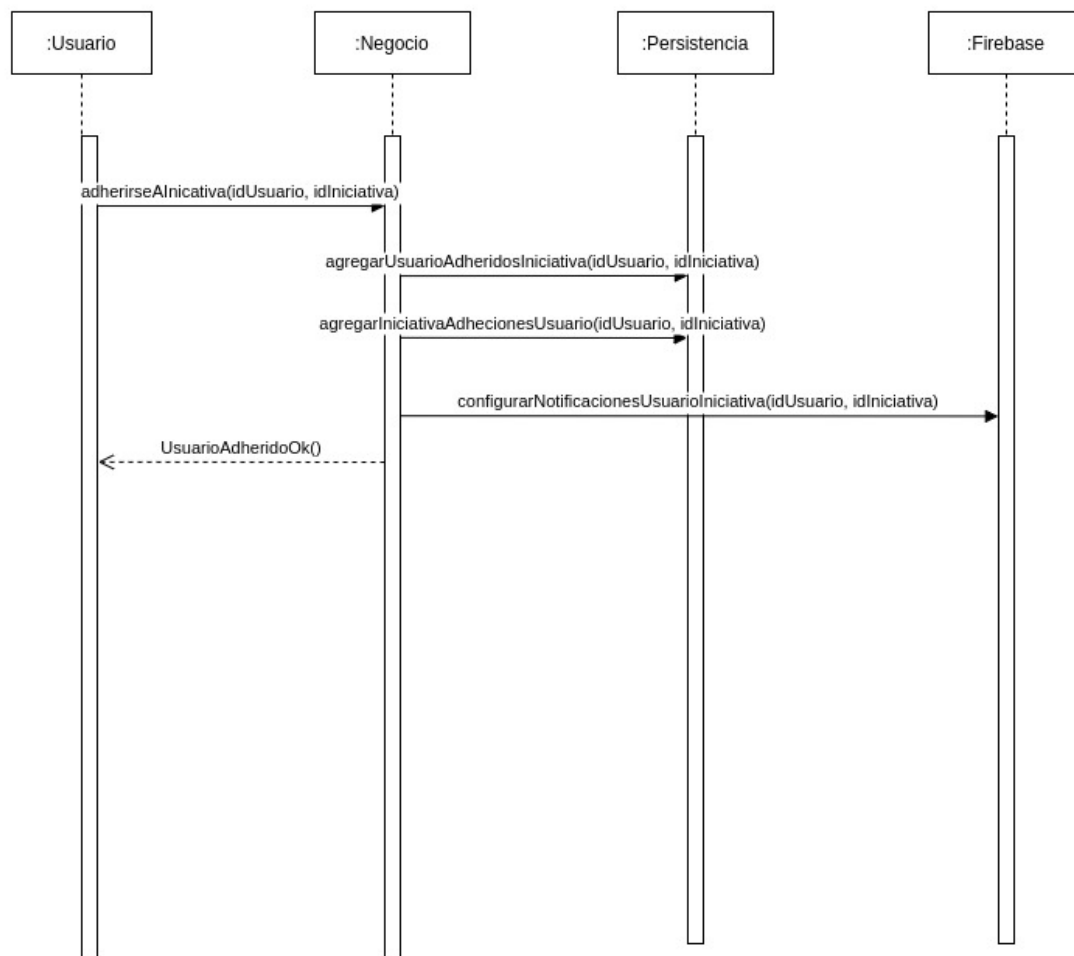


Figura 9: Diagrama de caso de uso crítico 3: Adherirse a Iniciativa

Reportar proceso participativo

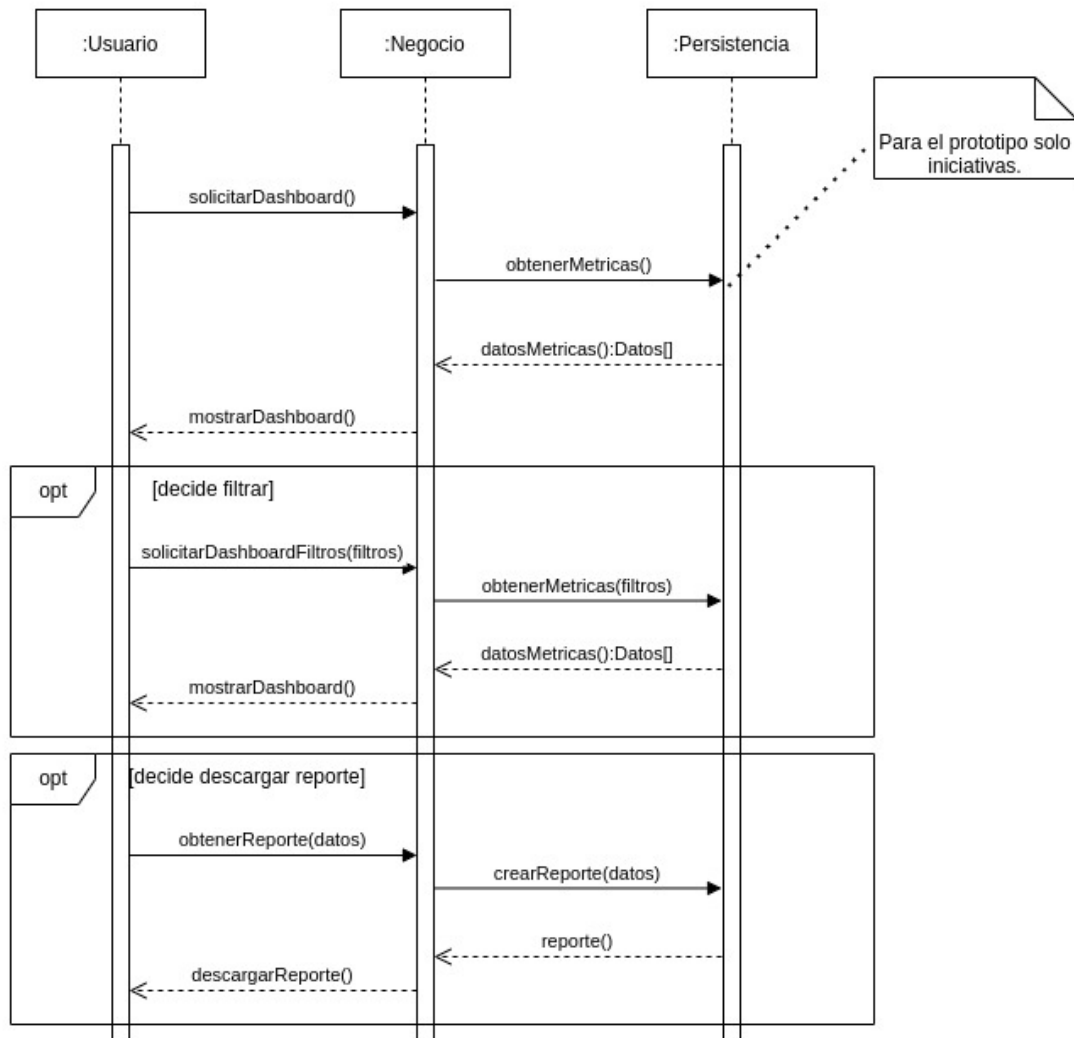


Figura 10: Diagrama de caso de uso crítico 4: Reportar proceso participativo

Obtener certificado de participacion

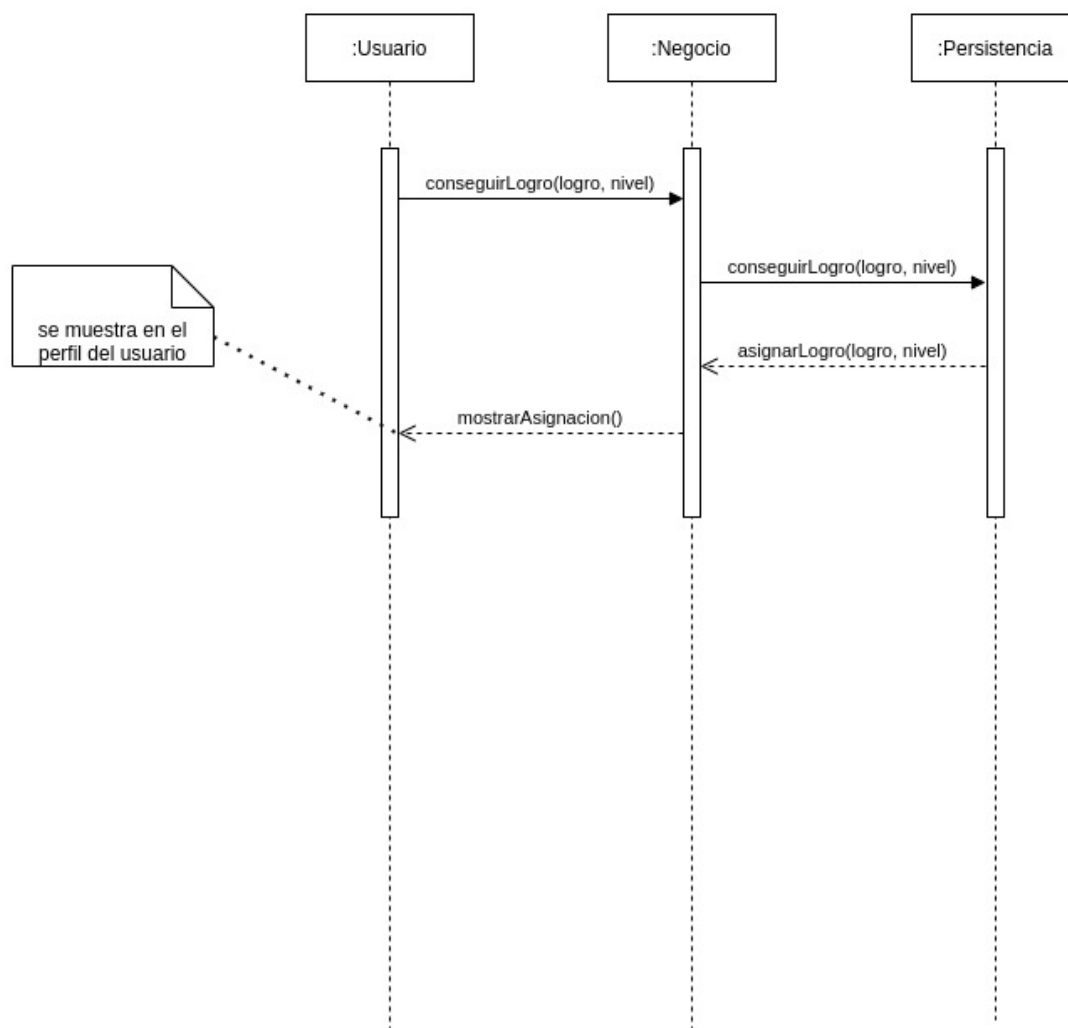


Figura 11: Diagrama de caso de uso crítico 5: Obtener certificado de participacion

7. Vista de Distribución

En la a vista de despliegue se muestra la arquitectura del sistema eparticipation.uy desde el punto de vista de la distribución de sus componentes del software, así como la relación entre ellos. Dichos componentes pueden ser tanto nodos físicos como nodos de software

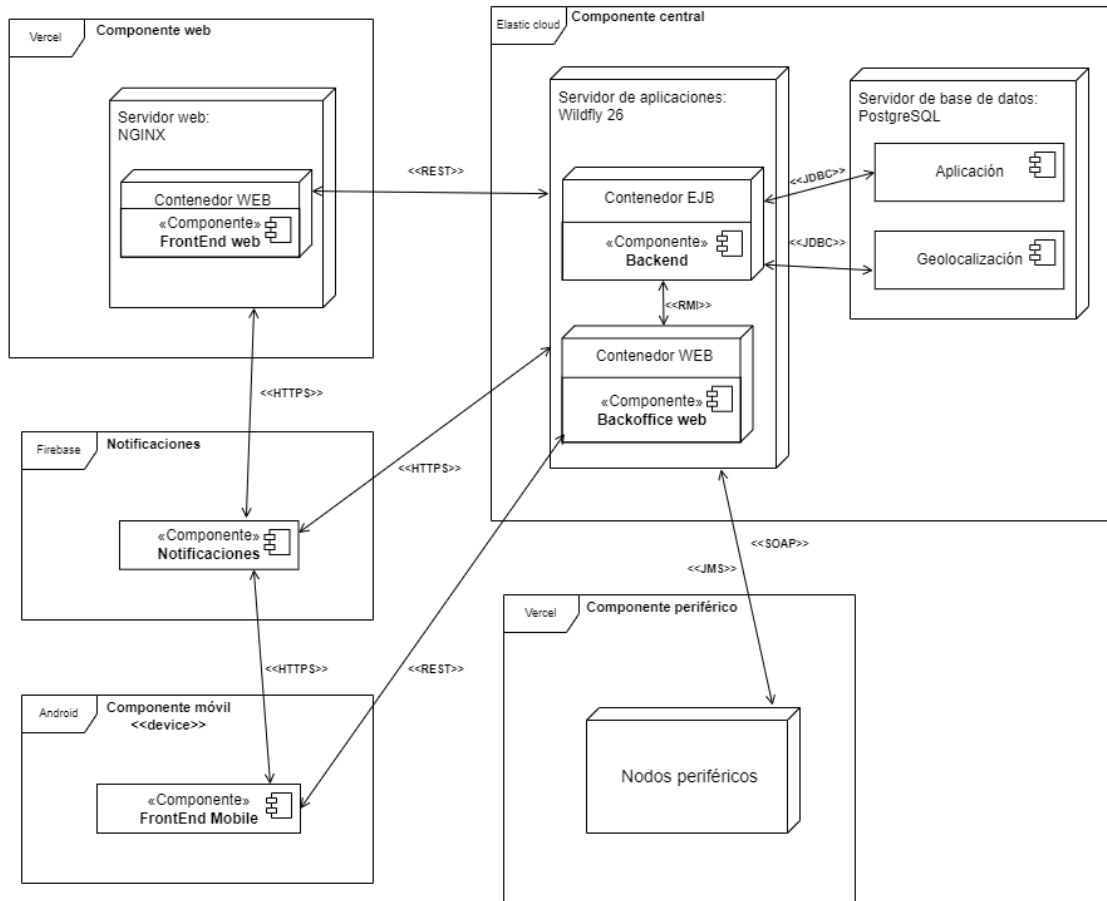


Figura 12: Diagrama de despliegue eparticipation.uy.

7.1. Escenario 1

Performance entre componente central y componente móvil - Se debe dar una comunicación veloz y sin interrupciones la cual permitirá a los usuarios del componente móvil tener una interfaz fluida. Esto se logrará a través de servicios web REST expuestos por el componente central.

7.2. Escenario 2

Escalabilidad del sistema - Se debe contemplar el uso del componente central en caso de tener más conexiones de las que nuestro servidor permite. Se configurará un escalamiento horizontal que permite aumentar la carga que recibirá el servidor de aplicaciones del componente central.

7.3. Escenario 3

Procesamiento de coordenadas - Se contempla el caso en que se deban guardar y/o procesar ubicaciones o coordenadas, para lo cual usamos PostGis, que es una extension de Postgre en la cual procesamos coordenadas para sacarle trabajo al Frontoffice y de esa forma darle dinamismo al uso de la aplicacion.

8. Vista de Implementación

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

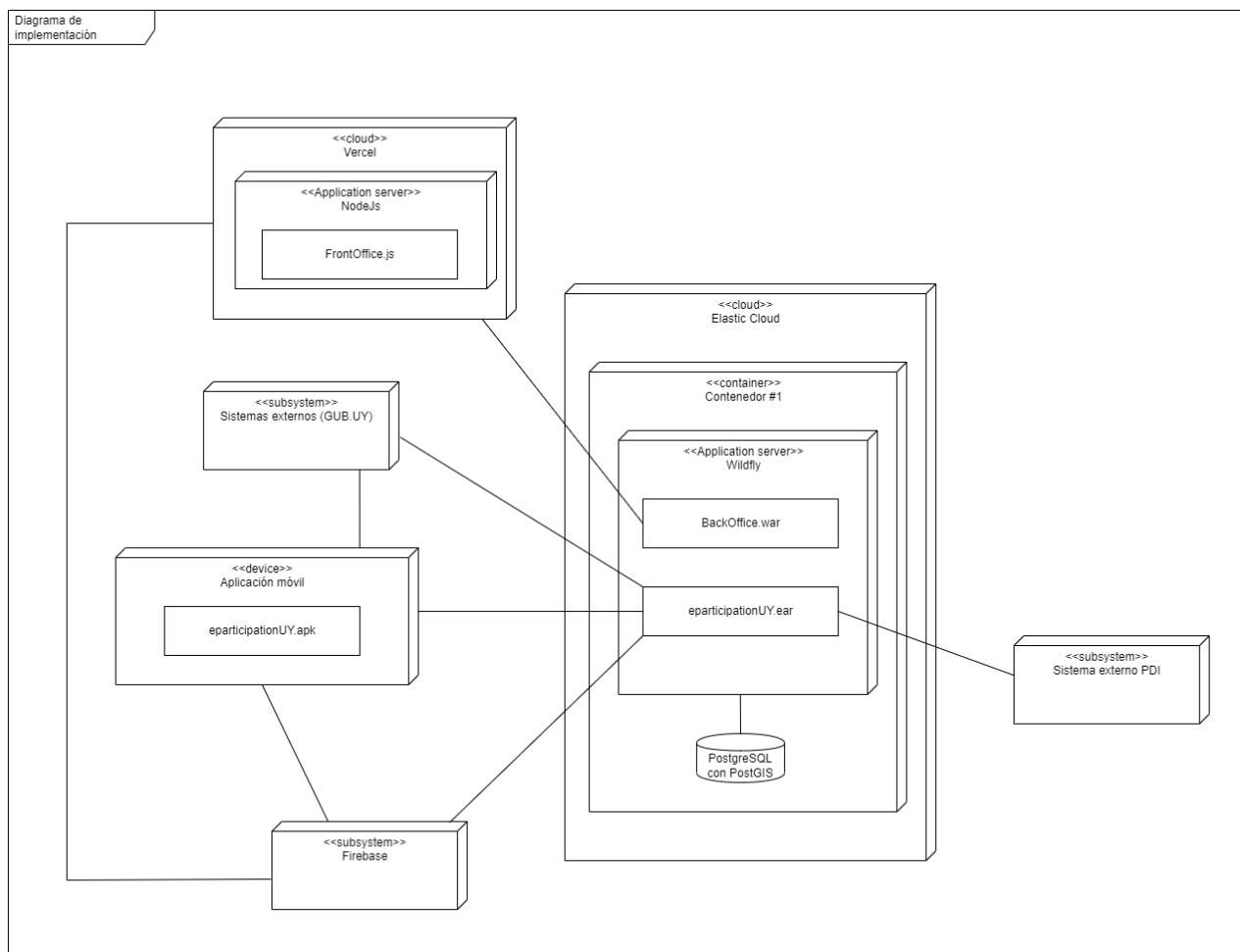


Figura 13: Diagrama de implementación

9. Vista de Decisiones de Arquitectura

La Vista de Decisiones de Arquitectura presenta y describe las principales decisiones de arquitectura tomadas [4][5].

Tabla 2: Decisión de Arquitectura 1

Identificador	D01
Nombre	Plataforma
Categorías	Componente central, Componente móvil
Problema	La solución eparticipation.uy debe contar con un componente central y un componente móvil que interctúan con nodos periféricos y externos. Estos elementos permiten brindar funcionalidades a cuatro tipos de usuarios: ciudadanos, funcionarios, autoridades y administradores.
Alternativas	- - -
Decisión	Se implementará una solucion con JakarteEE y Android
Justificación	Requerimiento del cliente.
Decisiones Relacionadas	D02.

Tabla 3: Decisión de Arquitectura 2

Identificador	D02
Nombre	Tecnologías de desarrollo
Categorías	Componente central, Componente móvil
Problema	Se debe definir el software que satisfaga los requerimientos del sistema a desarrollar.
Alternativas	JakartaEE, PostgreSQL, PostGIS***, Java(Android), Kotlin(Android)
Decisión	JakartaEE, PostgreSQL, PostGIS***, Kotlin(Android)
Justificación	Requerimiento del cliente. Sistema empresarial de software libre.
Decisiones Relacionadas	D03, D04, D05.

Tabla 4: Decisión de Arquitectura 3

Identificador	D03
Nombre	Distribucion de sistema
Categorías	Plataforma
Problema	Distruibir o no distruibir
Alternativas	- - -
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	Componente de acceso a datos
Problema	Visualizar espacios participativos que refieren a zonas cercanas a la ubicación del ciudadano.
Alternativas	- - -
Decisión	Se implementa un módulo en la base de datos que gestione el Sistema de información Geografica, PostGIS.
Justificación	Se evalúa que PostGIS es una buena solución libre y es la más adecuada para solucionar el problema que nos plantea el cliente.
Decisiones Relacionadas	- - -

Tabla 6: Decisión de Arquitectura 5

Identificador	D05
Nombre	Servidor de aplicaciones
Categorías	Plataforma
Problema	Determinar el servidor de aplicaciones a utilizar.
Alternativas	WildFly, GlassFish, Open Liberty, ManageFish
Decisión	Wildfly
Justificación	Requerimiento del cliente.
Decisiones Relacionadas	- - -

Tabla 7: Decisión de Arquitectura 6

Identificador	D06
Nombre	Autenticación de ciudadanos y funcionarios
Categorías	Componente central, Componente móvil
Problema	Se debe autenticar el ciudadano y el funcionario mediante el sistema externo GUB.UY.
Alternativas	- - -
Decisión	Se integra el sistema a través de llamadas HTTP GET y POST.
Justificación	Requerimiento del cliente.
Decisiones Relacionadas	- - -

Tabla 8: Decisión de Arquitectura 7

Identificador	D07
Nombre	Framework Frontend
Categorías	Componente web
Problema	Se debe decidir la tecnología frontend a utilizar para el desarrollo de la solución.
Alternativas	ReactJS, Angular, VueJS
Decisión	Evalutando las opciones decidimos utilizar ReactJS.
Justificación	Experiencia del grupo de desarrollo en la tecnología.
Decisiones Relacionadas	- - -

Tabla 9: Decisión de Arquitectura 8

Identificador	D08
Nombre	Seguridad
Categorías	Componente central, componente web, componente mobile
Problema	Se debe utilizar una conexión segura entre las conexiones del componente central con los servicios externos, periféricos y demás componentes. Se debe encriptar las contraseñas.
Alternativas	- - -
Decisión	Se utilizan access token para el acceso de REST API y mediante canal HTTPS, JWT como formato para intercambio de información. En cuanto al manejo de contraseñas se encriptan mediante una función de hash BCrypt.
Justificación	JWT es un formato estándar, compacto y seguro de transmitir información entre diferentes sistemas. BCrypt es un algoritmo diseñado específicamente para hash de contraseñas que permite agregar un salt al proceso de generación del hash.
Decisiones Relacionadas	- - -

Tabla 10: Decisión de Arquitectura 9

Identificador	D09
Nombre	Escalabilidad
Categorías	Servidor de aplicaciones
Problema	Se requiere que el sistema pueda soportar una gran cantidad de usuarios activos sin que comprometa su tiempo de respuesta.
Alternativas	- - -
Decisión	Se dividió el sistema en varios módulos independientes para lograr la escalabilidad horizontal. En el caso de Wildfly se puede escalar utilizando las facilidades de Elastic Cloud, generando un modelo de escalado por varios nodos dependiendo del uso.
Justificación	Se hace necesaria la escalabilidad para poder responder a las necesidades de rendimiento conforme el número de usuarios y las transacciones aumentan y la base de datos comienza a sufrir degradamiento de la performance por las cargas crecientes.
Decisiones Relacionadas	- - -

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] LINS. *Letra Trabajo Laboratorio 2022 - Taller de Sistemas Empresariales*. Inf. téc. 2022.
- [4] 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.
- [5] 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.