



Upc-pre-202202-si720-WS32-Pet Solution-report-tf

Diseño y Patrones de Software (Universidad Peruana de Ciencias Aplicadas)



Scan to open on Studocu

Universidad Peruana de Ciencias Aplicadas

Ingeniería de Software

SI720 - Diseños y patrones de software – WS32

Docente: Efraín Ricardo Bautista Ubillús

MAG 28 C

FINAL PROJECT

Informe de Trabajo Final

StartUp: PetSolution

Nombre del producto:

PetCare



Team Members

Ayasta Leonardo, Valery Ruth **U20211C231**

Espejo Macuri, Paolo Andre **U20211B786**

Espinosa Quispe, Jennifer Mary **U202120911**

Seminario Garbín, Carlo Luca **U20211A475**

Soriano Huapaya, Isabella María **U20211C242**

Ciclo 2022- 02

Link de video de exposición: <https://youtu.be/N72K9OdHXto>

This document is available free of charge on



Downloaded by Paul Sulca Gonzales (kyrtobi320@gmail.com)

REGISTRO DE VERSIONES DEL INFORME

Entregable	Fecha	Participantes	Observaciones
TB1	01/09/22	Ayasta Leonardo, Valery Ruth Espejo Macuri, Paolo Andre Espinoza Quispe, Jennifer Mary Seminario Garbín, Carlo Luca Soriano Huapaya, Isabella María	Descripción de la información e introducción del proyecto. Además de los objetivos, requisitos funcionales, no funcionales y los escenarios AS-IS / TO-BE.
TB2	16/09/22	Ayasta Leonardo, Valery Ruth Espejo Macuri, Paolo Andre Espinoza Quispe, Jennifer Mary Seminario Garbín, Carlo Luca Soriano Huapaya, Isabella María	Redacción de las Historias de Usuario, Product Backlog, Sprint Backlogs, Application Wireframes y los diagramas de código como Diagrama de Actividades, Diagrama de Clase, Diagrama de Objetos, Diagrama de secuencia, Diagrama de Estados
TP	28/09/22	Ayasta Leonardo, Valery Ruth Espejo Macuri, Paolo Andre Espinoza Quispe, Jennifer Mary Seminario Garbín, Carlo Luca Soriano Huapaya, Isabella María	Mejora de los requisitos funcionales y aplicación de patrones creacionales y de comportamiento al diagrama de clases.
TB3	28/10/22	Ayasta Leonardo, Valery Ruth Espejo Macuri, Paolo Andre Espinoza Quispe, Jennifer Mary Seminario Garbín, Carlo Luca Soriano Huapaya, Isabella María	Redacción de los módulos del sistema. Creación de los diagramas C4: Diagrama de contexto, Diagrama de contenedores, Diagrama de Componentes. Redacción des los patrones empleados.
TB4	11/11/22	Ayasta Leonardo, Valery Ruth Espejo Macuri, Paolo Andre Espinoza Quispe, Jennifer Mary Seminario Garbín, Carlo Luca Soriano Huapaya, Isabella María	Redacción del DDD Estratégico y DDD Táctico
TF	25/11/22	Ayasta Leonardo, Valery Ruth Espejo Macuri, Paolo Andre Espinoza Quispe, Jennifer Mary Seminario Garbín, Carlo Luca Soriano Huapaya, Isabella María	Se incluyeron las secciones de API REST, pruebas realizadas, fuera de alcance, restricciones, supuestos y principales riesgos

STUDENT OUTCOME

Criterio Específico	Acciones realizadas	Conclusiones
Demuestra ética profesional en el ejercicio de la ingeniería de software.	<p><u>Ayasta Leonardo, Valery Ruth</u></p> <p>TB1: Para esta entrega me encargué de redactar el propósito y alcance del proyecto que como grupo definimos, además de basarme en fuentes para ello.</p> <p>TB2: Para esta entrega me encargué de redactar el Product Backlog y los Sprint Backlogs ayudándome en crear los artefactos de materia académico confiable.</p> <p>TP: Realice mejoras a los requisitos funcionales revisando nuevamente el artefacto y sus componentes.</p> <p>TB3: Realice la redacción de los módulos que compondrán nuestra aplicación</p> <p>TB4: Realice la redacción de los diagramas correspondientes al Bounded Context de Security y Services, además del prototipo del Landige Page</p> <p>TF: Realice la redacción de las conclusiones y recomendaciones con respecto a los aprendizajes obtenidos con este trabajo.</p>	<p>TB1:</p> <p>Se concluye que la recolección de fuentes, así como aquellas brindadas por la universidad para elaborar los distintos artefactos nos permite lograr un mejor conocimiento del problema que deseamos solucionar y con ello la solución que buscamos brindar a un público objetivo.</p> <p>TB2:</p> <p>Concluimos que con la información brindada por las historias de usuario y el producto backlog podemos entender mejor a nuestro segmento objetivo y sus inquietudes.</p> <p>TP:</p> <p>Con las mejoras de los requisitos funcionales se concluye que es posible especificar mejor aquellas cualidades que nuestros clientes buscarían de nuestra aplicación.</p> <p>TB3:</p> <p>Se respeta el código de ética al emplear patrones empresariales que garantizarán que nuestro producto cumpla los mayores estándares profesionales posibles.</p> <p>TB4:</p> <p>Con el DDD estratégico y táctico logramos definir el dominio del negocio en el que busca involucrarse nuestra aplicación, de manera que se establece la lógica y la táctica a implementar para el desarrollo de esta.</p> <p>TF:</p> <p>Concluimos que con las secciones implementadas de la API REST, pruebas realizadas, fuera de alcance, restricciones, supuestos y principales riesgos; podemos cumplir mejor las metas para con este proyecto.</p>
	<p><u>Espejo Macuri, Paolo Andre</u></p> <p>TB1: Realice los requisitos funcionales y no funcionales apoyándose en la teoría de la creación del artefacto de mi curso de especificación y análisis y requerimientos del ciclo pasado.</p> <p>TB2: Realice el diagrama de secuencia y diagrama de objetos siguiendo la explicación del profesor e investigando más sobre el artefacto.</p> <p>TP: Incluyó al diagrama de clases los patrones de diseño enseñados en clase</p> <p>TB3: Realice la redacción de todos los patrones empleados, así como la creación de los diagramas de clases por modulo</p>	

	<p>TB4: Realice la redacción de todos los diagramas correspondientes al Bounded Context de Complaints book, Invitations, Opinions, así como los Bounded Context Canvas.</p> <p>TF: Realice la redacción del código de la API REST</p> <p style="text-align: center;"><u>Espinosa Quispe, Jennifer Mary</u></p> <p>TB1: Redacte los objetivos del proyecto previamente discutidos y acordados en mi grupo</p> <p>TB2: Grafique el diagrama de clases y el Application Wireframes prototipado de la aplicación e investigue más sobre los artefactos para elaborarlos correctamente.</p> <p>TP: Añadí al diagrama de clases los patrones de diseño presentados en clase</p> <p>TB3: Codifique los diagramas de componentes para cada módulo de nuestra aplicación.</p> <p>TB4: Añadí al EventStorming los diagramas de flujo del dominio y de los diagramas correspondientes al Bounded Context de Pet Sitters y Pets.</p> <p>TF: Añadí las secciones de supuestos y principales riesgos correspondientes a nuestro proyecto</p> <p style="text-align: center;"><u>Seminario Garbín, Carlo Luca</u></p> <p>TB1: Realice los diagramas AS-IS / TO BE apoyándose en la teoría de la creación del artefacto del material pedagógico ofrecido por la universidad.</p> <p>TB2: Realice los diagramas de actividades y para las historias de usuario recolecte información de las funcionalidades ofrecidas por la competencia.</p> <p>TP: Añadí mejoras al diagrama de clases para que se ajuste más al prototipo y funcionalidades de nuestra aplicación.</p>	
--	--	--

	<p>TB3: Realice la redacción de todos los patrones empleados, así como la creación de los diagramas de clases por modulo</p> <p>TB4: Realice la redacción de todos los de los diagramas correspondientes al Bounded Context de Advertise, Appointment, Benefits, así como los Bounded Context Canvas.</p> <p>TF: Realice la redacción del código de la API REST</p> <p><u>Soriano Huapaya, Isabella María</u></p> <p>TB1: En esta entrega redacte el resumen ejecutivo del proyecto correspondiente a la información del mismo basándome para ello en material de estudio de cursos anteriores.</p> <p>TB2: Grafiqué el diagrama de clases y los diagramas de estados en relación a las reglas de negocio más destacables en relación a nuestro proyecto</p> <p>TP: Realice mejoras a los requisitos funcionales revisando nuevamente el artefacto y sus componentes</p> <p>TB3: Codifique los diagramas de contexto y contenedores.</p> <p>TB4: Añadí al EventStorming los diagramas de flujo del dominio y de los diagramas correspondientes al Bounded Context de Owner y Pays.</p> <p>TF: Añadí las secciones de fuera de alcance y restricciones correspondientes a nuestro proyecto</p>	
Demuestra responsabilidad profesional para el logro de los objetivos	<p><u>Ayasta Leonardo, Valery Ruth</u></p> <p>TB1: Realice mis secciones señaladas a tiempo y correctamente.</p> <p>TB2: Entregue las secciones asignadas a mi persona a tiempo y acorde a la información brindada por mis compañeros</p>	

	<p>TP: Trabaje en par con mi compañera para lograr la mejora de los requisitos funcionales</p> <p>TB3: Realice mis secciones señaladas a tiempo y correctamente.</p> <p>TB4: Realice mis secciones señaladas a tiempo y correctamente</p> <p>TF: Realice mis secciones señaladas a tiempo y correctamente.</p> <p><u>Espejo Macuri, Paolo Andre</u></p> <p>TB1: Cumplí con entregar mis secciones en la fecha acordada por mi grupo.</p> <p>TB2: Cumplí en entregar a la fecha los diagramas que me fueron asignados</p> <p>TP: Con mi compañera termine de incluir los patrones de diseño a tiempo para la entrega del trabajo</p> <p>TB3: Cumplí con entregar mis secciones en la fecha acordada por mi grupo.</p> <p>TB4: Cumplí con las secciones asignada en la fecha acordada por mi grupo.</p> <p>TF: Cumplí con entregar mis secciones en la fecha acordada por mi grupo.</p> <p><u>Espinoza Quispe, Jennifer Mary</u></p> <p>TB1: Brinde oportunamente la entrega de mi sección para complementar satisfactoriamente la información sobre el proyecto a realizar.</p> <p>TB2: Termine con mi compañera el diagrama de clases a tiempo según los acordado en el grupo.</p> <p>TP: Con mi compañero termine de incluir los patrones de diseño a tiempo para la entrega del trabajo</p> <p>TB3: Realice la entrega de mi sección para complementar satisfactoriamente los diagramas de componentes</p>	<p>TB1: Concluimos que basándonos responsablemente en referencias de materiales académicos nos es posible trabajar en equipo armoniosamente.</p> <p>TB2: Concluimos que con la ayuda de plataformas especializadas para crear distintos diagramas de código nos es posible entregar un trabajo presentable y adecuado.</p> <p>TP: Se concluye con el uso de plataformas que permiten el uso compartido para trabajar en simultáneo nos beneficia en colaborar más eficientemente como equipo.</p> <p>TB3: Se concluye que, para la creación de los diagramas contextual, contenedores y componentes se utilizó Structurizr con lo cual pudimos trabajar los diagramas como código de modo que pudimos aplicar nuestros conocimientos de código para simplificar el trabajo de estos diagramas.</p> <p>TB4: Concluimos que el desarrollar el método del DDD fue más sencillo de elaborar gracias a la aplicación Miro, en donde nos facilita las gráficas para lo que fueron el EventStorming, los diagramas de flujo y los Bounded Context Canvas</p> <p>TF: Se concluye con el uso de plataformas como lo es Visual Studio, la codificación de la API termina siendo más limpia, ya que esta herramienta nos señala los errores que cometemos y por ende los podemos arreglar.</p>
--	--	---

	<p>TB4: Termine con mi compañera el EventStorming y los diagramas del flujo del dominio a tiempo según los acordado en el grupo.</p> <p>TF: Cumplí con entregar mis secciones en la fecha acordada por mi grupo.</p> <p style="text-align: center;"><u>Seminario Garbín, Carlo Luca</u></p> <p>TB1: Hice entrega oportuna de los diagramas de los escenarios AS- IS / TO BE en relación a lo acordado con mi grupo.</p> <p>TB2: Cumplí con redactar los diagramas de actividades y los User Stories oportunamente para que mi compañera pueda proseguir con el producto backlog y los sprint</p> <p>TP: Realizar las mejoras del diagrama de clases fue breve y entregado eficientemente al grupo</p> <p>TB3: Hice entrega oportuna de los diagramas de clase por modulo, así como la descripción de patrones usados</p> <p>TB4: Cumplí con las secciones asignada en la fecha acordada por mi grupo.</p> <p>TF: Cumplí con entregar mis secciones en la fecha acordada por mi grupo.</p> <p style="text-align: center;"><u>Soriano Huapaya, Isabella María</u></p> <p>TB1: Complete mi sección asignada puntualmente y apoye en complementar las ideas sobre la finalidad del proyecto</p> <p>TB2: Termine de completar con mi compañera el diagrama de clases a tiempo según los acordado en mi grupo.</p> <p>TP: Trabaje en par con mi compañera para lograr la mejora de los requisitos funcionales</p>	
--	--	--

	<p>TB3: Complete mi sección asignada puntualmente y apoye en la lógica de los diagramas de componentes</p> <p>TB4: Termine con mi compañera el EventStorming y los diagramas del flujo del dominio a tiempo según los acordado en el grupo.</p> <p>TF: Complete mi sección asignada puntualmente y apoye en complementar las ideas sobre la finalidad del proyecto</p>	
Emite juicios considerando el impacto de las soluciones de ingeniería de software en el contexto global, impacto social, ambiental y económico	<p><u>Ayasta Leonardo, Valery Ruth</u></p> <p>TB1: Al redactar el propósito y alcance del proyecto nos ayuda a comprender mejor el impacto que queremos alcanzar.</p> <p>TB2: Al redactar el Product Backlog y los Sprint Backlogs entiendo mejor las funcionalidades de la aplicación para brindar la solución deseada.</p> <p>TP: Con la mejora de los requisitos funcionales especificamos más las facilidades que la app brindara a mis usuarios.</p> <p>TB3: Con la redacción de los módulos llegamos a mostrará los beneficios de nuestra solución tecnológica</p> <p>TB4: Con la redacción de los diagramas para los Bounded Context logramos mostrará la táctica a seguir para la implementación de la aplicación.</p> <p>TF: Con la redacción de las conclusiones y redacciones nos damos cuenta de todos los logros obtenidos con este proyecto.</p> <p><u>Espejo Macuri, Paolo Andre</u></p> <p>TB1: Al redactar los requisitos funcionales y no funcionales pude considerar mejor las facilidades que la app brindara a mis usuarios.</p> <p>TB2: Los diagramas de secuencia y de objeto ofrece un mejor panorama del diseño de software para la aplicación</p>	<p>TB1: Concluimos que la problemática que deseamos solucionar tiene un gran impacto social y moral, ayudando al sector de cuidados de mascotas a crecer.</p> <p>TB2: Concluimos que la solución tecnológica propuesta para solucionar la problemática tiene un impacto social y moral; además que ayudara al negocio de las mascotas y todos sus involucrados.</p> <p>TP: Concluimos que la solución tecnológica propuesta tiene un impacto social y moral; además que ayudara al negocio de las mascotas y todos sus involucrados.</p> <p>TB3: Concluimos que la solución tecnológica propuesta tiene un impacto social y moral; además que ayudara al negocio de las mascotas y todos sus involucrados.</p> <p>TB4: Concluimos que la implementación del DDD Estratégico y Táctico tiene un impacto social y moral; además, ayudará al negocio de las mascotas y todos sus involucrados.</p> <p>TF: Concluimos que el desarrollo de una parte de la API de nuestro proyecto permitirá una guía para que el equipo de desarrollo se encargue de implementar totalmente. Con ello nuestra aplicación podrá hacerse realidad, y no se quedará como un prototipo.</p>

	<p>TP: Con la integración de los patrones, el diseño de software propuesto para la aplicación es más consistente y funcional</p> <p>TB3: Con la redacción de patrones empleados y la implementación de los diagramas de clases, el diseño de software propuesto para la aplicación es más consistente y funcional</p> <p>TB4: Con la redacción de los Bounded Context Canvas logramos mostrará la estrategia a seguir para la implementación de la aplicación.</p> <p>TF: Con la codificación de la API podemos dar una guía para que el equipo de desarrollo se encargue de implementar totalmente.</p> <p><u>Espinoza Quispe, Jennifer Mary</u></p> <p>TB1: Considero que los objetivos del proyecto son prescindibles porque es en base a ello que brindamos la solución deseada a la problemática</p> <p>TB2: Considero que el diagrama de clases brinda una visión organizada del diseño de la solución. Así mismo el Application Wireframes proporciona el ambiente ameno que el usuario tendrá al hacer uso de la app.</p> <p>TP: Incluir los patrones al diagrama de clases brinda al diseño de software consistencia y funcionalidad.</p> <p>TB3: Considero que la visualización de los diagramas de componentes es prescindible porque nos expresamos la lógica para la creación de nuestra aplicación</p> <p>TB4: Con la redacción del EventStorming y los diagramas del flujo del dominio logramos mostrará la estrategia a seguir para la implementación de la aplicación.</p> <p>TF: Con la redacción de los supuestos y principales riesgos nos damos cuenta de nuestras limitaciones para con este proyecto.</p>	
--	--	--

	<p><u>Seminario Garbín, Carlo Luca</u></p> <p>TB1: Teniendo como base los escenarios AS-IS / TO BE podemos conocer la perspectiva de nuestros usuarios frente a la problemática y lo que esperan de una solución tecnológica para ello.</p> <p>TB2: Realizar los diagramas de actividades permite conocer los pasos y procesos del algoritmo que va a seguir la aplicación.</p> <p>TP: Realizar las mejoras al diagrama de clases afianzara más el diseño del software para una eficaz solución.</p> <p>TB3: Con la redacción de patrones empleados y la implementación de los diagramas de clases, el diseño de software propuesto para la aplicación es más consistente y funcional</p> <p>TB4: Con la redacción de los Bounded Context Canvas logramos mostrará la estrategia a seguir para la implementación de la aplicación.</p> <p>TF: Con la codificación de la API podemos dar una guía para que el equipo de desarrollo se encargue de implementar totalmente.</p>	
	<p><u>Soriano Huapaya, Isabella María</u></p> <p>TB1: Al redactar el resumen ejecutivo del proyecto comprendemos como grupo mejor el impacto y solución que queremos brindar a la problemática.</p> <p>TB2: El diagrama de clases brinda una visión más organizada del diseño de la solución. Así mismo los diagramas de estado nos permite conocer las distintas etapas de una clase a lo largo de su ejecución.</p> <p>TP: Mejorar los requisitos funcionales permite especificar aún más las facilidades que la app brindara a mis usuarios.</p> <p>TB3: Considero que la visualización de los diagramas de contexto y contenedores son prescindibles porque nos</p>	

	<p>expresamos la lógica para la creación de nuestra aplicación</p> <p>TB4: Con la redacción del EventStorming y los diagramas del flujo del dominio logramos mostrará la estrategia a seguir para la implementación de la aplicación.</p> <p>TF: Con la redacción de los supuestos y principales riesgos nos damos cuenta de nuestras limitaciones para con este proyecto.</p>	
--	--	--

CONTENIDO

1.	Introducción	14
1.1.	Información del proyecto	14
1.1.1.	Startup Profile	14
1.1.2.	Descripción de la Startup	14
1.1.3.	Perfiles de integrantes del equipo	14
1.1.4.	Solution Profile	17
1.1.5.	Nombre del producto	17
1.2.	Objetivos del proyecto	18
1.3.	Propósito	18
1.4.	Alcance	18
1.5.	Siglas	19
1.6.	Fuera de alcance	19
1.7.	Restricciones	19
1.8.	Supuestos	20
1.9.	Principales riesgos	20
2.	REQUISITOS del producto	21
2.1.	Escenario AS-IS / TO-BE	21
2.2.	Requisitos funcionales	22
2.3.	Requisitos no funcionales	24
2.4.	Historias de usuario	24
2.5.	Product Backlog	34
2.6.	Sprint Backlog	36
3.	Diseño del producto	40
3.1.	Prototipos	40
3.2.	Diseño estratégico	45
3.2.1.	Event Storming	45
3.2.2.	Domain Message Flow Modelling	50
3.2.3.	Bounded Contexts	55
3.2.4.	Bounded Context Canvas	56
3.2.5.	Diagrama de Contexto	62
3.2.6.	Diagrama de Contenedores	64
3.3.	Diseño táctico	65
3.3.1.	Bounded Context Advertise	65
3.3.2.	Bounded Context Appointment	68
3.3.3.	Bounded Context Benefits	71
3.3.4.	Bounded Context Complaints book	74
3.3.5.	Bounded Context Invitations	77

3.3.6.	Bounded Context Opinions	80
3.3.7.	Bounded Context Owner.....	83
3.3.8.	Bounded Context Pays	86
3.3.9.	Bounded Context Pet Sitters.....	89
3.3.10.	Bounded Context Pets	92
3.3.11.	Bounded Context Security	95
3.3.12.	Bounded Context Services.....	98
3.3.13.	Diagrama de actividades.....	101
3.3.14.	Patrones empleados.....	104
4.	Construcción del producto.....	115
4.1.	Landige Page	115
4.2.	API REST	118
4.2.1.	Construcción.....	118
4.2.2.	Documentación.....	124
4.3.	Pruebas unitarias.....	124
	Conclusiones	127
	Recomendaciones.....	128
	Referencias bibliográficas	130
	Anexos.....	132

1. INTRODUCCIÓN

1.1. Información del proyecto

1.1.1. Startup Profile

NOMBRE DEL GRUPO: PetSolution

Somos un grupo de jóvenes estudiantes universitarios que hemos decidido, a través de un procedimiento de selección de ideas, optar por la creación de un servicio enfocado en la comodidad de los dueños de mascotas en encontrar establecimientos o personal especializado para que puedan atender a sus queridos animales.

Nombre del Startup:

PetSolution

1.1.2. Descripción de la Startup

PetSolution, es una empresa emergente o Startup especializada en personas con mascotas, la cual consiste en poder encontrar distintos establecimientos que ofrecen servicios para los animales, tales como veterinarias, hoteles, tiendas de productos propios e incluso poder encontrar personal especializado de confianza que pueda cuidar a una mascota. También se cuenta con una Alianza estratégica con negocios los cuales deseen ser promocionados en nuestro servicio. Adicionalmente los usuarios podrán obtener roles tales como dueño de mascota, ser parte del teamcare e incluso ser algún invitado, asimismo podrán brindar reseñas de los servicios que han recibido de manera que se pueda retroalimentar errores que encuentren y brindar consejos que ansíen de su implementación.

1.1.3. Perfiles de integrantes del equipo

Alumno: Ayasta Leonardo, Valery Ruth

Código: U20211C231

Carrera: Ingeniería de Software



Tengo 18 años y curso el cuarto ciclo de la carrera de Ingeniería de Software en la UPC. Me gusta programar y leer. Me considero una persona solidaria y empática. En mis tiempos libres me gusta escuchar podcasts. Siento que puedo aportar al equipo con mis conocimientos en C++, Lean UX Process y HTML5. En un futuro quisiera especializarme en Informática Forense.

Alumno: Espejo Macuri, Paolo Andre

Código: U20211B786

Carrera: Ingeniería de Software



Estudiante de Ingeniería de Software.

Hábil con las TICs y una persona creativa para tener nuevas ideas.

Alumno: Espinoza Quispe, Jennifer Mary

Código: U202120911

Carrera: Ingeniería de Software



Soy Jennifer Espinoza, me encuentro en el 3er ciclo de la carrera de Ing. de Software. Escogí esta carrera porque siempre me ha atraído la tecnología, el crear soluciones con un computador me sigue pareciendo muy interesante. En lo personal, me agrada aprender cosas nuevas, como nuevos lenguajes de programación e idiomas. Me considero una persona creativa, responsable y paciente, habilidades que ayudarán en el presente proyecto.

Alumno: Seminario Garbín, Carlo Luca

Código: U20211A475

Carrera: Ingeniería de Software



Mi nombre es Carlo Luca Seminario Garbín tengo 19 años de edad y actualmente curso el cuarto ciclo de la carrera de ingeniería de software en la Universidad Peruana de Ciencias Aplicadas (UPC). Considero que soy una persona que es capaz de trabajar bajo presión, además de ser responsable y perseverante. En cuanto a cualidades para la realización del trabajo considero que soy bueno identificando problemáticas y buscando soluciones, a lo que le puedo sumar los conocimientos previos de los ciclos anteriores en cursos de programación y de empresas.

Alumno: Soriano Huapaya, Isabella María

Código: U20211C242

Carrera: Ingeniería de Software



Mi experiencia con los negocios la tengo gracias al curso de Organización y Dirección de empresas cursado en mi segundo ciclo de carrera. Ello me sirvió para saber identificar un problema y solucionarlo mediante la creación de un proyecto de emprendimiento y, en esta oportunidad, una aplicación que se desempeñará como servicio. Las habilidades que poseo y pondré a disposición en el equipo son: capacidad de retención y análisis de información, liderazgo y organización.

1.1.4. Solution Profile

En Perú se registró que poco más del 60% de hogares limeños tiene en casa una mascota, por ello es imprescindible que les ofrezcan a estos animales los cuidados necesarios para una vida adecuada (La República, 2019). En ese sentido, encontrar distintos establecimientos y/o personal experimentado que puedan satisfacer estos cuidados a las mascotas según la comodidad, disponibilidad y conveniencia de los dueños; se necesita una manera para que el conocimiento de la existencia de estos servicios puedan llegar rápidamente a las personas, con lo cual proponemos a "PetCare", la cual servirá como un intermediario para generar contacto entre los dueños de mascotas y aquellos negocios y/o personal que ofrecen sus servicios. Para hacer uso de la app, los usuarios deberán:

- Crearse un perfil
- Podrá elegir entre un apartado de opciones de negocios y/o personal que ofrecen servicios y productos para mascotas
- La app se encarga de intermediarios para que el usuario pueda llegar a ellos y contratarlos.

Téngase en cuenta que para el funcionamiento de la app se tendrá que realizar alianzas con negocios, servicio y personal experimentado con relación al cuidado de mascotas. Es así que los ingresos de la empresa serán percibidos por una cuota mensual o anual por parte de negocios y personal suscrito a la misma. Sin embargo, dicha suscripción aún está en proceso, ya que se necesita establecer los ingresos y gastos que se generarían al salir al mercado como empresa.

1.1.5. Nombre del producto

Nombre de la app y logotipo



1.2. Objetivos del proyecto

"PetCare" busca brindar una nueva forma segura, sencilla y rápida de atender a tu mascota en diversos aspectos de su vida diaria. También, busca minimizar el número de animales sin hogar. Para lograr ello, nuestros objetivos son los siguientes:

- Ser los intermediarios más confiables para búsqueda de un cuidador de animales a nivel nacional.
- Tener una de mejores tasas de efectividad al publicitar la adopción responsable y supervisada de animales sin hogar.
- Realizar convenios con centros veterinarios para poder llevar a nuestros usuarios ofertas exclusivas y más bajas del mercado.

1.3. Propósito

Nuestra empresa, a través de su aplicación "PetCare" ofrecerá a sus usuarios una forma más rápida y cómoda de buscar servicios y adquirir productos para sus mascotas. Con ello, se busca solucionar los problemas que tienen los dueños de mascotas al momento de atenderlas o cuidarlas.

Esta aplicación busca reducir el tiempo que se toman las personas para encontrar centros de atención y cuidado animal. A su vez, ofrece tiendas en donde comprar artículos necesarios para el cuidado de sus mascotas.

También, ofrece beneficios como, por ejemplo, la obtención de puntos por cada vez que un usuario compra un producto o solicita un servicio, con los cuales podrá canjear descuentos en los artículos de las tiendas asociadas.

Adicionalmente, cuenta con una interfaz amigable y sencilla para facilitar aún más la tarea de criar a una mascota.

1.4. Alcance

La aplicación busca convertirse en el principal intermediario a nivel local entre los dueños de mascotas y los negocios que brindan servicios especializados en los próximos tres años. Según la encuestadora Ipsos, en el 2016, el 55% de los hogares limeños tenía alguna mascota,

de los cuales, se registra que el 41% les compra juguetes (Ipsos, 2016). Es decir, en nuestra región existe un público que desea los servicios especializados en mascotas y requieren un intermediario para facilitar su ubicación.

Deseamos entregar una app donde los usuarios se sientan cómodos y puedan navegar con tranquilidad, concentrándose en encontrar lo que necesitan para su mascota.

Para ejecutar este proyecto, este se ha dividido en 4 entregables a lo largo del presente ciclo, los cuales tendrán diferentes avances hasta poder llegar al producto final, el cual se centrará en el diseño Back-end, donde tendremos la Arquitectura y Diseño detallado de la solución, así como un adelanto del Front-end.

1.5. Siglas

- DDD: Domain Driven Design
- ORM: Object Relational Mapping
- API: interfaz de programación de aplicaciones
- CQRS: Command and Query Responsibility Segregation
- SMART: específicos, mensurables, alcanzables, relevantes y temporales

1.6. Fuera de alcance

Nosotros como startup fuimos contratados para diseñar de manera estratégica y táctica el 100% del sistema de la aplicación “PetCare” siguiendo el enfoque DDD, sin embargo, solo nos solicitaron implementar el 10% de la codificación que sirva de guía para el equipo de desarrollo que se va a encargar de implementar la API completa. Por lo tanto, está fuera de nuestro alcance el completar el 90% restante de la codificación del sistema.

1.7. Restricciones

En cuanto a las restricciones técnicas empleadas para la codificación que desarrollaremos del sistema, tenemos que:

- El enfoque de desarrollo empleado para el sistema ha de ser Domain Driven Design (DDD), de tal manera que sea usado también para la programación del API.

- La plataforma de desarrolla emplear es el server NodeJS.
- Se utilizará también el framework de NestJS, en conjunto con ello utilizaremos TypeORM
- El Lenguaje de programación empleado es TypeScript
- Para trabajar nuestra base de datos emplearemos MySQL 8
- Por último, el sistema deberá poder ejecutarse en el sistema operativo de Windows

1.8. Supuestos

Para el presente trabajo partimos con estos principales supuestos:

Uso de APIs externas como Ciber protector, Google Maps, PayPal, Worldpay y Stripe de fácil integración.

Nosotros partimos del supuesto de un correcto manejo y uso de dichas APIs. También, considerando que todos los sistemas externos poseen Rest API.

Conocimiento completo del lenguaje de programación TypeScript y total manejo de MySQL8.

1.9. Principales riesgos

Nuestros principales riesgos para nosotros como una nueva startup son:

Falta de experiencia aplicando Domain Driven Design, conocimiento básico del framework de NestJs para construir una aplicación web. Poca familiaridad con MySQL8, TypeORM y la respectiva migración de base de datos.

Por último, falta experiencia usando el patrón Command and Query Responsibility Segregation (CQRS), la cual es una de nuestras restricciones para el desarrollo de nuestra API.

2. REQUISITOS DEL PRODUCTO

2.1. Escenario AS-IS / TO-BE

[Ver enlace 1](#)

		AS-IS						
Phases		Adopción de mascotas	Compra de comida y productos relacionados al cuidado de la mascota	Servicio de veterinario	Encargar el cuidado de sus mascotas			
Doing	Buscar centros de adopción	Buscar conocidos cuyas mascotas hayan tenido crías y las den en adopción	Buscar e ir a petshops	Pedir los productos por delivery	Reservar cita con el veterinario local según disponibilidad	Buscar un familiar o conocido al cual encargarlo	Contratar un servicio para el cuidado de la mascota	
	Buscar un centro de adopción puede tomar mayor tiempo del deseado	No en todo momento hay algún conocido que de mascotas en adopción	Nunca lograré encontrar las orejas de mi mascota y se moverán a los demás	Tener que buscar e ir a tiendas cercanas me quita tiempo que podría invertir en otras cosas	Pedir los productos por delivery tiene un costo mayor	Alguna vez puedo necesitar otro veterinario por disponibilidad o emergencia	Ojalá que una persona de confianza pueda cuidar a mi mascota mientras yo no pueda	
	Perdida de motivación para adoptar una mascota			Estrés, porque se tiene que balancear el gasto económico y el tiempo invertido en la compra	Mosquitos e intranquilidad debida a que tiene que buscar un veterinario cercano y de confianza		Inconveniente, porque no podré encargar a alguien el cuidado de su mascota	
Thinking								
Feeling								

TO-BE								
Phases		Adopción de mascotas	Compra de comida y productos relacionados al cuidado de la mascota	Servicio de veterinario	Encargar el cuidado de sus mascotas			
Doing	Los usuarios publican imágenes e información de las mascotas que dan en adopción	Se puede conectar entre usuarios para coordinar la adopción bajo la supervisión del PetCare	Se busca en centros de adopción aliados	Buscar tiendas cercanas a tu ubicación desde PetCare	Buscar desde PetCare por ofertas exclusivas	Contactar para hacer delivery	Donde PetCare selecciona a un veterinario cercano a tu ubicación y realiza una reserva según disponibilidad	Contactar con personas verificadas por PetCare para dejarles el cuidado de tu mascota
	Que bueno que pueda tener la información de la mascota y el dueño	Tengo mayor variedad de opciones para la adopción	Puedo ahorrar al buscar ofertas	Tengo mayor comodidad y facilidad para buscar o pedir en tiendas cercanas	Tengo mayor cantidad de opciones para reservar una cita			Si me voy de viaje o no puedo cuidar de mi mascota puedo contactar con algunas de ellas inmediata
	Seguridad, puesto que cuentan con la información del perfil del dueño y mascota	Motivado a adoptar puesto que hay mayores opciones y es más sencillo	Alivio económico al poder encontrar ofertas	Tranquilo porque no tiene problemas para conseguir los productos que necesita	Relajado y seguro, porque no tiene complicaciones para reservar una cita			Agradecida de poder dejar encargados a sus mascotas
Thinking								
Feeling								

2.2. Requisitos funcionales

CÓDIGO	REQUISITO
GRUPO FUNCIONAL: GESTIÓN DE LA CUENTA	
RFU-001	LA APLICACIÓN TENDRÁ LA OPCIÓN DE CREAR UNA CUENTA REGISTRANDO TAMBIÉN A SU MASCOTA.
RFU-002	LA APLICACIÓN BRINDARÁ LA OPCIÓN DE QUE EL USUARIO PUEDA ACTUALIZAR SU PERFIL
GRUPO FUNCIONAL: SERVICIO	
RFU-003	LA APLICACIÓN BRINDARÁ UN APARTADO DONDE SE ENCONTRARÁ UNA LISTA DE TODOS LOS USUARIOS QUE OFRECEN SUS SERVICIOS EN LA APLICACIÓN.
RFU-004	AL CONTRATAR UN SERVICIO DE CUIDADO, LA APLICACIÓN BRINDA LA UBICACIÓN DE LA MASCOTA MEDIANTE EL USO DE GPS.
RFU-005	AL FINALIZAR EL SERVICIO EL USUARIO PODRÁ CALIFICAR EL SERVICIO Y HACER UN COMENTARIO AL RESPECTO.
RFU-006	EL USUARIO PODRÁ CONTRATAR A UN TRABAJADOR POR MEDIO DE LA APP PARA QUE CUIDE Y/O SUPERVISE A SU MASCOTA.
RFU-007	LA APLICACIÓN BRINDA AL USUARIO Y AL TRABAJADOR, AL CONCRETAR EL CONTRATO, LOS MEDIOS DE CONTACTO PERTINENTES COMO EL NÚMERO DE CELULAR Y REDES SOCIALES DE AMBAS PARTES.
RFU-008	LA APP BRINDA UN CONTRATO EN DONDE EL USUARIO Y LOS TRABAJADORES PUEDEN ESTABLECER LAS CONDICIONES DEL MISMO.
RFU-009	LA APLICACIÓN NOTIFICA AL USUARIO SI EL TRABAJADOR CON QUIEN REALIZÓ EL CONTRATO CANCELA LA CITA.

RFU-010	La aplicación brindará a todos los usuarios el acceso a un libro de reclamaciones
RFU-011	LA APLICACIÓN TENDRÁ UN APARTADO PARA UNA RED SOCIAL DONDE LOS USUARIOS DE LA COMUNIDAD PODRÁN INTERACTUAR ENTRE ELLOS.
RFU-012	LA APLICACIÓN BRINDARÁ UN APARTADO DONDE SE ENCONTRARÁ UNA LISTA DE NEGOCIOS ACTIVOS Y AFILIADOS A LA APLICACIÓN.
RFU-013	EL USUARIO AL REALIZAR ACTIVIDADES PROMOVIDAS POR PETCARE RECIBIRÁ POR ELLO PUNTOS_BENEFICIO.
RFU-014	EL USUARIO PODRÁ CANJEAR LOS PUNTOS BENEFICIO POR PRODUCTOS TANGIBLES OFRECIDOS POR LA MISMA APLICACIÓN.

2.3. Requisitos no funcionales

CÓDIGO	REQUERIMIENTO	TIPO
RNF-001	LA APLICACIÓN DEBE DE SER CAPAZ DE SOPORTAR A UNA CANTIDAD DE 100000 USUARIOS A LA VEZ.	EFICIENCIA
RNF-002	LA BASE DE DATOS DEBE DE PODER SER ACTUALIZADA AL INSTANTE LUEGO DE UN NUEVO REGISTRO, CON UN TIEMPO LÍMITE DE 1 MINUTO.	EFICIENCIA
RNF-003	LA BASE DE DATOS DEBE DE REALIZAR UN RESPALDO DE LOS DATOS CADA 24 O 48 HORAS.	SEGURIDAD
RNF-004	TODA LA INFORMACIÓN BRINDADA POR LOS USUARIOS NO SERÁ EXPUESTA Y SOLO SE USARÁ DURANTE LOS PROCESOS EN LA APLICACIÓN.	SEGURIDAD
RNF-005	LA APLICACIÓN DEBE DE TENER UNA OPCIÓN DE AYUDA EN LÍNEA.	USABILIDAD
RNF-006	LA INTERFAZ GRÁFICA DE LA APLICACIÓN DEBE DE ESTAR BIEN DISEÑADA.	USABILIDAD

2.4. Historias de usuario

Historia de Usuario			
Número:	HU01	Usuario:	Cliente (Sin mascota)
Nombre HU:	Contactar una persona para adoptar una mascota		
Prioridad en el negocio:	Bajo	Riesgo en desarrollo:	Medio
Horas estimadas:	5h	Iteración asignada:	Sprint 4
Programador responsable:	Paolo Espejo		
Descripción:	Como persona interesada en una mascota quiero poder contactar con dueños de mascotas que den crías en adopción para adoptar una mascota.		

Criterios de aceptación:	<p>Escenario 1: Usuario busca adoptar una mascota de otro usuario</p> <p>Dado que el sistema muestra un apartado para adoptar mascotas Cuando el usuario seleccione la opción adopta una mascota Y el usuario escoja una mascota Entonces el usuario llena un formulario con sus datos personales Y se envía una solicitud de adopción al dueño de la mascota Y el dueño de la mascota evalúa la solicitud Y el dueño de la mascota se pone en contacto con el usuario para llevar a cabo la adopción</p>
--------------------------	--

Historia de Usuario			
Número:	HU02	Usuario:	Cliente (Sin mascota)
Nombre HU:	Contactar a un centro de adopción		
Prioridad en el negocio:	Bajo	Riesgo en desarrollo:	Medio
Horas estimadas:	5h	Iteración asignada:	Sprint 4
Programador responsable:	Valery Ayasta		
Descripción:	Como persona interesada en una mascota quiero poder contactar con centros de adopción para adoptar una mascota.		
Criterios de aceptación:	<p>Escenario 1: Usuario busca adoptar una mascota de un centro de adopción</p> <p>Dado que el sistema muestra un apartado para adoptar mascotas Cuando el usuario seleccione la opción adopta una mascota Y el usuario escoga una mascota Entonces el usuario llena un formulario con sus datos personales Y se envía una solicitud de adopción al centro de adopción Y el centro de adopción evalúa su solicitud Y el centro de adopción le envía la respuesta a la bandeja de entrada del usuario</p>		

Historia de Usuario			
Número:	HU03	Usuario:	Cliente (con mascota)
Nombre HU:	Dar en adopción una mascota		
Prioridad en el negocio:	Bajo	Riesgo en desarrollo:	Bajo
Horas estimadas:	3h	Iteración asignada:	Sprint 4
Programador responsable:	Jennifer Espino		

Descripción:	Como dueño de una mascota quiero contactar con personas que busquen adoptar una para ofrecer la adopción de las crías de mi mascota.
Criterios de aceptación:	<p>Escenario 1: Usuario dueño de una mascota quiere dar en adopción las crías de su mascota</p> <p>Dado que un usuario tiene mascotas que quiere dar en adopción Cuando el usuario ingrese al apartado de dar en adopción Y complete el formulario con la información de la mascota en cuestión Entonces se registrará el estado de adopción en la plataforma Y otros usuarios podrán hacer solicitudes de adopción Y el dueño decide ponerse en contacto con algún solicitante para dar la mascota en adopción</p>

Historia de Usuario			
Número:	HU04	Usuario:	Cliente (con mascota)
Nombre HU:	Actualizar el estado de mi mascota		
Prioridad en el negocio:	Bajo	Riesgo en desarrollo:	Medio
Horas estimadas:	5h	Iteración asignada:	Sprint 4
Programador responsable:	Paolo Espejo		
Descripción:	<p>Como dueño de una mascota quiero actualizar el estado de mi mascota para dar a conocer si se ha perdido, se ha muerto, ha tenido crías o busco cruzarla.</p>		
Criterios de aceptación:	<p>Escenario 1: La mascota de un usuario se ha perdido</p> <p>Dado que la mascota registrada de un usuario se ha perdido Cuando el usuario vaya al apartado de perfil Y seleccione actualizar estado de mi mascota Entonces se pedirá una autenticación para habilitar la edición de los datos Y el usuario pondrá el estado de perdido a su mascota</p> <p>Escenario 2: La mascota de un usuario se ha muerto</p> <p>Dado que la mascota registrada de un usuario falleció Cuando el usuario vaya al apartado de perfil Y seleccione actualizar estado de mi mascota Entonces se pedirá una autenticación para habilitar la edición de los datos Y el usuario pondrá el estado de fallecido a su mascota Y el usuario tiene la opción de eliminar el registro de su mascota</p> <p>Escenario 3: La mascota de un usuario ha tenido crías</p> <p>Dado que la mascota registrada de un usuario tuvo crías Cuando el usuario vaya al apartado de perfil Y seleccione actualizar estado de mi mascota Entonces se pedirá una autenticación para habilitar la edición de los datos</p>		

	<p>Y el usuario pondrá el estado de crías recientes a su mascota Y la mascota ya no estará habilitada a tener crías por un tiempo</p> <p>Escenario 4: El dueño de una mascota busca cruzarla</p> <p>Dado que el dueño de una mascota registrada busca otra mascota para cruzarla Cuando el usuario vaya al apartado de perfil Y seleccione actualizar estado de mi mascota Entonces se pedirá una autenticación para habilitar la edición de los datos Y el usuario pondrá el estado de busca cruce Y el usuario podrá recibir solicitudes cruce</p>
--	---

Historia de Usuario			
Número:	HU05	Usuario:	Cliente (con mascota)
Nombre HU:	Buscar pet-shops cercanas		
Prioridad en el negocio:	Medio	Riesgo en desarrollo:	Alto
Horas estimadas:	7h	Iteración asignada:	Sprint 3
Programador responsable:	Paolo Espejo		
Descripción:	<p>Como dueño de una mascota quiero buscar pet-shops cercanas a mi ubicación para comprar productos relacionados al cuidado de mi mascota.</p>		
Criterios de aceptación:	<p>Escenario 1: Dueño de mascota busca pet-shop cercanos a su ubicación</p> <p>Dado que un usuario quiere comprar productos para el cuidado de su mascota Cuando el usuario ingrese al apartado de productos Y el usuario seleccione tiendas cercanas a mi ubicación Entonces se mostrará un mapa con tiendas de cuidado de mascotas cercanas a su ubicación Y podrá hacer filtros de búsqueda según el tipo de tienda de su preferencia</p>		

Historia de Usuario			
Número:	HU06	Usuario:	Cliente (con mascota)
Nombre HU:	Acceso a ofertas en productos de cuidado animal		
Prioridad en el negocio:	Medio	Riesgo en desarrollo:	Medio
Horas estimadas:	5h	Iteración asignada:	Sprint 3
Programador responsable:	Jennifer Espinoza		

Descripción:	Como dueño de una mascota quiero recibir ofertas en productos y servicios relacionados al cuidado de mi mascota para brindar un alivio económico.
Criterios de aceptación:	<p>Escenario 1: El dueño de una mascota quiere comprar productos o servicios para su mascota a precios más accesibles</p> <p>Dado que el dueño de una mascota la registra Cuando el cliente quiera comprar productos relacionados a su mascota Entonces el cliente es beneficiario de una serie de ofertas con tiendas aliadas</p>

Historia de Usuario			
Número:	HU07	Usuario:	Cliente (con mascota)
Nombre HU:	Acceder al servicio de veterinaria		
Prioridad en el negocio:	Alto	Riesgo en desarrollo:	Medio
Horas estimadas:	4h	Iteración asignada:	Sprint 2
Programador responsable:	Isabella Soriano		
Descripción:	Como dueño de una mascota quiero tener disponibilidad de un veterinario en todo momento para llevar a mi mascota ante cualquier eventualidad.		
Criterios de aceptación:	<p>Escenario 1: La mascota de un usuario necesita de un veterinario</p> <p>Dado que un usuario necesita un veterinario para su mascota Cuando ingrese al apartado de centros veterinarios Entonces tendrá el nombre, la disponibilidad, reseñas, ubicación del centro Y podrá hacer la reserva</p>		

Historia de Usuario			
Número:	HU08	Usuario:	Cliente (con mascota)
Nombre HU:	Recibir recompensas		
Prioridad en el negocio:	Bajo	Riesgo en desarrollo:	Medio
Horas estimadas:	5h	Iteración asignada:	Sprint 4
Programador responsable:	Isabella Soriano		
Descripción:	Como dueño de una mascota quiero obtener recompensas por realizar actividades promovidas por PetCare para estar motivado a participar en ellas.		

Criterios de aceptación:	<p>Escenario 1: PetCare incita a sus usuarios a participar en actividades beneficiarias para mascotas</p> <p>Dado que un usuario participó en una actividad promovida por PetCare Cuando el usuario suba evidencia de su participación mediante una publicación Entonces el usuario será beneficiario de PetPoints Y podrá canjearlo por descuentos u otros beneficios</p>
--------------------------	--

Historia de Usuario			
Número:	HU09	Usuario:	Cliente (con mascota)
Nombre HU:	Cuidado de mascotas		
Prioridad en el negocio:	Alto	Riesgo en desarrollo:	Medio
Horas estimadas:	5h	Iteración asignada:	Sprint 1
Programador responsable:	Valery Ayasta		
Descripción:	<p>Como dueño de una mascota quiero poder encargar a una persona el cuidado de mi mascota para que la supervise y le de el cuidado necesario cuando yo no pueda.</p>		
Criterios de aceptación:	<p>Escenario 1: El dueño de una mascota no tiene tiempo para darle cuidado a su mascota</p> <p>Dado que el usuario dueño de una mascota está ocupada Y no puede brindarle cuidado a su mascota Cuando ingrese al apartado de paseo de perros Y seleccione la fecha, hora, cuidador e ubicación Entonces se le asignará un cuidador Y recibirá un mensaje en su bandeja de entrada</p>		

Historia de Usuario			
Número:	HU10	Usuario:	Cliente (con mascota)
Nombre HU:	Contacto con el cuidador		
Prioridad en el negocio:	Alto	Riesgo en desarrollo:	Bajo
Horas estimadas:	3h	Iteración asignada:	Sprint 1
Programador responsable:	Carlo Seminario		
Descripción:	<p>Como dueño de una mascota quiero tener todos los medios de contacto al instante con el encargado del cuidado de mi mascota para contactarlo o ser contactado ante cualquier eventualidad.</p>		

Criterios de aceptación:	Escenario 1: La mascota de un usuario está siendo cuidada Dado que como usuario quiero conocer el estado de mi mascota Cuando ingreso al apartado de paseo de perros Y ingreso a paseo en curso Entonces puedo ver el estado del paseo Y ponerme en contacto con el cuidador a través de mensajes, llamada o videollamada
--------------------------	--

Historia de Usuario			
Número:	HU11	Usuario:	Cliente (con mascota)
Nombre HU:	Notificar el estado de la mascota		
Prioridad en el negocio:	Alto	Riesgo en desarrollo:	Bajo
Horas estimadas:	3h	Iteración asignada:	Sprint 1
Programador responsable:	Paolo Espejo		
Descripción:	Como dueño de una mascota quiero ser notificado si el encargado de cuidar a mi mascota cancela el servicio para contactar con otro cuidador.		
Criterios de aceptación:	Escenario 1: Un usuario contrata el cuidado de su perro pero le cancelan Dado que un usuario contrata el servicio de un cuidador Cuando el cuidador cancela el servicio Entonces el usuario recibe una notificación en la bandeja de entrada Y puede contratar a otro cuidador		

Historia de Usuario			
Número:	HU12	Usuario:	Cliente (con mascota)
Nombre HU:	Acceso para reclamos		
Prioridad en el negocio:	Alto	Riesgo en desarrollo:	Bajo
Horas estimadas:	3h	Iteración asignada:	Sprint 2
Programador responsable:	Isabella Soriano		
Descripción:	Como dueño de una mascota quiero tener un libro de reclamaciones para mostrar alguna discrepancia que tenga.		
Criterios de aceptación:	Escenario 1: Un usuario tiene alguna queja Dado que un usuario tiene una discrepancia Cuando ingrese al apartado centro de ayuda Entonces podrá presentar sus reclamos Y recibir ayuda con respecto al tema		

Historia de Usuario			
Número:	HU13	Usuario:	Cliente (Tienda de cuidado de mascotas)
Nombre HU:	Clientes para la tienda de productos de cuidado animal		
Prioridad en el negocio:	Medio	Riesgo en desarrollo:	Medio
Horas estimadas:	5h	Iteración asignada:	Sprint 3
Programador responsable:	Valery Ayasta		
Descripción:	<p>Como tienda de productos para el cuidado de mascotas quiero un intermediario que capte a mi público objetivo para que compren mis productos.</p>		
Criterios de aceptación:	<p>Escenario 1: Una tienda aliada quiere incrementar el numero de sus clientes Dado que una tienda aliada quiere incrementar sus ventas Cuando los usuarios dueños de una masctoa ingresen a buscar productos Entonces la tienda saldrá como opción según el perfil del usuario</p>		

Historia de Usuario			
Número:	HU14	Usuario:	Cliente (Tienda de cuidado de mascotas)
Nombre HU:	Calidad de productos en la tienda de productos para mascotas		
Prioridad en el negocio:	Medio	Riesgo en desarrollo:	Alto
Horas estimadas:	7h	Iteración asignada:	Sprint 3
Programador responsable:	Carlo Seminario		
Descripción:	<p>Como tienda de productos para el cuidado de mascotas quiero que mis clientes se sientan a gusto con la calidad y precio de mis productos para que se vuelvan clientes habituales.</p>		
Criterios de aceptación:	<p>Escenario 1: Una tienda quiere brindar un servicio de calidad Dado que una tienda busca tener clientes habituales Cuando un usuario dueño de una mascota quiera comparar algo nuevamente Entonces le aparecerá las tiendas donde a comprado previamente Y podrá tener descuentos especiales por cliente frecuente</p>		

Historia de Usuario			
Número:	HU15	Usuario:	Cliente (veterinaria)
Nombre HU:	Clientes para veterinaria		
Prioridad en el negocio:	Alto	Riesgo en desarrollo:	Alto
Horas estimadas:	7h	Iteración asignada:	Sprint 2
Programador responsable:	Paolo Espejo		
Descripción:	Como veterinaria quiero un intermediario que consiga clientes para poder obtener el mayor número de citas posibles por día.		
Criterios de aceptación:	<p>Escenario 1: Una veterinaria busca tener más citas</p> <p>Dado que una veterinaria no ha completado la disponibilidad de horarios del día</p> <p>Cuando el cliente ingrese a buscar veterinarias</p> <p>Entonces esa veterinaria tendrá mayor prioridad al estar más disponible</p>		

Historia de Usuario			
Número:	HU16	Usuario:	Cliente (cuidador)
Nombre HU:	Negociación		
Prioridad en el negocio:	Alto	Riesgo en desarrollo:	Medio
Horas estimadas:	4h	Iteración asignada:	Sprint 1
Programador responsable:	Jennifer Espinoza		
Descripción:	Como cuidador quiero poder negociar con el dueño de la mascota para llegar a un precio justo.		
Criterios de aceptación:	<p>Escenario 1: Se contacta un cuidador con un usuario</p> <p>Dado que un usuario contacta con un cuidador</p> <p>Cuando se realiza la reserva para el cuidado</p> <p>Entonces se llega a un acuerdo por el precio</p> <p>Y PetCare se lleva un 10% de comisión</p>		

Historia de Usuario			
Número:	HU17	Usuario:	Cliente (cuidador)
Nombre HU:	Previsión de problemas		
Prioridad en el negocio:	Alto	Riesgo en desarrollo:	Medio
Horas estimadas:	5h	Iteración asignada:	Sprint 1
Programador responsable:	Valery Ayasta		
Descripción:	Como cuidador quiero tener garantías por el servicio brindado para evitar eventualidades por cancelaciones o problemas con el pago		
Criterios de aceptación:	<p>Escenario 1: Un usuario realiza el pago por medio de la aplicación</p> <p>Dado que un usuario contrata a un cuidador por un servicio Y paga por medio de la app Y el cuidador reciba el 50% del pago Cuando pasen 48h de haber brindado el servicio Y verificar la conformidad de ambas partes Entonces el cuidador recibirá el 50% restante</p>		
	<p>Escenario 2: Un cuidador le cancelan el servicio 12h antes del horario acordado</p> <p>Dado que un usuario contrata a un cuidador por un servicio Cuando el usuario cancela el servicio con 12 o más horas de anticipación Entonces el usuario recibe un rembolso completo</p>		
	<p>Escenario 3: Un cuidador le cancelan el servicio con menos de 12h de aviso</p> <p>Dado que un usuario contrata a un cuidador por un servicio Cuando el usuario cancela el servicio con menos de 12 horas de anticipación Entonces el cuidador recibe un 6% del costo correspondiente al servicio</p>		

2.5. Product Backlog

#Orden	User Story ID	Título	Descripción	Story Points (1/2/3/5/8)
1	HU09	Cuidado de mascotas	Como dueño de una mascota quiero poder encargar a una persona el cuidado de mi mascota para que la supervise y le dé el cuidado necesario cuando yo no pueda.	5
2	HU10	Contacto con el cuidador	Como dueño de una mascota quiero tener todos los medios de contacto al instante con el encargado del cuidado de mi mascota para contactarlo o ser contactado ante cualquier eventualidad.	8
3	HU16	Negociación	Cómo cuidador quiero poder negociar con el dueño de la mascota para llegar a un precio justo	5
4	HU17	Previsión de problemas	Cómo cuidador quiero tener garantías por el servicio brindado para evitar eventualidades por cancelaciones o problemas con el pago	8
5	HU11	Notificar el estado de la mascota	Como dueño de una mascota quiero ser notificado si el encargado de cuidar a mi mascota cancela el servicio para contactar con otro cuidador.	5
6	HU07	Acceder al servicio de veterinaria	Como dueño de una mascota quiero tener disponibilidad de un veterinario en todo momento	3

			para llevar a mi mascota ante cualquier eventualidad.	
7	HU04	Actualizar el estado de mi mascota	Como dueño de una mascota quiero actualizar el estado de mi mascota para dar a conocer si se ha perdido, se ha muerto, ha tenido crías o busco cruzarla.	3
8	HU12	Acceso para reclamos	Como dueño de una mascota quiero tener un libro de reclamaciones para mostrar alguna disconformidad que tenga.	5
9	HU05	Buscar pet-shops cercanas	Como dueño de una mascota quiero buscar pet-shops cercanas a mi ubicación para comprar productos relacionados al cuidado de mi mascota.	8
10	HU06	Acceso a ofertas en productos de cuidado animal	Como dueño de una mascota quiero recibir ofertas en productos y servicios relacionados al cuidado de mi mascota para brindar un alivio económico.	5
11	HU08	Recibir recompensas	Como dueño de una mascota quiero obtener recompensas por realizar actividades promovidas por PetCare para estar motivado a participar en ellas.	5
12	HU03	Dar en adopción una mascota	Como dueño de una mascota quiero contactar con personas que busquen adoptar una para ofrecer la adopción de las crías de mi mascota.	3
13	HU01	Contactar una persona para adoptar una mascota	Como persona interesada en una mascota quiero poder contactar con dueños de	5

			mascotas que den crías en adopción para adoptar una mascota.	
14	HU02	Contactar a un centro de adopción	Como persona interesada en una mascota quiero poder contactar con centros de adopción para adoptar una mascota.	5
15	HU15	Clients para veterinaria	Como veterinaria quiero un intermediario que consiga clientes para poder obtener el mayor número de citas posibles por día.	8
16	HU13	Clients para la tienda de productos de cuidado animal	Como tienda de productos para el cuidado de mascotas quiero un intermediario que capte a mi público objetivo para que compren mis productos.	5
17	HU14	Calidad de productos en la tienda de productos para mascotas	Como tienda de productos para el cuidado de mascotas quiero que mis clientes se sientan a gusto con la calidad y precio de mis productos para que se vuelvan clientes habituales.	3

2.6. Sprint Backlog

Sprint 1		Sprint 1						
User Story		Work-Item / Task						
ID	Title	ID	Title	Description		Estimation (Hours)	Assigned To	Status
HU09		T01	Registro de dueños	Programar interfaz para la el		5	Valery Ayasta	To-Do

				registro de los dueños			
Cuidado de mascotas	T02	Registro de cuidadores	Programar la interfaz para el registro de cuidadores	5	Isabella Soriano	To-Do	
	T03	Contacto entre dueños y cuidadores	Programar contratación de un cuidador	5	Jennifer Espinoza	To-Do	
HU10	Contacto con el cuidador	T01	Perfil y datos del cuidador	Programar la interfaz para que el usuario observe la información de contacto del cuidador	3	Carlo Seminario	To-Do
HU11	Notificar el estado de la mascota	T01	Estado del cuidador	Establecer un atributo para que se muestre y se actualice la condición de un cuidador.	3	Paolo Espejo	To-Do
HU16	Negociación	T01	Acuerdo	Conectar al cuidador con el dueño para que negocien	4	Jennifer Espinoza	To-Do
HU17	Previsión de problemas	T01	Garantía del servicio	Establecer lineamientos	5	Valery Ayasta	To-Do

Sprint 2	Sprint 2							
User Story	Work-Item / Task							
ID	Title	ID	Title	Description	Estimation (Hours)	Assigned To	Status	
HU07	Acceder al servicio de veterinaria	T01	Registro de veterinarias	Programar el registro de veterinarias en la aplicación	4	Jennifer Espinoza	To-Do	

		T02	Acceso al servicio	Programar la reserva en una veterinaria	5	Isabella Soriano	To-Do
HU15	Clientes para veterinaria	T01	Recomendar veterinaria	Añadir el apartado de salud animal para recomendar la veterinaria	7	Paolo Espejo	To-Do
HU12	Acceso para reclamos	T01	Libro de quejas y sugerencias	Habilitar una opción para que el usuario ingrese sus comentarios	3	Isabella Soriano	To-Do

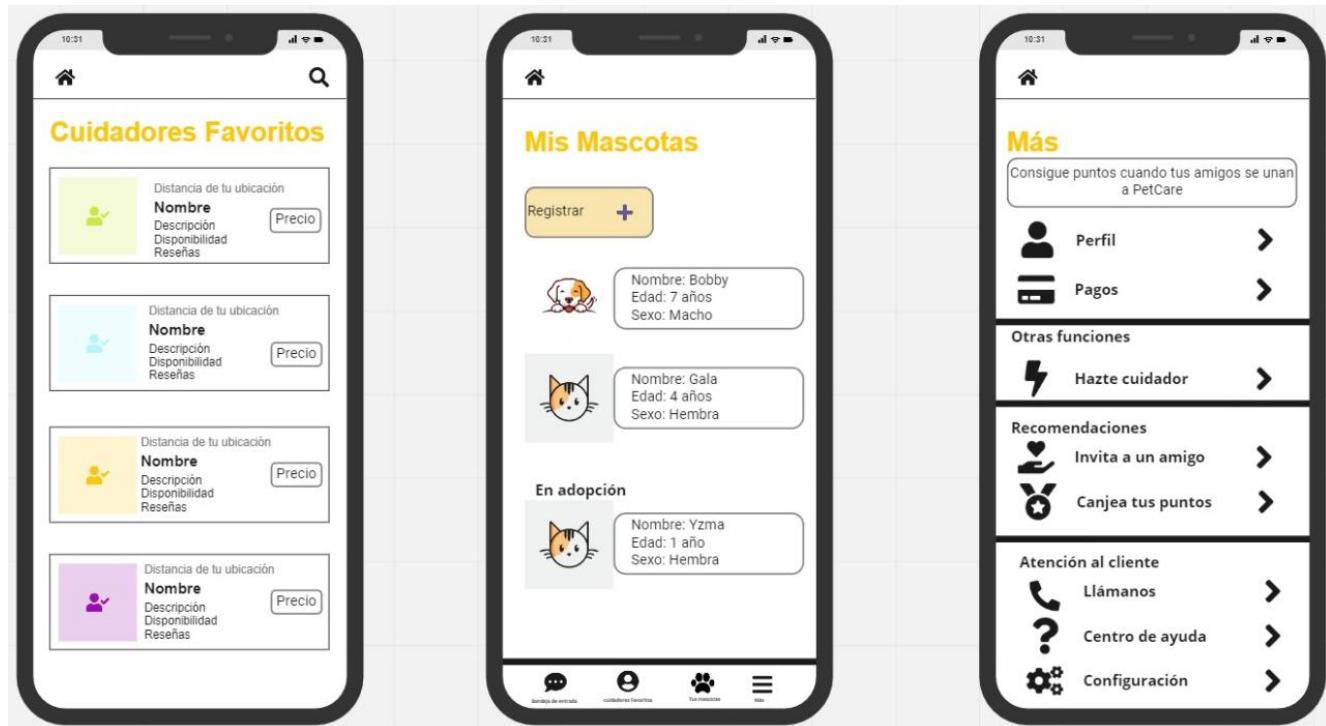
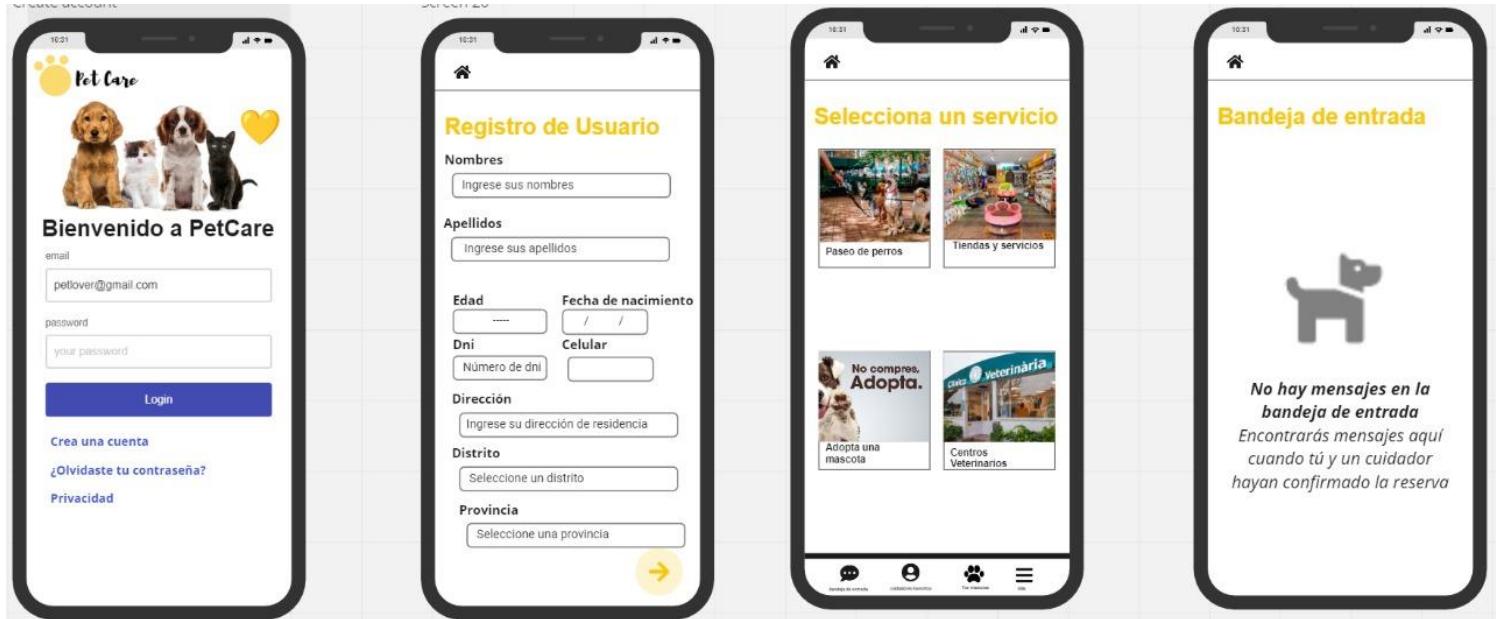
Sprint 3		Sprint 3						
User Story		Work-Item / Task						
ID	Title	ID	Title	Description		Estimation (Hours)	Assigned To	Status
HU05	Buscar pet-shops cercanas	T01	Localización de pet-shops	Programar el mapa que muestre las pet-shops más cercanas según la ubicación del dueño		7	Paolo Espejo	To-Do
HU06	Acceso a ofertas en productos de cuidado animal	T01	Obtención de ofertas	Programar la obtención de puntos para obtener ofertas	5	Jennifer Espinoza	To-Do	
		T02	Canje de ofertas	Programar el canje de los beneficios en las tiendas asociadas	5	Isabella Soriano	To-Do	
HU13	Clientes para la tienda de productos de cuidado animal	T01	Tienda virtual	Programar la tienda virtual de productos de cuidado animal	5	Valery Ayasta	To-Do	
HU14	Calidad de productos en la tienda de productos para mascotas	T01	Organizar productos	Programar la interfaz de la tienda para ordenar los productos y mostrar los más recomendados según reseñas	7	Carlo Seminario	To-Do	

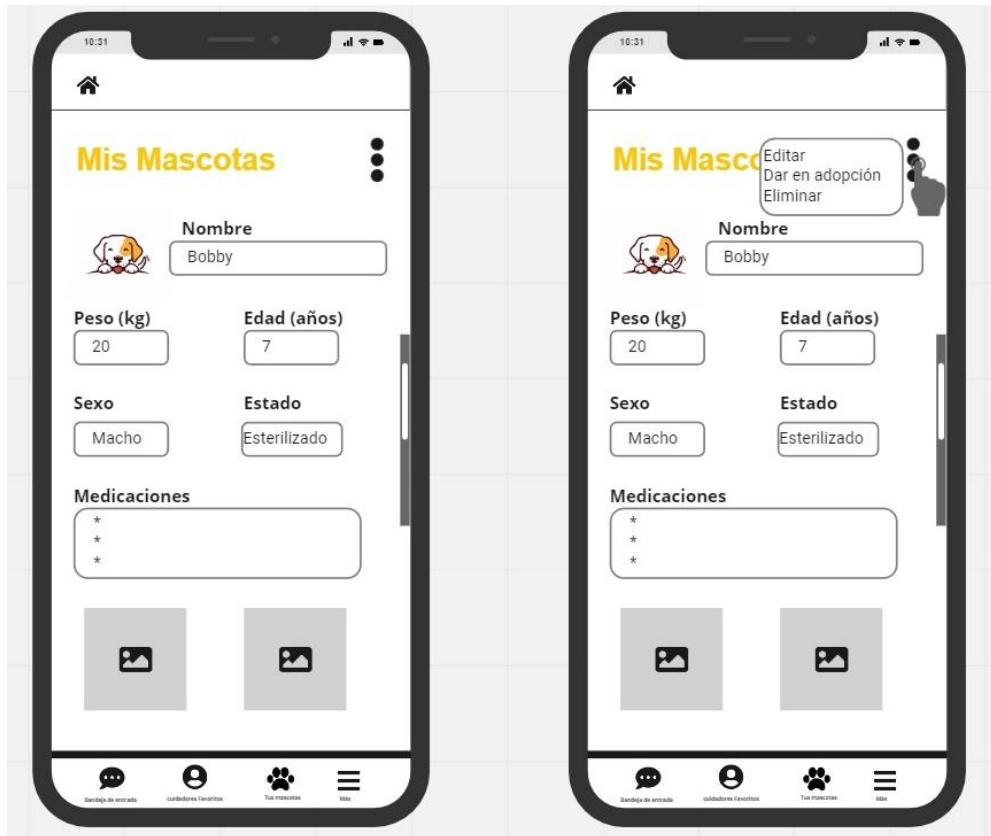
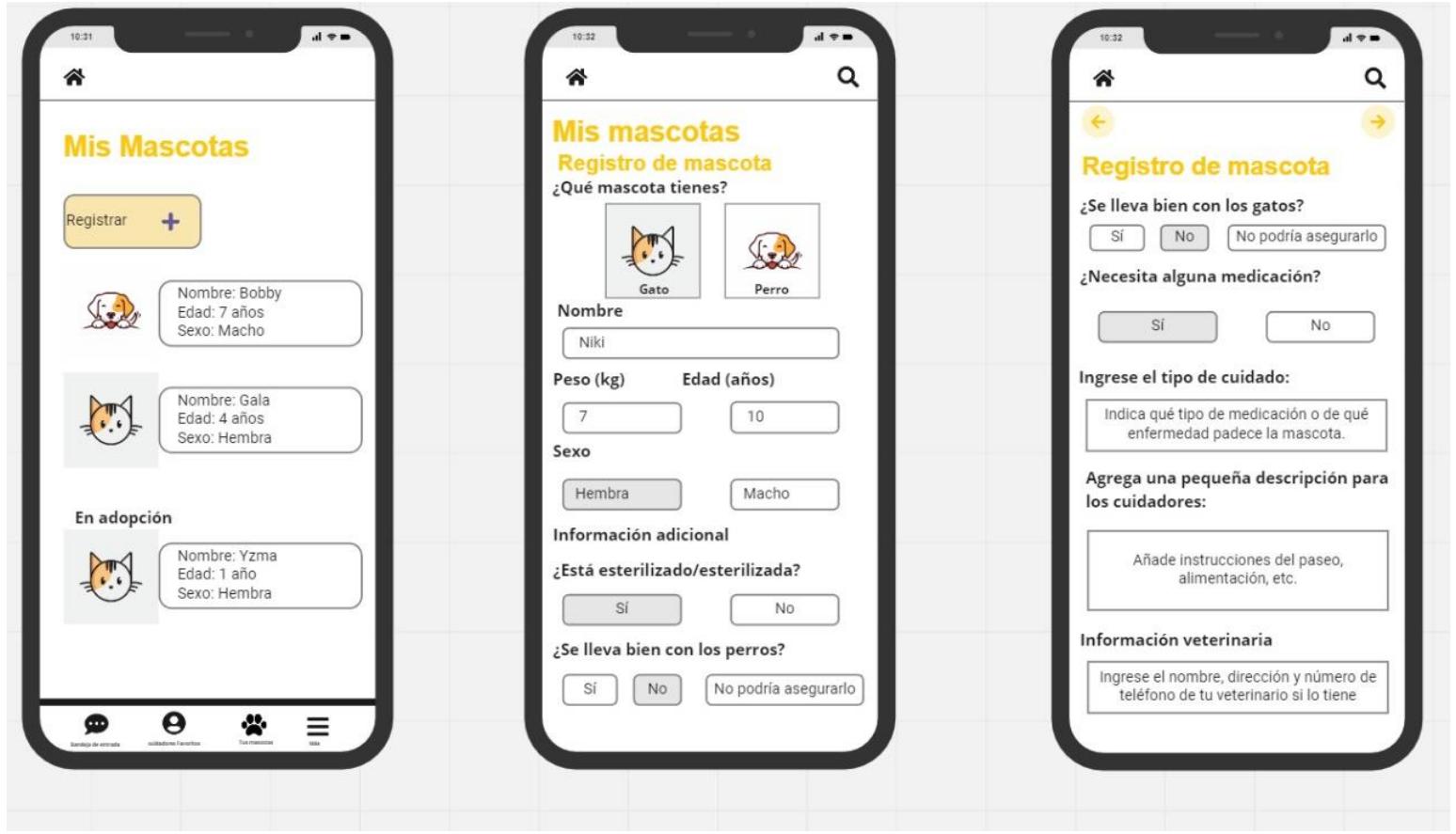
Sprint 4		Sprint 4						
User Story		Work-Item / Task						
ID	Title	ID	Title	Description	Estimation (Hours)	Assigned To	Status	
HU04	Actualizar el estado de mi mascota	T01	Perfil de la mascota	Programar la interfaz del perfil de la mascota	5	Paolo Espejo	To-Do	
HU08	Recibir recompensas	T01	Asignar recompensas	Añadir recompensas en la aplicación	5	Isabella Soriano	To-Do	
HU03	Dar en adopción una mascota	T01	Dar en adopción	Programar la interfaz de anuncios	3	Jennifer Espino	To-Do	
HU01	Contactar una persona para adoptar una mascota	T01	Contacto con adoptador	Programar la aplicación para contactar al adoptador	5	Paolo Espejo	To-Do	
HU02	Contactar a un centro de adopción	T01	Contacto centro	Programar la interfaz de contacto a un centro de adopción	5	Valery Ayasta	To-Do	

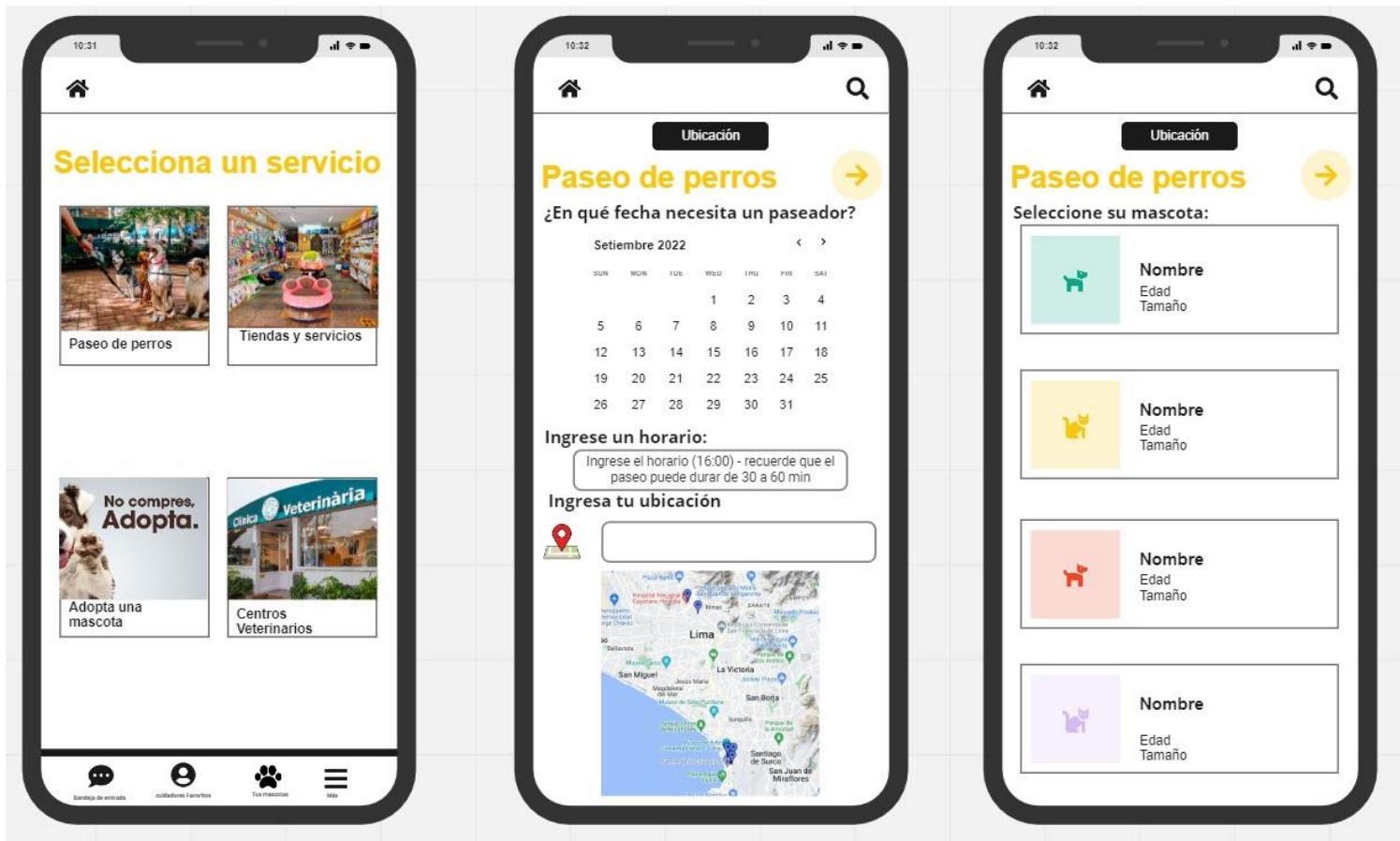
3. DISEÑO DEL PRODUCTO

3.1. Prototipos

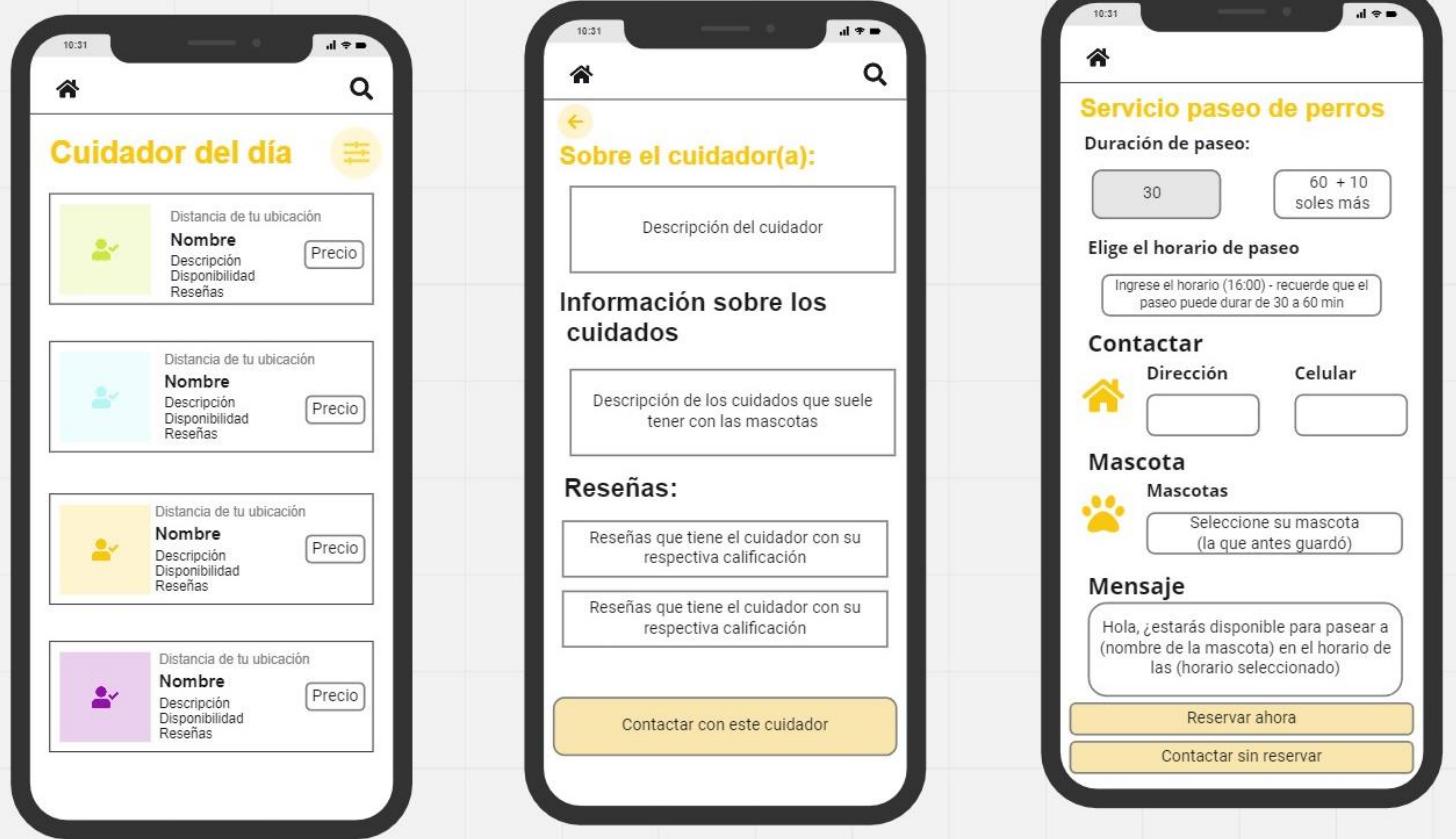
[Ver enlace 2](#)

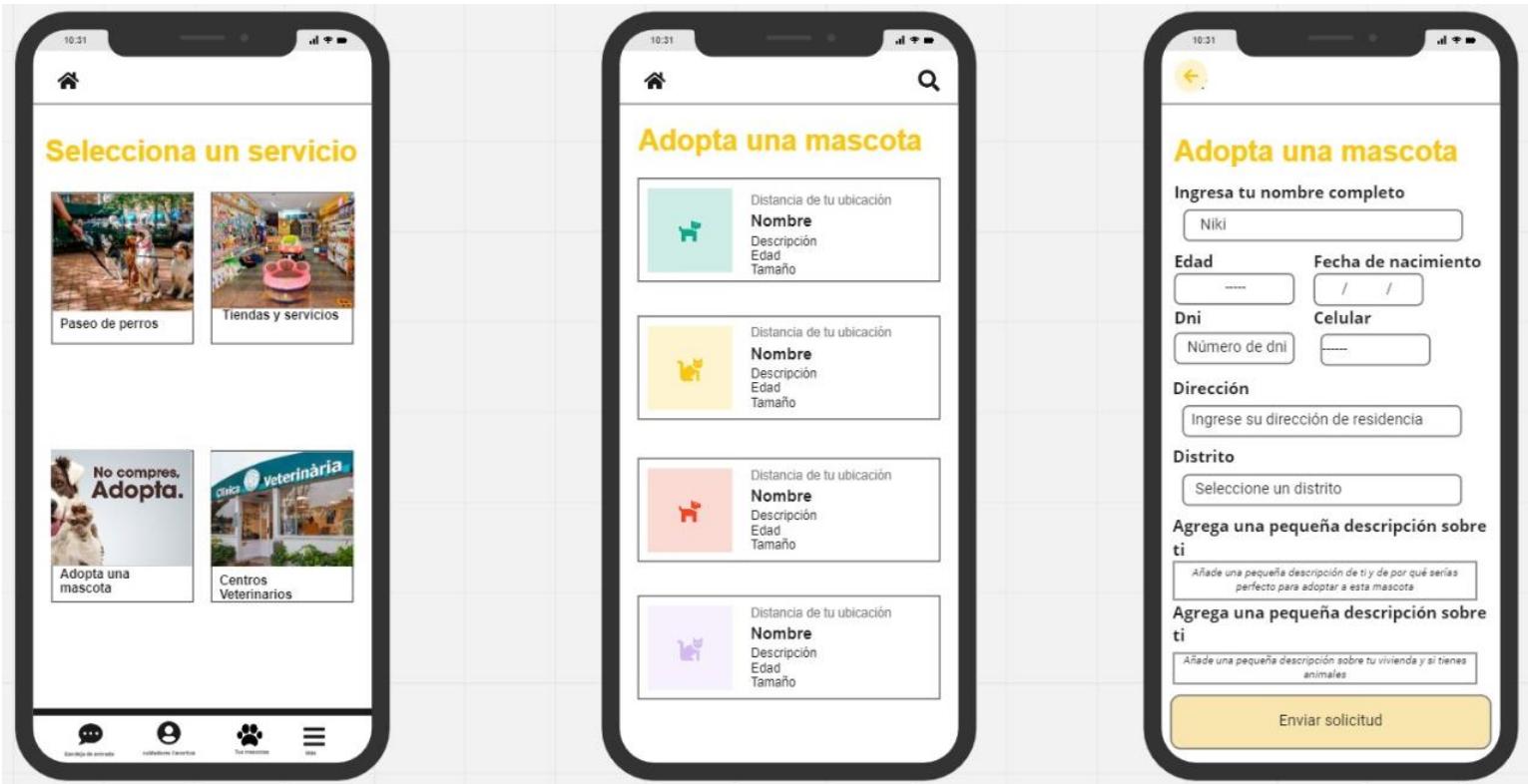
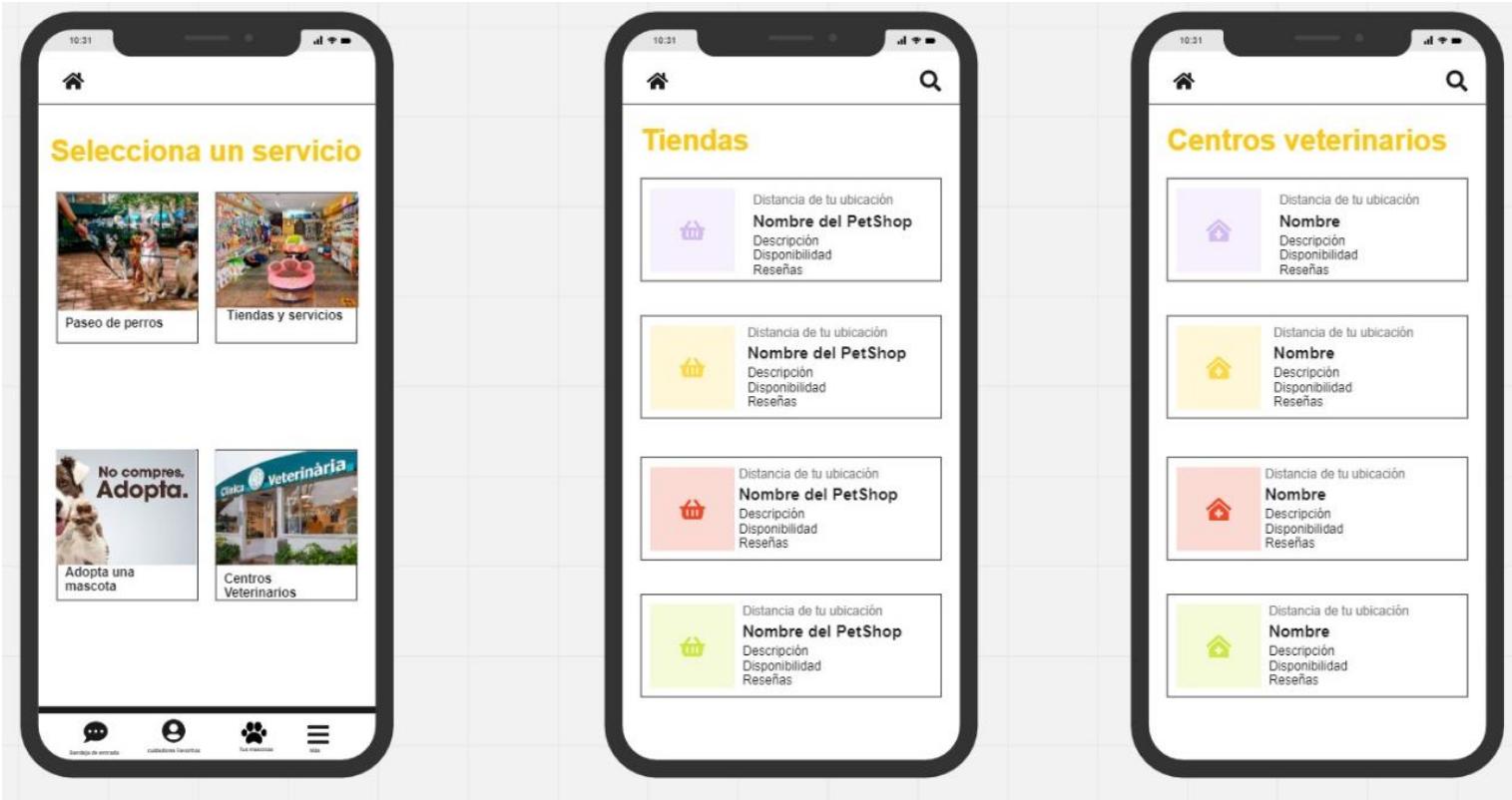


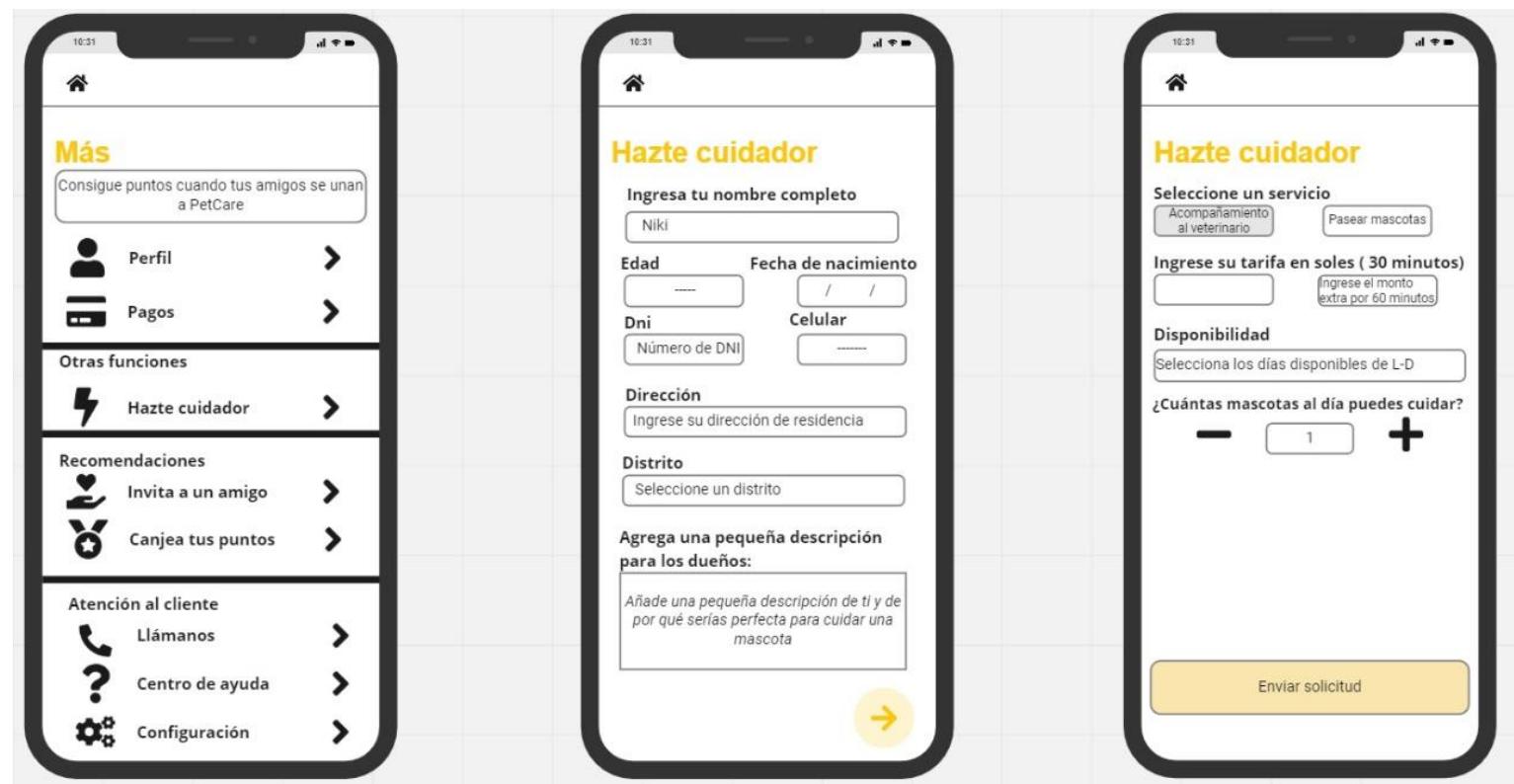
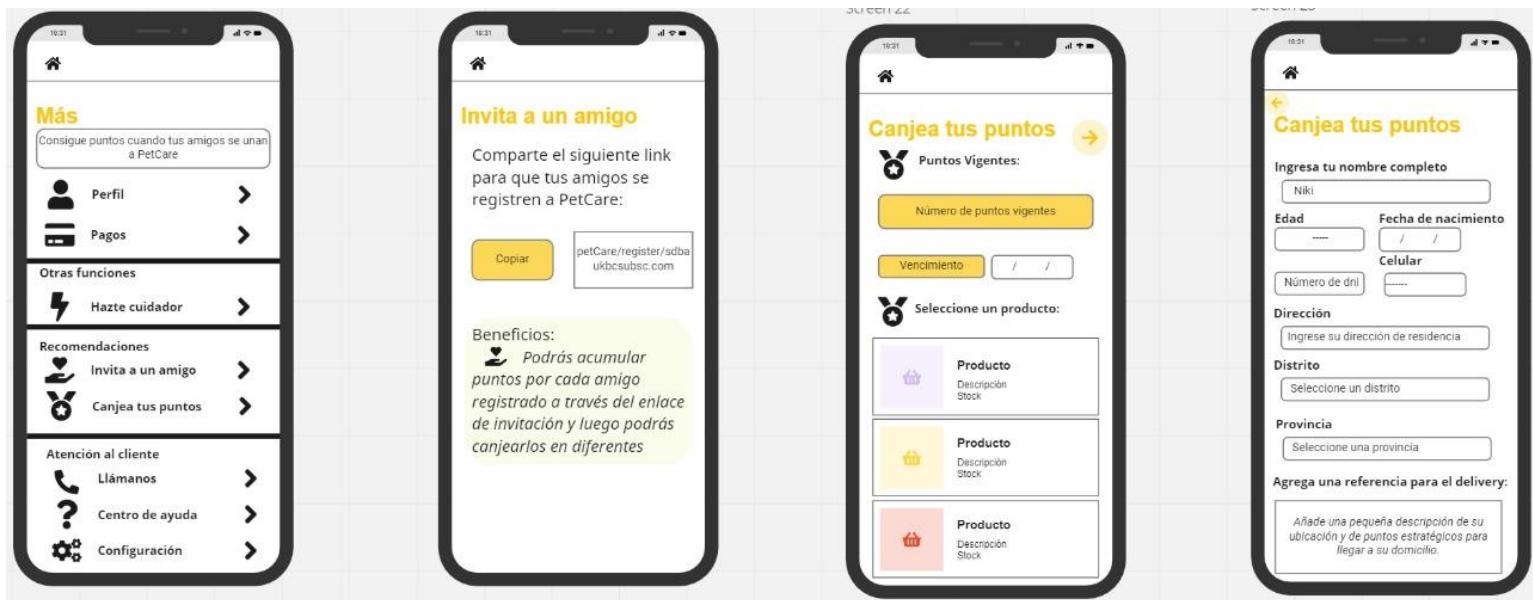


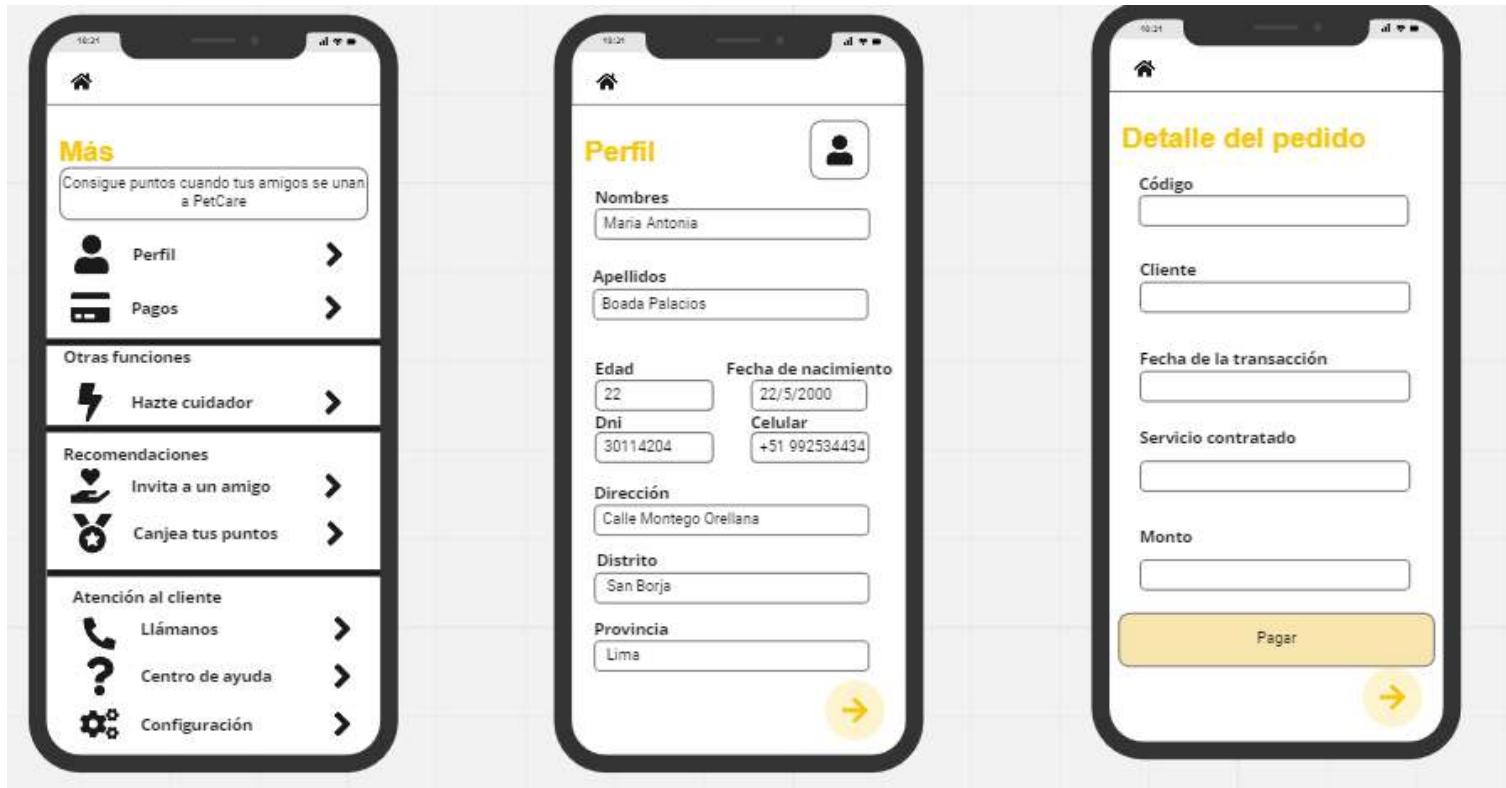


Screen 25





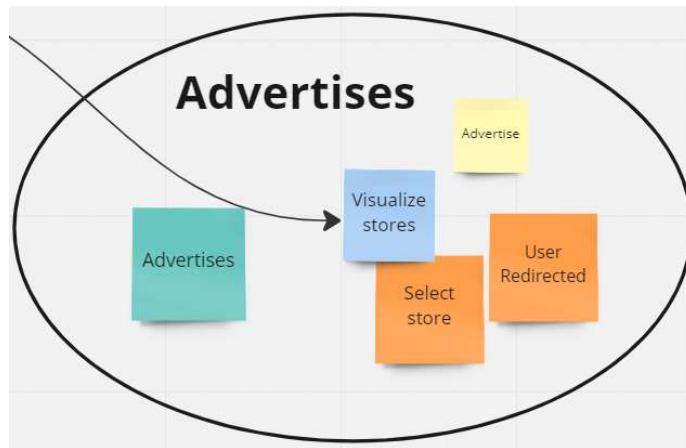


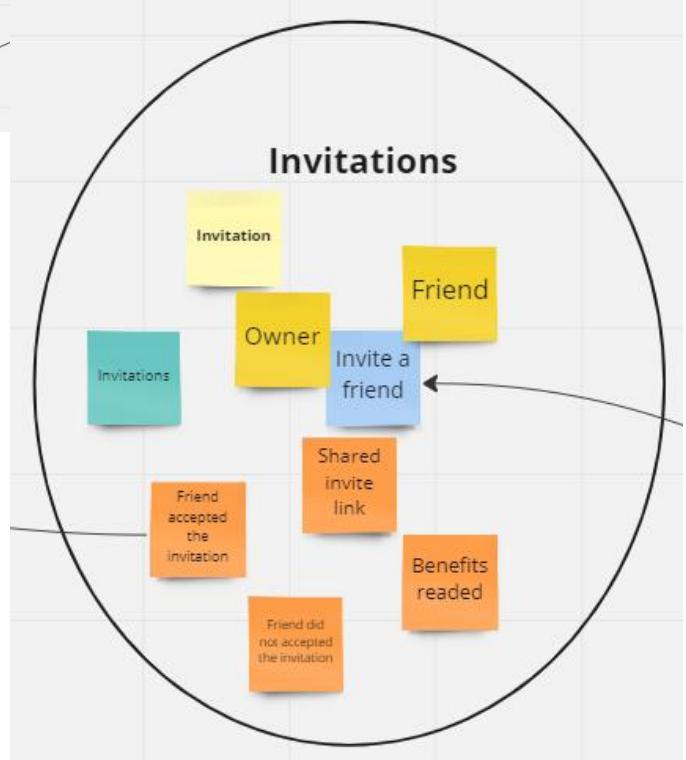
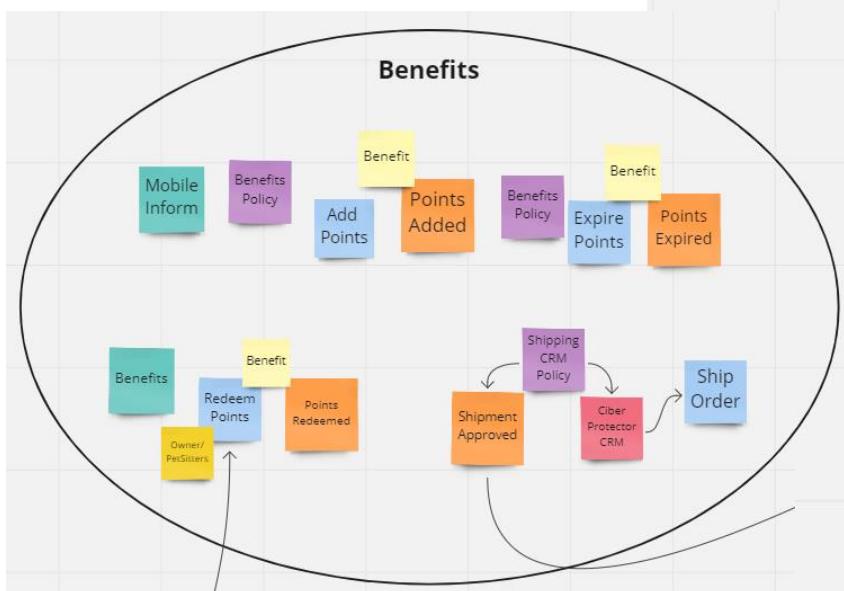
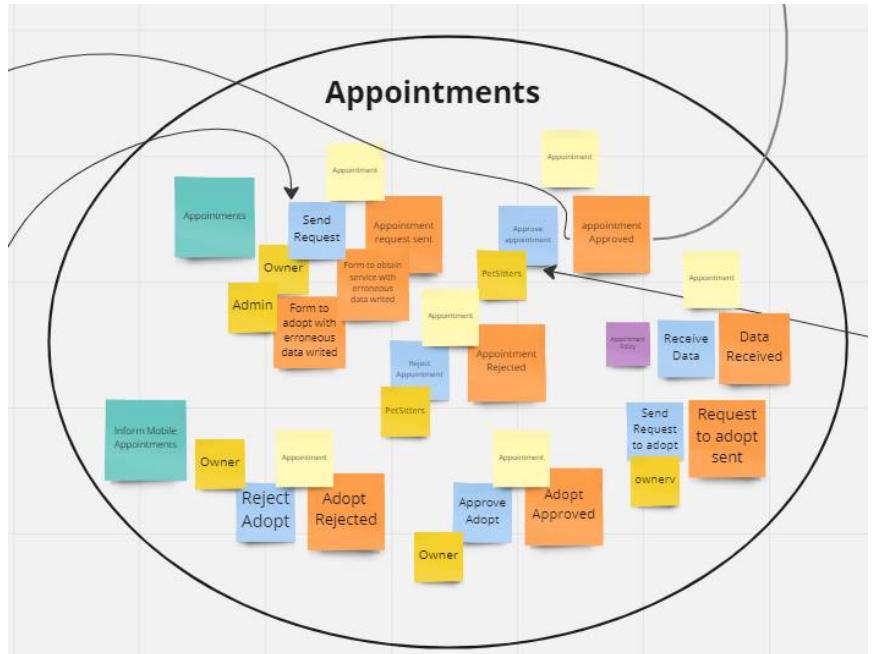


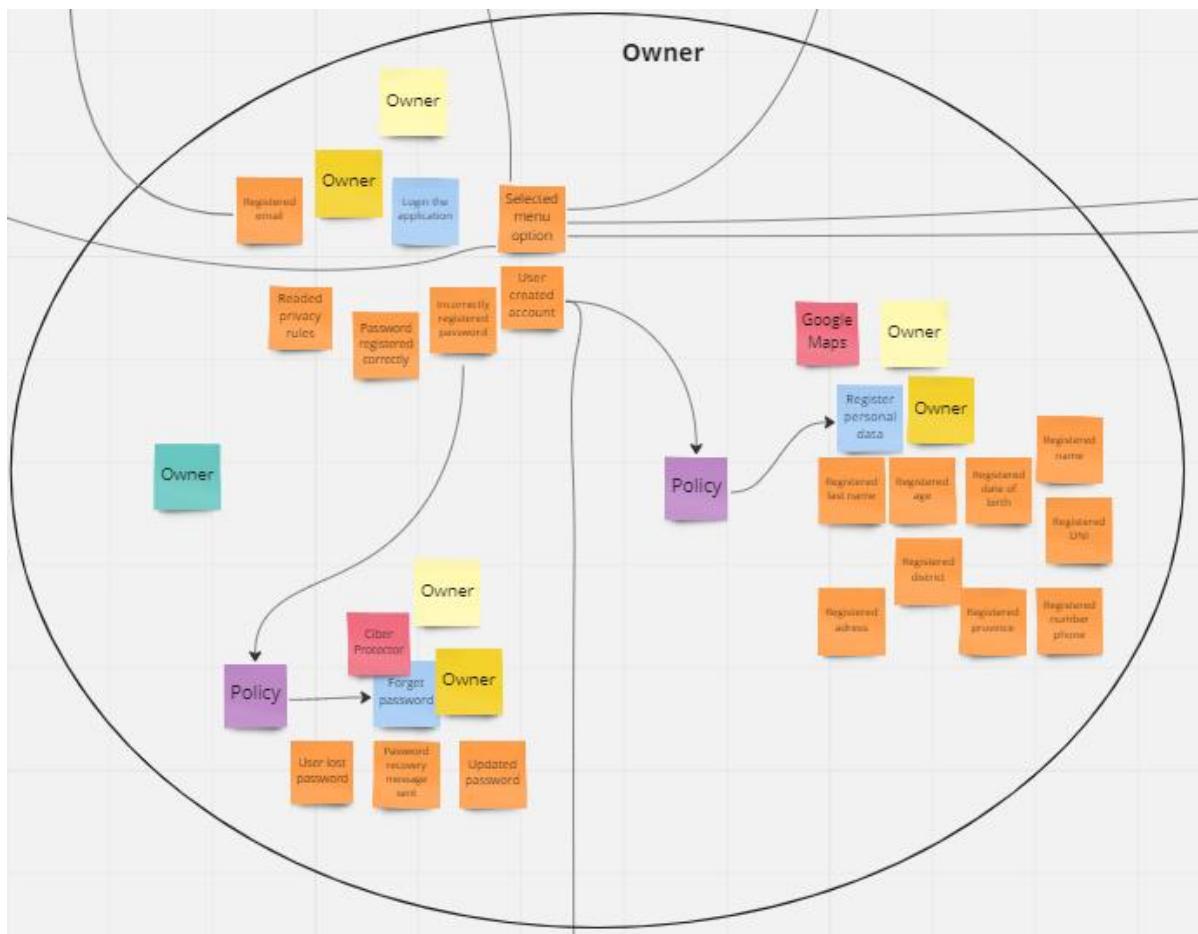
3.2. Diseño estratégico

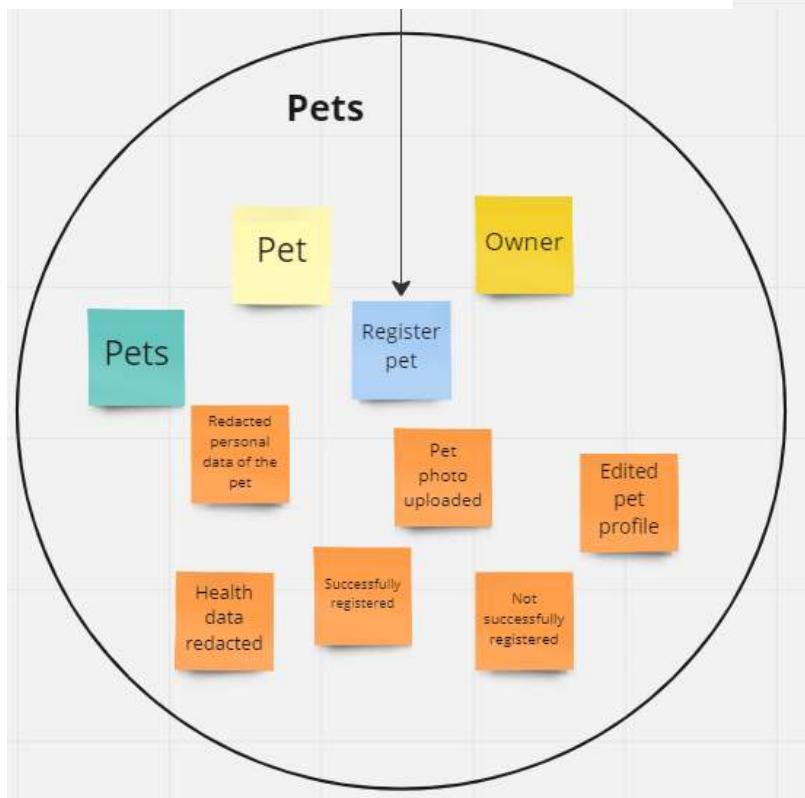
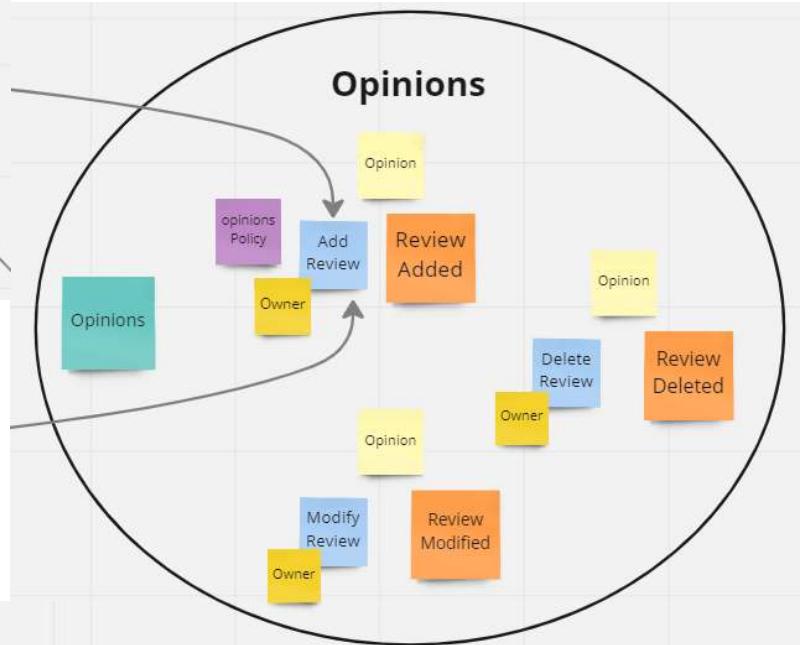
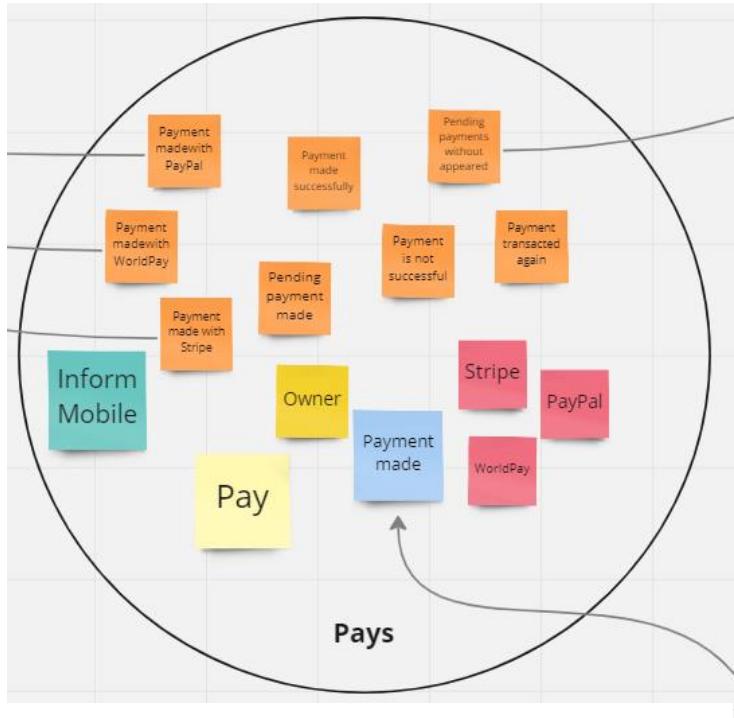
3.2.1. Event Storming

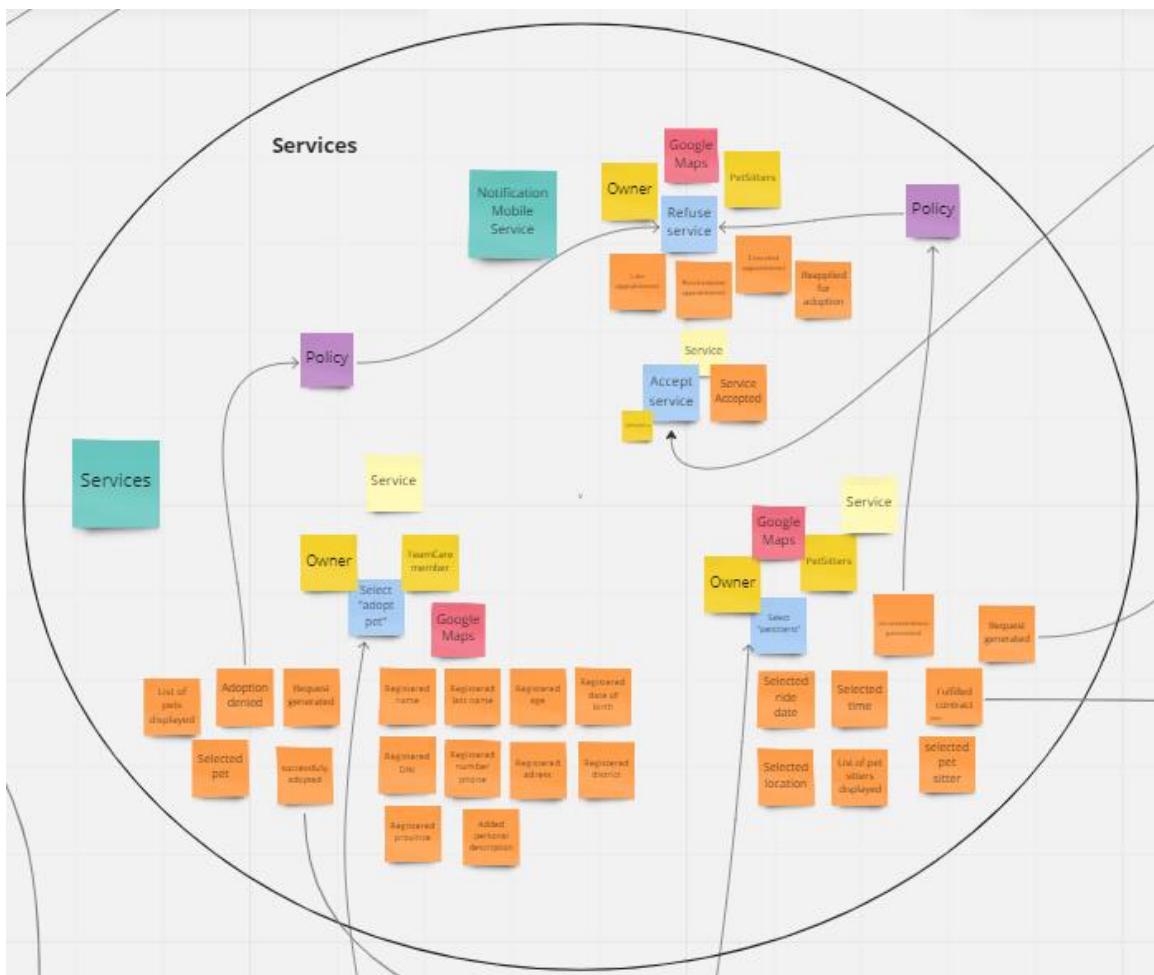
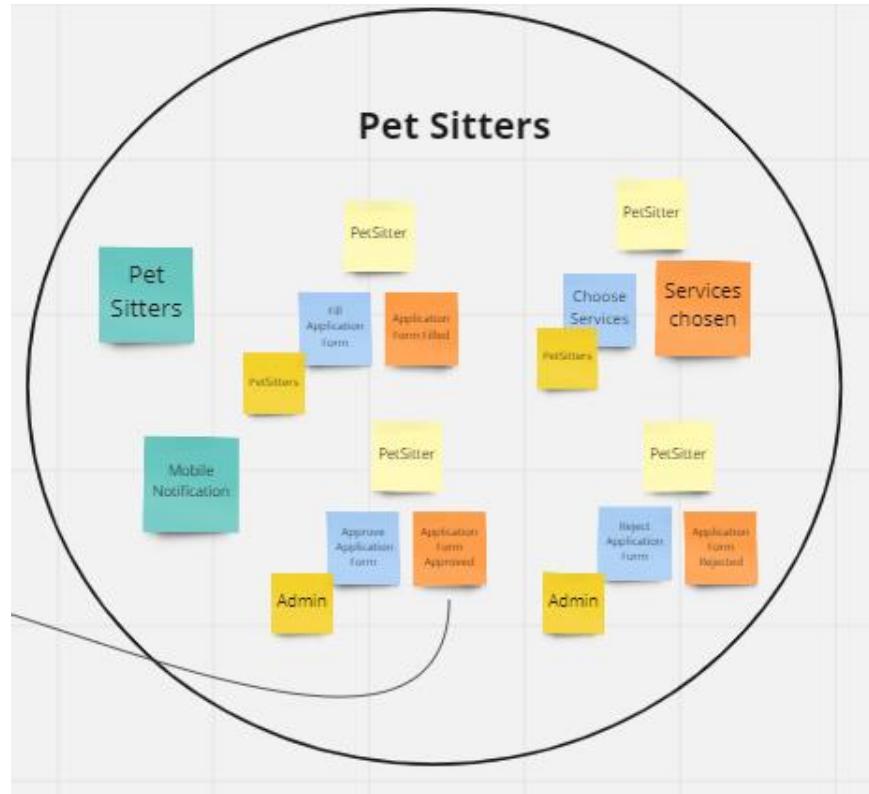
[Ver enlace 3](#)

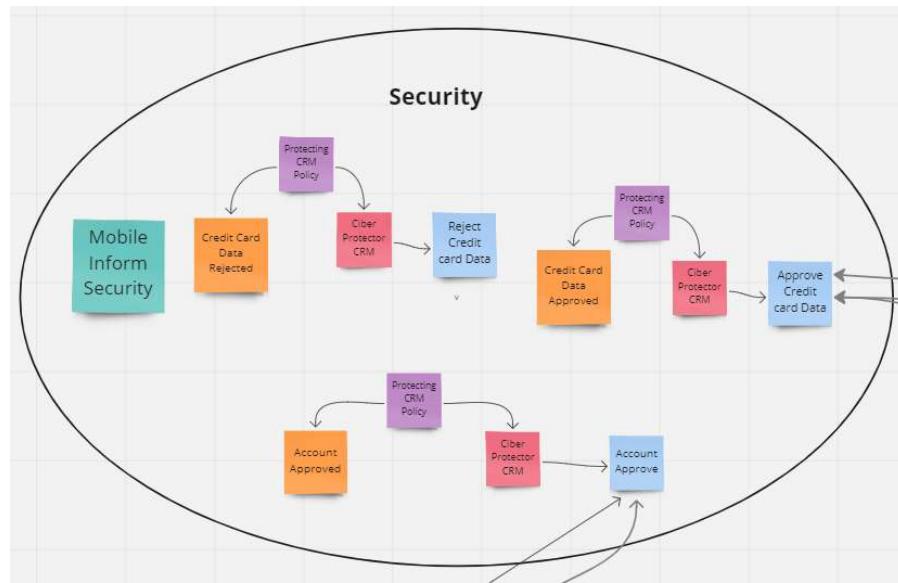








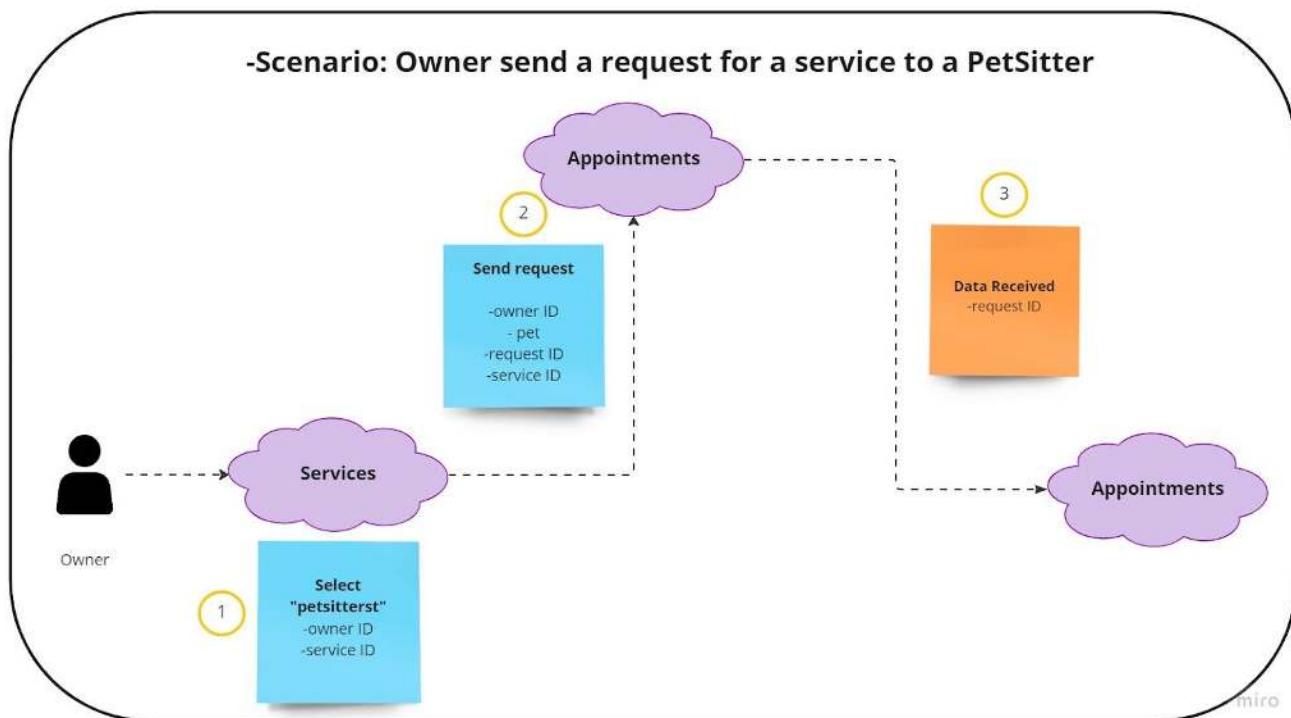


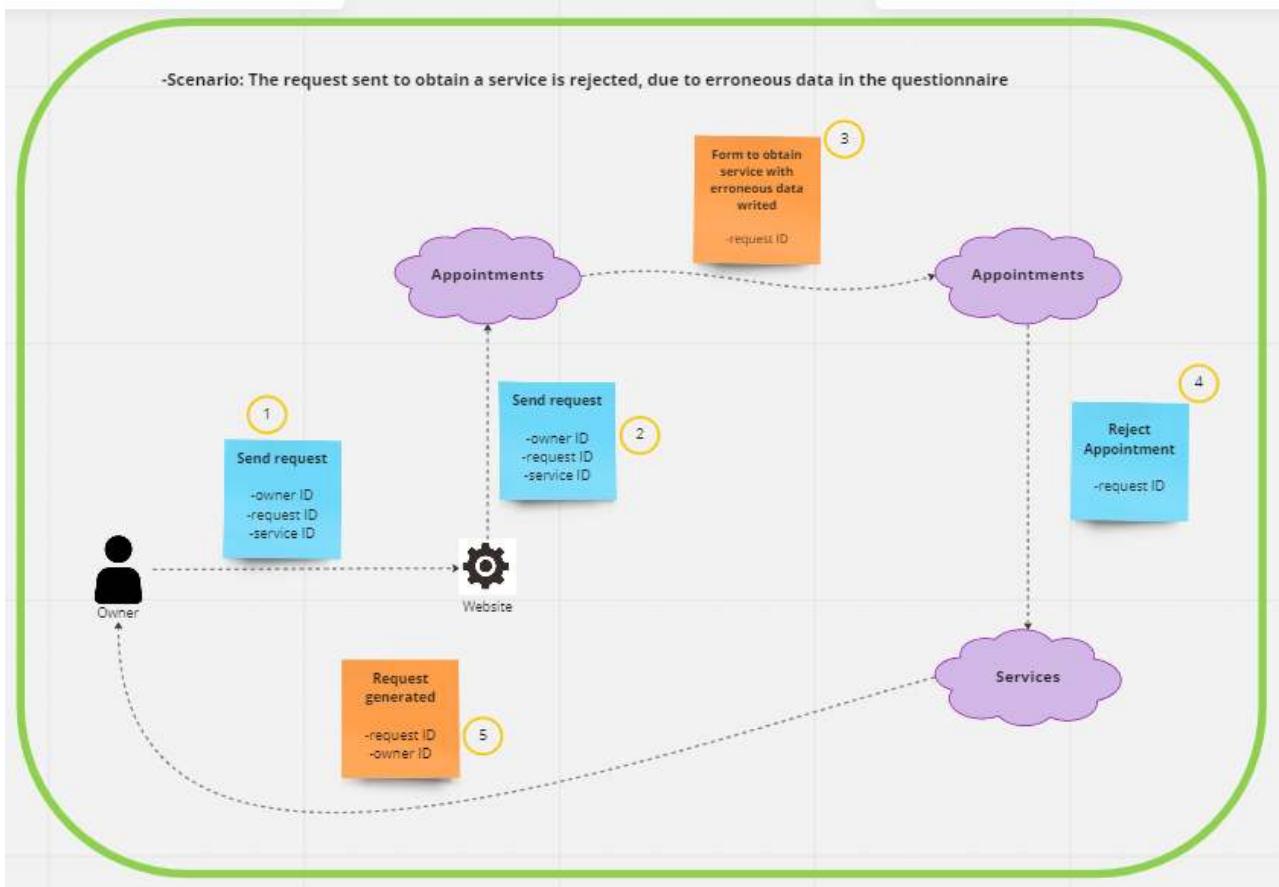
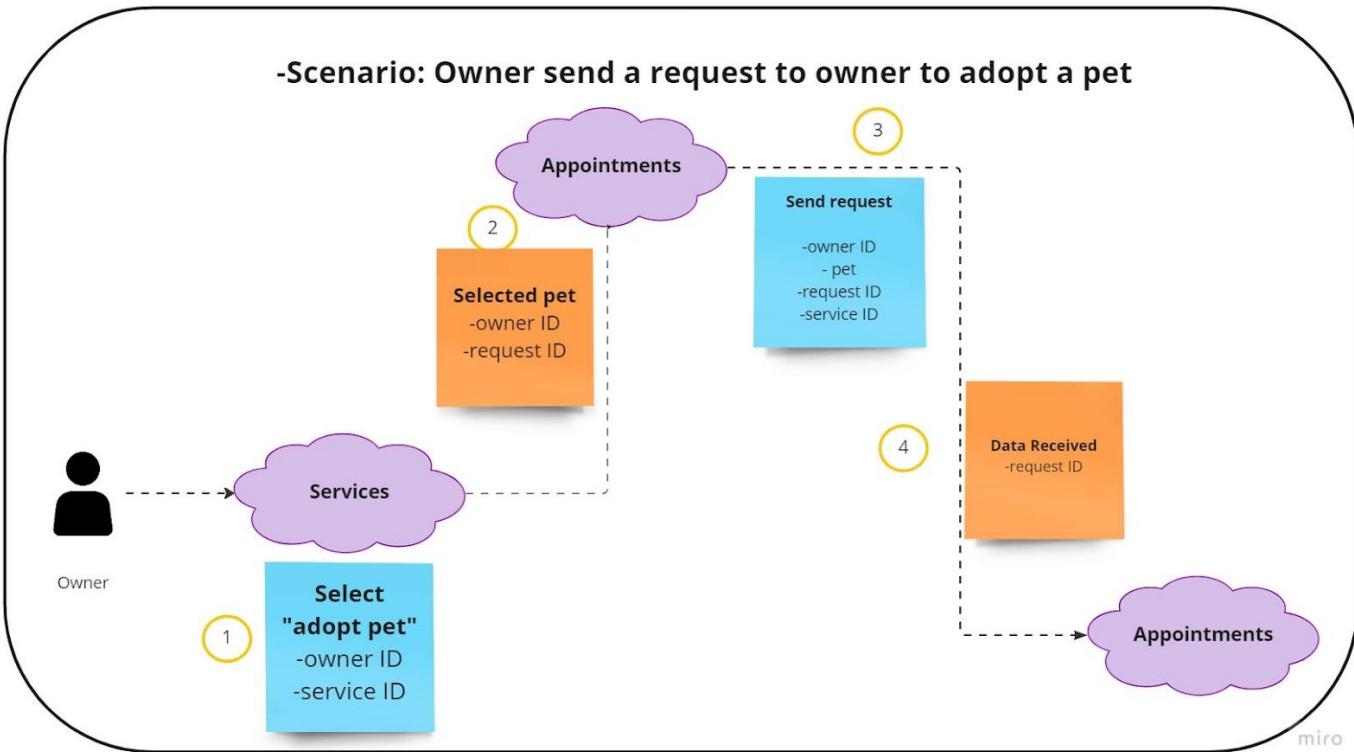


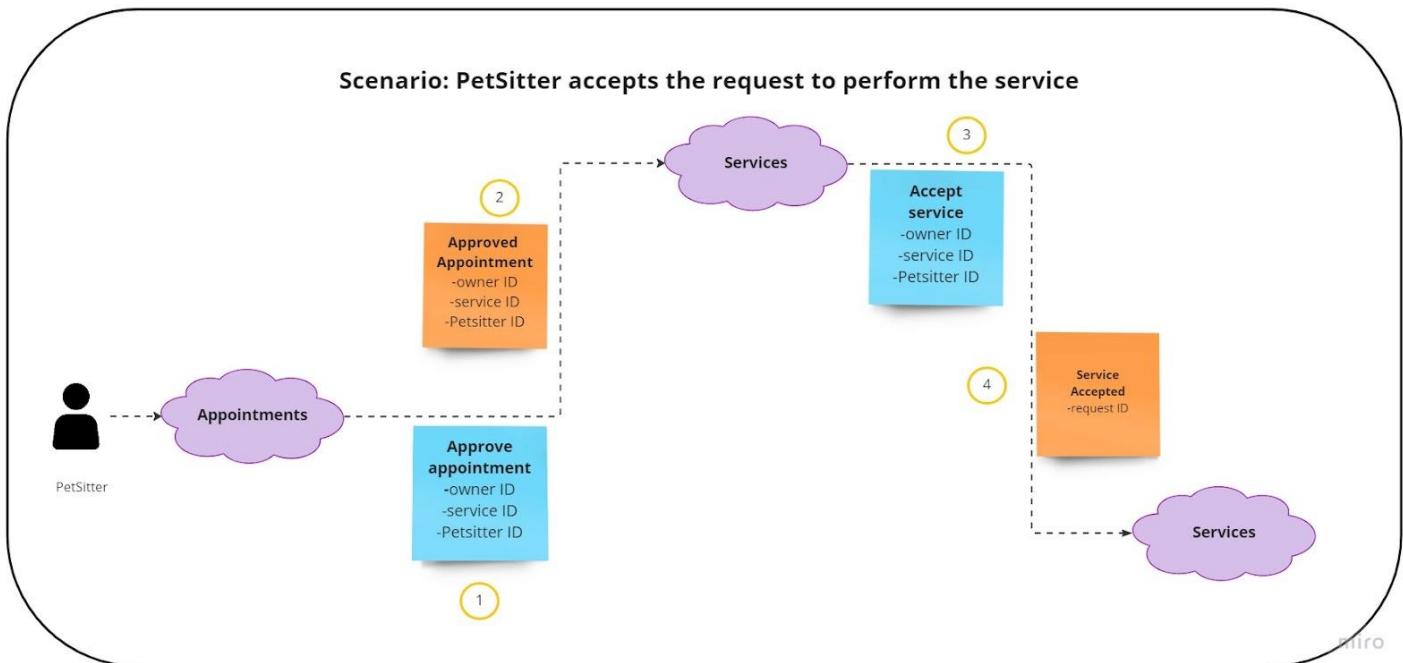
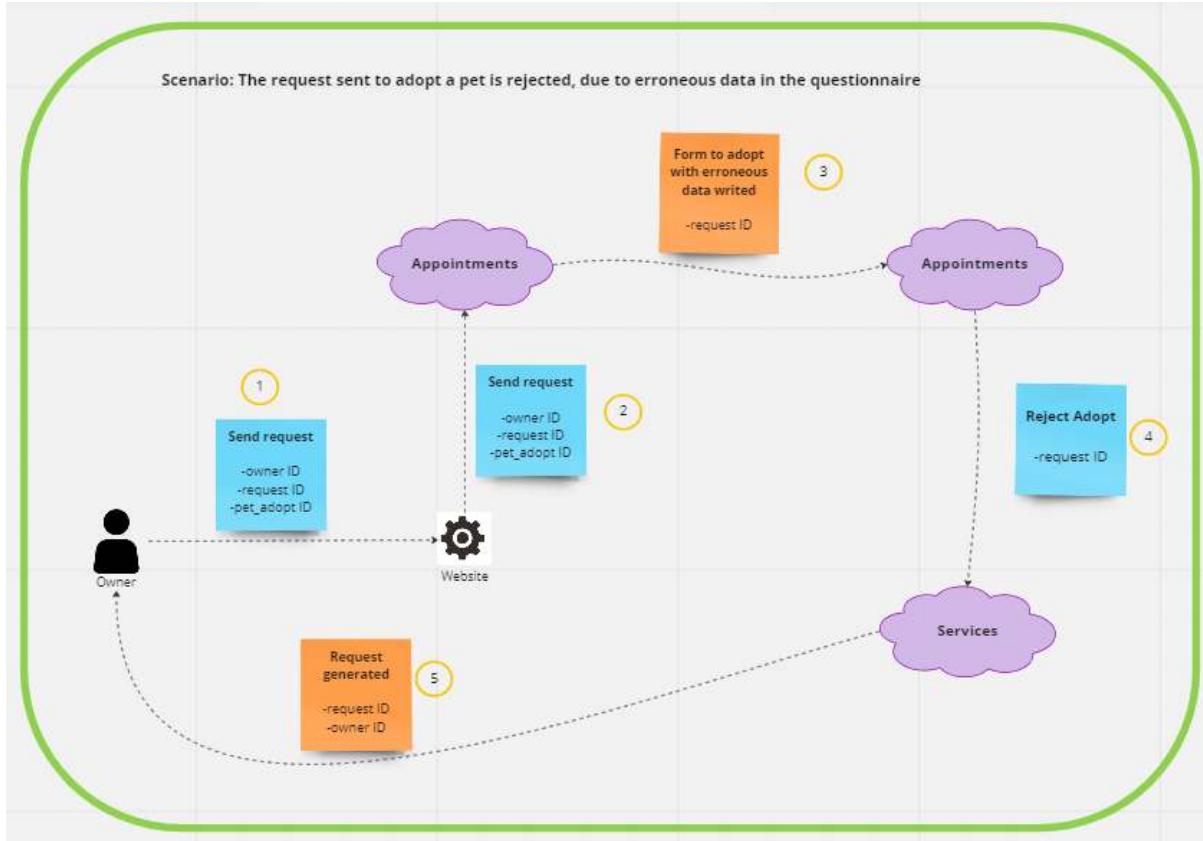
3.2.2. Domain Message Flow Modelling

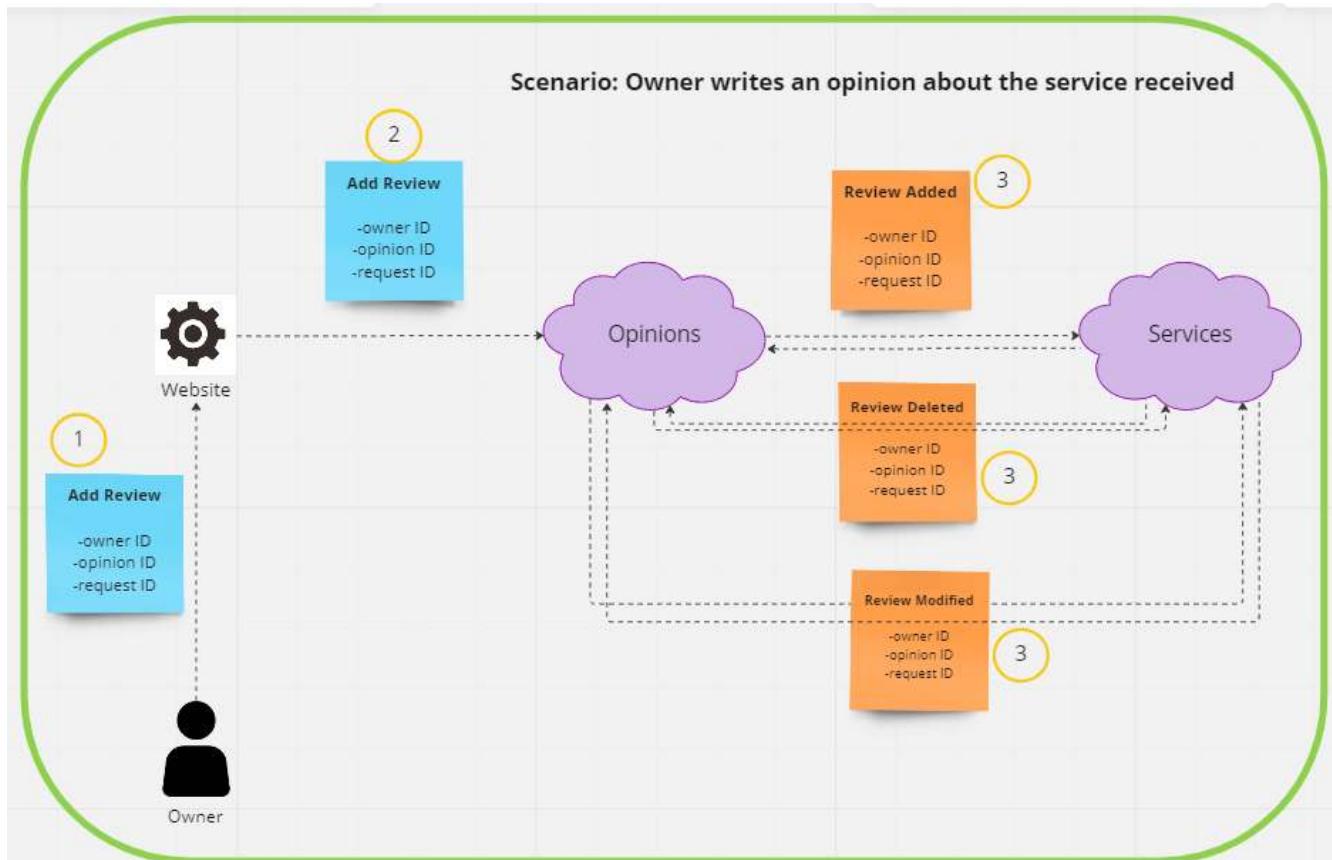
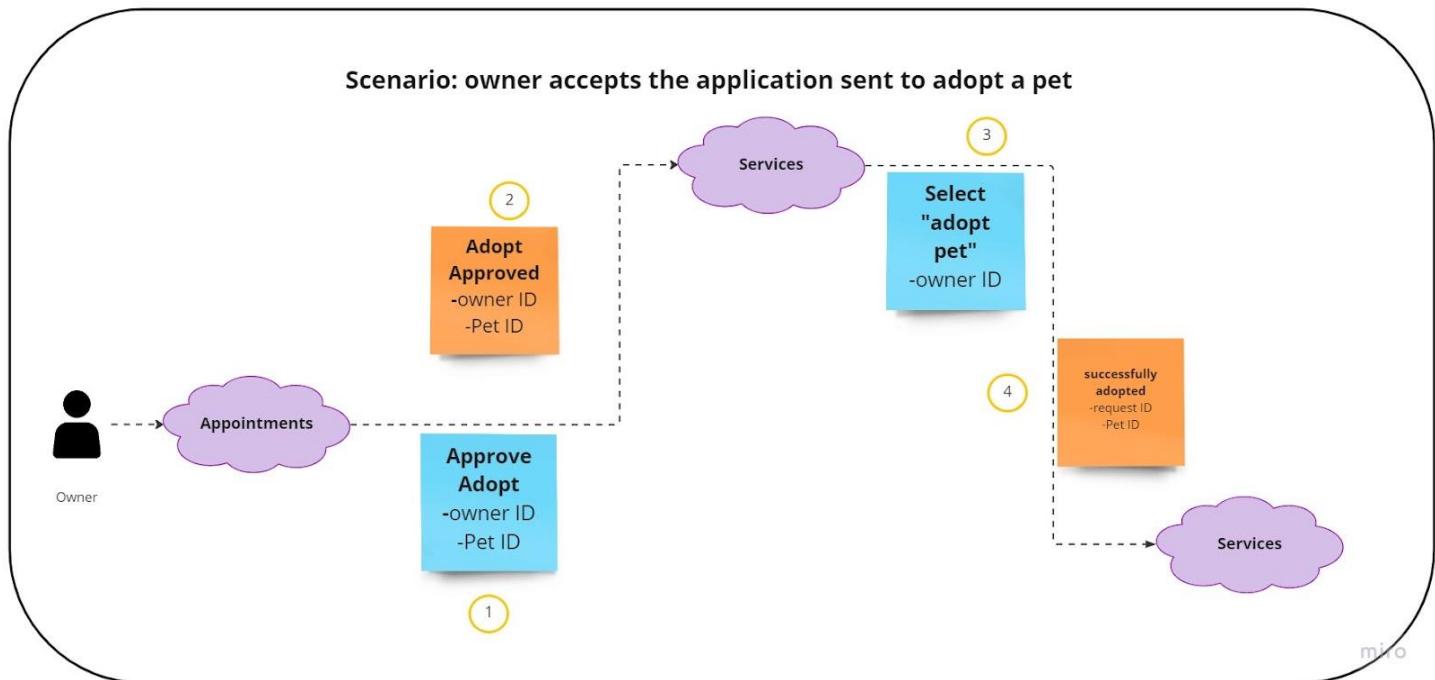
[Ver enlace 4](#)

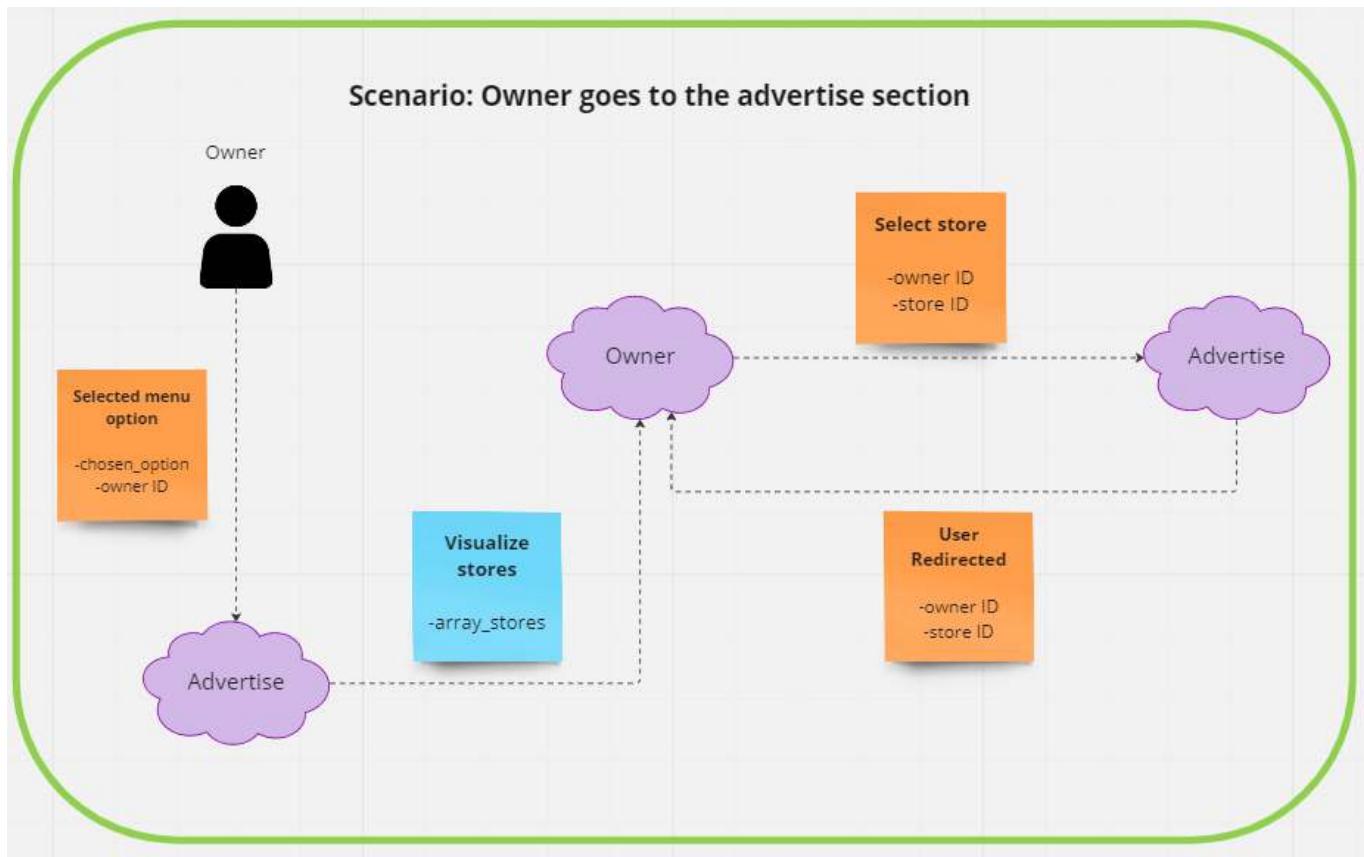
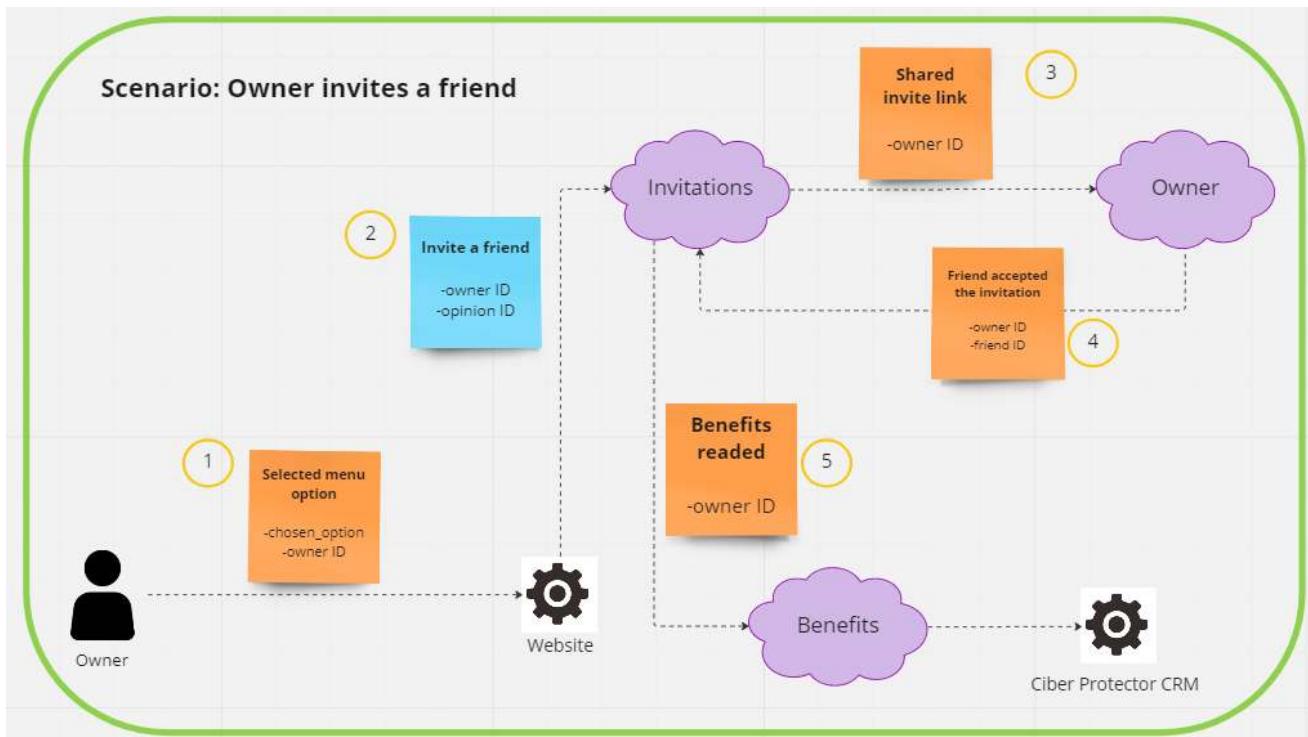
Para esta sección hemos considerado los diez escenarios que destacan nuestra lógica del negocio.

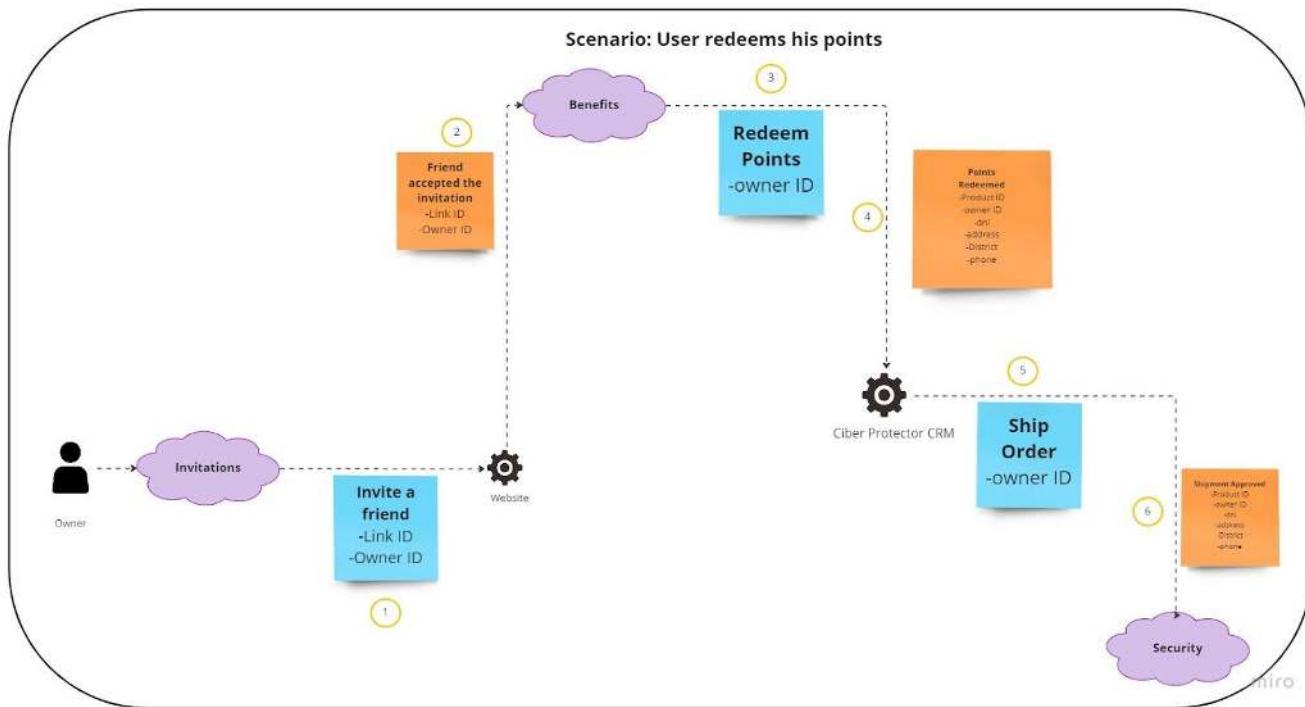












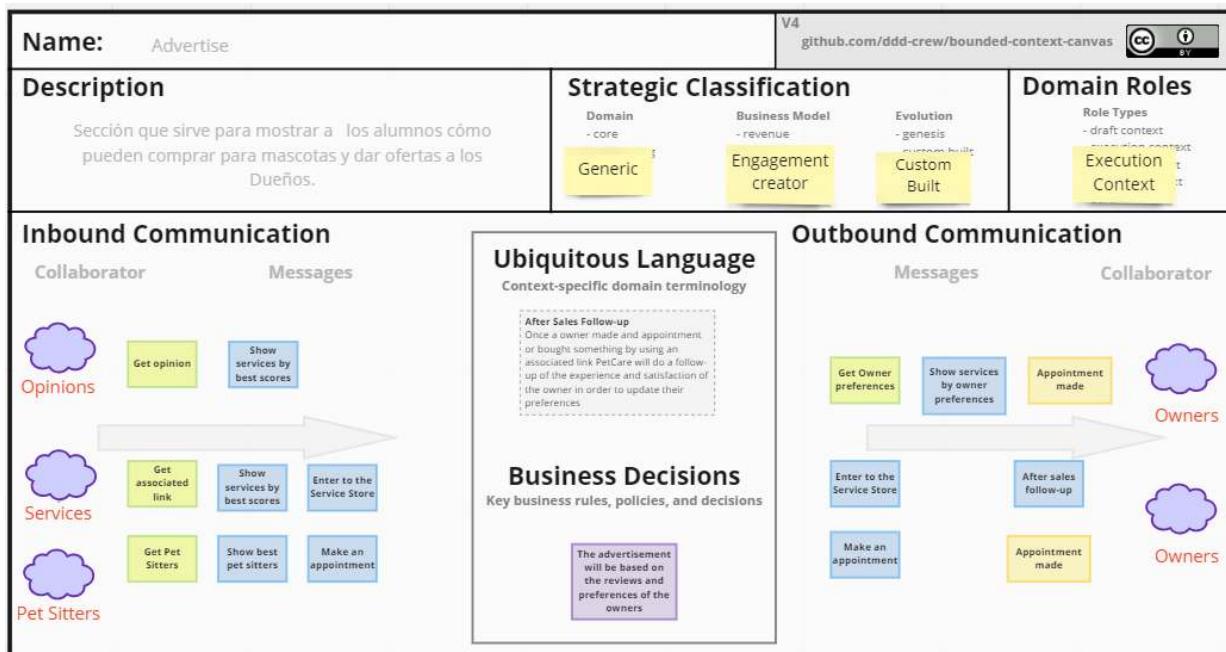
3.2.3. Bounded Contexts

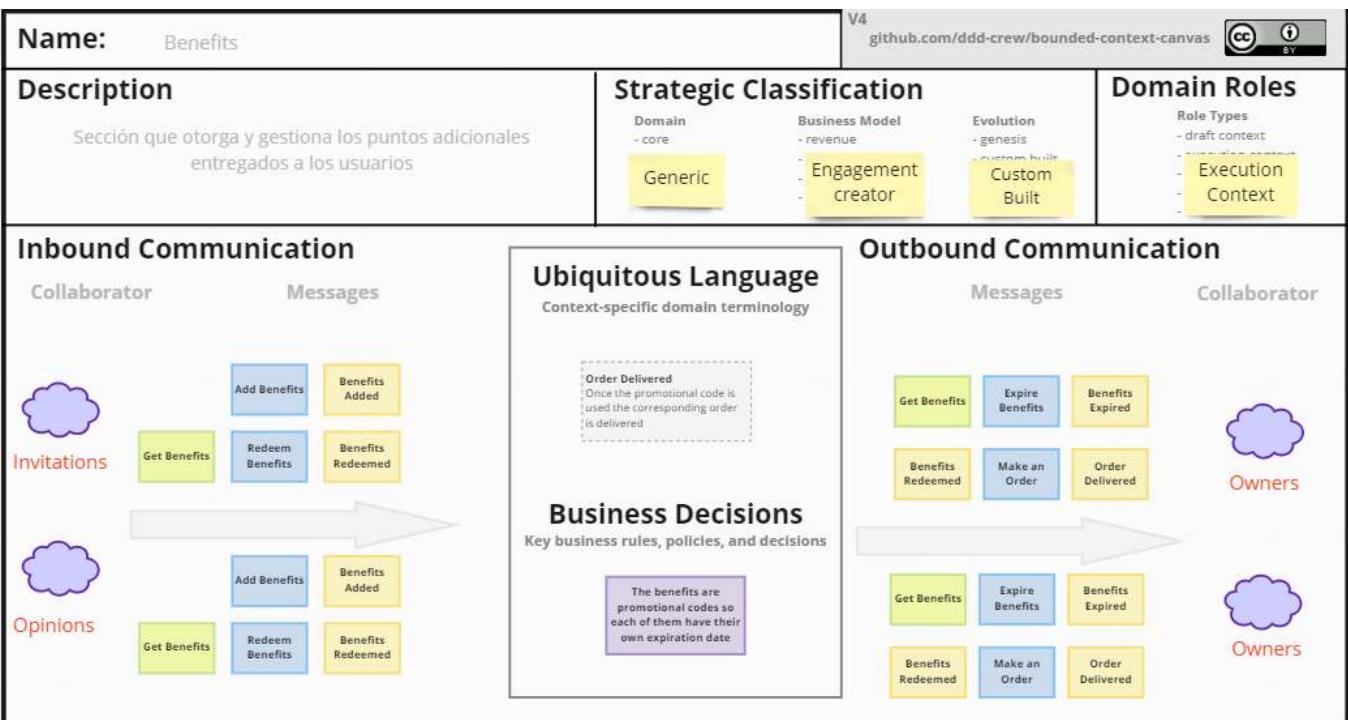
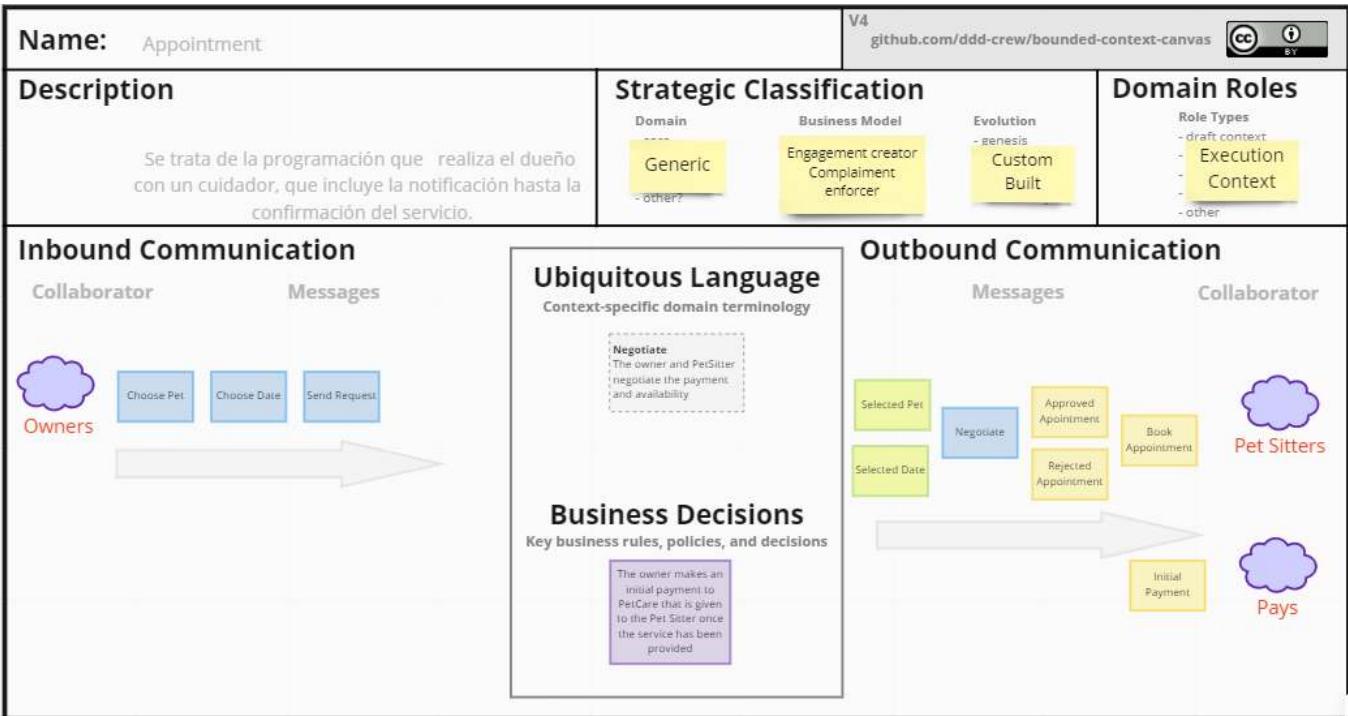
Bounded Context	Descripción
Owners	Se centra en el registro y el ingreso de un dueño de mascota en la aplicación, donde se le crea un perfil y se realizan las validaciones necesarias.
Services	Incluye los servicios que un cuidador puede ofrecer al dueño de una mascota, a la vez que realiza el seguimiento
Appointment	Se trata de la programación que realiza el dueño con un cuidador, que incluye la notificación hasta la confirmación del servicio.
PetSitters	Perfiles de los miembros PetCare que se desempeñan como cuidadores, también realiza el seguimiento de los contratos.

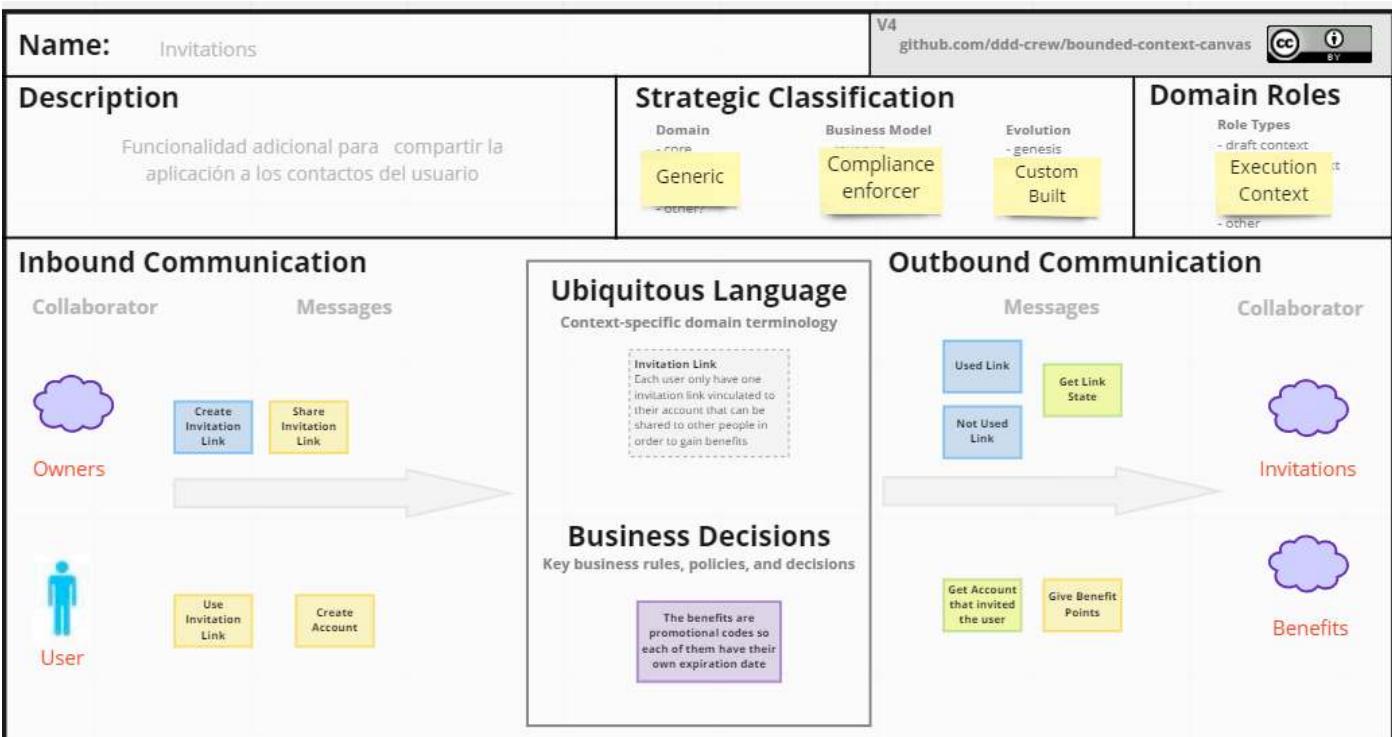
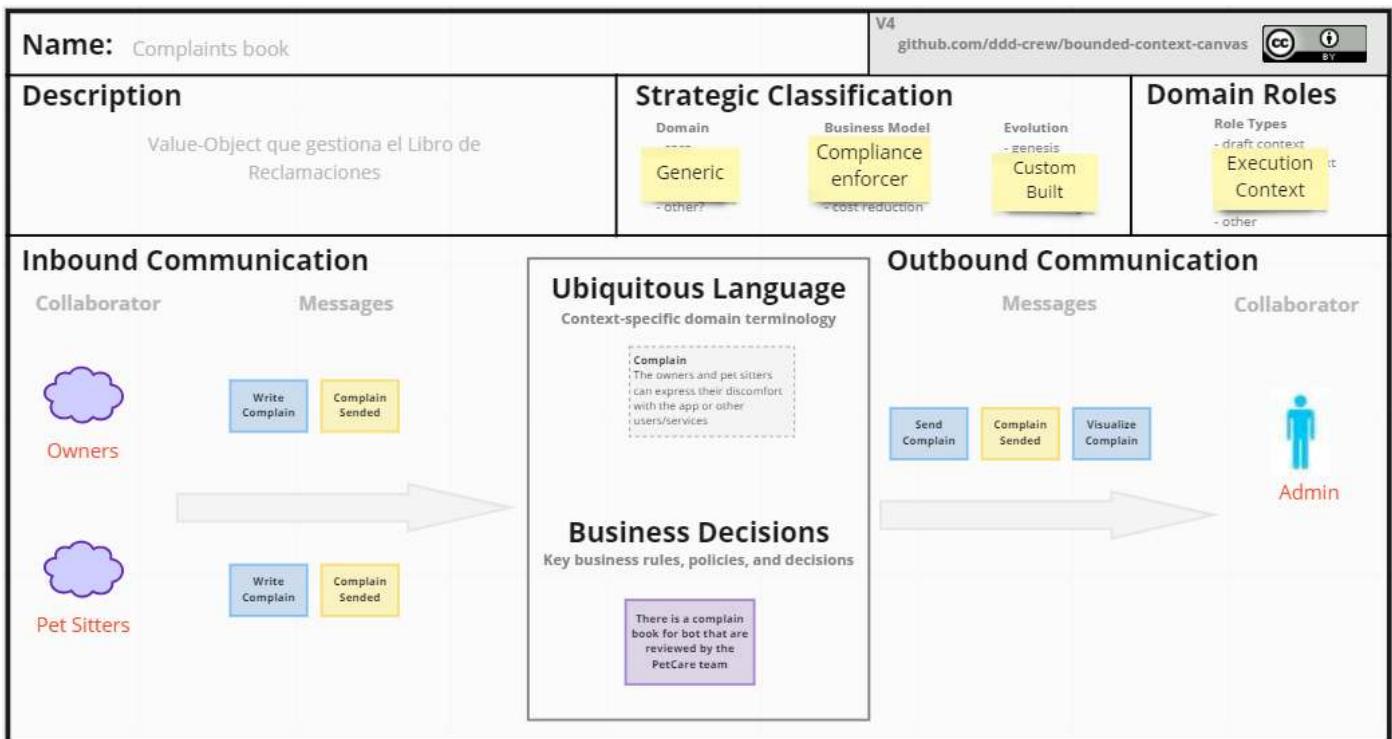
Invitations	Funcionalidad adicional para compartir la aplicación a los contactos del usuario
Pets	Módulo que abarca las características de las mascotas del dueño y su perfil para cuidado
Pays	Funcionalidad para el pago entre contratante y contratado.
Benefits	Sección que otorga y gestiona los puntos adicionales entregados a los usuarios
Opinions	Funcionalidad que permite realizar reseñas
Complaints book	Value-Object que gestiona el Libro de Reclamaciones
Advertise	Sección que sirve para promocionar a los dueños de mascota los negocios que son socios de la app
Security	Procedimientos de seguridad en la aplicación

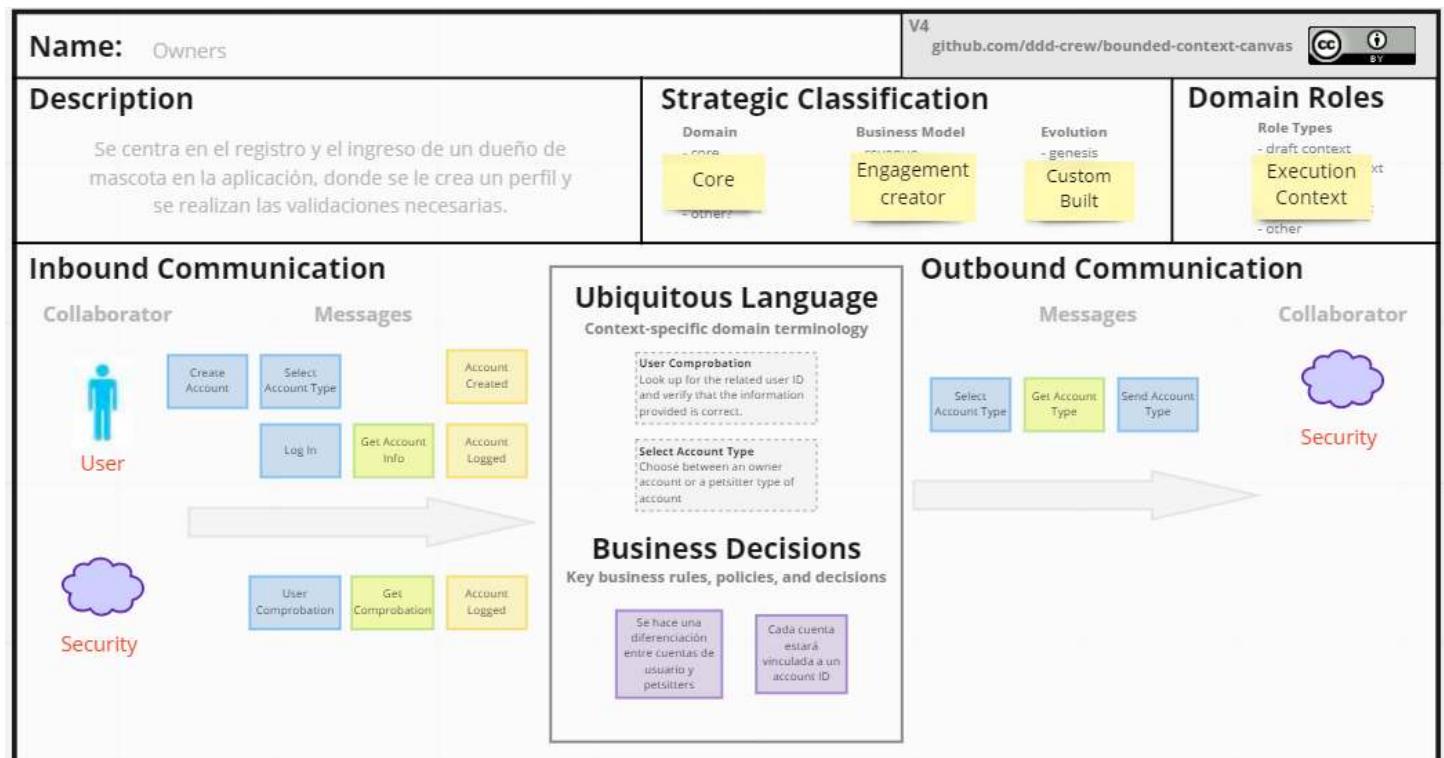
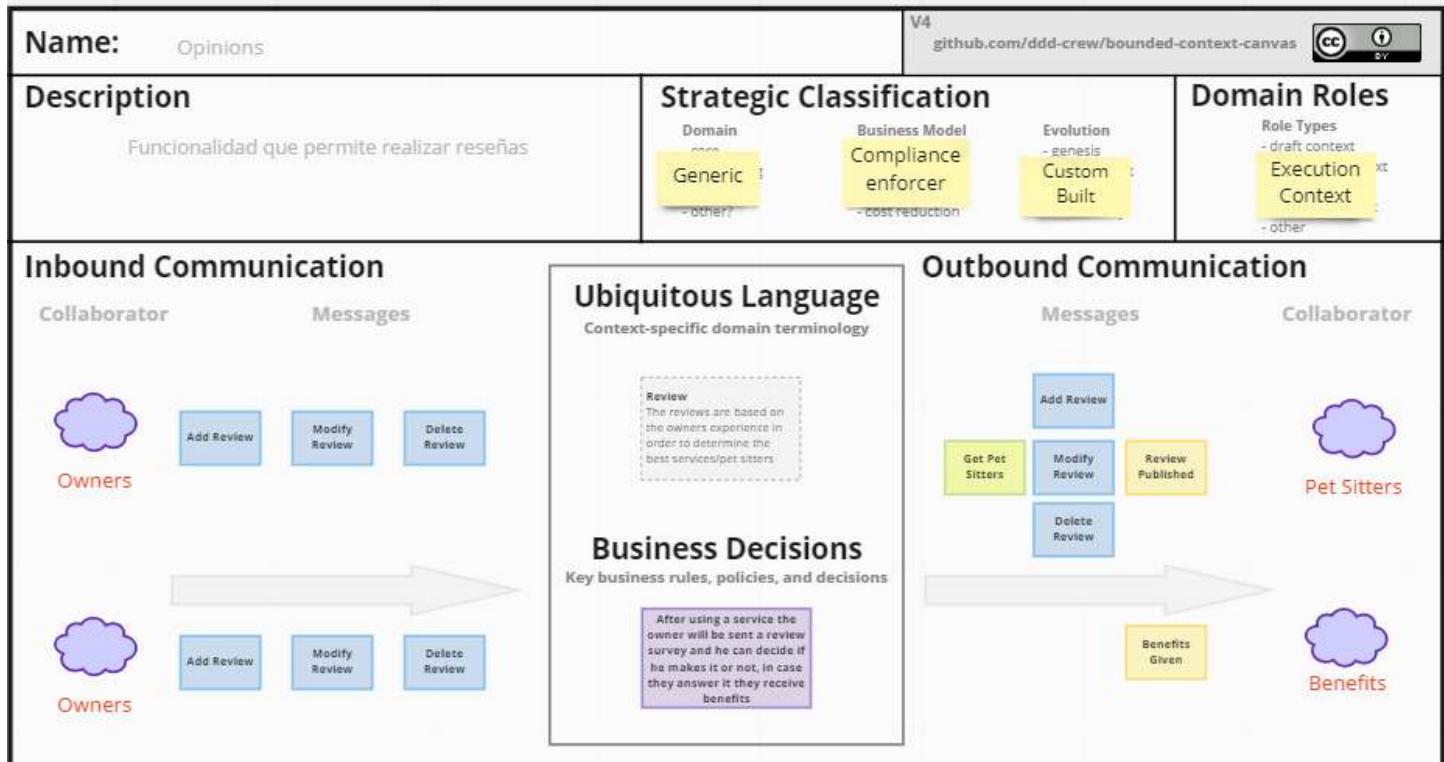
3.2.4. Bounded Context Canvas

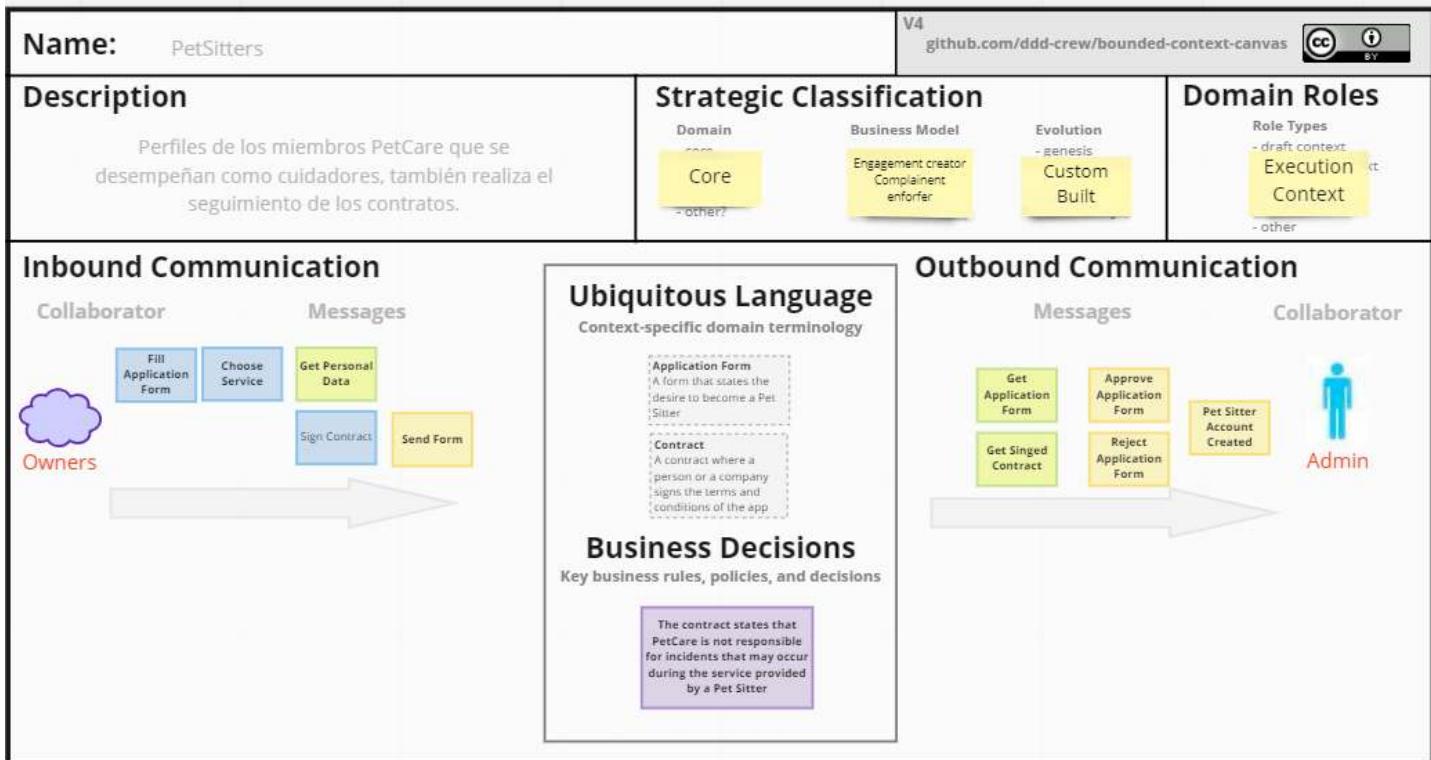
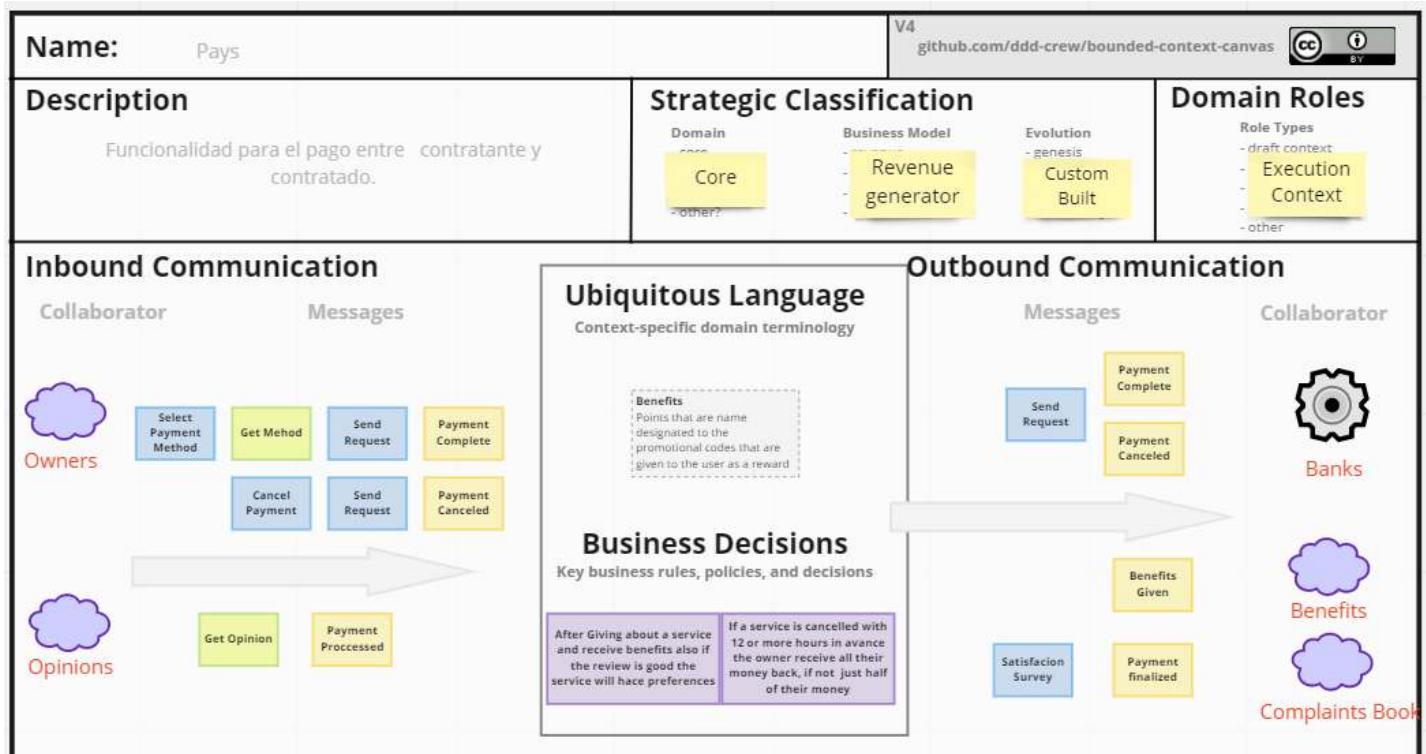
[Ver enlace 5](#)

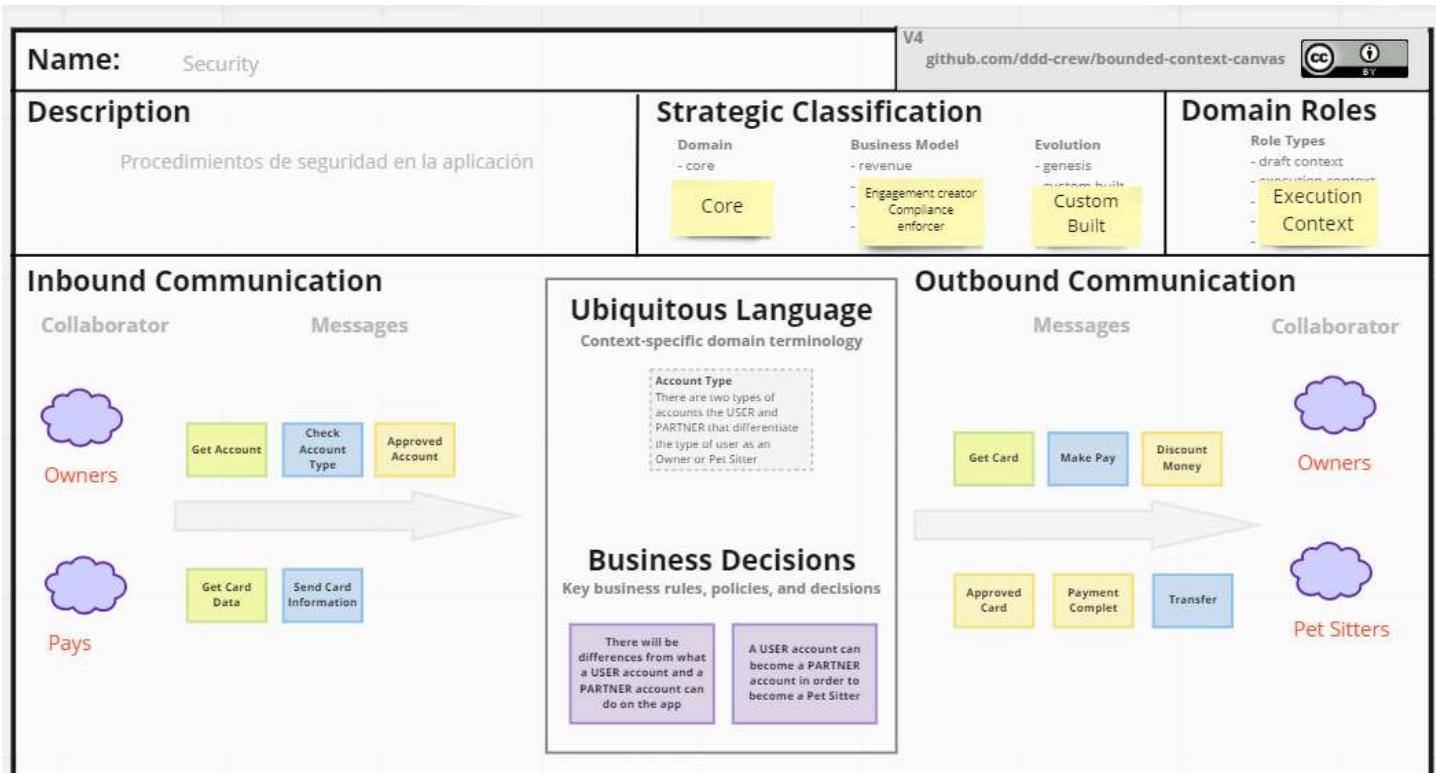
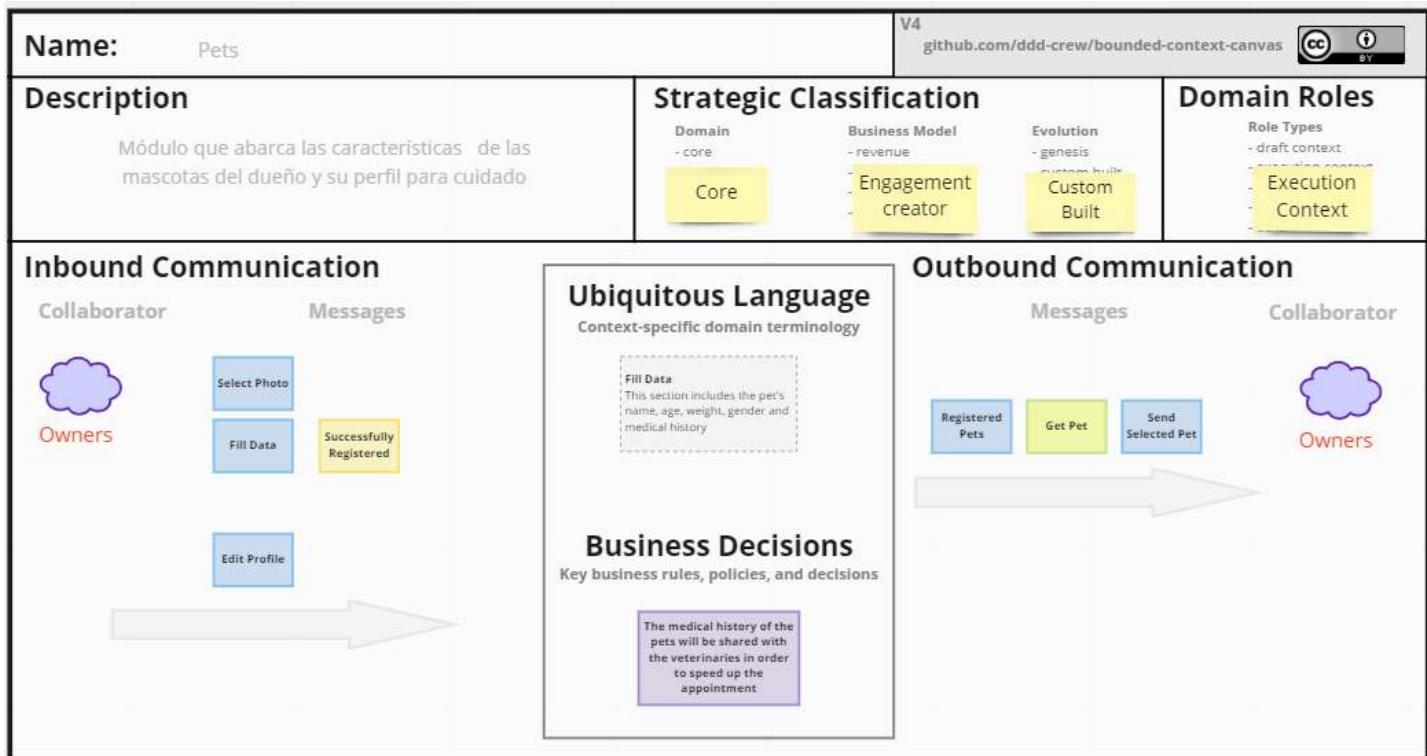


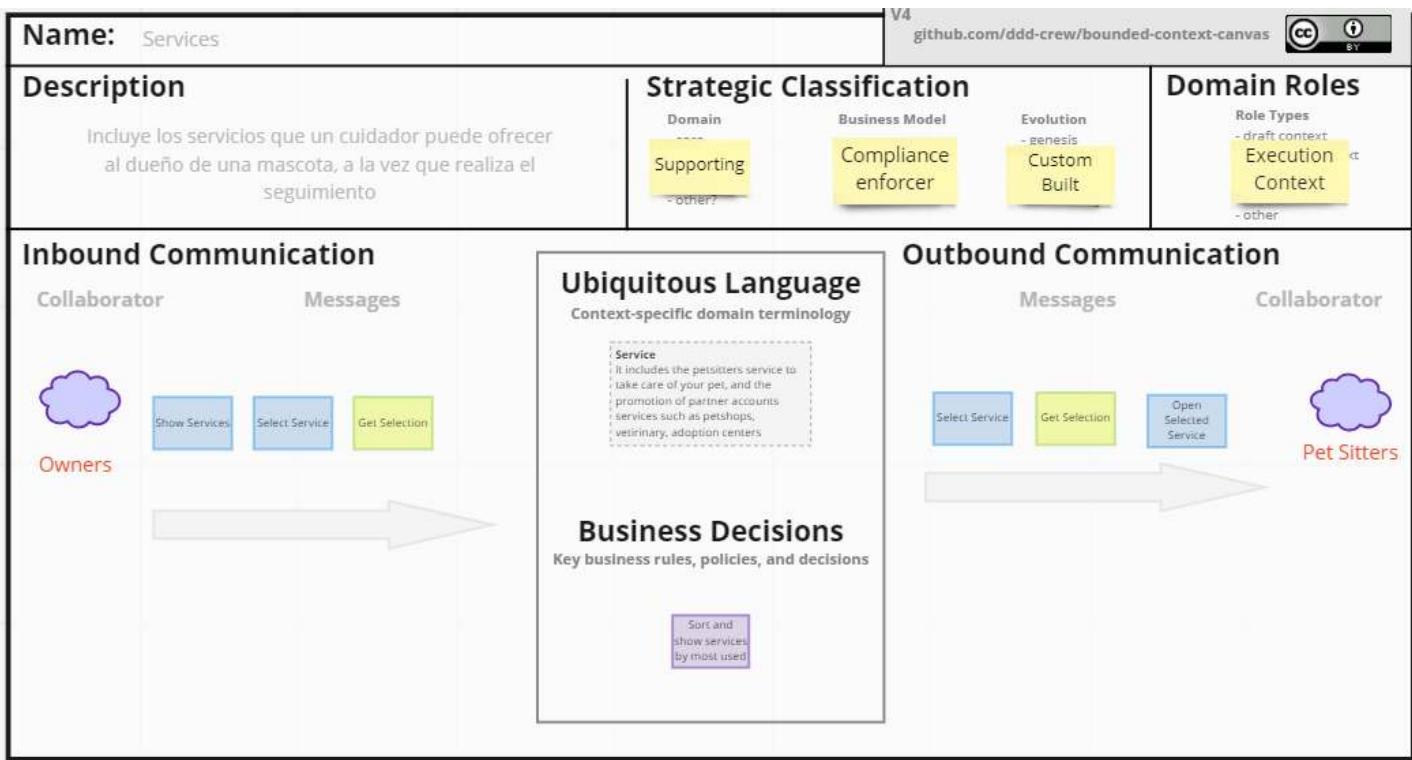






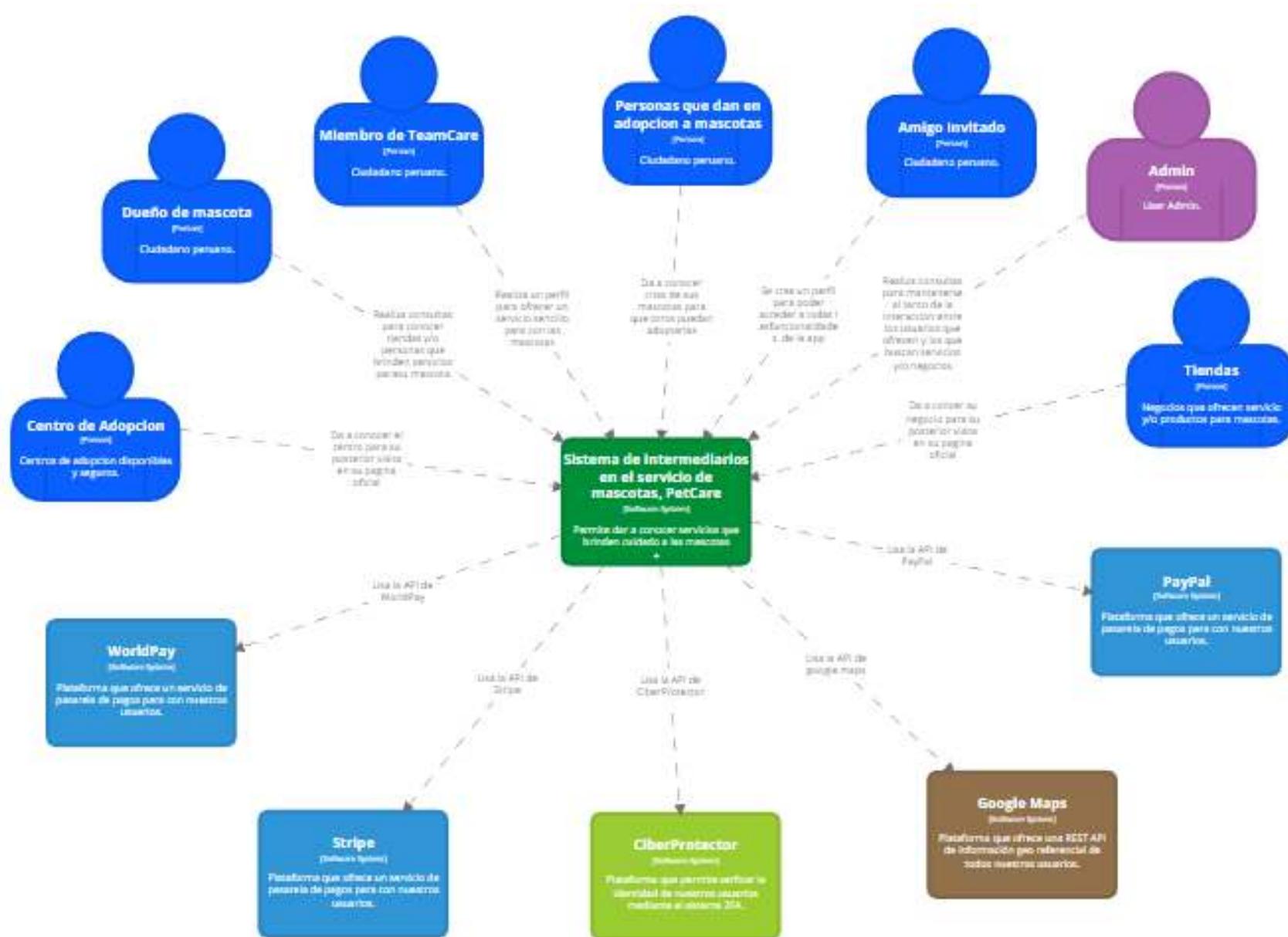






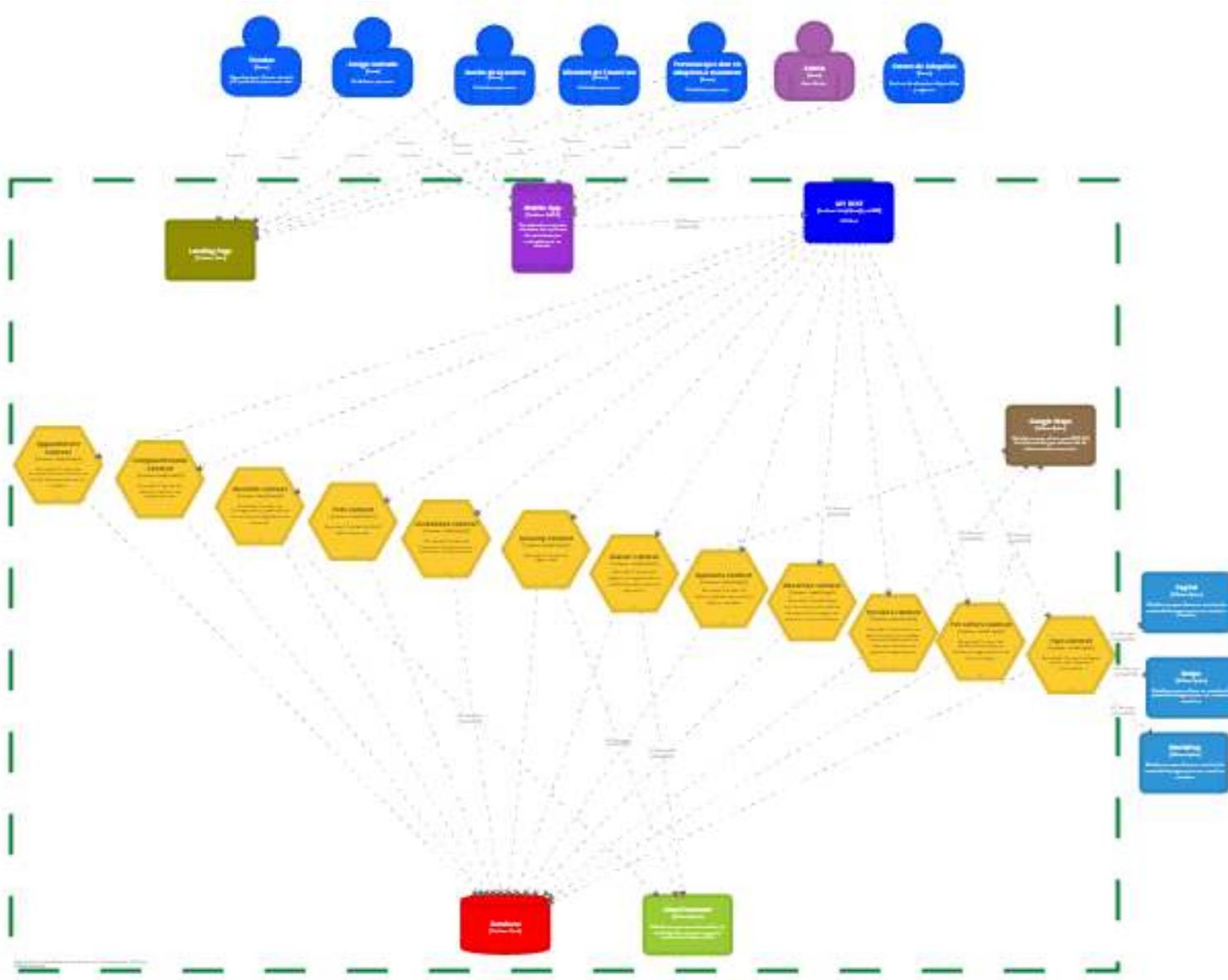
3.2.5. Diagrama de Contexto

Link Diagramas C4: [Ver enlace 6](#)



[System Context] Sistema de intermediarios en el servicio de mascotas, PetCare

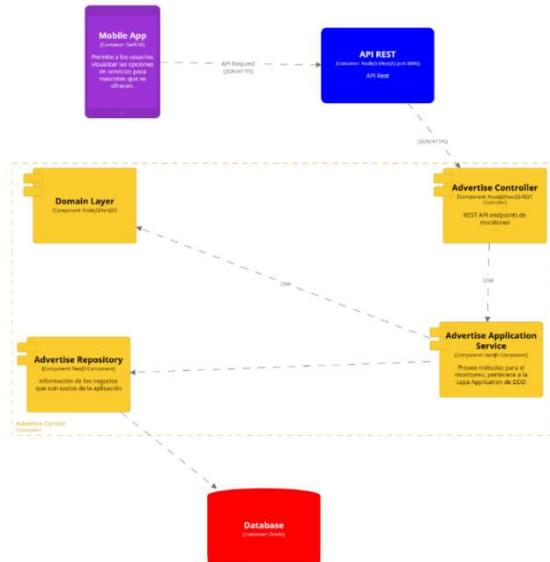
3.2.6. Diagrama de Contenedores



3.3. Diseño táctico

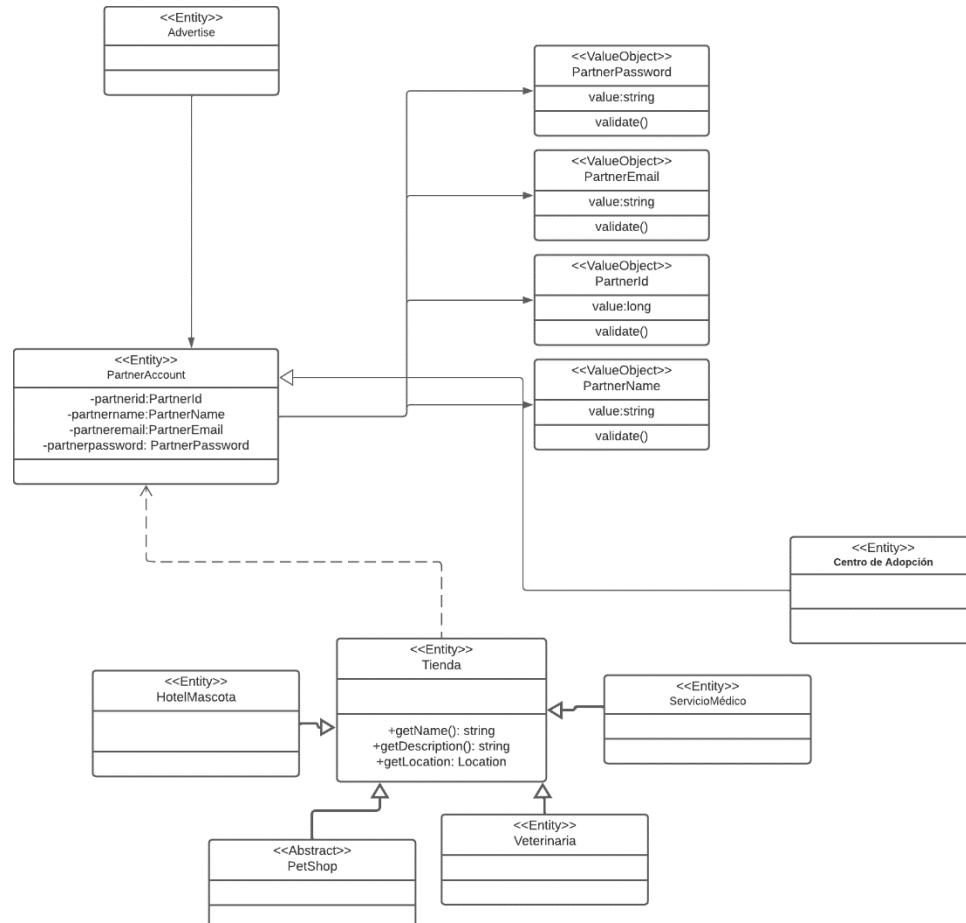
3.3.1. Bounded Context Advertise

3.3.1.1. Diagrama de Componentes

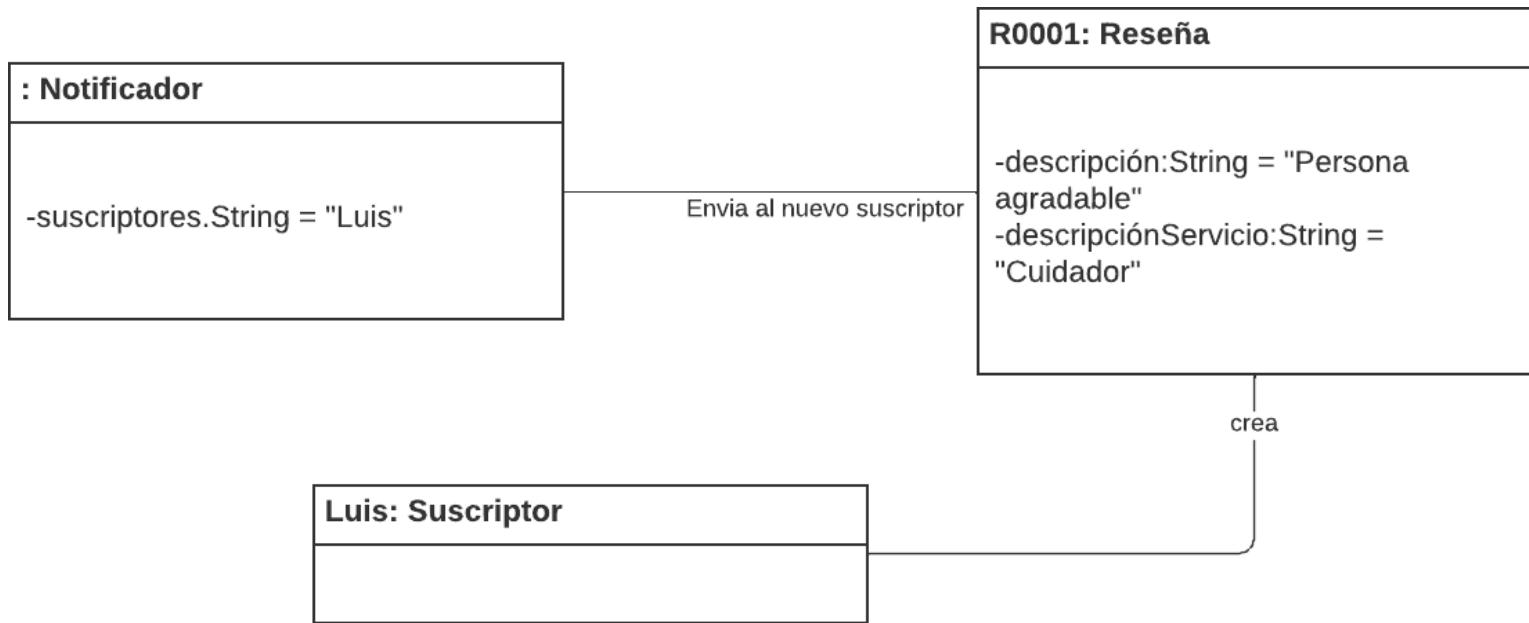


3.3.1.2. Diagramas de código

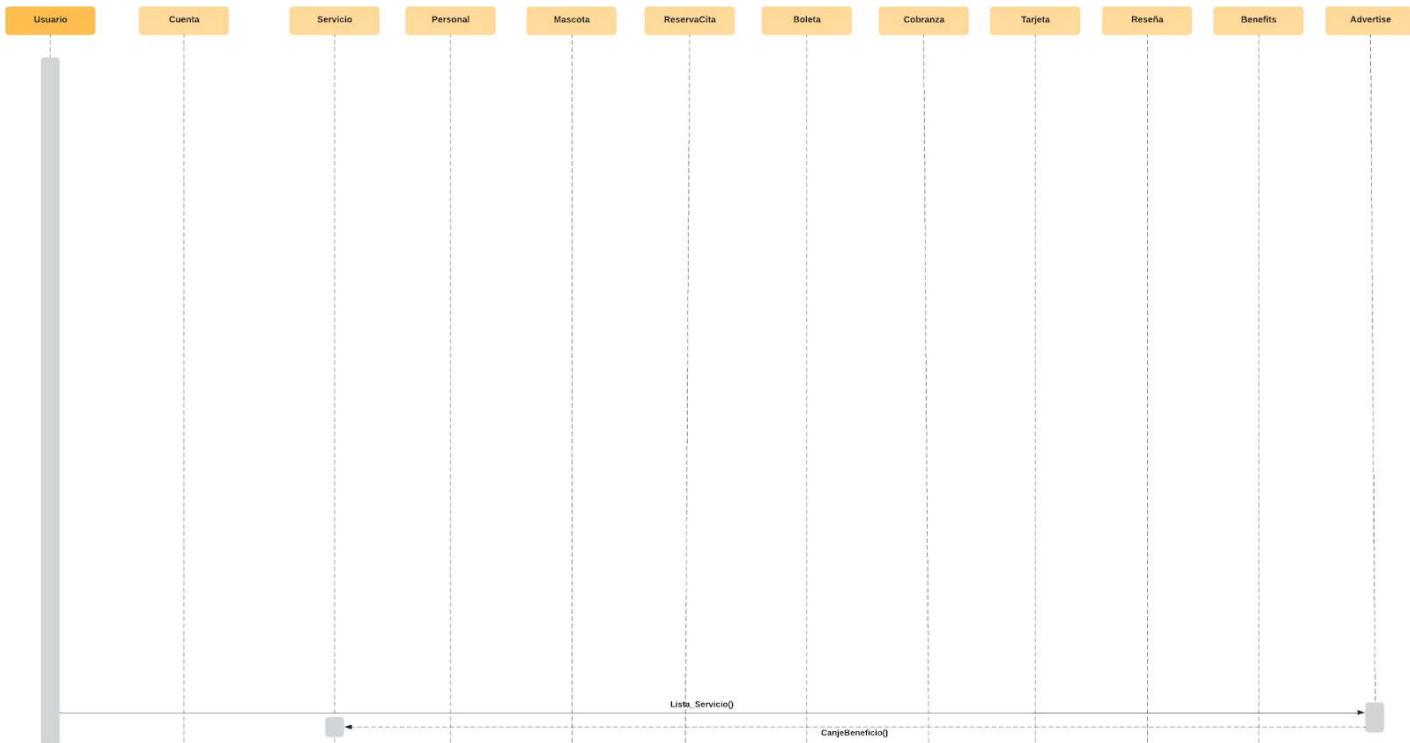
3.3.1.2.1. Diagrama de Clases



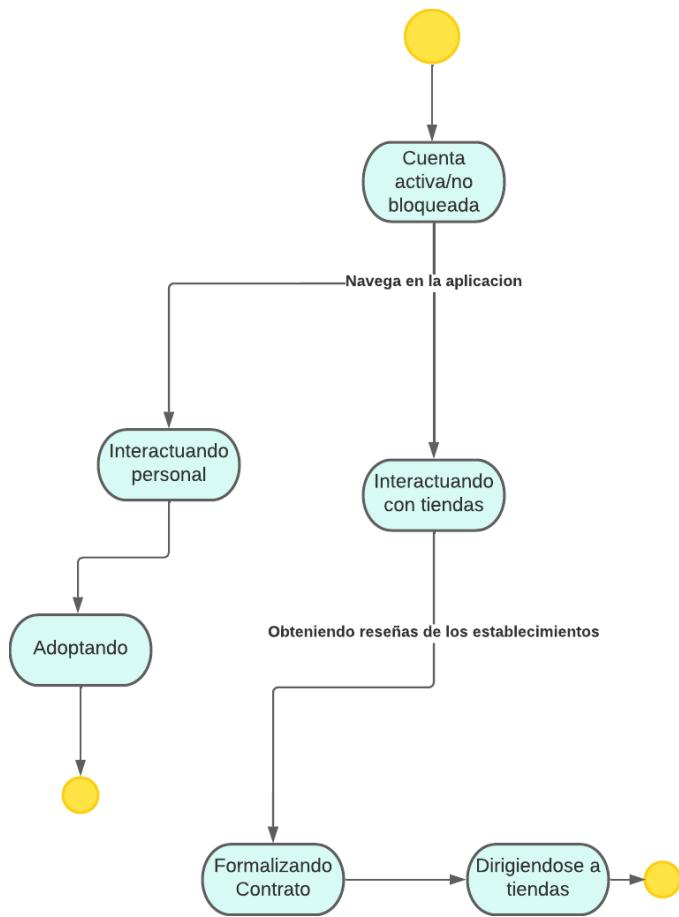
3.3.1.2.2. Diagrama de Objetos



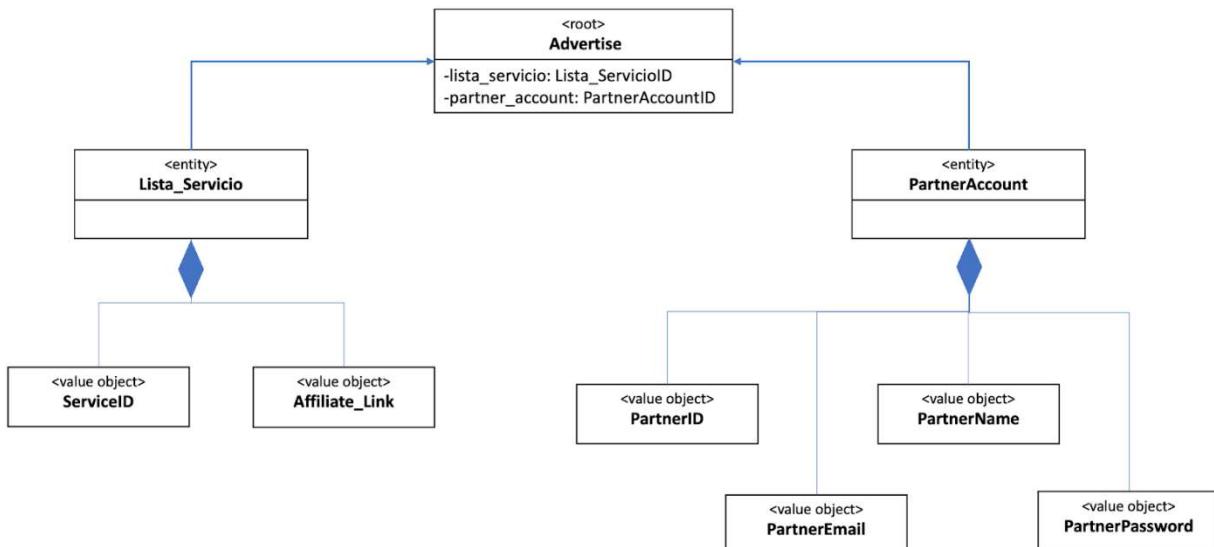
3.3.1.2.3. Diagrama de Secuencia



3.3.1.2.4. Diagrama de Estados

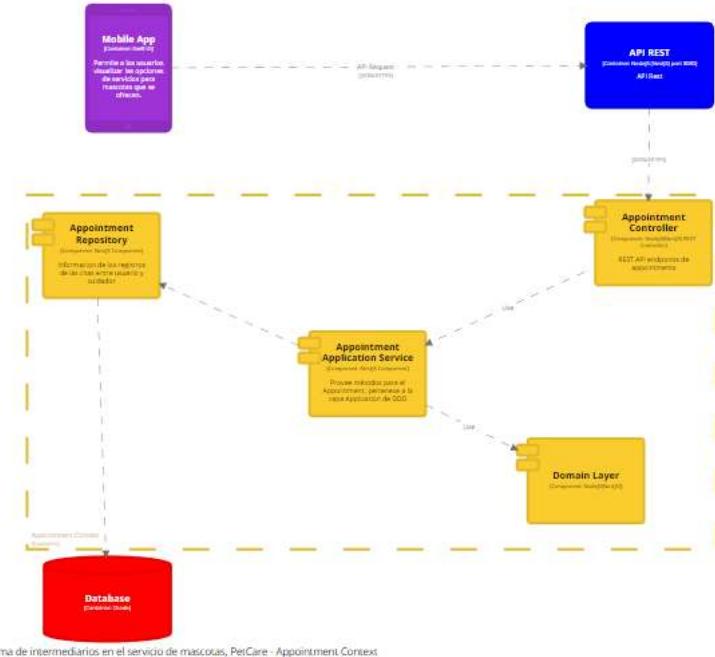


3.3.1.3. Diagrama de Agregados: Advertise



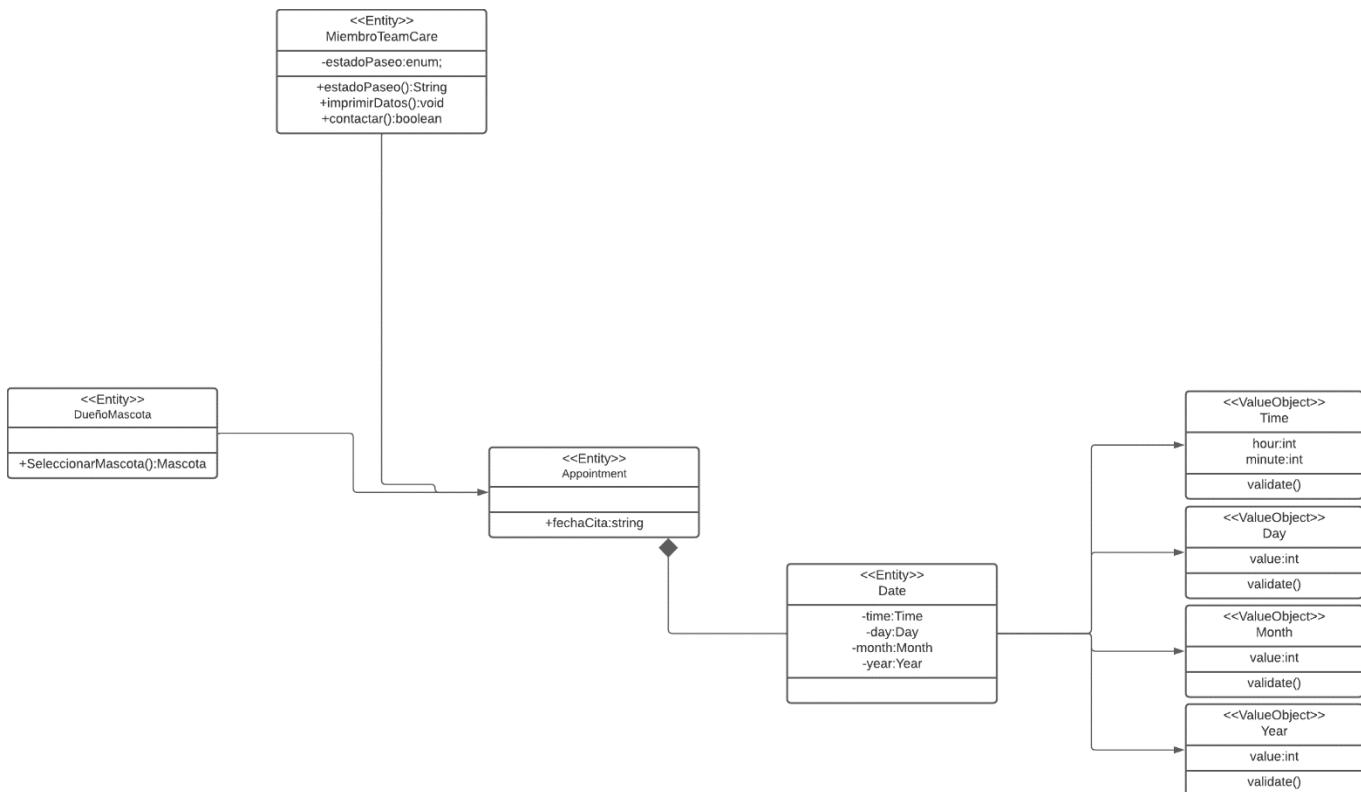
3.3.2. Bounded Context Appointment

3.3.2.1. Diagrama de Componentes

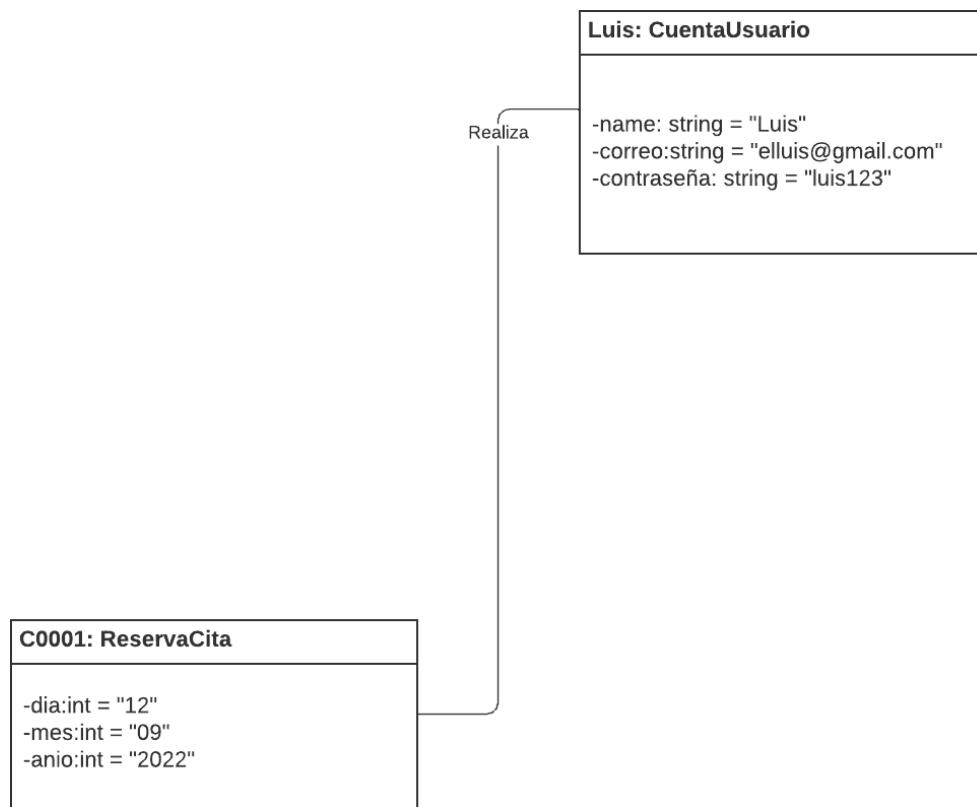


3.3.2.2. Diagramas de código

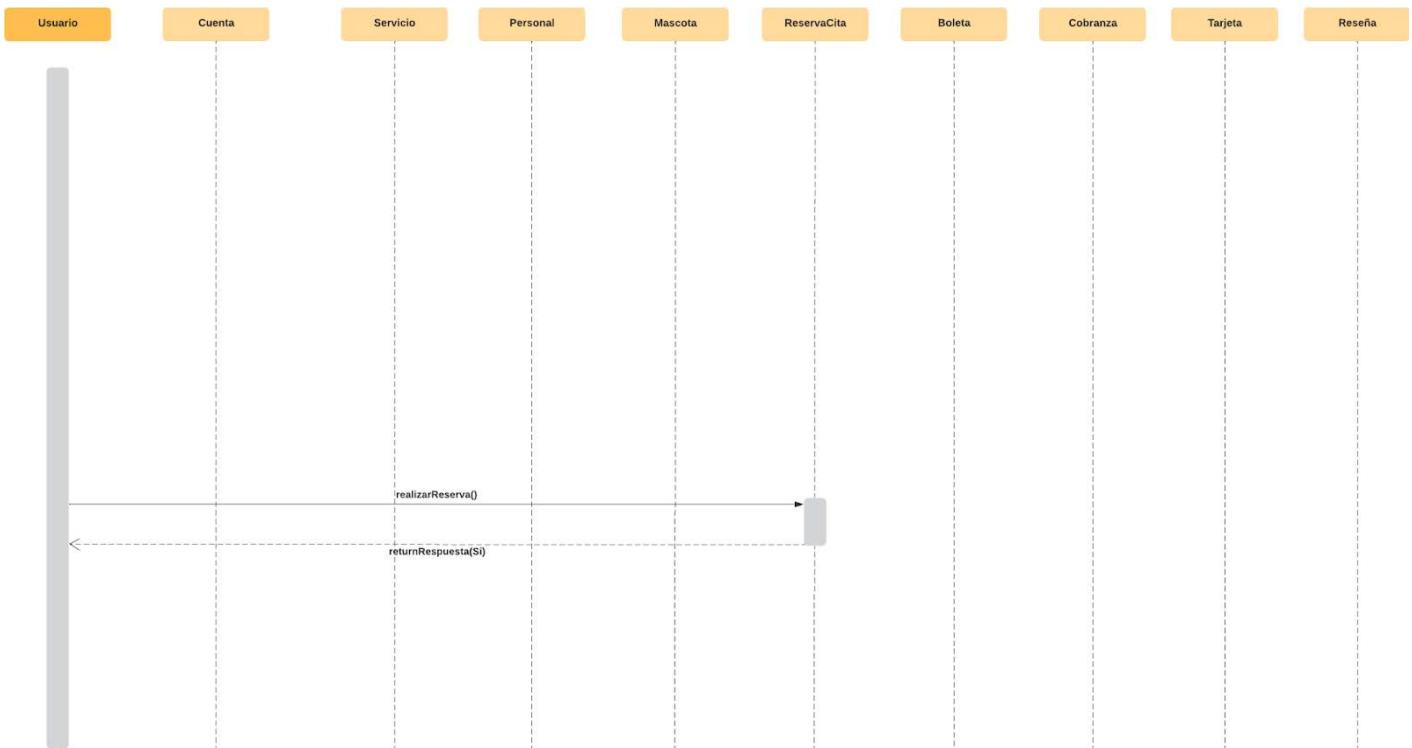
3.3.2.2.1. Diagrama de Clases



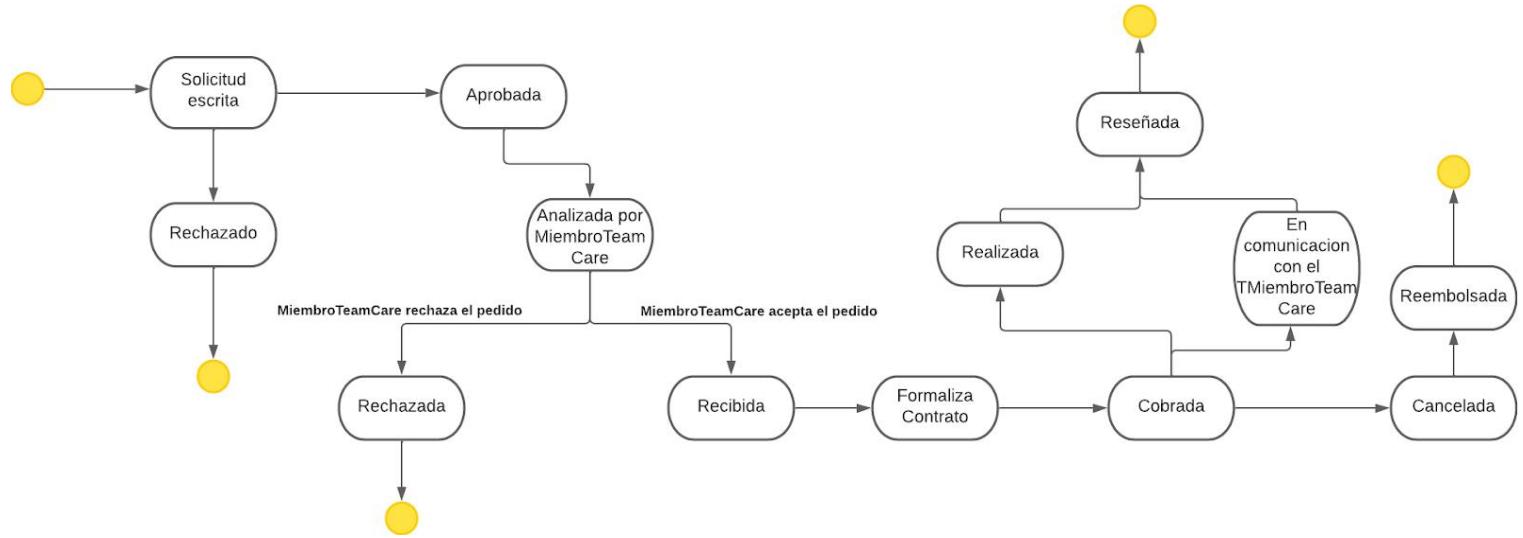
3.3.2.2.2. Diagrama de Objetos



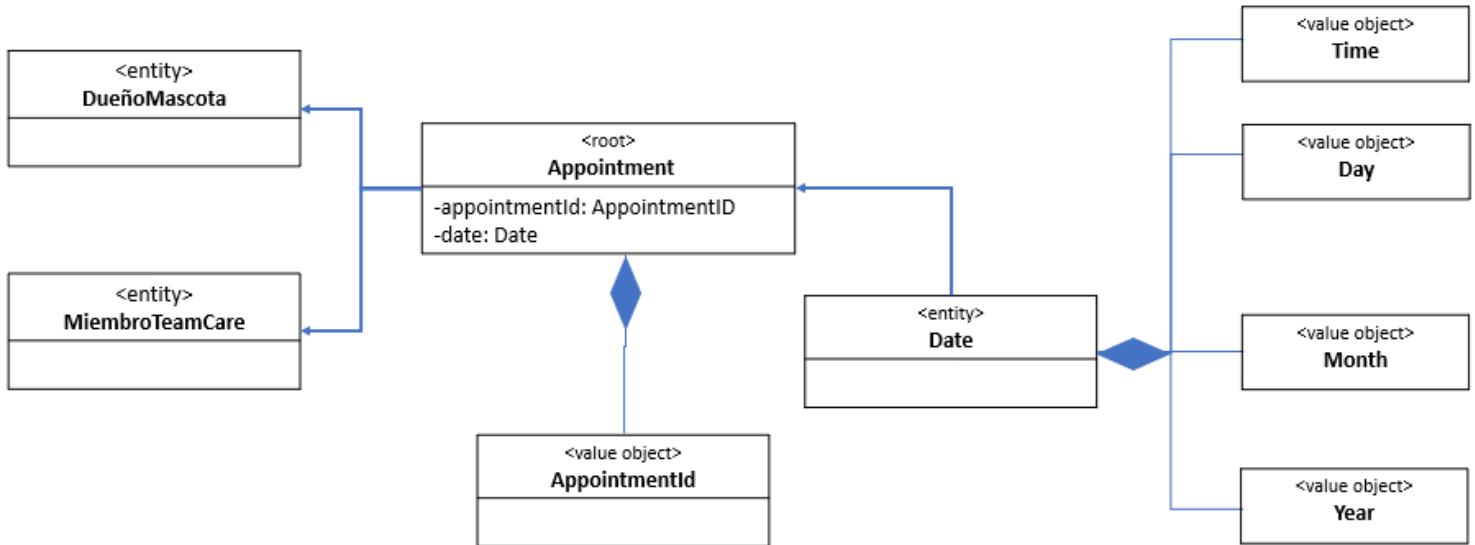
3.3.2.2.3. Diagrama de Secuencia



3.3.2.2.4. Diagrama de Estados

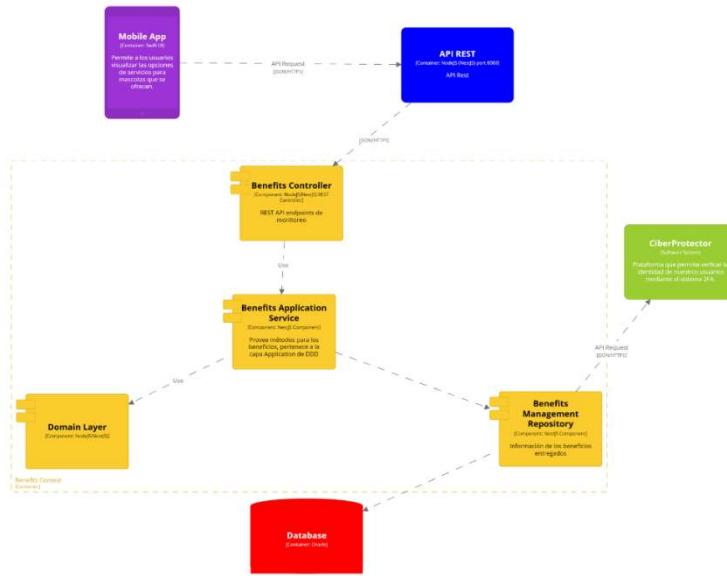


3.3.2.3. Diagrama de Agregados: Appointment



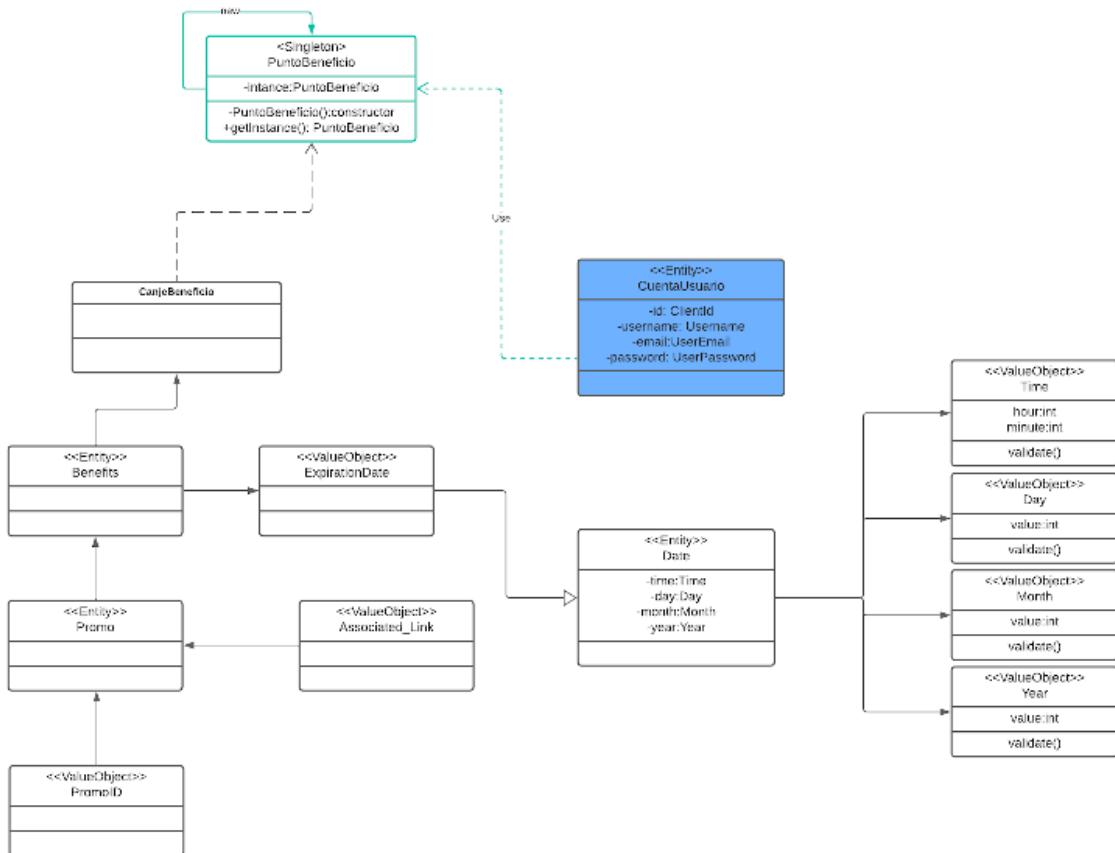
3.3.3. Bounded Context Benefits

3.3.3.1. Diagrama de Componentes

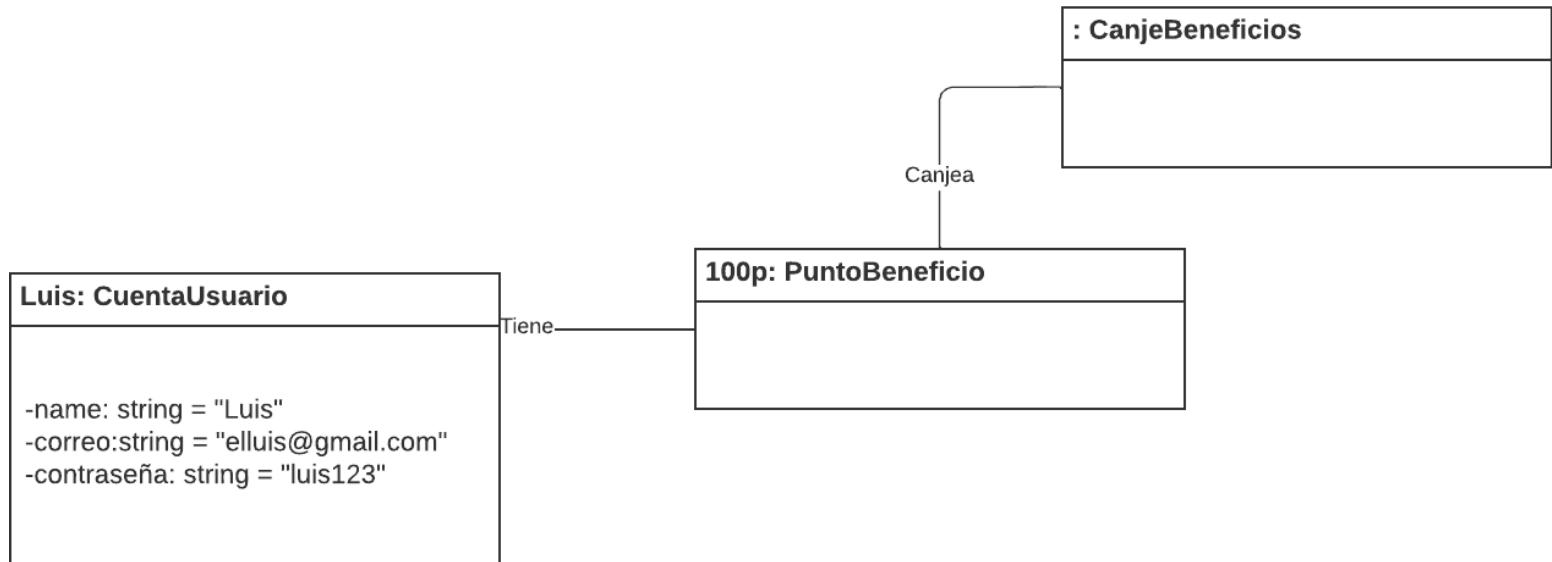


3.3.3.2. Diagramas de código

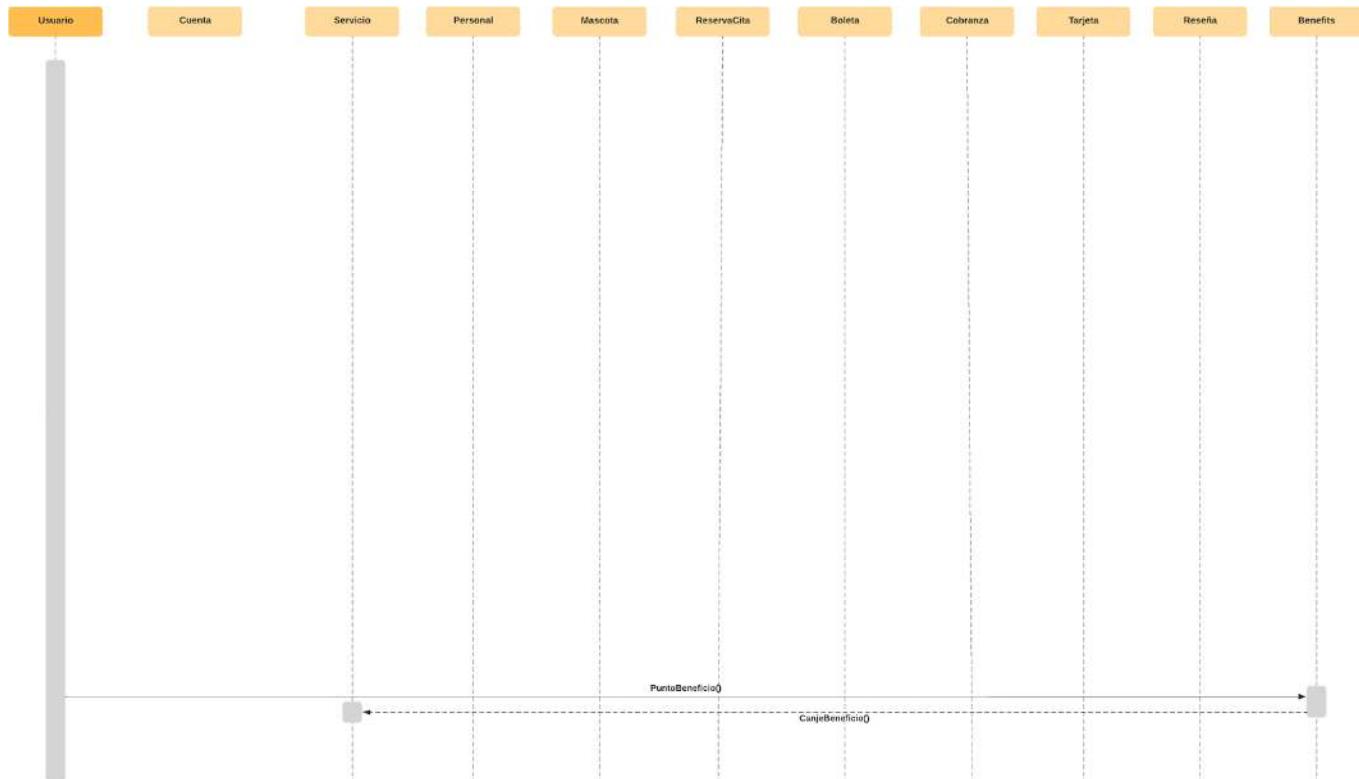
3.3.3.2.1. Diagrama de Clases



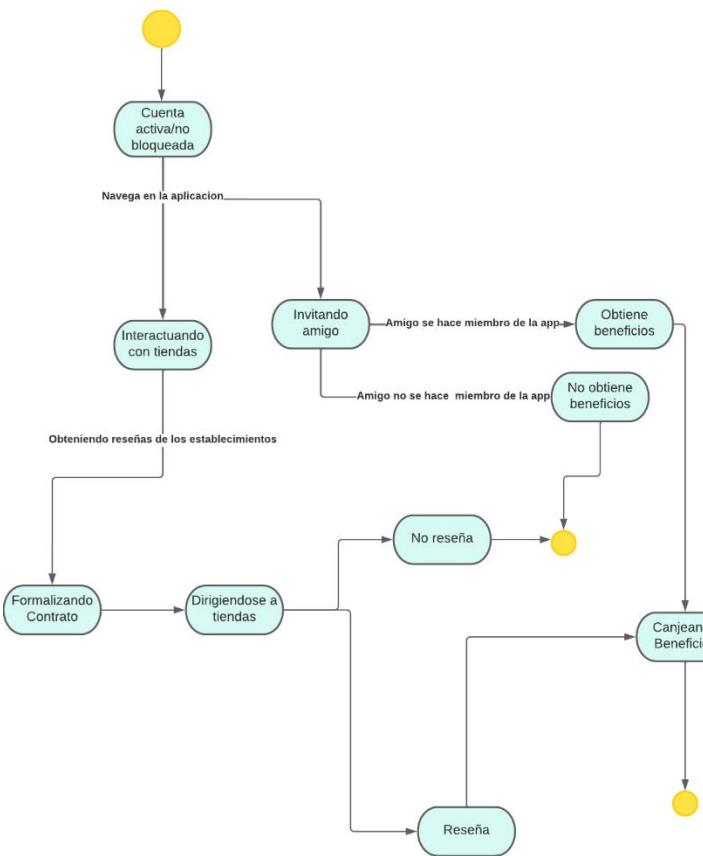
3.3.3.2.2. Diagrama de Objetos



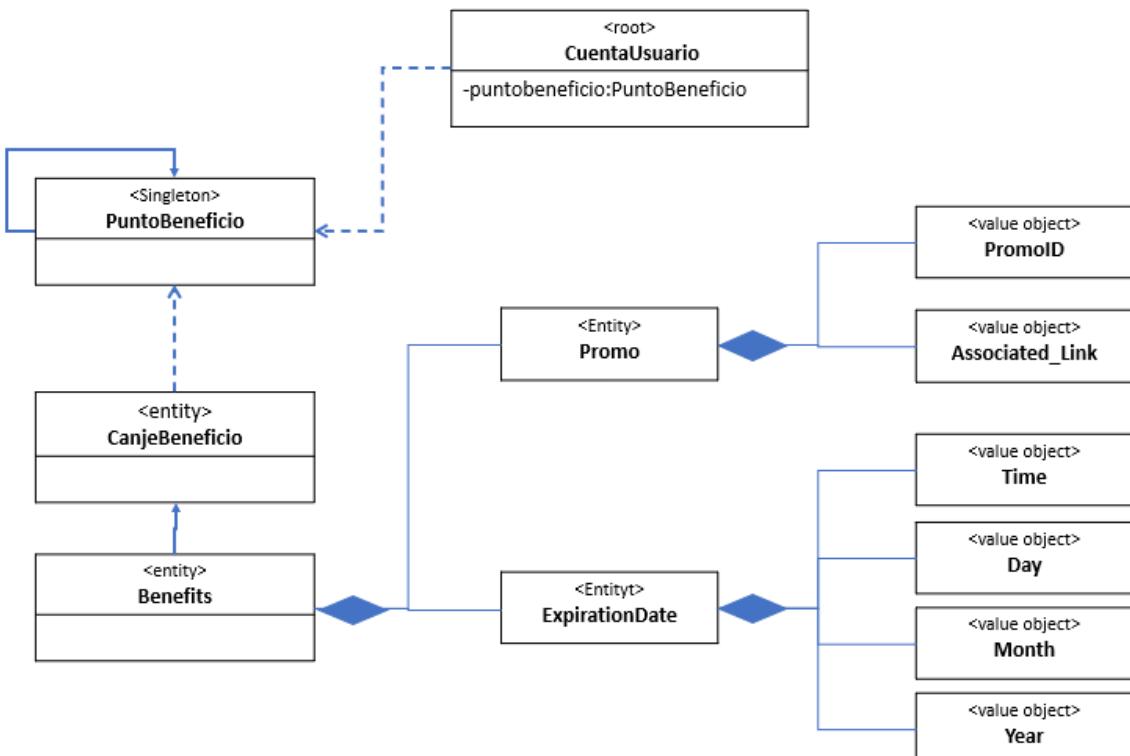
3.3.3.2.3. Diagrama de Secuencia



3.3.3.2.4. Diagrama de Estados

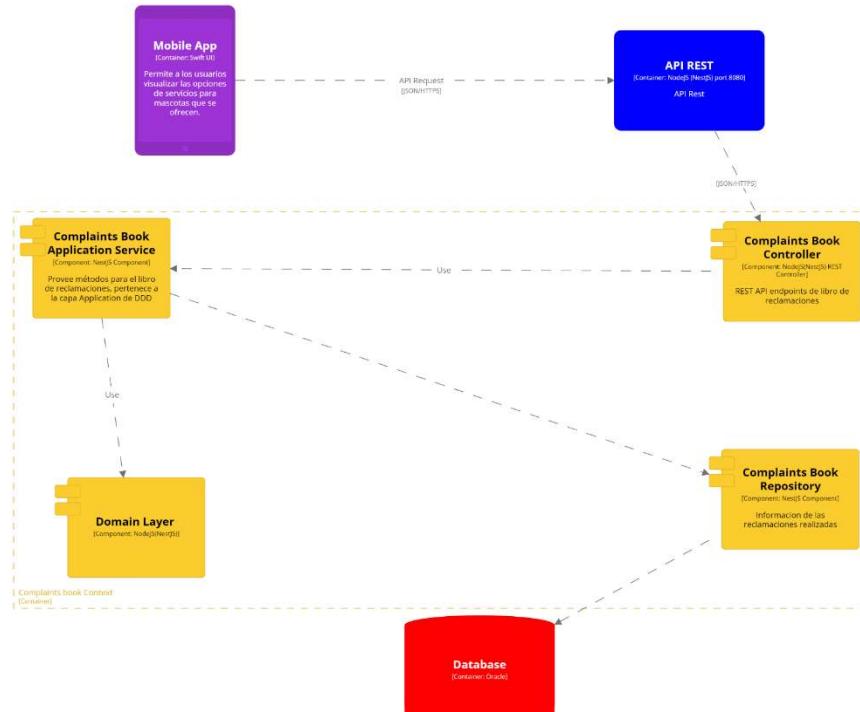


3.3.3.3. Diagrama de Agregados: Benefits



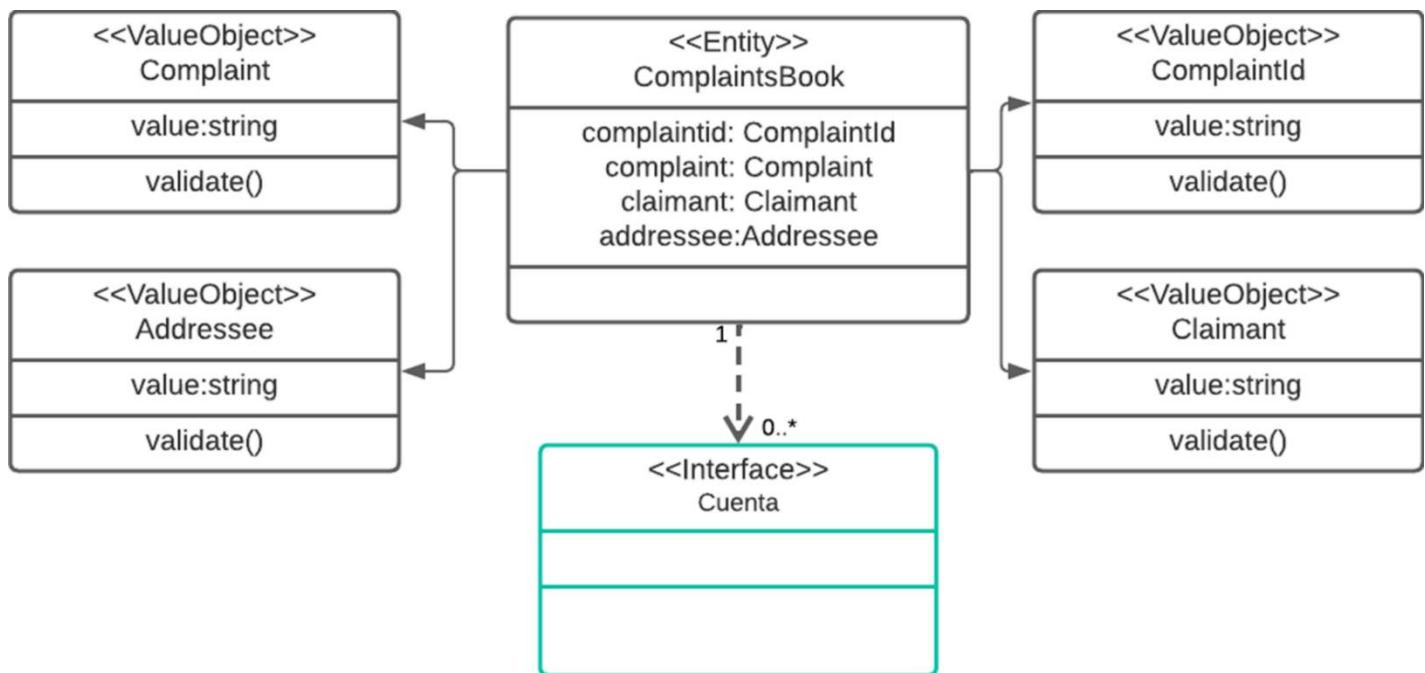
3.3.4. Bounded Context Complaints book

3.3.4.1. Diagrama de Componentes

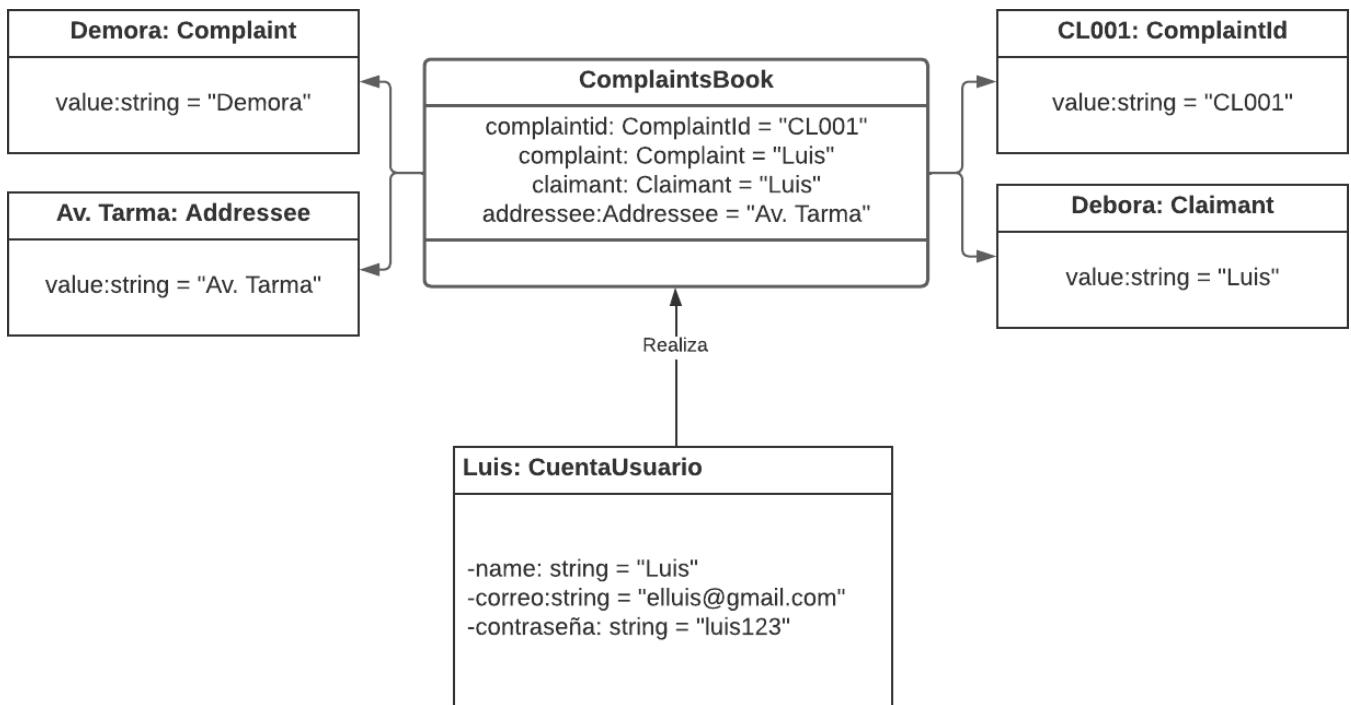


3.3.4.2. Diagramas de código

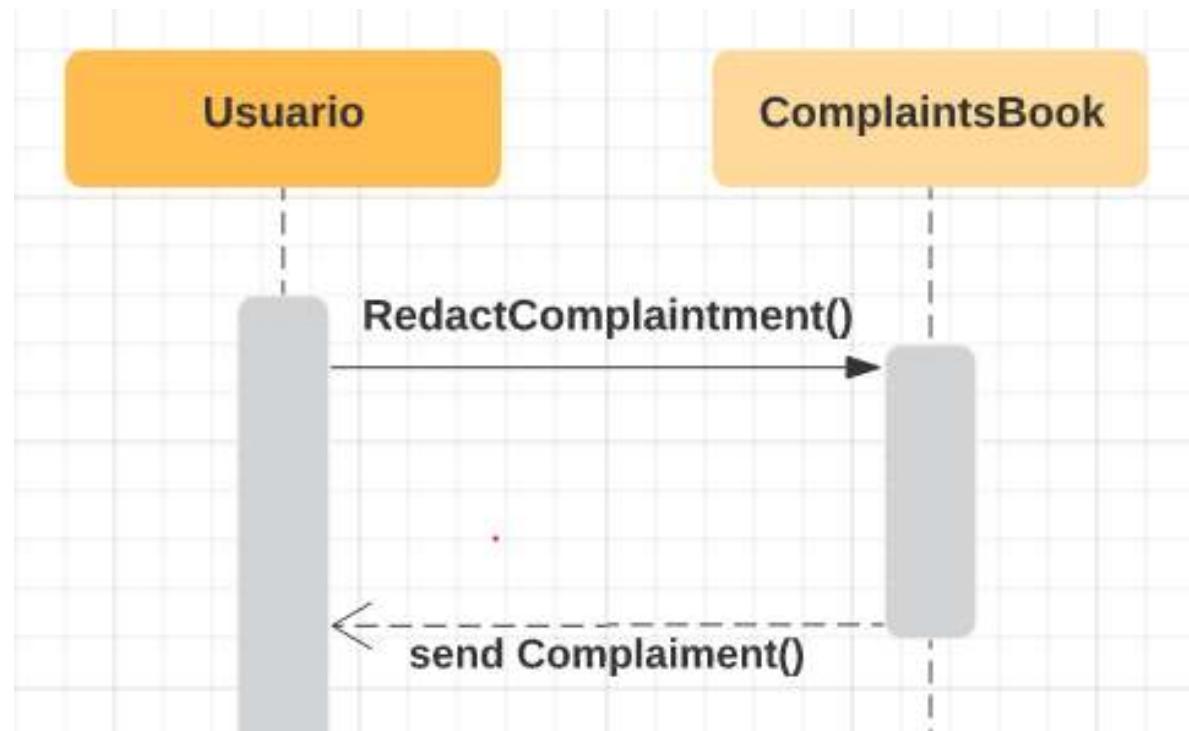
3.3.4.2.1. Diagrama de Clases



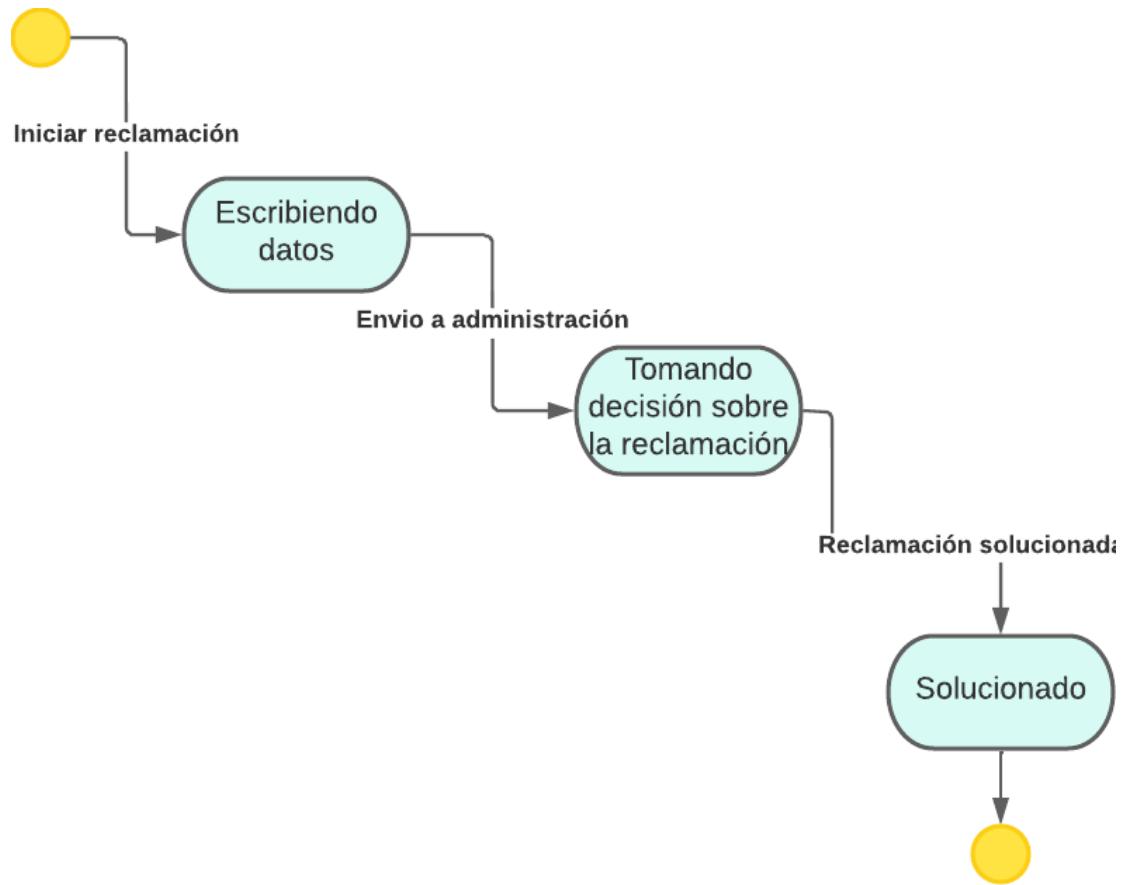
3.3.4.2.2. Diagrama de Objetos



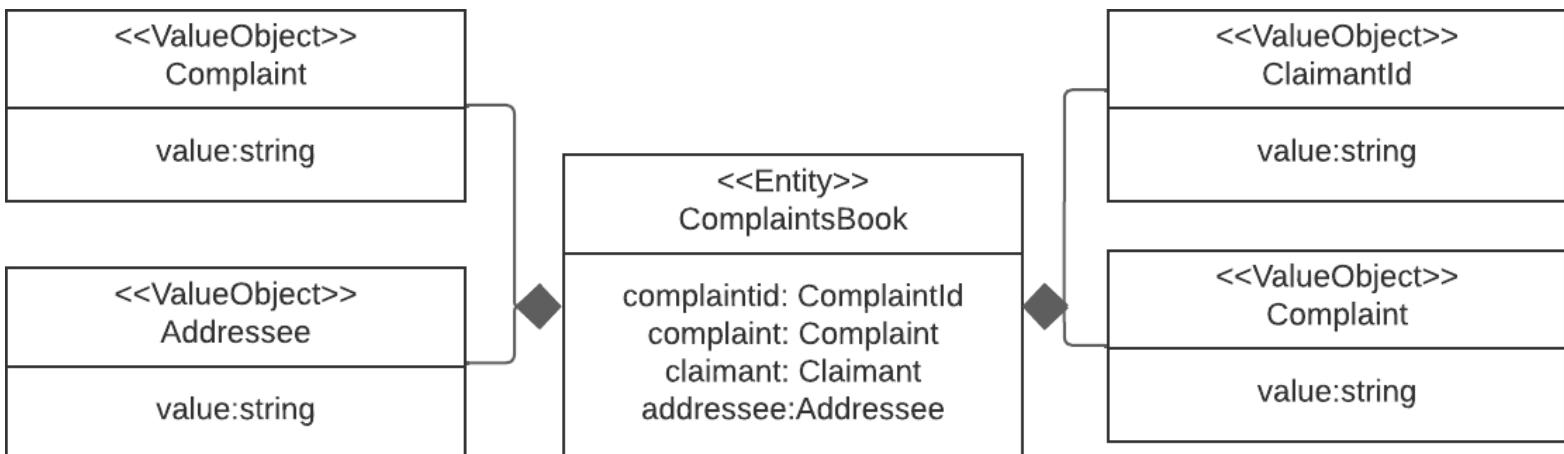
3.3.4.2.3. Diagrama de Secuencia



3.3.4.2.4. Diagrama de Estados

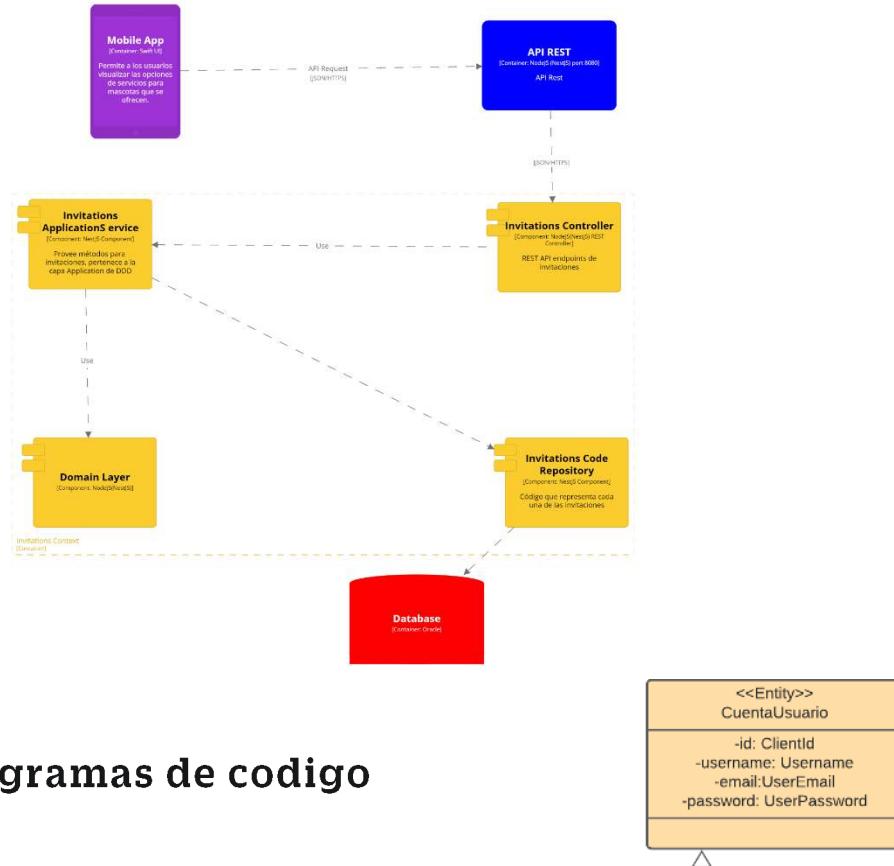


3.3.4.3. Diagrama de Agregados: Complaints book



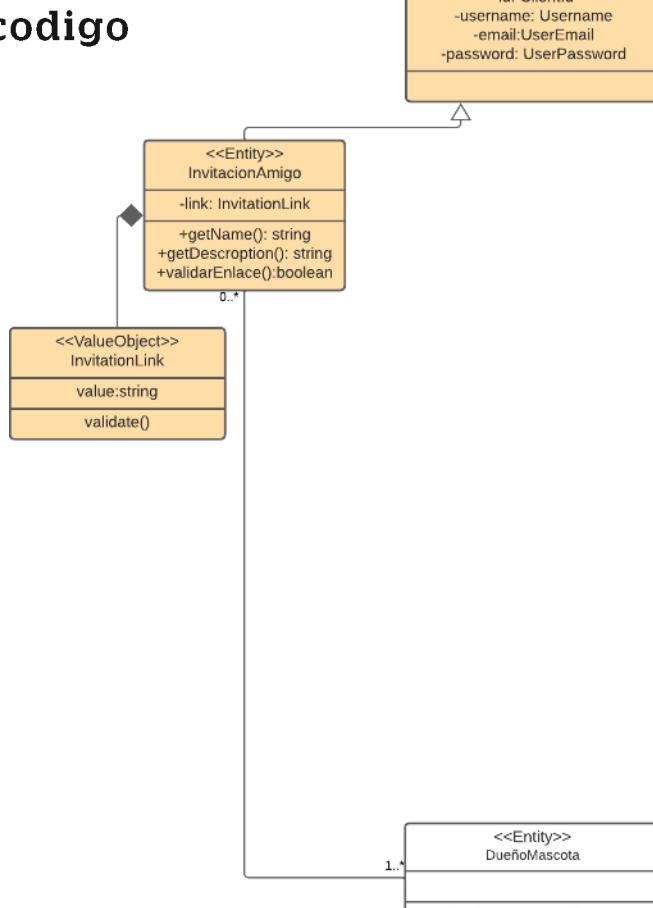
3.3.5. Bounded Context Invitations

3.3.5.1. Diagrama de Componentes

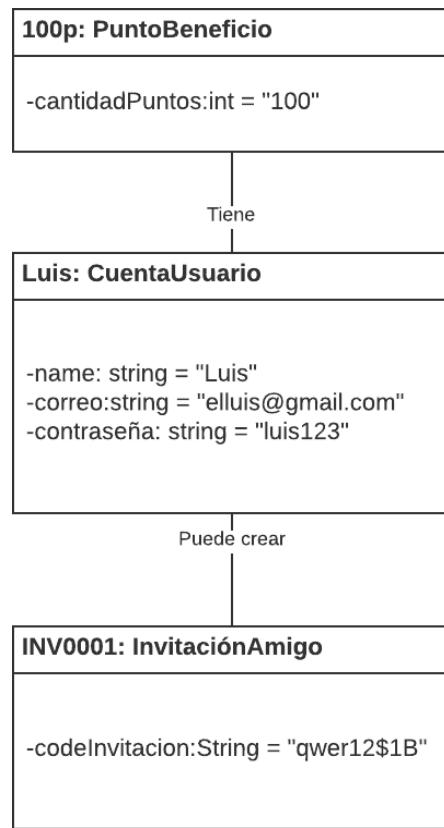


3.3.5.2. Diagramas de código

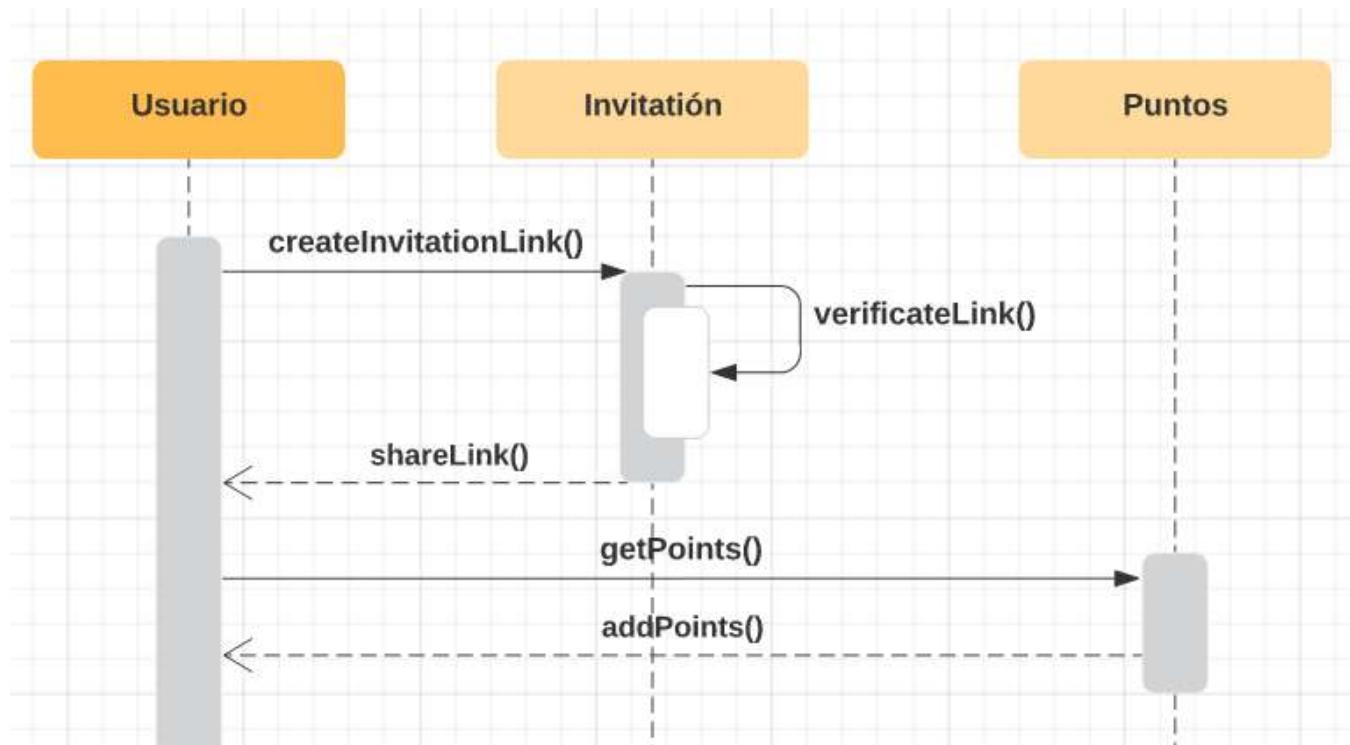
3.3.5.2.1. Diagrama de Clases



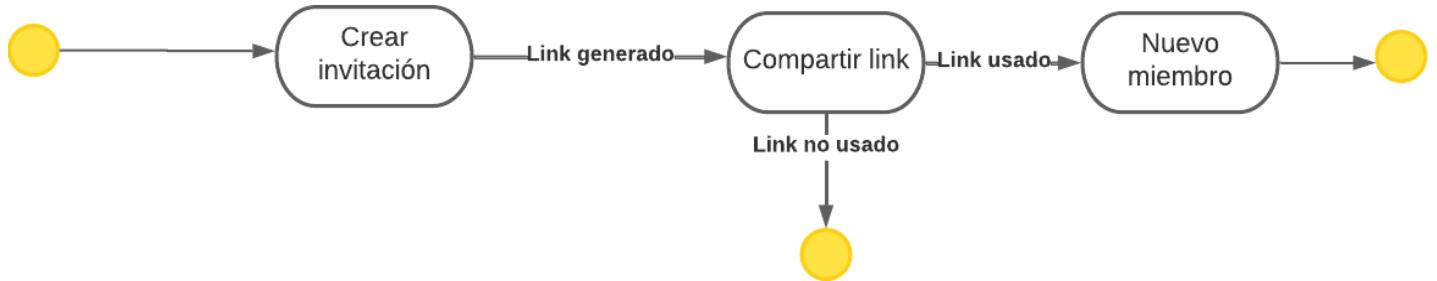
3.3.5.2.2. Diagrama de Objetos



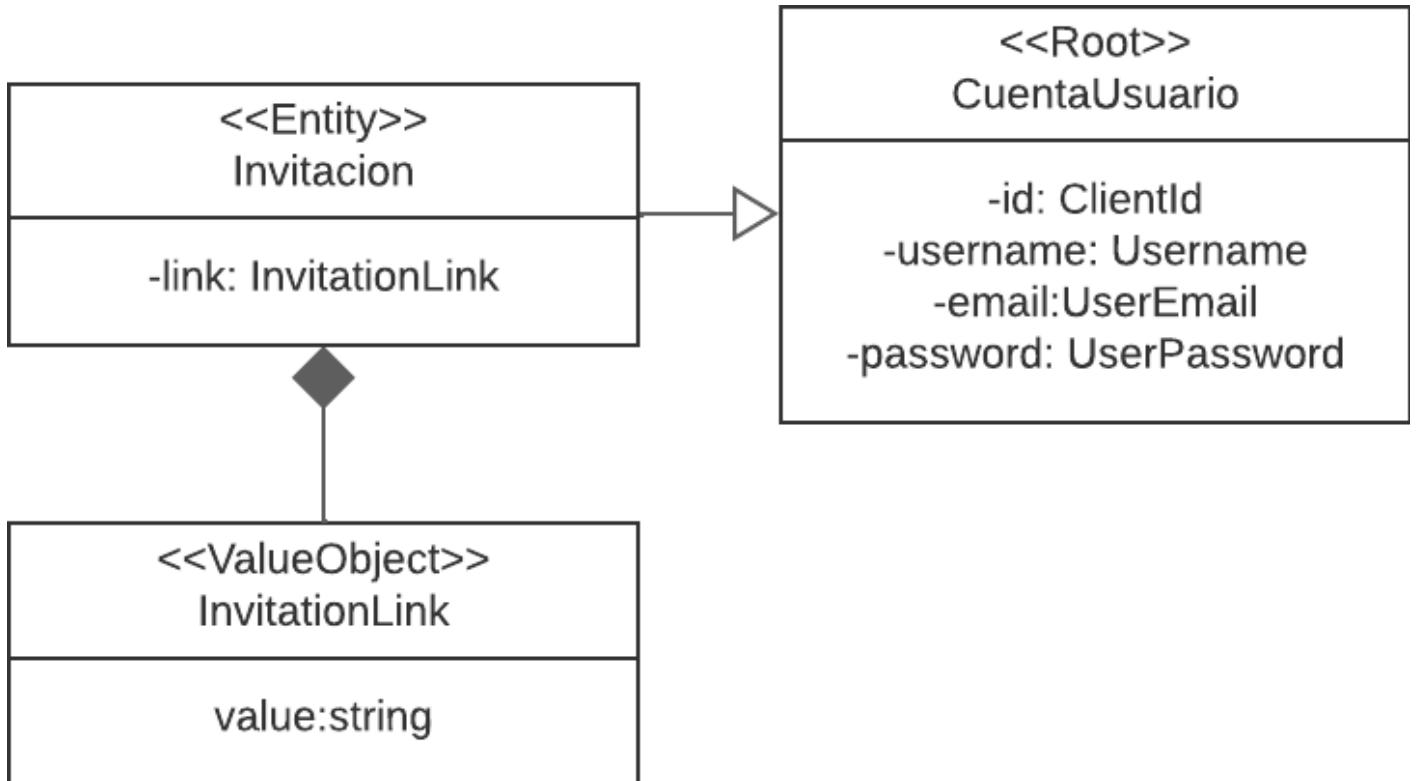
3.3.5.2.3. Diagrama de Secuencia



3.3.5.2.4. Diagrama de Estados

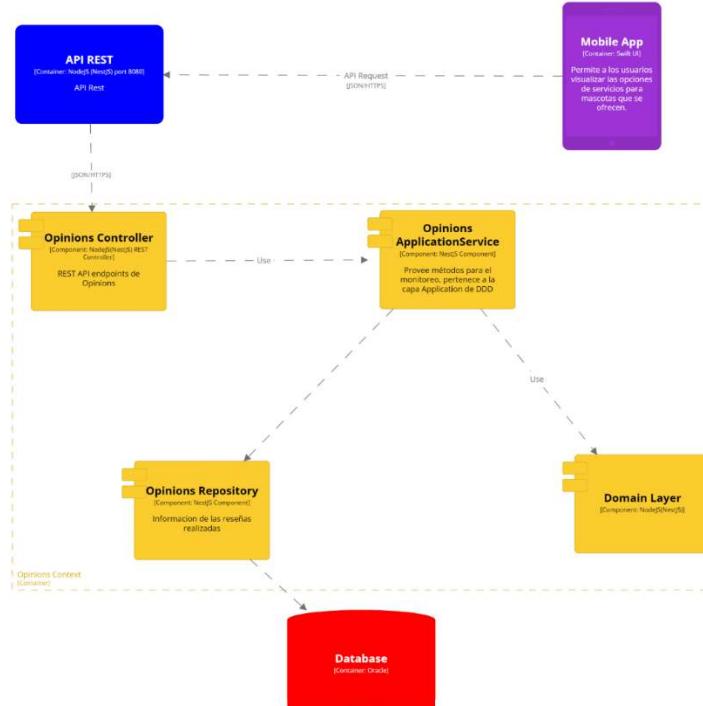


3.3.5.3. Diagrama de Agregados: Invitations



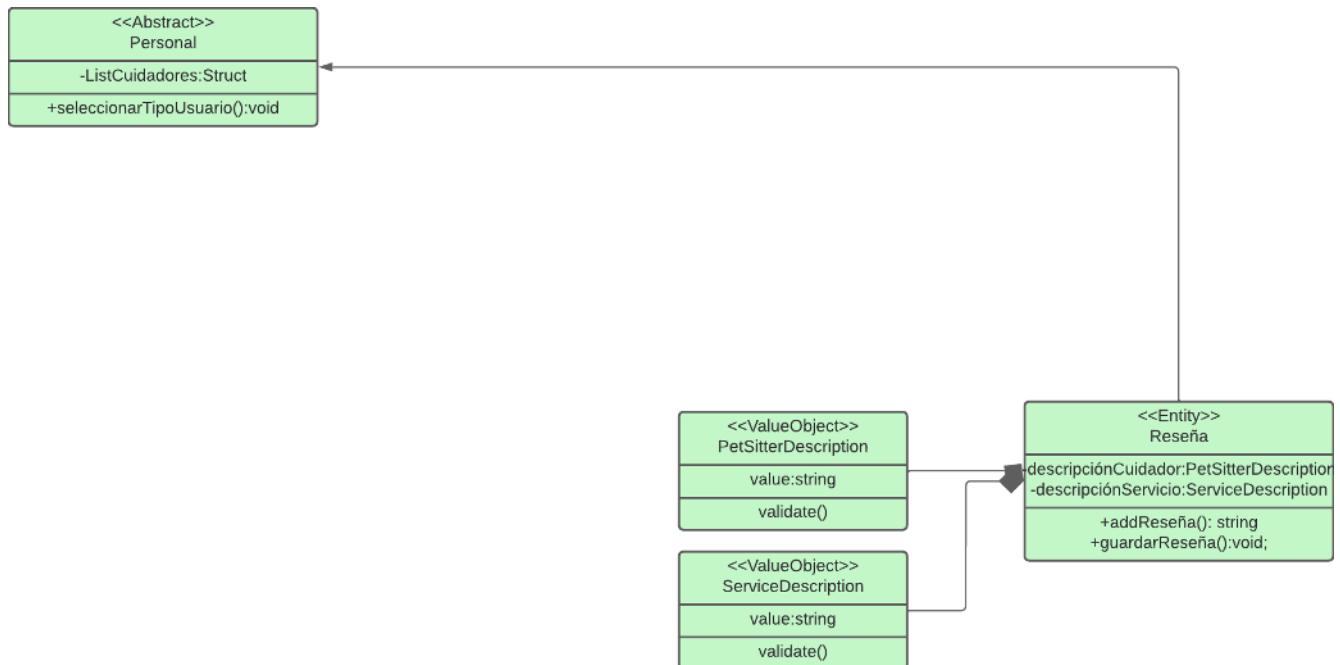
3.3.6. Bounded Context Opinions

3.3.6.1. Diagrama de Componentes

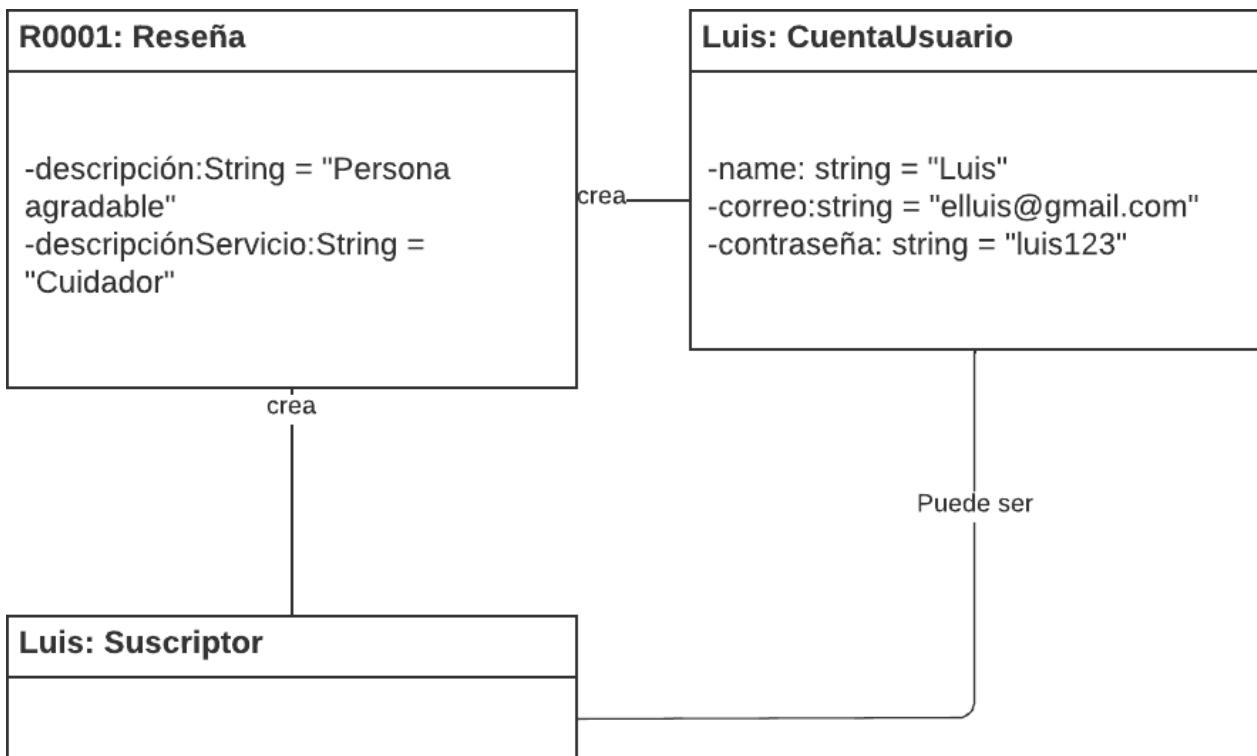


3.3.6.2. Diagramas de código

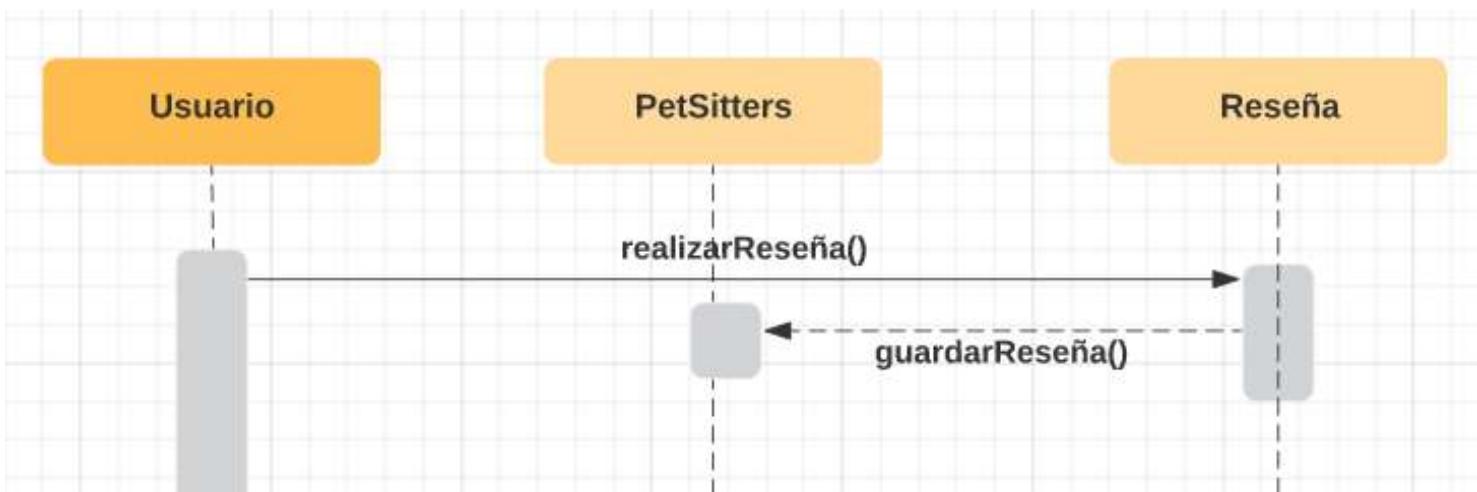
3.3.6.2.1. Diagrama de Clases



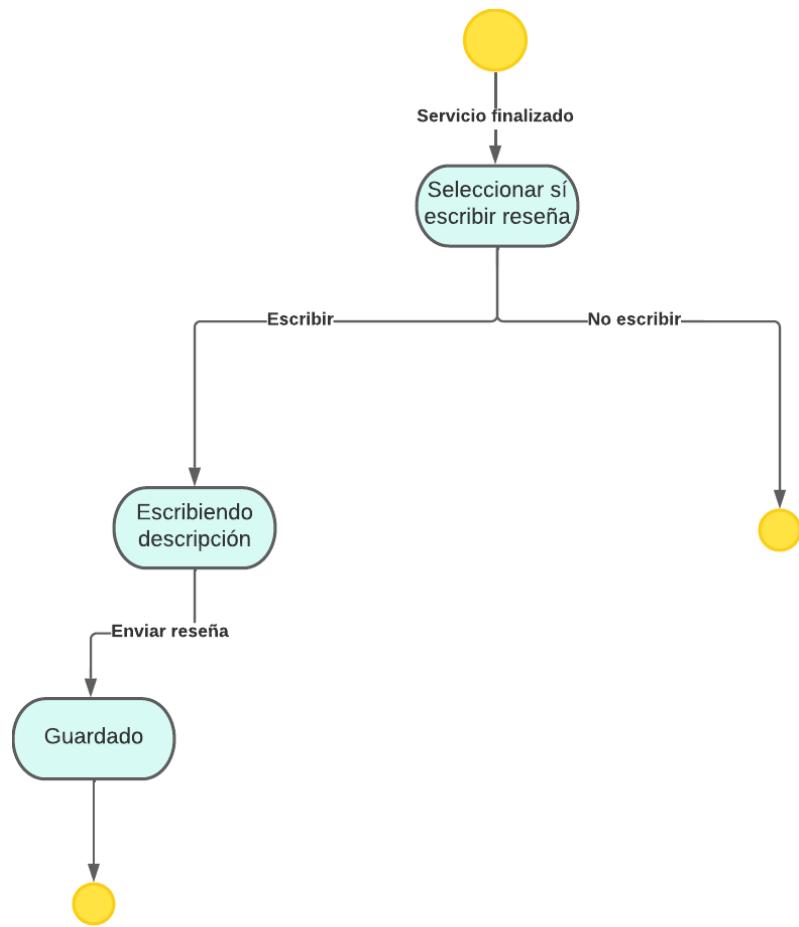
3.3.6.2.2. Diagrama de Objetos



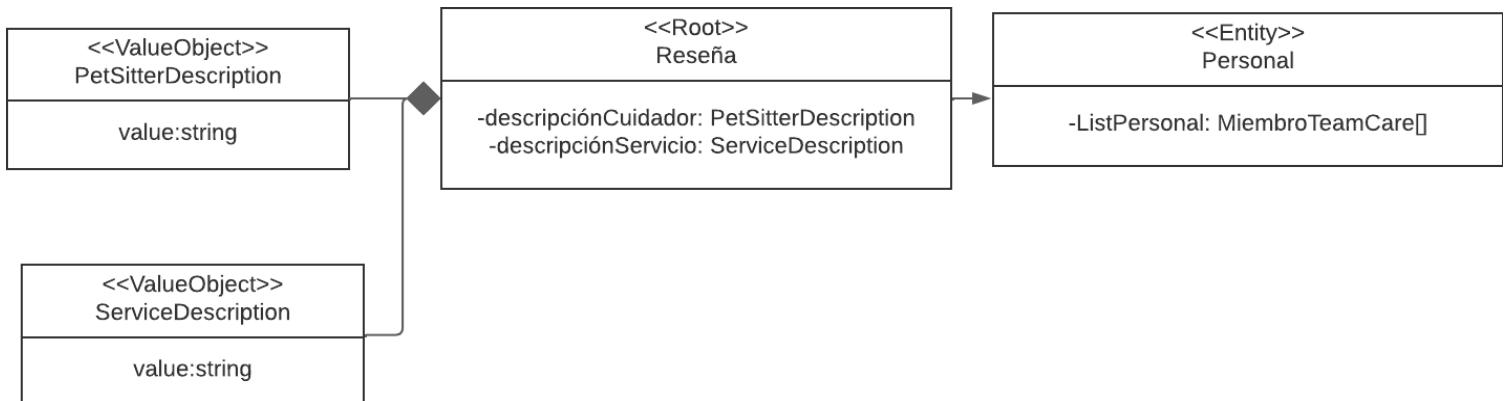
3.3.6.2.3. Diagrama de Secuencia



3.3.6.2.4. Diagrama de Estados

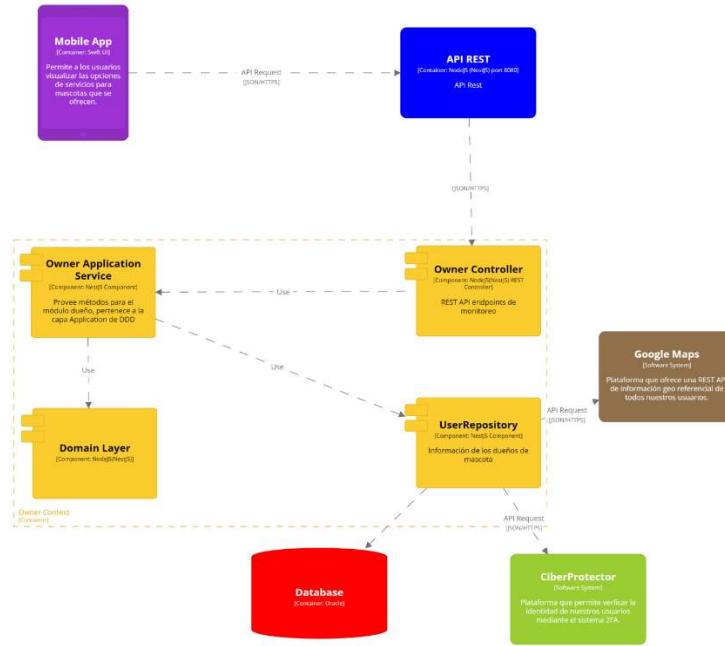


3.3.6.3. Diagrama de Agregados: Opinions



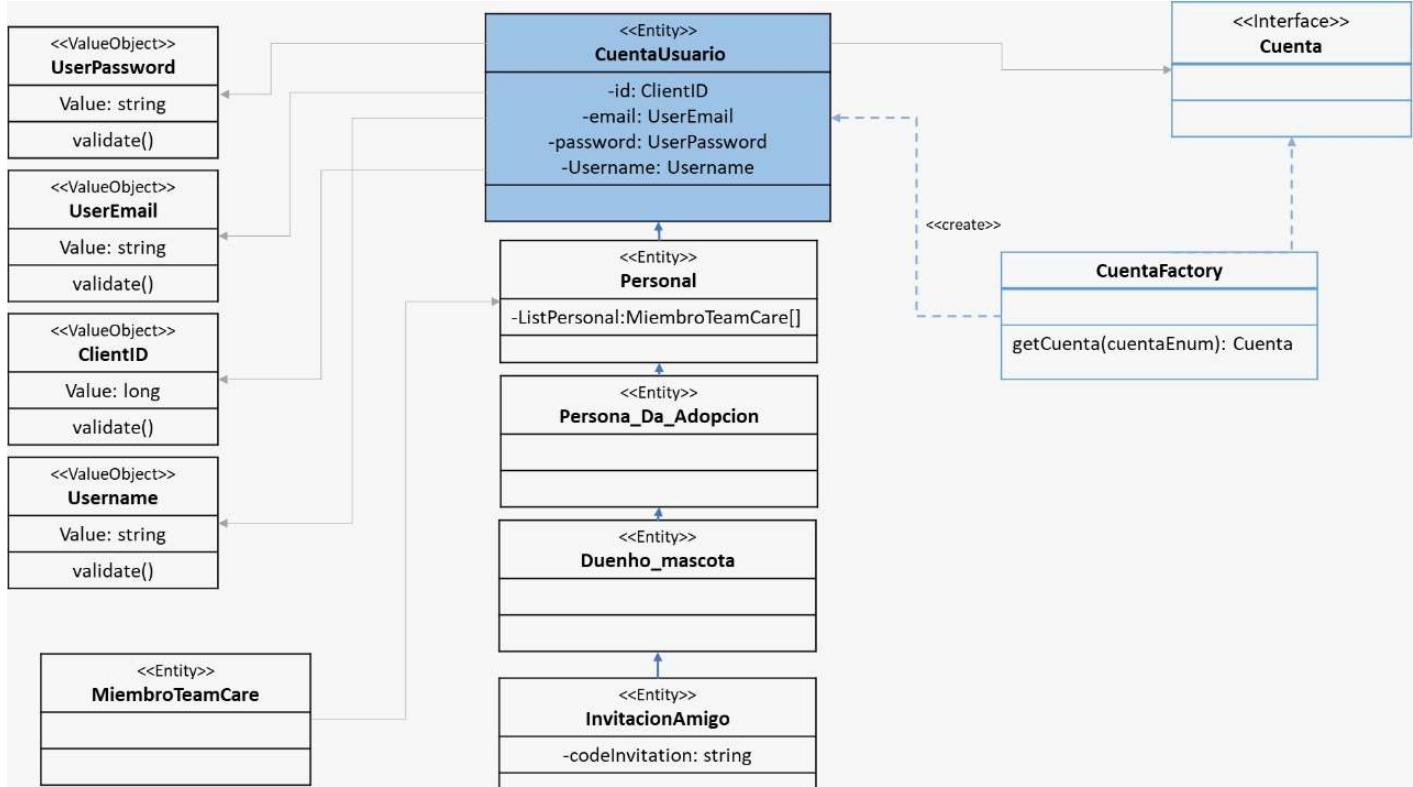
3.3.7. Bounded Context Owner

3.3.7.1. Diagrama de Componentes

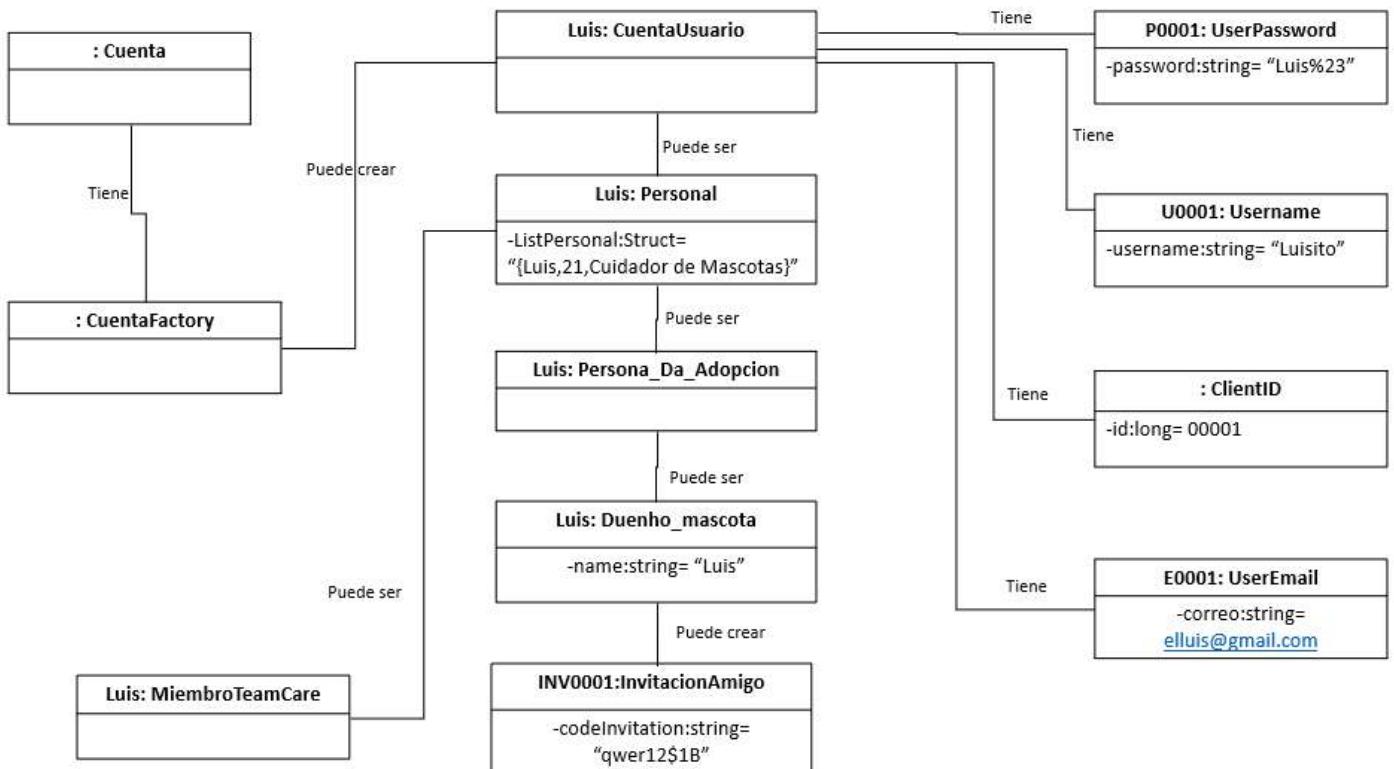


3.3.7.2. Diagramas de código

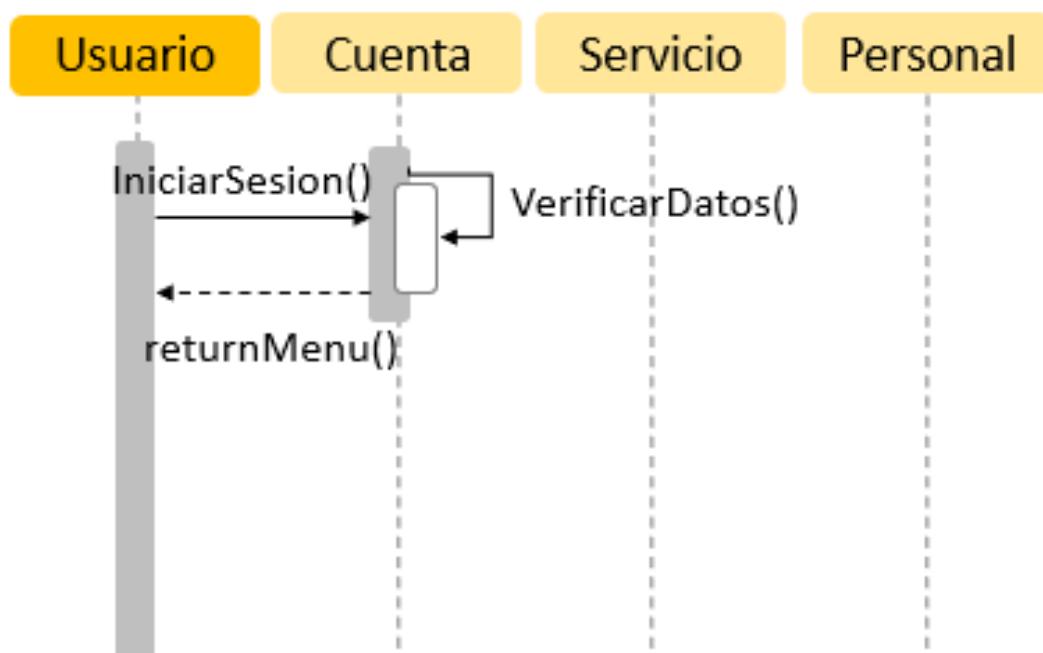
3.3.7.2.1. Diagrama de Clases



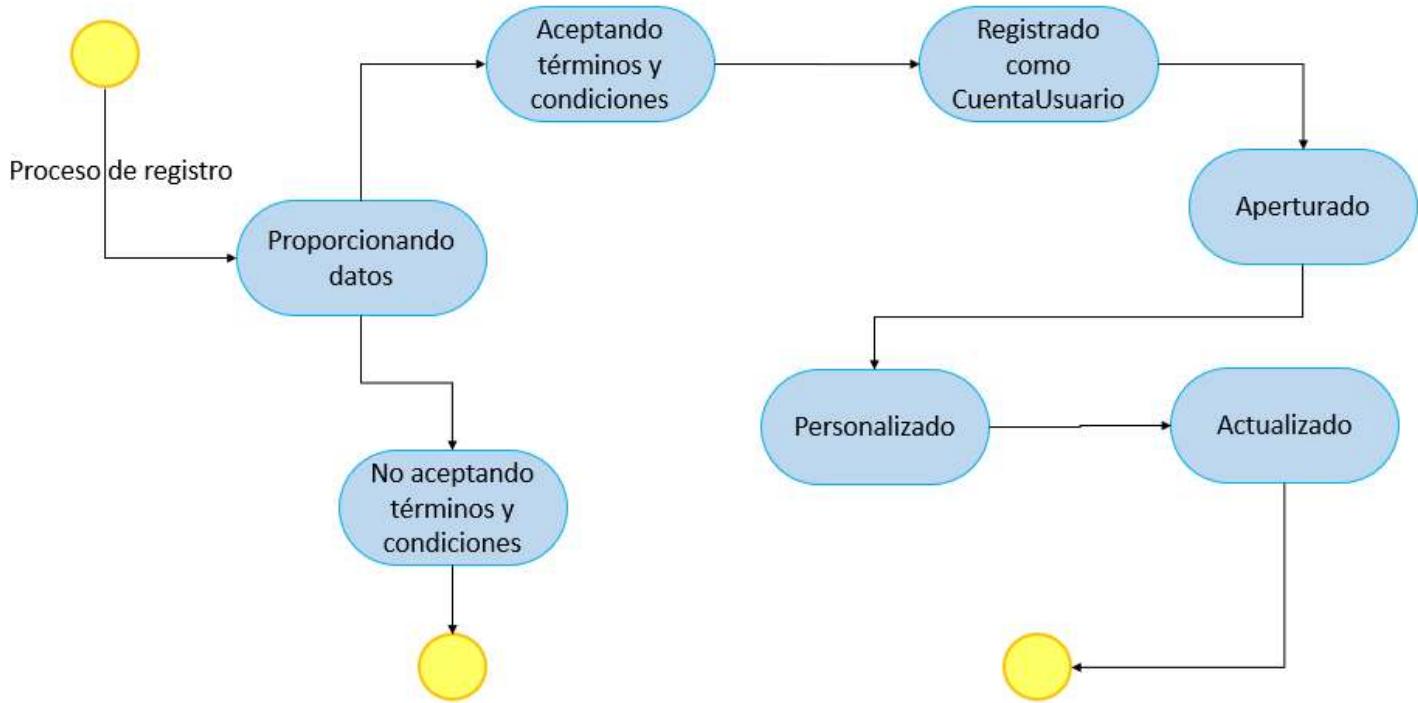
3.3.7.2.2. Diagrama de Objetos



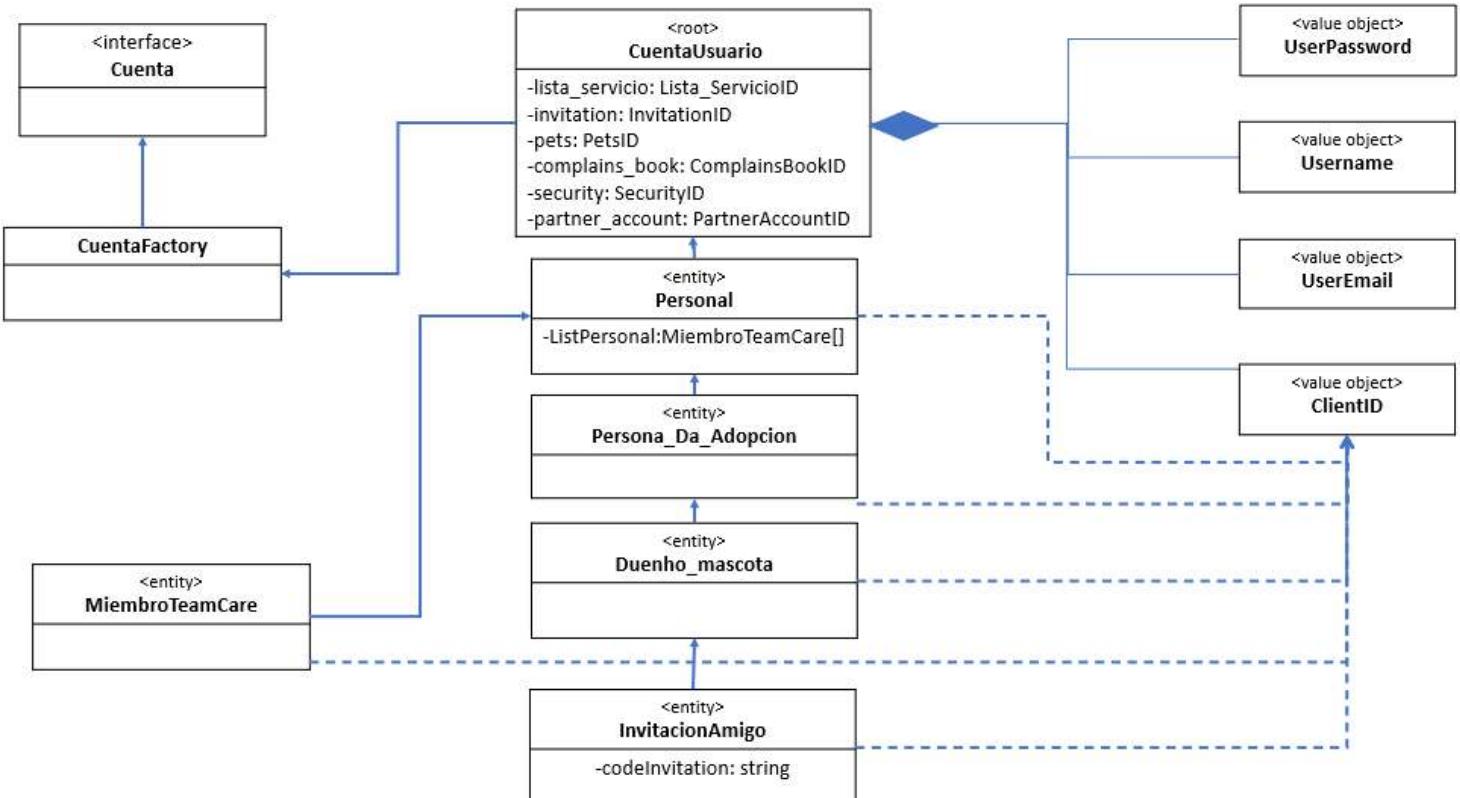
3.3.7.2.3. Diagrama de Secuencia



3.3.7.2.4. Diagrama de Estados

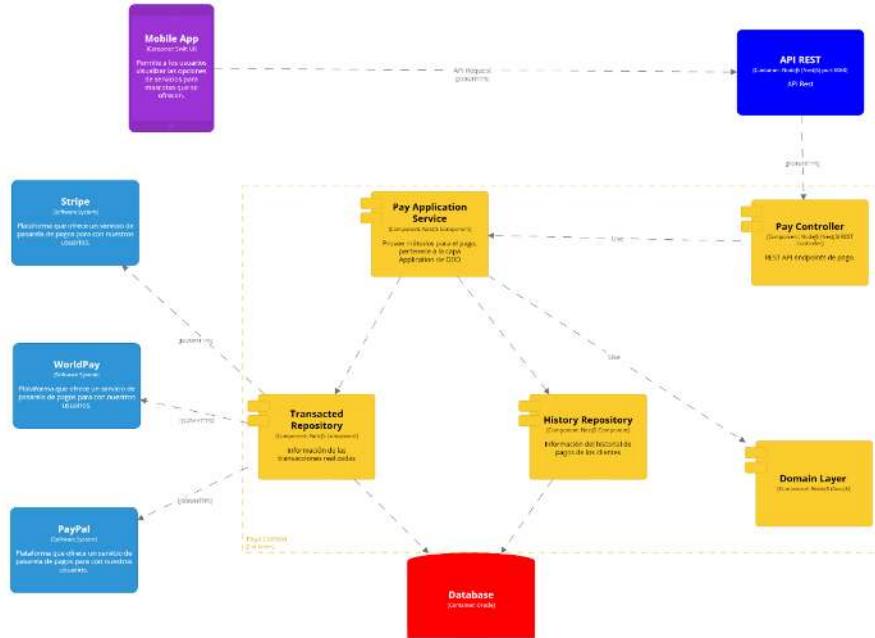


3.3.7.3. Diagrama de Agregados: Owner



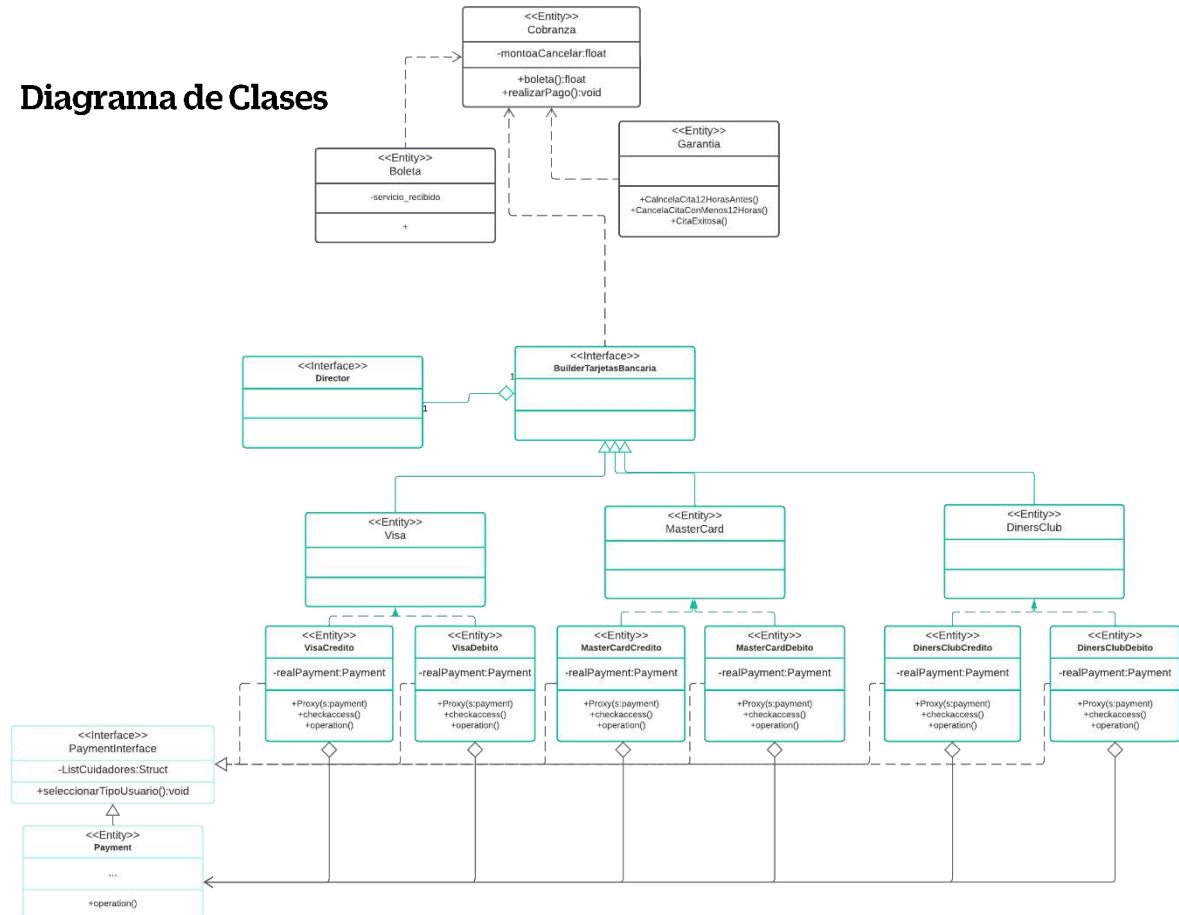
3.3.8. Bounded Context Pays

3.3.8.1. Diagrama de Componentes

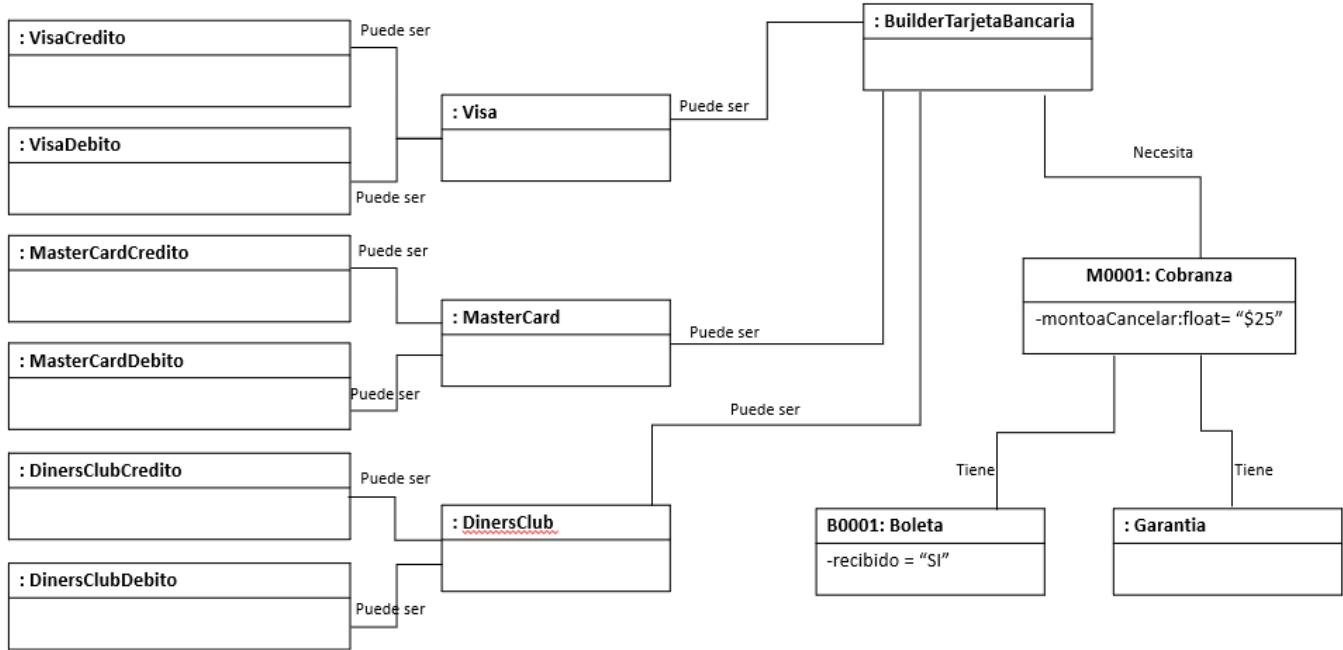


3.3.8.2. Diagramas de código

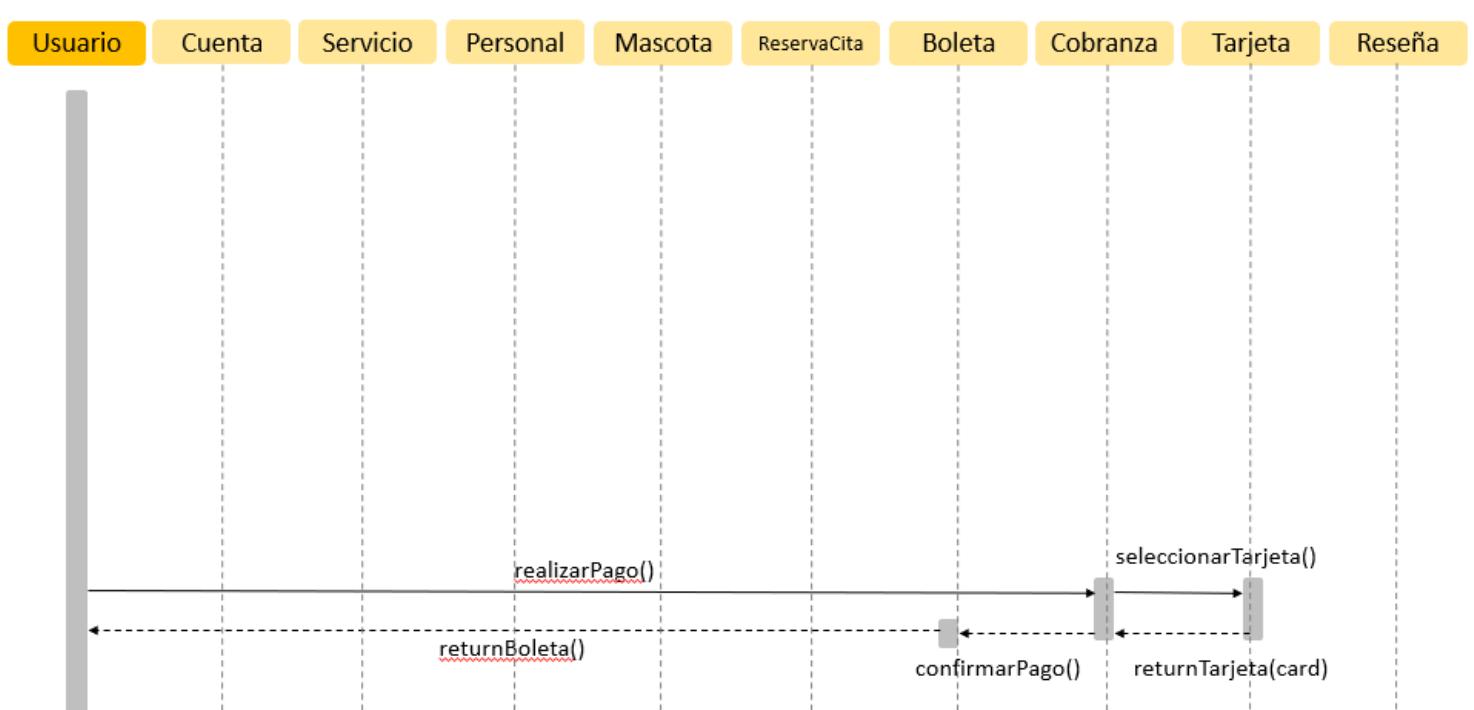
3.3.8.2.1. Diagrama de Clases



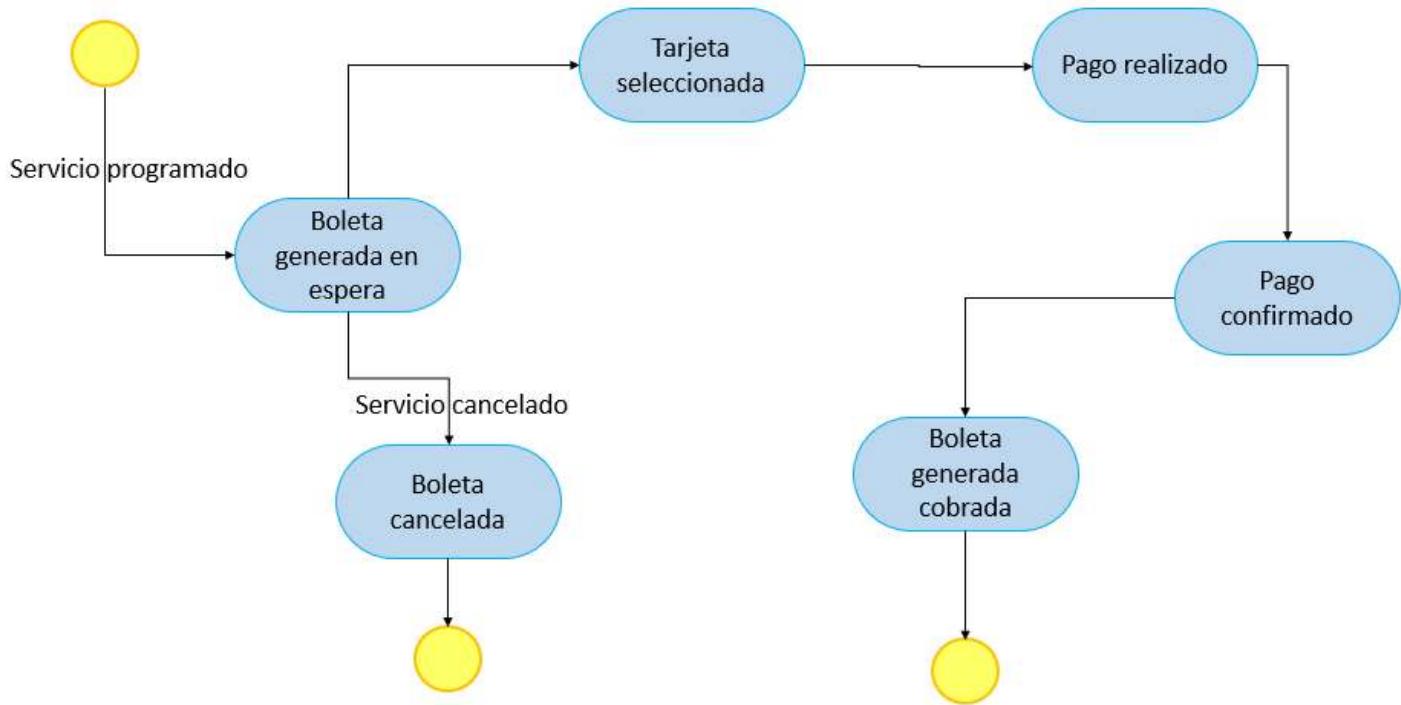
3.3.8.2.2. Diagrama de Objetos



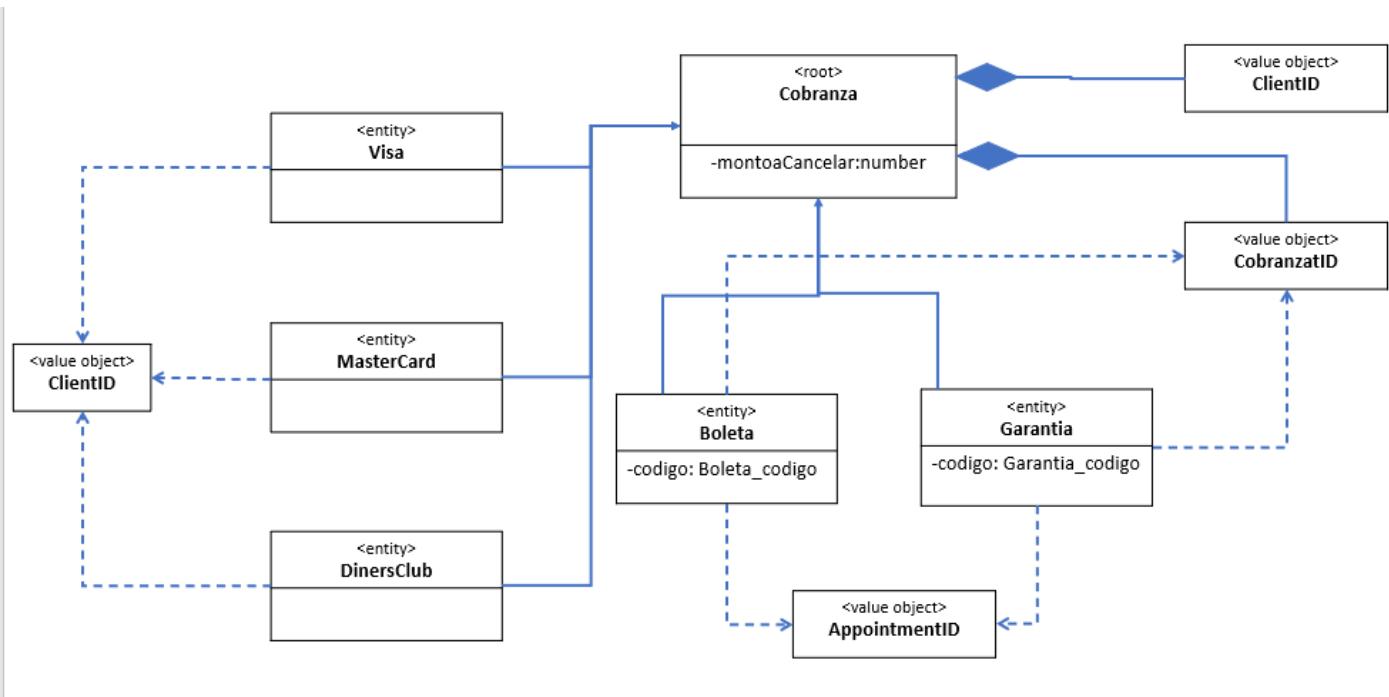
3.3.8.2.3. Diagrama de Secuencia



3.3.8.2.4. Diagrama de Estados

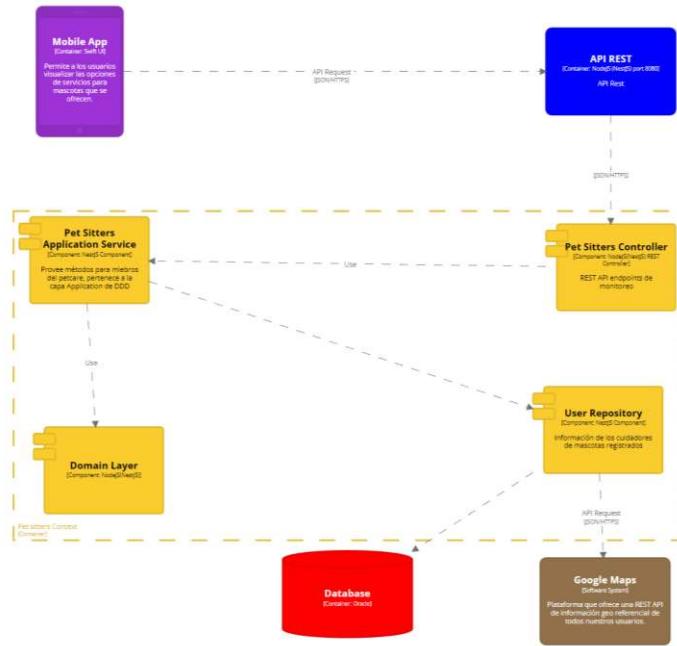


3.3.8.3. Diagrama de Agregados: Pays



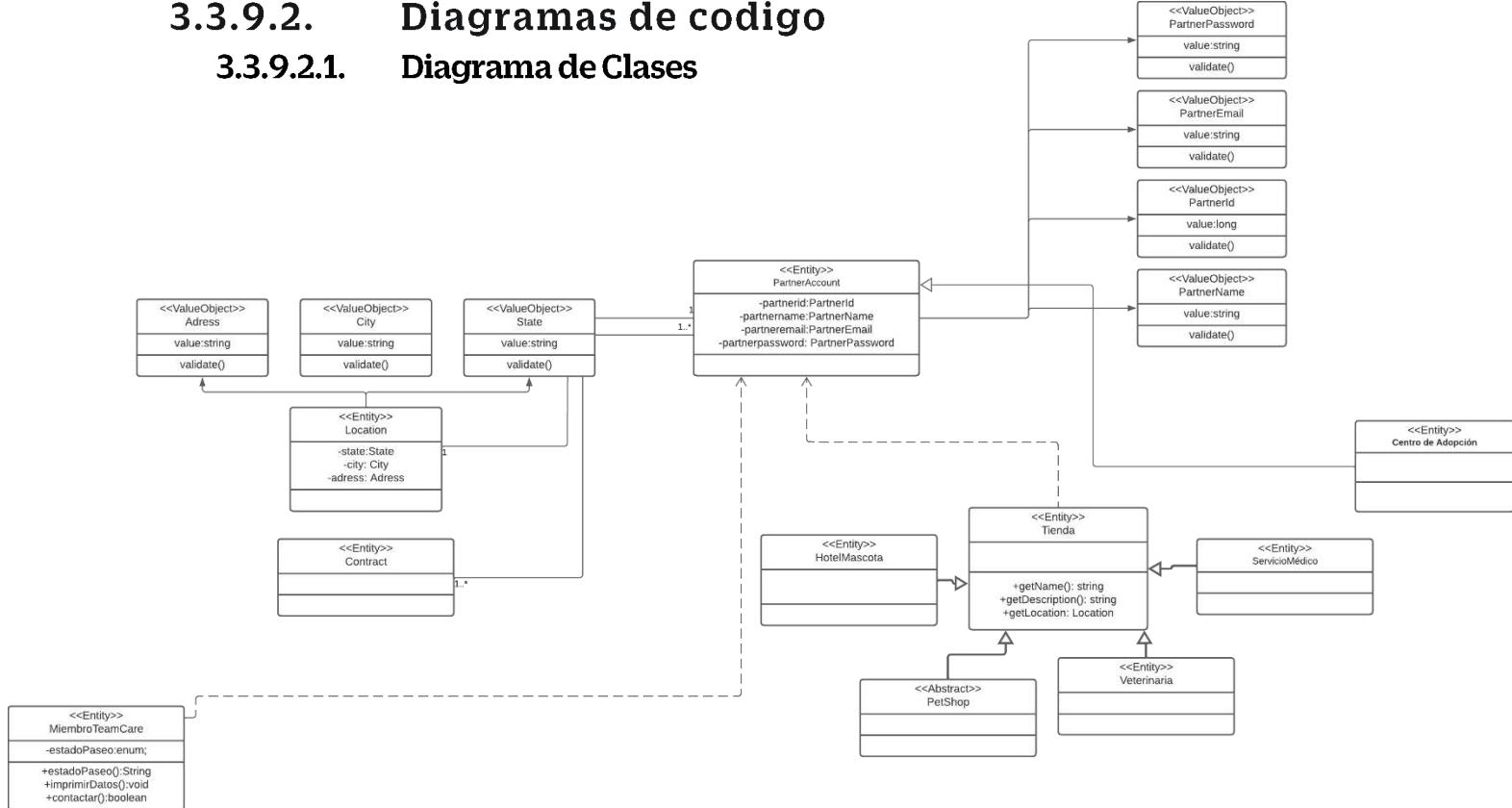
3.3.9. Bounded Context Pet Sitters

3.3.9.1. Diagrama de Componentes

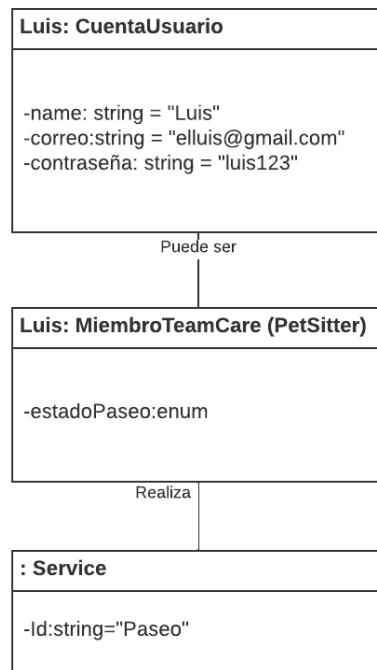


3.3.9.2. Diagramas de código

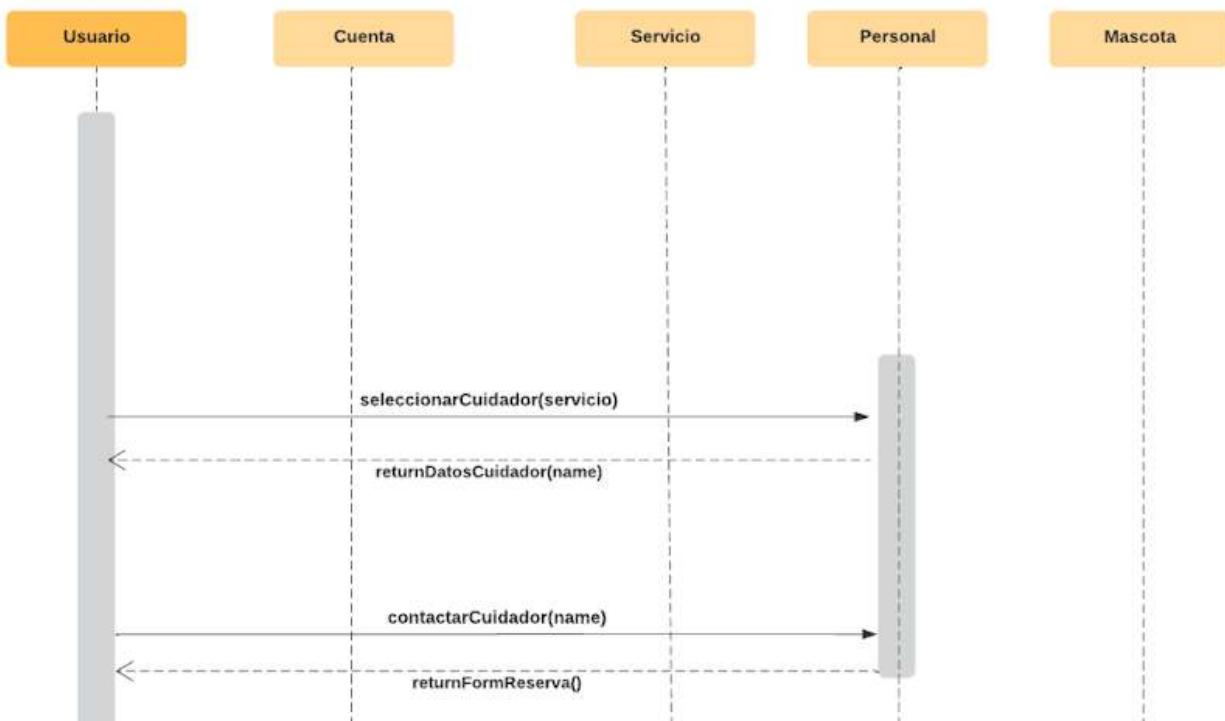
3.3.9.2.1. Diagrama de Clases



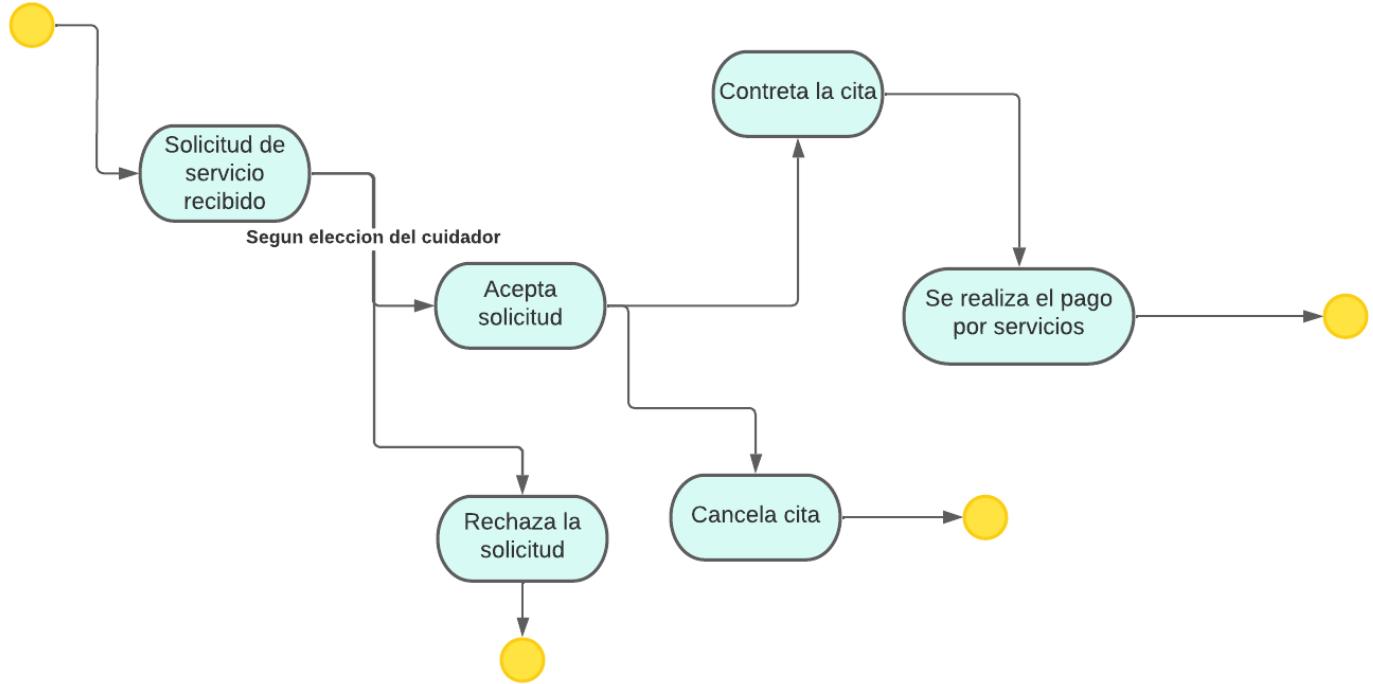
3.3.9.2.2. Diagrama de Objetos



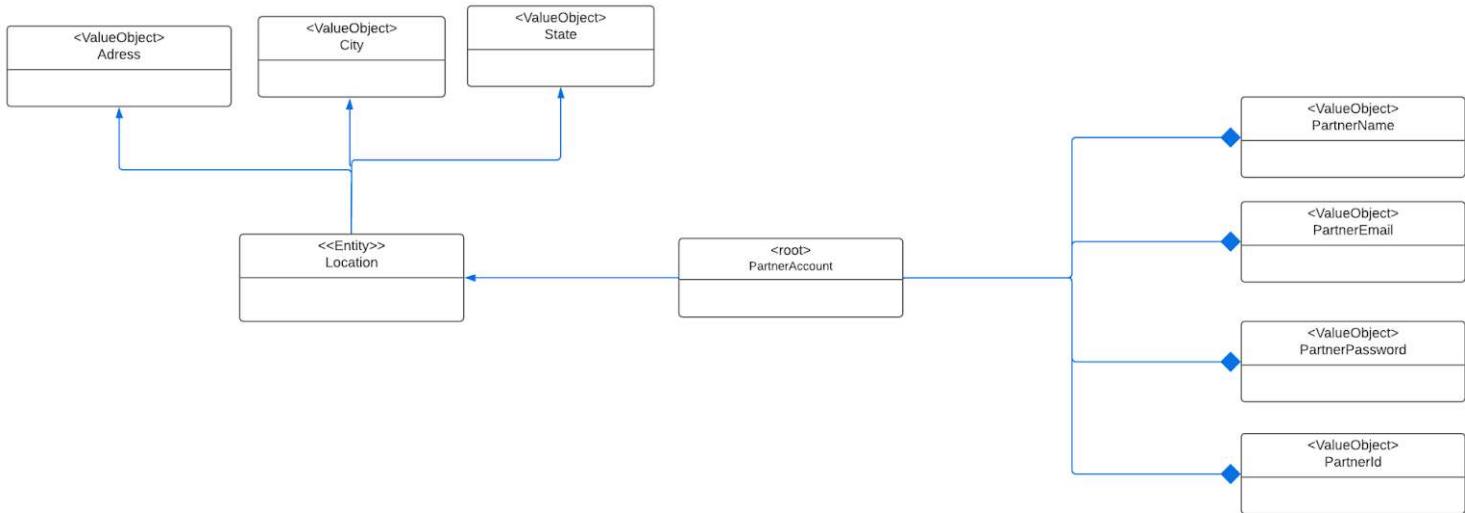
3.3.9.2.3. Diagrama de Secuencia



3.3.9.2.4. Diagrama de Estados

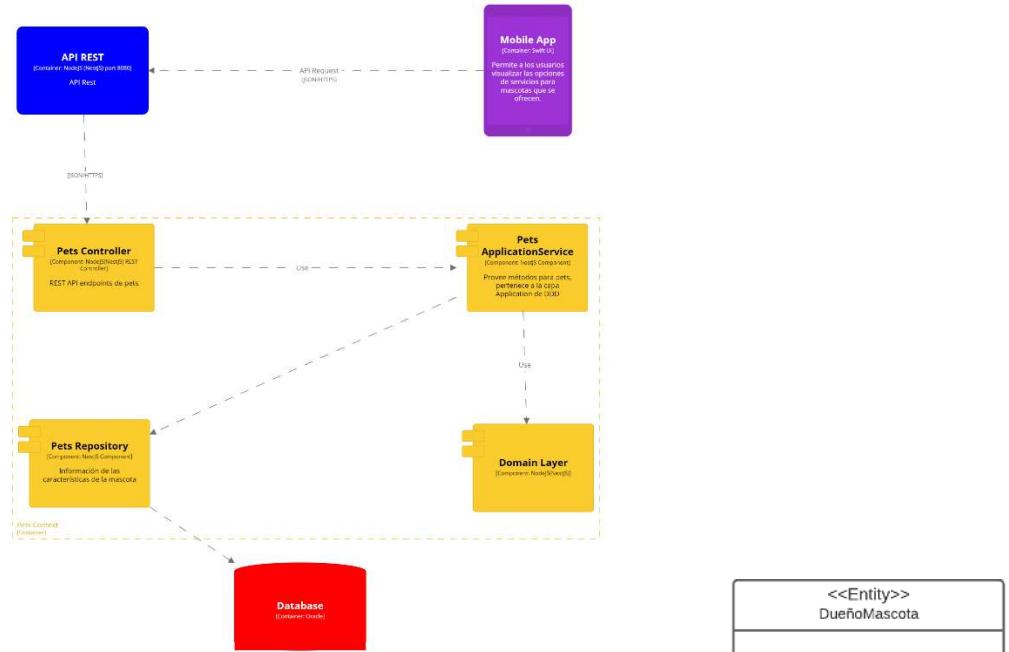


3.3.9.3. Diagrama de Agregados: Pet Sitters



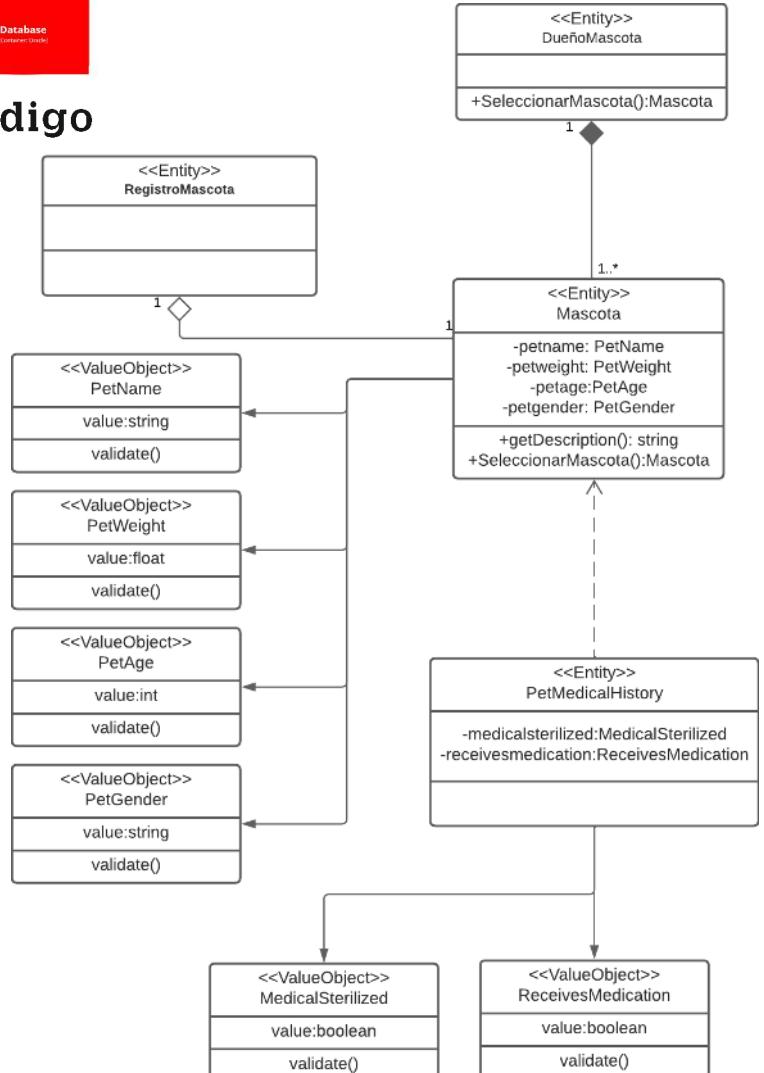
3.3.10. Bounded Context Pets

3.3.10.1. Diagrama de Componentes

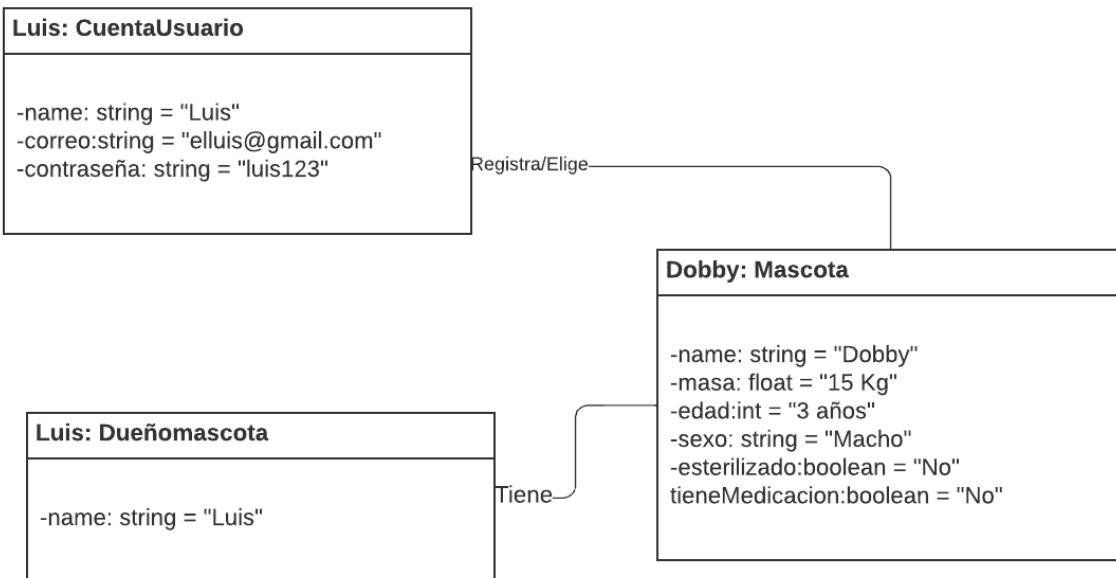


3.3.10.2. Diagramas de código

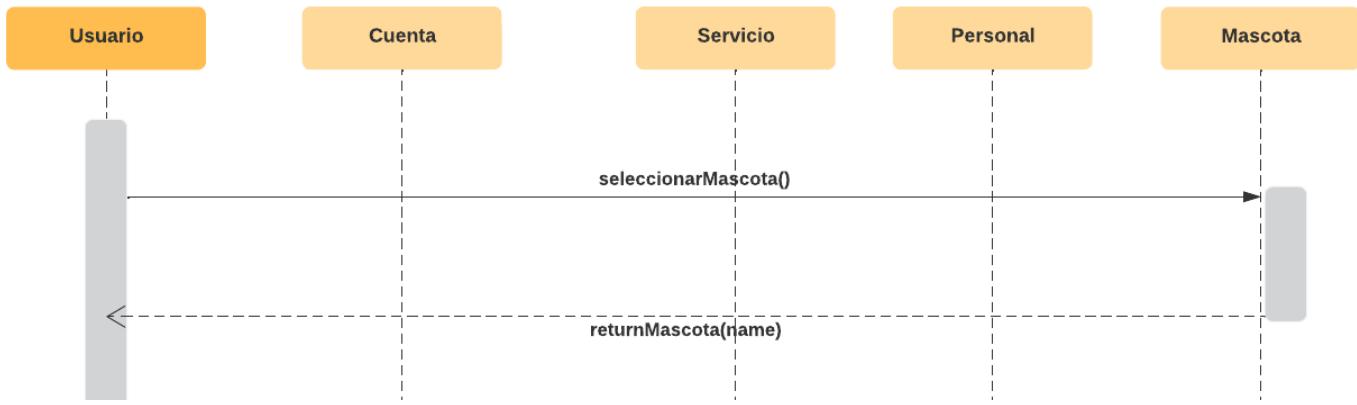
3.3.10.2.1. Diagrama de Clases



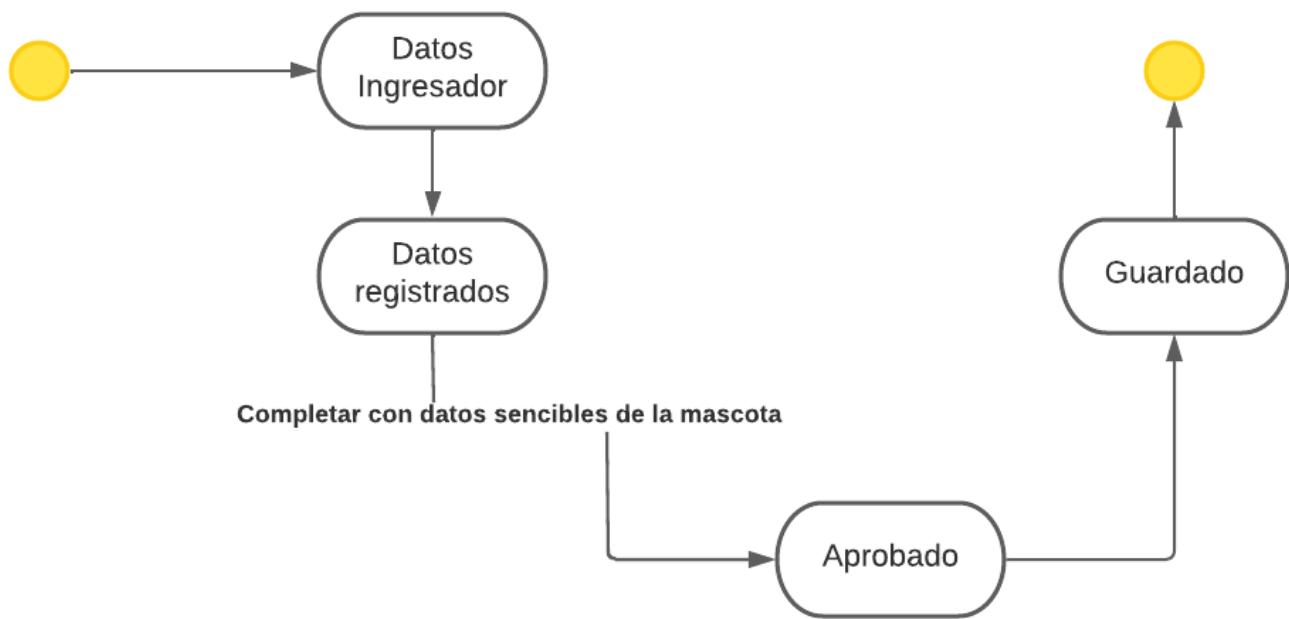
3.3.10.2.2. Diagrama de Objetos



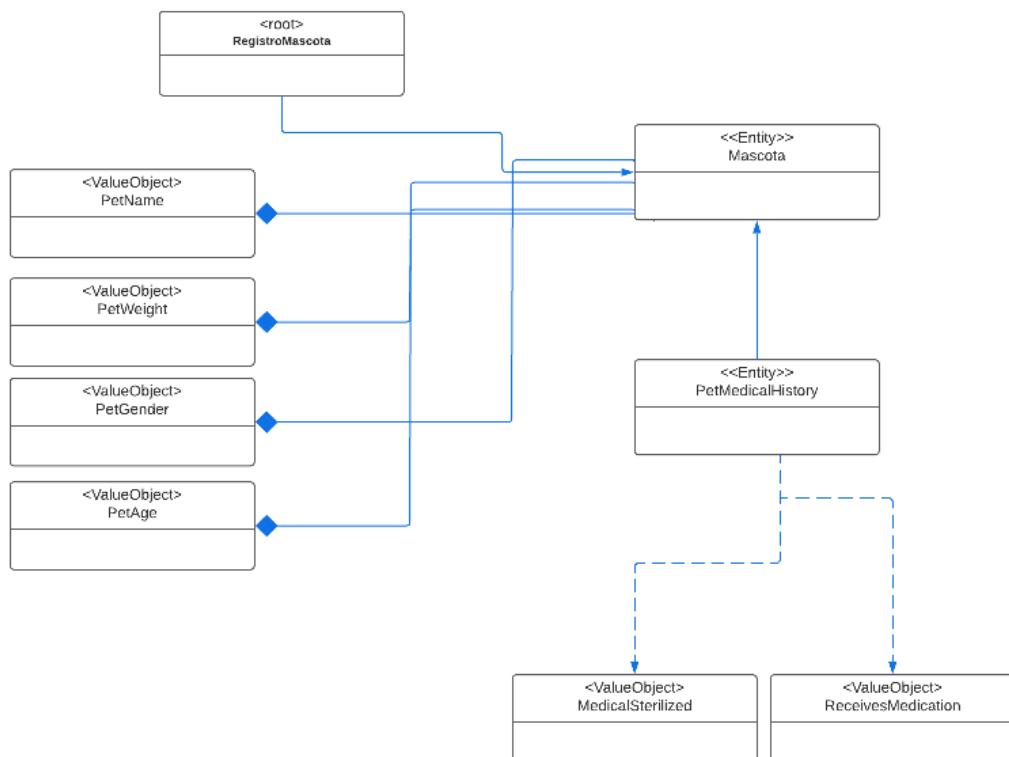
3.3.10.2.3. Diagrama de Secuencia



3.3.10.2.4. Diagrama de Estados

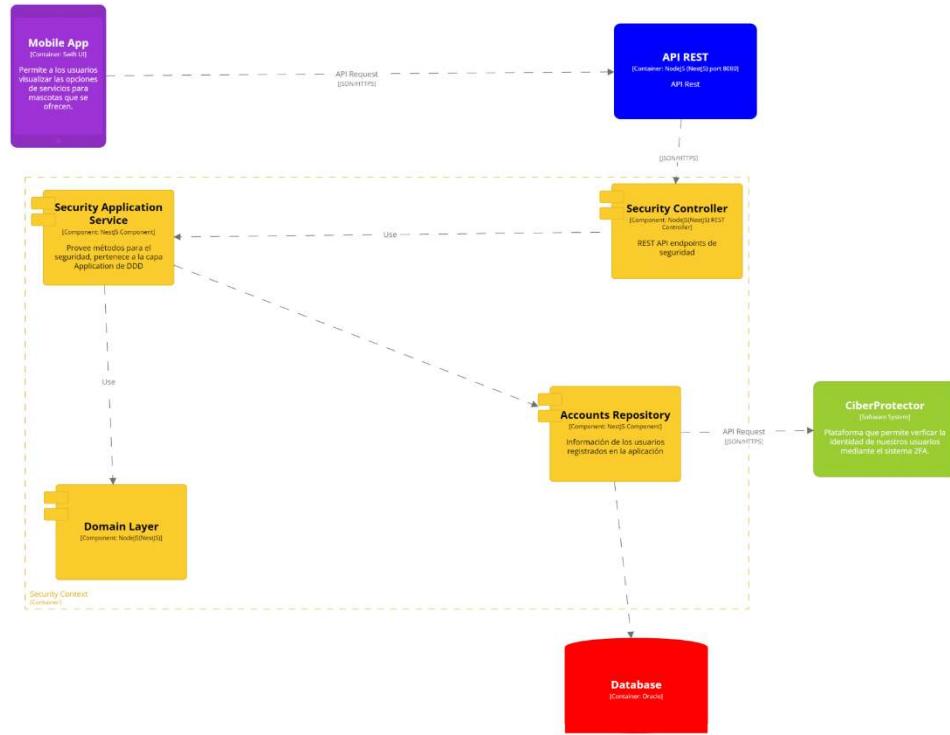


3.3.10.3. Diagrama de Agregados: Pets



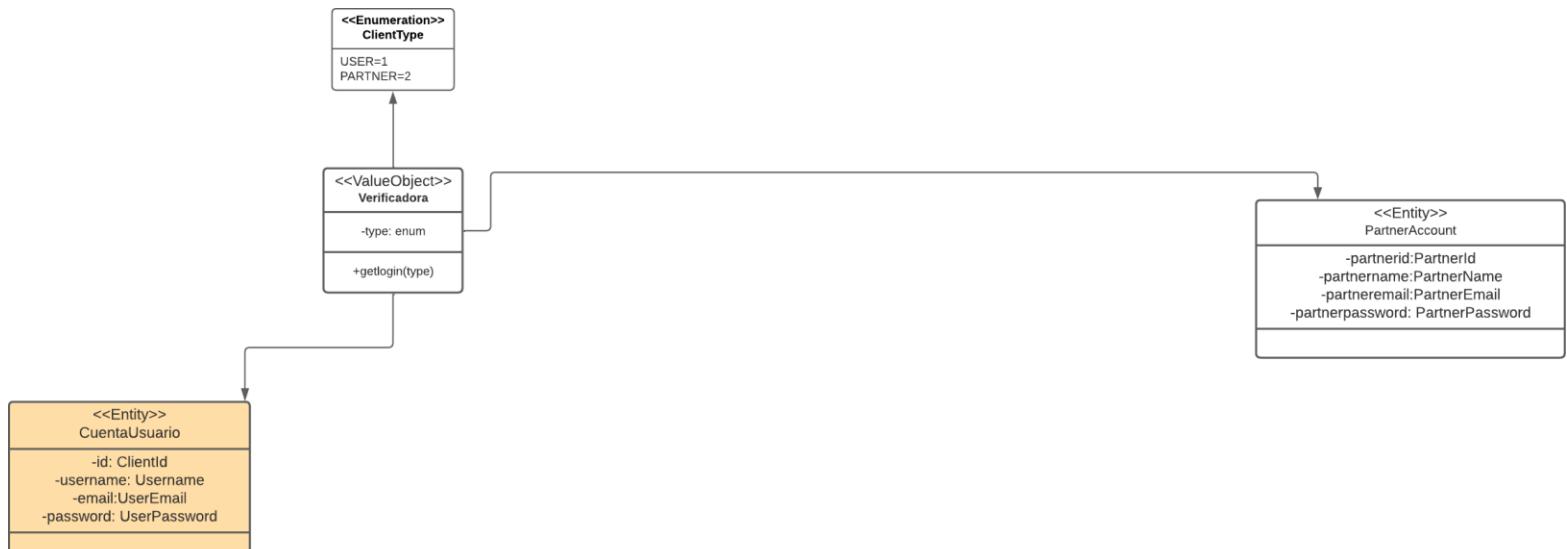
3.3.11. Bounded Context Security

3.3.11.1. Diagrama de Componentes

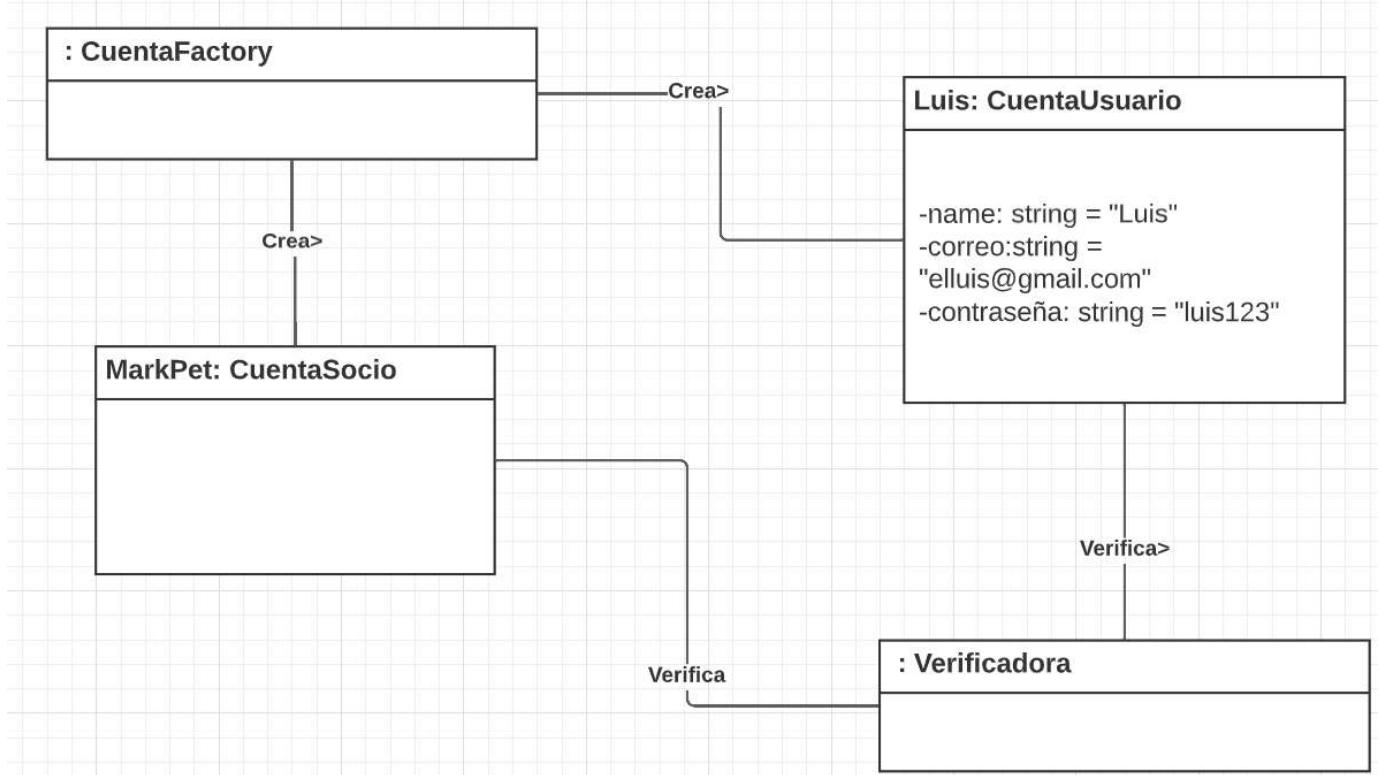


3.3.11.2. Diagramas de código

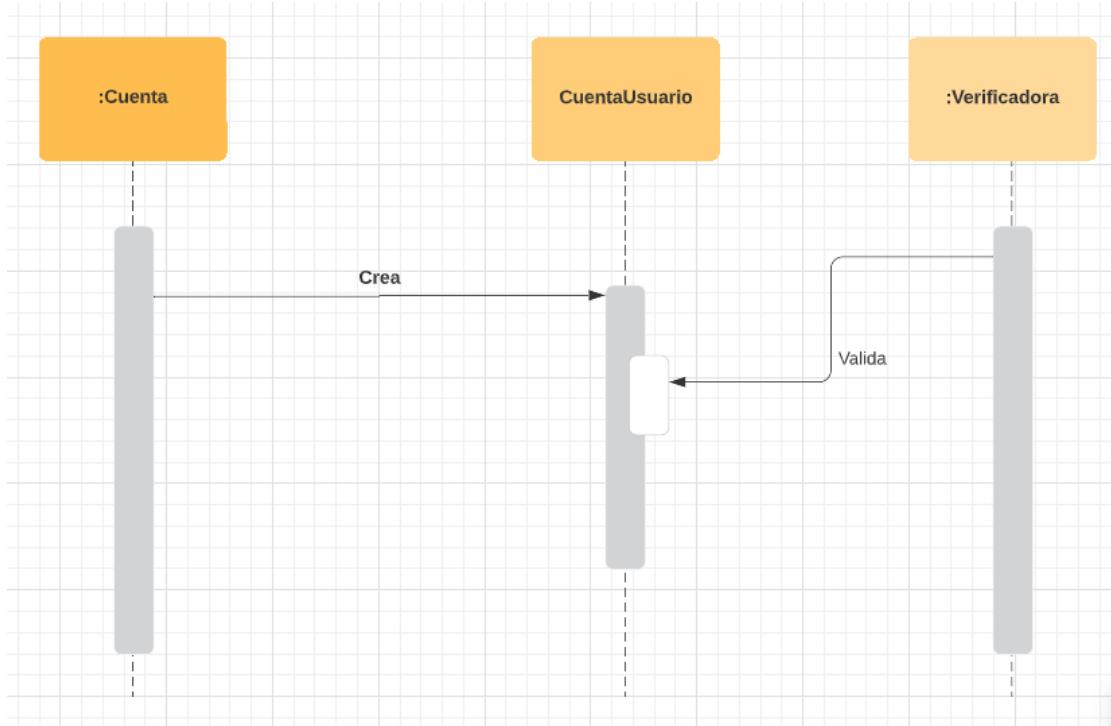
3.3.11.2.1. Diagrama de Clases



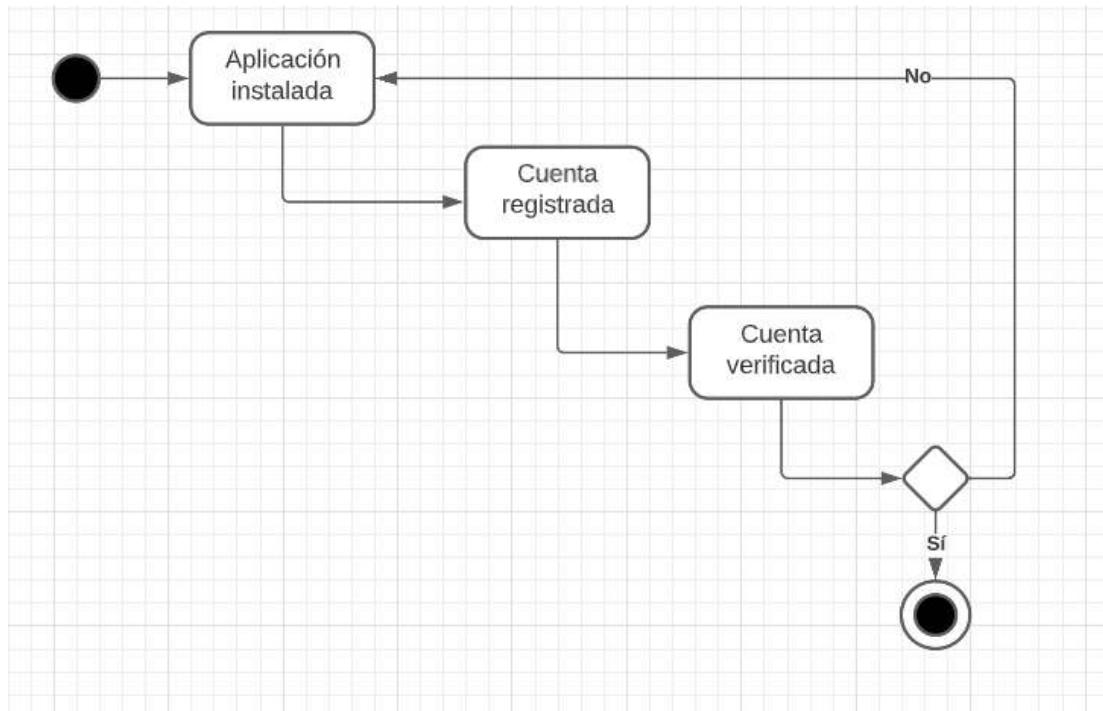
3.3.11.2.2. Diagrama de Objetos



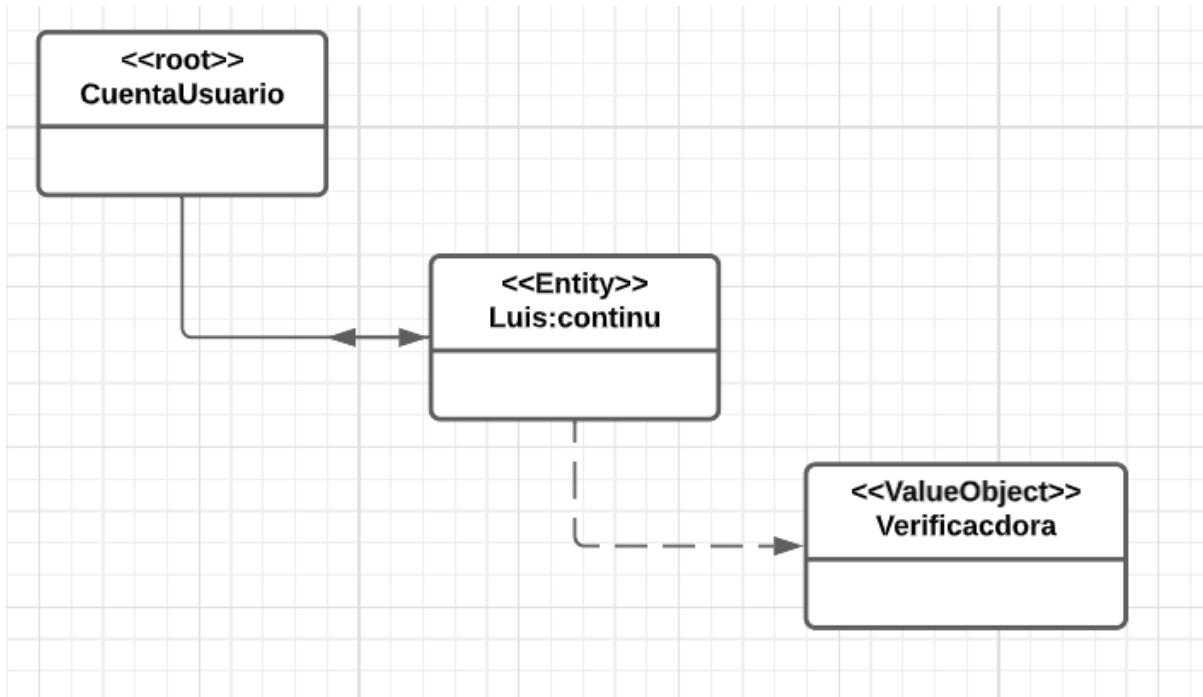
3.3.11.2.3. Diagrama de Secuencia



3.3.11.2.4. Diagrama de Estados

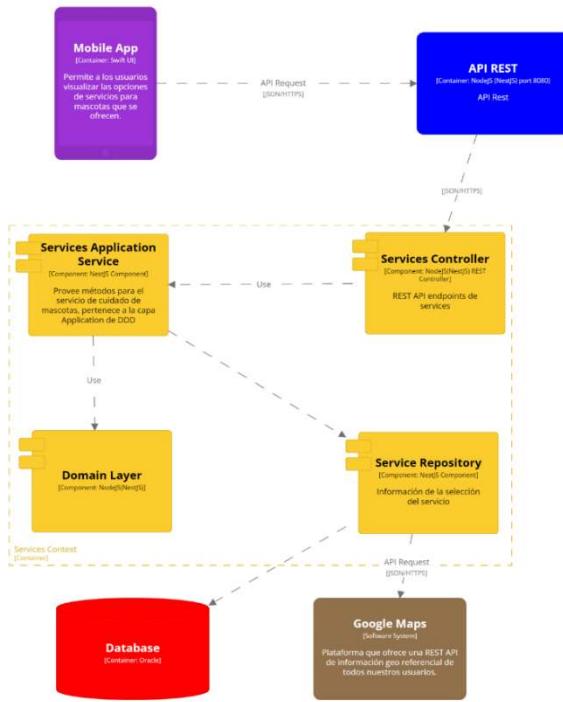


3.3.11.3. Diagrama de Agregados: Security



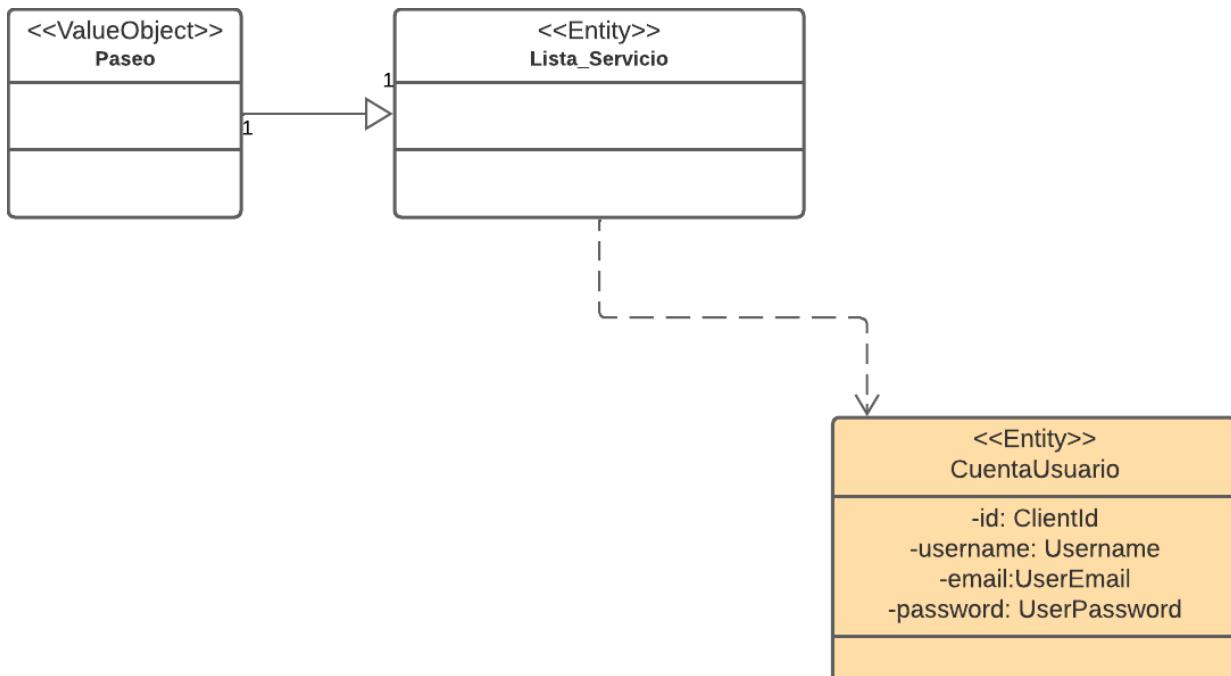
3.3.12. Bounded Context Services

3.3.12.1. Diagrama de Componentes

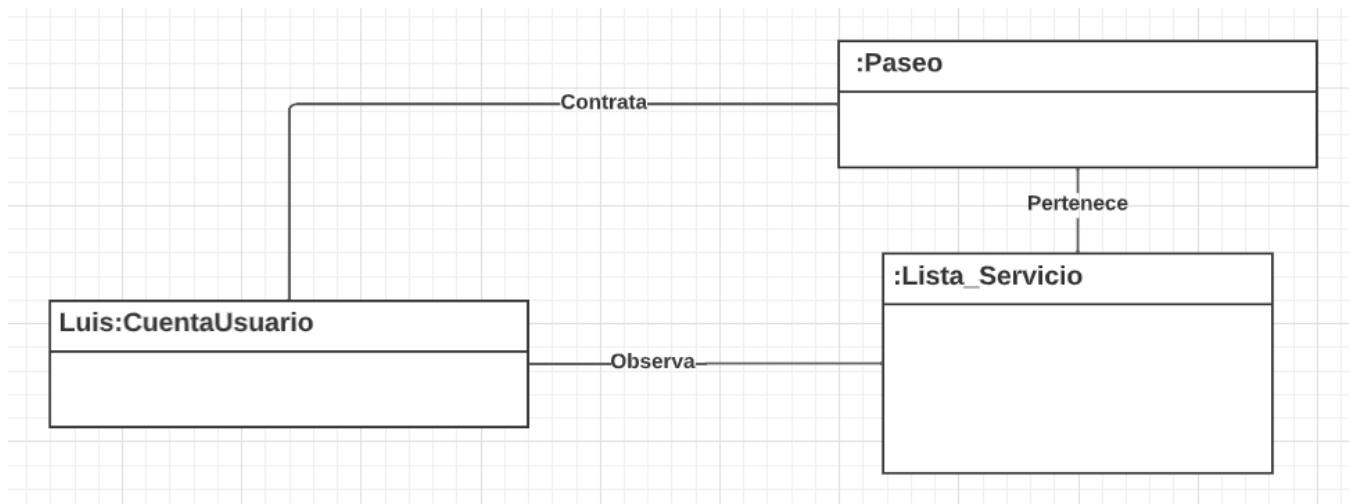


3.3.12.2. Diagramas de código

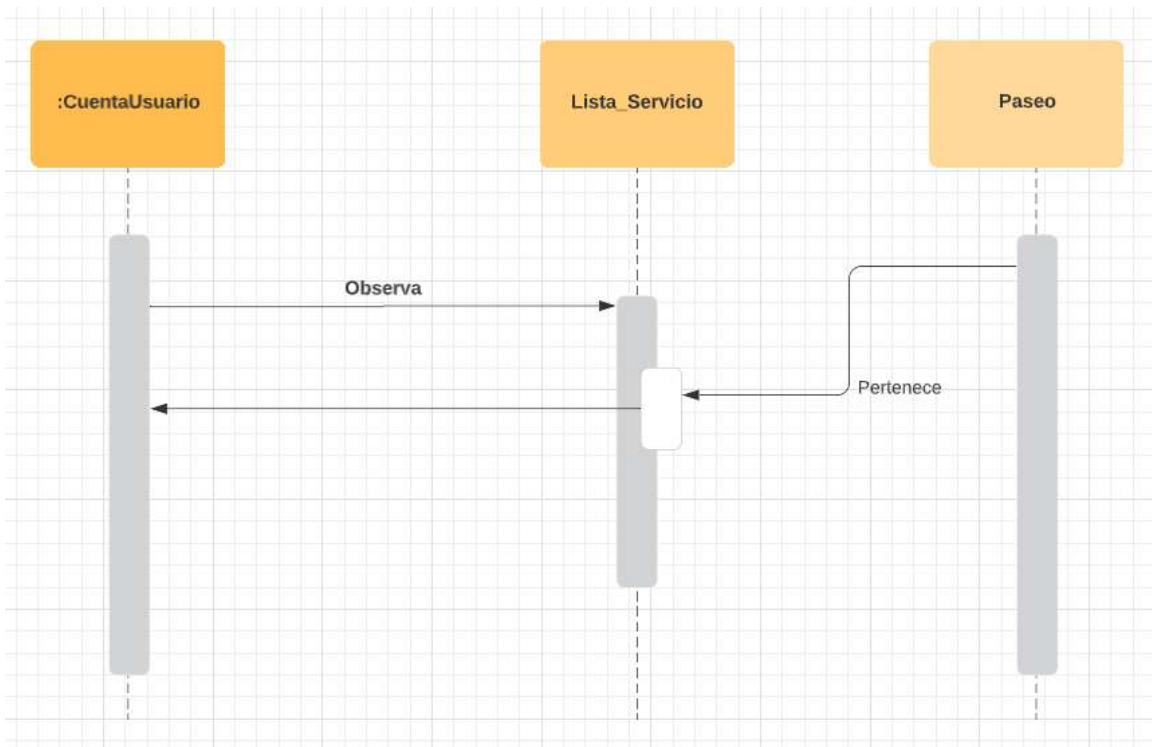
3.3.12.2.1. Diagrama de Clases



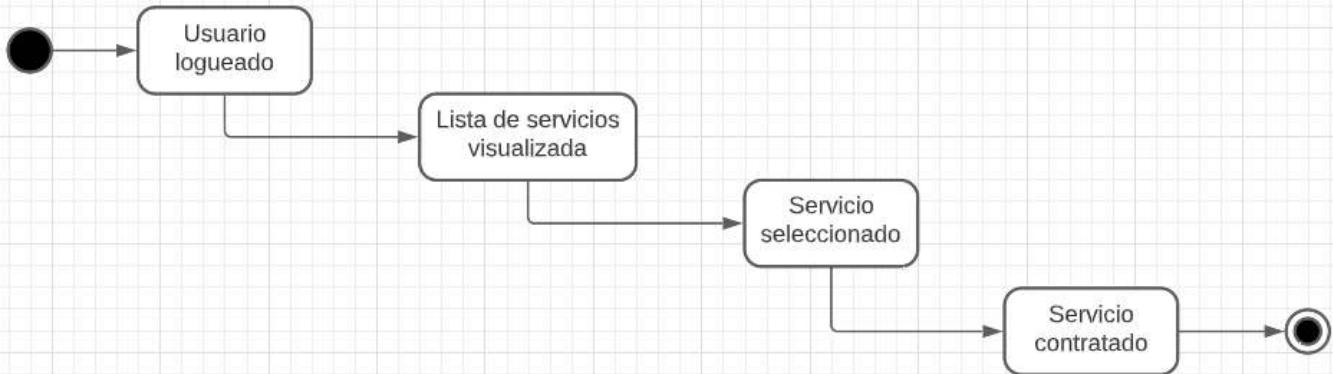
3.3.12.2.2. Diagrama de Objetos



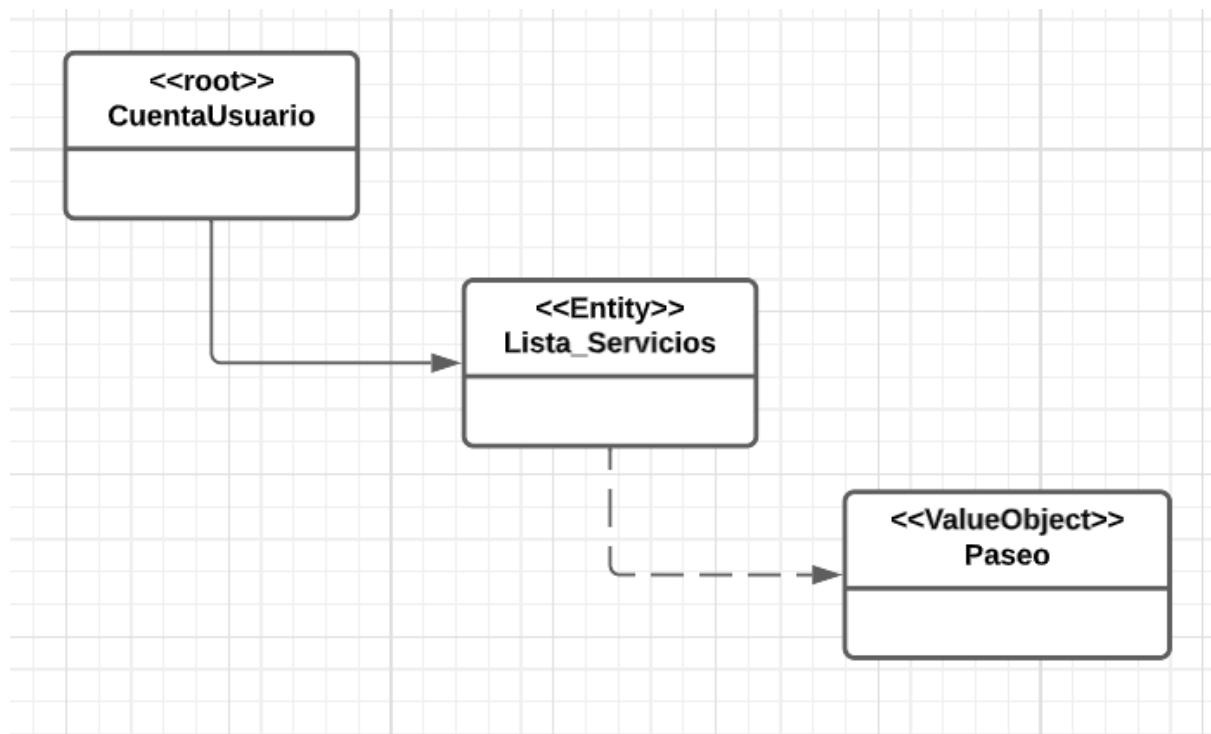
3.3.12.2.3. Diagrama de Secuencia



3.3.12.2.4. Diagrama de Estados



3.3.12.3. Diagrama de Agregados: Services



3.3.13. Diagrama de actividades

Diagrama 1: [Ver enlace ?](#)

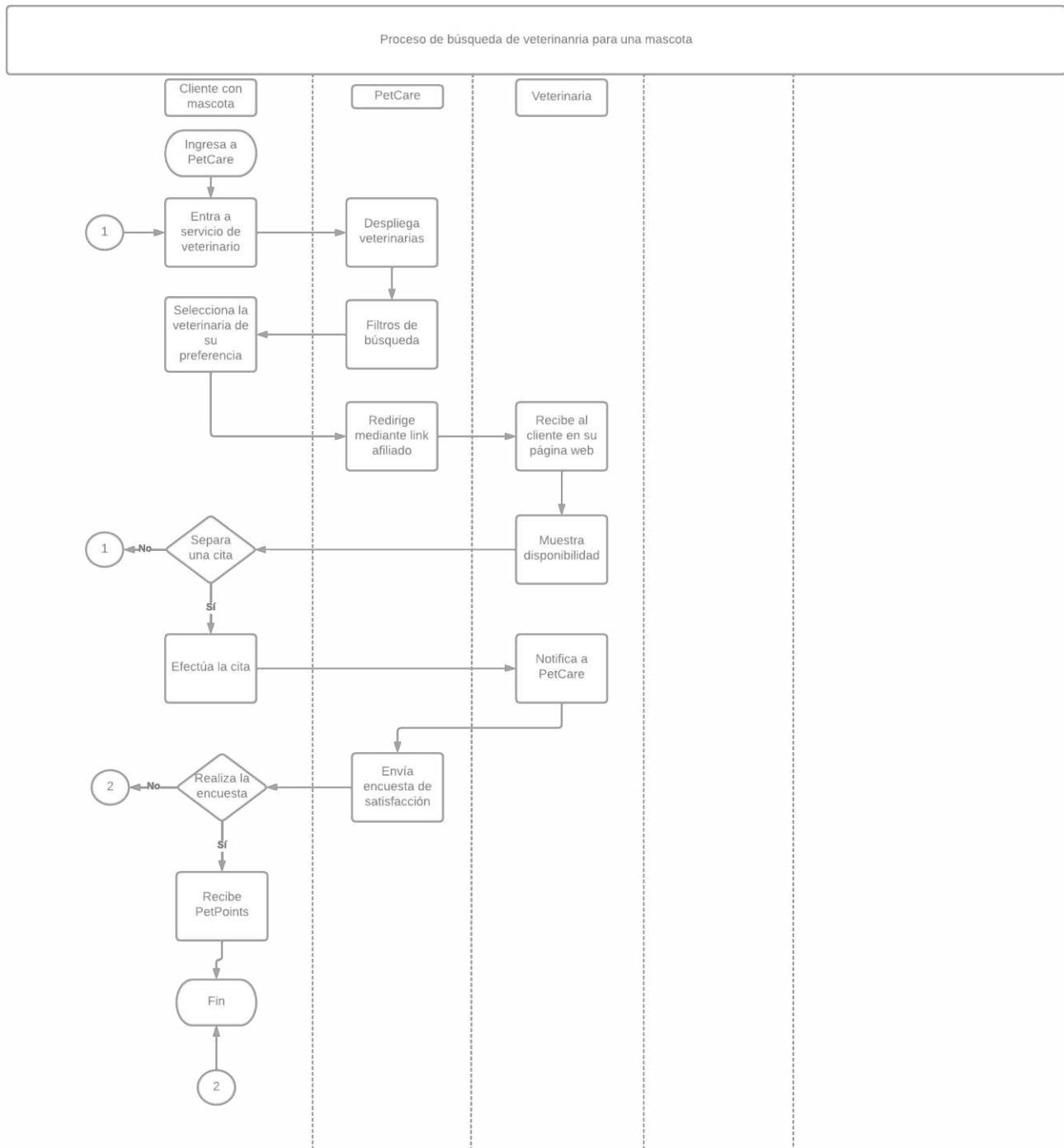


Diagrama 2: [Ver enlace 8](#)

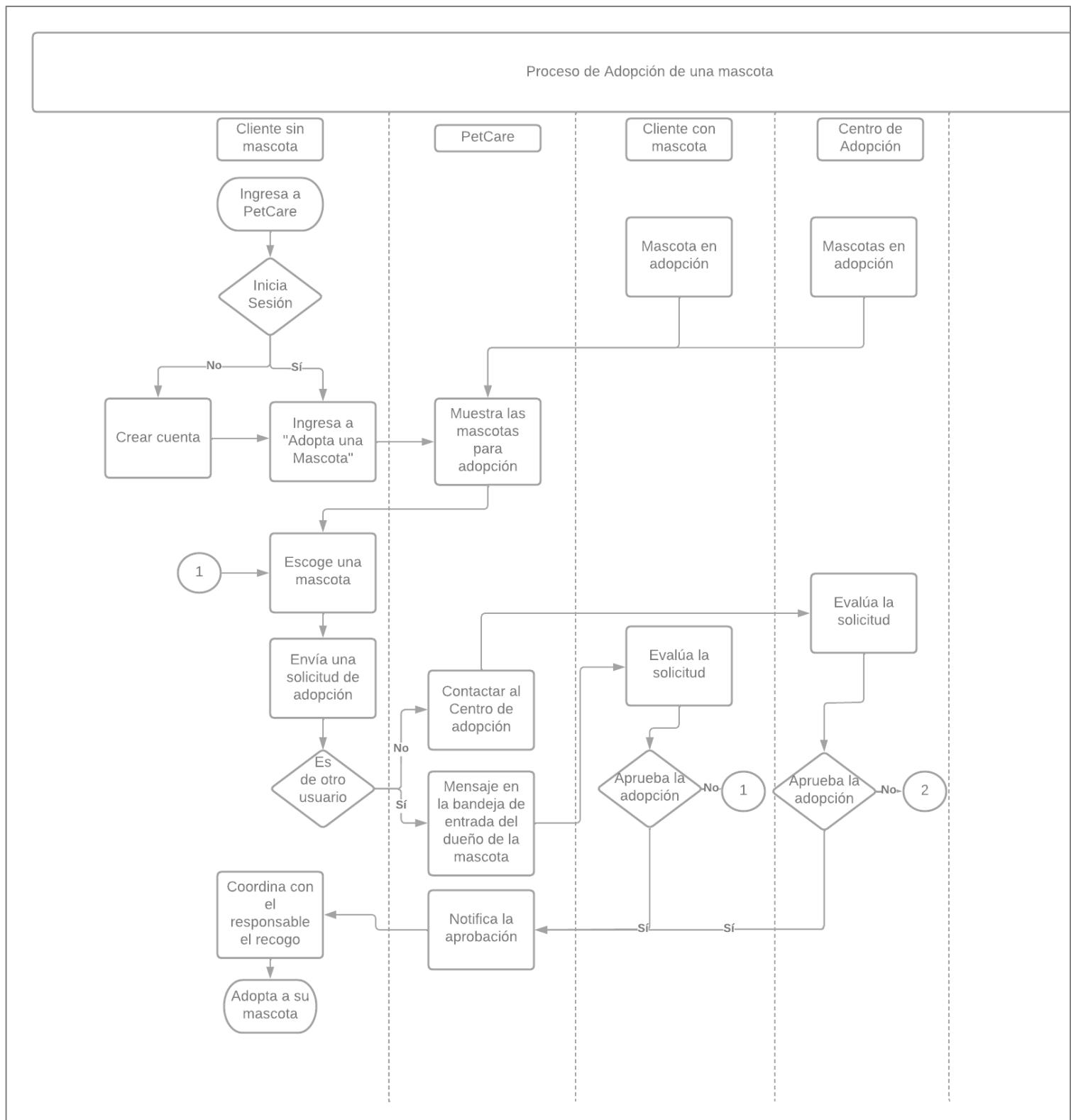
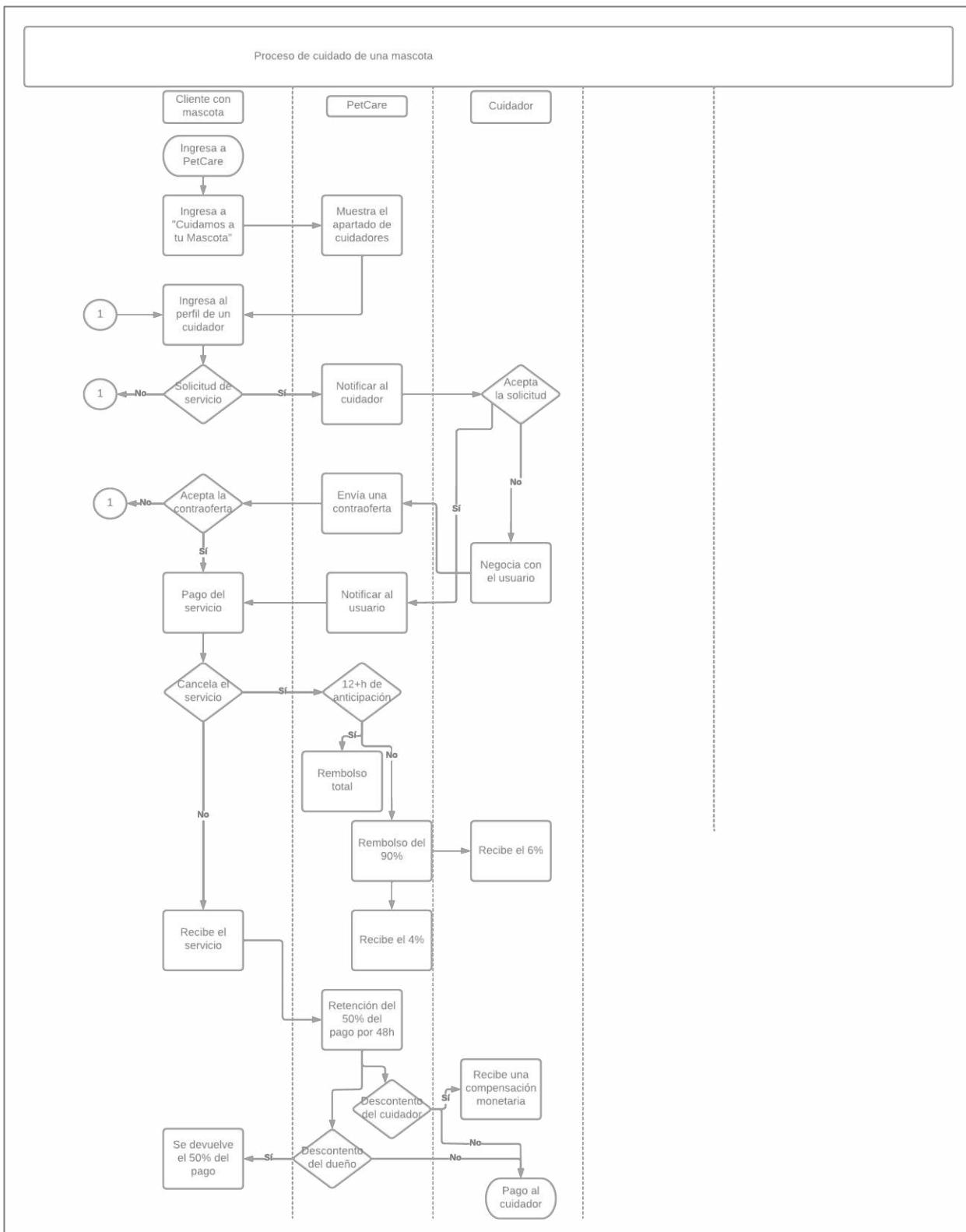


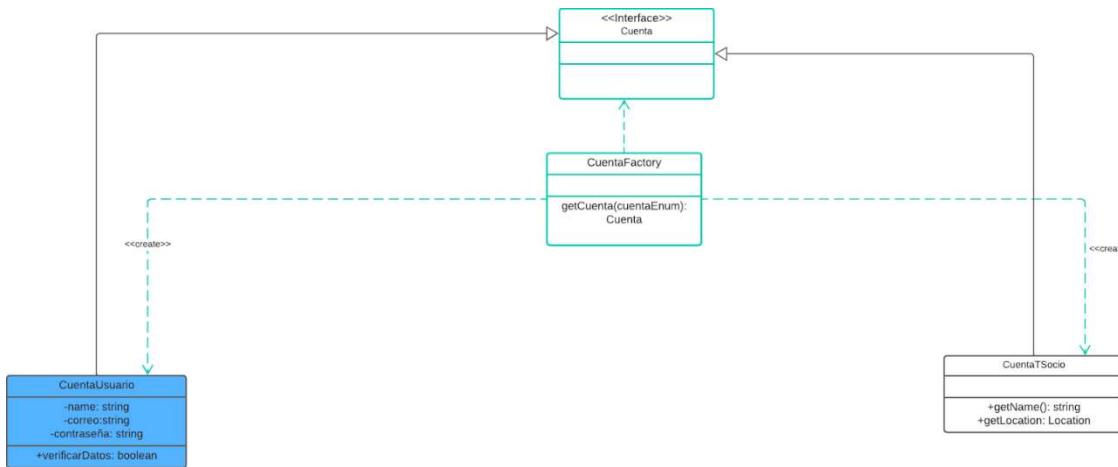
Diagrama 3: [Ver enlace 9](#)



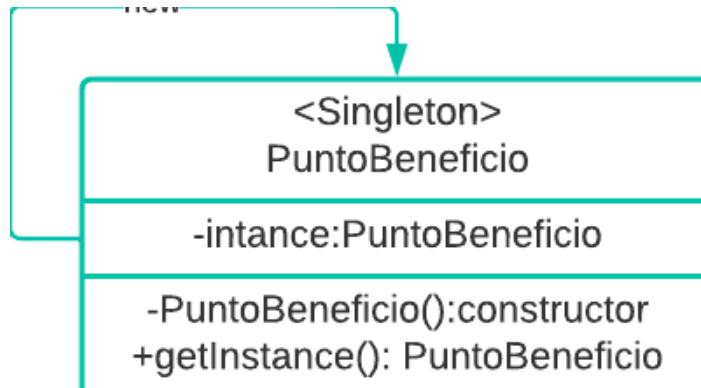
3.3.14. Patrones empleados

3.3.14.1. Creational

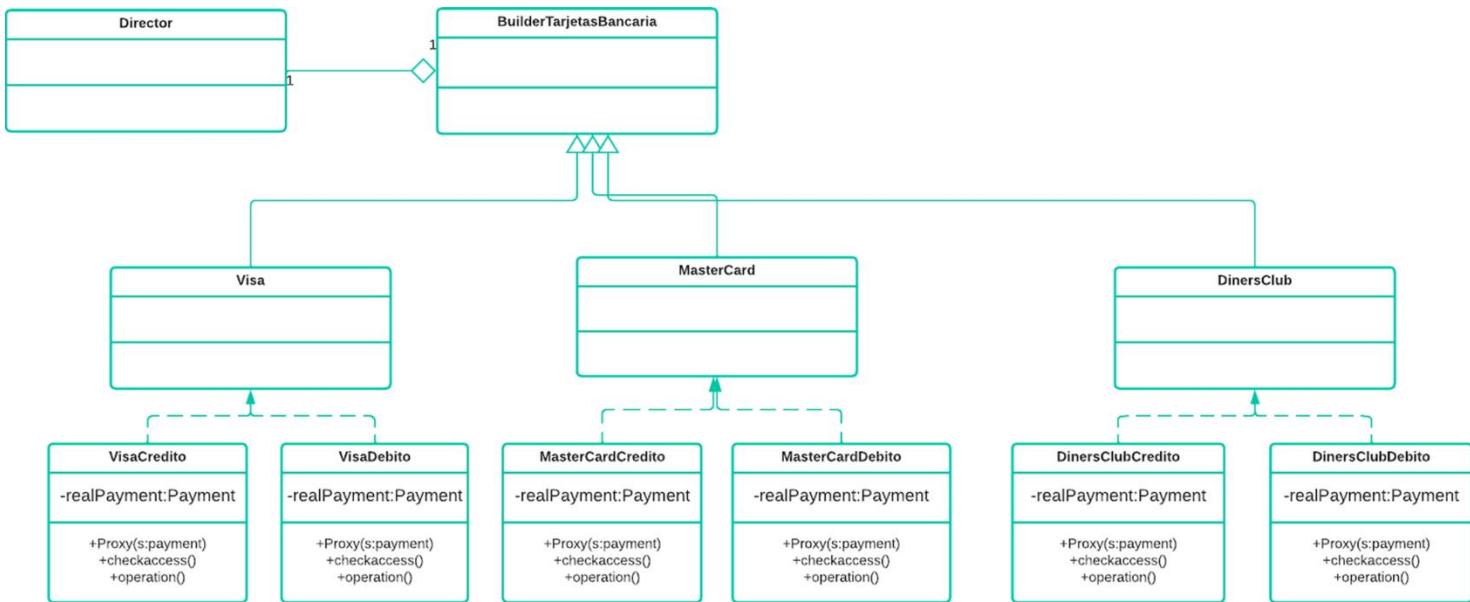
Factory Method: Decidimos implementar el patrón creacional Factory Method para poder ahorrar recursos del sistema al reusar los objetos cuentaUsuario y cuentaTSocio. Ello porque identificamos que el trabajar con if para cada clase hubiese llenado nuestro código de bad smell.



Singleton: Para la clase PuntoBeneficio observamos que se agregará una cierta cantidad constante de puntos por cada Usuario que se logre registrar de forma exitosa con el enlace compartido por el usuario principal. Por ello, implementamos el patrón creacional Singleton para sólo brinde los puntos una sola vez por usuario nuevo y si algún otro usuario necesite los puntos retorne la misma cantidad constante, evitándonos el crear un nuevo objeto PuntoBeneficio por el mismo usuario.

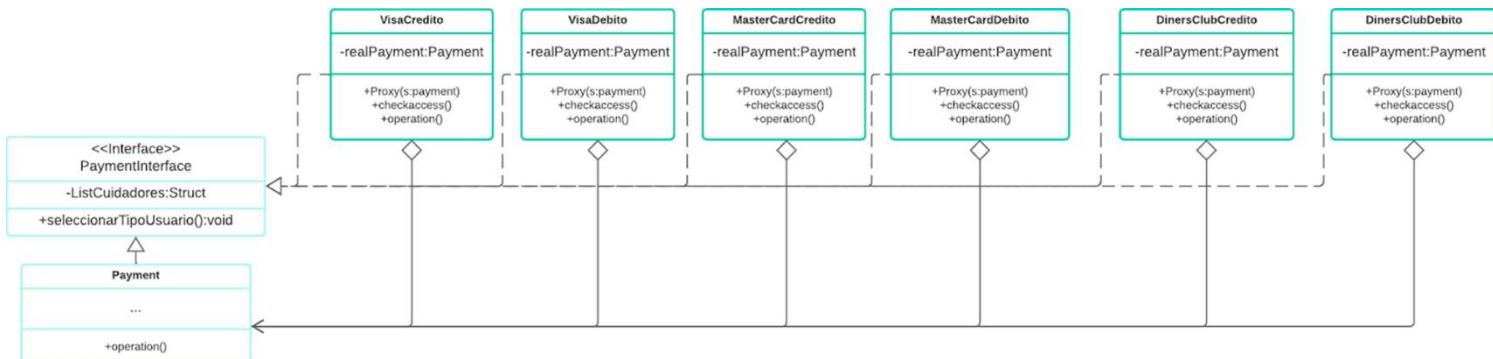


Builder: Nuestra aplicación recibirá medios de pago con tarjetas de crédito o débito, por ello implementamos el patron Builder para que podamos crear un registro de todas las tarjetas con un código base y obtener de ello productos concretos que serian las tarjetas registradas en nuestra base de datos.



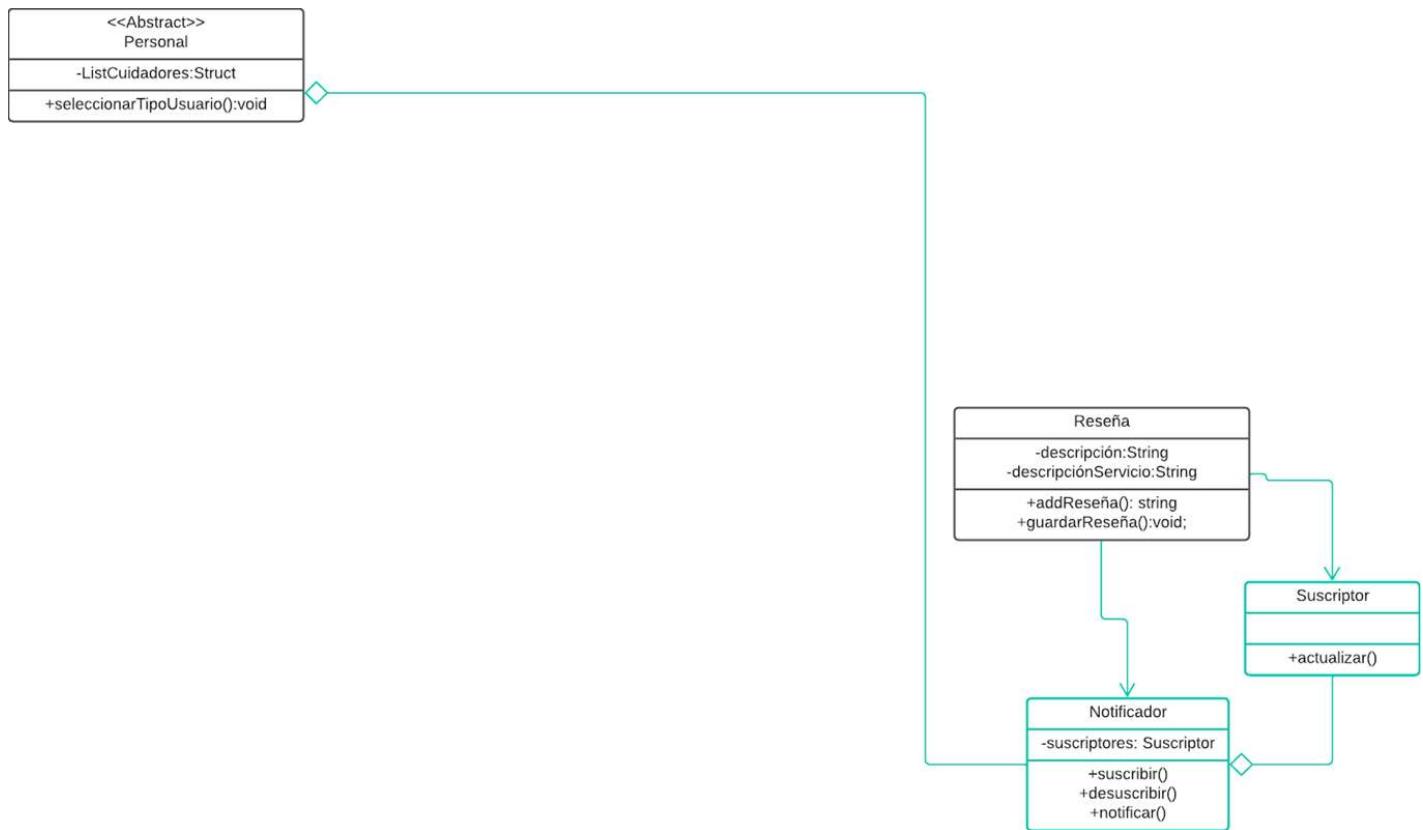
3.3.14.2. Structural

Proxi: Se usó el patrón estructural Proxy de forma remota, es decir se hace uso de un servicio ubicado en un servidor remoto. En nuestro caso, el realizar el pago con una tarjeta de crédito o débito, realizando una solicitud de pago a la entidad bancaria que está encargada de gestionar la tarjeta correspondiente y que se pueda proceder con el pago del servicio.



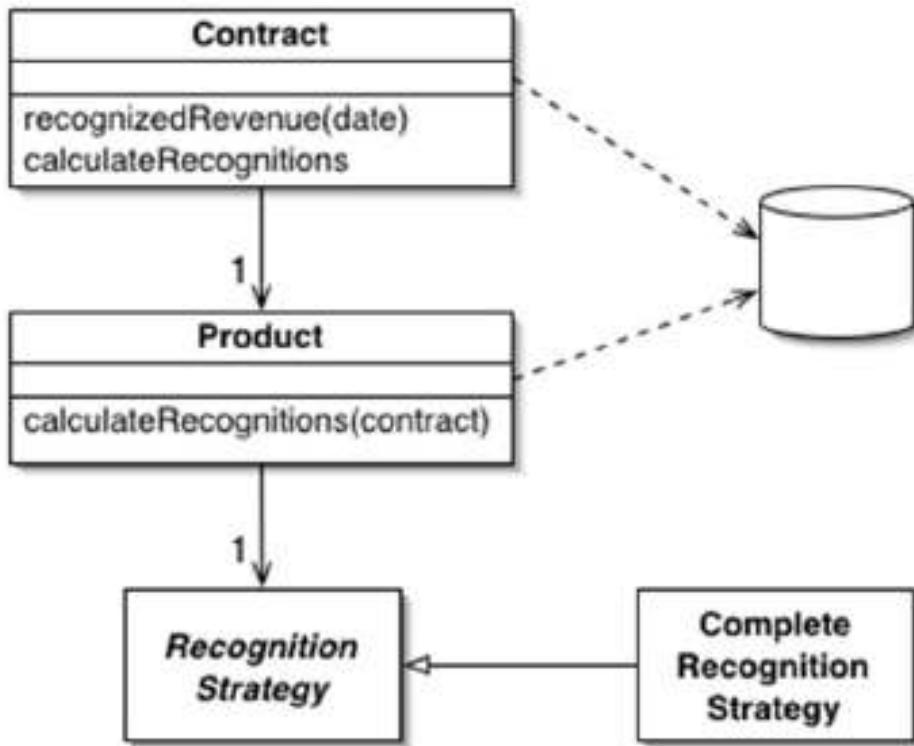
3.3.14.3. Behavioral

Observer: Se uso el patrón de comportamiento Observer para que se pueda realizar notificaciones en nuestra app. Donde en base a la clase reseña se tomará en cuenta si es que el servicio externo que tuvo fue bueno o malo. En caso de que sea bueno, se agregara al usuario como un suscriptor y se enviara este a la clase notificadora para poder registrarla y que pueda recibir notificaciones del servicio que le agrado. En caso de que el usuario no quiera recibir las notificaciones puede retirarlas, debido a que la lista de suscriptores es dinámica es decir que pueden desactivar la suscripción si es que lo quieren.

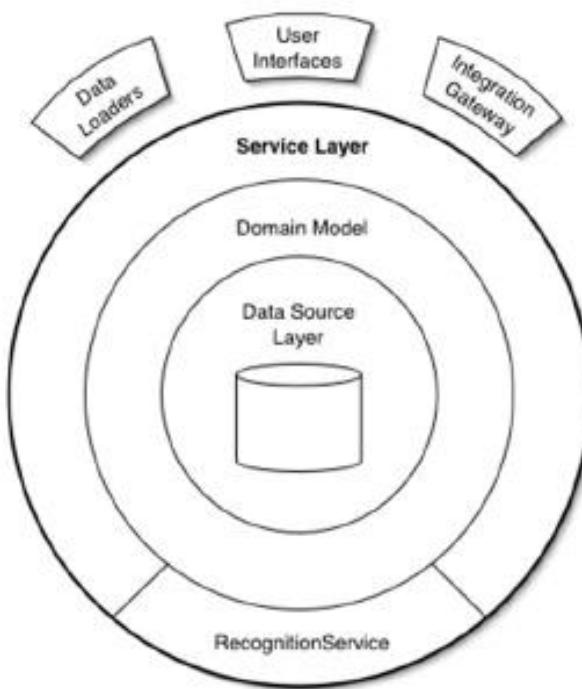


3.3.14.4. Domain Logic Patterns

Domain Model: Usaremos este patrón para abordar la complejidad formando una red de objetos que se interconectan. En este caso, todas nuestras entidades, sus atributos y relaciones se unen para crear la solución de software según la lógica del negocio.

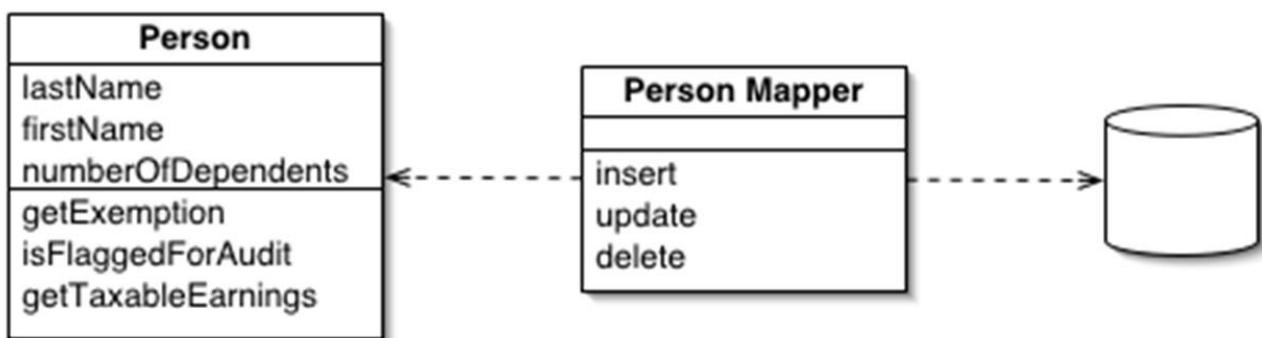


Service Layer: El patrón Service layer, permite encapsular nuestra lógica de negocios el cual se encuentra en el modelo del dominio. Su funcionalidad le permite iniciar, confirmar o abortar una transacción. Con la finalidad de que el modelo del dominio no tenga contacto con la tecnología.



3.3.14.5. Datta Source Architectural Patterns

Data Mapper: Utilizaremos este patrón para que las clases sean agnósticas a la base de datos. En lugar de ello, se utilizará un Mapper, en este caso un Repository, que dependa de las clases y realice las transacciones en la base de datos.



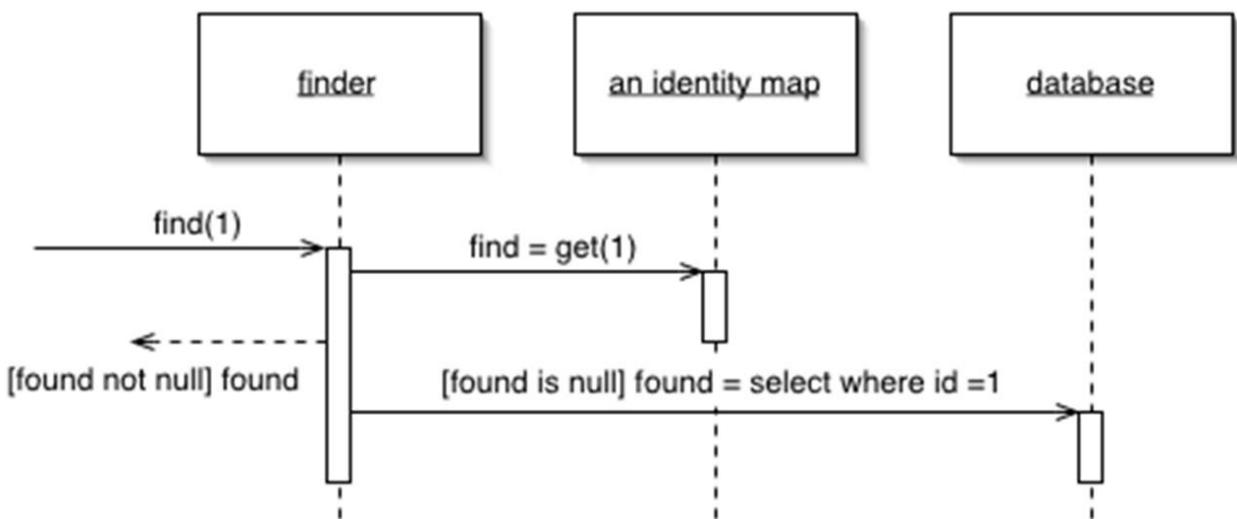
3.3.14.6. Object-Relational Behavioral Patterns

Unit of work: Este patrón se aplicará para realizar los cambios en la base de datos. Es decir, habrá una clase que tenga los métodos para actualizar la base de datos. De esta manera se mantendrá la consistencia de objetos cuando se realicen las transacciones.

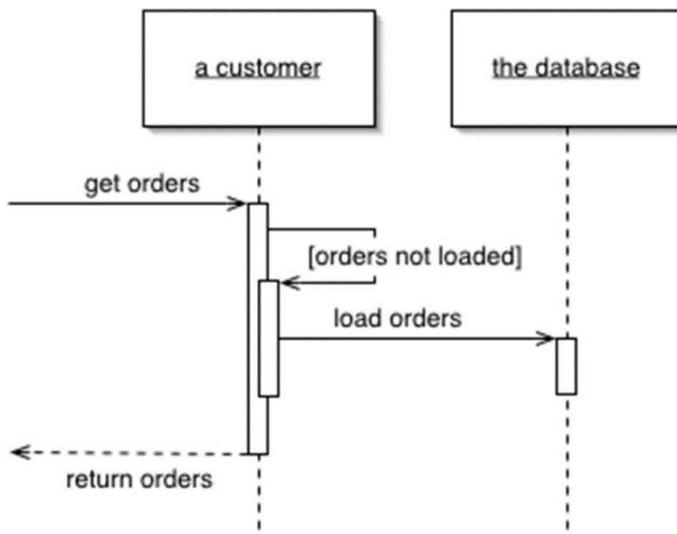
En este caso, este patrón no se implementará desde cero, pues el Framework ya lo tiene implementado.

Unit of Work
registerNew(object) registerDirty(object) registerClean(object) registerDeleted(object) commit rollback

Identity Map: Utilizaremos el siguiente patrón, el cual consiste de que cada tabla en nuestra base de datos tenga un identificador “ID”, de manera que cada objeto se cargue solo una vez, ello será posible porque se tendrá un diccionario con muchas identidades cargadas en un mapa, para no estar cargando constantemente de la base de datos. Es así que cada que necesitemos hacer una búsqueda de un objeto en específico acudiremos a dicho mapa.

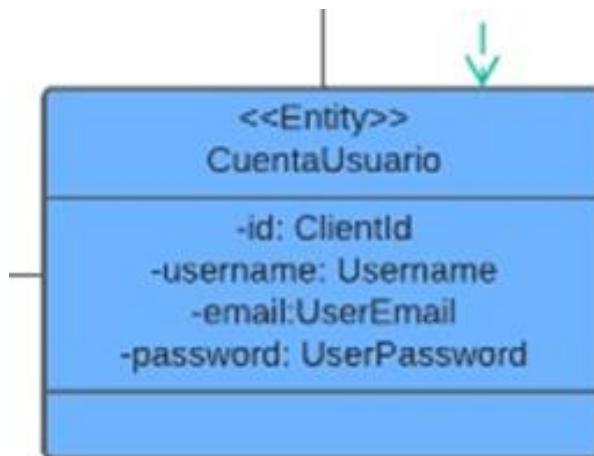


Lazy Load: El siguiente patrón nos permitirá que un objeto que no contiene todos los datos que necesita, sepa cómo obtenerlos, ello con la finalidad que solo hagamos uso de la información que se necesite en ese momento.

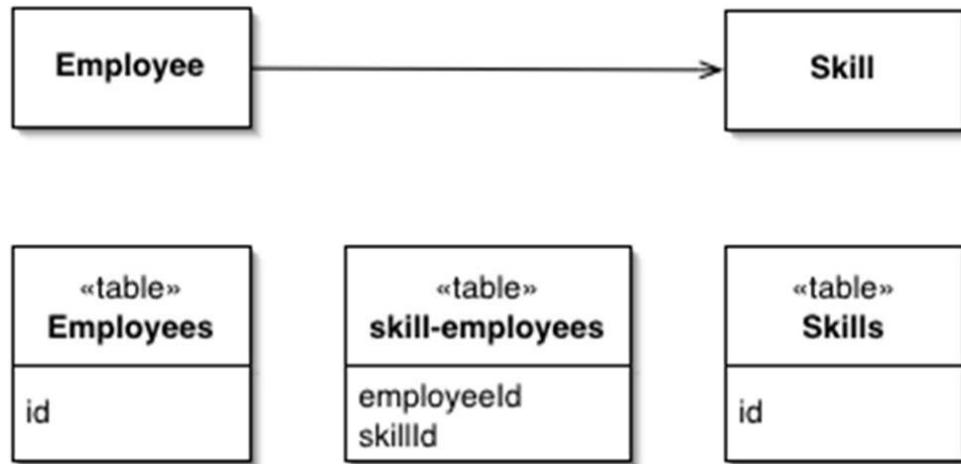


3.3.14.7. Object-Relational Structural Patterns

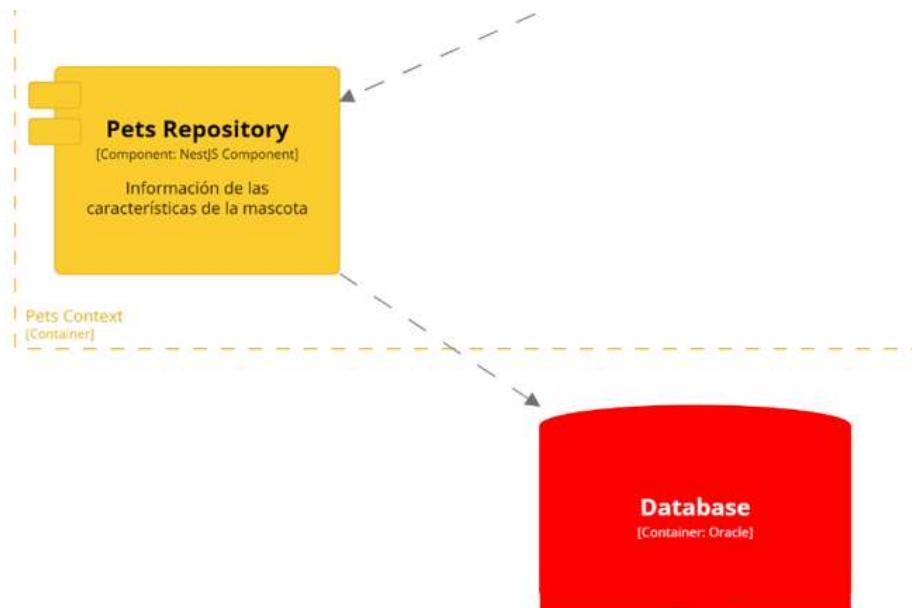
Identity Field: Este patrón consiste en que se guardará un campo ID, o también llamado PrimaryKey, de la base de datos en cada objeto. Con ello nos aseguramos que un objeto posea identidad entre la memoria y una fila de la base de datos. Para efectos de nuestro trabajo hemos colocado un "ID" a cada clase, en nuestros diagramas de clases por módulo.



Association Table Mapping: En caso contemos con relaciones de muchos a muchos en los objetos relacionales, crearemos una asociación o tabla intermedia que tenga referencias de las otras dos a través de llaves foráneas; tal como lo indica este patrón. La información adicional sería nuestra tabla asociada en el mundo de objetos y asimismo la tabla intermedia en el mundo relacional.



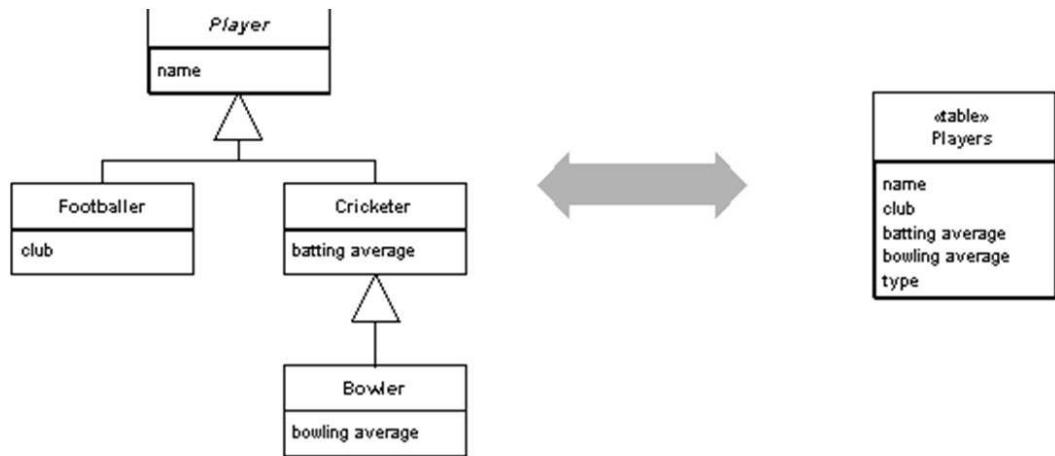
Dependent Mapping: El siguiente patrón será utilizado para realizar una asignación de la base de datos para una clase secundaria. Nuestro patrón “Repository”, que se menciona líneas a continuación, será el intermediario entre nuestras clases y base de datos



Embedded Value: El patrón a continuación nos dice que se ha de mapear un objeto en varios campos de la tabla de otro objeto. Para ello identificamos en nuestro diagrama de clases quienes serán las “Entity”, las que a su vez tendrán como atributos “Value Object”. Las “Entity” serán las únicas que posean el “ID” y no trabajarán con datos primitivos, sino que con referencias a las clases u objetos. Los “Value Object” tendrán un método “validar”, con la finalidad de cerciorarse que la información añadida es correcta para este tipo de objeto.

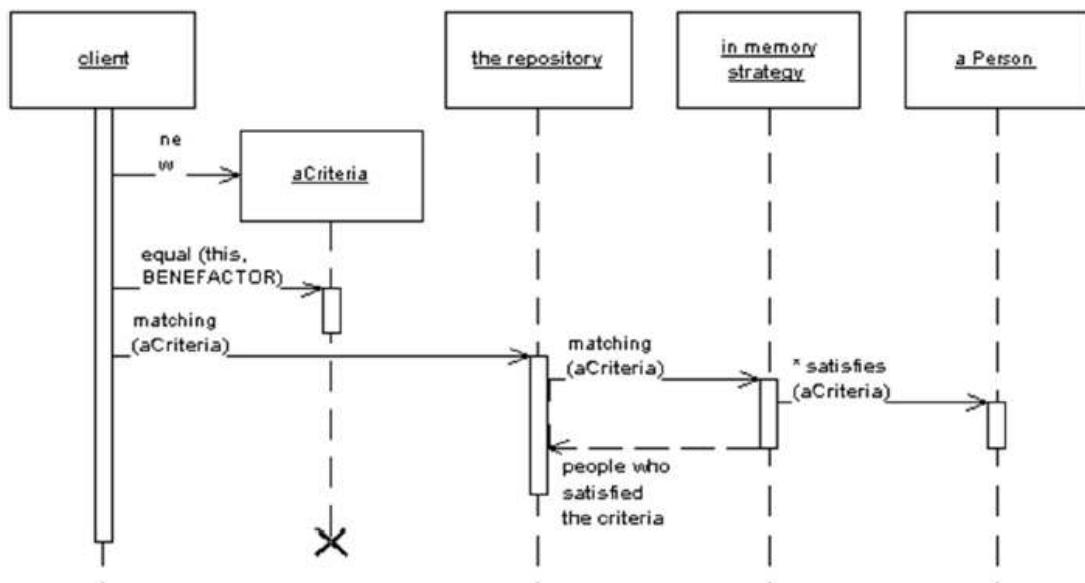


Single Table Inheritance: En nuestro proyecto, se usó el patrón Single Table Inheritance, pues, para mapear la herencia, cada jerarquía de clases será representada como una tabla única en el modelo relacional, la cual tendrá columnas para todos los campos de las diversas clases.



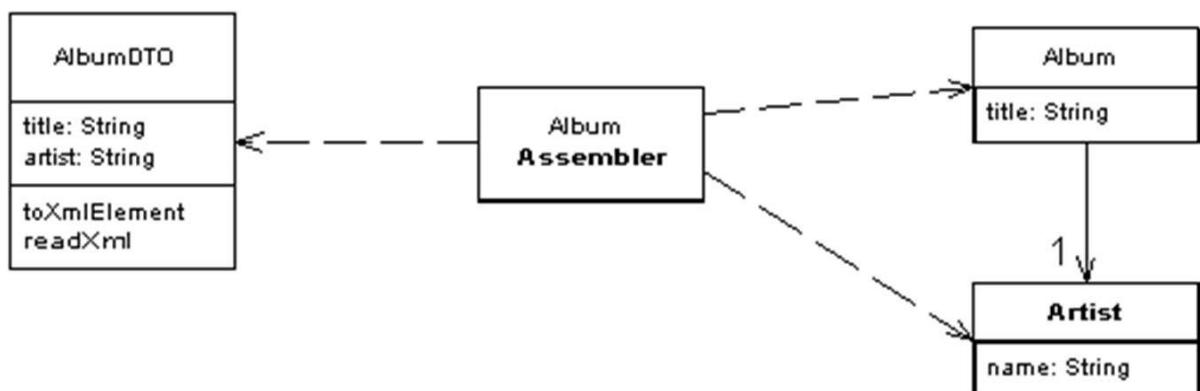
3.3.14.8. Object-Relational Metadata Mapping Patterns

Repository: El patrón empresarial de objetos relacionales Repository servirá de medio de comunicación entre el dominio de nuestros módulos y las capas de mapeos de datos.



3.3.14.9. Distribution Patterns

Data Transfer Object: En este patrón, tendremos un objeto ensamblador que será un conversor de datos a objetos o viceversa. Servirá tanto para crear datos como para consultar datos y poder retornarlos eficientemente. De esta manera, se reducirá la cantidad de llamadas a métodos.



4. CONSTRUCCION DEL PRODUCTO

4.1. Landige Page

[Ver enlace 10](#)

PetCare busca ser el lazo entre tú y los mejores servicios para tu mascota

Nuestra empresa PetSolution busca ser intermediario entre los dueños de mascotas y los negocios o personas que brindan servicios especializados. De esta manera, lograremos reducir el tiempo que se toman las personas para encontrar centros de atención y cuidado animal.



Contrata cuidadores

En PetCare tú y tu mascota tendrán un perfil personalizado, con el cual podrán unirse a la comunidad. De este modo, tendrán la opción de buscar y contratar cuidadores.



swipe up!

Ubica tiendas especializadas

Podrás elegir entre un apartado de opciones de negocios y/o personal que ofrecen servicios y productos para mascotas como veterinarias, Pet Spas, hoteles para mascotas, entre otros.

swipe up!





Obtén beneficios

También, ofrecemos beneficios como, por ejemplo, la obtención de puntos por cada vez que compres un producto o solicites un servicio. Con ellos, podrás canjear descuentos en los artículos de las tiendas asociadas.

Disponible para descargar en las principales tiendas digitales



Por: PetSolution

4.2. API REST

4.2.1. Construcción

Evidencia del repositorio GitHub del código

[Ver enlace 11](#)

The screenshot shows a GitHub repository page for 'CarloLSG / petcare-ddd-scratch'. The repository is public and has 2 branches and 0 tags. The main branch is selected. The commit history shows the following files and their first commits:

File	Commit Message
src	Co-authored-by: u20211b786 (Espejo Macuri, Paolo Andre) <U20211B786@u...
test	first commit
.eslintrc.js	first commit
.gitignore	first commit
.prettierrc	first commit
README.md	first commit
nest-cli.json	first commit
package-lock.json	first commit
package.json	first commit
tsconfig.build.json	first commit
tsconfig.json	first commit

Toma de capturas de las distintas capas del código

```
└── src
    ├── appointments
    │   ├── application
    │   ├── domain
    │   ├── infrastructure
    │   ├── interface
    │   └── TS appointments.module.ts
    ├── application
    │   ├── dtos
    │   │   ├── request
    │   │   │   └── TS edit-appointment-request.dto.ts
    │   │   │   └── TS register-appointment-request.dto.ts
    │   │   ├── response
    │   │   │   └── TS appointment.dto.ts
    │   │   └── TS register-appointment-response.dto.ts
    │   ├── handlers
    │   │   ├── commands
    │   │   │   └── TS register-appointment.handler.ts
    │   │   ├── queries
    │   │   │   └── TS get-appointments.handler.ts
    │   │   └── mappers
    │   │       └── TS appointment.mapper.ts
    │   ├── messages
    │   │   ├── commands
    │   │   │   └──
    │   │   ├── queries
    │   │   └──
    │   └── services
    │       └── TS appointment-application.service.ts
    └── validators
        └── TS register-appointment.validator.ts
```

```
    ↘ domain
        ↘ aggregates/client
            TS appointment-id.value.ts
            TS appointment.entity.ts
            TS appointment.repository.ts
            TS appointment.root.entity.spec.ts
        ↘ events
            TS appointment-registered.event.ts
        ↘ factories
            TS appointment.factory.ts
    ↘ infrastructure/persistence
        ↘ entities
            TS appointment.entity.ts
        ↘ repositories
            TS appointment.repository.ts
        ↘ values
            TS day.value.ts
            TS month.value.ts
            TS year.value.ts
    ↘ interface/rest
        TS appointment.controller.ts
    ↘ shared
        > application
        > domain
        > infrastructure
        > interface
```

```
    ↘ application
      TS app-settings.ts
      TS app-util.ts
      TS app.error.ts
      TS app.notification.ts

    ↘ domain/values
      TS audit-trail.value.ts
      TS date-time.value.ts
      TS day.value.ts
      TS email.value.ts
      TS id-number.value.ts
      TS month.value.ts
      TS password.value.ts
      TS person-name.value.ts
      TS year.value.ts

    ↘ infrastructure/persistence
      ↘ migrations
        TS 1669297643709-InitialSchema.ts
        TS 1669298211225-MasterData.ts
        └ initial-schema.sql
        └ master-data.sql

    ↘ interface/rest
      TS api.controller.ts
      TS envelope.ts
```

```
    ↘ users
        ↘ domain/aggregates/user
            TS user-id.value.ts
            TS users.module.ts
        ↘ notifications
            ↘ application/handlers/events
                TS person-registered.handler.ts
                TS notifications.module.ts
        ↘ owners
            > application
            > domain
            > infrastructure
            > interface
            TS owners.module.ts
        ↘ application
            ↘ dtos
                ↘ request
                    TS edit-person-request.dto.ts
                    TS register-person-request.dto.ts
                ↘ response
                    TS person-owner.dto.ts
                    TS register-person-response.dto.ts
            ↘ handlers
                ↘ commands
                    TS register-person.handler.ts
                ↘ queries
                    TS get-person-owners.handler.ts
            ↘ mappers
```

```
TS person.mapper.ts
  ✓ messages
  ✓ commands
    TS register-person.command.ts
  ✓ queries
    TS get-person-owners.query.ts
  ✓ services
    TS person-application.service.ts
  ✓ validators
    TS register-person.validator.ts
  ✓ domain
    ✓ aggregates/owner
      TS owner-id.value.ts
      TS owner-type.enum.ts
      TS owner.root.entity.spec.ts
      TS owner.root.entity.ts
      TS person.entity.ts
      TS person.repository.ts
    ✓ events
      TS person-registered.event.ts
    ✓ factories
      TS person.factory.ts
    ✓ infrastructure/persistence
      ✓ entities
        TS owner.entity.ts
        TS person.entity.ts
      ✓ repositories
        TS person.repository.ts
      ✓ values
        TS email.value.ts
        TS owner-id.value.ts
        TS password.value.ts
        TS person-name.value.ts
        TS user-name.value.ts
    ✓ interface/rest
      TS person.controller.ts
```

4.2.2. Documentación

Documentación del API en una interface web

The screenshot shows the PetCare DDD API documentation interface. At the top, it says "PetCare DDD API 1.0 OAS3". Below that, there's a section for "person owners" with three endpoints: a green "POST /owners/person" for Register Person Owner, and two blue "GET /owners/person" and "GET /owners/person/{id}" endpoints. There's also a "default" section with a single blue "GET /" endpoint. Under "appointments", there are three blue "GET /appointments" and "GET /appointments/{id}" endpoints. At the bottom, there's a "Schemas" section with links to "RegisterPersonRequest" and "RegisterAppointmentRequest".

4.3. Pruebas unitarias

Pruebas Unitarias de la capa del dominio, cabe resaltar que nos centramos en el Bounded Context de Owner y Appointment

Para la ejecución del código y visualizar que las pruebas unitarias se ejecutan correctamente usamos el comando “npm run test”

```
[Carlos-MacBook-Air:~/Desktop/workspace-upc/petcare-ddd] carlolucas% npm run test
> petcare-ddd-nest@0.0.1 test
> jest

PASS  src/owners/domain/aggregates/owner/owner.root.entity.spec.ts
PASS  src/appointments/domain/aggregates/client/appointment.root.entity.spec.ts

Test Suites: 2 passed, 2 total
Tests:       2 passed, 2 total
Snapshots:   0 total
Time:        1.74 s, estimated 3 s
Ran all test suites.
[Carlos-MacBook-Air:~/Desktop/workspace-upc/petcare-ddd] carlolucas%
```

Capturas de pantalla de la ejecución de las pruebas unitarias en Postman

The screenshot shows a Postman request to `http://localhost:3000/appointments`. The request method is `POST`. The `Body` tab is selected, showing the following JSON payload:

```

1  {
2    "day": "12",
3    "month": "11",
4    "year": "2022"
5  }

```

The response tab shows the following JSON response:

```

1  {
2    "result": {
3      "id": 1,
4      "day": "12",
5      "month": "11",
6      "year": "2022",
7      "createdAt": "2022-11-25T14:52:24Z",
8      "createdBy": 1
9    },
10   "errors": []
11 }

```

The screenshot shows the Postman application interface. At the top, there is a header bar with the method "POST", the URL "http://localhost:3000/owners/person", and a "Send" button. Below the header, there are tabs for "Params", "Authorization", "Headers (8)", "Body", "Pre-request Script", "Tests", and "Settings". The "Body" tab is selected, showing the following JSON payload:

```
1
2   ...
3     "userName": "Fabrizzio",
4     "email": "fab@upc.edu",
5     "password": "fabito"
```

Below the body editor, there are tabs for "Body", "Cookies", "Headers (7)", and "Test Results". The "Test Results" tab is selected, showing the response details: Status: 201 Created, Time: 44 ms, Size: 386 B, and a "Save Response" button. The response body is displayed in a "Pretty" JSON format:

```
1
2   "result": {
3     "id": 1,
4     "userName": "Fabrizzio",
5     "email": "fab@upc.edu",
6     "password": "fabito",
7     "createdAt": "2022-11-25T14:51:16Z",
8     "createdBy": 1
9   },
10  "errors": []
```

CONCLUSIONES

1. Nuestra aplicación PetCare estará centrada en ofrecerles a aquellas personas con mascotas el encontrar y contratar servicios para sus queridos compañeros. La app será un intermediario para que los dueños puedan hacer contacto con aquellos que ofrezcan servicios para las mascotas, y estos contarán con una reseña de calificación para poder conocer la calidad del servicio ofrecido por aquellas entidades.
2. Es importante delimitar el objetivo y el alcance del proyecto, pues es parte de la planificación que lleva a diseñar un software de calidad.
3. Los requisitos son la base principal para definir cómo es que la aplicación va a funcionar, porque es donde se definen todos los criterios para un correcto funcionamiento.
4. Las historias de usuario son la clave para centrarse en diseñar la aplicación cubriendo las necesidades del usuario.
5. El Product Backlog fue importante para priorizar historias de usuario en la primera versión de la aplicación y asignar los Story Points a cada una de ellas.
6. El Sprint Backlog fue determinante para la organización del proyecto y dar un plazo para los entregables.
7. Se aplicó un diseño centrado en el dominio, de esta manera se logra una mantenibilidad a largo plazo y el software es más cercano al cliente.
8. DDD nos permitió dividir el dominio en contextos, lo cual nos reduce la complejidad y en un futuro, el código estará más organizado.
9. Se realizó el EventStorming y Bounded Canvas para cada uno de los módulos con el fin de organizar el DDD estratégico.
10. El modelo C4 es importante para dividir el software desde el alto nivel, en la arquitectura, hasta el diseño detallado.
11. El diagrama de contexto nos ayudó a ubicar y relacionar nuestro software entre los sistemas externos y los usuarios que lo utilizarán.
12. El diagrama de contenedores nos permitió tener una visión amplia de los artefactos software de nuestro proyecto y API Rest.
13. Lo más difícil fue elaborar los diagramas de componentes para cada uno de los Bounded Context pues se debía diseñar patrones empresariales para cada uno de ellos.
14. Concluimos que es necesario elaborar los diagramas de agregados en cada módulo, pues nos ayuda a organizar las entidades y objeto-valor de la aplicación.
15. Al realizar los diferentes diagramas de código para cada contexto se puede valorar su relevancia en el funcionamiento de cualquier sistema de una empresa. Pues son una guía para el desarrollo de un software.
16. Realizar un diagrama de secuencias ayuda a comprender más el funcionamiento de la aplicación porque al contrario del diagrama de clases, este relaciona las clases realizando un simulacro de las funciones.
17. En cuanto a los diagramas de estado se evidencia para las clases creadas los distintos eventos que puede atravesar a lo largo de la ejecución de la aplicación, gracias a ello está

la referencia de cómo las clases deberán ser programadas para así cumplir con los objetivos en cuanto al comportamiento deseado para estas entidades.

18. El haber implementado patrones de diseño al diagrama de clases nos permite realizar un diseño y estructura de código más eficiente orientado a la aplicación que queremos crear. Con ello aseguramos que la aplicación contará con un soporte sólido y a largo plazo las implementaciones de mejora no destruirán la lógica inicial base del proyecto.

19. Lo que más ayudó a organizar los diagramas de componentes fueron los patrones empresariales, que sirvieron para diseñar cómo la API Rest se conectaría con la Base de Datos.

20. El trabajo realizado es de importancia debido a que nos cambia la percepción de que todo es código y nos muestra una realidad del diseño y planificación antes de empezar a codificar formalmente.

RECOMENDACIONES

1. Realizar una completa observación a los problemas de la sociedad y escoger un problema con una alta cantidad de usuarios y amplia información sobre él

2. Delimitar alcance del proyecto de forma realista considerando restricciones, de tal manera que se proyecten más objetivos a lo largo del tiempo.

3. Plantear objetivos SMART para planificar de mejor forma cómo se alcanzarán

4. Redactar los requisitos antes de iniciar con la ejecución del proyecto, pues es parte la planificación para diseñar un software de calidad.

5. Crear los escenarios suficientes para considerar en su totalidad los flujos que tendrá la aplicación.

6. Es necesario priorizar las User Stories en el Product Backlog para determinar cuáles son necesarias de implementar primero y tener en cuenta su complejidad, riesgo y repetición.

7. Se recomienda realizar el Application Wireframes porque permite plasmar las ideas del negocio, así como relacionar mejor los servicios a ofrecer con el usuario. También, ayuda a mejorar la interfaz del usuario y su diseño.

8. Se recomienda centrarse primero en el DDD estratégico, pues se la lógica del dominio es primordial para desarrollar un software de calidad.

9. Se recomienda elaborar el modelo C4 para organizar la lógica del dominio en diseño de alto nivel y bajo nivel.

10. Se recomienda dividir el dominio en Bounded Context para reducir la complejidad y realizar las pruebas unitarias por separado.

11. Se recomienda realizar los diagramas de componentes y agregados para cada módulo.

12. Se recomienda realizar los diferentes diagramas de código siguiendo un orden para facilitar el trabajo. De esta manera, garantizar que todo esté relacionado y sin confusiones entre diagramas.

13.Los diagramas de estados que han sido redactados en su momento son de aquellas clases que resaltan las reglas del negocio aplicadas al proyecto. Por lo tanto, es necesario un asesoramiento con respecto a la redacción y con ello pulir aún más tales diagramas, ya que son los que darán sentido a la programación de la aplicación que deseamos ofrecer al público.

14.Para los diagramas de flujo se podrían realizar más diagramas y relacionarlos entre sí, es decir cuando acaba un proceso inicia otro.

15.Se recomienda aplicar patrones empresariales en donde las clases del dominio se mantengan agnósticas a la tecnología, pues se permitirá separar la base de datos y utilizar Repositories.

REFERENCIAS BIBLIOGRAFICAS

- Freeman. (2021). *Essential TypeScript 4: From Beginner to Pro*. Recuperado de https://upc.alma.exlibrisgroup.com/leganto/readinglist/lists/88656_32190003391?institute=51UPC_INST&auth=LOCAL [Consulta: 10 de noviembre de 2022].
- Grynhaus, Hudgens, J., Hunte, R., Morgan, M., & Stefanovski, W. (2021). *The TypeScript Workshop: A Practical Guide to Confident, Effective TypeScript Programming*. Recuperado de https://upc.alma.exlibrisgroup.com/leganto/readinglist/lists/88656_32190003391?institute=51UPC_INST&auth=LOCAL [Consulta: 10 de noviembre de 2022].
- Institut de Publique Sondage d'Opinion Secteur (IPSOS). (2016). *Mascotas 2016*. Lima: IPSOS. Recuperado de <https://www.ipsos.com/sites/default/files/2017-03/Mascotas%202016.pdf> [Consulta: 28 de agosto de 2022].
- Khononov. (2019). *What is domain-driven design?* (First edition.). Recuperado de https://upc.alma.exlibrisgroup.com/leganto/readinglist/lists/88656_32190003391?institute=51UPC_INST&auth=LOCAL [Consulta: 9 de noviembre de 2022].
- Kouraklis. (2019). *Introducing Delphi ORM Object Relational Mapping Using TMS Aurelius* (1st ed. 2019.). Recuperado de https://upc.alma.exlibrisgroup.com/leganto/readinglist/lists/88656_32190003391?institute=51UPC_INST&auth=LOCAL [Consulta: 6 de noviembre de 2022].
- La República. (26 de mayo de 2019). El 62.4% de los hogares limeños tienen una mascota. *La República*. Recuperado de <https://larepublica.pe/sociedad/808674-el-624-de-los-hogares-limenos-tienen-una-mascota/> [Consulta: 27 de agosto de 2022].
- Martin. (2008). *Clean code: a handbook of agile software craftsmanship*. Recuperado de

https://upc.alma.exlibrisgroup.com/leganto/readinglist/lists/8865632190003391?institute=51UPC_INST&auth=LOCAL [Consulta: 8 de noviembre de 2022].

Morgan, Samaroo, A., Williams, P., Hambling, B., & Thompson, G. (2019). *Software testing: an ISTQB-BCS certified tester foundation guide*. Recuperado de https://upc.alma.exlibrisgroup.com/leganto/readinglist/lists/8865632190003391?institute=51UPC_INST&auth=LOCAL [Consulta: 7 de noviembre de 2022].

Revista TicNews. (22 de octubre de 2013). Diagrama de Estados [Archivo de video]. Recuperado de <https://www.youtube.com/watch?v=hbWfk -57qc&t=239s> [Consulta: 16 de septiembre de 2022].

Theo Despoudis. (2021). *TypeScript 4 Design Patterns and Best Practices*. Recuperado de https://upc.alma.exlibrisgroup.com/leganto/readinglist/lists/8865632190003391?institute=51UPC_INST&auth=LOCAL [Consulta: 10 de noviembre de 2022].

Tsui, & Bartlet. (2013). *Essentials de ingeniería de software 3rd (tercera)*. Recuperado de https://upc.alma.exlibrisgroup.com/leganto/readinglist/lists/8865632190003391?institute=51UPC_INST&auth=LOCAL [Consulta: 9 de noviembre de 2022].

Vlad Khononov. (2021). *Learning Domain-Driven Design*. Recuperado de https://upc.alma.exlibrisgroup.com/leganto/readinglist/lists/8865632190003391?institute=51UPC_INST&auth=LOCAL [Consulta: 9 de noviembre de 2022].

ANEXOS

Enlace 1.

https://miro.com/app/board/uXjVPc0f16c=/?share_link_id=781532666646

Enlace 2.

https://miro.com/app/board/uXjVPSPC750=/?share_link_id=417659250046

Enlace 3.

https://miro.com/app/board/uXjVPG5cn9Y=/?share_link_id=489401383276

Enlace 4.

https://miro.com/app/board/uXjVPGHQjNY=/?share_link_id=551923833554

Enlace 5.

https://miro.com/app/board/uXjVPGdcD10=/?share_link_id=748743413219

Enlace 6.

<https://structurizr.com/share/77664/c678eb13-e266-45ad-bc81-3f6f3dc079e1>

Enlace 7.

https://lucid.app/lucidchart/54eb17c4-aae8-4f80-a28c-d06a29885cbd/edit?viewport_loc=-989%2C-27%2C3689%2C1842%20&invitationId=inv_60d28e7e-60c6-4c54-b558-ef776602469d#

Enlace 8.

https://lucid.app/lucidchart/da0a618b-1a56-45a4-8f7b-a2097e2d4cd3/edit?viewport_loc=-899%2C-96%2C3614%2C1805%20&invitationId=inv_b6213427-ad43-448c-a2f9-d865df54f29c#

Enlace 9.

https://lucid.app/lucidchart/4efb027c-bb42-43e1-b28e-256aeb25375e/edit?viewport_loc=-548%2C1350%2C2724%2C1360%20&invitationId=inv_bfa0c566-32cb-409a-a99b-c1408171067a#

Enlace 10.

<https://petsolution.my.canva.site/petcare>

Enlace 11.

<https://github.com/CarloLSG/PetCare-DDD>