

**DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
CARRERA DE INGENIERÍA DE SOFTWARE**

**TRABAJO DE INTEGRACIÓN CURRICULAR, PREVIO A LA OBTENCIÓN
DEL TÍTULO DE INGENIERA DE SOFTWARE**

TEMA:

**IMPLEMENTACIÓN DE LOS MICROSERVICIOS ORIENTADOS AL
AGENDAMIENTO DE TURNOS, CITAS Y SEGURIDADES DEL SISTEMA
USANDO EL PARADIGMA DE LÍNEA DE PRODUCTO SOFTWARE (LPS)
ENFOCADO A UN DESARROLLO CO-LOCALIZADO (DGS)**

AUTORAS:

**ROMO GAVILANEZ, PAOLA LISSETTE
SÁNCHEZ PICO, ANA LISSETTE**

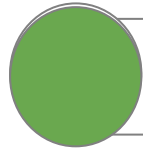
DIRECTOR:

Dr: JACOME GUERRERO, PATRICIO SANTIAGO

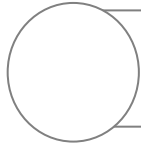
LATACUNGA FEBRERO, 2023



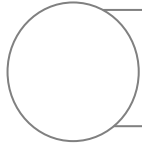
Contenido



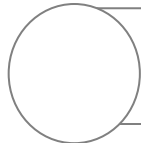
Planteamiento del Problema



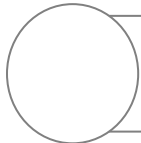
Fundamentación Teórica



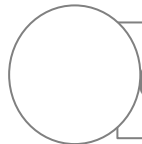
Metodología



Desarrollo del Sistema



Validación del Sistema



Conclusiones y Recomendaciones

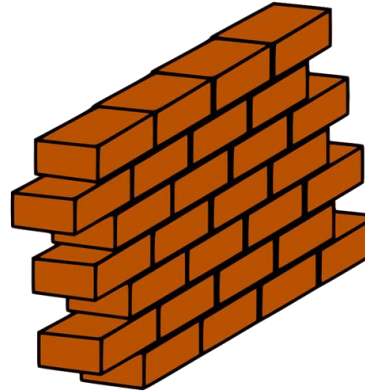


Planteamiento del problema

- Recolección de información.
- Falta de organización de los expedientes físicos.
- Registros no legibles por la calidad de la caligrafía.
- Pérdida de carnets
- Pérdida de clientes por procesos lentos.
- Sistemas costosos.



Justificación e importancia



Plan Nacional del Buen Vivir (2017 - 2021), objetivo número 3

Construcción de nuevos productos disminuyendo costos, tiempo y recursos.

Generar software que cumpla sus necesidades específicas y generales, con un costo accesible



Objetivos

General:

Implementar los microservicios orientados al agendamiento de turnos, citas y seguridades del sistema.

Específicos:

- Revisión del estado del arte sobre Línea de Producto Software (LPS), arquitectura de microservicios y desarrollo co-localizado.
- Elaborar la línea de producto de software basada en un análisis de dominio en clínicas veterinarias.
- Implementar los microservicios orientados al agendamiento de turnos, citas, y seguridades del sistema.
- Validación de los microservicios orientados al agendamiento de turnos, citas, y seguridades del sistema.



Hipótesis

La utilización de línea de producto software (LPS) para desarrollar software, permite implementar soluciones que se ajustan al requerimiento de las clínicas veterinarias.



Contenido

- Planteamiento del Problema
- **Fundamentación Teórica**
- Metodología
- Desarrollo del Sistema
- Validación del Sistema
- Conclusiones y Recomendaciones



Línea de Producto Software (LPS)

Definición:

Es un paradigma que propone la reutilización planificada, para aprovechar elementos comunes y variables que tienen productos de software que pertenecen a un mismo dominio.

Características :

- Permite crear diversas variantes del software utilizando una base común.
- Está diseñada y adaptada para cumplir con necesidades específicas del cliente o mercado.
- El proceso de construcción consta de tres fases: ingeniería de dominio, ingeniería de aplicación, gestión de la configuración como se muestra en la Figura 1.



Figura 1: Actividades esenciales para el desarrollo de LPS

Análisis de Dominio Orientado a Características (FODA)

Definición:

Es una técnica usada en LPS para identificar y analizar las características más importantes o dominantes que posteriormente se convertirán en funcionalidades del sistema.

Características:

- Representa los diferentes componentes del dominio y la relación que existe entre ellos.
- Las características pueden ser de tipo opcional u obligatorio en base a los objetivos del cliente como se muestra en la Figura 2.

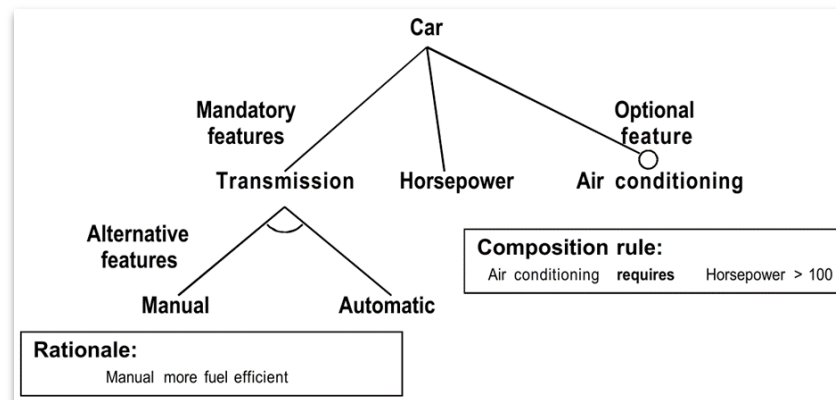


Figura 2: Ejemplo de diagrama de características mostrando la composición de un auto

Microservicios

Definición:

Es un enfoque de diseño de software en el que un sistema se divide en múltiples servicios independientes los cuales realizan una función específica dentro del sistema.

Características:

- Permite una mayor flexibilidad y escalabilidad.
- Están contruidos alrededor de funcionalidades de negocio y con independencia de despliegue.

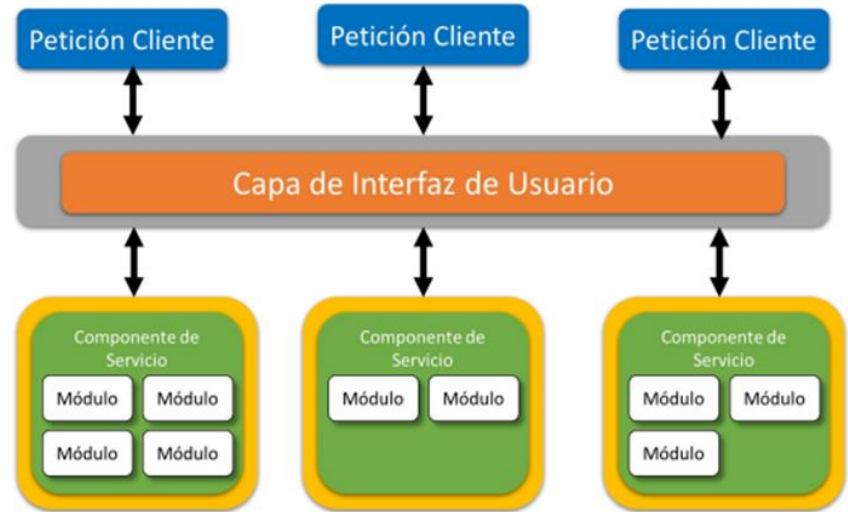


Figura 3: Ejemplo de arquitectura de microservicios.

Desarrollo Global de Software (DGS)

Definición:

Es el proceso de desarrollar aplicaciones de software utilizando un equipo distribuido que se encuentra en diferentes regiones geográficas o países.

Características:

- Se requiere un marco sólido de comunicación y gestión de proyectos, herramientas y técnicas de colaboración eficaces
- Existe una rama denominada DGS co-localizado, el mismo abarca grupos de trabajo dentro de una zona geográfica.



Figura 4: Representación Gráfica del Desarrollo Global de Software



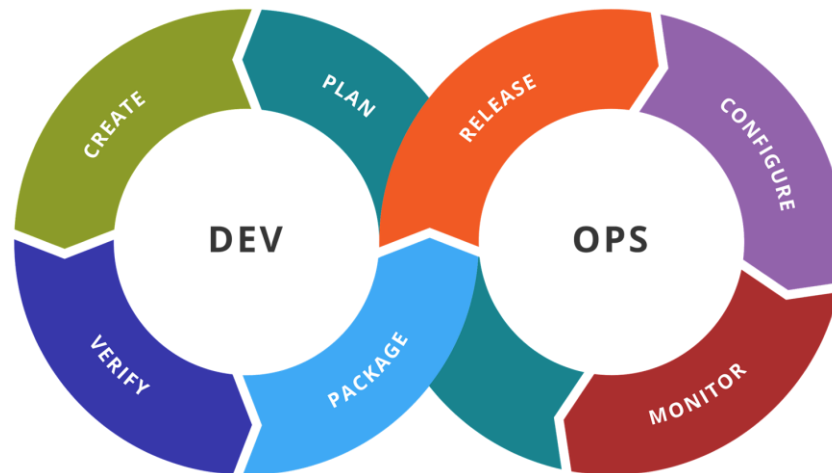
DevOps

Definición:

Enfoque de desarrollo de software que combina el desarrollo de software (**Dev**) y las operaciones de tecnología de la información (**Ops**) para permitir un desarrollo y una entrega de software más rápidos y eficientes.

Características:

- Se centra en la automatización de la entrega de software y los cambios de infraestructura, con énfasis en la colaboración, la comunicación y la integración entre los equipos de desarrollo y operaciones..



Marco Ágil Escalado - SAFe

Definición:

SAFe es un marco que proporciona un conjunto de prácticas y herramientas para que las organizaciones escalen metodologías ágiles en toda su empresa.

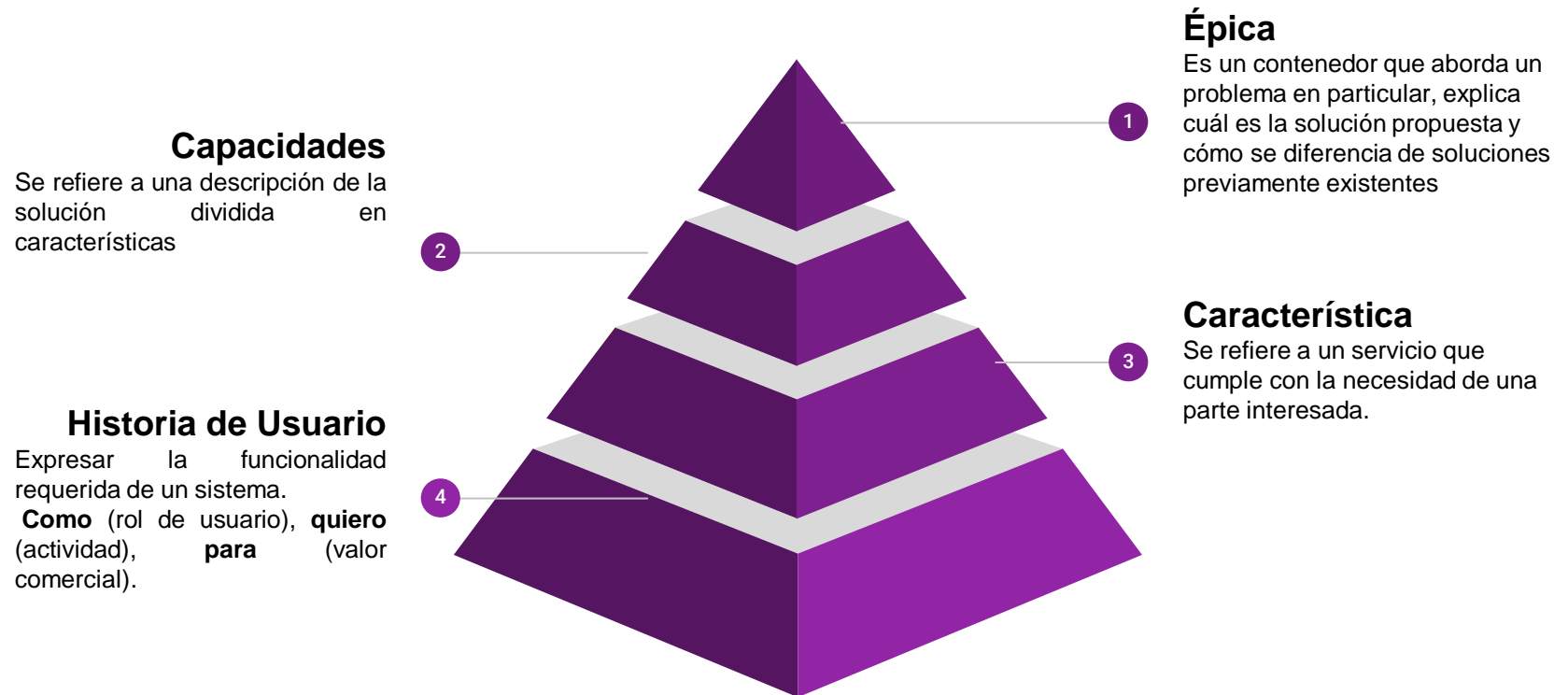


Figura 6: Niveles jerárquicos de SAFe



Metodología SCRUM

Definición:

Es un marco ágil para administrar y completar proyectos complejos, especialmente en el desarrollo de software.

Características:

- Divide proyectos complejos en piezas de trabajo más pequeñas y manejables, que luego se planifican, ejecutan y revisan en iteraciones cortas llamadas sprints.
- Cada sprint comienza con una sesión de planificación de sprint, donde el equipo planifica el trabajo que se completará durante el sprint.
- Product Owner, Scrum Master y Development Team, y artefactos Product Backlog, Sprint Backlog .

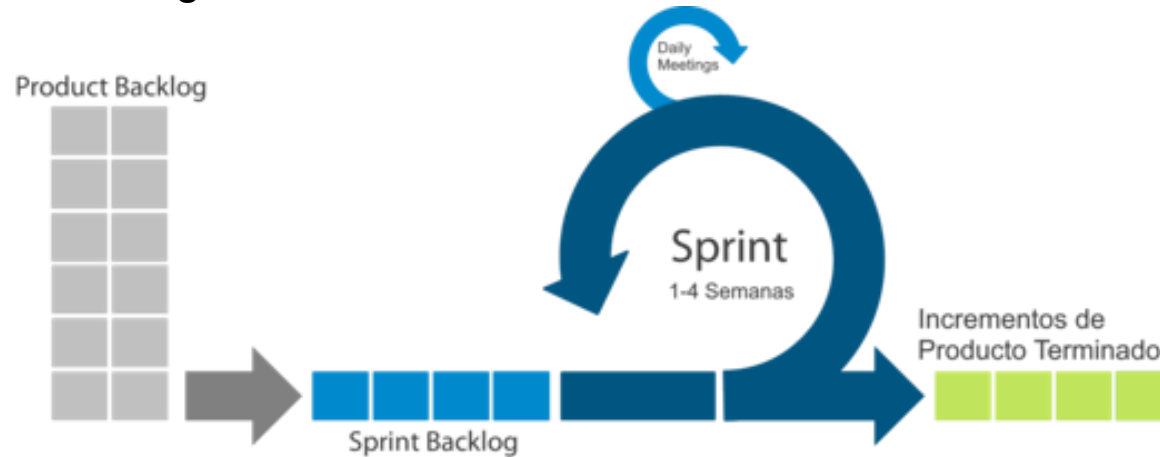


Figura 7: Representación de la metodología SCRUM



Autenticación del Sistema con Identity Server

Definición:

Es un sistema de autenticación y Autorización basado en token, permite implementar el inicio de sesión único y el control de acceso de aplicaciones web modernas además de API que usen protocolos como OpenID Connect y OAuth2

Características:

- Es el responsable de crear un servicio de autenticación completo con una entrada y salida de sesión única para varios tipos de aplicaciones

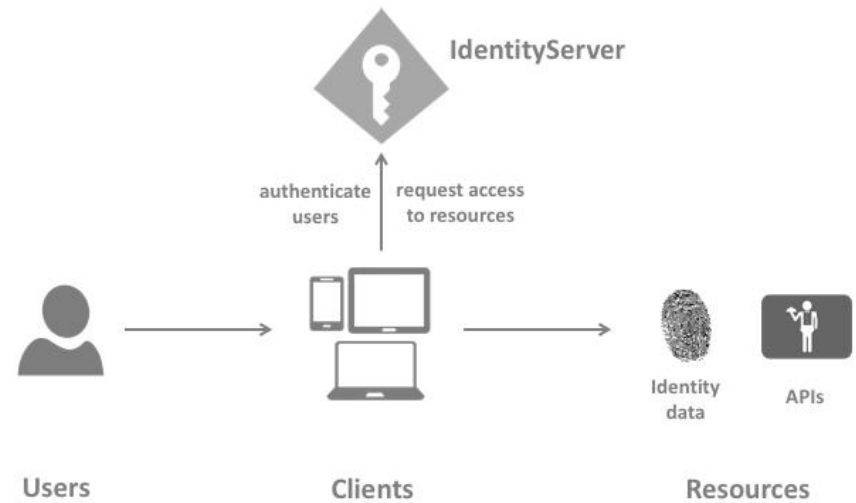


Figura 8: Flujo de trabajo de Identity Server

Contenido

○ Planteamiento del Problema

○ Fundamentación Teórica

● Metodología

○ Desarrollo del Sistema

○ Validación del Sistema

○ Conclusiones y Recomendaciones



Metodología - Asignación de equipos y roles

Equipo	Integrantes	Localidad	Responsabilidades
A	- Julio Castro - José Poveda	Ambato	Desarrollo del front end del sistema
B	- Martín Camacho - José Núñez	Ambato – Latacunga	Desarrollo del back end del sistema y despliegue
C	- Paola Romo - Ana Sánchez	Ambato - Latacunga	Desarrollo del back end y seguridades



Metodología - Asignación de equipos y roles

Rol	Integrante	Función
Product Owner	Dr. Santiago Patricio Jácome Guerrero	Determina el cumplimiento de las actividades del proyecto y garantizar que el producto se entregue funcional, correctamente validado y que cumpla con los requerimientos establecidos.
Scrum Máster	Paola Lissette Romo Gavilánez	Encargado de que los equipos de trabajo alcancen los objetivos hasta llegar a la fase final del sprint.
Team Development	Ana Lissette Sánchez Pico Paola Lissette Romo Gavilánez	Personas encargadas del análisis, diseño, desarrollo, despliegue y pruebas del sistema



Contenido

- Planteamiento del problema y objetivos
- Fundamentación teórica
- Metodología
- **Análisis y diseño del sistema**
- Desarrollo y validación del sistema
- Conclusiones y recomendaciones



1 Entrevistas

Servicios generales (Royal-Hound)

- Consultas
- Vacunas
- Desparasitaciones
- Cirugía, traumatología, y neurocirugía
- Hospitalización
- Ambulancias
- PetGrooming
- Venta de accesorios y alimento

Servicios generales (Vital-Pet)

- Consultas
- Hospitalización
- Laboratorio Clínico
- Radiografías
- Cirugías
- Venta de accesorios
- Farmacia
- PetGrooming
- Vacunación

Servicios generales (Cannopolis)

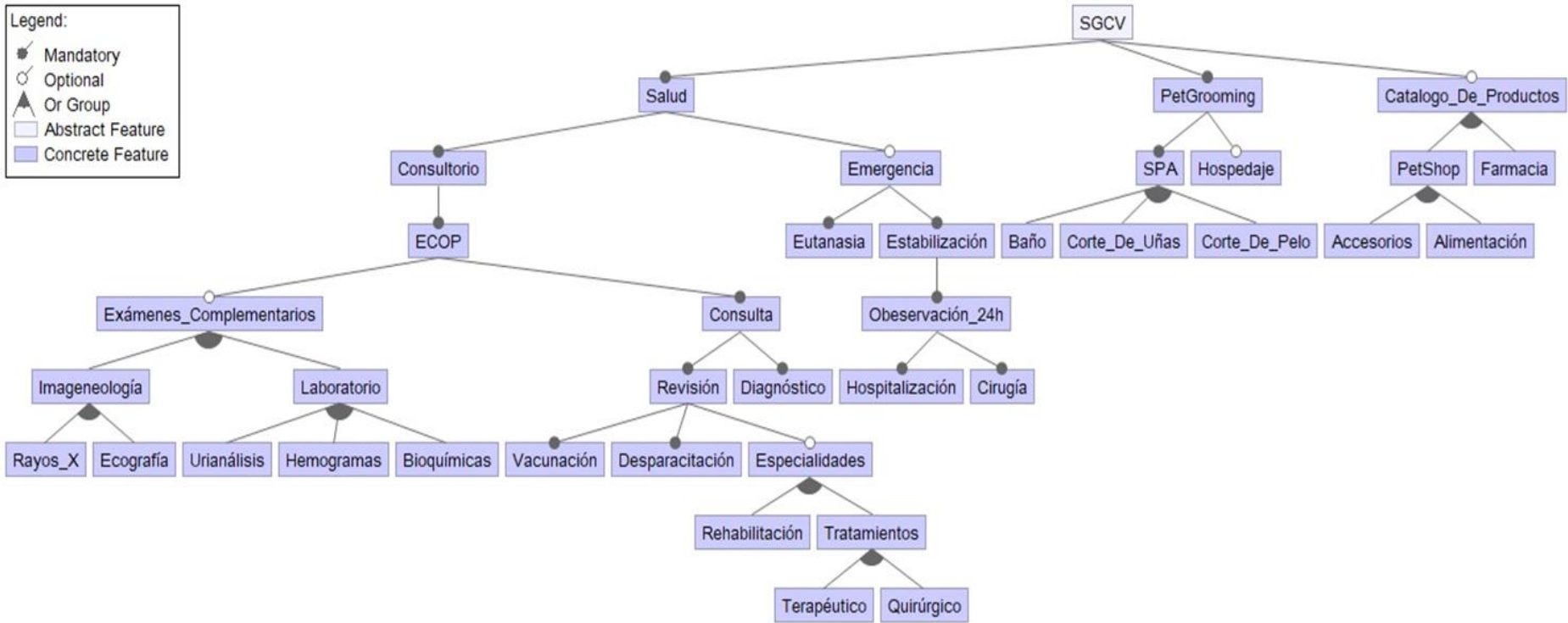
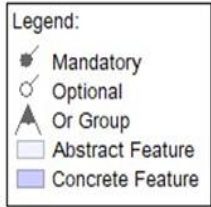
- Consulta
- Cirugía general
- Petshop
- Farmacia
- Laboratorio
- Ecografías
- Planes de vacunación
- Hospitalización

Servicios generales (PETS-CLI-LAB)

- Tienda
- Consultorio
- Hospitalización
- PetGrooming
- Eutanasia
- Administracion de seguimiento de medicamentos
- Hospedaje



2 Diagrama de Características



Observación_24h ⇒ Especialidades

Especialidades ⇒ Exámenes_Complementarios

Terapéutico ⇒ Hospitalización

Quirúrgico ⇒ Cirugía

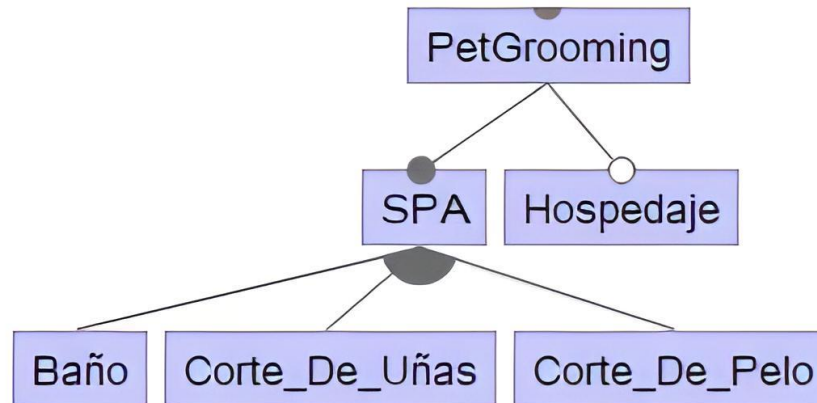
ECOP: Examen clínico orientado a problemas.

PetGrooming: Cuidado estético de mascotas

PetShop: Tienda de mascotas



PetGrooming



PetGrooming: Cuidado estético de mascotas

PetShop: Tienda de mascotas

Análisis del sistema - Implementación de SAFe

3

Una vez realizado el FODA, se aplica SAFe y se obtienen las épicas, capacidades, características e historias de usuario que tendrá el sistema.

Épica	Características	Historias de Usuario
PetGrooming	Gestión Paciente - Tutor	Gestión Tutor
		Gestión Paciente
	Agendamiento de Turno	Gestión Turno
	Spa	Gestión Spa
	Hospedaje	Gestión Habitación
		Gestión Huésped
		Gestión Ficha Registro
	Autenticación	Autenticación del sistema

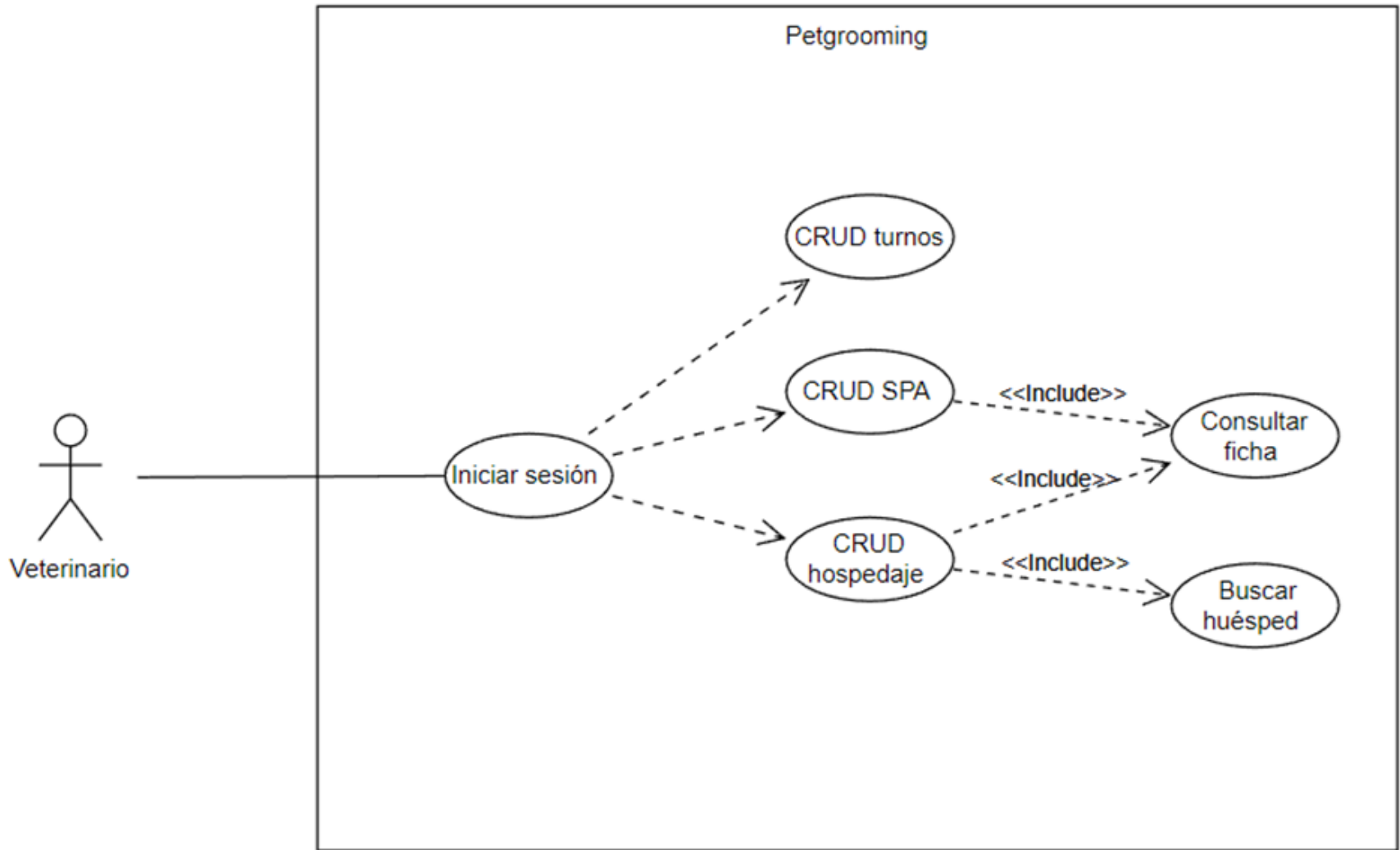


Análisis del sistema - Product Backlog

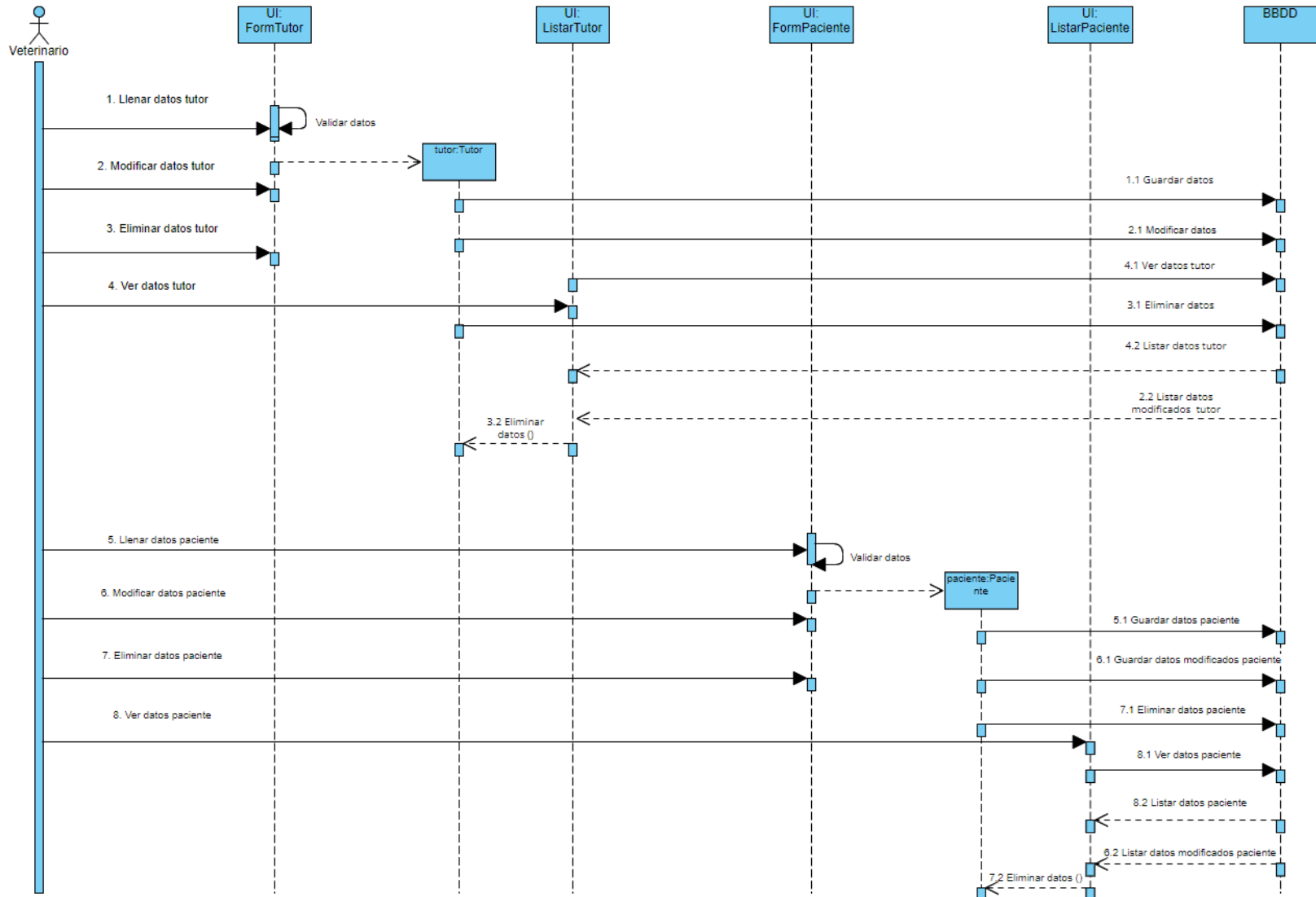
Historia de Usuario	Estimación	Fecha Inicio	Fecha Fin	Sprint	Actividades Realizadas
HU001	8	7/10/2022	14/10/2022	1	- Creación de Tablas en base de datos
HU002	8	20/10/2022	27/10/2022	1	- Creación de Repositorio y ramas
HU009	5	2/11/2022	6/11/2022	2	- Conexión de base de datos
HU010	5	15/10/2022	19/11/2022	2	- Creación del proyecto e instalación de dependencias
HU011	3	28/11/2022	30/12/2022	2	- Desarrollo de rutas, controladores.
HU012	3	12/12/2022	14/12/2022	2	- Creación de validaciones.
HU013	3	19/12/2022	21/12/2022	2	- Pruebas en Postman.
HU015	13	6/1/2023	18/1/2023	3	- Despliegue de microservicios.
					- Pruebas de Despliegue
					- Instalación de Framework Identity Server
					- Configuración de protocolo oauth2.0.



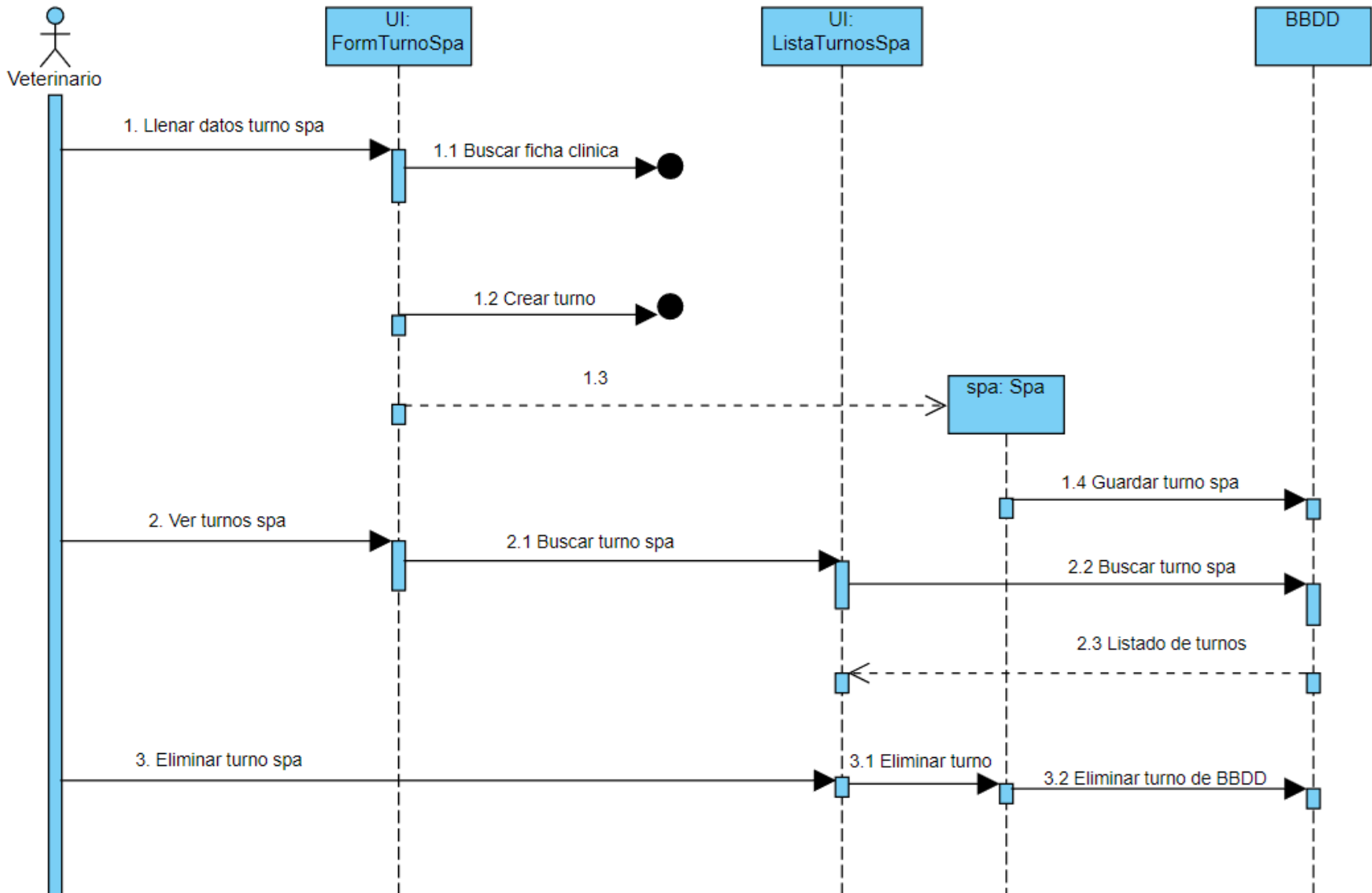
Diseño del sistema - Diagrama de casos de uso



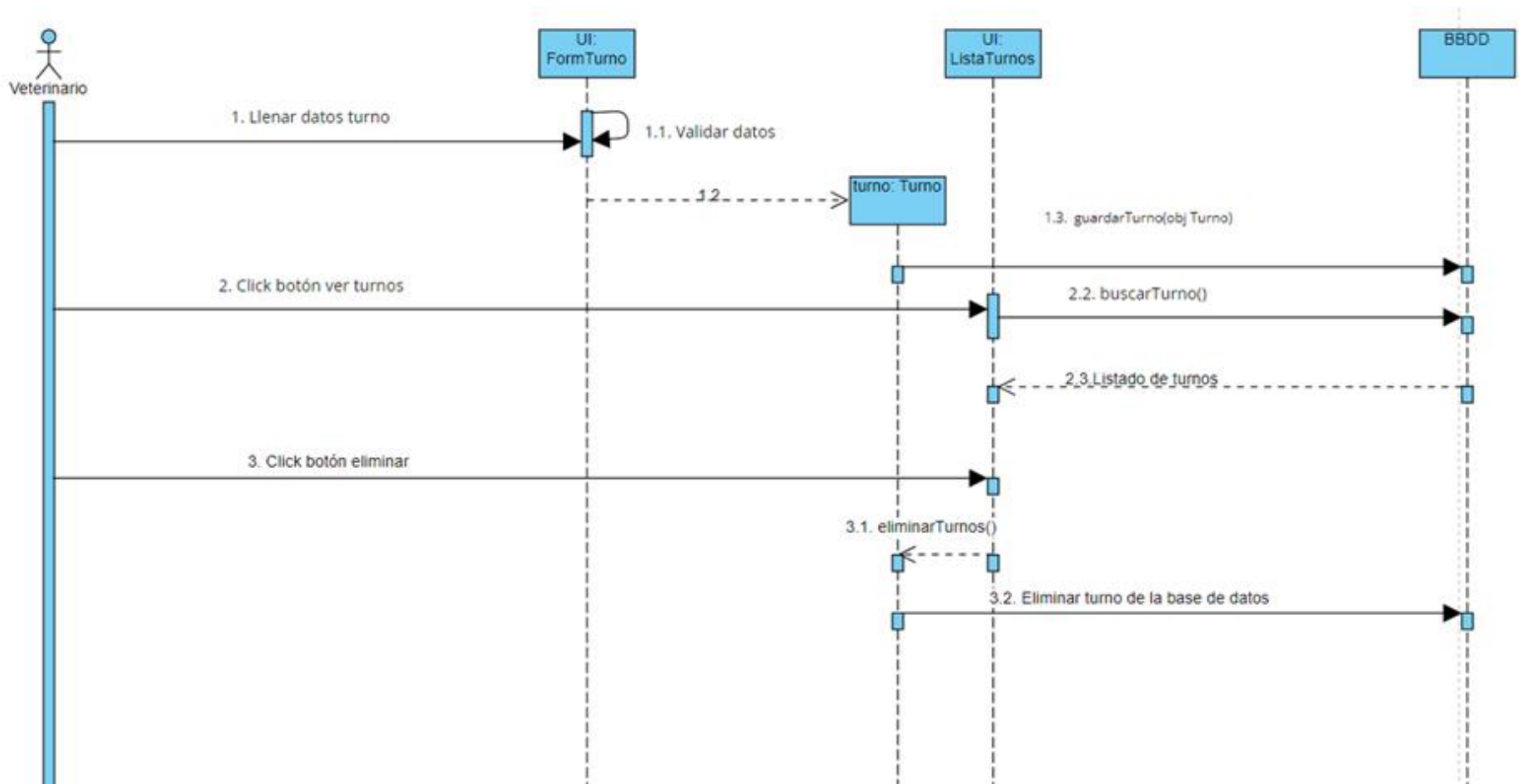
Diseño del sistema - Diagrama de secuencia Paciente Tutor



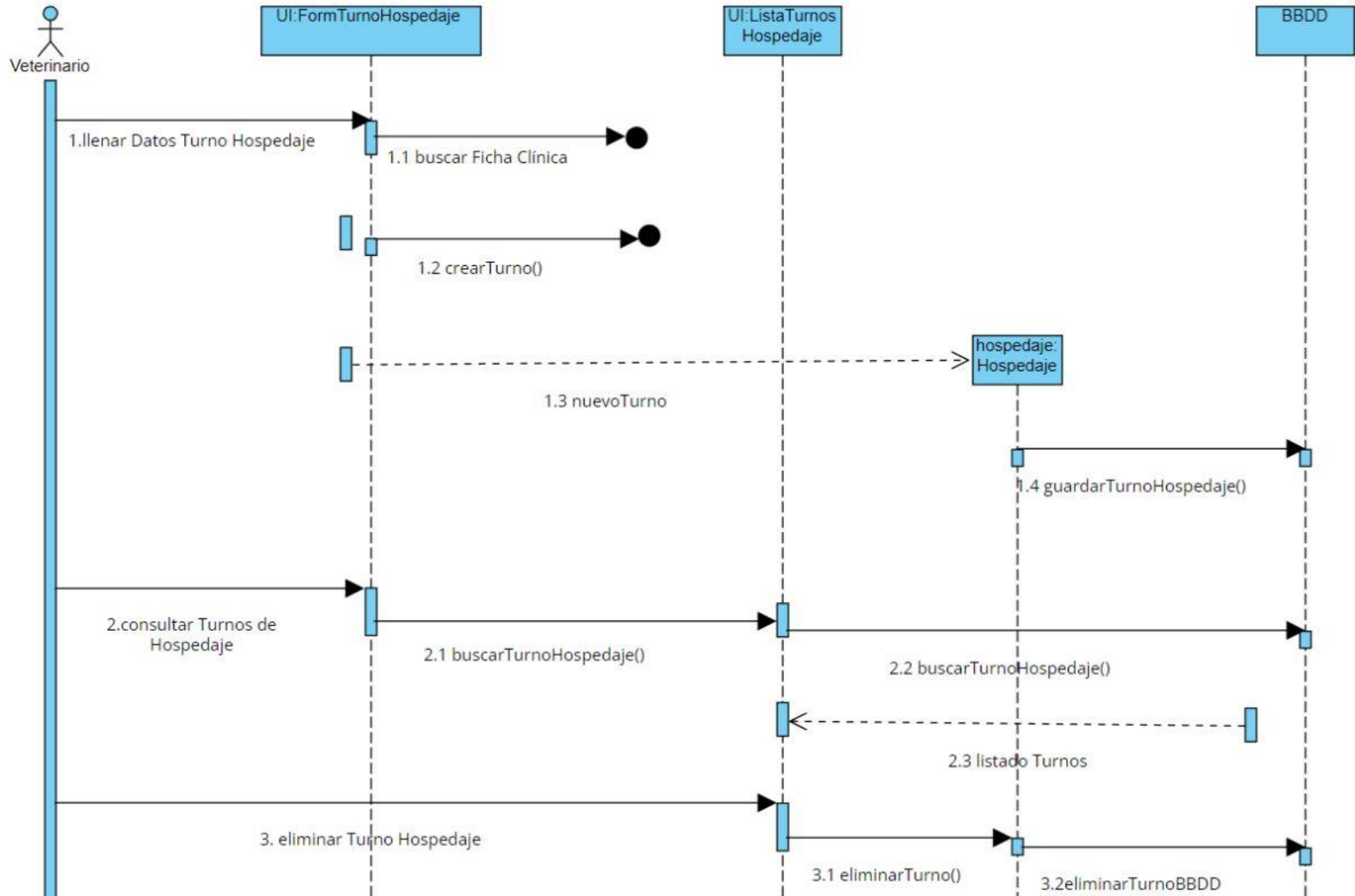
Diseño del sistema - Diagrama de secuencia SPA



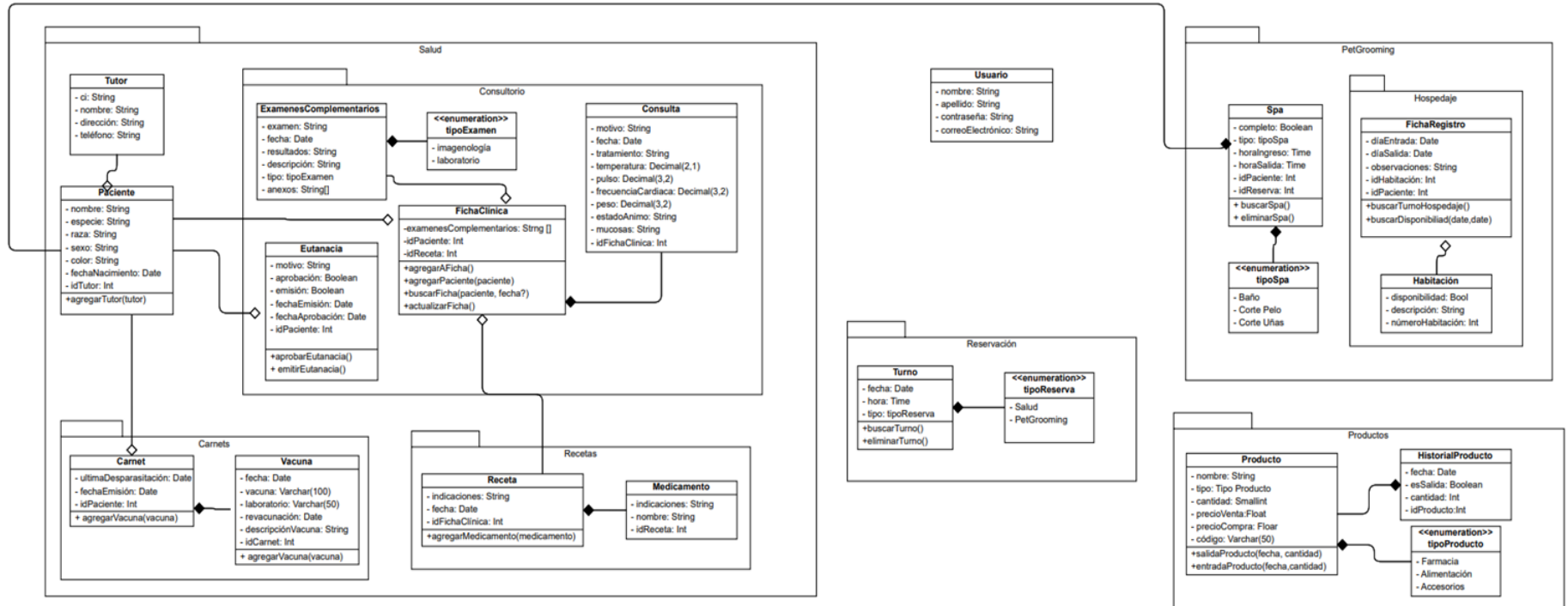
Diseño del sistema - Diagrama de secuencia Agendamiento Turno



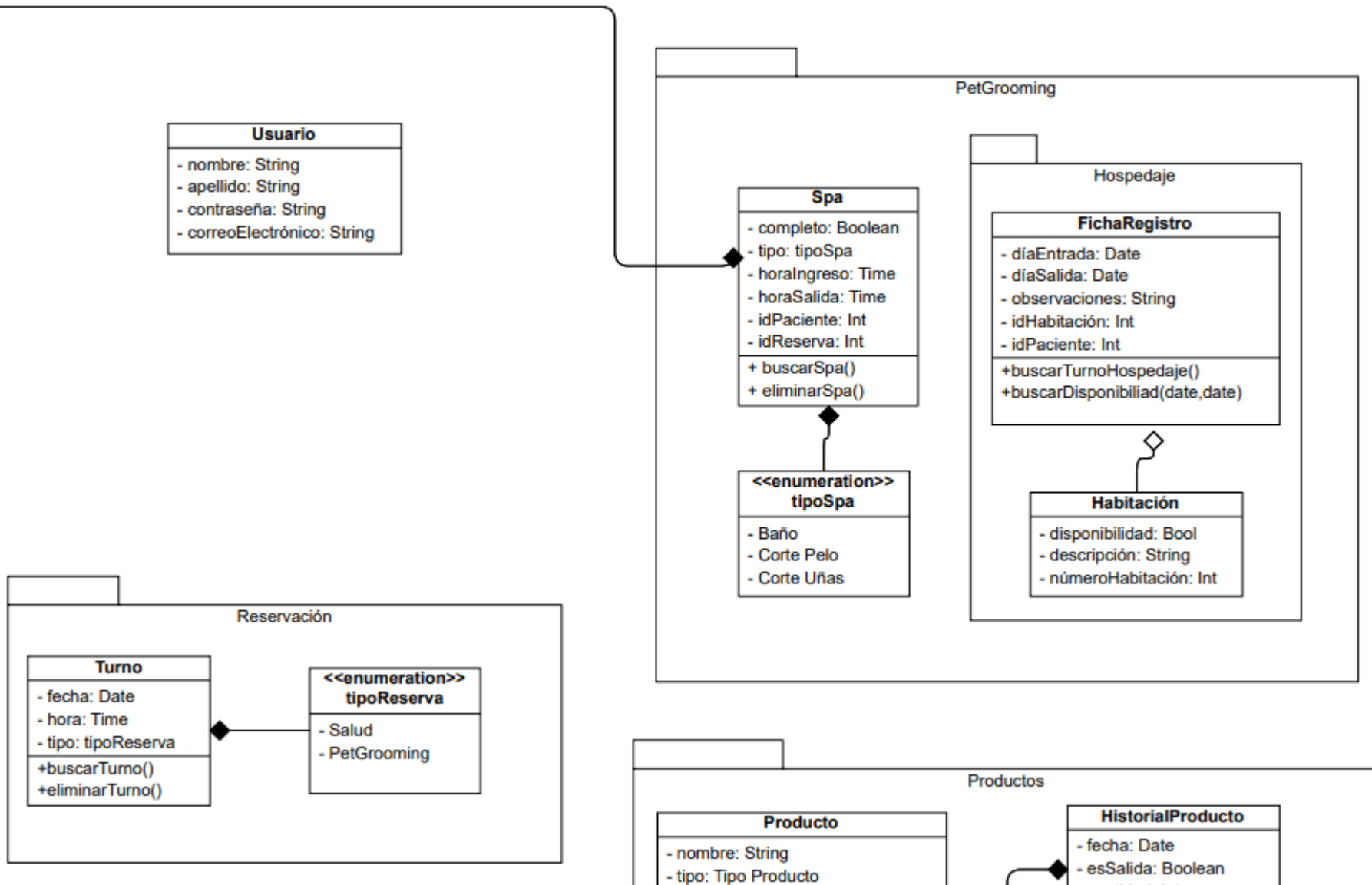
Diseño del sistema - Diagrama de secuencia Hospedaje



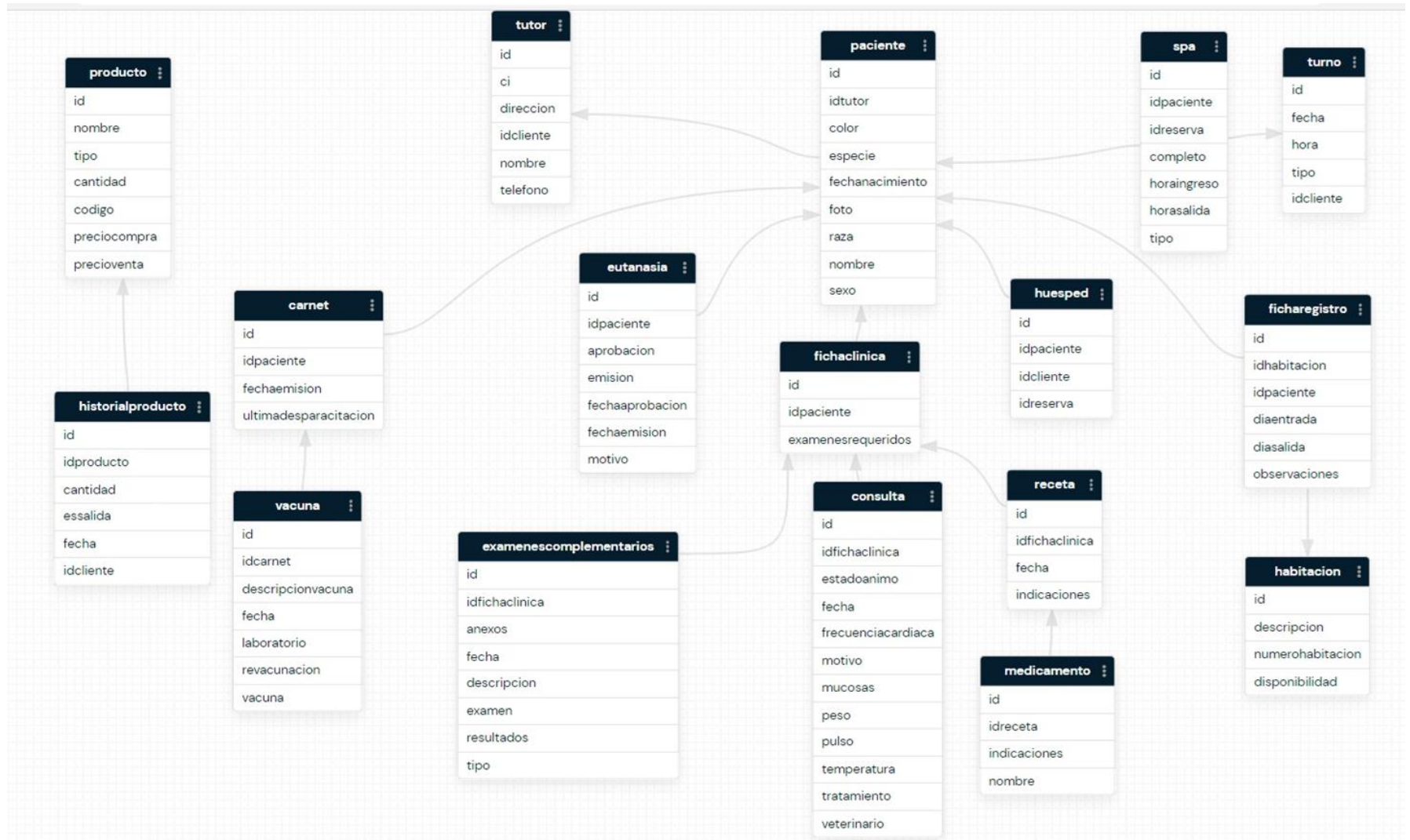
Diseño del sistema - Diagrama de clases



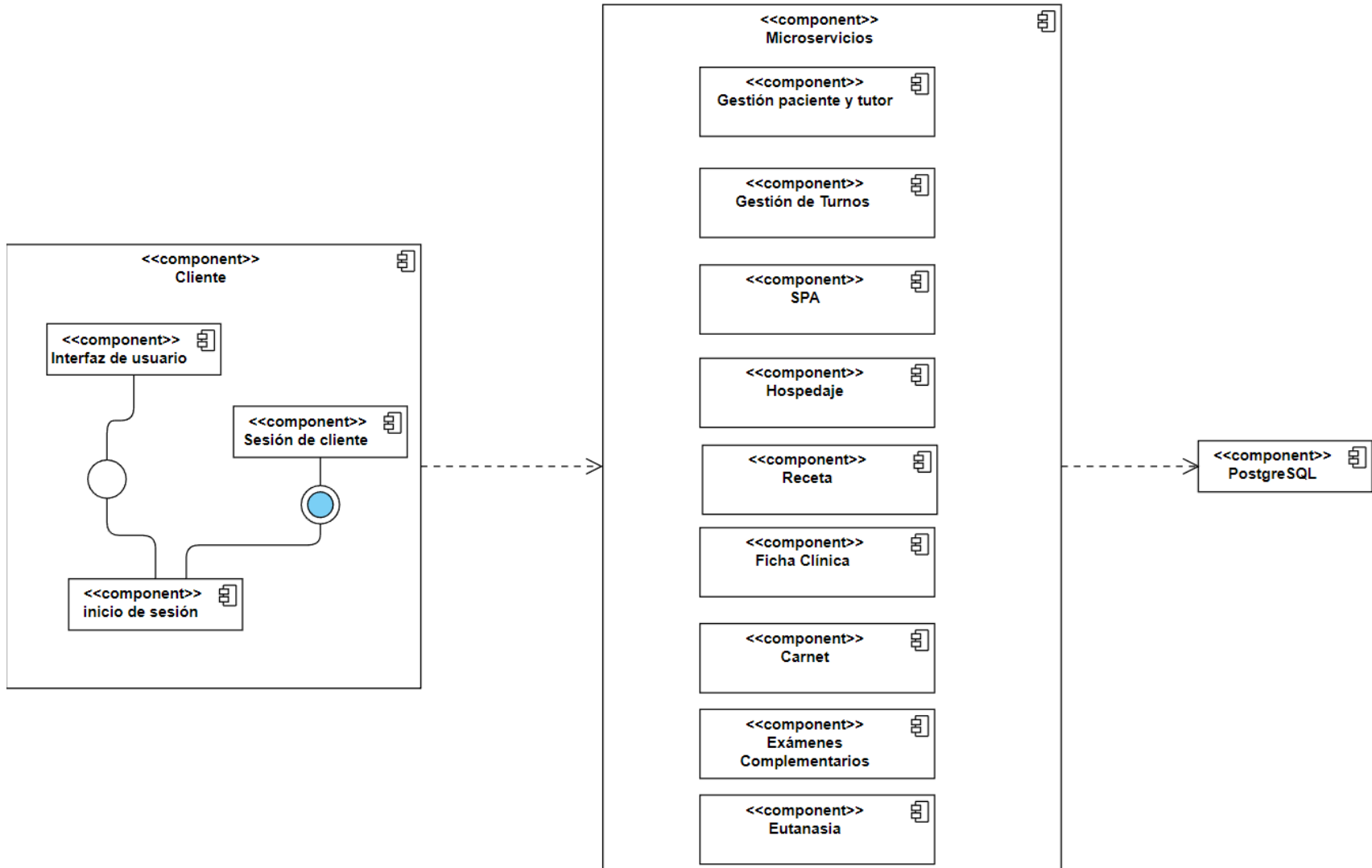
Diseño del sistema - PetGrooming, Reserva y Usuario



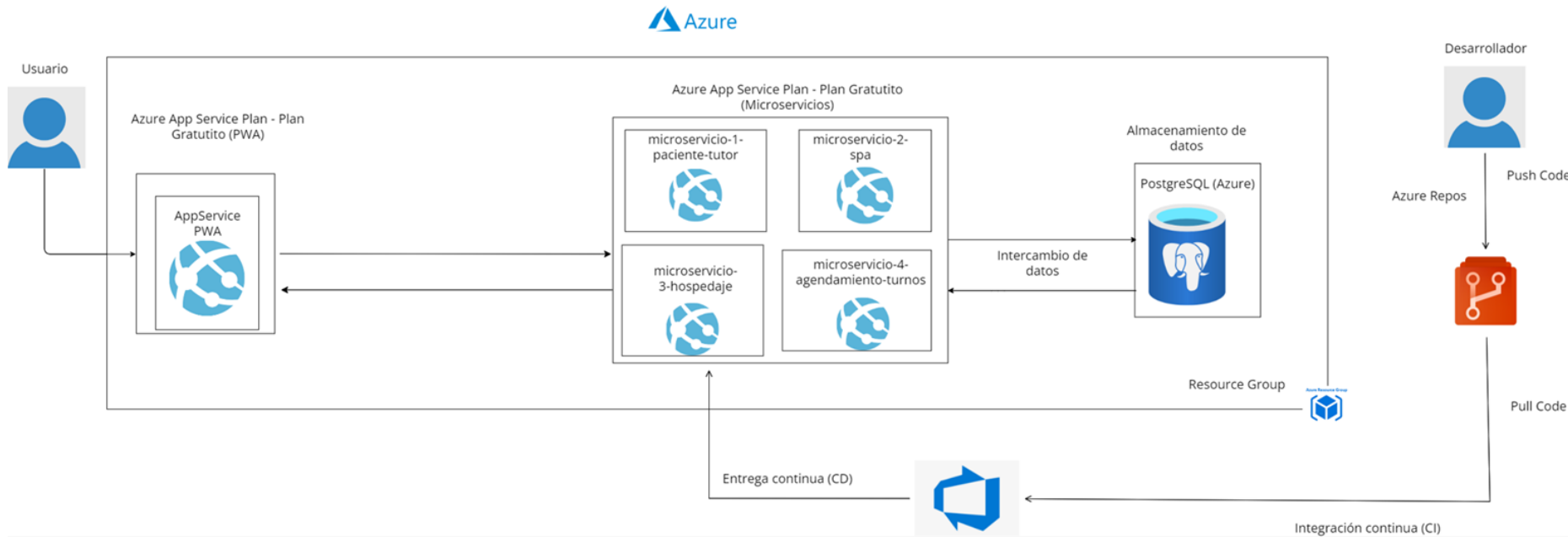
Diseño del sistema - Modelo Entidad Relación



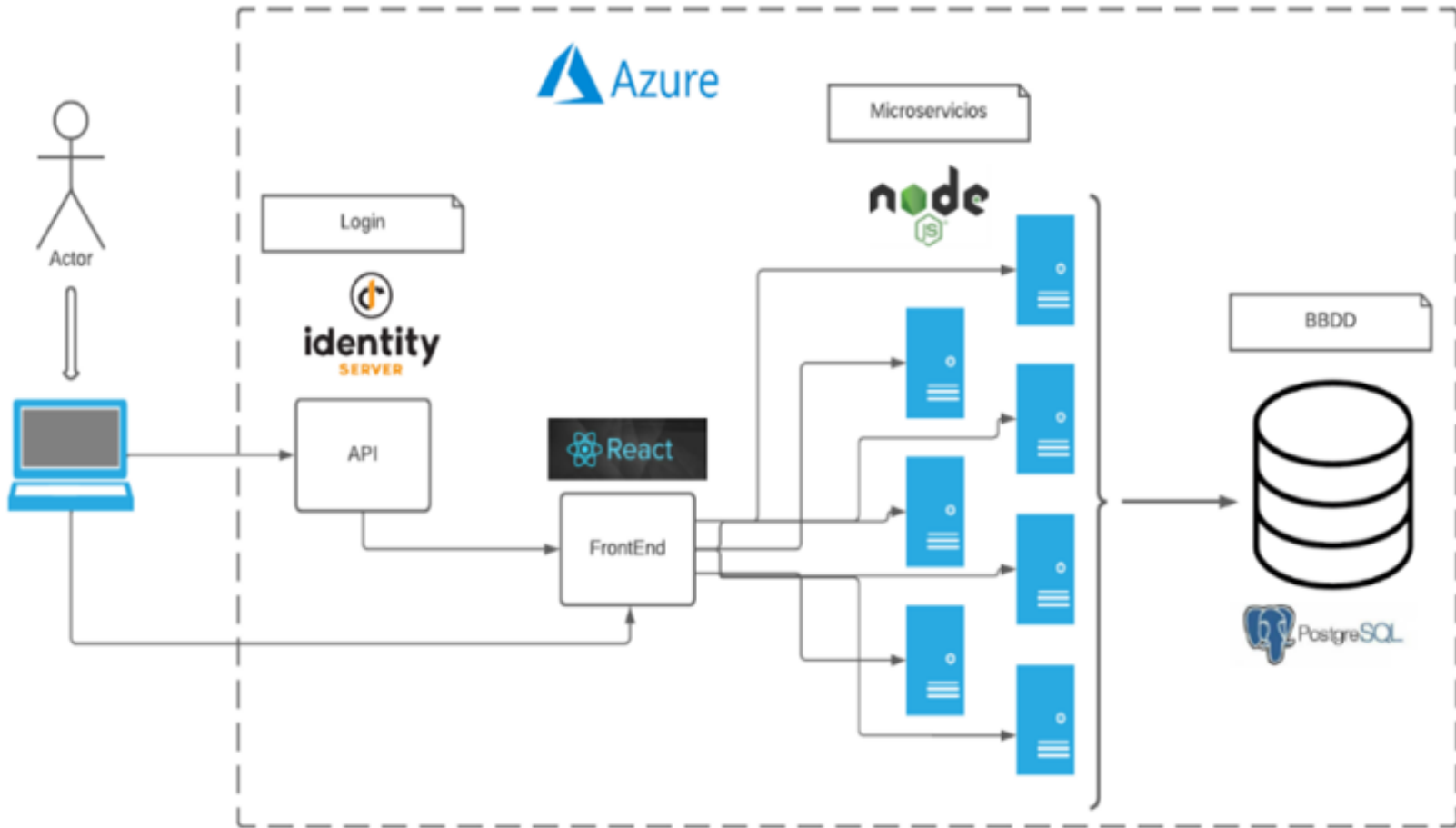
Diseño del sistema - Diagrama de componentes



Diseño del sistema - Diagrama de despliegue



Arquitectura



Contenido

- Planteamiento del problema y objetivos
- Fundamentación teórica
- Metodología
- Análisis y diseño del sistema
- Desarrollo y validación del sistema
- Conclusiones y recomendaciones



Herramientas de desarrollo

- Node JS
- Express Validator
- GitFlow
- Azure DevOps
- Azure



Microservicios

- Postman
- Locust



Pruebas

- PostgreSQL
- Liquibase



Base de Datos

- C#



Autenticación



Gestión de la base de datos sprint 1 & 2

1

Clonar el repositorio de la base de datos

2

Se crean las ramas correspondientes a cada microservicio.

3

Sincronizar los cambios con la rama develop.

4

Subir los cambios a la base de datos con “**Liquibase Update**”.

```
develop / changelog.sql
changelog.sql
Contents History Compare Blame
293 --changeset paola.romo:15 labels:agregar-tabla-agendamiento-turnos context:41-h42
294 --comment: se agrega la tabla de agendamiento de turnos
295
296 CREATE TYPE tipoReserva AS ENUM ('Salud', 'PetGrooming');
297
298 create table turno (
299     id INT GENERATED ALWAYS AS IDENTITY,
300     fecha date,
301     hora time,
302     tipo tipoReserva,
303     idcliente int
304 );
305 --rollback drop table turno;
306
307 --changeset paola.romo:17 labels:agregar-tabla-agendamiento-turnos-clave-primaria context:41-h42
308 --comment: se agrega clave primaria a la tabla de agendamiento de turnos
309
310 ALTER TABLE turno add primary key (id);
311
312 --rollback ALTER TABLE turno add primary key id
313 --changeset ana.sanchez:18 labels:agregar-tabla-spa context:45-h46
314 --comment: se agrega la tabla de spa
315
316 CREATE TYPE tipoSpa AS ENUM ('Baño', 'Corte Pelo', 'Corte Uñas');
317
318 create table spa (
319     id INT GENERATED ALWAYS AS IDENTITY,
320     idPaciente int not null,
321     idReserva int not null,
322     completo boolean,
323     tipo tipoSpa,
```



Desarrollo de microservicios sprint 1 & 2

Crear un repositorio en Azure DevOps por cada microservicio.

1

Crear las ramas de acuerdo a las historias de usuario

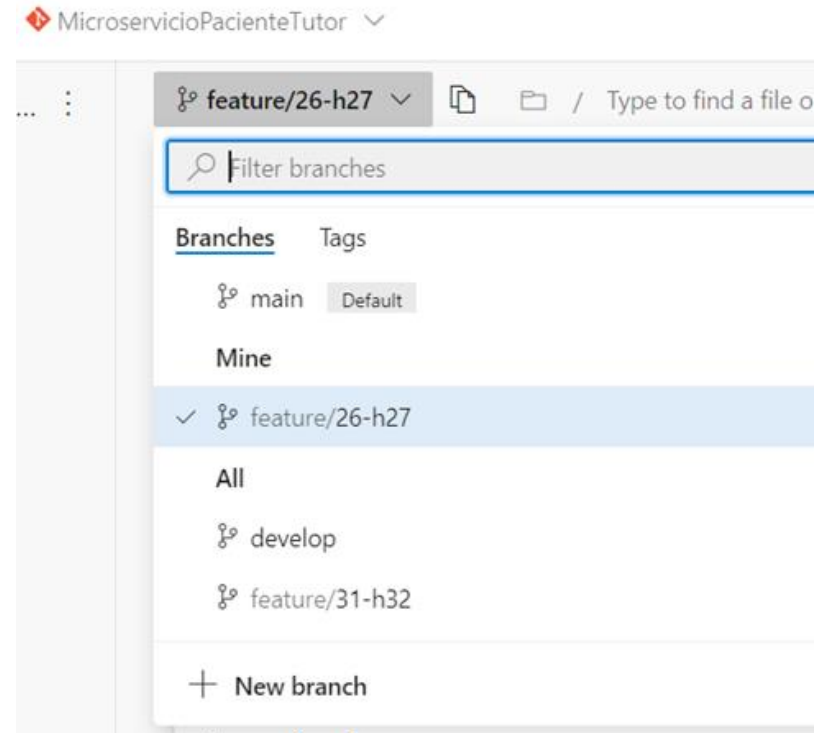
2

Crear rutas, controladores, middlewares, validaciones e índices

3

Sincronizar las ramas con la rama develop.

4



Despliegue de microservicios sprint 1 & 2

Se prepara el entorno, en el archivo index se configura el puerto **app.listen (process.env. PORT || 8181)**. En el archivo package json se pone **"start": "node ./src/index.js"**

1. Configuración

En Azure se selecciona el grupo de recursos SGCV, y en buscar se selecciona aplicación web.

3. Creación de App Service

2. Sincronización de Ramas

Se sincronizan las ramas develop hacia la rama main.

4. Configuración de App Service

Se llenan los datos requeridos que son asignarle un nombre al microservicio, escoger el lenguaje de programación y el sistema operativo, el plan de pago y la zona horaria.



Validación de los microservicios - Postman sprint 1 & 2

1

Desplegar los
microservicios en
Azure

2

Validar la funcionalidad
de todas las rutas de
acuerdo a las
peticiones HTTP: GET,
POST, PUT, DELETE.

3

En caso de ser
necesario validar que
se muestran los
mensajes requeridos
en caso de existir
error.



Validación de los microservicios - Locust sprint 1 &2

La herramienta se encargará de las pruebas de rendimiento. Para las mismas primero se crea un archivo con las peticiones GET y POST, este archivo llamado locustfile.py, se lo corre en la terminal con el comando locust. Se pedirá que se haga las pruebas de ingreso de 10 usuarios cada 1 segundo. Locust empezará con las pruebas y enviará una cantidad masiva de datos. Se genera un reporte.

```
Get Started locustfile.py X
locustfile.py > Tutor > crear_tutores
1 from locust import HttpUser, task
2
3 class Tutor(HttpUser):
4     @task
5     def listar_tutores(self):
6         self.client.get("/tutores")
7     @task
8     def listar_tutores_porID(self):
9         self.client.get("/tutores/5")
10    @task
11    def crear_tutores(self):
12        self.client.post("/tutores", json={
13
14            "nombre": "Andrea Jimenez",
15            "ci": "0501724306",
16            "direccion": "Ambato, Huchi Chico",
17            "telefono": "0991452428",
18        })
```

Start new load test

Number of users (peak concurrency)

Spawn rate (users started/second)

Host (e.g. http://www.example.com)

Advanced options



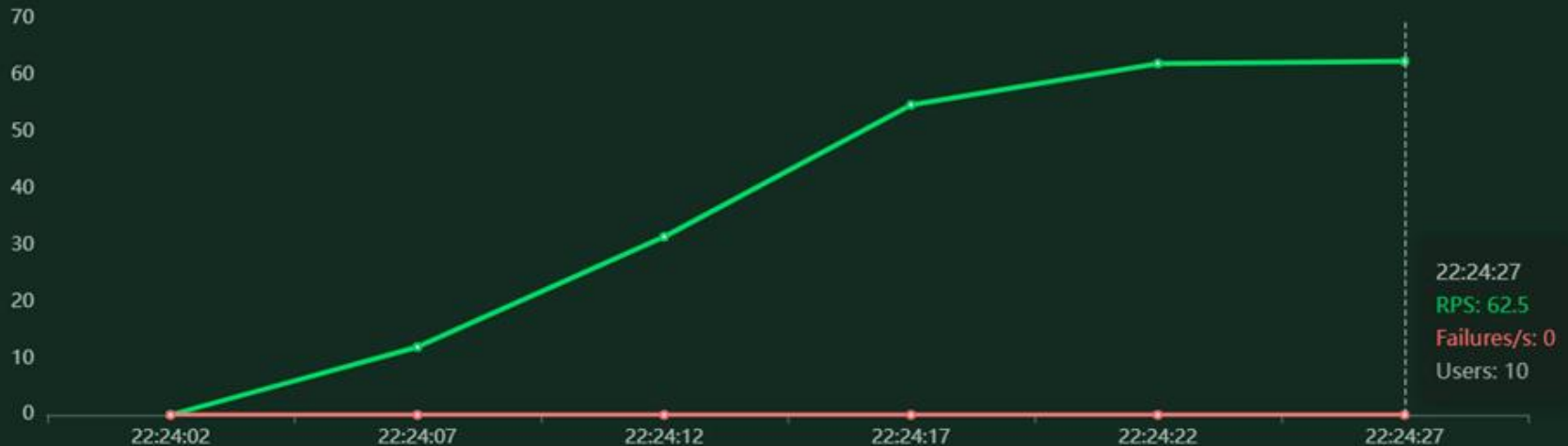
Pruebas de rendimiento - Locust sprint 1 & 2

Request Statistics

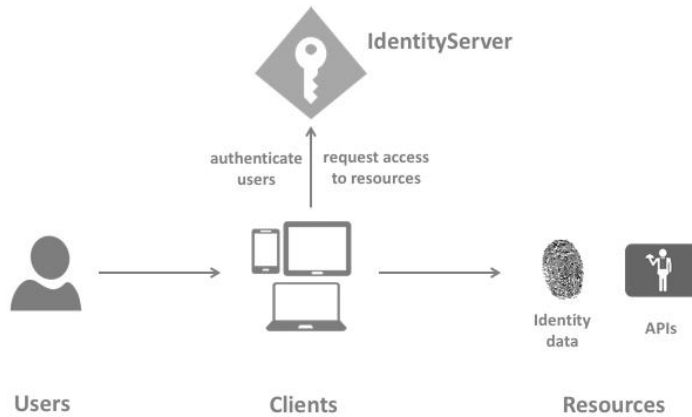
Method	Name	# Requests	# Fails	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	RPS	Failures/s
POST	/tutores	475	0	170	142	1314	26	15.9	0.0
GET	/tutores/5	513	0	159	135	539	112	17.2	0.0
GET	/tutores	518	0	172	140	740	32071	17.4	0.0
Aggregated		1506	0	167	135	1314	11077	50.5	0.0

Total Requests per Second

● RPS ● Failures/s



Sprint 3



Iniciar Sesión

Error
• Invalid username or password

Cuenta Local

Nombre de Usuario

El nombre de usuario es la primera parte del correo (previo al @).

Contraseña

Mantener sesión activa

Welcome to Duende IdentityServer (version 6.2.0)

IdentityServer publishes a [discovery document](#) where you can find metadata and links to all the endpoints, key material, etc.
[Click here](#) to see the claims for your current session.
[Click here](#) to manage your stored grants.
[Click here](#) to view your pending CIBA login requests.
Here are links to the [source code repository](#), and [ready to use samples](#).

Registrarse

Error

- La Contraseña debe tener al menos una letra minúscula.
- La Contraseña debe tener al menos una letra mayúscula.
- La Contraseña debe tener al menos un caracter especial:
!@#%\$^&*()_+~\|{}:;<>|/?,-

Crear un nuevo Usuario

Nombres *

Ana

Apellidos *

Sanchez

Correo electrónico *

alsanchez5@espe.edu.ec

Contraseña *

Contraseña

Confirmar contraseña *

Vuelva a escribir la contraseña

Escoja las características adicionales: *

Exámenes complementarios

Exámenes complementarios

Hospedaje

Hospedaje



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

Listas de chequeo



- Mediante la técnica de listas de chequeo, se verifica el cumplimiento de los requerimientos planteados ya que cada criterio de aceptación de las historias de usuario planteadas se convierte en una prueba.
- Las pruebas realizadas evidencian el cumplimiento de los requisitos planteados previamente como historias de usuario.



Contenido

- Planteamiento del problema y objetivos
- Fundamentación teórica
- Metodología
- Análisis y diseño del sistema
- Desarrollo y validación del sistema
- Conclusiones y recomendaciones



CONCLUSIONES

Se desarrollaron los microservicios de agendamiento de turnos, citas y seguridades pertenecientes al SGCV, se determina que el análisis realizado con LPS utilizando FODA, permite tener una mejor visión de las necesidades de las clínicas veterinarias. La utilización de SAFe permitió que las características obtenidas, se implementen a través de microservicios de manera correcta, clara y ordenada.

Al trabajar aplicando Desarrollo de Software Global co-localizado se debe organizar aplicando una metodología enfocada a equipos de trabajo en este caso se aplicó SCRUM lo que permitió realizar el desarrollo de planificada en base a los tiempos de cada sprint, este trabajo lo facilitó la herramienta Azure DevOps.

Se utilizó GitFlow para el control del flujo de trabajo, esto ayudó a que se sincronice de mejor manera las ramas creadas por cada miembro del equipo, permitiendo así optimizar la integración del código.

El modelado UML aplicado en el proyecto permitió apreciar de forma visual el comportamiento y la estructura del sistema.



CONCLUSIONES

Al realizar un desarrollo global de software co-localizado se necesitaba una base de datos robusta que permita gestionar cambios de varias personas al mismo tiempo por lo cual se eligió el gestor de base de datos PostgreSQL ya que se adapta a la tecnología Liquibase la cual permite sincronizar la base de datos y mantener un registro de los cambios que se realizan entre los desarrolladores.

Al aplicar las pruebas de rendimiento se pudo comprobar que los microservicios generados, responden de manera rápida ante una carga masiva de datos. Logra estabilizarse evitando así los cuellos de botella. Se obtienen unos microservicios estables y funcionales.

Al aplicar las pruebas de funcionalidad se concluye que los microservicios cumplen con los requisitos planteados y funcionan de manera correcta.

Al aplicar las listas de chequeo se comprueba que la LPS cumple con los requerimientos de los usuarios a través de soluciones que se ajustan a las necesidades de las clínicas veterinarias..



RECOMENDACIONES

Es recomendable utilizar el paradigma LPS cuando se requiere generar software que gestione múltiples procesos ya que a partir de la reutilización de elementos comunes se pueden generar varios productos en un menor tiempo de producción, con la reducción de recursos y costes al generar nuevos productos

Se recomienda aplicar LPS en equipo de alta afinidad ya que la comunicación es un punto clave durante todo el desarrollo ya que siempre existirá dependencia de unos con otros.

En el desarrollo se recomienda utilizar software para el control de versiones como Gitflow para llevar un control de los cambios que se realizan en el código y para una mejor integración. Se recomienda crear ramas para cada característica, una rama develop para la integración y la rama main que será la rama de producción y desplegarlo sin inconvenientes.



RECOMENDACIONES

Se recomienda la utilización de Azure DevOps para una mejor gestión de las actividades que se va a realizar durante el desarrollo de cada sprint

Se recomienda la utilización de SAFe ya que nos permite modelar de forma jerárquica las características comunes y específicas, facilitando la obtención de los requerimientos a través de las historias de usuario.

Se recomienda la utilización de la metodología SCRUM ya que para el proceso de construcción de la LPS se requiere de equipos para un desarrollo planificado lo cual se obtiene mediante el uso de sprint, los cuales requieren una planificación mediante el product backlog, además posee los roles y funcionalidades bien definidos lo cual evita ambigüedades al momento de distribuir el trabajo.



Gracias!



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA