**Sistema de Gestión y Agendamiento de Servicios de Jardinería Clean & Garden**

**(DAS) Documento Arquitectura Sistema**

**Versión 1.0.4**

**Identificación de Documento**

| **Identificación** |  |
| --- | --- |
| **Proyecto** | Sistema de Gestión y Agendamiento de Servicios de Jardinería Clean & Garden |
| **Versión** | 1.0.4 |

| **Documento mantenido por** | Marcela Gonzalez |
| --- | --- |
| **Fecha de última revisión** | 03/10/2025 |
| **Fecha de próxima revisión** | 08/10/2025 |

| **Documento aprobado por** | Alvaro Morales |
| --- | --- |
| **Fecha de última aprobación** | 08/10/2025 |

**Historia de Revisiones**

| **Fecha** | **Versión** | **Descripción** | **Autor** |
| --- | --- | --- | --- |
| 29/09/2025 | 1.0.1 | Se completa información de introducción,metas y restricciones de la arquitectura,d[ecisiones de diseño y selección de alternativas](#_heading=h.6fdp1eqtzbc8),a[nálisis de reutilización](#_heading=h.qgyjcdz1k24l). | Marcela Gonzalez |
| 01/10/2025 | 1.0.2 | Se completa Vista física | Marcela Gonzalez |
| 03/10/2025 | 1.0.3 | Se completa Vista de Escenarios | Marcela Gonzalez |
| 08/10/2025 | 1.0.4 | Se completa Vista lógica y puntos incompletos del documento | Vicente Zapata y Álvaro Morales |
|  |  |  |  |

**Tabla de Contenidos**

[**1**](#_heading=h.mfwyqztv1olp) **Introducción 3**

[1.1](#_heading=h.kwoppq8n1z5) Contexto del Problema 3

[1.2](#_heading=h.3vglxd23p6u6) Propósito 3

[1.3](#_heading=h.reebgxv6jttb) Ámbito 3

[1.4](#_heading=h.n7w2coiy0lij) Definiciones, acrónimos y abreviaciones 3

[1.5](#_heading=h.q8w38vg6cf6b) Referencias 3

[1.6](#_heading=h.lj4dlmt2md2n) Resumen ejecutivo 3

[1.7](#_heading=h.k4kdisfjahca) Representación 4

[**2**](#_heading=h.vnvzd4txuoip) **Metas y Restricciones de la Arquitectura 5**

[2.1](#_heading=h.pznqzrl6b13v) Metas de la arquitectura 5

[2.2](#_heading=h.3kfwg5hdgp35) Restricciones de la Arquitectura 5

[2.3](#_heading=h.ekf9psx863ro) Otros antecedentes y consideraciones 5

[**3**](#_heading=h.j4q55wlhwst4) **Vista de Escenarios 6**

[3.1](#_heading=h.wu44eqr0h8bo) Modelo de Casos de Uso 6

[3.2](#_heading=h.ffsrkidpadjl) Casos de Usos Extendidos 6

[3.3](#_heading=h.jdsdudybfq4l) Especificación de los Escenarios de Calidad Relevantes 7

[**4**](#_heading=h.20md54croa4y) **Vista de Procesos 8**

[**5**](#_heading=h.3w4uloce7o4r) **Vista Lógica 9**

[5.1](#_heading=h.p118lpgbokg) Parte Estructural ( Diagrama de Clases y Diagrama Relacional) 9

[*5.1.1*](#_heading=h.bti1r9z07tz) *Descripción de Clases 9*

[*5.1.2*](#_heading=h.eq2r7zy4vxca) *Descripción de Tablas 10*

[5.2](#_heading=h.a4afujx4cou5) Parte Dinámica (Diagrama de Secuencias) 11

[**6**](#_heading=h.qdubvkjcawbx) **Vista de Desarrollo o Despliegue 12**

[**7**](#_heading=h.7b6knok1o7vr) **Vista Física 13**

[**8**](#_heading=h.6fdp1eqtzbc8) **Decisiones de Diseño y Selección de Alternativas 14**

[**9**](#_heading=h.qgyjcdz1k24l) **Análisis de Reutilización 15**

1. **Introducción**
   1. **Contexto del Problema**

Clean & Garden es una empresa familiar que ofrece variados servicios de jardinería y limpieza en la comuna Santo Domingo de la región de Valparaíso.

Actualmente, la gestión de los servicios se realiza de manera manual mediante un cuaderno y la comunicación con los clientes se realiza principalmente mediante llamadas o mensajes por whatsapp. Esta forma de operar genera múltiples problemas como son el riesgo de pérdida de información, duplicidad de reservas, dificultad para controlar demanda en temporadas de alta carga de trabajo y ausencia de un historial ordenado que facilite el seguimiento de clientes y servicios.

Además, la empresa carece de una presencia digital sólida que le permita mostrar su portafolio de trabajos y fortalecer la relación con nuevos clientes. En consecuencia, la falta de herramientas tecnológicas limita su capacidad de crecimiento, eficiencia operativa y competitividad frente a otras empresas que ya cuentan con soluciones digitales.

* 1. **Propósito**

El propósito del sistema es diseñar y desarrollar una plataforma tecnológica integral, compuesta por una aplicación web y una aplicación móvil, que permita a Clean & Garden modernizar la gestión de sus servicios y fortalecer la relación con sus clientes.

El sistema busca resolver las limitaciones actuales a través de funcionalidades que aseguren:

* Una agenda digital eficiente y libre de duplicidades.
* Un canal de comunicación centralizado con notificaciones automáticas.
* La posibilidad de exhibir un portafolio digital de trabajos realizados.
* La generación de reportes y estadísticas para apoyar la toma de decisiones.
* Una experiencia de usuario segura, confiable y accesible desde cualquier dispositivo.

De este modo, el sistema contribuirá directamente a la eficiencia interna de la empresa, a la fidelización de los clientes actuales y a la captación de nuevos clientes mediante una presencia digital profesional.

* 1. **Ámbito**

Clean & Garden ha decidido invertir en la modernización de sus procesos internos con el fin de mejorar la gestión de sus servicios de jardinería y limpieza, garantizando un control más eficiente de las agendas y fortaleciendo la comunicación con sus clientes. Para ello, se definió el desarrollo de una solución tecnológica que permita digitalizar las operaciones de la empresa, ofrecer una imagen más profesional frente al mercado y apoyar el crecimiento sostenido del negocio en un entorno cada vez más competitivo.

* 1. **Definiciones, acrónimos y abreviaciones**

| **ACRÓNIMO** | **DESCRIPCIÓN** |
| --- | --- |
| *DAS* | Documento de Arquitectura del Sistema |
| *ERS* | Especificación de Requerimientos del Sistema. Documento que detalla los requerimientos funcionales y no funcionales. |
| *API* | Conjunto de funciones o rutas que permiten la comunicación entre el frontend y el backend. |
| *CRUD* | Operaciones básicas de datos: Crear, Leer, Actualizar y Eliminar. |
| *BD (Base de datos)* | Repositorio donde se almacenan los datos del sistema. En este proyecto, PostgreSQL en Supabase. |
| *JWT (Json Web Token)* | Token usado para autenticar usuarios y mantener sesiones seguras. |

* 1. **Referencias**

A continuación, se listan las referencias a otros documentos :

* [**Documento ERS**](https://docs.google.com/document/d/1KWY7x9CvlATxloMtIo4n8zV-gUEZiW5U/edit?usp=sharing&ouid=112046847387388169157&rtpof=true&sd=true)
  1. **Resumen ejecutivo**

El presente documento expone los lineamientos técnicos, organizativos y de diseño para el desarrollo de la solución tecnológica de Clean & Garden. El proyecto tiene como propósito modernizar la gestión de servicios de jardinería y limpieza de la empresa, superando las limitaciones derivadas del uso de métodos manuales y comunicación fragmentada con los clientes.

La arquitectura propuesta busca establecer una plataforma integral, conformada por una aplicación web y una aplicación móvil, que garantice eficiencia operativa, seguridad en el manejo de la información y una experiencia de usuario accesible y confiable. Este documento describe el contexto del problema, los objetivos de la solución y las metas de calidad, incorporando consideraciones de desempeño, escalabilidad y mantenibilidad.

De este modo, el DAS se convierte en una guía fundamental para asegurar que las decisiones técnicas y de diseño estén alineadas con los objetivos estratégicos de Clean & Garden, entregando un sistema robusto que aporte valor al negocio y facilite su crecimiento sostenido en un mercado competitivo.

* 1. **Representación**

La arquitectura del sistema Sistema de Gestión y Agendamiento de Servicios de Jardinería Clean & Garden está representada siguiendo el enfoque del framework 4+1 y las recomendaciones del proceso unificado. Las vistas incluidas en esta versión del documento son:

* **Vista de Escenarios**: Describe los casos de uso más significativos, presenta los actores y una descripción de sus casos de uso asociados. De igual forma describe los escenarios de calidad más relevantes para la arquitectura.
* **Vista de Procesos**: Describe los procesos involucrados para darle sentido a la ejecución del sistema, así como sus relaciones de comunicación y sincronización.
* **Vista Lógica**: Describe la arquitectura del sistema presentando varios niveles de refinamiento. Indica los módulos lógicos principales, sus responsabilidades y dependencias.
* **Vista de Desarrollo o Despliegue**: Describe los componentes de deployment construidos y sus dependencias.
* **Vista Física**: Describe restricciones tecnológicas, normativas, estándares, etc., los cuales influyen sobre las decisiones arquitectónicas, del producto y del proceso de desarrollo.

1. **Metas y Restricciones de la Arquitectura**

A continuación, se revisan las metas y restricciones de la arquitectura.

* 1. **Metas de la arquitectura**

De acuerdo a las reuniones y al análisis de los requerimientos, se listan los principales conductores iniciales de la arquitectura los cuales corresponden a las metas arquitectónicas iniciales ( atributos de calidad)

* **Seguridad**: proteger información, credenciales y manejar autenticación segura.
* **Disponibilidad**: Tanto en web como en móvil, los usuarios (cliente, administrador, jardinero) deben acceder sin interrupciones al sistema.
* **Usabilidad**: Interfaces intuitivas → facilidad de uso en web y móvil.
* **Escalabilidad**: Capacidad de soportar más usuarios y servicios.
* **Interoperabilidad**:Integración con servicios externos (Flow para pagos, Gmail para recuperación de contraseña,confirmación de correo,notificaciones,etc y Expo Push).
* **Rendimiento**: Respuestas rápidas en visualización de portafolio ,manejo de citas/usuarios,etc tanto en web como en móvil.
* **Confiabilidad**:manejar errores y caídas de servicios externos
  1. **Restricciones de la Arquitectura**

Existen restricciones que han sido levantadas con los stakeholders, las cuales se presentan a continuación:

* **Tiempo de construcción:** se cuenta con un plazo de tiempo para su construcción, 4 meses y 1 semana según la planificación.
* **Tecnologías definidas:**
  + **El frontend web** debe realizarse con React + Next.js + TypeScript + Tailwind + DaisyUI.
  + **La app móvil** debe desarrollarse en React Native + TypeScript + Tailwind + DaisyUI.
  + **El backend** debe construirse con Node.js + Express + Prisma ORM + TypeScript.
  + **La base de datos** debe estar en Supabase (PostgreSQL en la nube).
* **Servicios externos obligatorios:**
  + **Flow** será el medio de pago.
  + **Gmail API** para recuperación de contraseñas,confirmación de correo,etc.
  + **Expo Push Notifications (FCM/APNs)** para notificaciones móviles.
* **Comunicación:** Toda la comunicación entre componentes y servicios externos se debe realizar mediante API REST.
* **Infraestructura:** La solución debe desplegarse en **servicios cloud gestionados**, sin posibilidad de usar servidores propios (on-premise).
* **Otros componentes de software**: no se considera la adquisición y licenciamiento de otros componentes de software.
  1. **Otros antecedentes y consideraciones**

La arquitectura definida se apoya en un conjunto de frameworks y librerías que permiten satisfacer los requerimientos arquitectónicos, asegurando seguridad, mantenibilidad, escalabilidad y rendimiento:

* **Framework Web y Móvil (React/Next.js y React Native/Expo):** Soportan la encapsulación y modularización de componentes, facilitando la mantenibilidad y reutilización de código. Además, optimizan el performance en tiempo de ejecución y permiten ofrecer interfaces modernas e intuitivas.
* **Framework Backend (Node.js + Express + Prisma):** Facilitan la escalabilidad y separación de responsabilidades en la lógica de negocio. Prisma ORM abstrae la capa de datos, mejorando la mantenibilidad y reduciendo la complejidad en el acceso a la base de datos Supabase.
* **Framework de Seguridad (JWT, Gmail API y Supabase RLS):** Aseguran la protección de credenciales e información sensible, soportando la meta de seguridad. El uso de JWT garantiza sesiones seguras, Gmail API permite confirmación de cuentas y recuperación de contraseñas, y Supabase implementa Row Level Security (RLS).
* **Servicios Externos (Flow y Expo Push Notifications):** Permiten cumplir con la meta de interoperabilidad. Flow garantiza la integración de pagos de servicios, mientras que Expo Push Notifications asegura la entrega de notificaciones móviles en tiempo real.

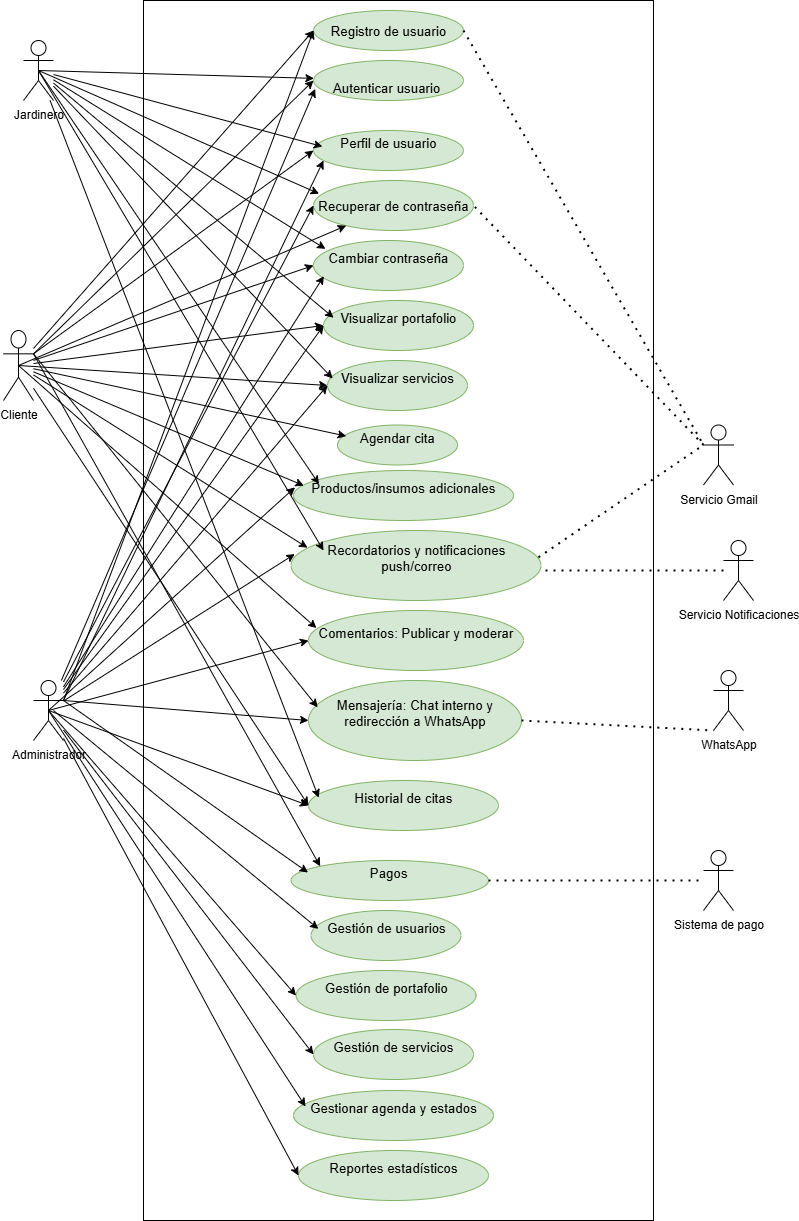
1. **Vista de Escenarios**

Esta sección describe en detalle el conjunto de escenarios funcionales y no funcionales que obtuvieron la mayor prioridad en el análisis. Para esto se presenta y describe el diagrama de casos de uso y los casos de uso prioritarios, así como los escenarios en que uno o más atributos de calidad se ven involucrados de manera significativa.

* 1. **Modelo de Casos de Uso**

Se agrega el modelo de caso uso general del sistema

**Ilustración 1: Diagrama de Caso Uso General del Sistema**



* 1. **Casos de Usos Extendidos**

Los casos de uso considerados son los más relevantes para el desarrollo de la arquitectura. Se adjunta el documento o planilla caso uso.

[**Se adjunta la planilla de casos de uso extendido**](https://docs.google.com/document/d/1bRuV53aFXXolZwpBedPNOj24XdTEjs47/edit?usp=drive_link&ouid=112046847387388169157&rtpof=true&sd=true)

A continuación, se listan los casos de uso relevantes, los cuales pueden ser encontrados con su especificación detallada en el documento “Casos de Uso Extendido”.

| **Código** | **Nombre** | **Actores** | **Prioridad** |
| --- | --- | --- | --- |
| CU01 | Registro de usuario | Cliente/Administrador. | Alto |
| CU02 | Autenticar usuario (Login) | Cliente/Administrador/Jardinero | Alto |
| CU03 | Perfil de usuario | Cliente/ Administrador/ Jardinero | Bajo |
| CU04 | Recuperar contraseña | Cliente/Administrador/Jardinero | Alto |
| CU05 | Cambiar contraseña | Cliente/ Administrador/ Jardinero | Bajo |
| CU06 | Visualizar portafolio | Cliente/Administrador/Jardinero | Alto |
| CU07 | Visualizar servicios (catálogo) | Cliente/Administrador/Jardinero | Alto |
| CU08 | Agendar cita (Cliente) | Cliente | Alto |
| CU09 | Productos/insumos adicionales | Cliente/Administrador/Jardinero | Medio |
| CU10 | Comentarios: Publicar y moderar | Cliente/Administrador | Medio |
| CU11 | Mensajería: Chat interno y redirección a WhatsApp | Cliente/ Administrador | Alto |
| CU12 | Historial de citas | Cliente/Administrador/Jardinero | Alto |
| CU13 | Gestión de usuarios (Admin) | Administrador | Alto |
| CU14 | Gestión de portafolio (Admin) | Administrador | Alto |
| CU15 | Gestión de servicios (Admin) | Administrador | Alto |
| CU16 | Gestionar agenda y estados (Admin) | Administrador | Alto |
| CU17 | Reportes estadísticos (exportable PDF/XLS) | Administrador | Alto |
| CU18 | Pagos | Cliente/ administrador | Alto |
| CU19 | Recordatorios y notificaciones push/correo | Sistema/Cliente/Administrador/jardinero | Alto |

* 1. **Especificación de los Escenarios de Calidad Relevantes**

Después de un análisis en conjunto con los stakeholders, los escenarios de calidad se expresan a continuación:

Escenario 1 – Inicio de sesión

| Identificador: EQ01 | | |
| --- | --- | --- |
| Escenario(s): | | Un usuario ingresa sus credenciales para acceder al sistema. |
| Atributos de Calidad relevantes: | | Seguridad, Rendimiento, Usabilidad. |
| Componentes del Escenario | Estímulos: | El usuario envía email y contraseña. |
| Fuente del estímulo | Cliente desde web/móvil. |
| Ambiente: | Carga normal del sistema. |
| Artefacto: | Módulo de autenticación (API login + BD). |
| Respuesta: | El sistema valida credenciales y devuelve acceso o error. |
| Medida de Respuesta | Tiempo de validación ≤ 2s; intentos fallidos limitados a 3. |

Escenario 2 – Registro de usuario

| Identificador: EQ01 | | |
| --- | --- | --- |
| Escenario(s): | | Un cliente crea una nueva cuenta en la aplicación. |
| Atributos de Calidad relevantes: | | Seguridad, Disponibilidad, Usabilidad. |
| Componentes del Escenario | Estímulos: | El cliente completa el formulario de registro. |
| Fuente del estímulo | Cliente desde frontend web/móvil. |
| Ambiente: | Carga normal, con validaciones en tiempo real. |
| Artefacto: | API de registro, base de datos. |
| Respuesta: | El sistema crea el usuario, envía correo de confirmación y redirige al login. |
| Medida de Respuesta | Confirmación enviada en ≤ 10s; 99% de registros procesados sin error. |

Escenario 3 – Recuperación de contraseña

| Identificador: EQ03 | | |
| --- | --- | --- |
| Escenario(s): | | Un usuario solicita recuperar su contraseña. |
| Atributos de Calidad relevantes: | | Seguridad, Confiabilidad. |
| Componentes del Escenario | Estímulos: | El usuario envía su correo en el formulario “Olvidé mi contraseña”. |
| Fuente del estímulo | Cliente desde web/móvil. |
| Ambiente: | Usuario fuera de sesión. |
| Artefacto: | Backend de forgot-password, servicio de email. |
| Respuesta: | El sistema genera un token único y envía un enlace seguro al correo. |
| Medida de Respuesta | Enlace válido 15 min; envío en ≤ 30s. |

Escenario 4 – Visualización de Portafolio

| Identificador: EQ04 | | |
| --- | --- | --- |
| Escenario(s): | | El cliente consulta trabajos realizados en el portafolio digital. |
| Atributos de Calidad relevantes: | | Usabilidad, Rendimiento, Disponibilidad. |
| Componentes del Escenario | Estímulos: | Cliente selecciona opción “Portafolio” desde menú. |
| Fuente del estímulo | Usuario web o móvil. |
| Ambiente: | Carga de datos multimedia (imágenes). |
| Artefacto: | API de portafolio, base de datos, almacenamiento de imágenes. |
| Respuesta: | El sistema devuelve la lista de trabajos con imágenes asociadas. |
| Medida de Respuesta | Tiempo de carga ≤ 3s con conexión estable; disponibilidad del servicio ≥ 99%. |

Escenario 5 – Pago de Servicios con Flow

| Identificador: EQ05 | | |
| --- | --- | --- |
| Escenario(s): | | El cliente paga un servicio mediante Flow. |
| Atributos de Calidad relevantes: | | Seguridad, Confiabilidad, Disponibilidad. |
| Componentes del Escenario | Estímulos: | Cliente confirma la compra de un servicio. |
| Fuente del estímulo | Usuario web/móvil. |
| Ambiente: | Proceso de transacción con pasarela externa (Flow). |
| Artefacto: | API de backend, integración con Flow. |
| Respuesta: | El sistema envía datos a Flow, recibe confirmación y actualiza el estado de la orden. |
| Medida de Respuesta | Transacción confirmada en ≤ 15s; 0% pérdida de integridad de datos. |

Escenario 6 – Notificaciones Push (Expo Push Notifications)

| Identificador: EQ06 | | |
| --- | --- | --- |
| Escenario(s): | | El sistema envía notificación a un jardinero sobre una nueva cita asignada. |
| Atributos de Calidad relevantes: | | Disponibilidad, Rendimiento. |
| Componentes del Escenario | Estímulos: | Se crea una nueva cita en el sistema. |
| Fuente del estímulo | Backend. |
| Ambiente: | Alta concurrencia (muchas notificaciones en paralelo). |
| Artefacto: | API de notificaciones + servicio de Expo Push. |
| Respuesta: | El jardinero recibe la notificación en su dispositivo móvil. |
| Medida de Respuesta | Notificación entregada en ≤ 5s; tasa de entrega ≥ 9%. |

Escenario 7 – Disponibilidad General del Sistema

| Identificador: EQ07 | | |
| --- | --- | --- |
| Escenario(s): | | El sistema debe estar disponible para clientes y jardineros durante la jornada laboral. |
| Atributos de Calidad relevantes: | | Disponibilidad, Confiabilidad. |
| Componentes del Escenario | Estímulos: | Cliente y jardinero intentan acceder al sistema en horario laboral. |
| Fuente del estímulo | Navegador web o aplicación móvil. |
| Ambiente: | Carga normal del sistema. |
| Artefacto: | Infraestructura en Supabase + backend. |
| Respuesta: | El sistema responde de forma continua sin caída del servicio. |
| Medida de Respuesta | Disponibilidad ≥ 99.5% mensual. |

Escenario 8 – Seguridad de Datos Personales

| Identificador: EQ08 | | |
| --- | --- | --- |
| Escenario(s): | | El sistema protege datos sensibles de los usuarios (ej. contraseñas, direcciones). |
| Atributos de Calidad relevantes: | | Seguridad, Privacidad. |
| Componentes del Escenario | Estímulos: | Un usuario registra/consulta datos sensibles. |
| Fuente del estímulo | Cliente web/móvil. |
| Ambiente: | Comunicación entre frontend y backend. |
| Artefacto: | API de backend + base de datos Supabase. |
| Respuesta: | Datos en tránsito cifrados (HTTPS), contraseñas almacenadas con hash seguro. |
| Medida de Respuesta | 100% de comunicaciones en HTTPS. |

Escenario 9 – Actualización de Perfil de Usuario

| Identificador: EQ09 | | |
| --- | --- | --- |
| Escenario(s): | | El usuario edita sus datos personales en el perfil. |
| Atributos de Calidad relevantes: | | Usabilidad, Seguridad, Confiabilidad. |
| Componentes del Escenario | Estímulos: | Usuario modifica nombre, dirección o teléfono. |
| Fuente del estímulo | Cliente web/móvil. |
| Ambiente: | Sesión activa y validada. |
| Artefacto: | API de usuario, BD Supabase. |
| Respuesta: | Los cambios se guardan y reflejan inmediatamente. |
| Medida de Respuesta | Confirmación ≤ 2s; sin pérdida de datos. |

Escenario 10 – Gestión de Citas/Servicios

| Identificador: EQ10 | | |
| --- | --- | --- |
| Escenario(s): | | El cliente agenda una cita de servicio de jardinería. |
| Atributos de Calidad relevantes: | | Confiabilidad, Disponibilidad. |
| Componentes del Escenario | Estímulos: | El cliente selecciona día, hora y servicio. |
| Fuente del estímulo | Cliente web/móvil. |
| Ambiente: | Varias solicitudes simultáneas. |
| Artefacto: | Módulo de reservas, API reservas,BD Supabase. |
| Respuesta: | El sistema registra la cita y notifica al jardinero. |
| Medida de Respuesta | Reservas confirmadas en ≤ 5s; 0% de duplicidad de reserva. |

Escenario 11 – Recuperación ante Fallo

| Identificador: EQ11 | | |
| --- | --- | --- |
| Escenario(s): | | El sistema debe recuperarse automáticamente en caso de caída del backend. |
| Atributos de Calidad relevantes: | | Disponibilidad, Confiabilidad. |
| Componentes del Escenario | Estímulos: | Interrupción en el backend. |
| Fuente del estímulo | Error en servidor / red. |
| Ambiente: | Producción. |
| Artefacto: | Backend en Express/Node.js, hosting en Vercel/Supabase. |
| Respuesta: | El sistema reanuda el servicio con mínima interrupción. |
| Medida de Respuesta | Recuperación rápida; pérdida de datos = 0. |

Escenario 12 – Escalabilidad bajo alta demanda

| Identificador: EQ12 | | |
| --- | --- | --- |
| Escenario(s): | | Múltiples usuarios acceden al sistema en horario pico. |
| Atributos de Calidad relevantes: | | Rendimiento, Escalabilidad. |
| Componentes del Escenario | Estímulos: | 100 usuarios simultáneos intentan iniciar sesión y agendar citas. |
| Fuente del estímulo | Clientes concurrentes. |
| Ambiente: | Producción en carga alta. |
| Artefacto: | Backend, BD Supabase. |
| Respuesta: | El sistema sigue respondiendo con tiempos aceptables. |
| Medida de Respuesta | Tiempo de respuesta ≤ 5s para 95% de peticiones. |

Escenario 13 – Confirmación de Reserva vía Correo

| Identificador: EQ13 | | |
| --- | --- | --- |
| Escenario(s): | | El cliente recibe un correo de confirmación al agendar un servicio. |
| Atributos de Calidad relevantes: | | Confiabilidad, Usabilidad. |
| Componentes del Escenario | Estímulos: | Cliente agenda una cita. |
| Fuente del estímulo | Cliente web/móvil. |
| Ambiente: | Conectividad normal. |
| Artefacto: | Backend + servicio de correo. |
| Respuesta: | El sistema envía un correo con los detalles de la reserva. |
| Medida de Respuesta | Envío en ≤ 1 min; entregabilidad ≥ 99%. |

Escenario 14 – Disponibilidad del Pago en Línea

| Identificador: EQ14 | | |
| --- | --- | --- |
| Escenario(s): | | El sistema debe asegurar la disponibilidad del módulo de pago con Flow. |
| Atributos de Calidad relevantes: | | Disponibilidad, Seguridad. |
| Componentes del Escenario | Estímulos: | Cliente inicia un pago. |
| Fuente del estímulo | Cliente web/móvil. |
| Ambiente: | Jornada laboral. |
| Artefacto: | API backend + integración Flow. |
| Respuesta: | El sistema procesa la transacción sin rechazos por caída del servicio. |
| Medida de Respuesta | Disponibilidad ≥ 99%; 0% pérdida de transacciones. |

Escenario 15 – Protección de Información Financiera

| Identificador: EQ15 | | |
| --- | --- | --- |
| Escenario(s): | | Los datos sensibles de pago no deben quedar almacenados en la BD. |
| Atributos de Calidad relevantes: | | Seguridad, Privacidad. |
| Componentes del Escenario | Estímulos: | Cliente paga un servicio con tarjeta. |
| Fuente del estímulo | Cliente web/móvil. |
| Ambiente: | Comunicación con Flow. |
| Artefacto: | Backend, integración Flow. |
| Respuesta: | El sistema envía datos a Flow y nunca guarda información financiera. |
| Medida de Respuesta | 100% de tokens temporales; 0% almacenamiento de datos sensibles. |

Escenario 16 – Cierre de Sesión Automático

| Identificador: EQ16 | | |
| --- | --- | --- |
| Escenario(s): | | El sistema debe cerrar sesión automáticamente tras inactividad prolongada. |
| Atributos de Calidad relevantes: | | Seguridad. |
| Componentes del Escenario | Estímulos: | Usuario permanece inactivo 15 min. |
| Fuente del estímulo | Cliente web/móvil. |
| Ambiente: | Sesión abierta. |
| Artefacto: | Backend de autenticación (JWT). |
| Respuesta: | El sistema invalida el token y cierra la sesión. |
| Medida de Respuesta | Tiempo máximo de inactividad = 15 min. |

Escenario 17 – Acceso a Catálogo de Servicios

| Identificador: EQ17 | | |
| --- | --- | --- |
| Escenario(s): | | El cliente visualiza la lista de servicios disponibles. |
| Atributos de Calidad relevantes: | | Usabilidad, Rendimiento. |
| Componentes del Escenario | Estímulos: | Cliente abre la sección “Catálogo”. |
| Fuente del estímulo | Cliente web/móvil. |
| Ambiente: | Conexión normal. |
| Artefacto: | API catálogo, BD Supabase. |
| Respuesta: | El sistema muestra la lista de servicios con precios e imágenes. |
| Medida de Respuesta | Tiempo de carga ≤ 2s; disponibilidad ≥ 99%. |

Escenario 18 – Gestión de Usuarios por el Administrador

| Identificador: EQ18 | | |
| --- | --- | --- |
| Escenario(s): | | El administrador crea, edita o desactiva cuentas de usuarios (clientes/jardineros). |
| Atributos de Calidad relevantes: | | Seguridad, Confiabilidad, Usabilidad. |
| Componentes del Escenario | Estímulos: | El administrador actualiza un perfil desde el panel de control. |
| Fuente del estímulo | Administrador autenticado (web). |
| Ambiente: | Sesión activa con permisos de rol. |
| Artefacto: | Backend + BD Supabase. |
| Respuesta: | Los cambios se aplican y quedan reflejados en el sistema. |
| Medida de Respuesta | Confirmación en ≤ 2s. |

Escenario 19 – Gestión de Servicios y Tarifas

| Identificador: EQ19 | | |
| --- | --- | --- |
| Escenario(s): | | El administrador modifica el catálogo de servicios y sus precios. |
| Atributos de Calidad relevantes: | | Confiabilidad, Usabilidad. |
| Componentes del Escenario | Estímulos: | El administrador agrega o edita un servicio. |
| Fuente del estímulo | Administrador web. |
| Ambiente: | Horario laboral, sin interrumpir la disponibilidad del catálogo. |
| Artefacto: | API de servicios, BD Supabase. |
| Respuesta: | El catálogo se actualiza de inmediato para todos los usuarios. |
| Medida de Respuesta | Reflejo de cambios en ≤ 1 min; 0% de pérdida de datos. |

Escenario 20 – Panel de Reportes y Estadísticas

| Identificador: EQ20 | | |
| --- | --- | --- |
| Escenario(s): | | El administrador consulta reportes de reservas, ingresos,etc. |
| Atributos de Calidad relevantes: | | Rendimiento, Usabilidad. |
| Componentes del Escenario | Estímulos: | El administrador genera un reporte mensual. |
| Fuente del estímulo | Administrador web. |
| Ambiente: | Base de datos con gran volumen de información. |
| Artefacto: | Módulo de reportes, BD Supabase. |
| Respuesta: | El sistema procesa y muestra el reporte en pantalla/exporta a Excel/PDF. |
| Medida de Respuesta | Generación ≤ 10s para 100 registros. |

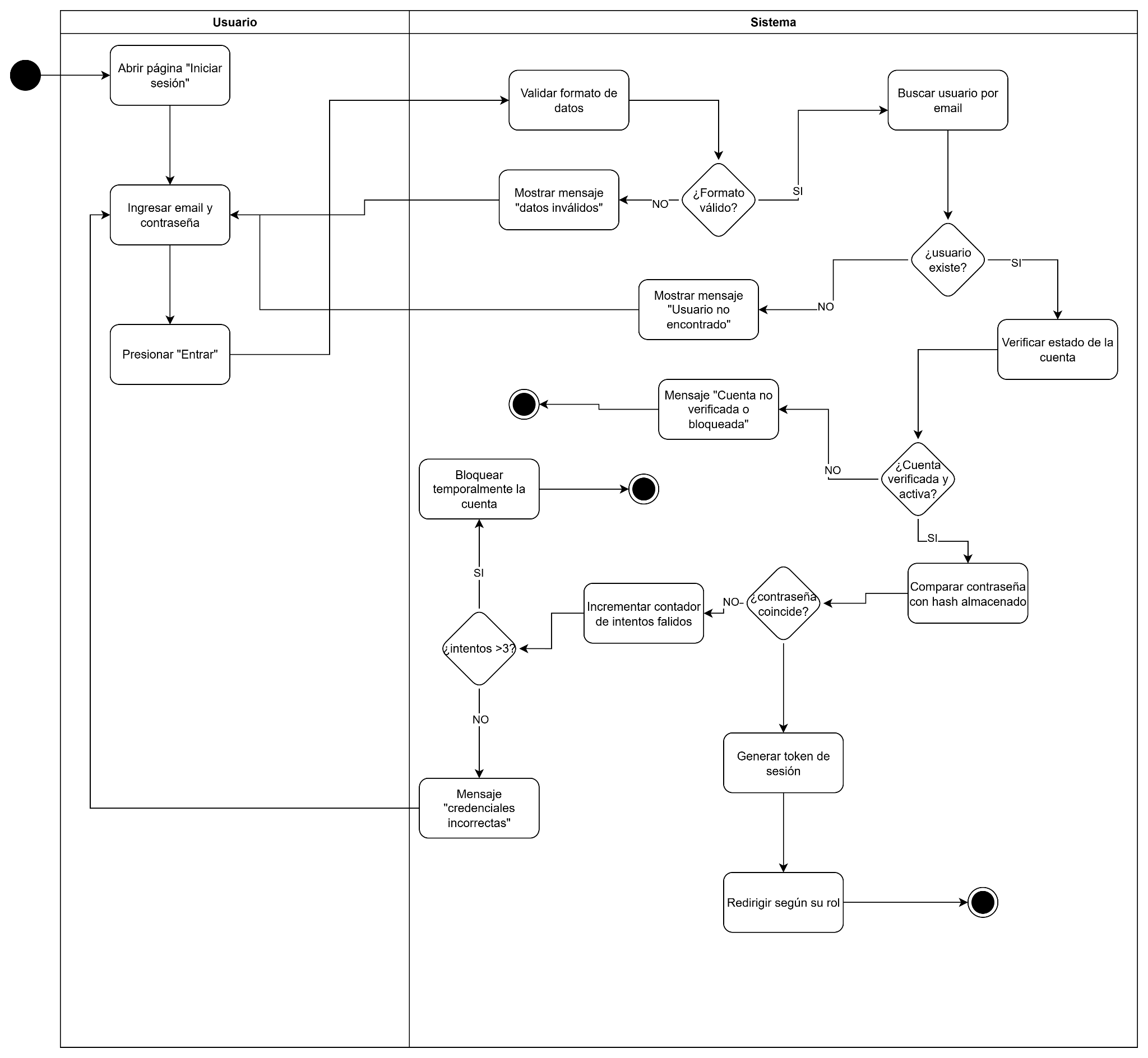
1. **Vista de Procesos**

**Ilustración 2: Diagramas de Actividades**

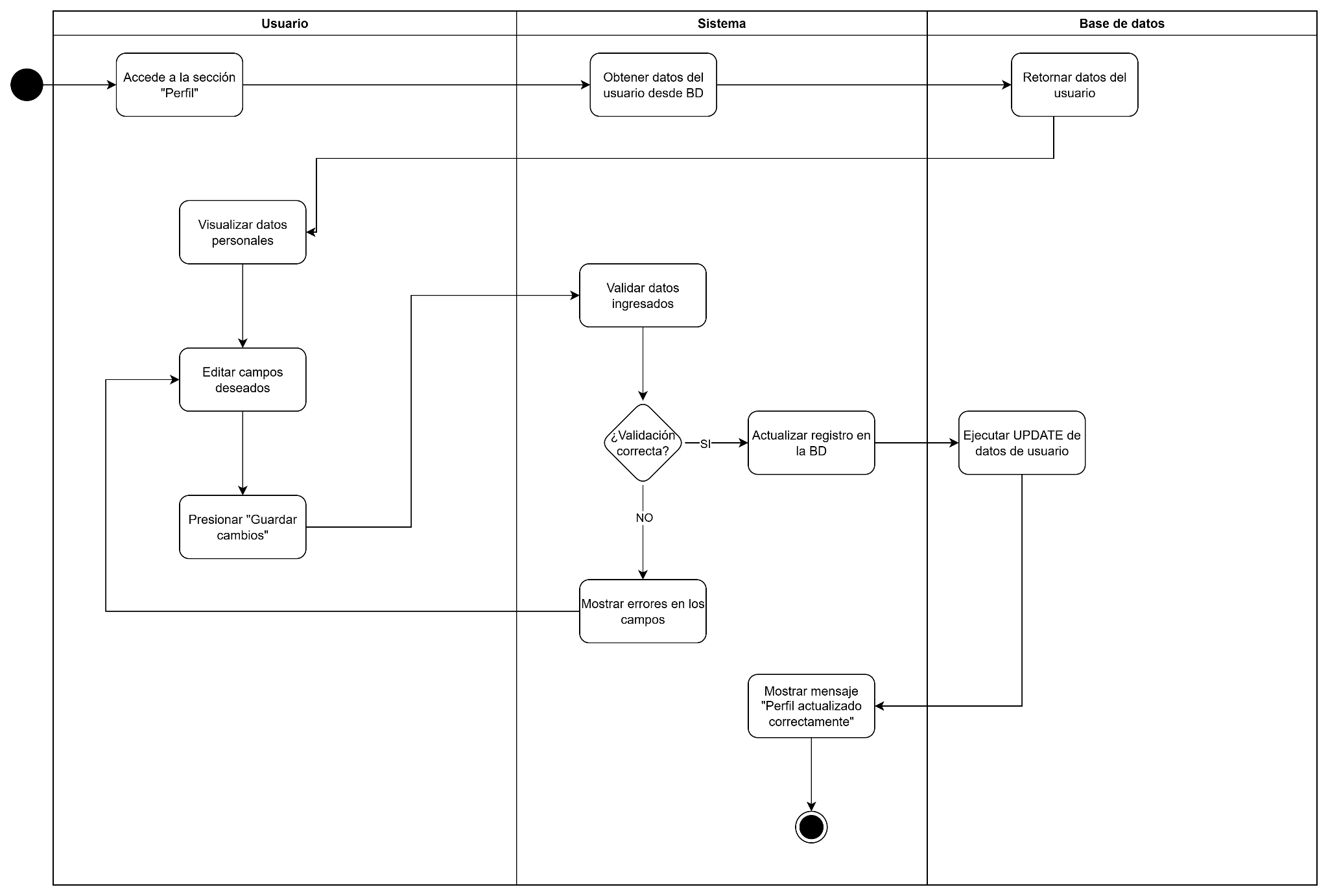
CU01 – Registro de usuario



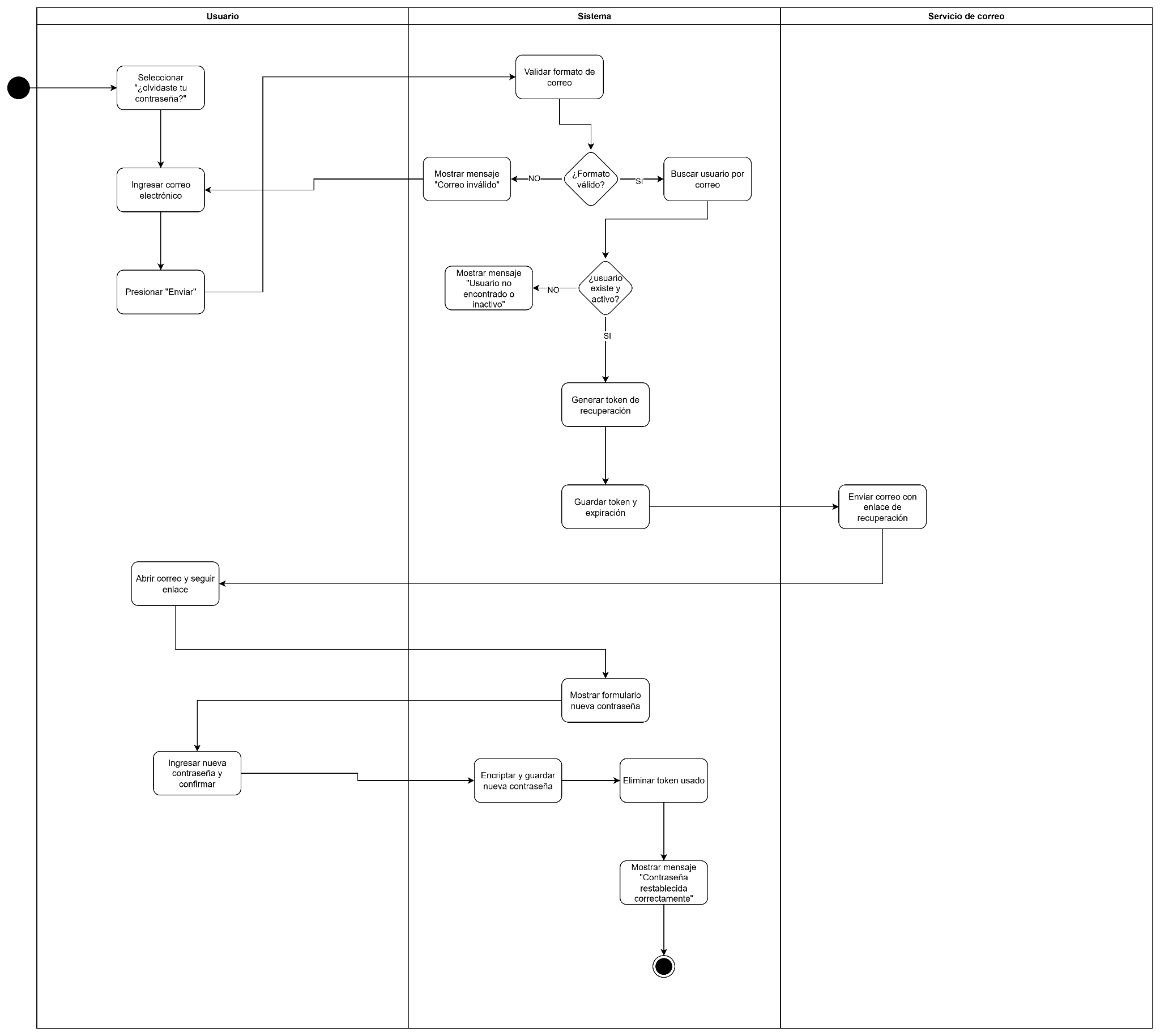
CU02 – Autenticar usuario (Login)



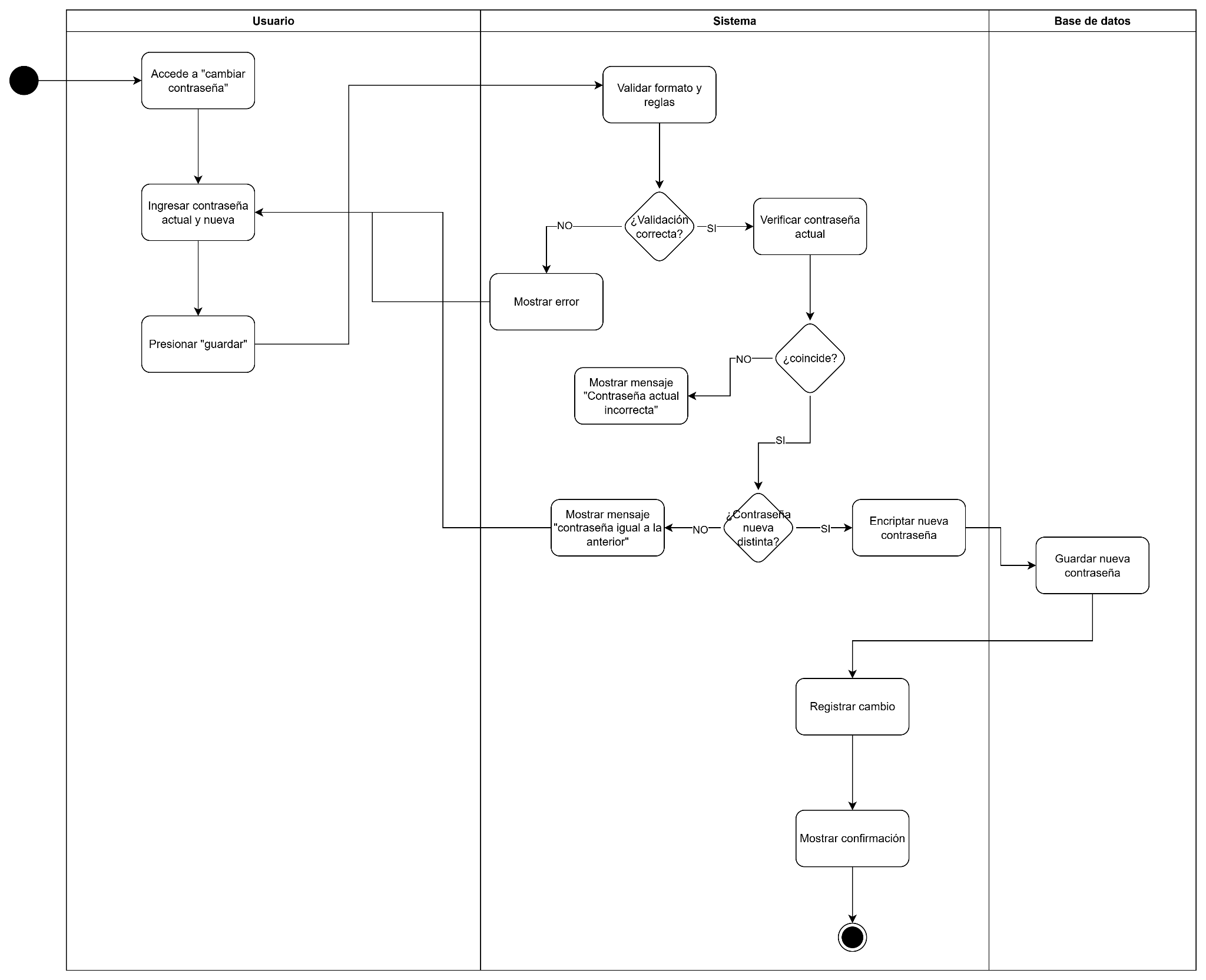
CU03 – Perfil de usuario



CU04 – Recuperar contraseña



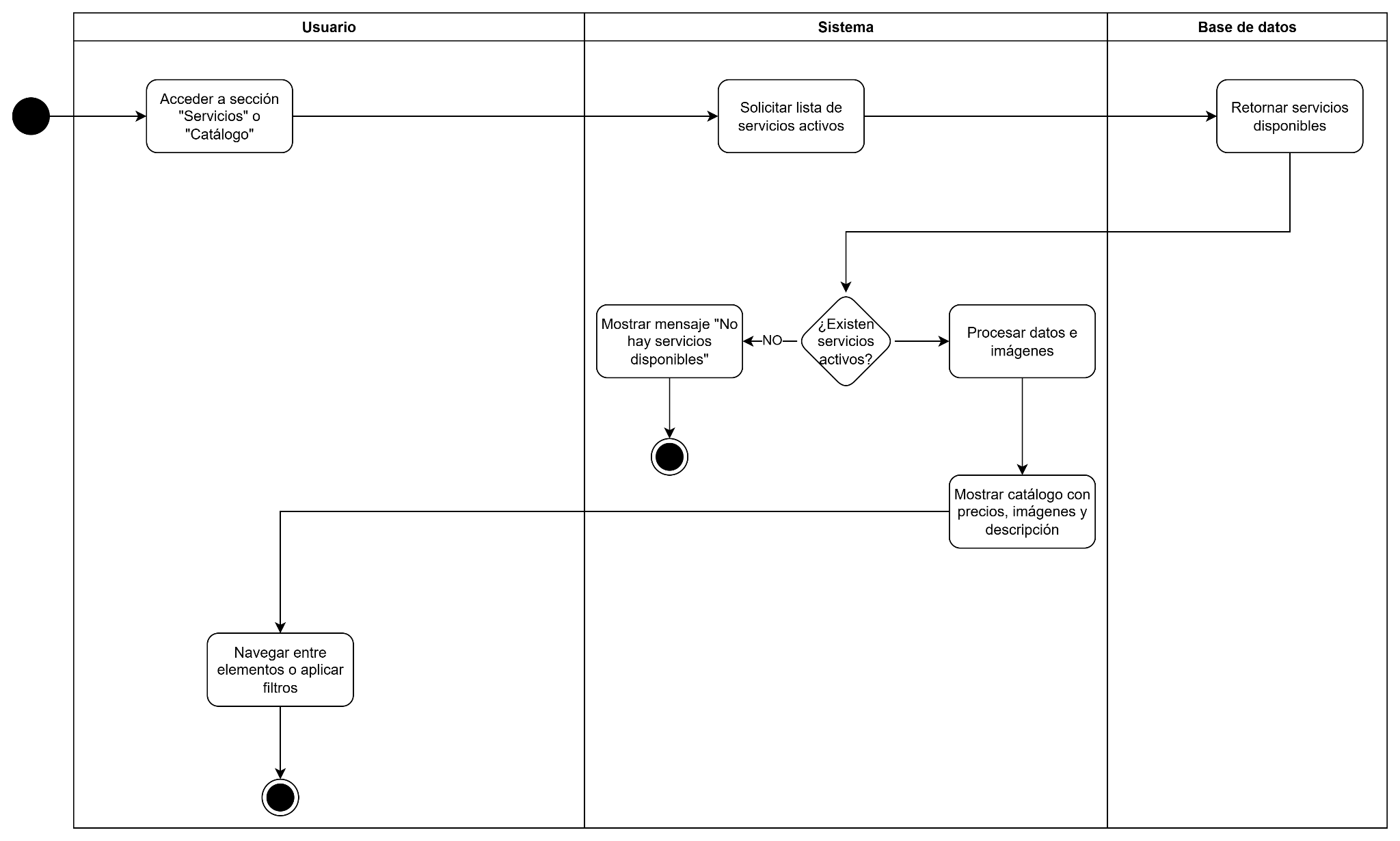
CU05 – Cambiar contraseña



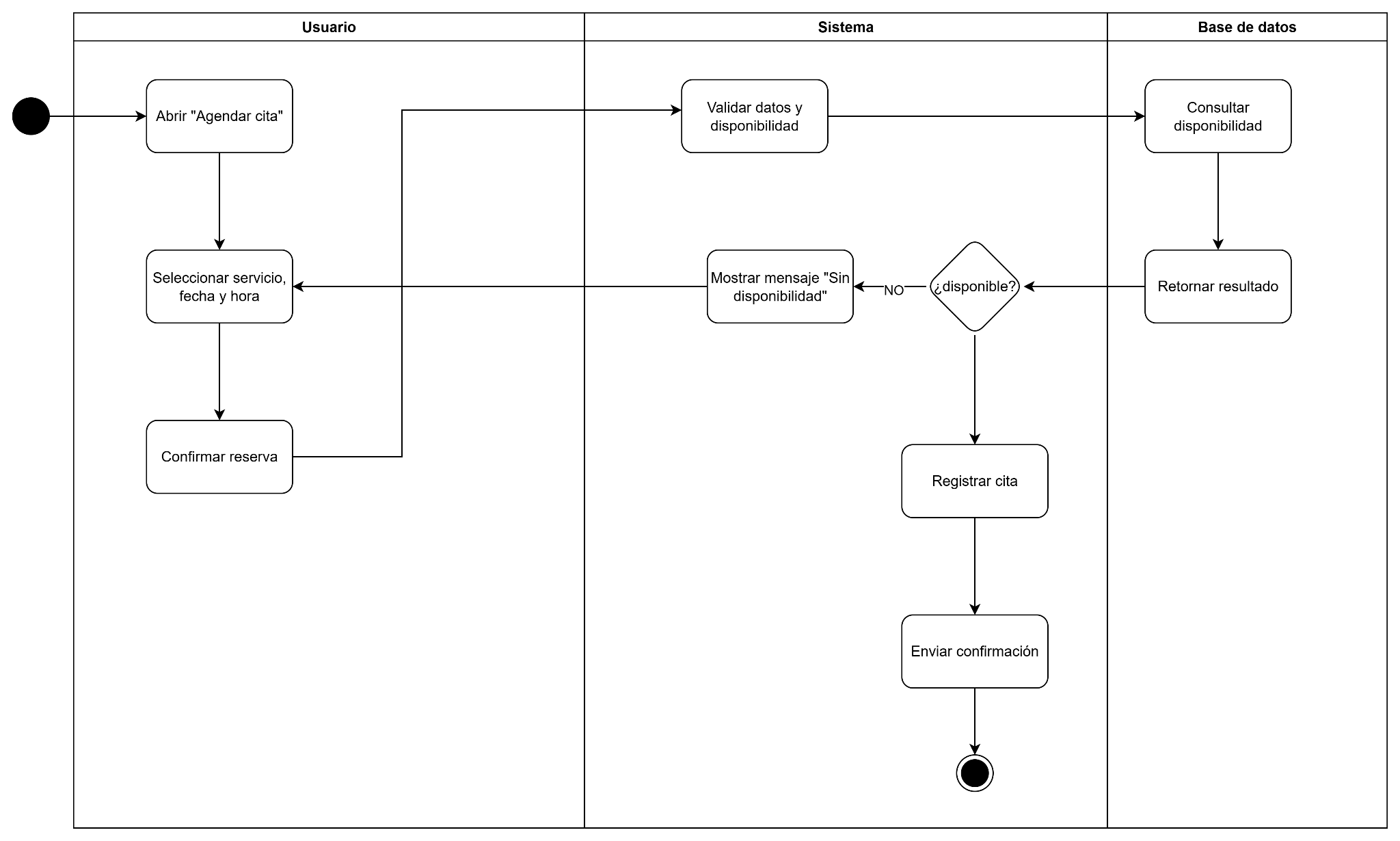
CU06 – Visualizar portafolio



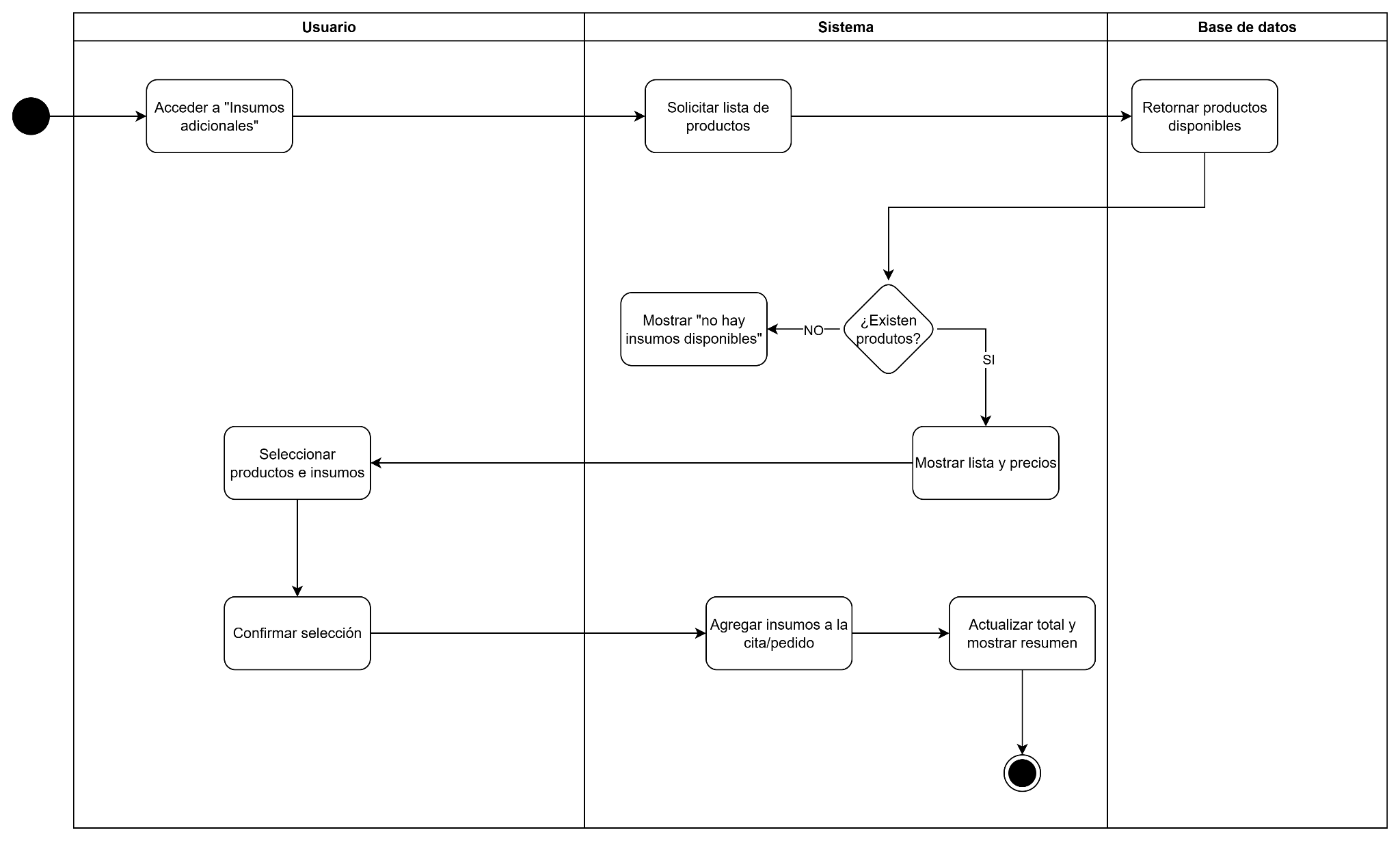
CU07 – Visualizar servicios (catálogo)



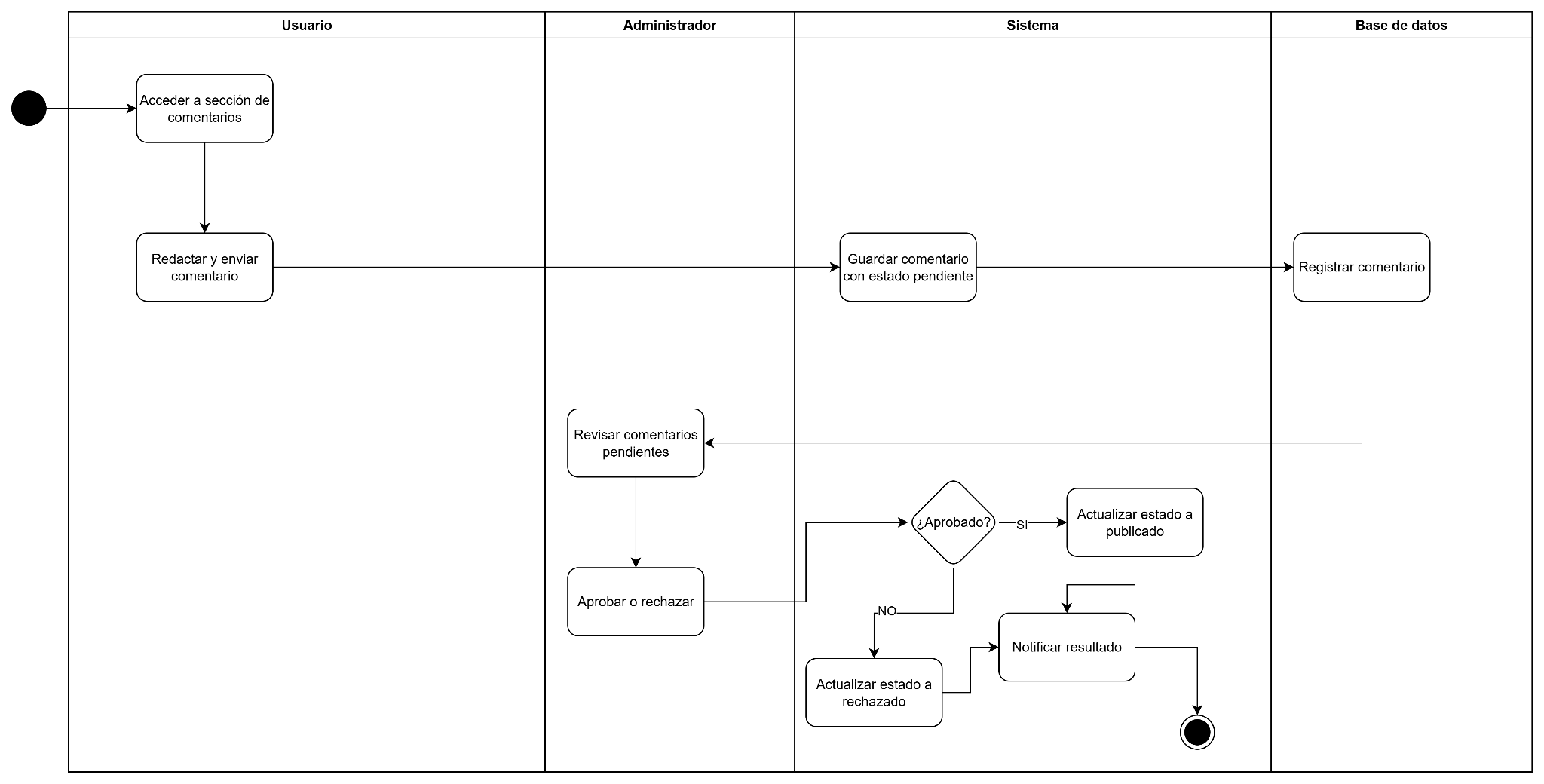
CU08 – Agendar cita (Cliente)



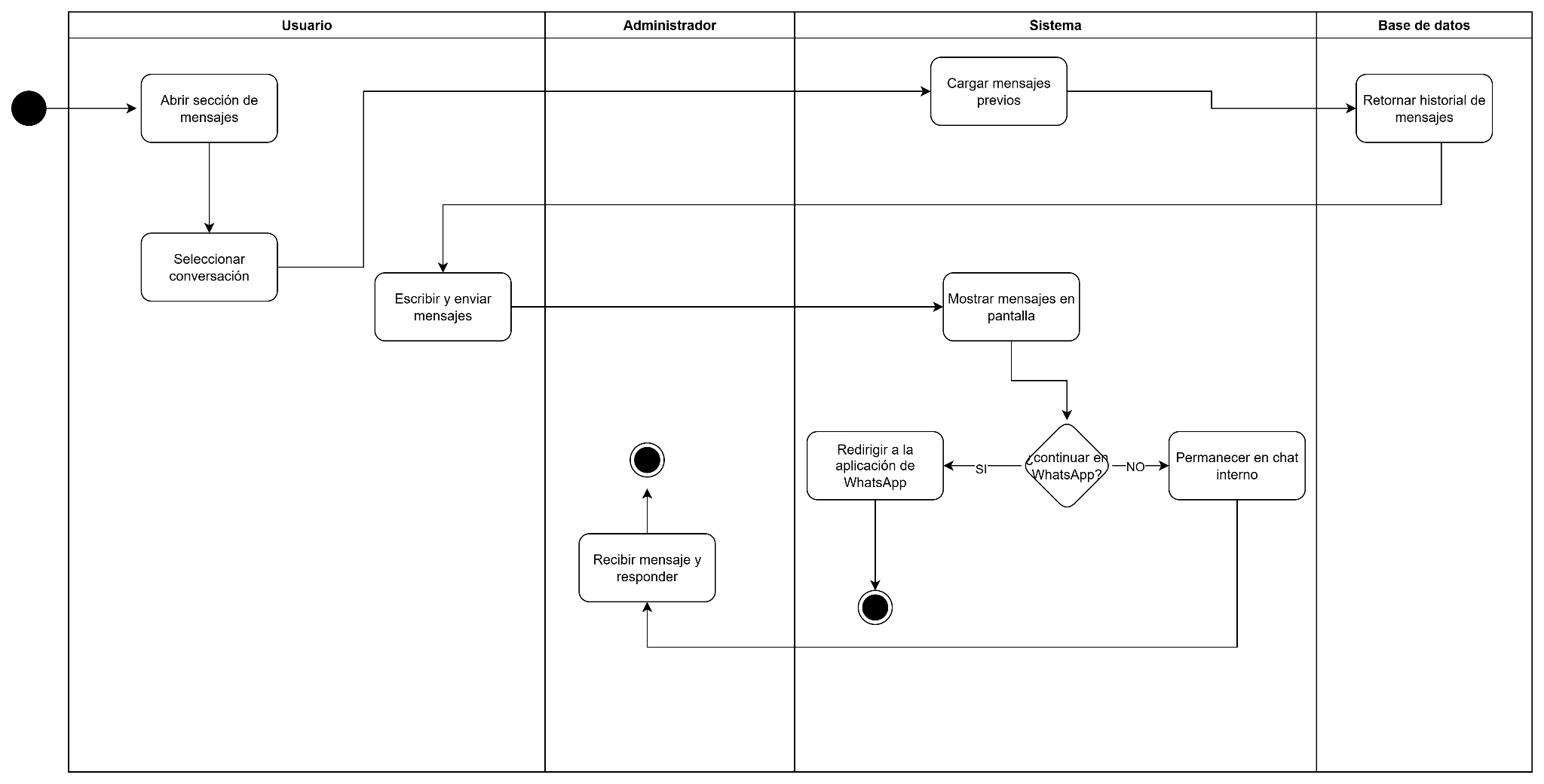
CU09 – Productos/Insumos adicionales



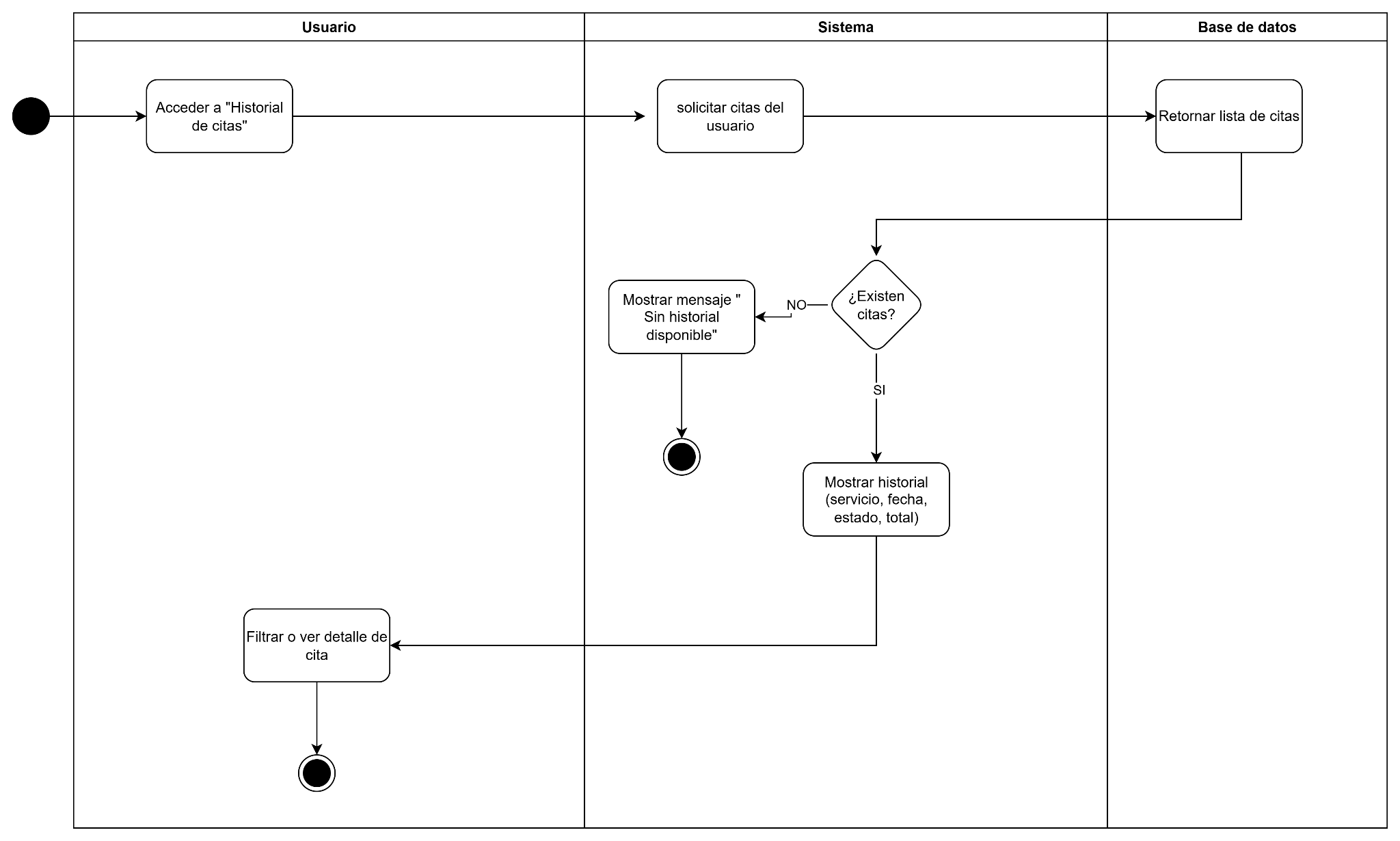
CU10 – Comentarios: Publicar y moderar



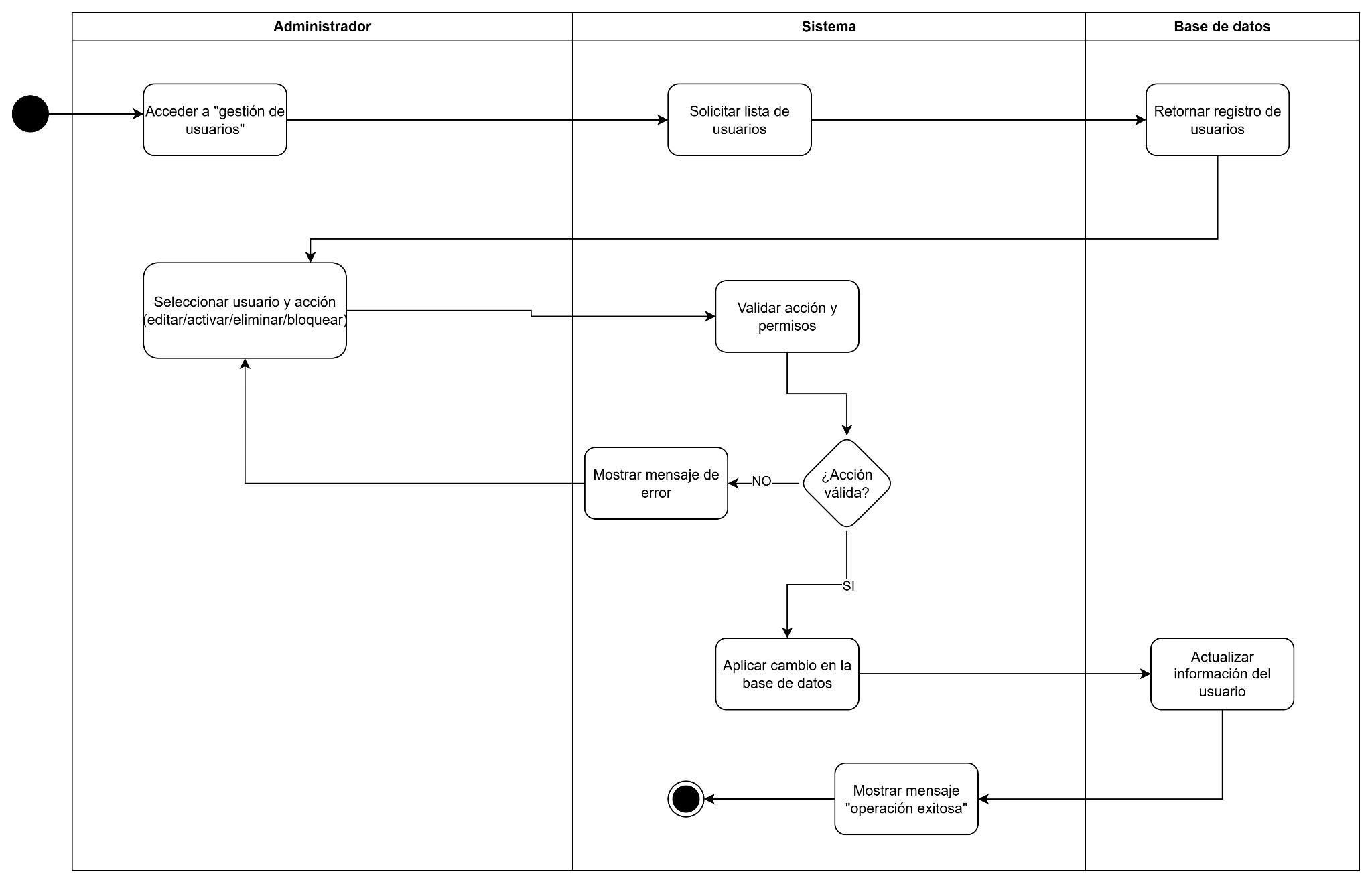
CU11 – Mensajería: Chat interno y redirección a WhatsApp



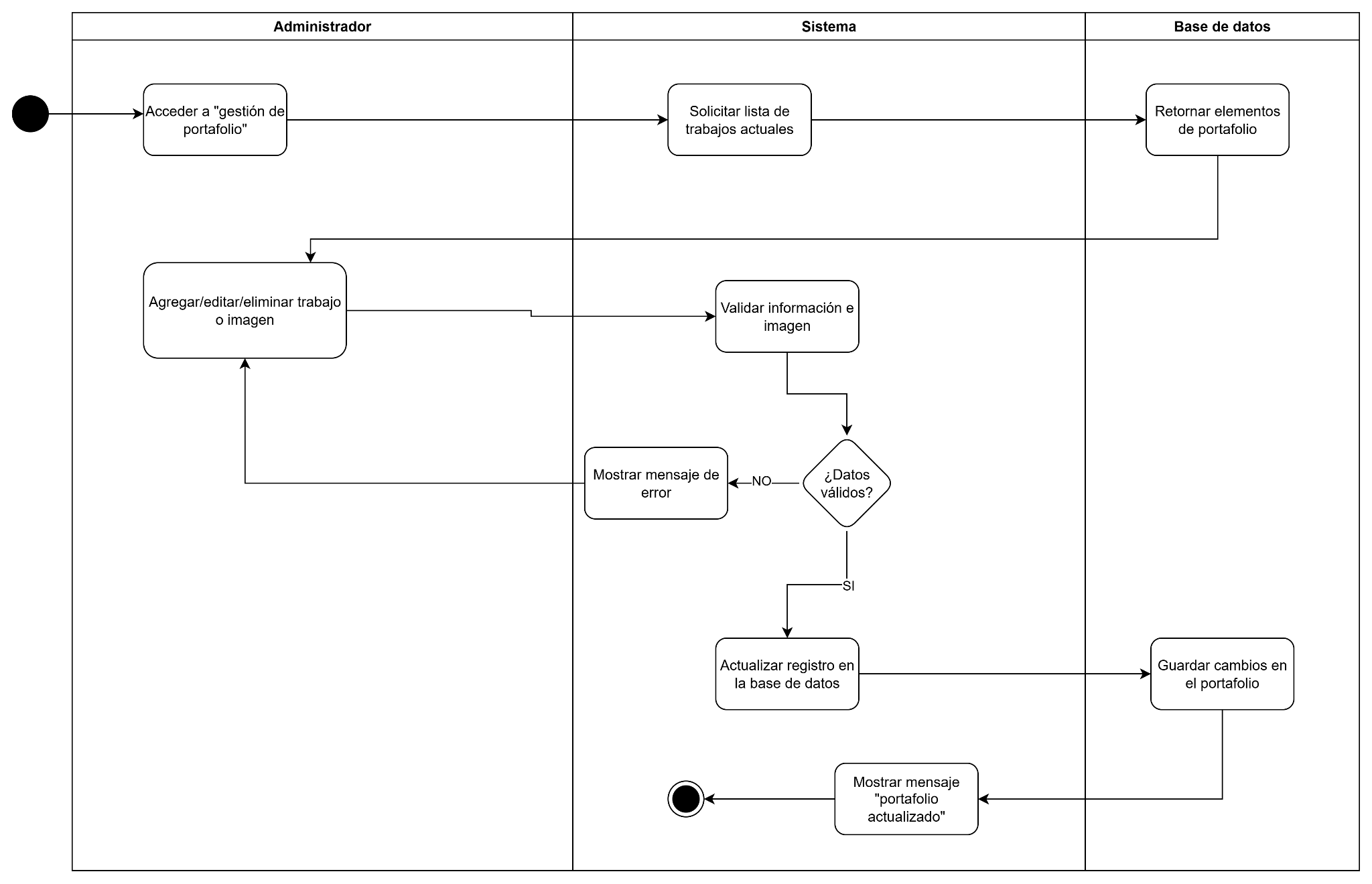
CU12 – Historial de citas



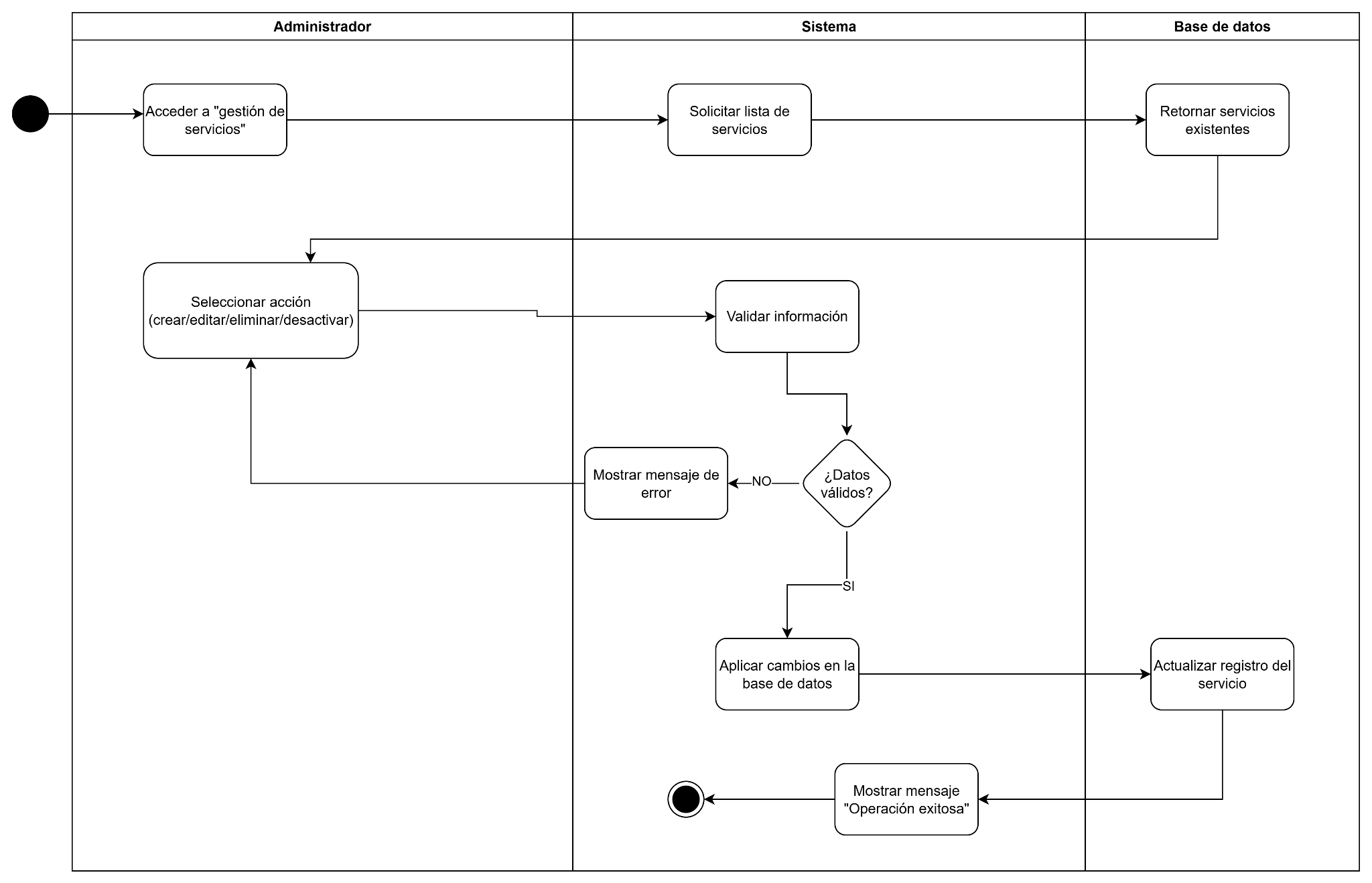
CU13 – Gestión de usuarios (Admin)



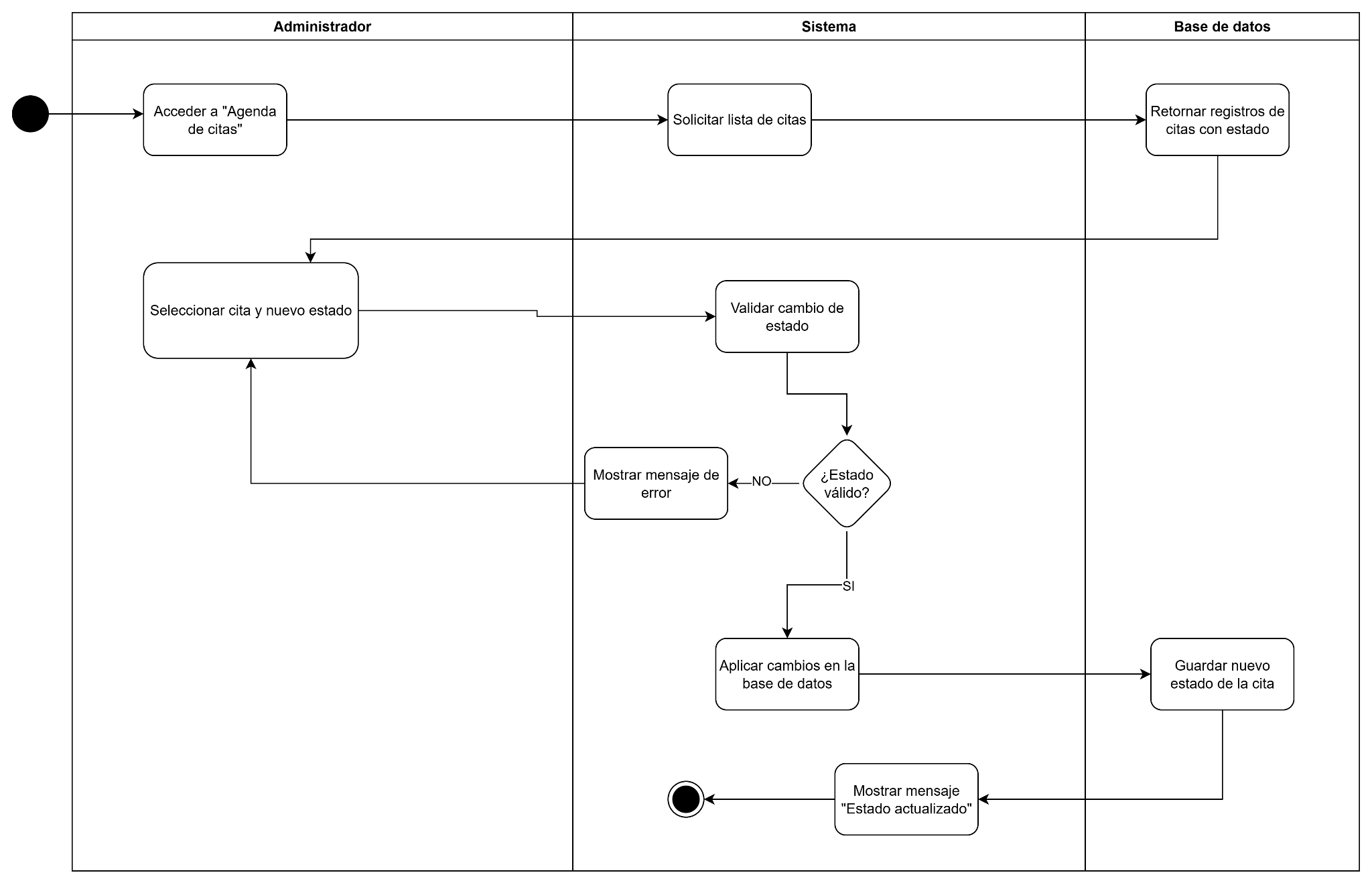
CU14 – Gestión de portafolio (Admin)



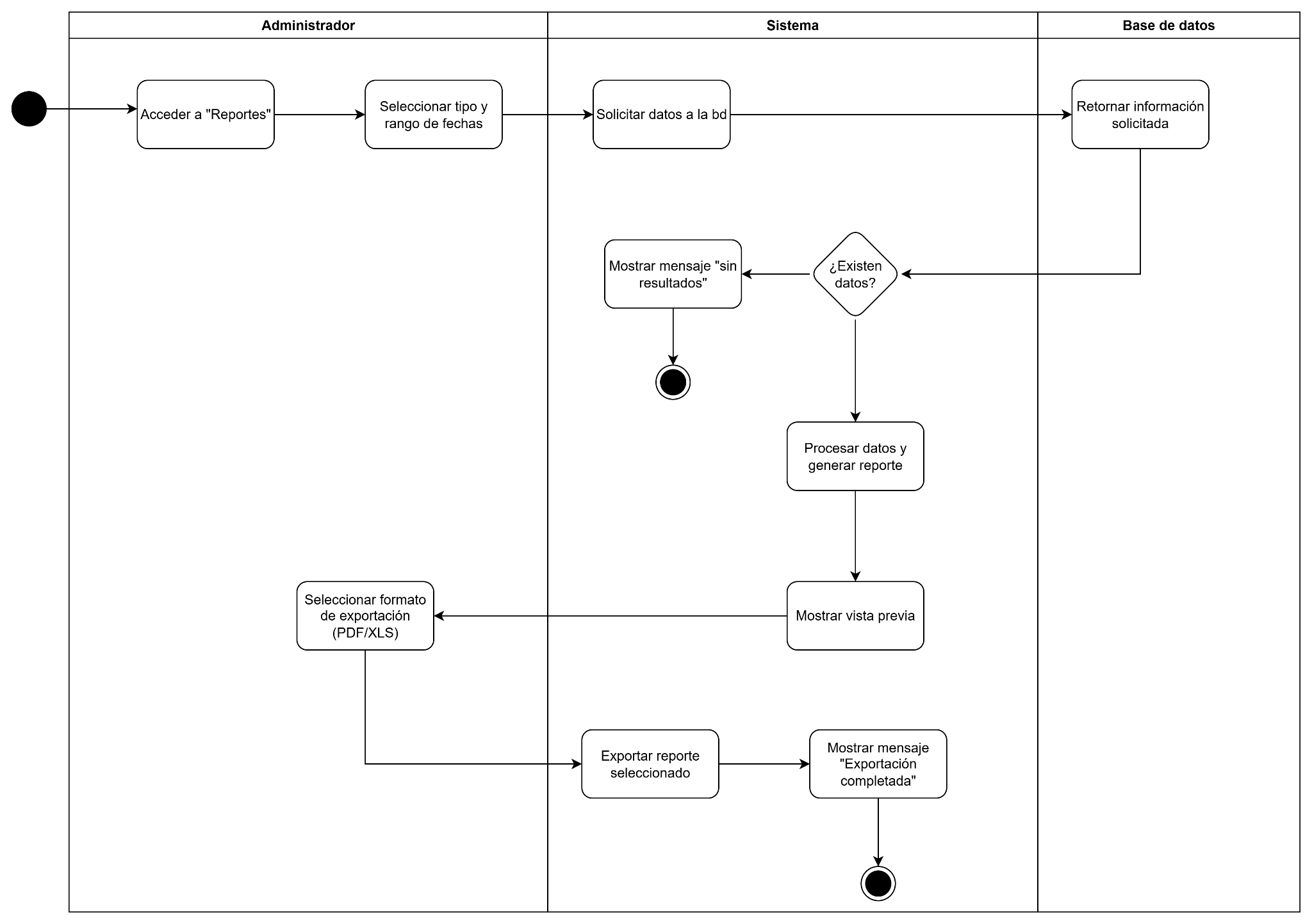
CU15 – Gestión de servicios (Admin)



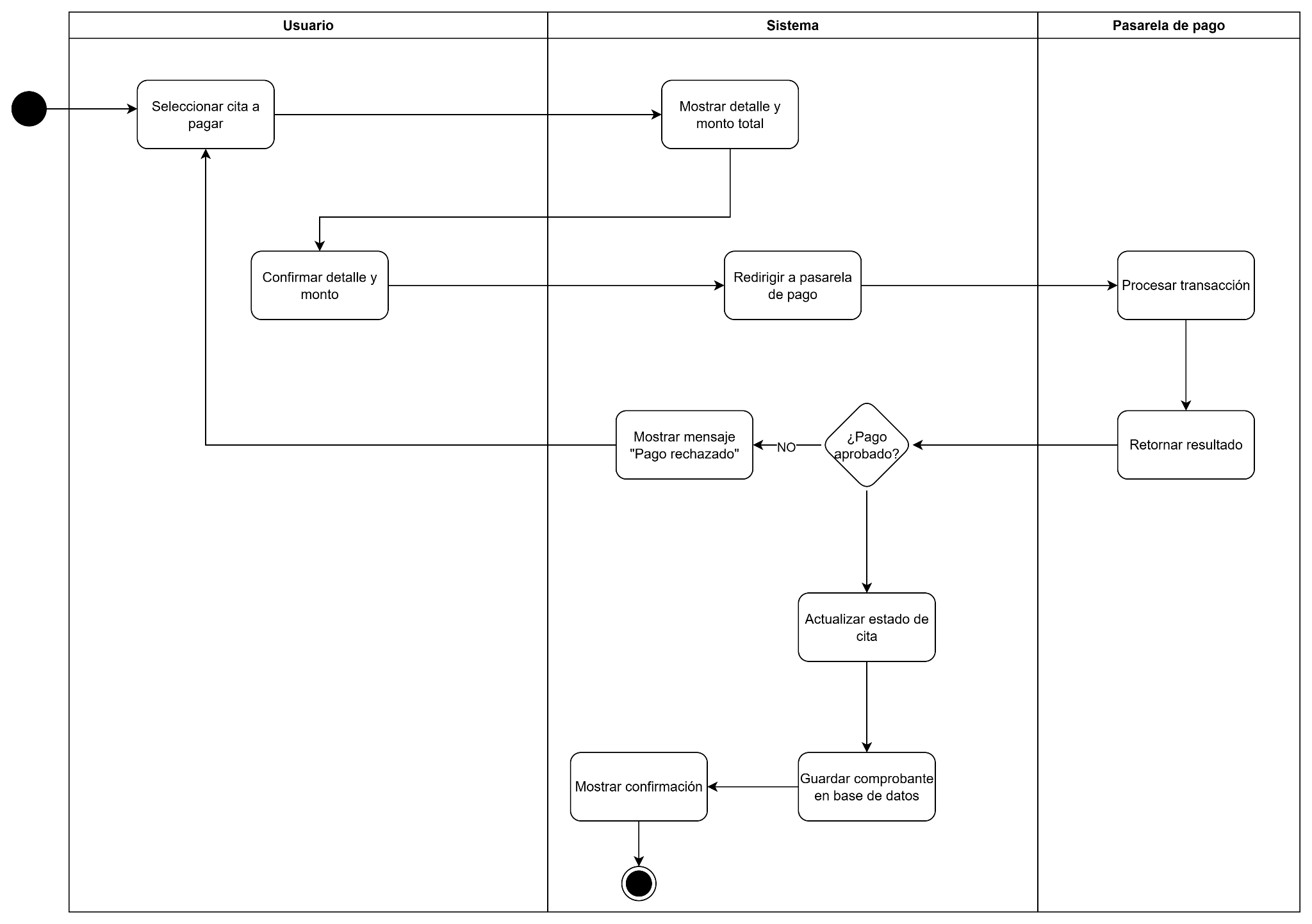
CU16 – Gestionar agenda y estados (Admin)



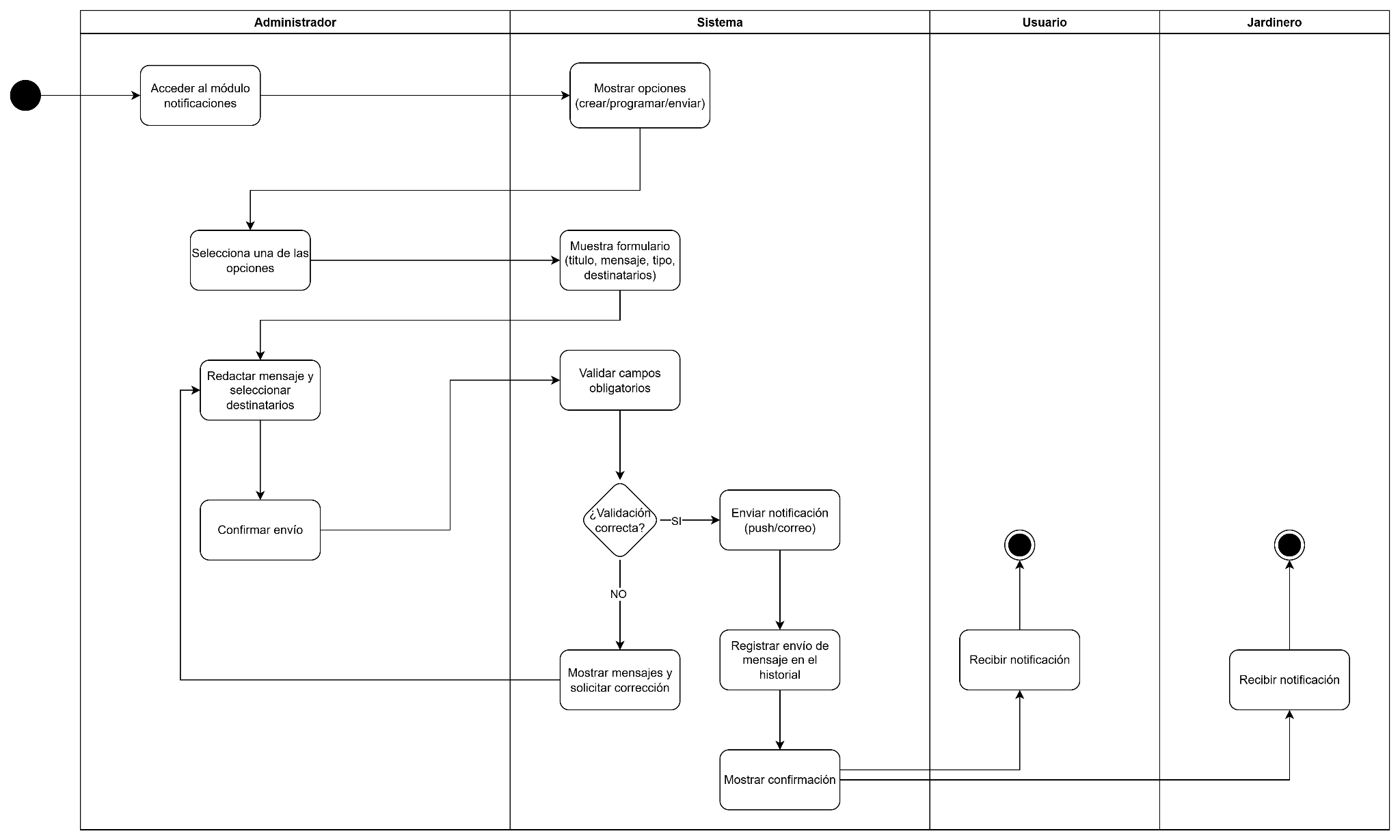
CU17 – Reportes estadísticos (exportable PDF/XLS)



CU18 – Pagos



CU19 – Recordatorios y notificaciones push/correo

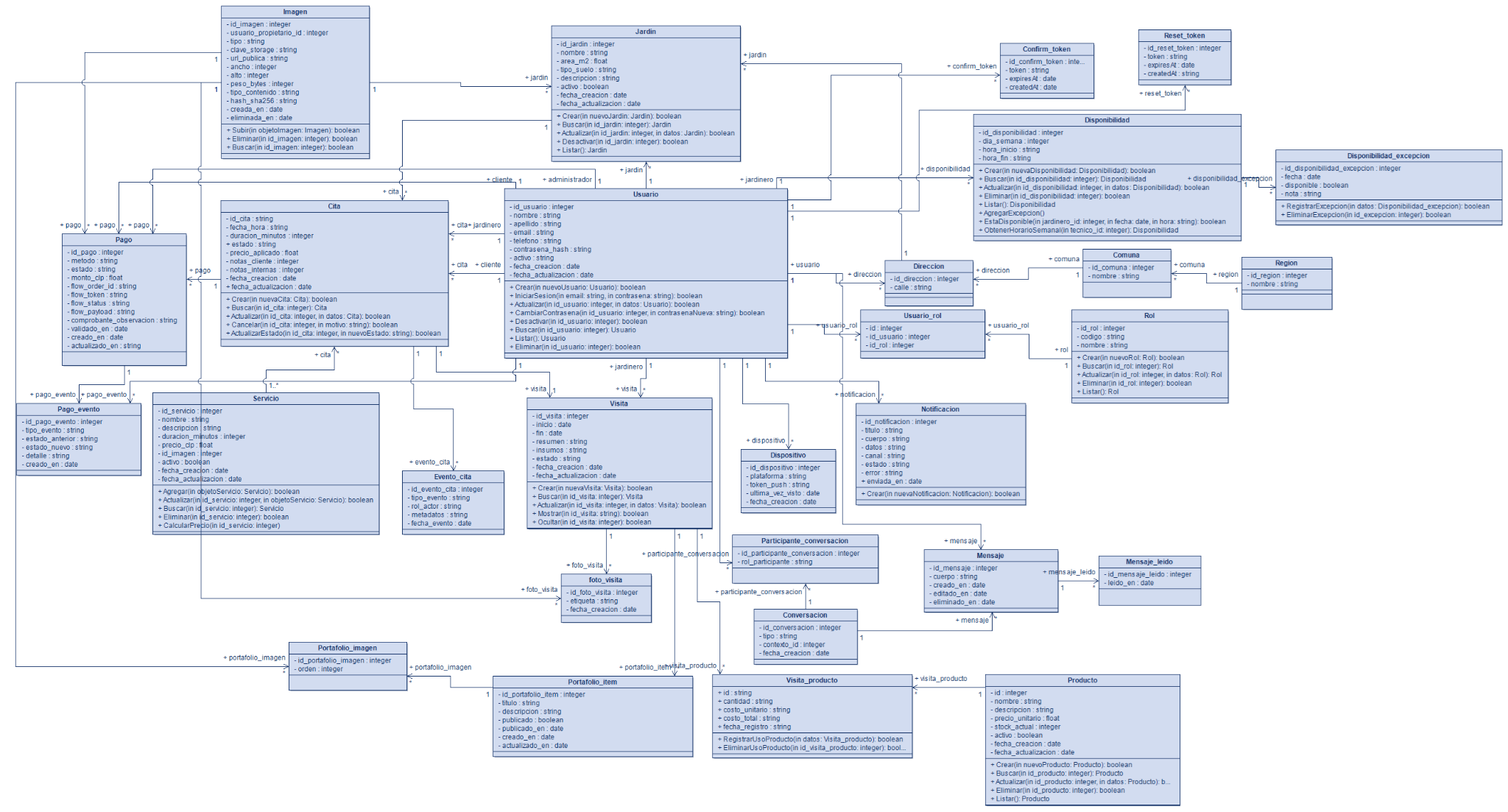


1. **Vista Lógica**

A continuación, se presenta una vista lógica de la aplicación expresado en tres diagramas, uno de ellos que muestra la parte estructural o estática de la aplicación (clases) y a la base de datos (modelo relacional).otra vista que representa la parte dinámica (secuencias).

* 1. **Parte Estructural ( Diagrama de Clases y Diagrama Relacional)**

**Ilustración 3: Diagrama de Clases**

****

### Descripción de Clases

| **Código** | **Nombre** | **Descripción** |
| --- | --- | --- |
| CL-001 | Usuario | Representa a cualquier persona que interactúa con el sistema (cliente, jardinero o administrador).  Contiene datos personales, credenciales y estado de la cuenta. |
| CL-002 | Rol | Define los permisos o responsabilidades que un usuario puede tener dentro del sistema (ej. Administrador, Jardinero, Cliente). |
| CL-003 | Usuario\_rol | Tabla intermedia que relaciona un usuario con uno o varios roles.  Permite la asignación flexible de permisos según perfil. |
| CL-004 | Jardin | Contiene la información del espacio físico donde se realizan los servicios (nombre, área, tipo de suelo, dirección).  Pertenece a un cliente (Usuario). |
| CL-005 | Servicio | Define los distintos tipos de servicios ofrecidos por la empresa (limpieza, poda, mantención, etc.), con su descripción, duración y precio. |
| CL-006 | Cita | Representa una reserva o agendamiento de servicio entre un cliente y un jardinero, incluyendo fecha, hora, duración, notas y estado (pendiente, realizada, cancelada, etc.). |
| CL-007 | Visita | Registra la ejecución efectiva de una cita por parte de un jardinero.  Contiene inicio y fin, resumen y estado de la visita. |
| CL-008 | Evento\_cita | Guarda los eventos o cambios ocurridos en una cita (ej. creación, cancelación, modificación de estado).  Permite trazar el historial completo de la atención. |
| CL-009 | Pago | Registra los pagos asociados a una cita, incluyendo método, monto, moneda y estado (pendiente, validado, rechazado). |
| CL-010 | Pago\_evento | Guarda los cambios o acciones sobre un pago (por ejemplo, “pago confirmado”, “validado por administrador”).  Ayuda a mantener un historial de los flujos financieros. |
| CL-011 | Producto | Representa los materiales o productos utilizados por los jardineros durante la realización de un servicio (ej. fertilizantes, bolsas de basura, repuestos). |
| CL-012 | Visita\_producto | Relaciona una visita con los productos que fueron utilizados en ella, registrando cantidad, costo unitario y total. |
| CL-013 | Disponibilidad | Registra los horarios semanales en los que un jardinero puede trabajar (día, hora de inicio y fin). |
| CL-014 | Disponibilidad\_excepcion | Permite registrar excepciones a la disponibilidad habitual (por ejemplo, vacaciones o ausencias). |
| CL-015 | Direccion | Almacena la dirección física asociada a un usuario o jardín (calle y comuna). |
| CL-016 | Comuna | Representa la comuna a la que pertenece una dirección, vinculada a una región. |
| CL-017 | Region | Define las regiones geográficas del país, utilizadas para clasificar comunas y direcciones. |
| CL-018 | Notificacion | Registra mensajes automáticos o manuales enviados a los usuarios (por correo, push o interno).  Contiene título, cuerpo, canal y estado de envío. |
| CL-019 | Conversacion | Representa un chat entre usuarios dentro del sistema (por ejemplo, cliente ↔ jardinero). |
| CL-020 | Participante\_conversacion | Asocia usuarios a una conversación determinada, con su rol (remitente o destinatario). |
| CL-021 | Mensaje | Almacena los mensajes enviados dentro de una conversación (texto, imagen, hora, remitente). |
| CL-022 | Mensaje\_leido | Registra qué usuario leyó cada mensaje y cuándo, permitiendo el seguimiento del estado de lectura. |
| CL-023 | Imagen | Almacena información de imágenes subidas al sistema (url, tipo, tamaño, hash).  Puede ser usada por servicios, jardines, mensajes o portafolio. |
| CL-024 | Foto\_visita | Relaciona imágenes con una visita, como evidencia del trabajo realizado. |
| CL-025 | Portafolio\_item | Representa un trabajo o proyecto completado que puede mostrarse públicamente en el portafolio del sistema. |
| CL-026 | Portafolio\_imagen | Relaciona imágenes con los ítems del portafolio, manteniendo un orden visual. |
| CL-027 | Confirm\_token | Token de verificación usado para confirmar la creación de cuenta o dirección de correo electrónico. |
| CL-028 | Reset\_token | Token temporal usado para restablecer la contraseña de un usuario. |

**Ilustración 4: Diagrama de Base Datos (Relacional)**



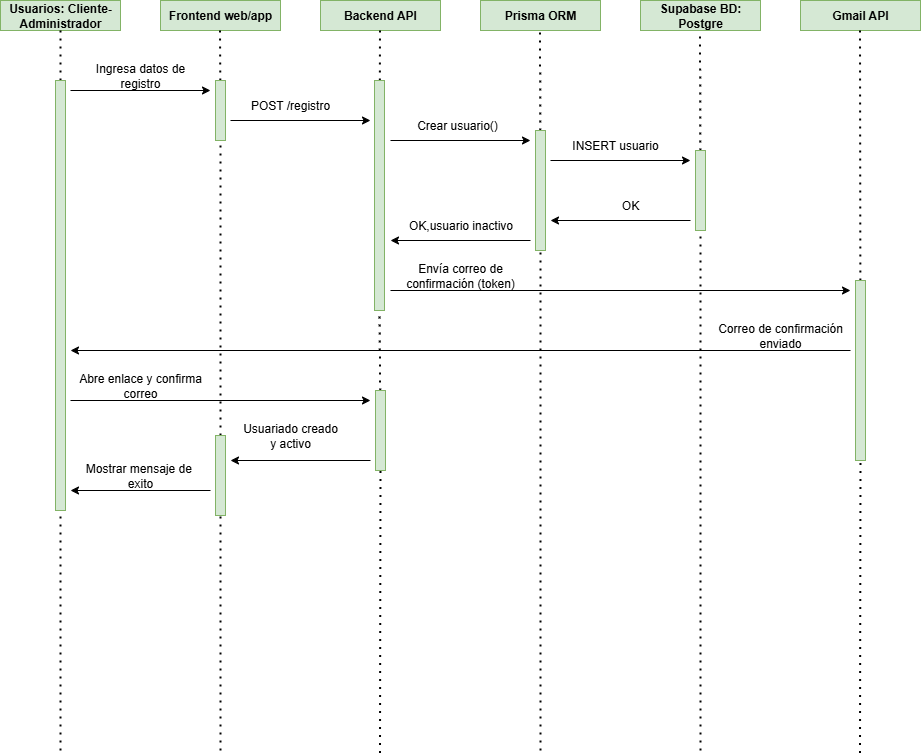
### Descripción de Tablas

| **Código** | **Nombre** | **Descripción** |
| --- | --- | --- |
| TB-001 | usuario | Contiene los datos principales de los usuarios del sistema, tanto clientes como técnicos o administradores. Almacena información de contacto, credenciales y estado de la cuenta. |
| TB-002 | rol | Define los distintos roles disponibles en el sistema (por ejemplo: administrador, técnico, cliente). Se utiliza para controlar los permisos y accesos. |
| TB-003 | usuario\_rol | Tabla intermedia que relaciona a los usuarios con sus roles asignados, permitiendo una relación de tipo muchos a muchos entre usuario y rol. |
| TB-004 | perfil\_cliente | Complementa la información de un usuario que actúa como cliente. Incluye datos adicionales como RUT y notas asociadas. |
| TB-005 | region | Registra las regiones geográficas del país, utilizadas para la ubicación de comunas y direcciones. |
| TB-006 | comuna | Lista las comunas existentes y las vincula con una región. Se utiliza como parte de la información geográfica de direcciones. |
| TB-007 | direccion | Contiene la información de dirección física de los usuarios o jardines, incluyendo calle y comuna. Se asocia opcionalmente a un usuario. |
| TB-008 | imagen | Almacena los metadatos de las imágenes subidas al sistema (ancho, alto, peso, tipo de contenido, hash, URL pública). Se usa para ilustrar perfiles, servicios, jardines y otros elementos. |
| TB-009 | jardin | Representa un jardín asociado a un cliente. Contiene datos como nombre, área, tipo de suelo, descripción, imagen principal y dirección. Es una de las entidades principales del sistema. |
| TB-010 | servicio | Registra los tipos de servicios ofrecidos (por ejemplo, limpieza, poda, riego). Incluye su descripción, duración, precio y una imagen asociada. |
| TB-011 | cita | Gestiona las citas o reservas de servicios entre clientes y técnicos. Almacena fecha, duración, estado, precio, notas y datos de cancelación. |
| TB-012 | visita | Registra la ejecución real de una cita, incluyendo el técnico asignado, horarios de inicio y fin, resumen de la atención, estado y detalles de los insumos utilizados. |
| TB-013 | visita\_producto | Relaciona las visitas con los productos utilizados en ellas, indicando cantidad, costos unitarios y totales. |
| TB-014 | producto | Lista los productos o materiales que se utilizan durante los servicios (por ejemplo, fertilizantes o herramientas). Contiene precio, stock y estado de disponibilidad. |
| TB-015 | disponibilidad | Define los rangos horarios en que un técnico está disponible para realizar servicios, segmentados por día de la semana. |
| TB-016 | disponibilidad\_excepcion | Registra excepciones a la disponibilidad normal de los técnicos, indicando fechas específicas en que estarán o no disponibles. |
| TB-017 | pago | Contiene los pagos asociados a las citas. Incluye información del método, estado, monto, moneda, comprobantes e integración con plataformas externas como Flow. |
| TB-018 | pago\_evento | Historial de eventos relacionados a un pago (por ejemplo, creación, validación, error). Permite auditar los cambios de estado y el usuario que los realizó. |
| TB-019 | evento\_cita | Registra eventos importantes del ciclo de vida de una cita (por ejemplo: creación, reasignación, cancelación), junto con el usuario actor y metadatos adicionales. |
| TB-020 | confirm\_token | Gestiona los tokens de confirmación utilizados en procesos como la verificación de cuentas. Contiene el token, usuario asociado y fecha de expiración. |
| TB-021 | reset\_token | Almacena los tokens generados para recuperación de contraseñas, con su usuario asociado y fecha de caducidad. |
| TB-022 | dispositivo | Registra los dispositivos móviles vinculados a los usuarios para el envío de notificaciones push. Incluye plataforma, token y última conexión. |
| TB-023 | notificacion | Contiene el historial de notificaciones enviadas a los usuarios (por ejemplo, alertas, recordatorios o avisos del sistema). Guarda título, cuerpo, canal, estado y datos adicionales. |
| TB-024 | conversacion | Representa un canal de comunicación entre usuarios (por ejemplo, cliente y técnico). Cada conversación puede tener múltiples mensajes asociados. |
| TB-025 | mensaje | Registra los mensajes enviados dentro de una conversación. Puede incluir texto o una imagen adjunta, junto con las marcas de edición o eliminación. |
| TB-026 | mensaje\_leido | Lleva el registro de qué usuarios han leído cada mensaje y la fecha en que lo hicieron. |
| TB-027 | participante\_conversacion | Asocia usuarios a una conversación, definiendo su rol dentro de ella (emisor, receptor, sistema). |
| TB-028 | foto\_visita | Asocia imágenes a una visita, permitiendo documentar el trabajo realizado o el estado del jardín con etiquetas descriptivas. |
| TB-029 | portafolio\_item | Contiene los elementos del portafolio de trabajos realizados (por ejemplo, proyectos destacados). Incluye título, descripción, imagen principal y fecha de publicación. |
| TB-030 | portafolio\_imagen | Relaciona múltiples imágenes con un ítem del portafolio, permitiendo mostrar galerías de fotografías asociadas a un mismo trabajo. |

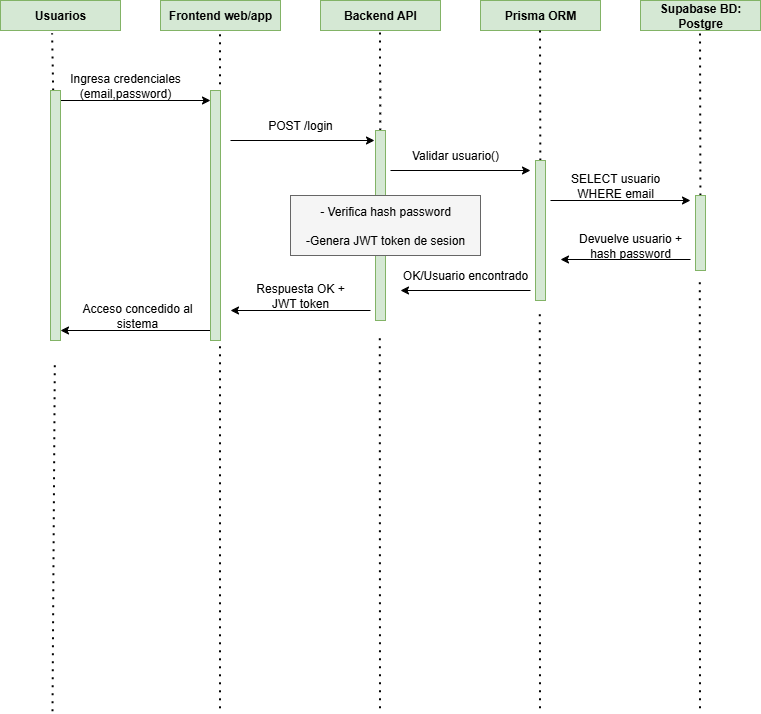
* 1. **Parte Dinámica (Diagrama de Secuencias)**

**Ilustración 5: Diagramas de Secuencias**

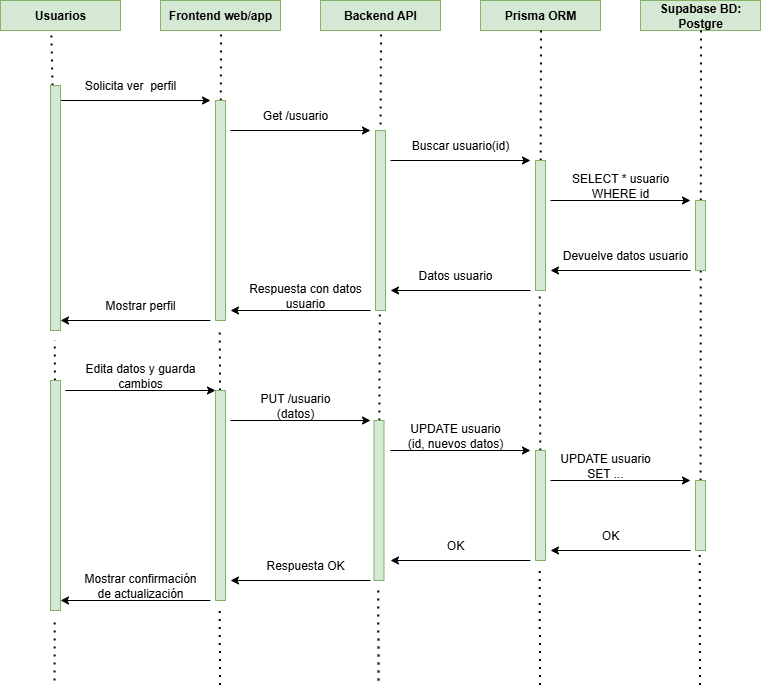
1.- Registro de usuario



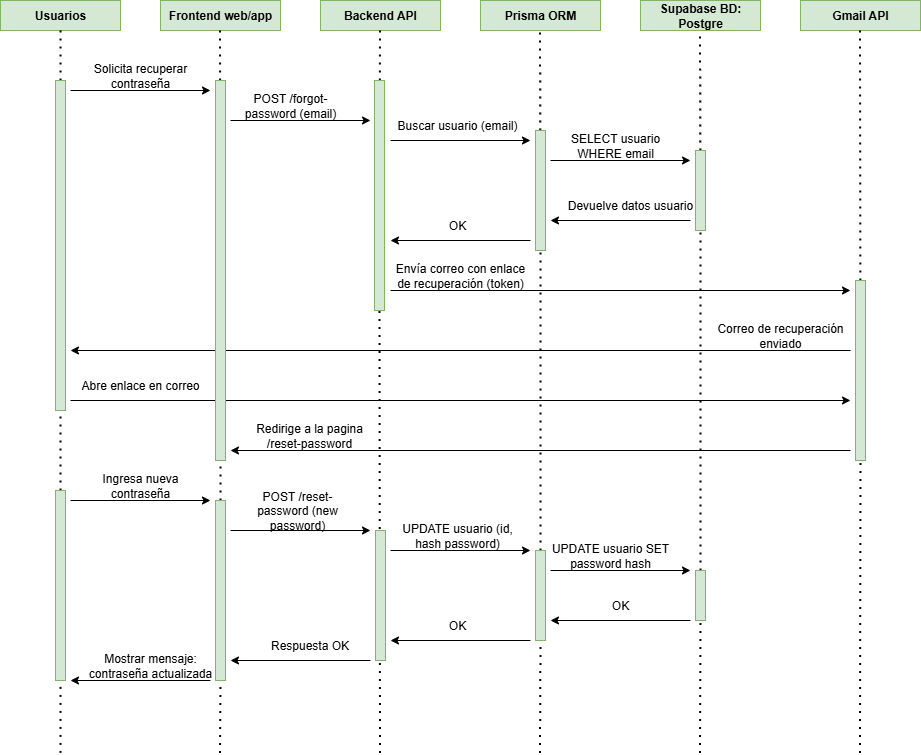
2.-Autenticar usuario (Login)



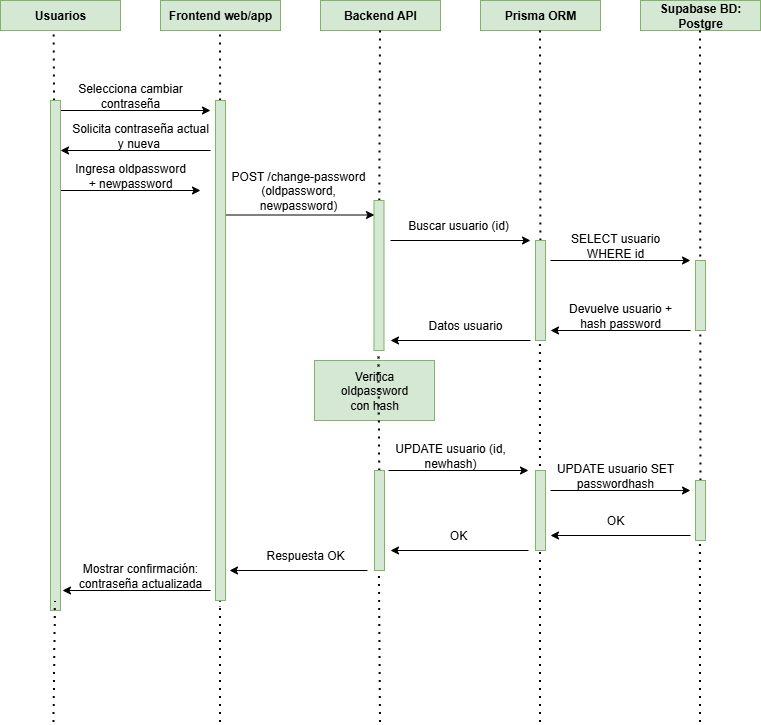
3.-Perfil de usuario



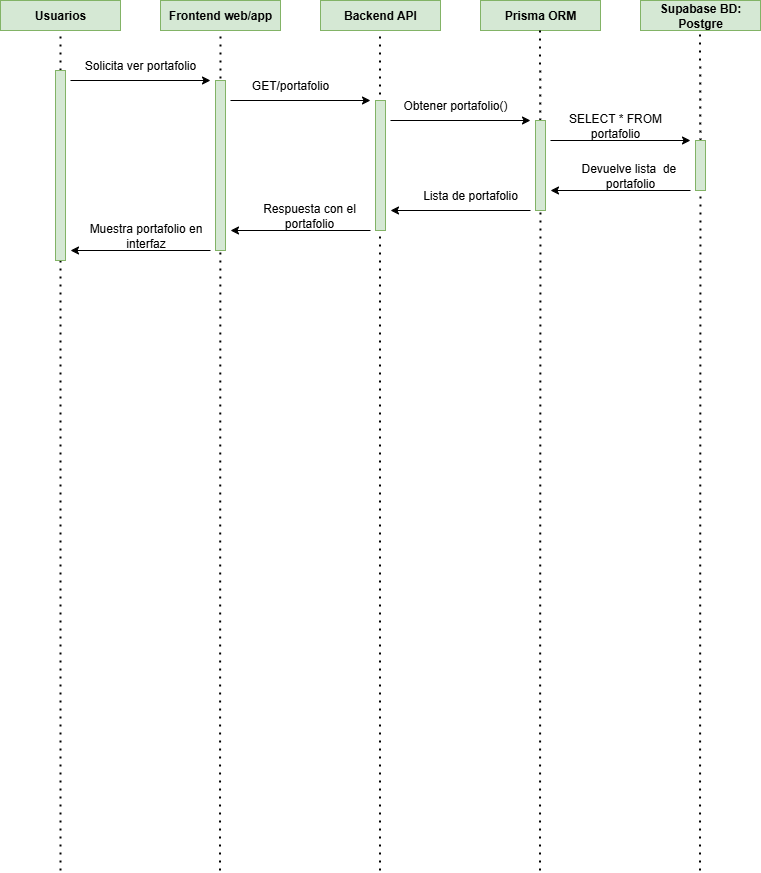
4.-Recuperar contraseña



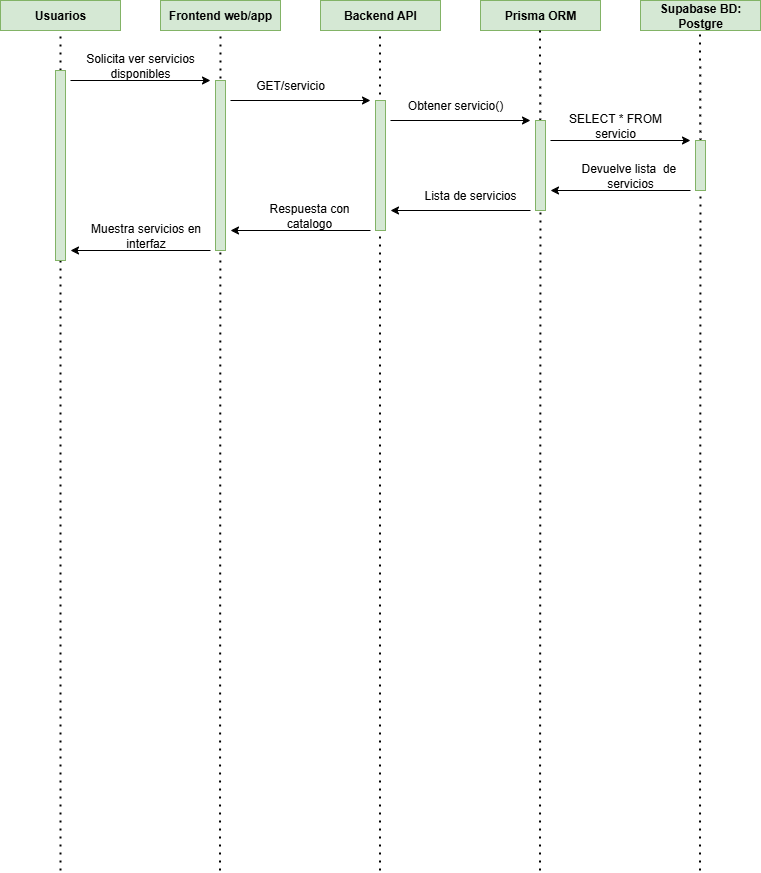
5.-Cambiar contraseña



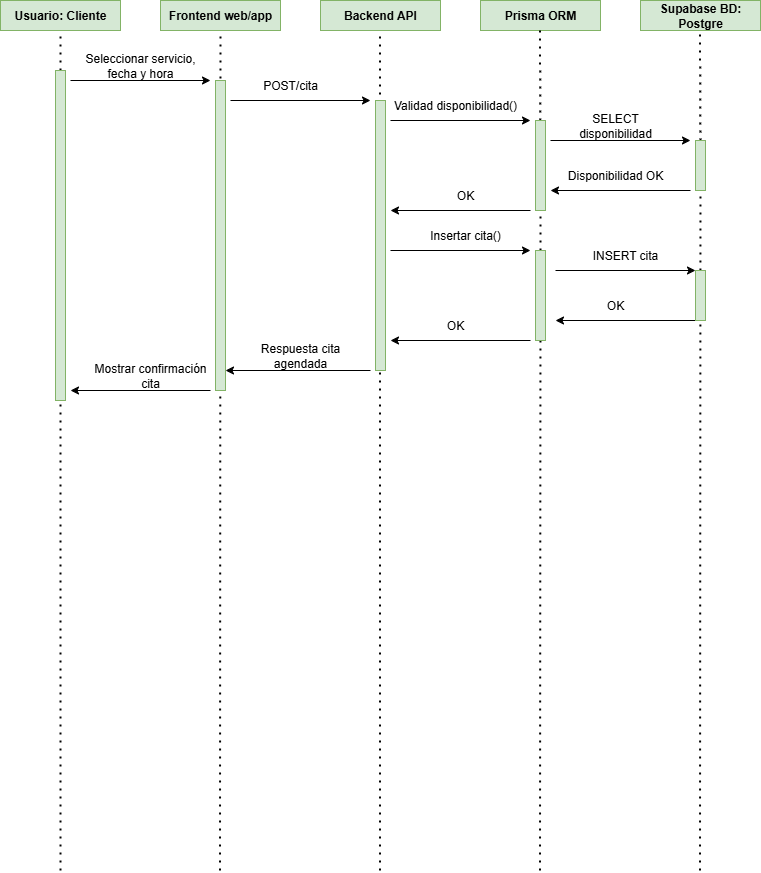
6.-Visualizar portafolio



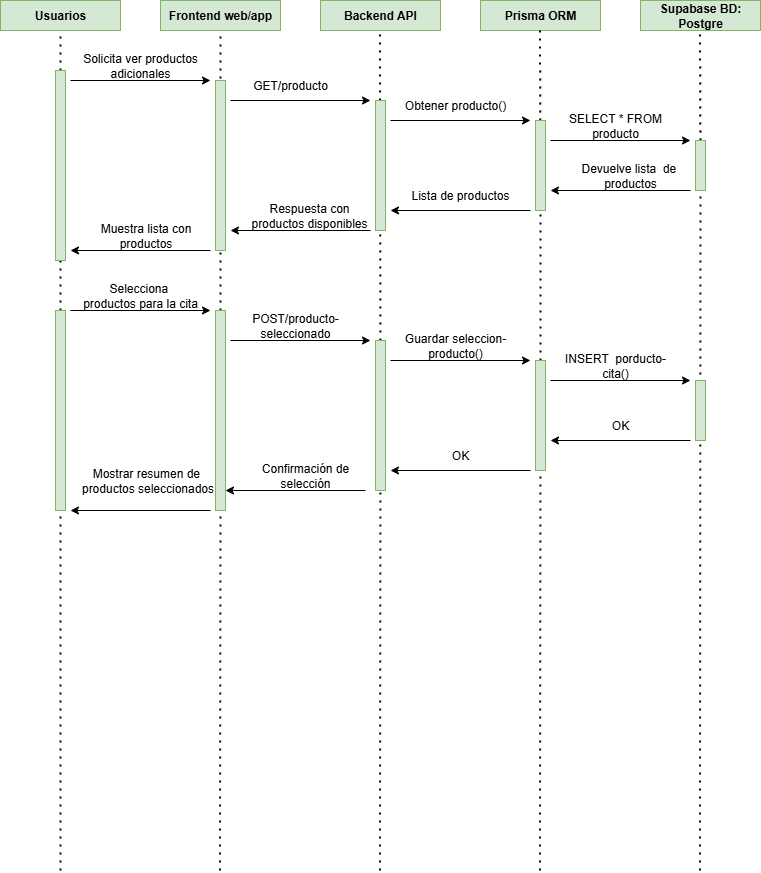
7.-Visualizar servicios (catálogo)



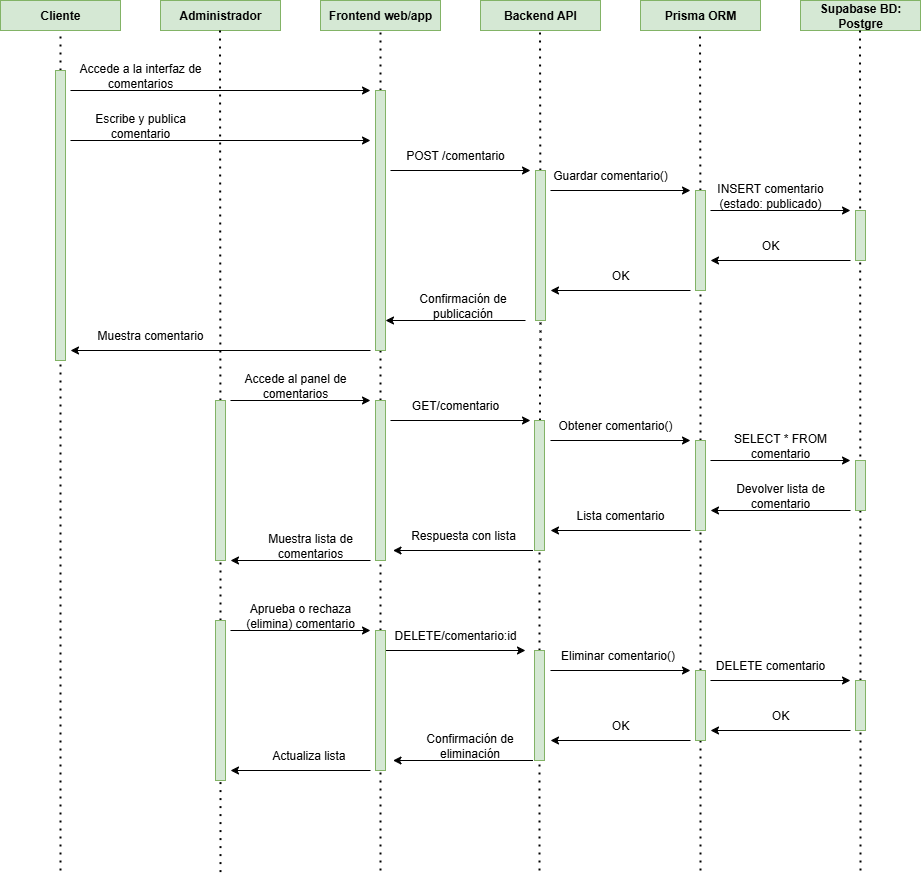
8.-Agendar cita (Cliente)



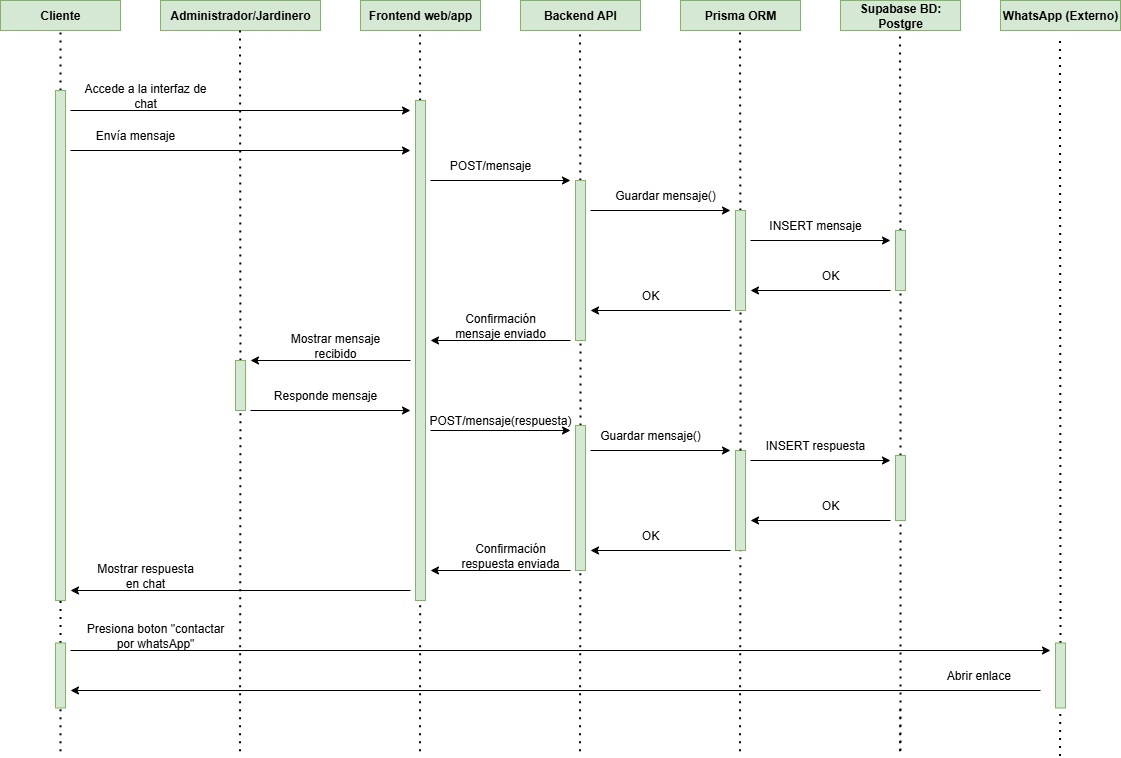
9.- Productos/insumos adicionales



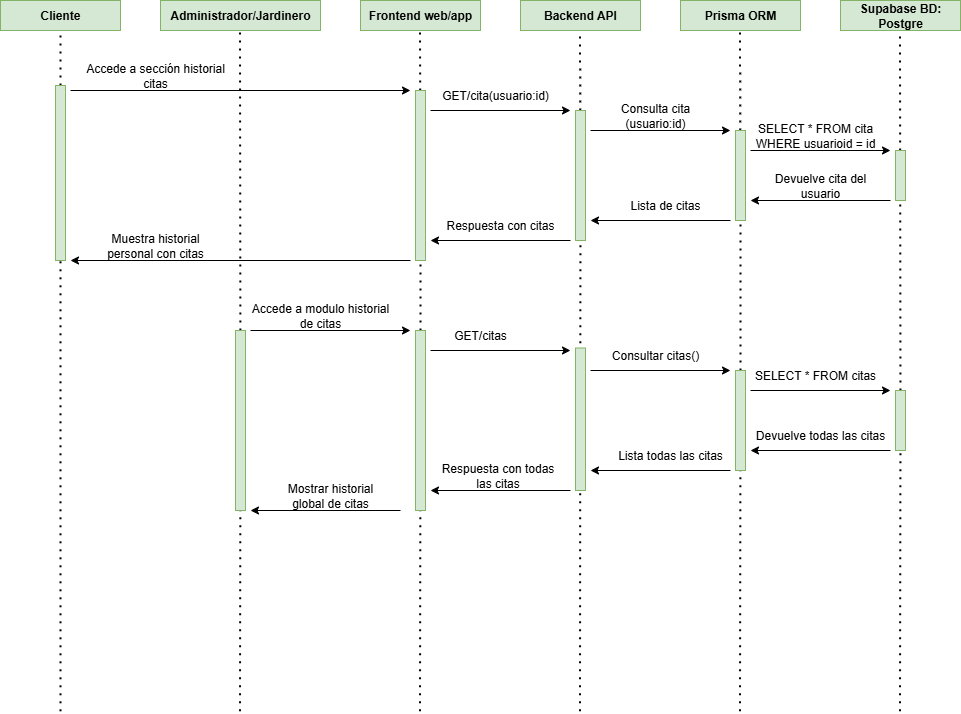
10.-Comentarios: Publicar y moderar



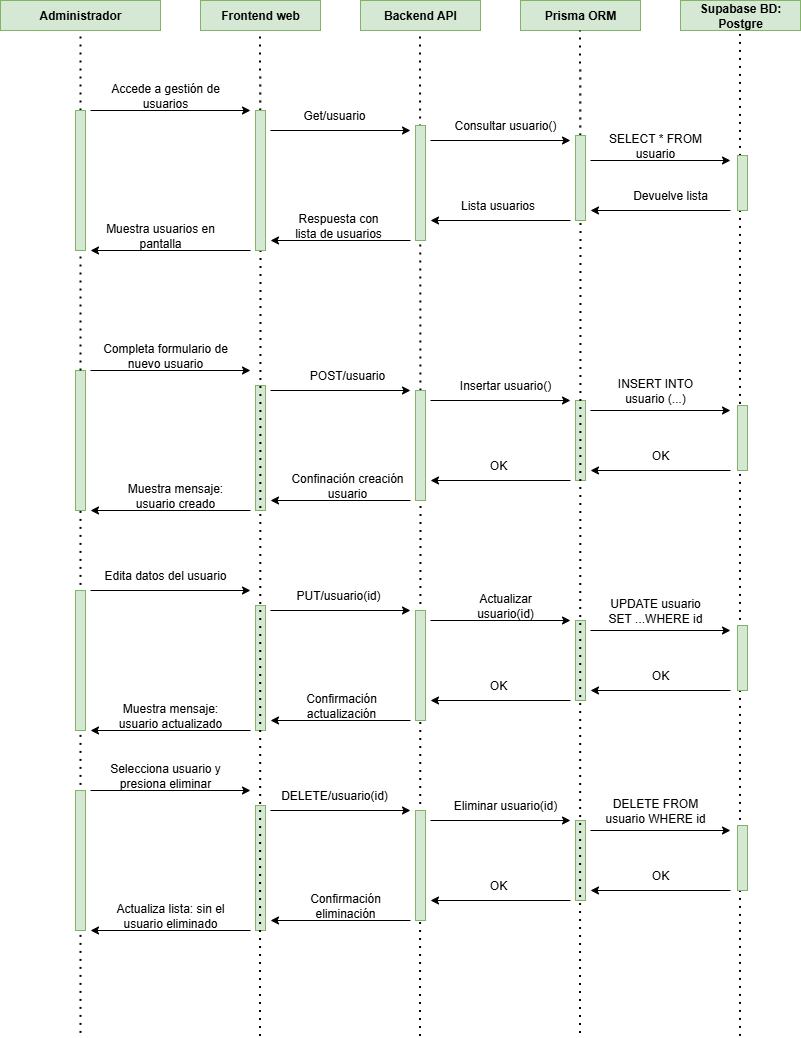
11.-Mensajería: Chat interno y redirección a WhatsApp



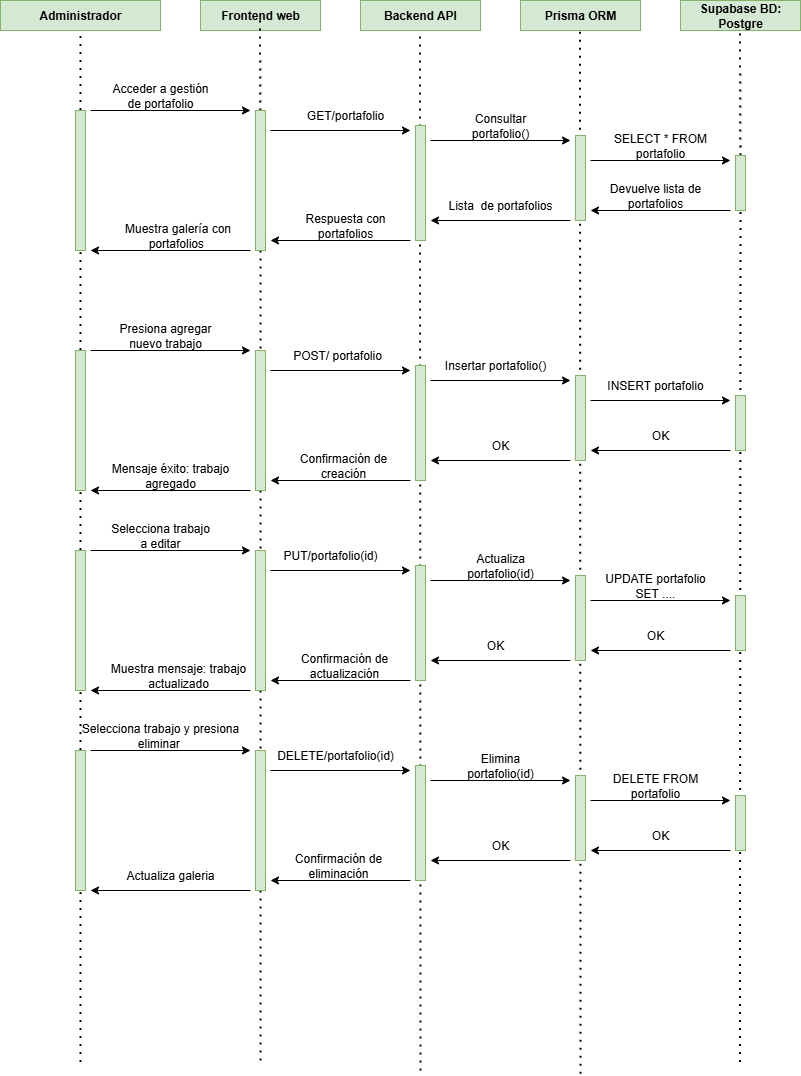
12.-Historial de citas



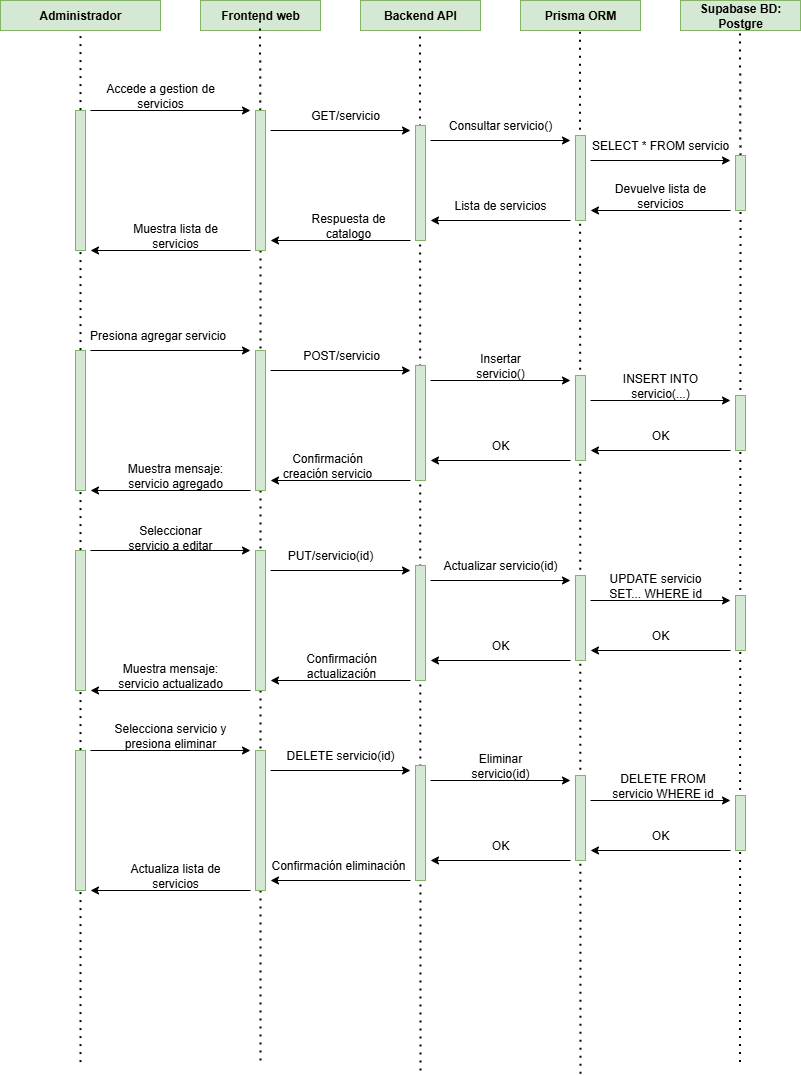
13.-Gestión de usuarios (Admin)



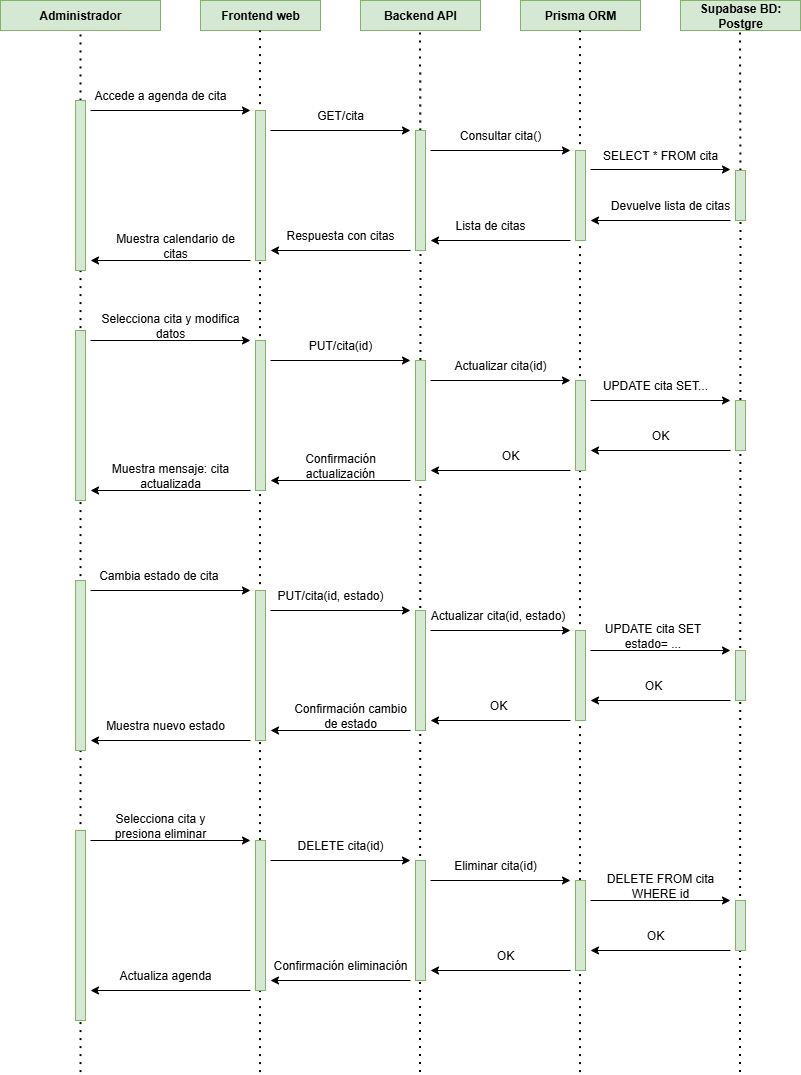
14.-Gestión de portafolio (Admin)



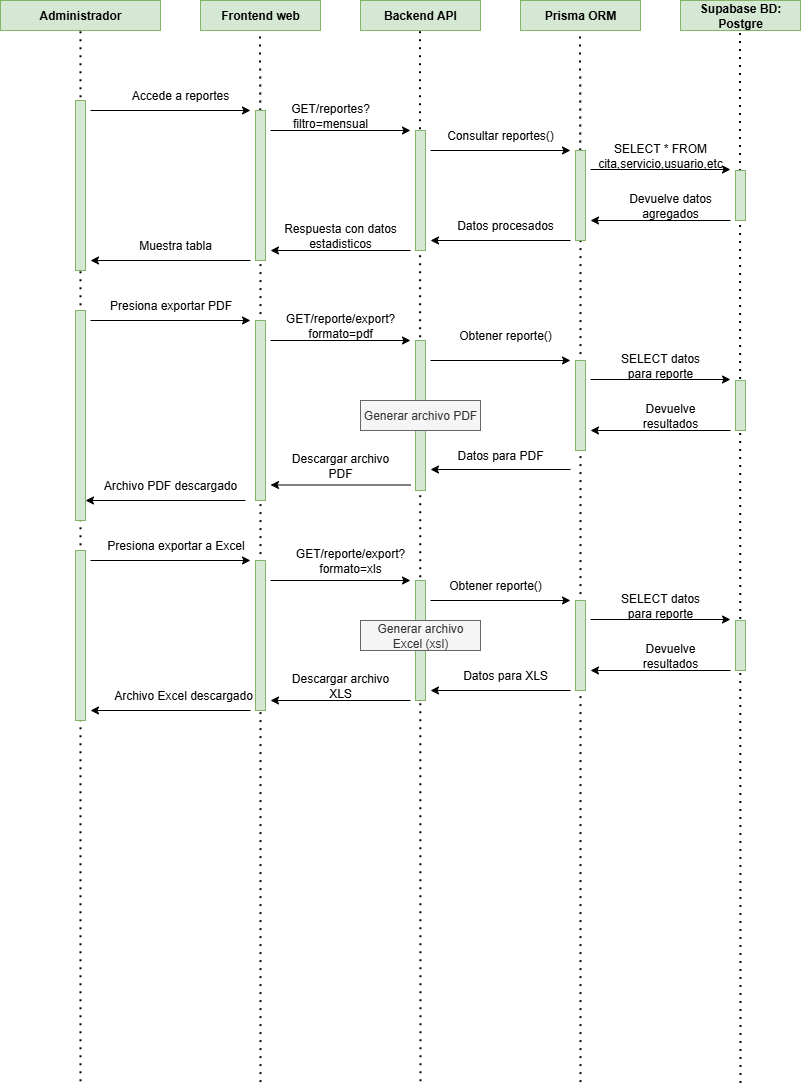
15.-Gestión de servicios (Admin)



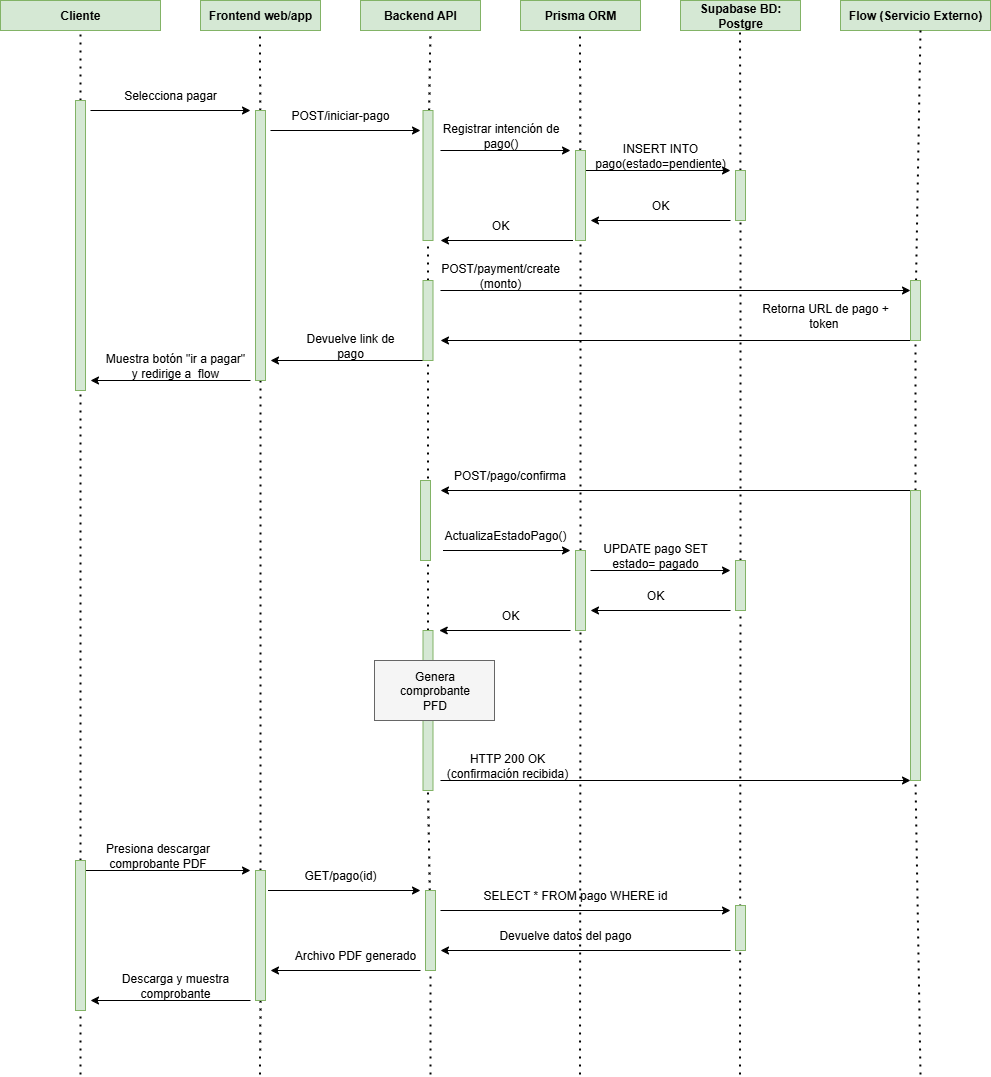
16.-Gestionar agenda y estados (Admin)



17.-Reportes estadísticos (exportable PDF/XLS)



18.-Pagos



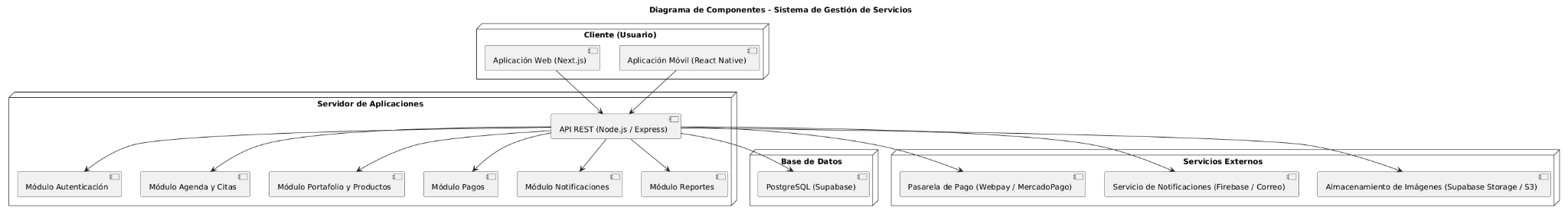
19.-Recordatorios y notificaciones push/correo



1. **Vista de Desarrollo o Despliegue**

En esta vista se describen las componente o modulos en las cuales se dividirá o implementará el sistema

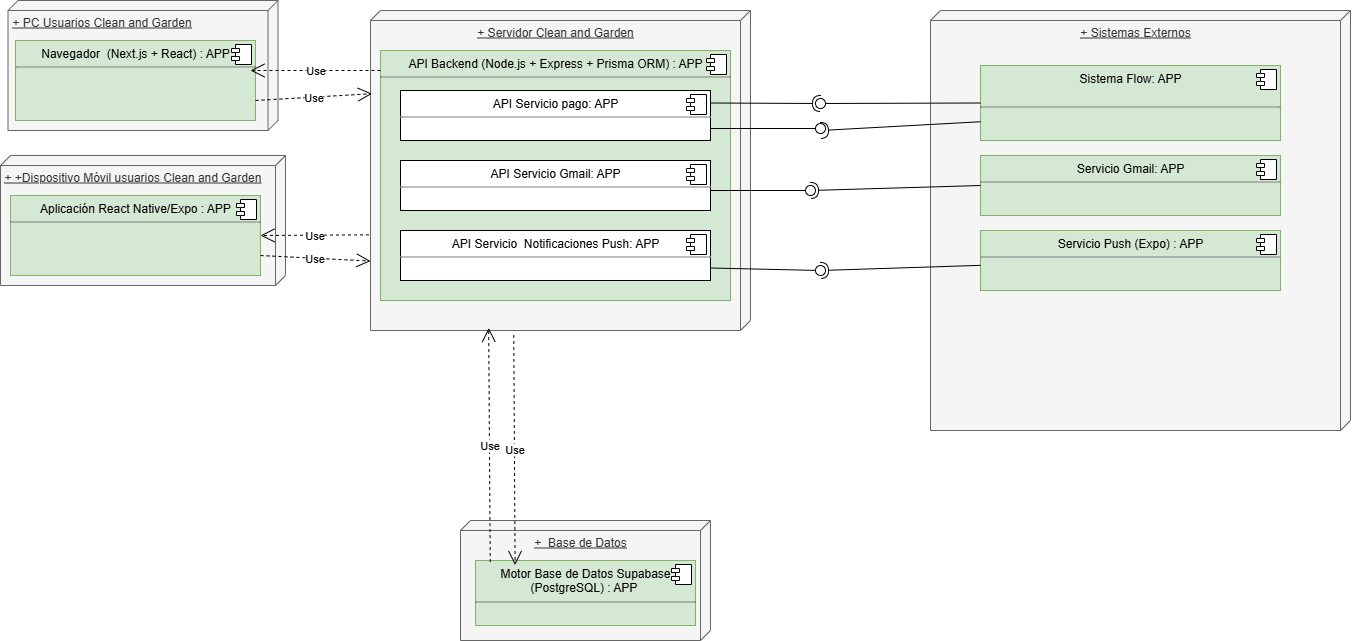
**Ilustración 6: Diagrama de componentes**



1. **Vista Física**

En esta vista se despliegan los nodos que participan con el sistema.

**Ilustración 7: Diagrama de Despliegue**

****

**Decisiones de Diseño y Selección de Alternativas**

Las principales decisiones arquitectónicas se tomaron considerando la restricción de tiempo de construcción, dado que el proyecto debe implementarse en un plazo de 4 meses y 1 semana. Por este motivo se privilegió la adopción de un stack tecnológico conocido, probado y de bajo riesgo en su implementación, que permite avanzar con rapidez y cumplir con los plazos establecidos.

La arquitectura fue diseñada de manera modular, de modo que cada componente (autenticación, usuarios, portafolio, pagos, notificaciones) pueda desarrollarse, probarse y mantenerse de forma independiente, disminuyendo la probabilidad de fallas y facilitando futuras extensiones del sistema.

Un segundo criterio fundamental en la selección fue la disponibilidad de conocimiento en el equipo. Las tecnologías seleccionadas (React, Next.js, React Native, Node.js, Prisma y Supabase) ya son familiares para los desarrolladores, lo que reduce la curva de aprendizaje y aumenta la productividad.

Otro aspecto considerado fue la interoperabilidad con servicios externos. Se eligieron herramientas que poseen SDKs y librerías oficiales (Flow para pagos, Gmail API para correos, Expo Push para notificaciones), garantizando una integración ágil y confiable.

Asimismo, se privilegió el uso de tecnologías open source y servicios cloud en modalidad gratuita o de bajo costo, lo que responde a la restricción de no adquirir licencias de software adicionales y permite asegurar la viabilidad económica del proyecto.

Finalmente, se prioriza una arquitectura escalable y mantenible, soportada en Supabase (PostgreSQL en la nube) y Prisma ORM, lo que facilita el crecimiento del sistema en el futuro y simplifica la administración de datos.

1. **Análisis de Reutilización**

El diseño de la arquitectura considera la **reutilización de frameworks, librerías y servicios cloud existentes**, con el fin de optimizar tiempos de desarrollo, reducir costos y asegurar la calidad de los componentes:

1. **Frameworks de frontend (React, Next.js y React Native/Expo)**
   * Se reutilizan componentes de interfaz (UI components) y librerías de la comunidad, evitando tener que construir elementos desde cero.
   * Facilita la consistencia entre la aplicación web y la móvil.
2. **Prisma ORM**
   * Permite reutilizar un mapeo de datos estandarizado en lugar de programar consultas SQL manuales.
   * Incrementa la mantenibilidad y reduce la probabilidad de errores.
3. **Base de Datos Supabase (PostgreSQL gestionado)**
   * Se reutilizan las capacidades nativas de PostgreSQL y los servicios gestionados de Supabase (autenticación, Row Level Security, almacenamiento).
   * Evita la necesidad de implementar un motor de base de datos propio.
4. **APIs externas (Flow, Gmail API, Expo Push Notifications)**
   * Se reutilizan servicios ya existentes para pagos, notificaciones y correos electrónicos.
   * Esto evita desarrollar módulos complejos de pago, servidores de correo o infraestructura de notificaciones en tiempo real.
5. **Estilos y componentes visuales (Tailwind CSS y DaisyUI)**
   * Se reutilizan librerías de estilos y componentes pre construidos que permiten acelerar el diseño de la interfaz, manteniendo coherencia visual.