Ejemplo

Las relaciones no están verificadas.

<https://github.com/RafaelSolier/REST_API_Estudiantes.git>

Descripción

Marketplace de Servicios de Salud

# **1. Introducción**

En la actualidad, muchas personas enfrentan dificultades para encontrar profesionales de la salud confiables o negociar tarifas justas, mientras que los médicos independientes carecen de una plataforma centralizada que les permita gestionar su disponibilidad y clientes de manera organizada. Este proyecto busca resolver ambos problemas mediante una aplicación intuitiva que integre herramientas clave como sistemas de pago en línea, gestión de reservas y horarios y valoraciones transparentes.

Con el crecimiento de la economía colaborativa y la creciente demanda de soluciones digitales para necesidades cotidianas, una plataforma de este tipo no solo facilita la interacción entre ambas partes, sino que también promovería la formalización de trabajos independientes. Además, al incluir funcionalidades como reservas en tiempo real y pagos seguros, se reduce la incertidumbre y se fomenta la confianza en las transacciones. El proyecto se alinea con tendencias tecnológicas actuales y se enfoca en entregar un producto mínimo viable (MVP) que demuestre su funcionalidad.

El objetivo es diseñar un software de marketplace para simplificar el proceso de búsqueda y reserva de servicios médicos, brindando una aplicación web eficiente, segura y accesible tanto para clientes como para profesionales de la salud.

# **2. Organización**

HabiMed es una plataforma digital que conecta profesionales de la salud con pacientes, ofreciendo agendamiento automático de citas mediante una página web o un chatbot dentro de la página web, integración con Google Calendar para manejo de reservas, procesamiento de pagos digitales y verificación de identidad.

El rubro de HabiMed es la tecnología para servicios de salud, facilitando y automatizando procesos en la interacción de usuarios y médicos.

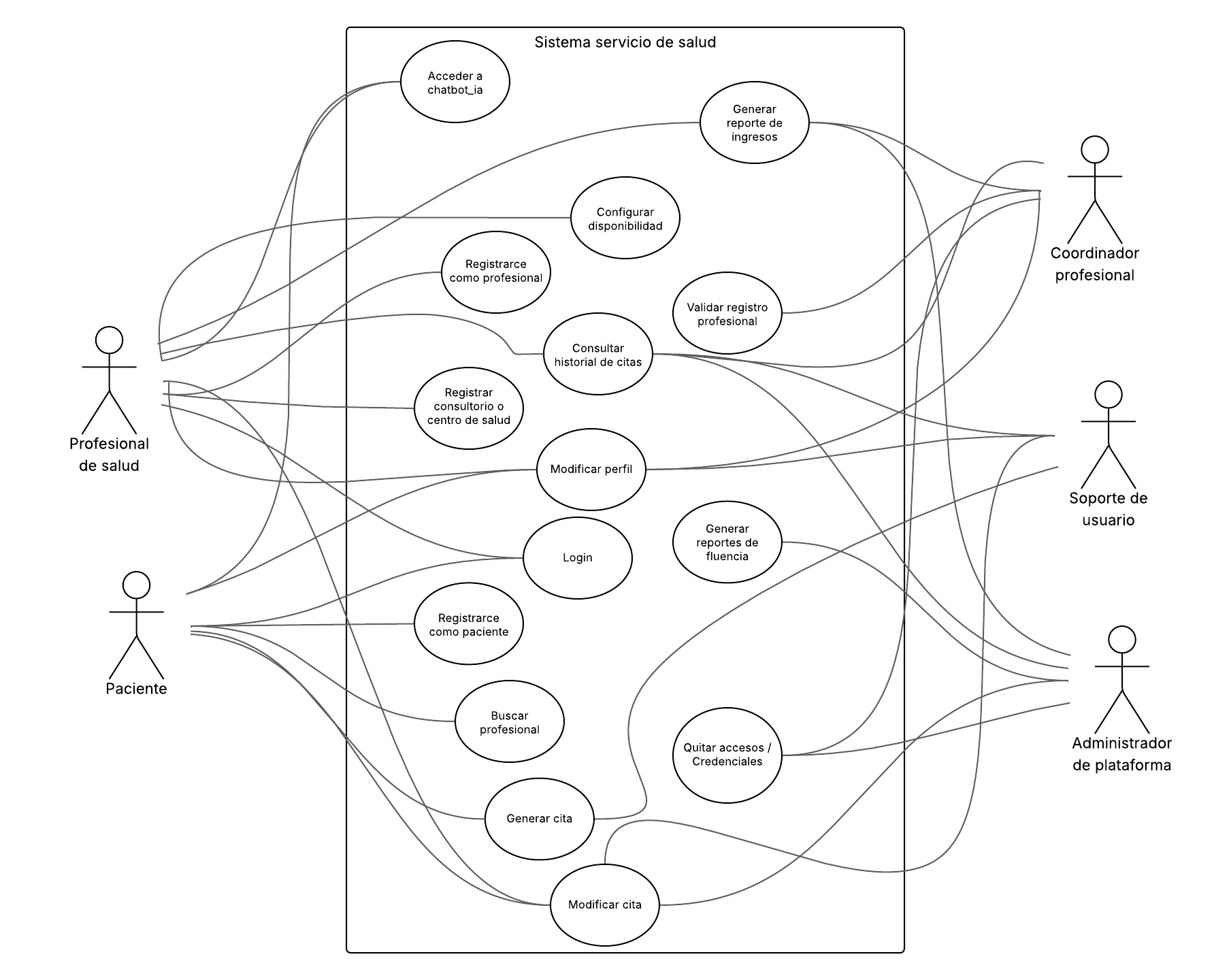
HabiMed es un proveedor de SaaS.

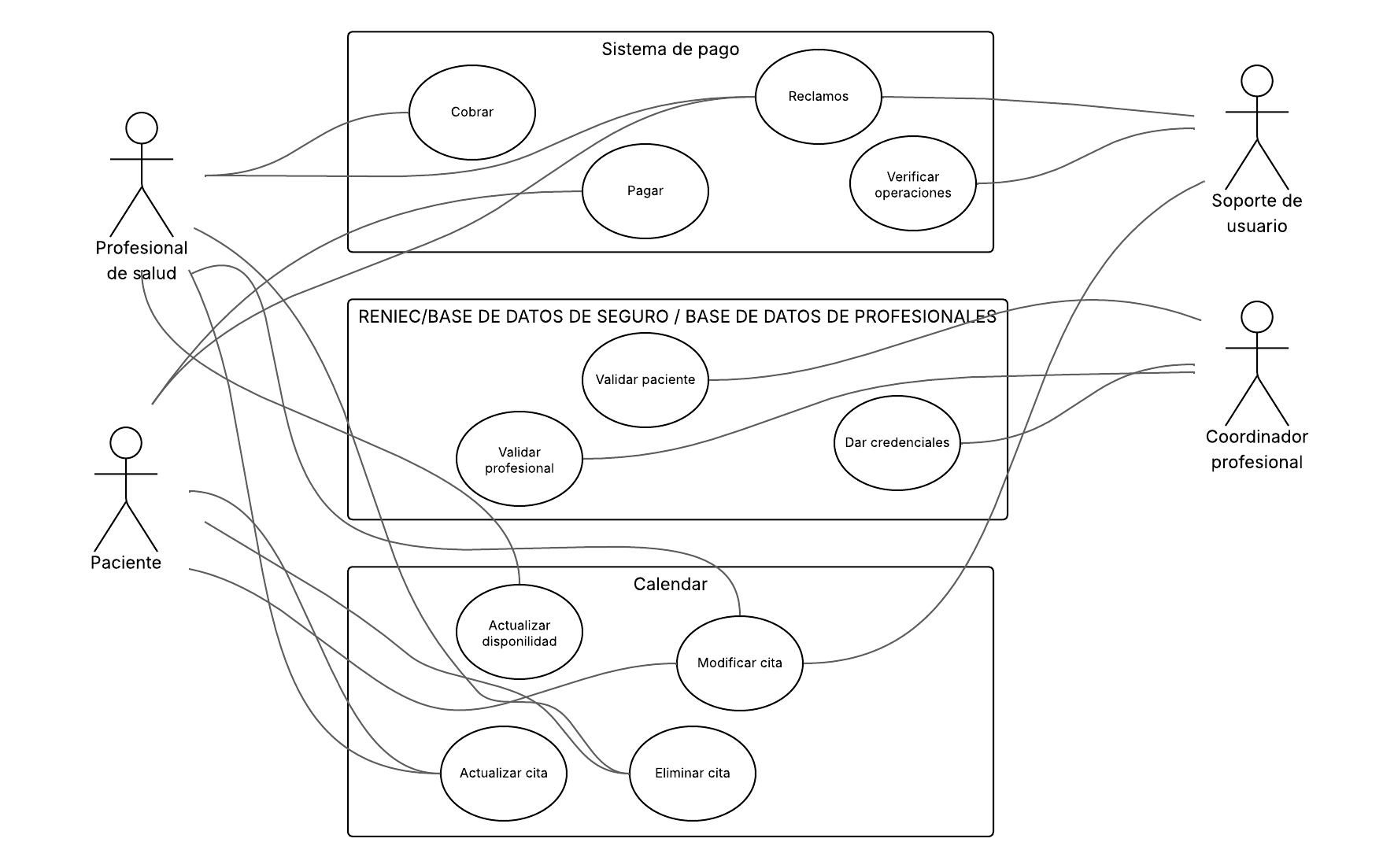
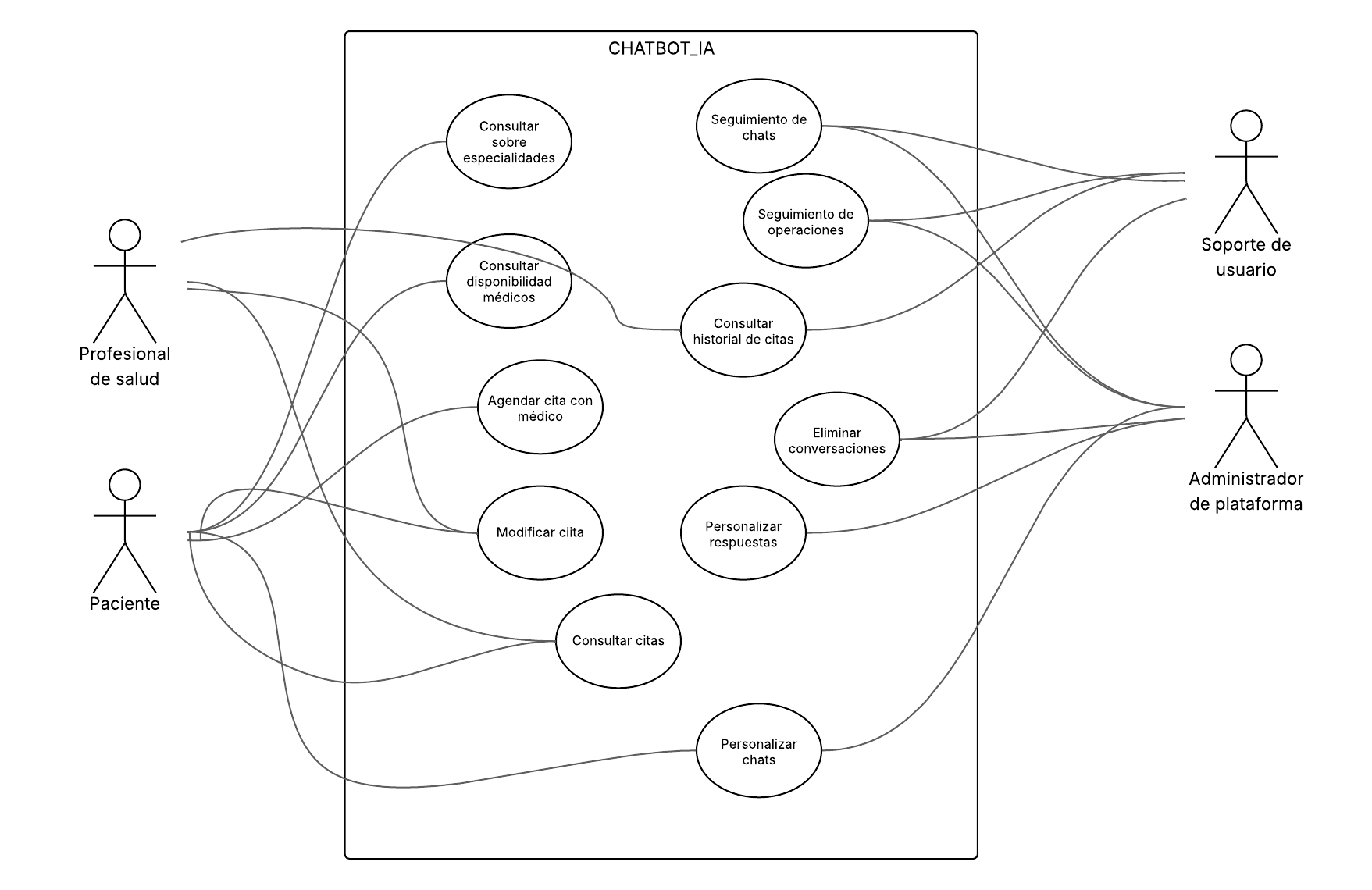
El modelo de negocio es Freemium (gratis para pacientes, suscripción/comisión para profesionales) permitiendo un acceso sin impedimentos a los pacientes y acercando este servicio básico a quienes lo necesiten.

# **3. Actores del Negocio y Casos de Uso**

| Actor | Descripción | Funciones |
| --- | --- | --- |
| Paciente | Persona que busca servicios de salud | * Registrarse como paciente e Iniciar sesión. * Buscar profesiona~~l o centro de salud.~~ * Interactuar con chatbot para diagnóstico preliminar, ayuda en búsqueda o consultas generales. * Agendar cita * Modificar cita * Realizar pago * Consultar historial de citas * Editar datos personales (no DNI, nombre ni apellido) * Acceder al chat de soporte |
| Profesional de Salud | Médico, odontólogo, oftalmólogo, etc que ofrece servicios de salud | * Registrarse como profesional e iniciar sesión. * Gestionar perfil y especialidades   - Agendar cita -aceptar citas   * Agregar información detallada de sus servicios (precio, riesgos, requisitos, etc) * Configurar disponibilidad horaria * Administrar agenda de citas * Generar reportes de ingresos |
| Administrador | Gestor de la plataforma | * Gestionar usuarios * Gestionar categorías/especialidades * Generar reportes globales * Configurar integraciones (pagos, RENIEC, Google) |
| Soporte del usuario | Encargado de atender incidencias básicas | * Consultar si hay peticiones de ayuda * Acceder a información de los usuarios * Acceder al chat de soporte |
| Coordinador de profecionales | Encargado de validar la información de los profesionales | * Acceder a la información de los profesionales * Actualizar información de los profesionales * Desactivar cuentas de profesionales |
| Sistema de Pago | Yape/Plin (externo) | * Gestionar el pago |
| RENIEC | Servicio de verificación de identidad (externo) por medio de DNI | * Verificar Los datos del usuario por DNI |
| Google Calendar | Sistema de calendario (externo) | * Registrar citas para un profesional |

# **4. Diagrama de casos de uso**





# **5. Identificar a los trabajadores del negocio**

1. **Paciente**

Es uno de los usuarios finales de la aplicación que realiza las reservas de citas, el pago de las mismas y administrar sus citas.

1. **Profesional**

Es uno de los usuarios finales que ofrece sus servicios de salud en un horario determinado. Puede definir información detallada de sus servicios, su información personal y profesional, registrar y actualizar su horario disponible.

1. **Administrador de Plataforma**

Es el encargado de gestionar la configuración global de la plataforma. Puede obtener reportes globales, gestionar usuarios y gestionar categorías o especialidades. También tiene acceso a la consola de administración y herramientas de monitoreo.

1. **Soporte Técnico Nivel 1**

Encargado de atender incidencias básicas de usuarios (pacientes o profesionales) como problemas al registrarse, iniciar sesión o algún inconveniente con las citas.

1. **Coordinador de Profesionales**

Valida credenciales de médicos, verifica su información profesional, realiza visitas imprevistas a los consultorios y gestiona altas/bajas de sus cuentas. Emplea sistema de verificación de colegiaturas y CRM interno. En el software puede consultar la información de los profesionales de salud, realizar cambios de su información o desactivar su cuenta.

# **6. Identificar las entidades del negocio**

* **Persona**

Son personas naturales que tienen como atributos **DNI (id)**, **nombres**, **apellidos, Correo, celular, dirección y contraseña.**

* **Usuario**

Son personas registradas en el sistema que pueden ser de los siguientes tipos: *paciente*, *doctor*, *administrador* y *coordinador*. Cada uno tendrá diferentes permisos dentro de la aplicación. Un doctor puede ser también paciente u otro tipo de usuario. Tienen los siguientes atributos **dniPersona** (primary key y foreign key de Persona)y **tipoUsuario** (primary key)**.**

* **Consultorio**

Es el lugar donde trabaja un **Doctor** y pueden trabajar varios doctores en un mismo consultorio. Tiene como atributos **idConsultorio** (primary key)**, nombre, ubicación (coordenadas), dirección y teléfono.**

* **Doctor\_Trabaja\_Consultorio**

Relación entre un **doctor** que trabaja en un **consultorio**, cada Doctor trabaja en un solo consultorio y en un consultorio pueden trabajar varios doctores. Tiene como atributos **idDoctor** (primary key y foreign key de Usuario), **idConsultorio** (foreign key de Consultorio), …. info profesional de doctor, ese código de los médicos, por ejemplo

* **Servicio**

Son los diferentes tipos de servicios que se ofrecen en un **consultorio**. Cada servicio pertenece a una sola **especialidad** y tiene como atributos **idServicio** (primary key), **idEspecialidad** (foreign key de Especialidad), **nombre**, **descripción**, **riesgos**, …Hay más características de un servicio?

* **Especialidad**

Son especialidades de salud que tienen diferentes **servicios** asociados. Tiene como atributos un identificador único **idEspecialidad**, **nombre** y **descripción**.

* **Horario\_Doctor**

Es el horario de trabajo que define el **doctor** a lo largo de la semana, por cada día de la semana. Se considera que el horario definido se mantiene para las semanas siguientes. De modo que, si el Doctor no podrá recibir pacientes por las tardes la sub-siguiente semana, por ejemplo, debería modificar su horario lo antes posible para evitar que un paciente pueda reservar una cita para esa semana. Tiene como atributos **idDoctor** (primary key y foreign key de Usuario), **horaSemana**, **duración** (indica la vigencia del horario en semanas). Las horas por cada día se guardan siguiendo un formato JSON en **horaSemana** para poder manejar distintas franjas horarias por cada día. Ejemplo de formato de horaSemana:

{

L: ["7-10", "15-18"],

M: ["7-9", "13-15", "17-20"],

X: ["7-10", "15-18"],

J: ["7-9", "13-15", "17-20"],

V: ["7-10", "15-18"],

S: ["7-9", "13-15", "17-20"],

D: ["7-10", "15-18"]

}

* **Cita**

Es la cita que agenda un **paciente** con algún **doctor** que tiene que ser aceptada por el doctor para recién generar el token de pago y después de pagado pasa a ser agendada en el calendario del doctor. Tiene como atributos **idCita** (primary key), **idPaciente** (foreign key de Usuario), **idDoctor** (foreign key de Usuario), **motivo**, **fechaInicio**, **fechaFin**, **estado** (solicitado, aceptado, pagado, modificado, realizado y cancelado) y **descripción**.

* **Reseña**

Es un comentario que otorga un **paciente** sobre un **doctor** con el que haya tenido una **cita** y califica la atención brindada. Tiene como atributos **idTrabajador** (primary key y foreign key de DoctorTrabajaConsultorio), **calificación**, **comentario** y **fecha**.

* **Detalle\_Pago**

Contiene la información sobre el pago realizado para una **cita** confirmada. Cada pago solo pertenece a una cita y al generarse una solicitud de pago se define un tiempo máximo en el que se debe de pagar para confirmar la reserva de la cita. Tiene como atributos **idCita** (primary key y foreign key de Cita), **monto**, **método**, **estado** (pendiente, pagado) y **duración**.

* **Diagnóstico**

Contiene la descripción técnica y detallada del diagnóstico realizado en la **cita**. Una cita no necesariamente tiene un diagnóstico. Tiene como atributos **idCita** (primary key y foreign key de Cita) y **descripción**.

* **Receta**

Contiene información detallada de la medicación recetada al paciente tras la **cita**, pero no todas las citas tienen una receta. Tiene como atributos **idCita** (primary key y foreign key de Cita) y **descripción**.

* **Recomendaciones**

Contiene información detallada de las recomendaciones o dosis de los medicamentos de la receta dada al paciente tras la **cita**, pero no todas las citas tienen una recomendación. Tiene como atributos **idCita** (primary key y foreign key de Cita) y **descripción**.

* **Permisos\_Historial**

Define los permisos brindados por un **paciente** a un **doctor** para poder ver su historial médico, donde el doctor podrá ver las recetas, recomendaciones y diagnósticos dados al paciente en sus diferentes citas. El paciente puede quitarle los permisos al doctor cuando lo desee. Los permisos otorgados serán registrados con fechas de inicio y fin. Tiene como atributos **idPermisoHistorial** (primary key), **idDoctor** (foreign key de Usuario), **idPaciente** (foreign key de Usuario), **fechaOtorgaPermiso**, **fechaDeniegaPermiso** y **estado** (otorgado o denegado).

# **7. Diagrama de clases del negocio**

# 

# **8. Reglas de negocio**

1. Todo profesional debe verificar su identidad con RENIEC antes de ofrecer servicios
2. Las citas deben confirmarse automáticamente si el pago se completa hasta en máximo 24h después de aceptada la solicitud de cita.
3. El chatbot solo sugiere profesionales y posibles servicios a consultar de acuerdo a los síntomas y diagnósticos del paciente, no da diagnósticos médicos.
4. Los pagos se dividen: 80% al profesional, 20% a la plataforma
5. Cancelaciones de citas con menos de 24h de antelación generan penalidad del 50%.
6. Profesionales deben tener al menos 5 slots disponibles por semana
7. Pacientes pueden calificar profesionales después de cada cita.
8. El orden de listado de los profesionales se hace de forma alfabética para evitar sesgos, pero la recomendación del chatbot se basa en la calificación de los médicos, mostrando los 5 mejores para el servicio.
9. Datos médicos son cifrados y solo accesibles por profesionales involucrados siempre y cuando el paciente dé su consentimiento, el cual puede revocar cuando desee.
10. Se considera que el horario definido se mantiene para las semanas siguientes. De modo que, si el Doctor no podrá recibir pacientes por las tardes la subsiguiente semana, por ejemplo, debería modificar su horario lo antes posible para evitar que un paciente pueda reservar una cita para esa semana.

# **9. Diagrama de actividades del negocio**

