MÓDULOS E INTERFACES

1.- Módulo de Autenticación de Usuarios

**Partes principales:**

* **Front-End (Web y Móvil):**
  + **Pantallas:**
    - Registro de usuarios: formulario para ingresar nombre, correo y contraseña.
    - Inicio de sesión: pantalla con opciones de ingreso (correo/contraseña o redes sociales).
    - Recuperación de contraseña: formulario para enviar un correo de recuperación.
  + **Validaciones:**
    - Comprobar formato correcto del correo.
    - Contraseñas seguras (mínimo 8 caracteres, con números y símbolos).

**Back-End:**

* **Servicios:**
  + Gestión de credenciales (registro, validación de contraseñas).
  + Generación de tokens JWT para sesiones seguras.
  + Integración con servicios de terceros (Google, Facebook, Apple) para autenticación.
* **Endpoints REST:**
  + POST /api/usuarios: Registrar un nuevo usuario.
  + PUT /api/usuarios/{id}: Editar información del usuario.
  + DELETE /api/usuarios/{id}: Eliminar un usuario.

2.- Módulo de Gestión de Mascotas

**Partes principales:**

* **Front-End (Web y Móvil):**
  + **Pantallas:**
    - Lista de mascotas: muestra las mascotas registradas por el usuario.
    - Detalles de mascota: permite visualizar información completa de cada mascota.
    - Formulario para añadir/editar mascotas.
  + **Interactividad:**
    - Validaciones en tiempo real para campos como peso, edad, o raza.
    - Subida de fotos de las mascotas (móvil).
* **Back-End:**
* **Endpoints REST:**
  + GET /api/mascotas: Obtener la lista de mascotas.
  + POST /api/mascotas: Registrar una nueva mascota.
  + PUT /api/mascotas/{id}: Editar información de una mascota.
  + DELETE /api/mascotas/{id}: Eliminar una mascota.
* **Servicios:**
  + Validación de datos antes de guardar en la base de datos.
  + Asignación de mascotas al usuario correspondiente.

3.- Módulo de Monitoreo

**Partes principales:**

**Front-End (Web y Móvil):**

* **Pantallas:**
  + **Panel de métricas:** Visualización en tiempo real de datos de salud (ritmo cardíaco, temperatura, nivel de actividad, etc.).
  + **Alertas de salud:** Notificaciones y advertencias de métricas fuera de rango.
  + **Historial de salud:** Gráficos y reportes con datos históricos.
  + **Configuración del collar:** Pantalla para emparejar el collar con la aplicación y personalizar alertas.
* **Interactividad:**
  + Representación gráfica de datos históricos de salud.
  + Opciones para personalizar umbrales de alerta según las necesidades de la mascota.

**Back-End:**

* **Servicios:**
  + Recepción y almacenamiento de datos de salud enviados por el collar.
  + Procesamiento de datos para identificar métricas fuera de rango.
  + Generación de notificaciones y reportes de salud.
* **Endpoints REST:**
  + **POST /api/monitoreo:** Registrar datos de salud en tiempo real enviados por el dispositivo.
  + **GET /api/monitoreo/mascota/{mascotaId}:** Consultar los datos de salud actuales de una mascota.

4.- Módulo de Historial Médico

**Partes principales:**

* **Front-End (Web y Móvil):**
  + **Pantallas:**
    - Visualización del historial médico de una mascota.
    - Formulario para registrar o actualizar vacunas, enfermedades, y tratamientos.
  + **Características:**
    - Mostrar información organizada por categoría (vacunas, enfermedades, tratamientos).
    - Notificaciones de cambios en el historial (móvil).
* **Back-End:**
* **Endpoints REST:**
  + GET /api/historialMedico/mascota/{mascotaId}: Obtener el historial médico de una mascota.
  + POST /api/historialMedico: Registrar información médica.
  + PUT /api/historialMedico/{id}: Editar el historial existente.
* **Servicios:**
  + Validación y categorización de datos.
  + Relación del historial médico con una mascota específica.

5.- Módulo de Recordatorios

**Partes principales:**

* **Front-End (Web y Móvil):**
  + **Pantallas:**
    - Lista de recordatorios.
    - Formulario para añadir/editar recordatorios.
  + **Notificaciones Push (móvil):**
    - Recordatorios configurados para enviar alertas en fechas/hora específicas.
* **Back-End:**
  + **Endpoints REST:**
    - GET /api/recordatorios/mascota/{mascotaId}: Obtener lista de recordatorios de una mascota.
    - POST /api/recordatorios: Crear un recordatorio.
    - PUT /api/recordatorios/{id}: Editar un recordatorio.
    - DELETE /api/recordatorios/{id}: Eliminar un recordatorio.
  + **Servicios:**
    - Programación de notificaciones basadas en el campo fecha

6.- Módulo de Primeros Auxilios

**Partes principales:**

* **Front-End (Web y Móvil):**
  + **Pantallas:**
    - Lista de guías organizadas por categorías (ej. heridas, intoxicaciones, etc.).
    - Visualización detallada de cada guía de primeros auxilios.
  + **Características:**
    - Función de búsqueda para localizar guías específicas.
    - Diseño optimizado para lectura en móvil.
* **Back-End:**
  + **Endpoints REST:**
    - GET /api/primeros-auxilios: Obtener lista de guías.
    - GET /api/primeros-auxilios/{id}: Obtener detalles de una guía específica.
  + **Servicios:**
    - Administración de contenido desde el back-end.

7.- Interfaces y Comunicación

* **API RESTful:**
  + La API sirve como puente entre las aplicaciones Web/Móvil y la lógica del servidor.
  + Proporciona endpoints accesibles mediante HTTPS.
  + Utiliza JSON para intercambiar datos estructurados.
* **Notificaciones Push (móvil):**
  + Implementación para recordatorios y alertas relevantes.
* **Base de Datos:**
* PostgreSQL como motor de base de datos.
* Acceso y consultas realizadas desde el back-end a través de Entity Framework Core.

MODELO DE LA BASE DE DATOS

Escala de tiempo

El contenido generado por IA puede ser incorrecto.

**Estructura del modelo:**

* **Usuario:** Maneja la interacción principal con la aplicación.
* **Mascota:** Es el centro de atención del sistema.
* **Monitoreo:** Almacena información y datos biométricos obtenidos del dispositivo inteligente, así como los datos de localización de la mascota.
* **Historial Médico:** Almacena información de salud detallada para cada mascota.
* **Recordatorios:** Gestiona notificaciones importantes para los dueños.
* **Veterinario y Primeros Auxilios:** Proporcionan recursos adicionales al usuario, como detalles de profesionales y guías de emergencias.

1.- Usuario

* **Atributos:**
  + Contiene datos básicos del usuario, como nombre, email, y contraseña, además de una relación con sus mascotas.
* **Métodos:**
  + iniciarSesion(), registrarse() y actualizarPerfil() permiten gestionar la autenticación y la administración de datos personales.
* **Relación:**
  + **1 a \* con Mascota:** Un usuario puede registrar múltiples mascotas, pero cada mascota pertenece a un solo usuario (idUsuario en Mascota referencia al Usuario).

2.- Mascota

* **Atributos:**
  + Información de cada mascota registrada, como nombre, fechaNacimiento, especie, raza, peso, e imagen asociada.
* **Métodos:**
  + Permiten añadir y actualizar información de la mascota.
* **Relaciones:**
  + **1 a 1 con HistorialMedico:** Cada mascota tiene un único historial médico.
  + **1 a \* con Recordatorio:** Cada mascota puede tener varios recordatorios asociados.
  + **1 a 1 con Usuario:** Relación ya descrita en la clase Usuario.

3.- HistorialMedico

* **Atributos:**
  + Almacena información específica del historial médico de cada mascota, como vacunas, enfermedades y tratamientos.
* **Métodos:**
  + agregarHistorial() y actualizarHistorial() permiten gestionar la información médica.
* **Relación:**
  + **1 a 1 con Mascota:** Cada mascota tiene su propio historial médico, y cada historial pertenece a una sola mascota.

4.- Recordatorio

* **Atributos:**
  + Define recordatorios configurados para las mascotas, incluyendo el tipo (vacuna, revisión, etc.), fecha, si es recurrente, y si hay una notificación activada.
* **Métodos:**
  + Permiten crear, enviar y editar recordatorios.
* **Relación:**
  + **1 a \* con Mascota:** Cada mascota puede tener múltiples recordatorios asociados.

5.- Veterinario

* **Atributos:**
  + Almacena datos del veterinario, como nombre, dirección, y informaciónContacto.
* **Métodos:**
  + verDetalles() para acceder a los datos del veterinario.
* **Relación:**
  + En este modelo no tiene conexión directa con otras clases, pero puede integrarse en el futuro con Mascota o HistorialMedico para registrar veterinarios específicos.

6.- PrimerosAuxilios

* **Atributos:**
  + Contiene titulo, contenido y categoria para organizar y mostrar guías de primeros auxilios.
* **Métodos:**
  + consultarGuia() y listarGuias() para acceder a las guías.
* **Relación:**
  + Es una clase independiente y no tiene relaciones directas con otras

7.- Monitoreo

* **Atributos:**
  + Contiene id, id de la mascota, y registra datos automáticos como la temperatura, el ritmo cardiaco, la variabilidad de frecuencia cardiaca, la respiración, actividad, y ubicación para la localización de la mascota.
* **Métodos:**
  + registrarMetrica(), obtenerUltimaMetrica(), alertaSalud(), para poder observar y tener un registro de todos los datos obtenidos, además de que la aplicación avise al usuario si detecta algo fuera de lo ordinario..
* **Relación:**
  + **1 a \* con Mascota:** Una mascota puede tener múltiples registros de monitoreo de salud.