**Sistema de Gestión para Clínica Veterinaria**

**Descripción General:** Los estudiantes deben desarrollar un sistema de gestión para una clínica veterinaria. Este sistema incluirá la creación de una base de datos en MongoDB, un backend en NodeJS (Express) que se conecte a la base de datos, y una API Rest que responda a una aplicación frontend.

**Requerimientos Específicos**

**1. Creación de la Base de Datos**

* **Nombre de la base de datos:** vet\_clinic\_management

**2. Creación de Colecciones**

* **Colección pets:**
  + Campos:
    - pet\_id (string, único)
    - name (string)
    - species (string)
    - breed (string)
    - age (número)
    - owner (documento anidado con campos name, email, phone)
    - appointments (arreglo de documentos con campos appointment\_id, date, reason)
* **Colección vets:**
  + Campos:
    - vet\_id (string, único)
    - name (string)
    - specialty (string)
    - email (string)
    - phone (string)
* **Colección appointments:**
  + Campos:
    - appointment\_id (string, único)
    - pet\_id (string, referencia a pets.pet\_id)
    - vet\_id (string, referencia a vets.vet\_id)
    - date (fecha)
    - reason (string)
    - diagnosis (string)
    - prescription (string)

**3. Manipulación de Documentos**

* **Inserción de Documentos:**
  + Inserta diez mascotas en pets.
  + Inserta cinco veterinarios en vets.
  + Inserta quince citas en appointments.
* **Eliminación de Documentos:**
  + Elimina una mascota específica (por ejemplo, con pet\_id = "pet123").
* **Actualización de Documentos:**
  + Actualiza el diagnóstico de una cita.
  + Actualiza la información de contacto de un propietario de mascota.

**4. Consultas Simples y Relacionadas**

* **Consultas Simples:**
  + Consulta todas las mascotas registradas.
  + Consulta todos los veterinarios especializados en una cierta área.
* **Consultas Relacionadas:**
  + Consulta todas las citas programadas para una mascota específica.
  + Consulta todas las citas atendidas por un veterinario específico.

**5. Aplicación de Filtros**

* **Filtros Condicionales:**
  + Consulta mascotas de una cierta especie.
  + Consulta citas programadas para una fecha específica.

**6. Documentos Anidados**

* **En la colección pets:**
  + Incluye un documento anidado con la información del propietario (owner).

**7. Colecciones con Arreglos**

* **En la colección pets:**
  + Incluye un arreglo de citas (appointments) para cada mascota.
* **En la colección vets:**
  + No aplica en este caso.

**8. Notación de Punto y Búsqueda en Documentos Anidados**

* **Notación de Punto:**
  + Buscar mascotas cuyo propietario tenga un correo electrónico específico (por ejemplo, pets.owner.email = "owner@example.com").
  + Actualizar la especialidad de un veterinario utilizando notación de punto (por ejemplo, cambiar specialty).

**9. Búsqueda en Arreglos**

* **Coincidencias en un Arreglo:**
  + Buscar citas que involucren a una mascota con cierto pet\_id.
  + Buscar citas diagnosticadas con una condición específica.

**Rúbrica de Evaluación**

| **Criterio** | **Puntos** |
| --- | --- |
| **Base de datos y colecciones (20 puntos)** |  |
| Creación correcta de la base de datos vet\_clinic\_management | 5 |
| Creación de las colecciones pets, vets, appointments | 15 |
| **Manipulación de documentos (30 puntos)** |  |
| Inserción correcta de documentos en cada colección | 10 |
| Eliminación correcta de documentos específicos | 10 |
| Actualización correcta de documentos | 10 |
| **Consultas y filtros (30 puntos)** |  |
| Realización de consultas simples | 10 |
| Aplicación de filtros condicionales | 10 |
| Búsqueda utilizando notación de punto | 10 |
| **Documentos anidados y arreglos (20 puntos)** |  |
| Creación de documentos anidados en pets | 10 |
| Creación de colecciones con arreglos en pets, appointments | 10 |
| **Total** | **100** |