

SANTA
BÁRBARA
CLÍNICA VETERINARIA

SANTA BÁRBARA APP

Derechos de autor

Usted es libre de:

- **Compartir** — copiar y redistribuir el material en cualquier medio o formato
- **Adaptar** — remezclar, transformar y construir a partir del material.



Bajo los siguientes términos:



- **Atribución** — Usted debe dar crédito de manera adecuada, brindar un enlace a la licencia, e indicar si se han realizado cambios. Puede hacerlo en cualquier forma razonable, pero no de forma tal que sugiera que usted o su uso tienen el apoyo de la licenciante.
- **No Comercial** — Usted no puede hacer uso del material con propósitos comerciales.



- **Compartir Igual** — Si remezcla, transforma o crea a partir del material, debe distribuir su contribución bajo la misma licencia del original.

No hay restricciones adicionales — No puede aplicar términos legales ni medidas tecnológicas que restrinjan legalmente a otras a hacer cualquier uso permitido por la licencia.

Avisos:

No tiene que cumplir con la licencia para elementos del material en el dominio público o cuando su uso esté permitido por una excepción o limitación aplicable.

No se dan garantías. La licencia podría no darle todos los permisos que necesita para el uso que tenga previsto. Por ejemplo, otros derechos como publicidad, privacidad, o derechos morales pueden limitar la forma en que utilice el material.

Prefacio

Querido usuario,

¡Bienvenido al Manual Técnico de **Veterinaria Santa Bárbara App**! Este documento ha sido creado para proporcionarte una guía detallada sobre cómo está estructurada nuestra aplicación móvil de manera efectiva

Nuestra aplicación ha sido cuidadosamente desarrollada para brindarte una experiencia intuitiva y satisfactoria, con el objetivo de brindar un mantenimiento correcto del aplicativo.

En este manual, encontrarás una descripción detallada de la base de datos, las funcionalidades clave y las opciones de configuración disponibles. Además, te proporcionaremos soluciones a problemas comunes que puedas encontrar y te ofreceremos consejos y mejores prácticas para mejorar tu experiencia.

¡Gracias por confiar en nuestra app! Esperamos que este manual sea una herramienta útil y que disfrutes de una experiencia satisfactoria al utilizar nuestra app móvil.

¡Que disfrutes explorando y utilizando nuestra aplicación!



Atentamente,

El equipo de Veterinaria Santa Bárbara App

Requerimientos de desarrollo

Visual Studio Code



Visual Studio Code (VS Code) es un editor de código fuente ligero pero potente, de código abierto, desarrollado por Microsoft. Ofrece soporte para una amplia variedad de lenguajes de programación y tecnologías, y cuenta con un sistema de extensiones que permite personalizar y ampliar sus capacidades. Integra funciones de depuración, control de versiones Git y un terminal integrado, además de una amplia variedad de herramientas de desarrollo. Su diseño es altamente personalizable y se ha ganado la preferencia de muchos desarrolladores por su velocidad, eficiencia y plataforma cruzada, funcionando en Windows, macOS y Linux. En nuestro proyecto se utilizó para el desarrollo del API en Python.

Android Studio



Android Studio es un entorno de desarrollo integrado (IDE) oficial para desarrollo de aplicaciones Android, ofreciendo herramientas para diseño de interfaz de usuario, depuración, pruebas y perfilado de rendimiento. Soporta lenguajes como Kotlin, Java y C++, y utiliza Gradle para la automatización de compilaciones. Android Studio facilita la emulación de dispositivos, la gestión de dependencias y la integración con sistemas de control de versiones. Para principiantes, dispone de plantillas de proyectos y una amplia gama de plugins que extienden su funcionalidad. Aprender a usarlo eficientemente puede requerir tiempo para adaptarse a su rica gama de características y flujos de trabajo.

Python



Python es un lenguaje de programación interpretado, de alto nivel y de propósito general. Destaca por su legibilidad, simplicidad y amplio soporte de módulos y librerías. Soporta varios paradigmas de programación, es ampliamente utilizado en desarrollo web, análisis de datos y aprendizaje automático. Es querido por su comunidad por ser accesible para principiantes, a la vez que poderoso para expertos. La gestión eficiente de paquetes y su portabilidad lo hacen versátil en diversas plataformas.

Kotlin



Kotlin es un lenguaje de programación moderno, seguro, conciso y orientado a objetos, interoperable con Java y orientado a la JVM, que soporta también la compilación a JavaScript y Native. Es oficial para el desarrollo de Android, favoreciendo la productividad con características como inferencia de tipos, funciones de extensión y null safety. Se utiliza ampliamente para desarrollo de aplicaciones móviles, backend y frontend, con un creciente ecosistema y una sintaxis que facilita la escritura de código limpio y mantenible.

Estructura base de datos



La imagen muestra el esquema de base de datos MySQL, la cual gestiona información de administradores, doctores, pacientes (mascotas) y citas.

MySQL es un sistema de gestión de bases de datos relacional (RDBMS) basado en SQL (Structured Query Language). Es ampliamente utilizado, de código abierto y permite almacenar, modificar y recuperar datos estructurados a través de consultas e instrucciones.

En el esquema:

admin: Guarda los datos de los administradores, incluyendo un identificador único (id_admin), nombre (nombre_completo) y datos de acceso como correo y contraseña.

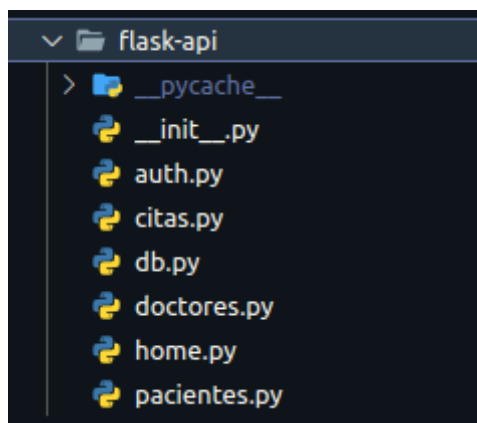
doctores: Almacena la información de los doctores, con un identificador único (id_doctor), datos personales, contacto, especialidad, horario laboral y control de acceso.

pacientes: Contiene los registros de las mascotas, con un identificador único (id_paciente), datos del animal y su propietario, incluyendo nombre, especie, raza y fecha de nacimiento, además de contacto y estado de la cuenta (habilitado/deshabilitado).

citas: Relaciona a pacientes y doctores, registrando cada cita con su propio identificador (id_cita), fechas, horas y el estado de la cita (habilitada/deshabilitada).

Las líneas entre las tablas indican las relaciones, sugiriendo que los doctores y pacientes están vinculados entre sí a través de las citas, lo que permite gestionar quién atiende a qué mascota y cuándo.

Estructura API Python



La estructura mostrada del API desarrollado en Flask, un framework de Python para desarrollar aplicaciones web. La organización de los archivos sigue un enfoque modular, donde cada aspecto de la aplicación tiene su propio script Python dedicado a manejar diferentes funciones y rutas de la API relacionadas con los modelos de datos de la aplicación. Aquí está el desglose:

`__init__.py`: Este archivo contiene la lógica de inicialización del paquete Flask. Aquí se define y configura la aplicación Flask, incluidos los blueprints.

`auth.py`: Este módulo se encarga de la autenticación y autorización dentro de la API. Maneja rutas para el registro de usuarios, inicio de sesión y cierre de sesión.

`citas.py`: Contiene la lógica para gestionar las citas a través de la API. Esto incluye funciones para crear, actualizar, eliminar y recuperar información de citas, haciendo uso de métodos HTTP como GET, POST, PUT y DELETE.

`db.py`: Este archivo es responsable de configurar la conexión a la base de datos y también contiene funciones de consulta, inserción y actualización de información en la base de datos.

`doctores.py`: Aquí se implementan las rutas y la lógica necesarias para manejar las operaciones relacionadas con los doctores, como obtener la lista de doctores, añadir nuevos registros, actualizar la información de un doctor o eliminarla.

`pacientes.py`: Similar a `doctores.py`, este módulo se enfoca en la gestión de pacientes, permitiendo operaciones CRUD (crear, leer, actualizar, eliminar) para los datos de los pacientes.

