

REQUERIMIENTOS DE SOFTWARE

PROYECTO
INPETS

PREPARADO POR
201235853 - ALEJANDRO JOSÉ SCHMIDT RAMÍREZ
201123291 - ARLEM GABRIEL BRENES AGUILAR
2016009280 - JEAN ANTHONY VEGA DÍAZ

INSTITUTO TECNOLÓGICO DE COSTA RICA
INTRODUCCIÓN AL DESARROLLO WEB

23/12/2019

Tabla de Contenido

1	Introducción	3
1.1	Propósito	3
1.2	Convenciones del Documento	3
1.3	Alcance General	4
1.4	Referencias	5
2	Descripción del Producto	5
2.1	Perspectiva	5
2.2	Clases de Usuario	5
2.3	Características generales	6
2.4	Entorno de Operación	7
2.5	Limitaciones	8
2.6	Suposiciones y Dependencias	9
3	Características del Sistema	9
3.1	Inicio de sesión	9
3.2	Registro de médico veterinario	10
3.3	Registro de dueños de mascota	11
3.4	Cerrar sesión	11
3.5	Recuperar Contraseña	12
3.6	Registro de mascota	13
3.7	Consultar expediente de una mascota	14
3.8	Solicitud de acceso al expediente de una mascota	14
3.9	Confirmación de acceso al expediente de una mascota	15
3.10	Revocación de acceso al expediente de una mascota	16
3.11	Agregar entrada al expediente de una mascota	17
3.12	Filtrar lista de mascotas	18
3.13	Filtrar expediente de una mascota	18
4	Interfaces Externas	19
4.1	Interfaces de Usuario	19
4.2	Interfaces de Hardware	22
4.3	Interfaces de Software	22
4.4	Interfaces de Comunicación	23
4.5	Estándar de las funciones	23
5	Requerimientos no Funcionales	24
5.1	Rendimiento	24
5.2	Seguridad	24
5.3	Otros Atributos de Calidad	25

A Glosario	25
B Modelos de Análisis	26
B.1 Diagrama de la arquitectura	26
C Diagrama de Hardware	26
D Imágenes del Prototipo	27

1 Introducción

1.1 Propósito

El presente documento especifica los requerimientos para la plataforma web **inPETS**, desarrollada durante el curso de Introducción al Desarrollo Web del Tecnológico de Costa Rica durante el periodo de Verano 2019/2020. **inPETS** es una plataforma que mantiene un registro médico de **mascotas**, de manera que tanto los **dueños** como los **médicos veterinarios** de las mascotas tengan acceso a toda la información centralizada; independientemente del centro veterinario donde asista la mascota.

En esta sección se planteará por encima la solución propuesta y se describirá la visión general para la plataforma. Se discuten temas como el alcance a nivel general, convenciones utilizadas a través de este documento y las referencias necesarias para la creación del mismo.

En la sección de Descripción del Producto se detalla más a fondo el contexto y origen del producto que se entregará como parte de la solución. Se incluyen secciones como la perspectiva de desarrollo, características puntuales de la aplicación, las diferentes clases o roles de usuario meta para la aplicación, el entorno donde operará la plataforma y finalmente las limitaciones, suposiciones y dependencias sobre las que depende el proyecto.

La sección de Características del Sistema simplemente lista todas las características detalladas que se implementarán. Se utilizarán casos de uso para representar cada una.

La sección de Interfaces Externas lista todo con lo que se va a comunicar el software de la plataforma. Esto incluye los requerimientos de interfaz de usuario, los requerimientos del hardware sobre el cual correrá dicho software, las interfaces lógicas necesarias para comunicarse con otros servicios y la manera en que se comunicarán.

Finalmente en la sección de Requerimientos no Funcionales se especifican lo que se espera del rendimiento, seguridad y otros atributos de calidad de software que deben ser asegurados al terminar el proyecto.

1.2 Convenciones del Documento

A través de este documento se utilizarán varias palabras para referirse al mismo concepto. Son las siguientes:

- **Médico, veterinario:** Los profesionales encargados de ingresar registros a los expedientes de mascotas.
- **Dueño:** Se refiere a los dueños de las mascotas que se registren.

- **Usuario:** Un usuario cualquiera, independientemente de ser médico veterinario o dueño de mascota(s).
- **Historial, expediente, registro:** Son el conjunto de entradas que hacen los veterinarios detallando la información de salud de cada mascota. En el caso de registro en particular a veces se refiere a solo una de esas entradas.
- **Plataforma, sistema, aplicación, proyecto:** Cuando se refiere a "este" proyecto, sistema, aplicación o plataforma se está hablando de inPETS.

Se le sugiere al lector revisar la sección de Glosario en el apéndice en caso de tener alguna confusión con alguno de los conceptos presentados. Además podrá notar que los nombres de las secciones mencionadas hasta ahora están subrayados. Si está leyendo el documento digital, puede hacer click en cualquiera de esos nombres para acceder directamente a la sección correspondiente. El índice, a pesar de no estar subrayado, también cuenta con esa funcionalidad. En caso de que este documento enlace a algún recurso en línea el enlace además tendrá color azul, así (ese enlace abre www.google.com como ejemplo).

1.3 Alcance General

En general la plataforma se encarga de mantener un expediente médico de mascotas. Los dueños de las mascotas crean perfiles para sí mismos, y después agregan cualquier cantidad de mascotas desde ese perfil. Una vez agregada una mascota se les asocia con un identificador único que es utilizado para compartir con los veterinarios para otorgar acceso al expediente de la mascota. El dueño puede revocar permisos de los veterinarios, pero **no puede alterar directamente los registros en el expediente**.

Los veterinarios tienen perfiles separados de los dueños, donde tienen acceso al expediente de todas las mascotas que están tratando y donde pueden agregar entradas nuevas. Hay tres tipos principales de entradas: **vacunas, diagnósticos y tratamientos**. La información necesaria en cada tipo de entrada difieren un poco de cada una, y será explicada es una sección posterior.

Un expediente por completo no necesariamente está atado a un solo veterinario, de lo contrario, pueden haber varios veterinarios haciendo entradas hasta en paralelo, siempre y cuando tengan acceso. Por otra parte, cada registro de ese expediente debe ser hecho por un solo veterinario y **no puede ser borrado una vez que es registrado**. Tanto los dueños como

los médicos autorizados pueden consultar cualquier entrada del expediente de las mascotas en cualquier momento.

Cabe recalcar que esta es una visión muy general del alcance, el cual será explicado más detalladamente en la sección de Descripción del Producto bajo la subsección de Características generales.

1.4 Referencias

La plantilla usada para darle estructura a este documento es **Software Requirements Specification** elaborada por Karl Wiegers de Process Impact y puede ser encontrado [aquí](#).

Previo a este documento se escribió una propuesta de proyecto que puede ser encontrada [aquí](#).

2 Descripción del Producto

2.1 Perspectiva

El mantenimiento de información de salud de mascotas es un proceso tedioso e incómodo. En la experiencia directa de los escritores de este documento, a menudo hay que mantener expedientes físicos que se pueden dañar, manchar o alterar sin mucho esfuerzo requerido. Es común también olvidarlo en camino al veterinario; o peor aún, perderlo por completo. Si se suma a esto las mascotas que van a múltiples veterinarias con expedientes separados, mantener control de todos se puede volver una tarea muy agotadora. Además, si la mascota estuvo en control para vacunación en alguna veterinaria y se cambia a otra es probable que esa información sea valiosa para el nuevo médico veterinario sin tener que preguntarle al cliente (que probablemente no sabe a nivel técnico) o pedirle una copia del expediente viejo. Si el expediente de la mascota está disponible en línea para ser accedido desde cualquier lugar con conexión a internet, basta con pedir acceso al dueño de la mascota y el veterinario puede planear como tratarla en base a su historial médico de inmediato. De ahí nace la inquietud por desarrollar inPETS y suplir este servicio; moviendo la información de las manos de cada centro veterinario a un único repositorio de expedientes disponible en línea y utilizable por cualquiera.

2.2 Clases de Usuario

Existen únicamente dos clases de usuario o actores relevantes para esta aplicación, que son:

- **Dueño de Mascota(s):** se refiere a todo usuario que registra su mascota en la plataforma. Todos estos usuarios tienen una cuenta o perfil personal donde pueden asociar varias mascotas, cada una con un perfil y expediente por aparte. Los usuarios pueden consultar la información de expedientes, pero no pueden agregar nuevos registros o modificar/eliminar los existentes.
- **Médico Veterinario:** se refiere a todo usuario encargado de agregar registros en los expedientes de las mascotas. Los dueños de las mascotas deben otorgar permiso a estos usuarios para acceder al expediente de cada mascota (o revocar los permisos de ser necesario). Los médicos veterinarios tampoco pueden modificar o eliminar registros previos, porque es de suma importancia que los registros no sean editados *a posteriori* maliciosamente. Se puede encontrar información detallada de esto en la sección de Requerimientos no Funcionales bajo la subsección de Seguridad.

2.3 Características generales

A continuación se listarán todas las características generales del sistema. Nótese que esta sección no corresponde directamente a lo que se va a implementar, sino un compendio de lo que se debe esperar del producto a alto nivel de abstracción. Posteriormente en este documento se encuentran las características puntuales del sistema en forma de casos de uso.

- Registro de usuarios, tanto médicos veterinarios como dueños de mascotas.
- Registro de mascotas (disponible solo para dueños).
- Consulta al expediente médico de mascotas (disponible para dueños y los médicos que autorizan).
- Sistema de autorización para acceso a expedientes de mascotas (los médicos piden acceso a los dueños). Esto implica los siguientes subrequerimientos relacionados con autorización:
 - Los médicos deben poder solicitar acceso al expediente de una mascota.
 - Los dueños deben poder garantizar acceso al expediente de la mascota a uno o varios médicos a la vez.
 - Los dueños deben poder revocar el acceso al expediente en cualquier momento.

- Los médicos veterinarios deben poder agregar registros al expediente de las mascotas a las cuales tienen permiso.

2.4 Entorno de Operación

Se planea mantener una arquitectura cliente - servidor web - servidor API. Las ventajas de esto comparado a tener un único servidor son varias:

- Desacopla la implementación del back-end mediante un API que sirve al servidor web (que solo se encarga de servir a los clientes).
- Dado a que se debe definir un API previo a la construcción de la plataforma, tanto el desarrollo del back-end como el front-end pueden ocurrir en paralelo sin problemas.
- Si se desea crear otras versiones de la aplicación basta con hacer pedidos al servidor API para acceder a todo el back-end.

Bajo este esquema entonces la arquitectura se puede visualizar como una pila, empezando desde el usuario hasta la base de datos. Se planean usar las siguientes tecnologías en cada capa de la pila:

Capa	Tecnología	Justificación
Browser	Chrome / Firefox / Edge / Safari	Estos browsers serán respaldados. No se puede asegurar funcionalidad en cualquier otro (en especial Internet Explorer).
Cliente	Bootstrap Javascript	y Es una aplicación pequeña, y por ende el equipo de trabajo piensa que no es necesario utilizar algún framework como Angular, React o Vue.
Servidor web	Node.js Express.js	y Esta es una aplicación de Node muy básica, encargada únicamente de servir las páginas web cliente necesarias. Se seleccionó Node con Express porque son rápidas de desarrollar y mantener.

Servidor API	Node.js y Express.js	Esta parte aloja la interfaz pública del API y maneja todos los pedidos. Tiene los drivers necesarios para comunicarse con las plataformas encargadas de mantener la base de datos y el almacenamiento de imágenes.
Almacenamiento de imágenes	Imgur	Imgur tiene un driver para almacenar imágenes anónimamente desde Node.js que es fácil de usar y comprender.
Base de datos	MongoDB Atlas	Mongo ofrece un driver muy cómodo para Node.js y lleno de funcionalidad para comunicarse con Atlas, la base de datos mantenida por ellos mismos. Ahí se alojará la base de datos.

Se utilizará [Heroku](#) para alojar tanto el servidor web como el del API; y proveer las interfaces públicas necesarias para que puedan ser accedidos mediante internet. Lo descrito anteriormente puede ser visualizado en el [diagrama de arquitectura](#) en el anexo de [Modelos de Análisis](#).

2.5 Limitaciones

Se lograron identificar las siguientes limitaciones al momento de escribir este documento:

- **Tiempo:** El equipo de trabajo cuenta con un total de un mes para desarrollar la aplicación. Esto no debería ser un problema dado al alcance limitado de la misma.
- **Costo:** Como es un proyecto académico no se cuenta con un presupuesto para desarrollar, mantener o pagar servicios para la aplicación. Afortunadamente, todas las tecnologías descritas anteriormente son gratuitas y deberían ser suficientes para alojar y operar la aplicación web.
- **Escalabilidad:** Por los dos puntos anteriores, no se puede garantizar que dado el hardware adquirido gratuitamente para alojar la aplicación pueda escalar adecuadamente si se desea. El software no debería ser un problema para lidiar con escalabilidad.

2.6 Suposiciones y Dependencias

Se supone para el desarrollo del proyecto y el mantenimiento del mismo lo siguiente:

- Las tecnologías y sub-plataformas utilizadas (Heroku, Imgur, MongoDB Atlas) continúan su operación y sus servicios gratuitos se mantienen en pie.
- Se supone que el alcance descrito en la [propuesta de anteproyecto](#) delimita correctamente el alcance del proyecto, ya que basado en ella se han definido las características de esta plataforma.

3 Características del Sistema

Como las características del sistema fueron discutidas a nivel general en la [sección previa](#), esta sección es simplemente una lista de casos de uso, uno por cada requerimiento puntual necesario para cubrir las características generales.

3.1 Inicio de sesión

Identificador	CU-1
Actor Primario	Cualquier usuario
Actor Secundario(s)	Front-end, servidor API
Descripción	Inicio de sesión tanto para médicos como dueños de mascotas.
Frecuencia	Una vez por sesión.
Activador	El usuario entró a la página principal y se le muestra el cuadro de iniciar sesión.
Precondiciones	Ninguna.
Postcondiciones	Ninguna.
Flujo Normal	<ol style="list-style-type: none">1. El usuario digita su correo electrónico, contraseña y tipo de usuario (dueño o veterinario).2. El usuario hace click en aceptar.3. El front-end genera una petición de inicio de sesión al servidor API.4. El servidor API revisa si las credenciales existen en la base de datos.5. El servidor API crea una sesión nueva, y envía su identificador al front-end.

	<p>6. El front-end guarda la sesión en una cookie para iniciar sesión automáticamente después.</p> <p>7. Se abre la página del <u>dashboard</u> de usuario correspondiente (dueño o veterinario).</p>
Excepciones	<p>1.E1. Si existe una cookie con un identificador de sesión entonces se intenta iniciar sesión automáticamente y se salta al paso 7 del flujo normal.</p> <p>4.E1. Si las credenciales no existen en la base de datos se devuelve un error.</p>

3.2 Registro de médico veterinario

Identificador	CU-2
Actor Primario	Médico veterinario
Secundario(s)	Front-end, servidor API
Descripción	El usuario digita la información requerida para crear una cuenta y; si no hay problemas, el servidor API los guarda como una cuenta en la base de datos.
Frecuencia	Una única vez por cuenta.
Activador	El usuario accedió a la página de creación de cuentas de veterinarios.
Precondiciones	Ninguna.
Postcondiciones	Una cuenta de veterinario nueva es creada en la base de datos.
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario introduce: su nombre, apellidos, identificación, correo electrónico, código de médico veterinario, contraseña para su cuenta y una confirmación para su contraseña. 2. El usuario hace click en registrarse. 3. El front-end genera una petición de creación de cuenta al servidor API. 4. El servidor API crea la cuenta en la base de datos y regresa una confirmación de creación de cuenta al front-end. 5. El front-end notifica al usuario que su cuenta fue creada.
Excepciones	<p>3.E1. Si el usuario no escribe una dirección de correo válida se muestra un mensaje de error.</p> <p>3.E2. Si el usuario no digitó lo mismo en los campos de contraseña y confirmación de contraseña se muestra un error.</p>

	4.E1. Si el e-mail seleccionado ya existe en la base de datos para veterinarios se devuelve un error.
--	---

3.3 Registro de dueños de mascota

Identificador	CU-3
Actor Primario	Dueño de mascota
Actor Secundario(s)	Front-end, servidor API
Descripción	El usuario digita la información requerida para crear una cuenta y; si no hay problemas, el servidor API los guarda como una cuenta en la base de datos.
Frecuencia	Una única vez por cuenta.
Activador	El usuario accedió a la página de creación de cuentas de dueños de mascota.
Precondiciones	Ninguna.
Postcondiciones	Una cuenta de dueño nueva es creada en la base de datos.
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario introduce: su nombre, apellidos, correo electrónico, contraseña para su cuenta y una confirmación para su contraseña. 2. El usuario hace click en registrarse. 3. El front-end genera una petición de creación de cuenta al servidor API. 4. El servidor API crea la cuenta en la base de datos y regresa una confirmación de creación de cuenta al front-end. 5. El front-end notifica al usuario que su cuenta fue creada.
Excepciones	<p>3.E1. Si el usuario no escribe una dirección de correo válida se muestra un mensaje de error.</p> <p>3.E2. Si el usuario no digitó lo mismo en los campos de contraseña y confirmación de contraseña se muestra un error.</p> <p>4.E1. Si el e-mail seleccionado ya existe en la base de datos para dueños se devuelve un error.</p>

3.4 Cerrar sesión

Identificador	CU-4
Actor Primario	Cualquier usuario
Actor Secundario(s)	Front-end, servidor API

Descripción	El usuario cierra sesión y se regresa a la página principal.
Frecuencia	Una vez por sesión.
Activador	El usuario selecciona la opción "Cerrar Sesión" en el menú de navegación.
Precondiciones	Debe haber una sesión abierta.
Postcondiciones	La sesión debe cerrarse a nivel de base de datos.
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario selecciona la opción de cerrar sesión desde el menú de navegación. 2. El front-end comunica al servidor API que se debe cerrar la sesión actual y le pasa el ID de sesión. 3. El servidor API cierra la sesión en la base de datos y regresa una confirmación al front-end. 4. El front-end cambia a la página principal.
Excepciones	Ninguna.

3.5 Recuperar Contraseña

Identificador	CU-5
Actor Primario	Cualquier usuario
Actor Secundario(s)	Front-end, servidor API
Descripción	El usuario cambia su contraseña a una aleatoria y se envía a su correo electrónico.
Frecuencia	No muy seguido, cada vez que el usuario olvide su contraseña.
Activador	El usuario entra a la página de "¿Olvidó su contraseña?".
Precondiciones	No debe haber una sesión iniciada.
Postcondiciones	Se cambia la contraseña del usuario por una aleatoria generada por el servidor API en la base de datos.
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario llena el campo de correo electrónico y selecciona su tipo de usuario (médico veterinario o dueño de mascota). 2. El front-end envía una petición de cambio de correo electrónico al servidor API. 3. El servidor API cambia la contraseña en la cuenta del usuario en la base de datos. 4. El servidor API envía un correo electrónico con la contraseña nueva. 5. El servidor API envía un mensaje de confirmación al front-end de correo enviado. 6. El front-end despliega el mensaje al usuario.

Excepciones	<p>2.E1. Si el usuario introduce un correo electrónico inválido el front-end despliega un mensaje de error.</p> <p>3.E1. Si el correo del usuario no es encontrado en la base de datos entonces se despliega un mensaje de error.</p> <p>4.E1. Si se cambió la contraseña pero no se pudo enviar el correo electrónico porque hubo un error al enviarlo entonces se notifica al usuario como una advertencia. Se le insta a resolver el problema con el correo e intentar de nuevo.</p>
--------------------	--

3.6 Registro de mascota

Identificador	CU-6
Actor Primario	Dueño de mascota
Actor Secundario(s)	Front-end, servidor API
Descripción	El dueño de mascota agrega una nueva mascota al sistema.
Frecuencia	No muy seguido, depende de la cantidad de mascotas del dueño.
Activador	El dueño selecciona la opción de agregar mascota en su dashboard.
Precondiciones	Ninguna.
Postcondiciones	Una mascota debe quedar agregada en la base de datos.
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario escribe el nombre de la mascota y una descripción. Selecciona su fecha de nacimiento, sexo, especie y raza. Finalmente le adjunta una fotografía de la mascota. 2. El usuario hace click en agregar. 3. El front-end convierte la imagen en su representación de Base64 para enviarla al servidor. 4. El front-end envía una petición de agregar mascota al servidor API. 5. El servidor API sube la imagen a Imgur y recibe un link para la imagen. 6. El servidor API agrega la mascota a la base de datos (se guarda el link de Imgur para la imagen). 7. El servidor API envía un mensaje de confirmación al front-end de mascota agregada. Este mensaje incluye la información de la mascota, incluyendo el identificador único de la mascota.

	8. El front-end agrega la mascota a la lista de mascotas del dueño.
Excepciones	5.E1. Si hay problemas subiendo la imagen a Imgur entonces se procede de la misma forma pero sin la imagen.

3.7 Consultar expediente de una mascota

Identificador	CU-7
Actor Primario Secundario(s)	Cualquier usuario con acceso al expediente de la mascota Front-end, servidor API
Descripción	El usuario ve toda la información de la mascota en un dashboard junto con su expediente médico.
Frecuencia	Muy seguido. Cada vez que el usuario selecciona una mascota para examinar su información.
Activador	El usuario selecciona la opción de ver expediente para alguna mascota en particular desde su dashboard.
Precondiciones	Debe existir al menos una mascota para ver en la lista de mascotas del dashboard del usuario (si es un dueño sería de sus mascotas, y si es un médico veterinario de las que tiene permiso para acceder).
Postcondiciones	Ninguna.
Flujo Normal	1. El front-end cambia a una página vacía sin información de mascota con el identificador de la mascota como parámetro del URL. 2. El front-end utiliza el identificador de la mascota para pedir toda la información de la mascota (información general y el expediente completo). 3. El servidor API busca la información en la base de datos con el identificador de la mascota y la regresa al front-end. 4. El front-end utiliza la información para llenar la página vacía de la mascota.
Excepciones	3.E1. Si la información de la mascota no es encontrada, entonces se muestra un mensaje de error.

3.8 Solicitud de acceso al expediente de una mascota

Identificador	CU-8
Actor Primario Secundario(s)	Médico veterinario Front-end, servidor API

Descripción	El médico veterinario actor solicita acceso al expediente médico de una mascota mediante su identificador único.
Frecuencia	Seguido. Cada vez que un médico requiera conseguir acceso a un expediente.
Activador	El médico selecciona la opción de solicitar acceso de mascota desde su dashboard.
Precondiciones	Ninguna.
Postcondiciones	Se debe guardar la solicitud en la base de datos.
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario escribe el identificador único de la mascota. 2. El usuario hace click sobre solicitar expediente. 3. El front-end envía una solicitud con el identificador del médico y la mascota al servidor API. 4. El servidor API agrega la solicitud al buzón de notificaciones del dueño. 5. El front-end muestra un mensaje de solicitud enviada.
Excepciones	<p>3.E1. Si la mascota no es encontrada en la base de datos se muestra un mensaje de error.</p> <p>3.E2. Si el médico veterinario no es encontrado en la base de datos entonces se muestra un mensaje de error.</p>

3.9 Confirmación de acceso al expediente de una mascota

Identificador	CU-9
Actor Primario	Dueño de mascota
Secundario(s)	Front-end, servidor API
Descripción	El dueño de la mascota acepta o rechaza la solicitud de acceso al expediente de la mascota.
Frecuencia	Depende de la cantidad de mascotas registradas y médicos a los que se les permitirá acceso, una vez por cada par mascota - médico.
Activador	El dueño de mascota selecciona la opción de notificaciones desde su dashboard.
Precondiciones	Debe existir una solicitud de acceso a expediente disponible.
Postcondiciones	Se debe dar acceso al médico desde la base de datos si el dueño la acepta. Si el dueño no la acepta entonces se elimina la notificación y la solicitud de la base de datos.
Flujo Normal	<ol style="list-style-type: none"> 1. El front-end muestra al usuario una lista de solicitudes.

	<p>2. El usuario selecciona aceptar de alguna de las solicitudes de la lista.</p> <p>3. El front-end notifica al servidor API que la solicitud fue aceptada.</p> <p>4. El servidor API elimina la solicitud de la base de datos y agrega un permiso con la información del médico veterinario correspondiente y la mascota.</p> <p>5. El servidor API responde que se agregó la solicitud exitosamente.</p> <p>6. El front-end notifica al usuario que se agregó el permiso de acceso al expediente y elimina la notificación de la lista.</p>
Flujo Alterno	<p>2.A. El usuario selecciona rechazar de alguna de las solicitudes de la lista.</p> <p>3.A. El front-end notifica al servidor API que la solicitud fue rechazada.</p> <p>4.A. El servidor API elimina la solicitud de la base de datos pero no agrega un permiso.</p> <p>5.A. El servidor API responde que se rechazó la solicitud exitosamente.</p> <p>6.A. El front-end notifica al usuario que se rechazó el permiso de acceso al expediente y elimina la notificación de la lista.</p>
Excepciones	Ninguna.

3.10 Revocación de acceso al expediente de una mascota

Identificador	CU-10
Actor Primario	Dueño de mascota
Actor Secundario(s)	Front-end, servidor API
Descripción	El dueño de la mascota decide revocar los permisos de acceso al expediente para un veterinario.
Frecuencia	No muy seguido, solo si el dueño de la mascota lo considera necesario por cuestiones de seguridad.
Activador	El dueño de mascota selecciona la opción de mostrar permisos desde su dashboard.
Precondiciones	El dueño de mascota debe haber dado algún permiso a un veterinario.
Postcondiciones	El permiso debe ser eliminado a nivel de base de datos.
Flujo Normal	<p>1. El front-end muestra al usuario una lista de permisos.</p>

	<ol style="list-style-type: none"> 2. El usuario selecciona revocar en algún permiso de la lista. 3. El front-end envía un identificador del permiso a revocar al servidor API. 4. El servidor API elimina el permiso. 5. El servidor API responde que eliminó el permiso al front-end. 6. El front-end le notifica al usuario que se eliminó el permiso.
Excepciones	<p>4.E1. Si el permiso no es encontrado en la base de datos, entonces se responde con un mensaje de error.</p>

3.11 Agregar entrada al expediente de una mascota

Identificador	CU-11
Actor Primario	Médico veterinario
Secundario(s)	Front-end, servidor API
Descripción	El médico veterinario agrega un nuevo registro en el expediente médico de la mascota.
Frecuencia	Muy seguido.
Activador	El médico veterinario selecciona la opción de agregar un nuevo registro desde el dashboard de la mascota.
Precondiciones	Ninguna.
Postcondiciones	Se debe agregar un registro nuevo en el expediente de la mascota en la base de datos.
Flujo Normal	<ol style="list-style-type: none"> 1. El médico veterinario selecciona algún tipo de entrada y escribe los datos necesarios dependiendo del tipo de registro. 2. El médico hace click en agregar. 3. El front-end genera una petición de creación de registro médico al servidor API. 4. El servidor API recibe todos los datos del front-end y guarda el registro en la base de datos. 5. El servidor API confirma la creación del registro al front-end y devuelve el registro completo. 6. El front-end agrega el registro a la lista en el dashboard de la mascota.

Flujo Alterno	1.A. Si el médico selecciona una entrada de tipo diagnóstico entonces debe suministrar un título para el diagnóstico, una descripción, el peso de la mascota y su temperatura.
Flujo Alterno	1.B. Si el médico selecciona una entrada de tipo vacuna o tratamiento debe indicar el nombre de lo que se le está suministrando a la mascota, la cantidad suministrada junto con su medida y finalmente una descripción.
Excepciones	Ninguna.

3.12 Filtrar lista de mascotas

Identificador	CU-12
Actor Primario	Cualquier usuario
Secundario(s)	Front-end
Descripción	El usuario filtra la lista de mascotas de su dashboard basado en cualquier criterio textual.
Frecuencia	Muy seguido en usuarios con muchas mascotas (especialmente veterinarios).
Activador	El usuario escribe en la barra de búsqueda cualquier texto.
Precondiciones	Ninguna.
Postcondiciones	Ninguna.
Flujo Normal	<ol style="list-style-type: none"> 1. El front-end captura el evento de la barra de búsqueda de texto cambiado. 2. Basado en ese texto, el front-end revisa la lista de mascotas y muestra únicamente a todas las que incluyen el texto en cualquiera de estos campos: identificador, nombre, sexo, especie y raza.
Excepciones	ninguna.

3.13 Filtrar expediente de una mascota

Identificador	CU-13
Actor Primario	Cualquier usuario
Secundario(s)	Front-end
Descripción	El usuario filtra los registros del expediente en el dashboard de la mascota basado en cualquier criterio textual.

Frecuencia	Depende de la cantidad de registros en el expediente, pero en general no debería ser muy seguido.
Activador	El usuario escribe en la barra de búsqueda, selecciona un tipo de registro de una lista con todos los tipos de registro o selecciona un médico de una lista que contiene a todos los médicos presentes en los registros.
Precondiciones	Ninguna.
Postcondiciones	Ninguna.
Flujo Normal	<ol style="list-style-type: none"> 1. El front-end captura el evento de la barra de búsqueda de texto cambiado, o si se selecciona un tipo de entrada o el nombre de un doctor. 2. El front-end genera un string de búsqueda basado en lo seleccionado (barra de búsqueda, tipo de entrada y doctor). 3. El front-end revisa los registros de la mascota y muestra únicamente a todos los que incluyen el texto en cualquiera de estos campos: tipo de registro, nombre, cantidad suministrada y medida de tratamiento o vacuna, título de diagnóstico, descripción, nombre de doctor y fecha.
Excepciones	Ninguna.

4 Interfaces Externas

En esta sección se discutirán las diferentes interfaces necesarias para el funcionamiento de la plataforma.

4.1 Interfaces de Usuario

- **Página de Inicio:** La página principal de inPETS, contará con una barra de navegación en la parte superior que permitirá al usuario navegar por las distintas opciones de la aplicación, además mostrará en la parte central el formulario de inicio de sesión para que el usuario pueda ingresar sus credenciales y acceder como usuario dueño o usuario veterinario. Para un mejor contexto de como sería visualmente se puede acceder [aquí](#), donde se muestra una imagen en la sección de los apéndices sobre como lucirá la página de inicio.
- **Página de Registro Veterinarios:** La página para registro de veterinarios mantendrá la barra de navegación en la parte superior, al costado izquierdo mostrará una imagen representativa y al costado

derecho el formulario exclusivo para el registro de usuarios veterinarios. Para un mejor contexto de como sería visualmente se puede acceder [aquí](#), donde se muestra una imagen en la sección de los apéndices sobre como lucirá la página de registro de veterinarios.

- **Página de Registro Dueños:** La página para registro de dueños mantendrá la barra de navegación en la parte superior, al costado izquierdo mostrará una imagen representativa y al costado derecho el formulario exclusivo para el registro de usuarios dueños de mascotas. Para un mejor contexto de como sería visualmente se puede acceder [aquí](#), donde se muestra una imagen en la sección de los apéndices sobre como lucirá la página de registro de dueños de mascotas.
- **Página de Dashboard Veterinario:** Para efectos de la página del tablero(dashboard) del veterinario, esta será visible una vez que un usuario veterinario registrado, inicie su sesión en la página de inicio de la aplicación. Esta página a diferencia de las de registro e inicio de sesión, cambiará su barra de navegación mostrando únicamente la opción de cerrar sesión. Ahora bien, en el encabezado se mostrarán los datos del veterinario que haya ingresado, así como la opción de modificar sus datos accediendo al ícono del lápiz o bien acceder a las notificaciones sobre respuestas a solicitudes a expedientes de mascotas. Por otra parte, en el cuerpo de la página se encontrará desplegada la lista de aquellas mascotas sobre las cuales el veterinario tiene permiso de acceso, una barra de búsqueda sobre dicho listado, la opción para solicitar acceso a una mascota en particular y por último abrir el expediente de una mascota mediante el ícono que se encuentra totalmente al lado derecho de los datos de la mascota. Para un mejor contexto sobre lo explicado se puede acceder [aquí](#), donde se muestra una imagen en la sección de los apéndices sobre como lucirá la página del tablero (dashboard) de veterinarios.
- **Página de Expediente desde Dashboard Veterinario:** Una vez ingresado a la opción de expediente que se encuentra en el tablero(dashboard) del veterinario, se mostrará la página del expediente de una mascota en particular donde el usuario veterinario podrá observar los datos de la mascota en el encabezado mientras que en el cuerpo de la página podrá encontrar todos aquellos tratamientos, vacunas y diagnósticos aplicados a la mascota, así como realizar búsquedas sobre estos datos por su tipo o por el médico veterinario que llevó a cabo el registro. Además se encuentra la opción de agregar, donde el médico podrá ingresar las entradas pertinentes en el expedi-

ente sobre alguna vacuna, tratamiento o diagnóstico realizado a dicha mascota. Para un mejor contexto sobre lo explicado se puede acceder [aquí](#), donde se muestra una imagen en la sección de los apéndices sobre como lucirá la página del expediente desde el tablero(dashboard) del veterinario.

- **Página de Dashboard Dueño:** Para efectos de la página del tablero(dashboard) del dueño, esta será visible una vez que un usuario dueño de mascota registrado, inicie su sesión en la página de inicio de la aplicación. Esta página a diferencia de las de registro e inicio de sesión, cambiará su barra de navegación mostrando únicamente la opción de cerrar sesión. Ahora bien, en el encabezado se mostrarán sus datos, así como la opción de administrar solicitudes de acceso mediante el ícono de la llave, modificación de sus datos mediante el ícono del lápiz y podrá observar si cuenta con notificaciones observando el ícono de la campana. Por otra parte, en el cuerpo de la página se encontrará desplegada la lista de aquellas mascotas que dicho usuario haya registrado, podrá realizar búsquedas sobre sus mascotas y también la opción de agregar una mascota nueva. Finalmente, podrá ingresar al expediente de una de sus mascotas en particular mediante el ícono de computadora que se encuentra totalmente al lado derecho de los datos de la mascota. Para un mejor contexto sobre lo explicado se puede acceder [aquí](#), donde se muestra una imagen en la sección de los apéndices sobre como lucirá la página del tablero (dashboard) de dueños de mascotas.
- **Página de Expediente desde Dashboard Dueño:** Una vez ingresado a la opción de expediente que se encuentra en el tablero(dashboard) del dueño, se mostrará la página del expediente de una mascota en particular donde el usuario dueño de mascota, podrá observar los datos de la mascota en el encabezado, así como modificar los datos de dicha mascota accediendo al ícono del lápiz en la esquina superior derecha, mientras que en el cuerpo de la página podrá encontrar todos aquellos tratamientos, vacunas y diagnósticos aplicados a la mascota, así como realizar búsquedas sobre estos datos por su tipo o por el médico veterinario que llevó a cabo el registro. Para un mejor contexto sobre lo explicado se puede acceder [aquí](#), donde se muestra una imagen en la sección de los apéndices sobre como lucirá la página del expediente desde el tablero(dashboard) del veterinario.

4.2 Interfaces de Hardware

Con respecto a las interfaces de hardware, la aplicación contará con distintos componentes según el diagrama de hardware que se puede observar [aquí](#), los cuales serán explicados a continuación:

- **Aplicación Web:** Esta es la interfaz con la cual el usuario tendrá interacción, en ella se desplegarán aquellas páginas construidas en HTML y CSS, que se encuentran almacenadas en el Servidor Web
- **Servidor Web:** El servidor Web contiene las páginas creadas en HTML, CSS para atender las consultas realizadas por el usuario, este se encuentra basado en Node con Express y únicamente se encarga de servir las páginas web al cliente.
- **Servidor API:** El servidor API, contiene toda la lógica de negocio que permite realizar todas las consultas necesarias para verificación u obtención de datos desde la base de datos en Mongo y en el caso de las imágenes, hacia el repositorio de Imgur
- **Repositorio Imgur:** Se encargará del almacenamiento anónimo de imágenes desde Node.js
- **Base de Datos Mongo:** Base de datos que se encargará de almacenar todos los datos registrados desde la aplicación web para realizar consultas sobre ellos.

4.3 Interfaces de Software

La aplicación, está desarrollado en un ambiente web, este tipo de aplicaciones nos permiten tener una mayor versatilidad al momento de usarlas, ya que no es necesario instalarla en un dispositivo para ejecutarla, basta con tener un navegador web.

Este software, hace uso de sistemas externos como lo es una base datos para almacenar de forma permanente los datos. En nuestro caso la base de datos utilizada será MongoDB, permitiéndonos aprovechar la flexibilidad y las ventajas que las bases de datos no relacionales no ofrecen.

En cuanto lenguajes de desarrollo la mayor parte del proyecto, tanto el front como el back end estará desarrollado en JavaScript, el cual nos ofrece muchas funcionalidades sin requerir a lenguajes mas pesados.

DE igual forma el servicio encargado de recibir y administrar las peticiones HTTP Request esta implementado en JavaScript.

4.4 Interfaces de Comunicación

La aplicación web, hará uso de llamado de funciones a un servidor de back-end, mediante el uso de peticiones HTTP que deben ser definidas en un documento por aparte llamado **Definición del servicio API**.

4.5 Estándar de las funciones

- **Nombre de las funciones:** las funciones encargadas de hacer las peticiones al servidor, iniciaran con el sufijo "request" seguido de un identificador relacionado con el procedimiento (en caso de ser mas de una palabra se, escribe pegado con una mayúscula al inicio de la palabra):

Ejemplo: `requestInsertPet()`

- **Parametros:** las funciones recibirán como único parámetro, un objeto de tipo JSON, el cual sera identificado con el sufijo "data" seguido de un identificador relacionado con los datos a enviar. (en caso de ser mas de una palabra se, escribe pegado con una mayúscula al inicio de la palabra):

Ejemplo: `requestInsertPet(data, namePet)`

- **Formato del JSON:** los atributos del JSON enviado en las funciones serán definidos de la siguiente forma, iniciaran con el sufijo "param" seguido de un identificador relacionado con el parámetro. (en caso de ser mas de una palabra se, escribe pegado con una mayúscula al inicio de la palabra)

Ejemplo: `paramNamePet`

- **Retorno de las funciones:** este tipo de funciones de igual forma, retornan un JSON según sea la definición de la función.
 - **Funciones de tipo 1:** realizan cambios en la base de datos, (no retornan un conjunto de datos). Estas retornan un JSON con 3 parametros:
 - * response: "True or False"
 - * message: "Mensaje de retorno"
 - * info: "Datos técnicos necesarios "
 - **Funciones de tipo 2:** realizan consultas a la base de datos y obtienen un conjunto de datos, retornan un JSON con los arreglos correspondientes al la información solicitada. (En caso de no contar con información para retornar el JSON se obtiene vacío)

5 Requerimientos no Funcionales

5.1 Rendimiento

Como no se espera una cantidad alta de usuarios el rendimiento en general no debería verse muy afectado. Esto depende mucho del hardware donde se instale la aplicación, y como fue mencionado antes eso está fuera del alcance de esta aplicación. Habiendo mencionado esto, debe asegurarse en medida de lo posible que el código escrito y base de datos diseñada sea tan escalable como sea posible. Los tiempos de respuesta para cualquier operación no deberían sobrepasar los diez segundos en cualquier operación.

5.2 Seguridad

La seguridad suministrada por la aplicación es sumamente básica. La aplicación debe contar con los siguientes mecanismos de seguridad para identificar usuarios y sesiones abiertas:

- Toda la información sensible será transmitida a través del cuerpo de documentos [HTML](#) a través de métodos [HTTP](#) con [HSTS](#). Esto implica que la información no puede ser afectada por intermediarios. Heroku se encarga de todo esto por default en sus servidores físicos.
- Las cuentas de usuarios todas tienen contraseñas, las cuales deben ser encriptadas mediante [SHA-256](#) y guardadas en lugar de en texto plano. Se recomienda usar un paquete de Node.js para esto llamado [crypto](#).
- Debe haber un sistema de sesiones, para que hayan operaciones que solo puedan ser realizadas por usuarios reales cuando llegue una petición al servidor API. Cuando un usuario inicia sesión se crea una sesión nueva en el servidor API. El identificador único de esta sesión es comunicado al front-end, que lo guarda en una cookie para iniciar sesión después sin tener que pedir credenciales. Cuando una sesión se cierra, se marca como tal y no se puede usar de nuevo. Para hacer cualquiera de las operaciones de la aplicación (con excepción de las relacionadas a crear cuentas e iniciar sesión) se requiere un identificador de sesión abierta o el servidor API rechazará la petición.
- A nivel interno, el servidor API manejará toda la información sensible (léase strings de conexión a bases de datos, contraseñas, etc) en archivos separados que no son sometidos a control de versiones en

línea (GitHub por ejemplo). Se recomienda usar un paquete de Node llamado [dotenv](#) para manejar variables de entorno.

5.3 Otros Atributos de Calidad

Los siguientes otros atributos de calidad también fueron tomados en cuenta:

- **Mantenibilidad:** La arquitectura del sistema definida en este documento (tener el servidor web separado del API) desacopla la implementación de la lógica de negocio de la interfaz gráfica. Para comunicar el front-end con el back-end, se debe proveer una especificación de métodos HTTP en un documento separado a este. Si se deseara, se puede modificar el servidor API a una arquitectura de microservicios para aliviar la carga en él, pero para el alcance actual esto no es necesario.
- **Portabilidad:** Al ser una plataforma web desarrollada para cualquier dispositivo, este atributo se asegurará probando la aplicación en diferentes tamaños de pantallas para asegurarlo. Las pruebas necesarias para esto se salen del alcance de este documento.
- **Robustez:** Bajo ninguna circunstancia ambos servidores deben tener fallos catastróficos y salir de operación dada la poca carga esperada.

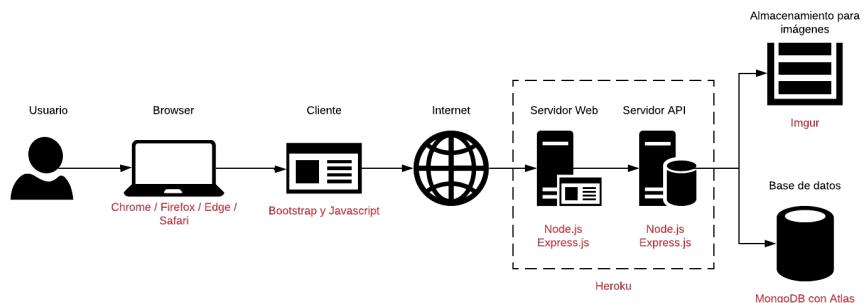
A Glosario

- **API:** Application Programming Interface. Es una interfaz entre diferentes sistemas de información para simplificar la implementación y mantenimiento del software.
- **Back-end:** Es la parte que el usuario final no ve de la aplicación, que maneja lógica de negocio, almacenamiento, etc.
- **Clases de Usuario:** Los roles de usuarios presentes en toda aplicación de software. En inPETS por ejemplo son dos: los dueños de las mascotas y los médicos veterinarios que las atienden.
- **Dashboard:** En el contexto web se refiere a la página principal que es central a toda la funcionalidad de la aplicación. El usuario parte de acá para hacer cualquier operación.
- **Express.js:** Paquete de Node.js para hacer más fácil el manejo de rutas en un servidor HTTP.

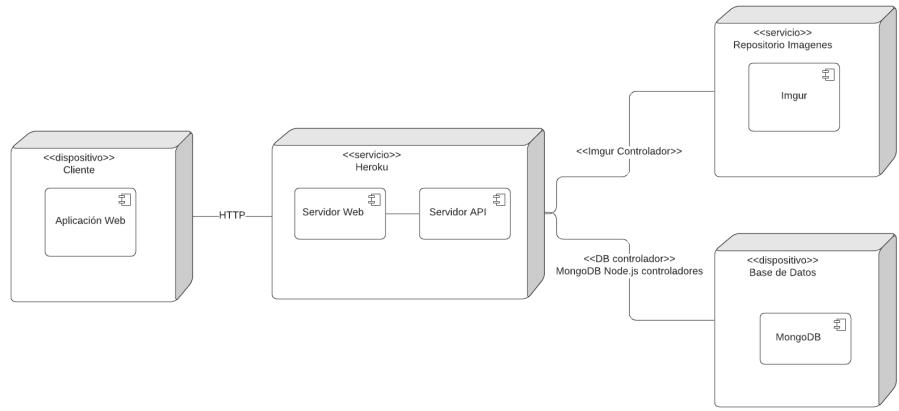
- **Front-end:** Es la parte de la aplicación con la que el usuario interacciona.
- **HTML:** HyperText Markup Language. Es un documento de texto que define la estructura de una página web.
- **HTTP:** HyperText Transfer Protocol. Es el protocolo de red mediante el cual se intercambian documentos HTML.
- **HSTS:** HTTP Strict Transport Security. Es una política de transmisión para HTTP que ayuda a sitios web a mantener la información transmitida segura.
- **Nivel de Abstracción:** El aislamiento de un elemento de su contexto. Entre más alto, más alejado está. Por ejemplo, en este documento la subsección de características bajo la sección de producto tiene un mayor nivel de abstracción que la sección de características.
- **Node.js:** Entorno de tiempo de ejecución de JavaScript capaz de crear servidores HTTP.
- **SHA-256:** Secure Hash Algorithm de 256 bits. Es un algoritmo determinístico para obtener un hash (una cadena semialeatoria de texto) de otra pieza de texto.

B Modelos de Análisis

B.1 Diagrama de la arquitectura



C Diagrama de Hardware



D Imágenes del Prototipo

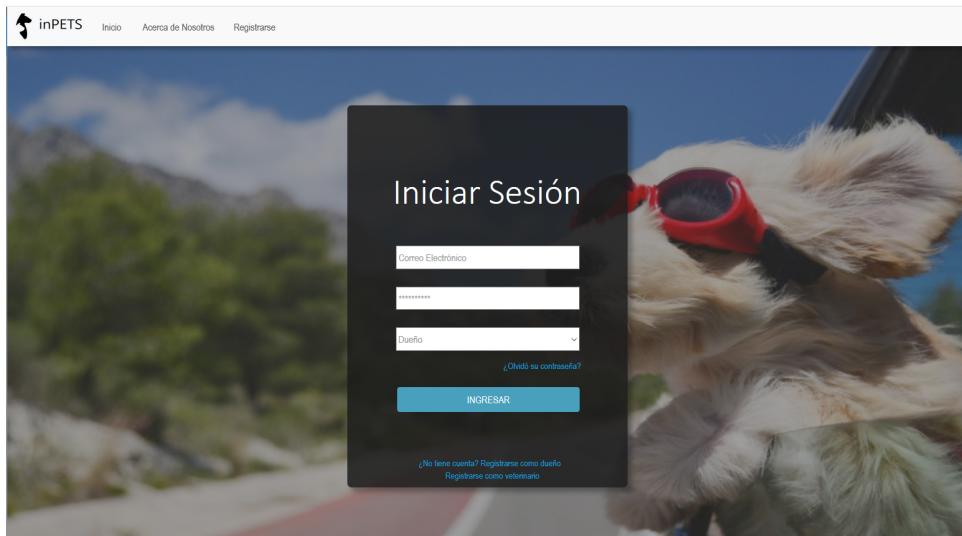
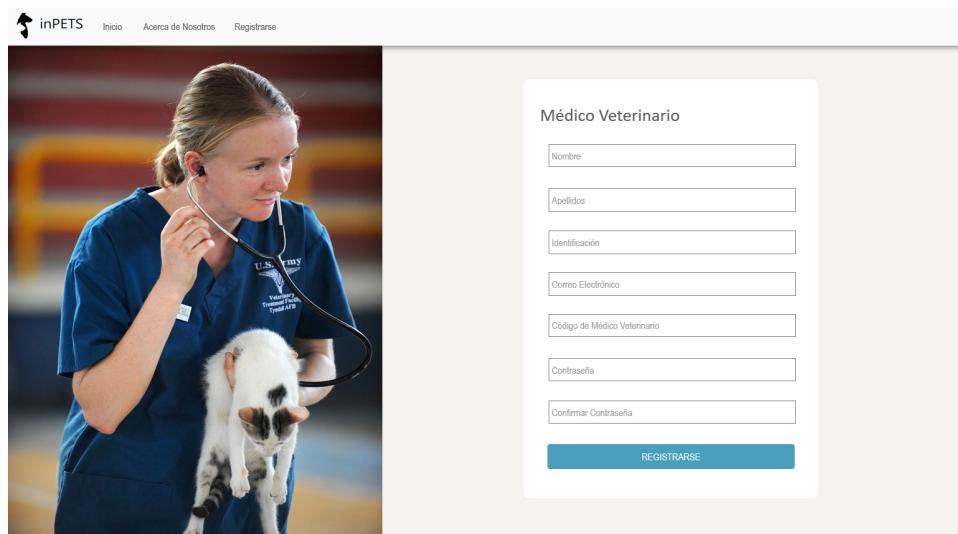


Figure 1: Página de Inicio



Médico Veterinario

Nombre

Apellidos

Identificación

Correo Electrónico

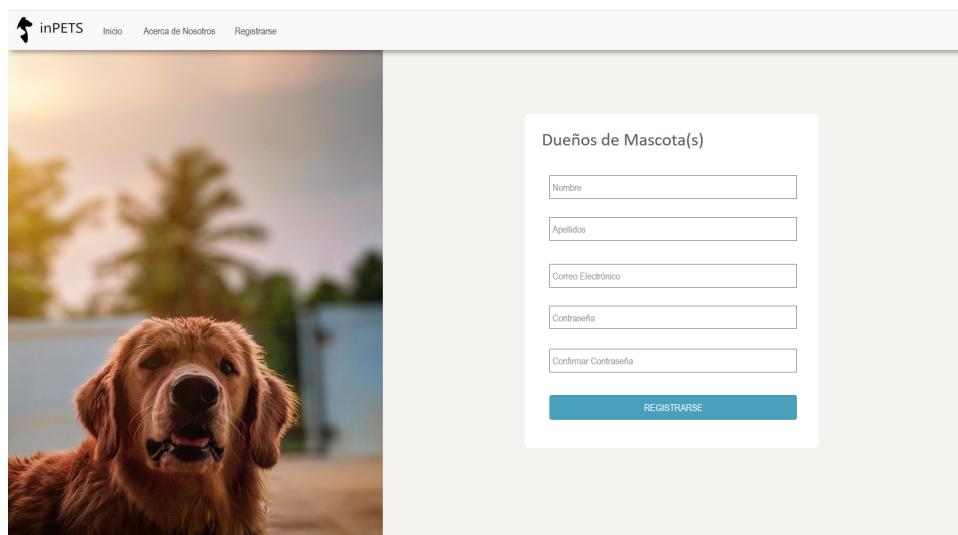
Código de Médico Veterinario

Contraseña

Confirmar Contraseña

REGISTRARSE

Figure 2: Registro Veterinarios



Dueños de Mascota(s)

Nombre

Apellidos

Correo Electrónico

Contraseña

Confirmar Contraseña

REGISTRARSE

Figure 3: Registro Dueños

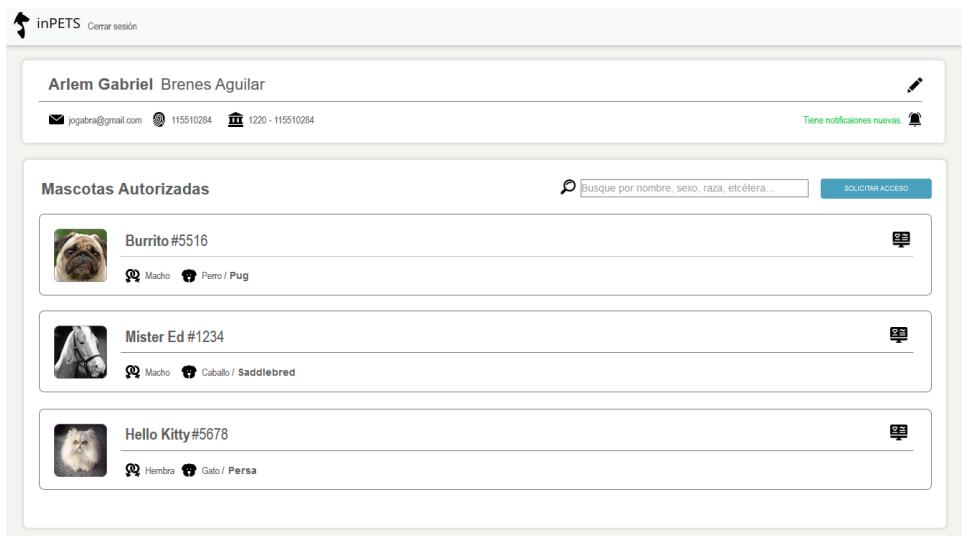


Figure 4: Dashboard Veterinario

The screenshot shows a list of medical records for the dog 'Burrito#5516'. Each record includes a small icon, the name of the treatment or service, the date it was performed, the gender of the animal (Male), the species (Dog / Pug), and the name of the veterinarian who performed it (Kenny Bell). The records are:

- Diazepam 30mg** on 25/09/2011 by Kenny Bell
- Tetravalente 10ml** on 25/09/2011 by Kenny Bell
- Revisión rutinaria de agosto** on 25/09/2011 by Kenny Bell

Below the records, there is a placeholder text: "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean euismod bibendum laoreet. Proin gravida dolor sit amet lacus accumsan et viverra justo commodo. Proin sodales pulvinar tempor. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Nam fermentum, nulla luctus pharetra vulputate, felis tellus mollis orci, sed rhoncus sapien nunc eget."

Figure 5: Expediente de Mascota desde Dashboard del Veterinario

The screenshot shows a list of owned pets for the user 'Alejandro José Schmidt Ramírez'. Each pet entry includes a small profile picture, the pet's name, its gender (Male or Female), species (Dog / Pug, Horse / Saddlebred, Cat / Persian), and a small edit icon.

- Burrito #5516** (Male, Dog / Pug)
- Mister Ed #1234** (Male, Horse / Saddlebred)
- Hello Kitty #5678** (Female, Cat / Persian)

Below the list, there is a placeholder text: "No tiene notificaciones nuevas." (No new notifications.)

Figure 6: Dashboard Dueño de Mascota

The screenshot displays a pet's dashboard within the inPETS application. At the top, there is a header with the inPETS logo, a 'Volver' (Return) link, and a 'Cerrar sesión' (Logout) link. Below the header, the main content area is titled 'Burrito#5516'. It features a thumbnail image of a pug dog, the name 'Burrito#5516', and a date '25/09/2011'. To the right of the date are icons for gender ('Macho') and species ('Perro / Pug'). A detailed status message follows: 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean euismod bibendum laoreet. Proin gravida dolor sit amet lacus accumsan et viverra justo commodo. Proin sodales pulvinar tempor. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Nam fermentum, nulla luctus pharetra vulputate, felis tellus mollis orci, sed rhoncus sapien nunc eget.' Below this, there is a search bar with placeholder text 'Busque por nombre de doctor, vacuna, tipo de registro, descripción, etcétera...', and dropdown menus for 'Todos' and 'Cuálquiera'. The main list contains four items:

- Diazepam 30mg**: Includes a small icon of a syringe and ampoule, the name 'Diazepam 30mg', the owner 'Kenny Bell', and the date '25/09/2011'. A detailed status message is present below it.
- Tetravalente 10ml**: Includes a small icon of a syringe and ampoule, the name 'Tetravalente 10ml', the owner 'Kenny Bell', and the date '25/09/2011'. A detailed status message is present below it.
- Revisión rutinaria de agosto**: Includes a small icon of a clipboard, the name 'Revisión rutinaria de agosto', the owner 'Kenny Bell', and the date '25/09/2011'. A detailed status message is present below it.

Figure 7: Expediente de Mascota desde Dashboard del Dueño