Desarrollo De Sistemas De Información Orientados A La Gestión Y Apoyo A Las Decisiones

Desarrollo De Un Sistema Para El Registro De Datos De Una Veterinaria

Prof.: Lautaro Muñoz

GRUPO 3 - *“Sistema de Registro ”*

-Di Napoli Federico Ronaldo

- Juan Ignacio Podestá

- Ariadna Menéndez

- Acevedo Mauricio Sebastián

CONTENIDO

CAPÍTULO 1

INTRODUCCIÓN

Este informe describe la problemática que existe en las clínicas veterinarias al momento de llevar un orden en sus registros de clientes-mascotas y los trabajadores de la clínica. El problema nace porque las fichas son llenadas sin un formato y por cualquier trabajador que no se identifica, además de ser almacenadas en distintos lugares. Como solución, se ofrece realizar un sistema que manejará esta información de manera más ordenada, en un sólo lugar, y con un sistema multi-usuarios.

LA CLÍNICA

La veterinaria San Agustín es una clínica veterinaria que se dedica al cuidado de animales, posee cuidados medicinales (rayos X, cirugías, vacunas, alimentación, farmacia, etc.), estéticos (baños, peluquería, etc.) y otros servicios (hospitalización, hotel, venta de productos). Fue creada por un grupo de médicos veterinarios que quería ofrecer la mejor medicina posible para perros, gatos y mascotas exóticas.
Actualmente la clínica reside en el barrio de Parque Patricio en la Ciudad Autónoma de Buenos Aires, donde además cuenta con tres veterinarios titulados de prestigiosas universidades, los cuales prestan servicios a la clínica y están disponibles para consultas a domicilio

1.4 LOS EMPLEADOS

La clínica veterinaria consta de tres médicos veterinarios encargados de la salud de las mascotas; una secretaria que registra a los clientes y mascotas al sistema; un administrador que se encarga del abastecimiento del material clínico y alimentario, hotelería y hospitalización; dos alumnos en práctica; y un encargado de mantener el aseo en la clínica.

CAPÍTULO 2

DEFINICIÓN DE OBJETIVOS

Con el problema medianamente planteado se decide realizar un objetivo general que será la meta principal del proyecto, y un listado de objetivos específicos los cuales son el medio para obtener este objetivo general.

OBJETIVO GENERAL

Modelar e implementar un sistema INFORMÁTICO para la gestión de clínicas veterinarias, que permita almacenar información acerca de clientes, pacientes, y servicios, de manera eficiente y modernizada.

OBJETIVOS ESPECÍFICOS

- Comprender la situación actual.
- Analizar los sistemas existentes para el manejo de información.
- Investigar tecnologías existentes y herramientas de sistemas web.
- Comprender la arquitectura, tecnologías, metodologías y paradigmas a utilizar.
- Modelar e implementar el nuevo sistema.
- Realizar pruebas con usuarios.

CAPÍTULO 3

SITUACIÓN ACTUAL

En la actualidad, existe escaso orden sobre la información almacenada, por lo que hace aún más difícil su búsqueda a la hora de necesitar algún dato de estos registros. Estos se irán detallando caso a caso, además se indicarán los problemas que pueda ocasionar.

− Registro de Mascotas y Clientes
Cuando un cliente visita por primera vez la clínica veterinaria se le asigna un registro y un número identificador que serán almacenados en una agenda.
Esta agenda sólo incluye datos del cliente y número identificador. Luego dependiendo de la cantidad de mascotas que posea, se le asigna un registro para cada una, que se ubica en un armario, la que incluye además el número identificador del cliente.
Cuando el cliente visita nuevamente, la secretaria deberá buscar en la agenda el registro del cliente para localizar su número identificador. Ya con este número, podrá buscar la ficha de la mascota en el armario. En esta ficha además se anota la prescripción médica por el doctor

− Procedimientos Efectuados
Ya con la ficha en las manos del doctor, éste anota los procedimientos que fueron realizados y los que se realizarán, en caso de necesitar, son registrados en la ficha de la mascota para así llevar un registro.

PROBLEMAS DETECTADOS

Ya mencionados los procedimientos que son realizados en la clínica veterinaria, se procede a describir cada problema que se ocasiona:
− Registro de Mascotas y Clientes
Cuando un funcionario de la clínica desea buscar una ficha de la mascota, ésta debe buscarla manualmente en el armario, esto se complica más aún cuando existen más pacientes. Lo cual toma mucho tiempo.
− Procedimientos Efectuados
Como los procedimientos que serán efectuados están escritos en la ficha de la mascota, la mayoría de las veces, la clínica no recuerda a sus pacientes. Por lo que no asisten a ser analizados por el especialista. Otro problema sucede que los doctores no se identifican, por lo que no se sabe quién hizo qué.

CAPÍTULO 4

PROPUESTA

En esta sección se detallará los procesos que serán realizados cuando el sistema ya se encuentre en funcionamiento. El nuevo sistema será gestionado a través de una aplicación de escritorio , que será instalado en la computadora de la veterinaria

¿Qué es una aplicación de escritorio?

Una aplicación de escritorio es aquella que se encuentra instalado en el ordenador o sistema de almacenamiento y podemos ejecutarlo sin internet en nuestro sistema operativo, al contrario que las aplicaciones en la nube que se encuentran en otro ordenador al que accedemos a través de la red o internet a su software.

Ventajas:

- Pueden ser más robustas
- Tiempo de respuesta más rápido
- Se puede hacer cualquier cosa que permita el Software (cuestión gráfica, control total de las entradas del usuario al momento de capturar)
- Facilita el uso de teclas en caliente (ejemplo: CTRL+G para grabar)

Desventajas:

- Requiere instalación en cada cliente
- Generalmente se hacen para un Sistema Operativo específico
- Se requiere actualizar en cada cliente

Características del Nuevo Sistema

Dada a los problemas que posee la clínica veterinaria, se han detallado varias funcionalidades que tendrá el nuevo sistema, las que serán resumidas a continuación:

− Registro de Mascotas y Clientes

El sistema poseerá un sólo almacén o base de datos, donde al consultar sobre un cliente (ya sea por nombre o número identificador) automáticamente mostrará las mascotas asociadas. Y así generar mayor rapidez al momento de la búsqueda.

-Autentificación

Ya que en todos los procedimientos registrados en la ficha de la mascota no se indica quién los realizó. El sistema permitirá registrar qué usuario y cuándo realizó dicha acción. De tal manera que haya un orden de los procedimientos.

Ventajas del Nuevo Sistema

Además de las ventajas mencionadas anteriormente el nuevo sistema tendrá las siguientes características:

• Disponibilidad: El sistema estará disponible para que cualquier funcionario de la veterinaria pueda utilizarlo. No se necesitará conectarse a internet para acceder al sistema.

• Usable: Proporcionará facilidades de uso, de tal manera de evitar frustraciones en el uso del sistema.

• Mayor Rapidez: Para acceder a la información.

• Interfaz intuitiva: Será sencillo de manejar.

DETALLE DE SOLUCIÓN

Ya teniendo la propuesta del nuevo sistema se debe definir cómo será resuelta así mismo eligiendo la metodología de desarrollo, el paradigma que incluye los modelos de desarrollo, las herramientas que se utilizarán y la arquitectura que poseerá el sistema cuando esté en funcionamiento.

METODOLOGÍA

Para que se pueda desarrollar correctamente el software, se debe determinar que metodología se utilizará. Por tal, se deberá elegir entre estructurada y orientada a objetos.

Desarrollo Estructurado

• El sistema se modela con un enfoque orientado al flujo de datos.

• Se pueden aplicar paradigmas de programación procedimental, modular o abstracción de datos para desarrollar software.

• Utilizar un método ascendente: descomposición funcional basada en subprocesos de procesos de niveles superiores.

Desarrollo Orientado a Objetos

• El sistema se modela con un enfoque orientado a objetos.

• Utiliza un paradigma de programación orientada a objetos.

• Utiliza un método ascendente: composición de clases basadas en abstracción de datos.

Elección de Metodología

Para el presente proyecto se decidió utilizar la metodología orientada a objetos, porque posee varias ventajas en relación a la otra metodología mencionada anteriormente:

• El enfoque orientado a objeto proporciona una mejor forma de validar los requerimientos.

• El problema se puede dividir en varios objetos, lo cual facilitará llevar a cabo el sistema.

• Permite la iteración durante el desarrollo de un proyecto, lo que conlleva a un mejor entendimiento y resolución del sistema a desarrollar.

• Además esta metodología va de la mano con la utilización del Proceso

Unificado como paradigma de desarrollo y UML como herramienta de modelado, lo que facilita aún más el desarrollo de un sistema informático.

METODOLOGÍA AGILE

Es un sistema de trabajo que está cambiando el desarrollo de proyectos de muchas empresas, entre ellas, Google, Amazon o Microsoft. Este nuevo concepto triunfa en todo el mundo y ha llegado para quedarse. ¿La clave de su éxito? Organizar y repartir el trabajo de una manera rápida y flexible entre diferentes equipos multidisciplinares.

La metodología Agile mantiene la dirección sin caer en la rigidez de los conocidos métodos en cascada o waterfall. Estos planean el trabajo desde el principio, sin lugar a imprevistos. De forma que cuando aparecen, resulta imposible reaccionar a tiempo. El agilismo, sin embargo, mantiene la capacidad de tomar la mejor opción en cada momento sin comprometer el proyecto. Los métodos Agile más populares del momento son Scrum y Kanban.

Desde sus inicios, la metodología Agile reivindica 4 valores:

- Las interacciones de las personas sobre los procesos y las herramientas.
- Un software en funcionamiento frente a documentación exhaustiva.
- La participación activa del cliente durante todo el proceso de desarrollo.
- La capacidad de respuesta ante los cambios e imprevistos.

https://www.progressalean.com/metodologia-agile/

HERRAMIENTAS Y TECNOLOGÍAS

En esta sección se detallarán todas las tecnologías utilizadas para el diseño, construcción y visualización de este nuevo sistema.

El proyecto se realizó en su totalidad con Windows Presentation Foundation es una tecnología de Microsoft, presentada como parte de Windows Vista. Permite el desarrollo de interfaces de interacción en Windows tomando características de aplicaciones Windows y de aplicaciones web

Calendarización y Planificación de Proyectos :

Trello es un software de administración de proyectos con interfaz web y con cliente para iOS y android para organizar proyectos.

Herramientas de Modelado y Diseño del Sistema:

-Draw.io

-Diagrams.net

Plataformas de Desarrollo:

-Visual Studio Code

-Visual Studio 2022

Motor de Base de Datos :

-Sql Server

-PhpMyAdmin

-Xampp

Lenguajes de Programación :

- C#

ESTUDIO DE FACTIBILIDAD

La finalidad de este estudio es determinar qué tan bueno será la implementación de este nuevo sistema en la clínica veterinaria antes mencionada. Además se verán los costos que tendrá que incurrir la clínica.

Las preguntas que orientan el estudio de factibilidad, entre otras, son: ¿Es realmente necesario? ¿Es beneficioso? ¿Es rentable, económica y socialmente? ¿Están las condiciones para emprender el proyecto?

TÉCNICA

Se deben considerar aspectos técnicos requeridos para el desarrollo del nuevo sistema, como hardware, software y recursos humanos.

Hardware

El sistema que se va a realizar debe poseer la capacidad técnica de manejar una cantidad de datos, por lo que se requiere de un computador servidor (arquitectura cliente-servidor) que soporte el almacenamiento de base de datos con todos los registros de la clínica veterinaria.

Para realizar las etapas de análisis de requerimientos, diseño, implementación, pruebas y documentación se cuenta con el siguiente hardware:

Factibilidad de Hardware.

Laptop Procesador: AMD Turion II

Memoria RAM: 4 Gb

Disco Duro: 500 Gb

Sistema Operativo: Windows 7 Ultimate

Tarjeta de Red: Gigabit Ethernet Broadcom Netlink

Tarjeta de Red Inalámbrica: Atheros ARB93 Wireless Network

Por lo que la clínica no deberá adquirir ningún sistema computacional nuevo para el desarrollo, ya que existe suficiente hardware para llevar a cabo el proyecto.

Recurso Humano

Para la realización de este proyecto se cuenta con personal que posee las siguientes competencias académicas:

• Lenguaje de Programación: C#.

• Modelamiento de Software: Lenguaje de modelado UML.

• Base de Datos: MySQL server.

OPERACIONAL

Esta factibilidad comprende una determinación de la probabilidad de que un nuevo sistema se use como se supone, y se evalúa el impacto que causará en la clínica.

Los empleados saben utilizar software de oficina como Word, Excel, etc. y conocimientos básicos de computación e internet. Por lo que se requiere de una pequeña capacitación a los usuarios de este nuevo sistema de tal manera que puedan utilizar el sistema en su totalidad.

Algunas características del nuevo sistema cuando esté en funcionamiento:

- Este sistema propone reemplazar el sistema anterior.

- Con el nuevo sistema aumentará la eficiencia en los procesos de los registros.

- Disminuirá los tiempos de respuesta del sistema actual.

PLANIFICACIÓN DEL PROYECTO

Etapas del desarrollo de metodologías de software ágiles

1.Determinación del alcance y la prioridad de los proyectos

Durante el primer paso del ciclo de vida del desarrollo de software con la metodología ágil, el equipo planifica y prioriza proyectos. Algunos equipos pueden trabajar en más de un proyecto al mismo tiempo, según la organización del departamento.

Para cada concepto, debes definir la oportunidad comercial y determinar el tiempo y el trabajo necesarios para completar el proyecto. A partir de esa información, puedes evaluar la viabilidad técnica y económica y decidir qué proyectos vale la pena seguir.

2. Diagrama de requisitos para el sprint inicial

Una vez que hayas identificado el proyecto, trabaja con las partes interesadas para determinar los requisitos. Puede ser útil usar diagramas de flujo de usuario o diagramas de UML de alto nivel para demostrar cómo debería funcionar la nueva función y cómo encajará en el sistema existente.

Luego, selecciona miembros del equipo para trabajar en el proyecto y asigna los recursos. Crea un cronograma o un mapa de proceso con carriles en Lucidchart para delinear responsabilidades y mostrar claramente cuándo se debe completar cierto trabajo durante el sprint.

3. Construcción/iteración

Una vez que un equipo ha definido los requisitos para el sprint inicial en función de los comentarios y requisitos de las partes interesadas, comienza el trabajo. Los diseñadores y desarrolladores de UX comienzan a trabajar en su primera iteración del proyecto, con el objetivo de tener un producto que funcione para lanzarse al final del sprint. Recuerda que el producto se someterá a varias rondas de revisión, por lo que esta primera iteración puede solo incluir la funcionalidad básica mínima.

El equipo puede tener (y los tendrá) sprints adicionales para expandir todo el producto.

4. Puesta en producción de la iteración

Ya casi estás listo para lanzar el producto al mundo. Finaliza esta iteración de software con los siguientes pasos:

- Prueba el sistema. Tu equipo de control de calidad (QA) debe probar la funcionalidad, detectar errores y registrar las cosas buenas y las cosas malas.
- Arregla cualquier defecto.
- Finaliza el sistema y la documentación del usuario. Lucidchart puede ayudarte a visualizar tu código a través de diagramas de UML o a mostrar los flujos de usuarios para que todos comprendan cómo funciona el sistema y cómo pueden desarrollarlo más a fondo.
- Pon la iteración en producción.

5. Producción y soporte continuo para la versión del software

Esta fase implica el soporte continuo para la versión del software. En otras palabras, tu equipo debe mantener el sistema funcionando sin problemas y mostrarles a los usuarios cómo se usa. La fase de producción finaliza cuando el soporte ha finalizado o cuando se planifica el retiro de la versión.

6. Fase de retiro

Durante la fase de retiro, eliminas la versión del sistema de la producción, normalmente cuando quieres reemplazar un sistema por una nueva versión o cuando el sistema se vuelve redundante, obsoleto o contrario a tu modelo de negocio.

Planificación de un sprint de desarrollo de software con la metodología ágil

Dentro del ciclo de vida del desarrollo de software (SDLC) ágil, el trabajo se divide en sprints, con el objetivo de producir un producto funcional al final de cada sprint. Un sprint suele durar dos semanas, o 10 días hábiles. El flujo de trabajo de un sprint debe seguir este esquema básico:

- Planificación. El sprint comienza con una reunión de planificación, donde los miembros del equipo se reúnen para establecer los componentes para la próxima ronda de trabajo. El director de producto prioriza el trabajo a partir de las tareas pendientes para asignar al equipo.
- Desarrollo. Diseño y desarrollo del producto de acuerdo con las pautas aprobadas.
- Prueba/QA. Pruebas exhaustivas completas y documentación de los resultados antes de la implementación.
- Implementación. Presentación del producto o software en funcionamiento a las partes interesadas y los clientes.
- Evaluación. Solicitud de comentarios al cliente y a las partes interesadas y recopilación de información para incorporar en el próximo sprint.

Además de las reuniones de planificación de sprints, el equipo debe reunirse diariamente para ver cómo van las cosas y ponerse al día con respecto al progreso, hablar sobre los conflictos y trabajar para que el proceso siga avanzando.

También es necesario mantenerse flexible y abierto al cambio. Después de todo, esta metodología se llama "ágil" por una razón.

Conclusión: el objetivo del ciclo de vida del desarrollo de software con la metodología ágil es crear e implementar software que funcione lo antes posible.

REQUERIMIENTOS DEL SISTEMA

Este capítulo comprende los requerimientos del sistema, donde primero se especifican los usuarios del sistema, más tarde mencionados como actores, luego se identifican los requerimientos del sistema y finalmente los casos de uso.

CARACTERISTICAS DE LOS USUARIOS

En esta sección se definirá las funciones que realizan cada uno de los distintos usuarios del sistema.

Médicos

Como el sistema está basado en una clínica veterinaria, en esta existen 3 médicos veterinarios además de alumnos en práctica que utilizarán este sistema identificándose como Médicos. Estas personas están encargadas gestionar la información del paciente sólo del área medicinal. Tienen conocimientos de palabras técnicas del área medicinal, pero poco conocimiento computacional.

Secretarias

Además se posee una secretaria, que es la encargada de almacenar la información del paciente, cuando éste recién llega, y administra la información del cliente. Debe poseer conocimientos básicos computacionales, y saber llenar formularios.

Requerimientos Funcionales

En esta sección se detallará las funcionalidades del nuevo sistema, los que serán divididos por los usuarios que lo utilizarán.

Secretarias

• El sistema debe permitir agregar, ver, eliminar, y modificar los datos de un cliente.

• Debe permitir agregar, ver, eliminar, y modificar los datos del paciente.

• El sistema debe permitir buscar a un cliente según su nombre, o apellido. Si el sistema no encuentra al cliente, deberá mostrar un mensaje de error y si encuentra más de uno deberá mostrar los que coinciden con la búsqueda.

• Debe permitir crear, buscar, ver, modificar y eliminar un registro del paciente.

Médicos

• El sistema permitirá crear, ver, eliminar y modificar registro de atención del paciente

Requerimientos No Funcionales

• El sistema debe permitir inicio de sesión segura, de tal manera que exista una correcta autenticación.

• Debe indicar el formato correcto de la información que debe ingresar el usuario al sistema en caso de que no corresponda a lo solicitado.

ANÁLISIS Y DISEÑO DEL SISTEMA

MODELOS DEL SISTEMA

Caso de Uso General:

GESTIONAR CLIENTES

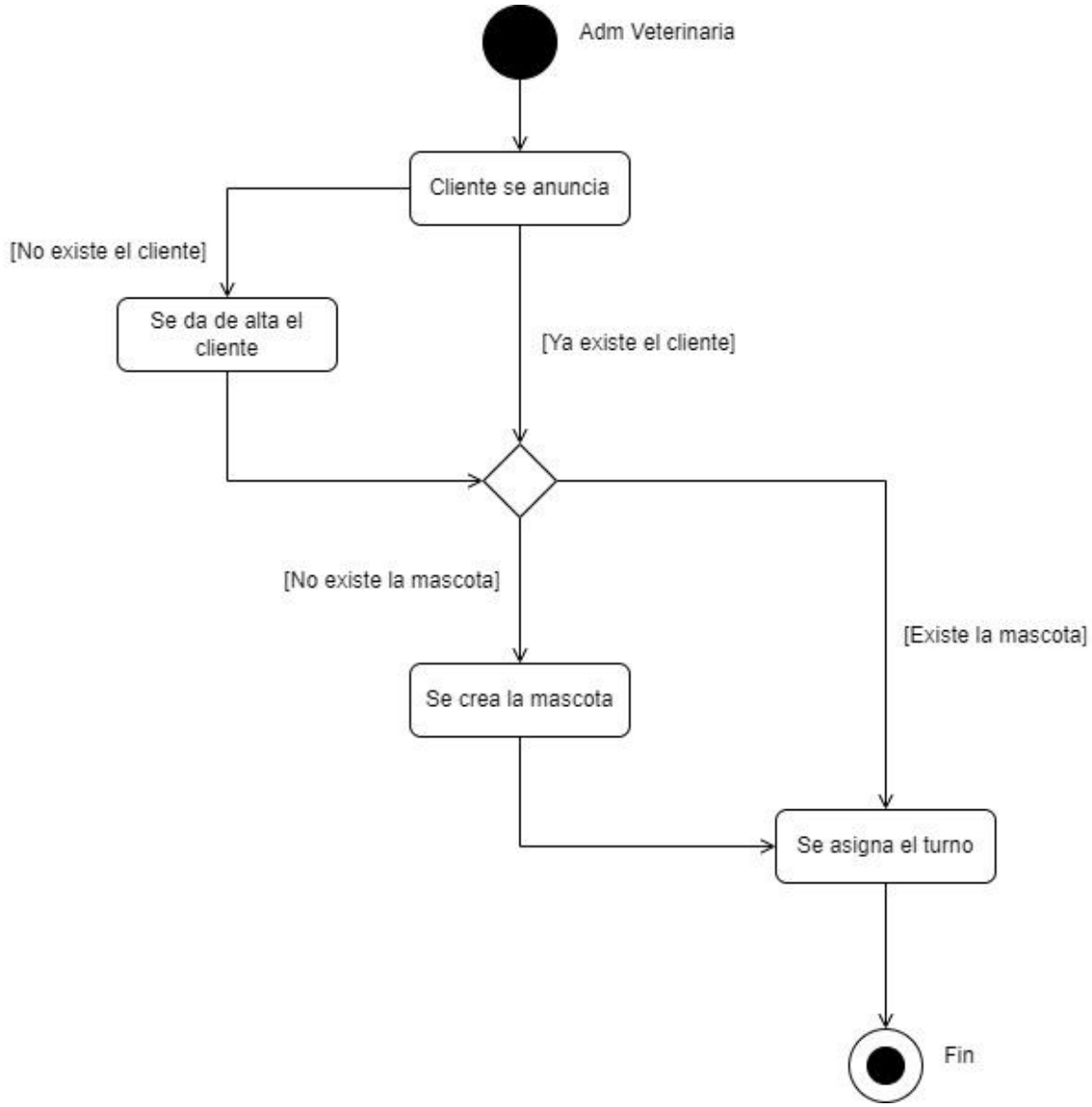
Buscar y Modificar Cliente

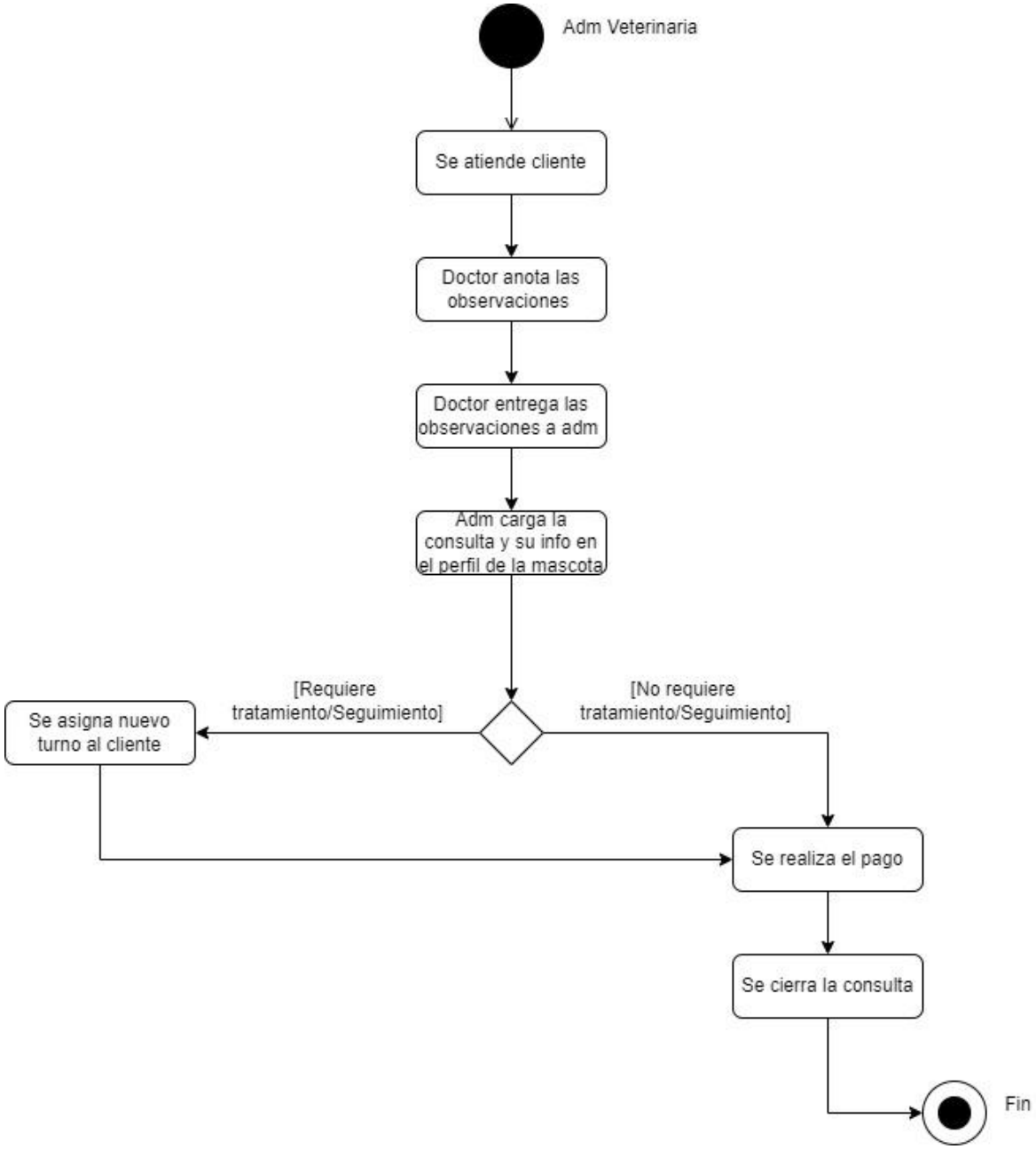
GESTIONAR PACIENTES

GESTIONAR REGISTROS

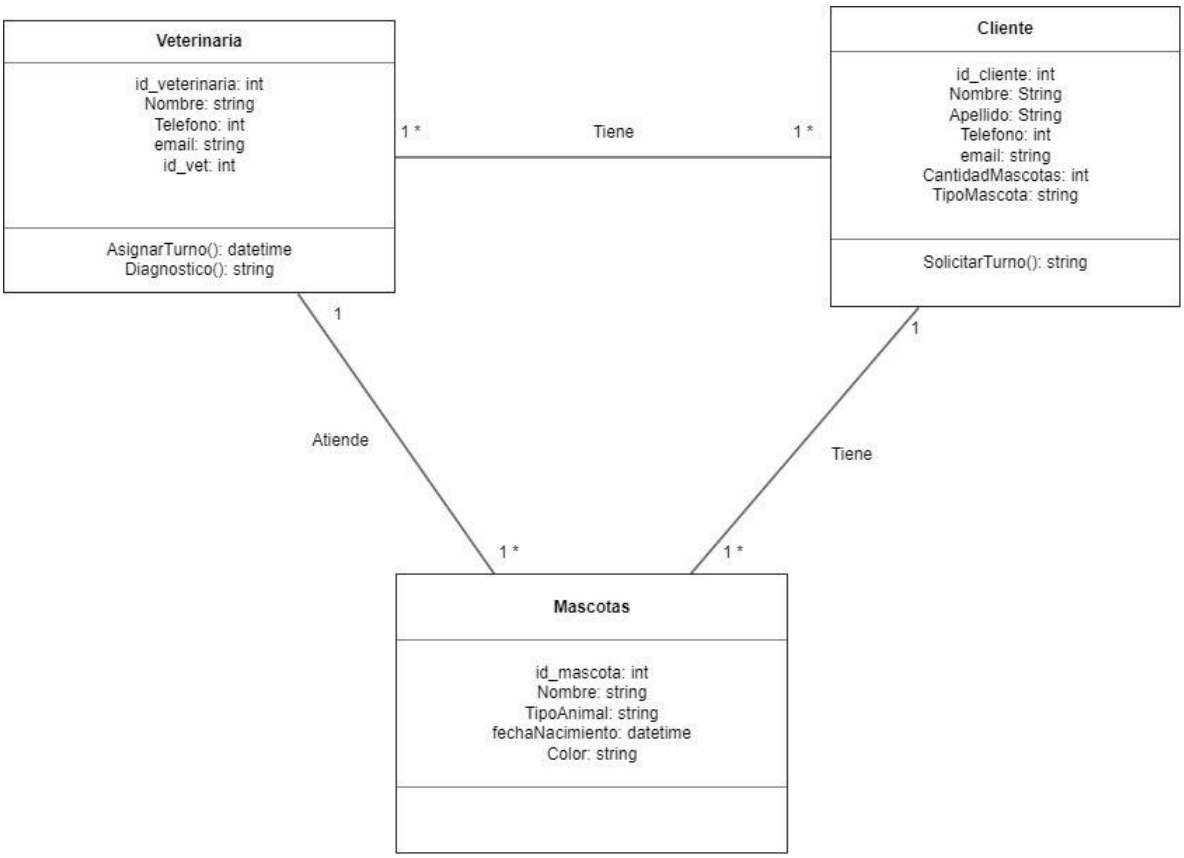
GESTIONAR USUARIOS

Diagrama de Actividades




Adm Veterinaria
Se atiende cliente
Doctor anota las observaciones
Doctor entrega las observaciones a adm
Adm carga la consulta y su info en el perfil de la mascota
[Requiere tratamiento/Seguimiento]
[No requiere tratamiento/Seguimiento]
Se asigna nuevo turno al cliente
Se realiza el pago
Se cierra la consulta
Fin

Con las interacciones de los usuarios en los modelos anteriores se obtiene el siguiente diagrama de clases:



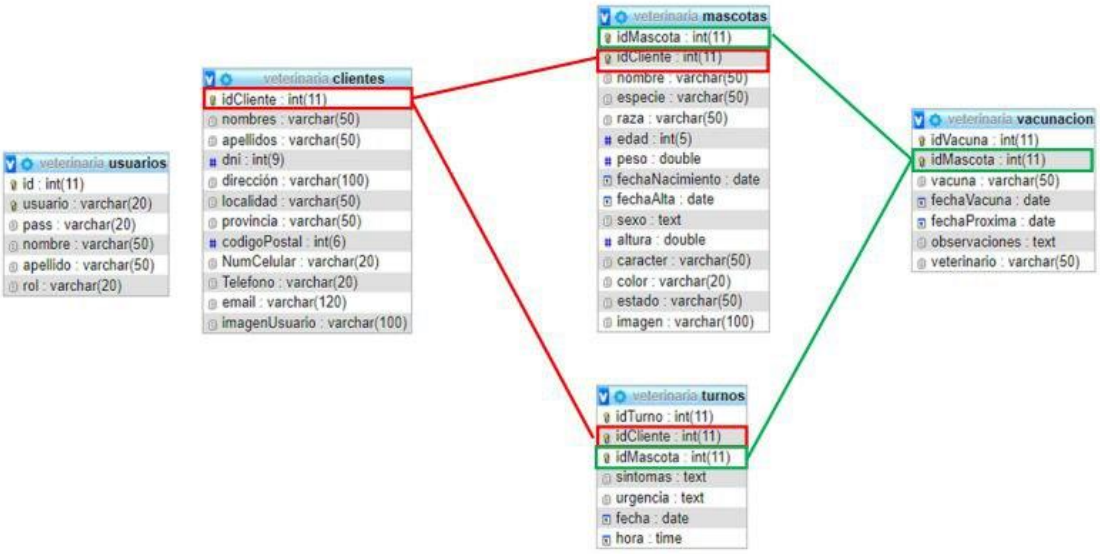
MODELO BASE DE DATOS

A partir del diagrama de Clases se han diseñado con las clases persistentes, y la relación que poseen cada una de ellas el siguiente diagrama de Entidad – Relación:

Modelo Entidad-Relación

Modelo Relacional Ya con el modelo de Entidad – Relación se puede obtener el Modelo Relacional, con este modelo se deben detallar las dependencias de cada una de las tablas, los campos, las claves primarias (PK) y claves foráneas (FK), y a partir de éste, se crea la base de datos que será finalmente utilizada por el sistema.

Diagrama Entidad-Relación



clientes

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	MIME
idCliente *(Primaria)*	int(11)	No				
nombres	varchar(50)	No				
apellidos	varchar(50)	No				
dni	int(9)	No				
dirección	varchar(100)	No				
localidad	varchar(50)	No				
provincia	varchar(50)	No				
codigoPostal	int(6)	No				
NumCelular	varchar(20)	No				
Telefono	varchar(20)	No				
email	varchar(120)	No				
imagenUsuario	varchar(100)	No				

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	idCliente	0	A	No	

mascotas

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	MIME
idMascota *(Primaria)*	int(11)	No				
idCliente *(Primaria)*	int(11)	No				
nombre	varchar(50)	No				
especie	varchar(50)	No				
raza	varchar(50)	No				
edad	int(5)	No				
peso	double	No				
fechaNacimiento	date	No				
fechaAlta	date	No				
sexo	text	No				
altura	double	No				
caracter	varchar(50)	No				
color	varchar(20)	No				
estado	varchar(50)	No				
imagen	varchar(100)	No				

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	idMascota	0	A	No	
				idCliente	0	A	No	

turnos

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	MIME
idTurno *(Primaria)*	int(11)	No				
idCliente *(Primaria)*	int(11)	No				
idMascota *(Primaria)*	int(11)	No				
sintomas	text	No				
urgencia	text	No				
fecha	date	No				
hora	time	No				

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	idCliente	0	A	No	
				idTurno	0	A	No	
				idMascota	0	A	No	

usuarios

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	MIME
id *(Primaria)*	int(11)	No				
usuario *(Primaria)*	varchar(20)	No				
pass	varchar(20)	No				
nombre	varchar(50)	No				
apellido	varchar(50)	No				
rol	varchar(20)	No				

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	id	0	A	No	
				usuario	0	A	No	

vacunacion

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	MIME

idVacuna *(Primaria)*	int(11)	No				
idMascota *(Primaria)*	int(11)	No				
vacuna	varchar(50)	No				
fechaVacuna	date	No				
fechaProxima	date	No				
observaciones	text	No				
veterinario	varchar(50)	No				

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	idVacuna	0	A	No	
				idMascota	0	A	No	

LA INTERFAZ DEL SISTEMA

La interfaz de una aplicación de escritorio es el conjunto gráfico que permite la presentación y la navegación del sistema. Esto se consigue con la inclusión de elementos gráficos comunes a todo el sistema que son estándares, haciendo que los usuarios tengan completo control sobre las funcionalidades desde el momento mismo de entrar a él sin que para ello deba tener amplios conocimientos ni preparación anterior alguna. Para lograr que la interacción con el usuario sea lo más intuitiva posible, se deben utilizar recursos como la gráfica, pictogramas, estereotipos, y símbolos, todo sin afectar el funcionamiento.

AUTENTIFICACIÓN

Para entrar al sistema, todo usuario debe ser autentificado mediante el sistema de log-in.

PANTALLA PRINCIPAL

Cuando ya está iniciada la sesión, la primera pantalla que se muestra es el INICIO.

MÓDULOS DEL SISTEMA

A continuación se detallarán los distintos módulos del sistema, vistos anteriormente en el menú lateral del sistema.

Gestionar Clientes
Se muestra un listado de los clientes, ordenado alfabéticamente, con la información más relevante y las mascotas que posee.

Gestionar Pacientes
Se muestra un listado de los pacientes, ordenados alfabéticamente, y con su respectivo dueño.

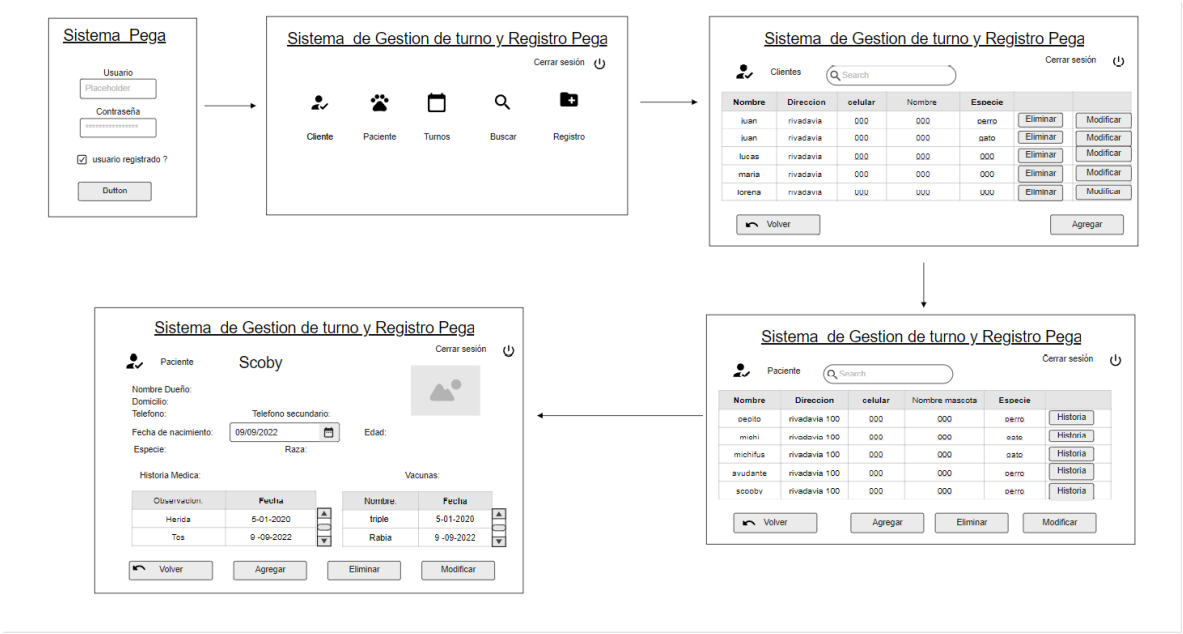
Gestionar Registros

Se detallan los registros realizados por los médicos, indicando el paciente, su dueño, la fecha de realización, y un resumen para tener una noción de que trata dicho registro.

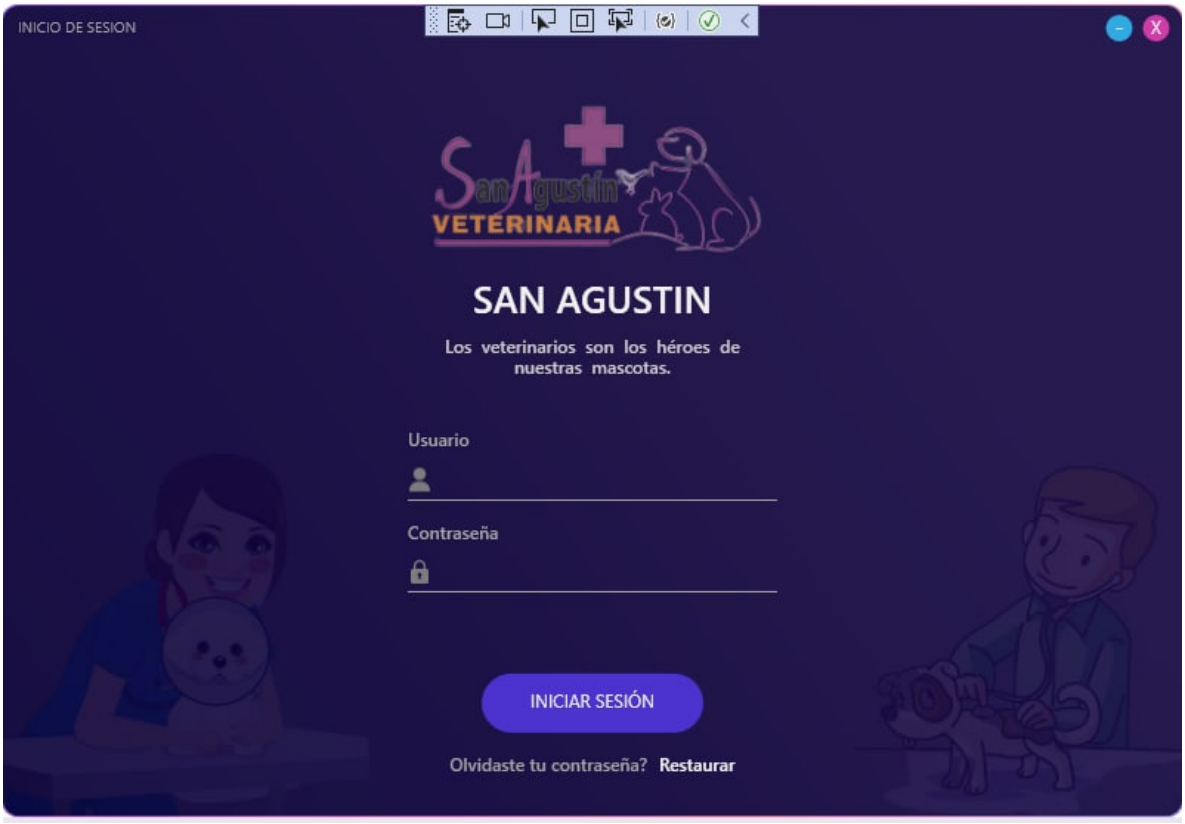
Usuarios

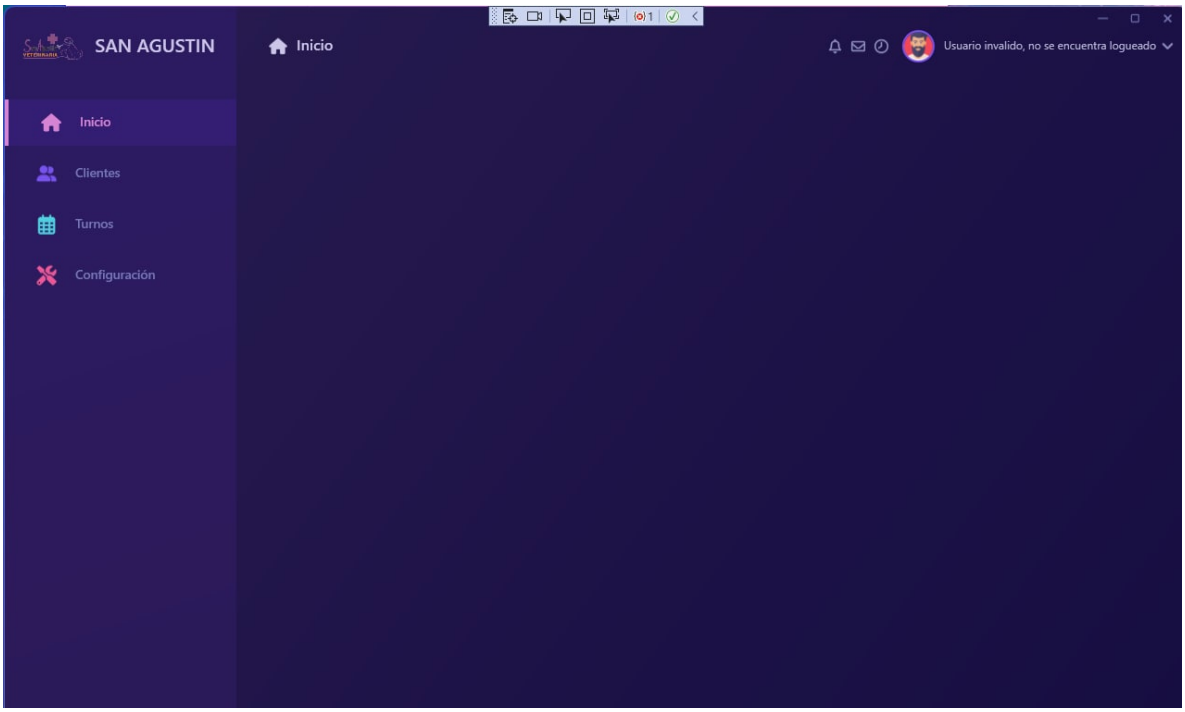
Módulo especialmente del administrador, donde se listan todos los usuarios del sistema, identificados por su perfil.

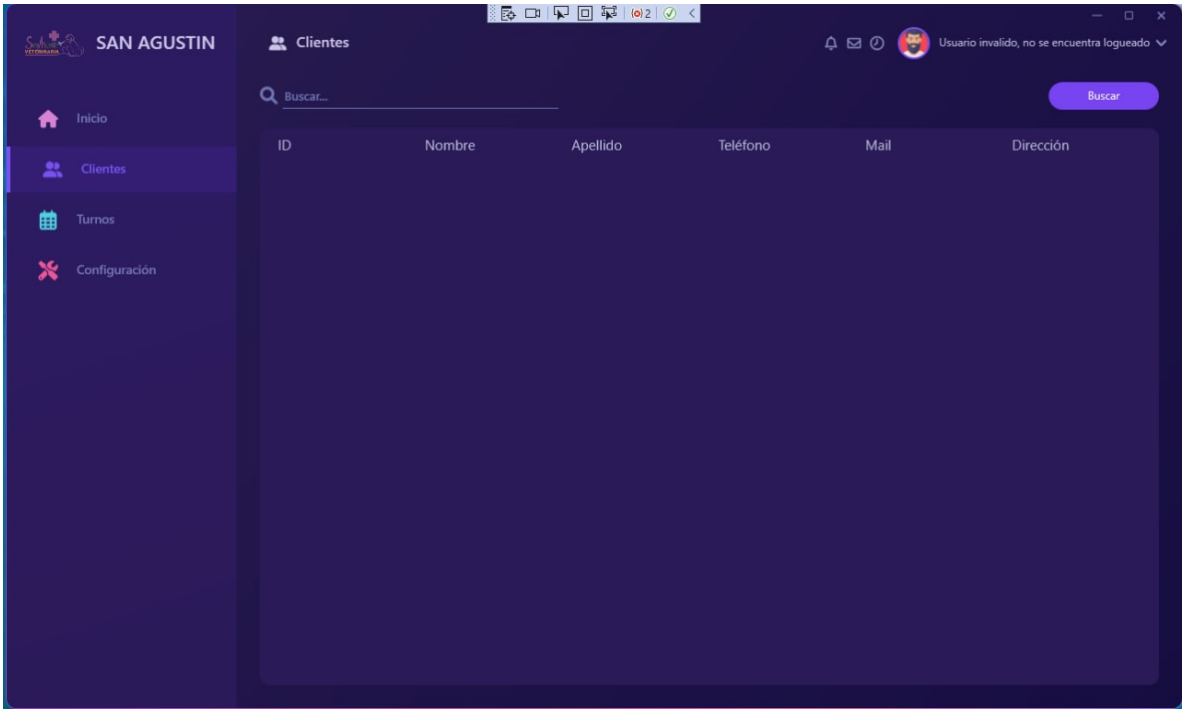
Maquetado Versión 1.0



Versión 2.0


INICIO DE SESION
SAN AGUSTIN
VETERINARIA
SAN AGUSTIN
Los veterinarios son los héroes de nuestras mascotas.
Usuario
Contraseña
INICIAR SESIÓN
Olvidaste tu contraseña? Restaurar


SAN AGUSTIN
Inicio
Usuario invalido, no se encuentra logueado
Inicio
Clientes
Turnos
Configuración



OTRAS FUNCIONALIDADES

Menú Lateral

Durante todo el sistema se considerará el uso de un menú lateral con los módulos importantes del sistema. Éste puede variar dependiendo el usuario.

Botones En cada módulo, se utilizan distintos botones que representan una acción.

PLAN DE PRUEBAS

El objetivo último del proceso de verificación y validación es establecer la seguridad de que el sistema software está “hecho para un propósito”. Esto significa que el sistema debe ser lo suficientemente bueno para su uso pretendido. El nivel de confianza requerido depende del propósito del sistema, las expectativas de los usuarios del sistema y el entorno de mercado actual del sistema.

OBJETIVOS

Las pruebas en un software son aplicadas como una etapa más del proceso de desarrollo del software y su objetivo es asegurar que el software cumpla con las especificaciones requeridas y eliminar los posibles defectos que éste pudiera tener.

Objetivos de las Pruebas

• Encontrar defectos en el software.

• Una prueba es considerada exitosa si se descubre un defecto.

• Una prueba fracasa si hay un defecto y no se descubre.

• Asegurar la calidad del software.