



UNIVERSIDAD DE ARTES, CIENCIAS Y COMUNICACIÓN
Facultad de Administración
Carrera de Ingeniería Informática Multimedia

Proyecto de Integración – Sistema de Información Agroveterinario

Proyecto para optar al Grado Académico de Licenciado/a en
Ingeniería Informática Multimedia y al Título Profesional de Ingeniero
Informático Multimedia

Profesor Guía:

Felipe Alejandro Montenegro Gonzalez

Estudiante:

Erick Alfonso Pinto Beltrán

Santiago de Chile, 21 de Marzo 2022



INFORME FINAL

PROYECTO DE INTEGRACIÓN

ERICK ALFONSO PINTO BELTRAN
Ingeniería Informática Multimedia

Tabla de contenido

I. PLANTEAMIENTO DEL PROBLEMA	3
1.1. MATRIZ FODA / ROLA.....	3
II. REFERENTES TEÓRICOS o MARCO TEÓRICO	5
III. REFERENTES METODOLÓGICOS	6
3.1. DISEÑO DE PANTALLAS	8
3.2. ATRIBUTOS Y ENTIDADES.....	9
3.3. ATRIBUTOS UNICOS	10
3.4. Modelo Entidad – Relación	11
3.5. DEFINICION DE ENTIDADES.....	12
IV. PLANIFICACIÓN Y DESARROLLO DEL PROYECTO	13
4.1. RECURSOS A UTILIZAR.....	13
4.2. ANALISIS	13
4.3. IMPLEMETACION	14
4.4. PRUEBAS	15
4.4.1. LIMITAR EL ALCANCE	15
4.4.2. PRUEBAS DE CONTENIDO	15
4.5. A MODO DE EJEMPLO SE REALIZARÁ LA SIGUIENTE PRUEBA DE CARGA.....	16
4.5.1. PRIMERA PRUEBA.....	16
4.5.2. SEGUNDA PRUEBA.....	16
4.5.3. GRAFICA DE SIMULACIÓN Y EJEMPLO DE RESULTADOS	17
4.5.4. PRUEBAS DE ESTRÉS	17
4.5.5. PRUEBAS DE CAPACIDAD	18
4.5.6. PRUEBAS DE RENDIMIENTO BAJA INTENSIDAD	18
4.5.7. PRUEBAS DE RENDIMIENTO DE ALTA INTENSIDAD	18
4.5.8. VISUALIZAR RESTRICCIONES DEL RENDIMIENTO	19
4.6. REQUERIMIENTOS FUNCIONALES	19
EFICIENCIA	19
SEGURIDAD LÓGICA Y DE DATOS	20
OTROS REQUERIMIETOS NO FUNCIONALES	20
USABILIDAD	20

I. PLANTEAMIENTO DEL PROBLEMA

Para presentar el problema que se identifica en el sistema de información de la empresa “MI MASKOTA”, se detalla la siguiente información:

1.1. MATRIZ FODA / ROLA

	RECURSOS	LIMITACIONES
AGROVETERINARIA “MI MASKOTA”	R1 Médico Veterinario y Auxiliar. R2 Computadores en funcionamiento. R3 Trabajo en Equipo. R4 Disposición para búsqueda de soluciones.	L1 Proceso de registro de pacientes de forma lenta. L2 Resguardo de la información de una forma no adecuada. L3 Falta de conocimiento en el área de informática, en especial los Sistemas de Información.
OPORTUNIDADES	ESTRATEGIAS RO	ESTRATEGIAS LO
O1 Comunidad atenta a Propuestas y Soluciones. O2 Instituto Universitario O3 Disposición al conocimiento.	Aprovechar los equipos de computación para el uso administrativo y el trabajo en equipo con vinculación a los estudiantes del IU. (R2, R3, O1, O2)	Aplicación de herramientas informáticas que faciliten un mejor proceso de registro y resguardo de la información. (O1, O2, L1, L2,) Adiestramiento en el uso de un Sistema de Información. (L3, O3)
AMENAZAS	ESTRATEGIAS RA	ESTRATEGIAS LA
A1 Fallas Eléctricas. A2 Daños Imprevistos.	Uso del Protector Eléctrico. (A1, A2, R2)	Concientizar al personal para que tomen precauciones con respecto a las fallas eléctricas y el uso del protector eléctrico. (A1, A2, L3)

Se concluye que la matriz anterior permite analizar elementos internos o externos de programas y proyectos.

Representa a través de una matriz de doble entrada, y su nombre es matriz FODA/ROLA, en donde se analiza los factores positivos y negativos, además se analiza los factores internos y por tanto controlables del programa o proyecto y los factores externos, considerados no controlables.

Las Fortalezas son todos aquellos elementos internos y positivos que diferencian al programa o proyecto de otros de igual clase. Las Oportunidades que se observan en una matriz FODA, son aquellas situaciones externas y positivas que se generan en el entorno y que una vez identificadas pueden ser aprovechadas. Por último, las amenazas que se identifican son situaciones negativas, externas al programa o proyecto, que pueden atentar contra éste, por lo que, llegado al caso, puede ser necesario diseñar una estrategia adecuada para poder sortearla.

En el cuadro mostrado anteriormente se puede observar que en la Agroveterinaria "Mi Maskota" posee una serie de recursos y oportunidades favorables como lo son el trabajo en equipo y una comunidad atenta a propuestas y soluciones. La agroveterinaria "Mi Maskota" tiene recursos y oportunidades valiosas, tiene limitaciones, el cual hacen que su trabajo sea un poco lento, en cuanto al proceso de registro de los pacientes, ya que no cuentan con un sistema de información dirigido al control de consulta de los pacientes que van al consultorio, ocasionando pérdida de información valiosa, adicional a ello.

Por esta razón, se busca solventar su problemática mediante estrategias adecuadas haciendo uso de los equipos automatizados que existen en la Agroveterinaria "Mi Maskota", aplicando herramientas informáticas que agilicen el proceso de registro y el trabajo.

En el presente informe se mostrarán los pasos a seguir para desarrollar un software que dará solución a una problemática específica de la Agroveterinaria "Mi Maskota", que entre otras cosas necesita automatizar sus procesos de atención al cliente y manejo de la información, como así mismo se elaborara el informe con las pautas y estructura solicitadas, y en base a ello se determinara la base de datos para el desarrollo del sistema y desarrollo del sistema de información automatizado (diseño y construcción).

II. REFERENTES TEÓRICOS o MARCO TEÓRICO

- Solé, M.(2019)*Pruebas de Caja blanca y Caja Negra*. Apunte de clase unidad 1, Proyecto de Integración IV, Universidad UNIACC.
- Moré, JJ, Garbow, BS y Hillstrom, KE (1981). Prueba de software de optimización sin restricciones. *Transacciones ACM en software matemático (TOMS)* , 7 (1), 17-41.
- 2021. Jen Oksnevad. Load View Pruebas de carga.
- Solé, M.(2018). Pruebas de Performance. Apunte de clase unidad 2, Proyecto de Integración IV, Universidad UNIACC.
- Toledo, M. A. (2019). HERRAMIENTAS PARA TESTING DE APLICACIONES WEB. *Conexiones*, 1(5), 140-144.
- 2021. Iván Torres IVE Consultores. 5 pasos para Realizar una Estandarización de Procesos en Tu Empresa.
- Sommerville, I. (2005). *Requerimientos del software*. Ingeniería del software, 7a ed., PEARSON EDUCACIÓN, Madrid, SPA, 109-110.

III. REFERENTES METODOLÓGICOS

El presente proyecto tiene como objetivo definir las características que satisfarán las necesidades la Agroveterinaria Mi Maskota, en base a los requerimientos presentados para buscar una solución a la necesidad de migrar sus procesos administrativos en el área de toda su gestión documental, y mejorar tiempos de respuesta en la gestión de pacientes, historiales de atención, administración de personal, entre otros.

El proyecto se desarrollará por un equipo de trabajo, involucrando áreas de la Ingeniería Informática, Analista Programador y Diseñador Gráfico.

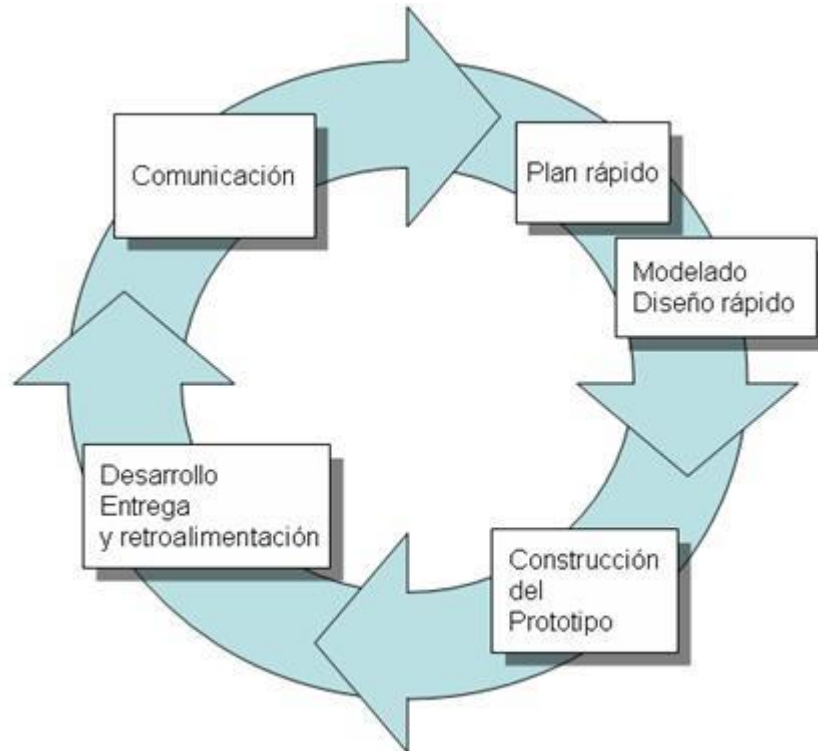
Se presentará en el siguiente informe, las áreas de factibilidad que posibilitaran la implementación efectiva del proyecto, por lo que se detallaran los costos, el tiempo necesario, el ámbito de la empresa, entre otros.

Además, y de acuerdo al planteamiento del problema, y la descripción de lo que se debe desarrollar, considero que el software que se debe implementar es de envergadura media, tomando en cuenta que se necesitan los siguientes módulos:

- Ingreso de Pacientes y datos de dueños de los animales.
- Creación de fichas de atención de los pacientes.
- Búsqueda y visualización de historiales de atención y medicación de los pacientes.
- Revisión de los profesionales y equipo técnico de atención a pacientes.
- Impresión de recetas de medicación y diagnósticos.

Considero que es un proyecto mediano, por lo que calificaría para aplicar el modelo de desarrollo evolutivo, ya que este tipo de modelo es para "*proyectos pequeños y medianos (aprox. 500.000 líneas de código)*". Además, considero que para el tipo de software que se está solicitando, podría desarrollarse de forma incremental, es decir se podría trabajar desde una "*implementación inicial y luego ir integrando implementaciones parciales de manera incremental*", hasta desarrollar e integrar todos los módulos solicitados por el cliente.

Creo también que las desventajas que tiene este modelo, como son la alta dinámica de desarrollo y estructura inestable no estaría presente en este caso, ya que, como lo mencione anteriormente la envergadura y complejidad son de término medio, por lo que no habría una alta complejidad y tampoco tantas entregas intermedias, como para generar costos altos o tareas difíciles de lograr.



3.1. DISEÑO DE PANTALLAS

- INGRESO DE PACIENTE

A Web Page

[Ingreso Paciente](#) [Ver Ficha Medica](#) [Ver Historial](#)

INGRESO FICHA PACIENTE

Nombre
 Especie
 Raza

DATOS MEDICO TRATANTE

Nombre
 RUT

- INGRESO FICHA DE CONTROL DE PACIENTE

A Web Page

INGRESO ATENCION A PACIENTE

Nombre
 Especie
 Raza

DETALLE DE ATENCION MEDICA

Peso Kg.

Diagnostico

Medicamentos Suministrados

- VER HISTORIAL DE ATENCIONES DEL PACIENTE

A Web Page

HISTORIAL DE ATENCION A PACIENTE

N°	NOMBRE	RAZA	FECHA	FAMILIAR	VER
1	PIPO	PERRO	01.11.2021	JUAN PEREZ	INGRESAR
2	LULU	GATO	05.09.2021	ANA MARIA	INGRESAR

3.2. ATRIBUTOS Y ENTIDADES

De acuerdo a la revisión de los enunciados de requerimiento se identifican las siguientes entidades y atributos:

- **Medico**

1. rut_medico
2. nombre_medico
3. apellido_medico
4. especialidad_medico

- **Auxiliar**

1. rut_auxiliar
2. nombre_auxiliar
3. apellido_auxiliar

- **Paciente**

1. id_paciente
2. nombre_paciente
3. raza_paciente

- **Familiar_Paciente**

1. rut_familiar
2. nombre_familiar
3. apellido_familiar
4. telefono_familiar
5. direccion_familiar

- **Ficha_Paciente**

1. id_ficha_paciente
2. fecha_nacimiento_paciente
3. raza_paciente

- **Historial_Paciente**

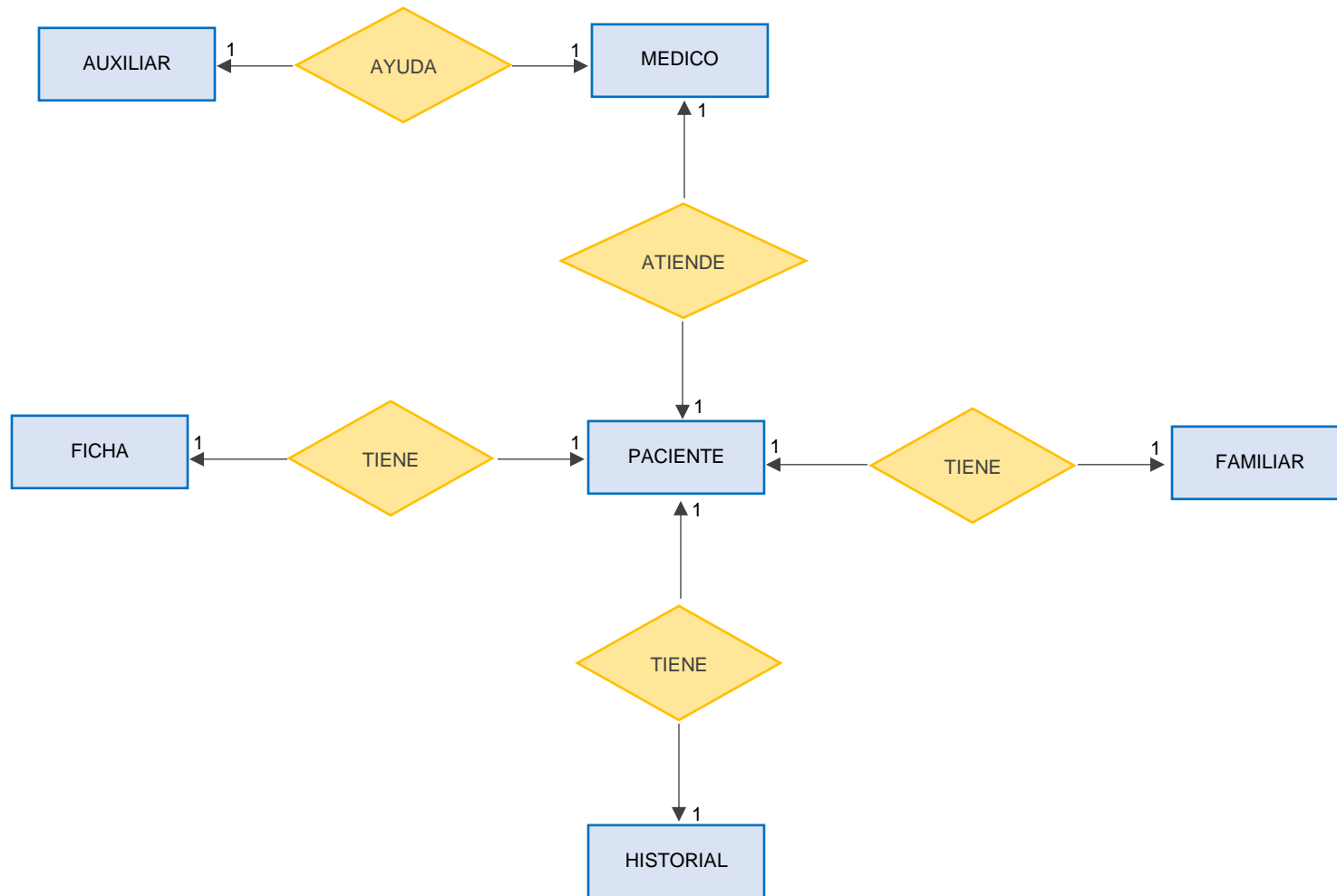
1. id_historial_paciente
2. fecha_control
3. peso_paciente
4. diagnostico_paciente
5. medicamento_suministrado
6. observaciones

3.3. ATRIBUTOS UNICOS

Para cada una de las entidades detalladas deberás identificar cuáles son los atributos únicos de identificación (Llave o *Primary key*).

N°	ENTIDADES	PRIMARY KEY
1	Medico	rut_medico
2	Auxiliar	rut_auxiliar
3	Paciente	id_paciente
4	Familiar_Paciente	rut_familiar
5	Ficha_Paciente	id_ficha_paciente
6	Historial_Paciente	id_historial_paciente

3.4. Modelo Entidad – Relación



3.5. DEFINICION DE ENTIDADES

MEDICO

NOMBRE ATRIBUTO	TIPO DE DATO	TAMAÑO	IDENTIFICADOR
rut_medico	INTEGER	15	PRIMARY KEY
nombre_medico	VARCHAR	30	
apellido_medico	VARCHAR	30	

AUXILIAR

NOMBRE ATRIBUTO	TIPO DE DATO	TAMAÑO	IDENTIFICADOR
rut_auxiliar	INTEGER	15	PRIMARY KEY
Nombre_auxiliar	VARCHAR	30	
apellido_apellido_auxiliar	VARCHAR	30	

PACIENTE

NOMBRE ATRIBUTO	TIPO DE DATO	TAMAÑO	IDENTIFICADOR
id_paciente	INTEGER	10	PRIMARY KEY
nombre_paciente	VARCHAR	30	
raza_paciente	VARCHAR	30	

FAMILIAR_PACIENTE

NOMBRE ATRIBUTO	TIPO DE DATO	TAMAÑO	IDENTIFICADOR
rut_familiar	INTEGER	15	PRIMARY KEY
nombre_familiar	VARCHAR	30	
apellido_familiar	VARCHAR	30	
teléfono_familiar	INTEGER	15	
dirección_familiar	VARCHAR	70	

FICHA_PACIENTE

NOMBRE ATRIBUTO	TIPO DE DATO	TAMAÑO	IDENTIFICADOR
id_ficha_paciente	INTEGER	15	PRIMARY KEY
fecha_nacimiento_paciente	VARCHAR	30	
raza_paciente	VARCHAR	30	

HISTORIAL_PACIENTE

NOMBRE ATRIBUTO	TIPO DE DATO	TAMAÑO	IDENTIFICADOR
id_historial_paciente	INTEGER	15	PRIMARY KEY
fecha_control	VARCHAR	30	
peso_paciente	VARCHAR	30	
diagnostico_paciente	VARCHAR	500	
medicamentos_suministrado	VARCHAR	200	
observaciones	VARCHAR	500	

IV. PLANIFICACIÓN Y DESARROLLO DEL PROYECTO

4.1. RECURSOS A UTILIZAR

Computador Personal: Este equipo servirá para realizar los informes de levantamiento inicial, además para llevar la documentación que se recopilará de forma ordenada. Además, servirá para la implementación del ambiente de programación si fuese necesario, una vez que se determine cual será, dependiendo del levantamiento de requerimientos.

Dispositivo de almacenamiento Externo: servirá para el transporte de documentación desde la institución y las personas designadas para las entrevistas, junto con ello se mantendrá en estos dispositivos externos toda documentación relativa al proyecto y que sea facilitada por los involucrados en la entrega de información.

Escáner: Servirá para digitalizar la documentación que se encuentre en papel, para obtener los formatos que actualmente se trabajan, para el ingreso de información por parte de la institución.

Impresora: servirá para imprimir los informes entregables, que se harán llegar al cliente, documentar el proyecto y presentar formatos preliminares y avances del proyecto.

Servidor Local: servirá para montar el software en la etapa de pruebas, ya que se tiene proyectado que la solución ofrecida se en entorno inicial a modalidad intranet.

Softwares de Producción: se adquirirán los softwares de SO, servidor web, Administrador de BD.

4.2. ANALISIS

De acuerdo al análisis para el desarrollo del sistema que se proyecta desarrollar, se propone que al determinar el beneficio que representará la implementación del proyecto, y considerando el extravío de la información, el consumo excesivo de papelería y el personal insatisfecho, creo que el beneficio no solo será en el aspecto económico, sino también en la capacidad de mejor gestión. Se adjunta tabla de costos asociados al recurso humano, para el desarrollo del proyecto.

4.3. IMPLMETACION

ETAPA	CANTIDAD DE PERSONAS	RECURSO	CANTIDAD DE HORAS	COSTO ROYECTADO
PLANIFICACION	1	Ingeniero Informático	45	351.000
	1	Analista Programador	45	234.000
	1	Administrativo	45	94.000
ANALISIS	1	Ingeniero Informático	35	276.000
	1	Analista Programador	45	234.000
DISEÑO	1	Ingeniero Informático	50	395.000
	1	Analista Programador	72	374.400
	1	Diseñador Grafico	72	338.400
IMPLEMENTACIÓN	1	Ingeniero Informático	15	118.500
	1	Analista Programador	27	140.400
PRUEBAS	1	Ingeniero Informático	15	118.500
	1	Analista Programador	27	140.400
AMBIENTE PRODUCCION	1	Ingeniero Informático	20	158.000
	1	Analista Programador	45	234.000
			TOTAL	3.206.600

CUADRO EXPLICATIVO VALOR EN PESOS CHILENOS	
TIPO DE RECURSO	VALOR HORA
Ingeniero Informático	7.900
Analista Programador	5.200
Diseñador Grafico	4.700
Administrativo	3.500

4.4. PRUEBAS

El plan de pruebas propuesto para el aseguramiento de calidad del producto final, es el siguiente:

PRUEBAS DE CONCEPTO

4.4.1. LIMITAR EL ALCANCE

Se realizarán pruebas de control de consultas, invitando a participar a un grupo de 50 pacientes(mascotas), de esta forma de controlar y recopilara información del comportamiento al momento de ingresar datos de los pacientes en las fichas digitales.

2. INFORMAR LOS AVANCES Y RESULTADOS A LOS STAKEHOLDER (INVERSIONISTAS)

Se mantendrá involucrado al cliente final (dueño de la veterinaria y usuarios finales), de los resultados obtenidos, a medida que el desarrollo del software avance, esto es para que puedan ir “construyendo” la idea de cómo será el producto final, esperando así obtener la retroalimentación necesaria para mejorar la calidad del software.

3. REUNIONES DIARIAS DE SEGUIMIENTO

Se realizarán reuniones semanales para mostrar documentación con el avance de las distintas etapas de desarrollo, lo que generara los documentos de conformidad y observaciones pertinentes.

4.4.2. PRUEBAS DE CONTENIDO

Las pruebas de contenido que se implementaran son las siguientes:

¿Es realmente precisa la información que se ofrece?

Se verificará que las fichas y formularios digitalizados estén claramente presentados en pantalla, así como los datos involucrados: nombre del dueño, rut del dueño, nombre mascota, edad mascota, raza, entre otros.

¿Es concisa y exacta la información mostrada en pantalla?

Se verificará que solo este la información relativa al proceso en pantalla, se realizaran revisiones para no solicitar datos redundantes, es decir si se solicitó un dato en un formulario que no se vuelva a solicitar en otro formulario.

¿Es fácil entender la plantilla de los usuarios?

Se revisará que la ficha de los pacientes y los datos asociados a esta, sean fácilmente identificables, entendible e intuitiva.

4.5. A MODO DE EJEMPLO SE REALIZARÁ LA SIGUIENTE PRUEBA DE CARGA

Considerare como primer ejemplo que se hará una prueba de carga, donde el objetivo es analizar si el sistema soporta 1.000 usuarios concurrentes (aclaración: no estoy teniendo en cuenta el tema de entorno de prueba, y qué tan similar es al de producción, y cómo definir el alcance en base a la diferencia de los ambientes).

4.5.1. PRIMERA PRUEBA

Un usuario no concurrente (esto puede servir como baseline, para comparar luego, puede ser con 1, 5, 10 o más, pero tiene que ser algo sumamente reducido para lo esperado en el sistema).

4.5.2. SEGUNDA PRUEBA

200 usuarios concurrentes (es decir, el 20% de la carga esperada). Desde este momento ya se puede obtener muchísima información sobre qué tan difícil va a ser completar la prueba, tanto en tiempo como en forma.

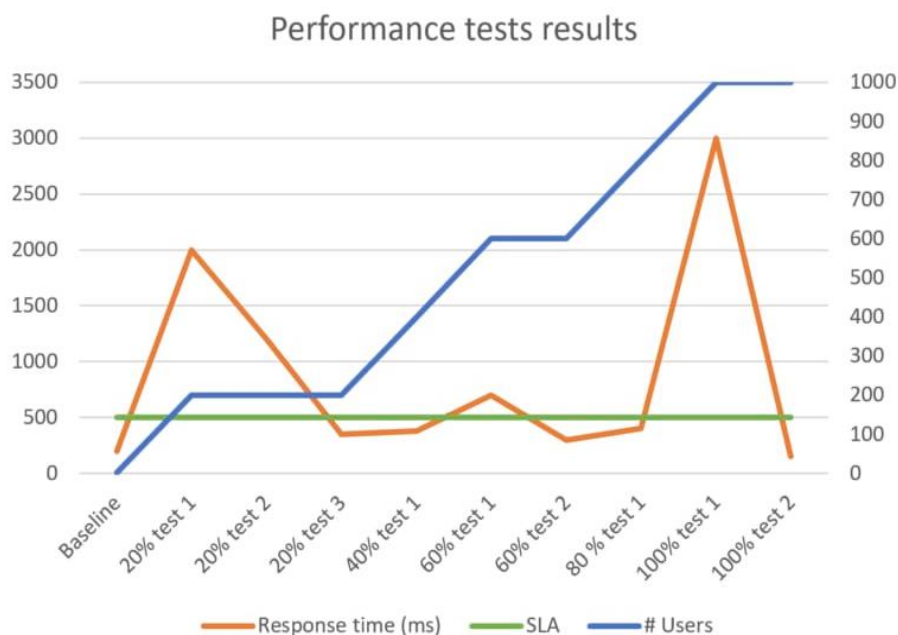
Al ejecutar estas primeras pruebas vamos a resolver los problemas más “gruesos”: las configuraciones por defecto (pools de conexión o tamaño del heap de Java por ejemplo). También vamos a tener una idea de cómo escala el sistema.

Una vez que se termina el análisis y resolución de problemas, se vuelve a ejecutar esta prueba hasta que se obtengan tiempos aceptables.

Según qué tan ajustados sean esos resultados se va a decidir si la tercera prueba será la del 40% (para seguir con incrementos de a 20) o si se hará con 50% de la carga (pensando en pasar luego al 75 y al 100), o si el sistema responde muy bien, quizá se pasará directamente a más.

En cualquier caso, lo que queremos tener al final es una gráfica que nos muestre los tiempos de respuesta obtenidos con cada prueba (con cada porcentaje de la carga esperada), y así podremos ver cómo fue evolucionando el sistema gracias a nuestro trabajo.

4.5.3. GRAFICA DE SIMULACIÓN Y EJEMPLO DE RESULTADOS EXTRAÍDO DE UNA SIMULACIÓN



En la anterior gráfica, se mostró cómo se ejecutaron distintas pruebas incrementando de a 20% la carga. Además, se puede observar que se repitieron las pruebas hasta alcanzar el acuerdo de nivel de servicio esperado (SLA o Service Level Agreement) en cada caso, y recién al alcanzarlo, se pasa al siguiente escalón.

4.5.4. PRUEBAS DE ESTRÉS

La siguiente prueba de Stress, tiene como finalidad probar el sistema y determinar donde se va más allá de la carga esperada para ver dónde se “rompe” el sistema. Se someterá al sistema a una carga más elevada de los límites solicitados de funcionamiento, se determinará el punto de ruptura del servicio y se analizarán las causas que tiene asociada.

Para eso se ejecutarán distintas pruebas, con distintas cantidades de usuarios, analizando si al aumentar la concurrencia sigue aumentando el throughput (si al aumentar la concurrencia no aumentara las transacciones por segundo, eso indicará que llegamos al punto de quiebre, ya que se estará saturando el sistema en algún punto, sin escalar).

Para ello, se ejecutará una prueba incremental de 0 a X, donde X es una cantidad de usuarios grande (digamos que 1.000 para considerar una generadora de carga sola) y que creemos que el quiebre tiene que estar en ese rango.

Lo que se puede hacer en cualquier herramienta de simulación de carga para ejecutar esta prueba es establecer un ramp-up (aumento de la velocidad de la prueba de carga) uniforme durante el tiempo de la prueba.

En otras palabras, si queremos que esa prueba dure una hora, hacemos una configuración para que la prueba comience con cero usuarios concurrentes, y al cabo de una hora tener 1.000. De esta manera, vamos a poder tener una primera aproximación hacia cuando el throughput del sistema se degrada. Si observamos que es alrededor de los 650 usuarios, ahí podemos comenzar a refinar ejecutando pruebas puntuales.

A modo de ejemplo, podríamos ejecutar una prueba con 500, otra con 600, otra con 700. Si efectivamente la prueba de 700 usuarios tiene menos throughput (tasa de transferencia efectiva) que la de 600, hay que refinar y ejecutar una con 650, y así seguimos con el punto medio, hasta mejorar la precisión.

4.5.5. PRUEBAS DE CAPACIDAD

Los siguientes ejemplos de pruebas de rendimiento, tienen como objetivo determinar el rendimiento del sistema bajo una carga de trabajo definida utilizando diferentes tipos de pruebas de rendimiento tales como pruebas de carga, de estrés y de estabilidad.

Tendremos en cuenta las consideraciones siguientes:

- Las prácticas recomendadas para realizar pruebas de rendimiento son las mismas que para las pruebas funcionales: probar en las fases iniciales y con frecuencia.
- Aprovecharemos los activos de prueba existentes creados en las pruebas funcionales.
- Utilizaremos analizadores para recopilar estadísticas de servidor, tales como utilización de CPU y de memoria.
- La instalación de agentes en varios sistemas para controlar la carga puede crear un perfil de generación de carga más realista y distribuido.
- Utilizaremos la acción "Registrar medida" para almacenar información adicional de servidores.
- Utilizaremos varios conjuntos de datos para combinar dos o más gráficos de rendimiento, tales como gráficos de tiempo de respuesta y gráficos de utilización de CPU.

4.5.6. PRUEBAS DE RENDIMIENTO BAJA INTENSIDAD

Durante la fase de evaluación, cuando ejecutemos pruebas funcionales a pequeña escala (pruebas de integración de unidad o componente), supervisaremos los tiempos de respuesta de transacciones individuales.

4.5.7. PRUEBAS DE RENDIMIENTO DE ALTA INTENSIDAD

Las pruebas de rendimiento de alta intensidad que se realizaran, se consideraran como pruebas de rendimiento instrumentadas. Esta fase de las pruebas se realiza en escenarios completos, tales como pruebas de estrés y pruebas de estabilidad realizadas en procesos de negocio globales.

4.5.8. VISUALIZAR RESTRICCIONES DEL RENDIMIENTO

Si el entorno de prueba consta de interacciones complejas entre componentes basadas en mensajes, las pruebas de rendimiento comprobarán que estos mensajes fluyen por el sistema a la velocidad necesaria. Para obtener una mejor comprensión de este concepto, es importante visualizar cómo los componentes procesan estos mensajes.

4.6. REQUERIMIENTOS FUNCIONALES

Ingreso de Nuevo Paciente	La aplicación deberá permitir el ingreso de nuevos datos de pacientes al sistema, asociándolos a un número único y datos personales
Actualización del Paciente	La aplicación deberá permitir la actualización de datos personales, no así del número único asociado.
Buscar Paciente	La aplicación deberá permitir la búsqueda de pacientes, ingresando: nombre, rut(dueño de la mascota), apellido(dueño de la mascota).
Ingreso de Ficha de Control de Pacientes	La aplicación deberá permitir el ingreso de nuevas fichas de control de pacientes al sistema, asociándolos a un número único.
Búsqueda Historial	La aplicación deberá permitir la búsqueda del historial de atención del paciente.
Impresión de Ficha	La aplicación deberá permitir la impresión de fichas de los pacientes, en donde aparecerá los datos asociados al animal.

Los requerimientos no funciones son los siguientes:

EFICIENCIA

- El sistema debe ser capaz de procesar 30 transacciones por segundo. Esto se medirá por medio de la herramienta SoapUI aplicada al Software Testing de servicios web.
- Toda funcionalidad del sistema y transacción de negocio debe responder al usuario en menos de 5 segundos.
- El sistema debe ser capaz de operar adecuadamente con hasta 100.000 usuarios con sesiones concurrentes.
- Los datos modificados en la base de datos deben ser actualizados para todos los usuarios que acceden en menos de 2 segundos.

SEGURIDAD LÓGICA Y DE DATOS

- Los permisos de acceso al sistema podrán ser cambiados solamente por el administrador de acceso a datos.
- El nuevo sistema debe desarrollarse aplicando patrones y recomendaciones de programación que incrementen la seguridad de datos.
- Todos los sistemas deben respaldarse cada 24 horas. Los respaldos deben ser almacenados en una localidad segura ubicada en un edificio distinto al que reside el sistema.
- Todas las comunicaciones externas entre servidores de datos, aplicación y cliente del sistema deben estar encriptadas utilizando el algoritmo RSA.
- Si se identifican ataques de seguridad o brecha del sistema, el mismo no continuará operando hasta ser desbloqueado por un administrador de seguridad

OTROS REQUERIMIENTOS NO FUNCIONALES

- El sistema será desarrollado para las plataformas PC y Macintosh.
- La aplicación debe ser compatible con todas las versiones de Windows, desde Windows XP.
- La aplicación deberá consumir menos de 500 Mb de memoria RAM.
- La aplicación no podrá ocupar más de 2 GB de espacio en disco.
- La nueva aplicación debe manejar fuentes del alfabeto en Inglés, Idiomas latinos (Español, Frances, Portugués, Italiano), Árabe y Chino.
- La interfaz de usuario será implementada para navegadores web únicamente con HTML5 y JavaScript.

USABILIDAD

- El tiempo de aprendizaje del sistema por un usuario deberá ser menor a 4 horas.
- La tasa de errores cometidos por el usuario deberá ser menor del 1% de las transacciones totales ejecutadas en el sistema.
- El sistema debe contar con manuales de usuario estructurados adecuadamente.
- El sistema debe proporcionar mensajes de error que sean informativos y orientados a usuario final.
- El sistema debe contar con un módulo de ayuda en línea.
- La aplicación web debe poseer un diseño "Responsive" a fin de garantizar la adecuada visualización en múltiples computadores personales, dispositivos tableta y teléfonos inteligentes.
- El sistema debe poseer interfaces gráficas bien formadas.

