Sistema de GESTION DE VETERINARIA

Plan de Desarrollo de Software

Versión 0.9

1.- **INTRODUCCION. -**

El proyecto ha sido solicitado por Fernando Montecinos dueño de la cadena de veterinarias PET-LAPAZ.

Estará basada en el tipo de desarrollo de software metodología ágil programación Extrema o XP

2.- **ANTECEDENTES. –**

El presente proyecto está desarrollado para el uso del personal de atención de una cadena de veterinarias, con el propósito de automatizar y mejorar el registro y administración de los datos de mascotas que serán atendidas, tomando como datos principales información del dueño y de la mascota, puesto que el manejo de datos en libros de registros escritos manualmente es un poco problemático y además moroso.

3.- **OBJETIVOS. –**

3.1.- **OBJETIVO GENERAL**. -

Desarrollar e implementar un sistema de SOFWARE para el desenvolvimiento del personal de la cadena de veterinarias, el cual permitirá contar con una adecuada, organización, un control más eficiente y optimización del tiempo en los procesos de administración, con el fin de contar con información oportuna.

3.1.- **OBJETIVOS ESPECIFICOS. –**

* Determinar los requerimientos del sistema registro de mascotas.
* Validar el sistema realizando la prueba registrando mascotas.

4.- **ALCANCE.** –

El sistema de gestión de la cadena de veterinarias PET-LAPAZ será desarrollado por el grupo de desarrollo de software.

El sistema permitirá al personal controlar todo lo relativo a la gestión de registro de mascotas para programar citas y servicios a las diferentes mascotas que necesitan atención. Además, también permitirá a los encargados de la veterinaria tener todo tipo de reportes.

EL plan de desarrollo de software, describe el plan global usado para el desarrollo del “Sistema de gestión de veterinarias.” Para este efecto se basa en la captura de requisitos por parte de las partes interesadas para realizar una estimación de recursos aproximada, una vez comenzando el proyecto y posterior avance del mismo. En caso de nuevas iteraciones esto ocasionara un nuevo versiona miento del documento.

AI desarrollar un sistema de información de control de una cadena de veterinarias como ser PET-LAPAZ, al entrar a un modelo de negocio de comercio esto llevara una adaptación de sistemas de tecnológicos que sean rápidas, seguras y sustentables.

El programa debe cumplir con los requerimientos del sistema, para eso debe cumplir con los subsistemas o módulos implicados en la gestión de control de sistema de veterinaria.

1. Desarrollar módulo de control de gestión de la cadena de veterinarias PET-LAPAZ.

* Gestión de información general de la cadena de veterinarias es decir de las 2 sucursales.
* Desarrollar módulo de información de facturación (a cargo de cajero).

5.- **ANALISIS**. -

**RESUMEN DE REQUERIMIENTOS**

Requerimientos de Sistema de gestión de veterinaria.

Según el diagnóstico realizado, se registra la información de la mascota, información del dueño, doctor que atenderá a la mascota que consiste en:

-Nombre de mascota

-Raza

-Color

-Nombre Dueño

- Nro. Carnet de Identidad de dueño

-Domicilio

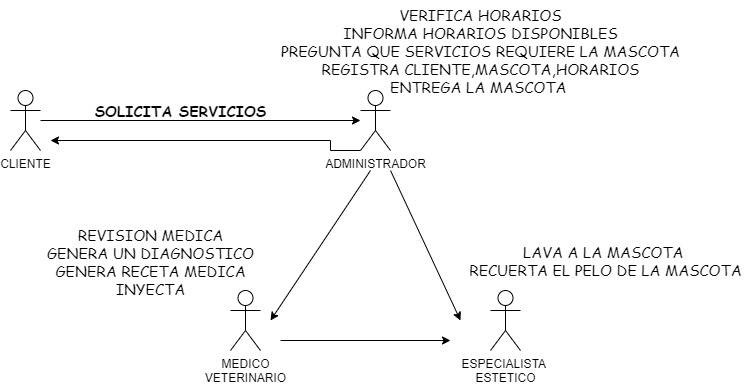
-Nro. de teléfono

-Fecha y hora de reserva

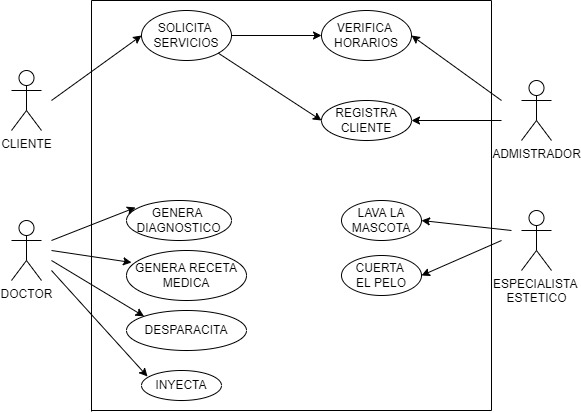
-Peluquero si es necesario.

* Gestión de veterinarias PET-LAPAZ para 2 sucursales
* Registro de empleados, datos de clientes registro.
* Debe tener sistema facturación computarizada, con código de control o nuevo código qr.
* Cliente debe poder ver todas las gestiones de operación de las 2 cadenas de veterinarias.
* Los empleados solo tienen la posibilidad de ver los datos del cliente, hora y fecha de atención, costo de la atención.
* El empleado solo puede ver información pertinente a los datos del cliente que fueron registrados.

**DIAGRAMA DE PROCESOS. -**

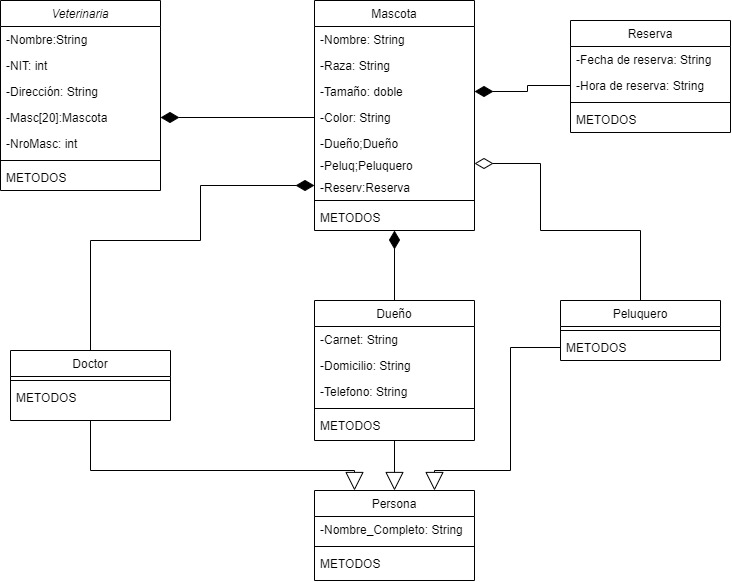


**DIAGRAMAS DE CASOS DE USO. –**



6.- **DISEÑO**. -

**DIAGRAMA DE CLASES. -**



7.- **TECNOLOGIA**. –

**FRAMEWORK**. –



**Spring Boot** es una tecnología que nos permite crear aplicaciones autocontenidas, con esto nos podemos olvidar de la arquitectura y enfocarnos únicamente en desarrollo, delegando a Spring Boot labores como configuración de dependencias, desplegar nuestro servicio o aplicación a un servidor de aplicaciones y enfocarnos únicamente en crear nuestro código.

Para esto Spring Boot utiliza internamente un servidor de aplicaciones embebido, por defecto utiliza Tomcat, pero también lo podemos hacer con jetty o undertow.

Y no solo esto, Spring Boot también nos provee un completo gestor de dependencias como maven o gradle, configuraciones automáticas y mucho más para que nuestra aplicación quede a la medida.

Spring Boot facilita la creación de aplicaciones independientes basadas en Spring de grado de producción que puede "simplemente ejecutar".

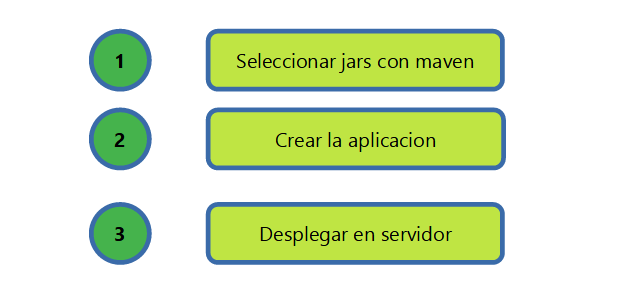
**Existen tres pasos a realizar.**

Primero se crea un proyecto Maven/Gradle y se descargan las dependencias necesarias.

En segundo lugar, se desarrolla la aplicación.

Tercero se despliega en un servidor.

Si nos ponemos a pensar un poco a detalle en el tema, **únicamente el paso dos es una tarea de desarrollo**. Los otros pasos están más orientados a infraestructura. No deberíamos tener que estar eligiendo continuamente las dependencias y el servidor de despliegue.



Básicamente, esto significa que es posible poner en funcionamiento una aplicación de Spring con muy poca configuración. La poca configuración que se necesita está en forma de anotaciones, y si se necesita, casi nada de XML.

Si bien es cierto que Spring Framework es muy potente, **la configuración inicial y la preparación de las aplicaciones para producción**son tareas bastante tediosas. **Spring Boot simplifica el proceso al máximo**gracias a sus dos principales mecanismos.

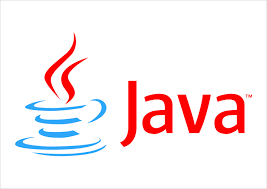
**BASE DE DATOS**. –



DBeaver permite la mayoría de funcionalidades básicas de cualquier gestor de bases de datos y mucho más. A través de su amigable interfaz podemos:

* Crear todos los componentes de una base de datos: esquemas, tablas, disparadores, funciones, usuarios, roles, etc.
* Realizar consultas SQL y NoSQL
* Crear/Modificar/Eliminar registros
* Exportar y migrar datos
* Generar backups
* Generar datos simulados para realizar pruebas
* Crear diagramas del modelo entidad-relación
* Visualización de información espacial

**LENGUAJE DE PROGRAMACION**. –



Java es una plataforma informática de lenguaje de programación creada por Sun Microsystems en 1995. Ha evolucionado desde sus humildes comienzos hasta impulsar una gran parte del mundo digital actual, ya que es una plataforma fiable en la que se crean muchos servicios y aplicaciones. Los nuevos e innovadores productos y servicios digitales diseñados para el futuro también siguen basándose en Java.

Aunque la mayoría de aplicaciones Java modernas combinan el tiempo de ejecución y la aplicación de Java, todavía existen algunas aplicaciones e incluso sitios web que no funcionan sin instalar Java para escritorio. El sitio web Java.com está pensado para consumidores que todavía necesitan Java en sus aplicaciones de escritorio, sobre todo las aplicaciones que tienen como destino Java 8. Los desarrolladores y los usuarios que quieran aprender sobre programación Java, deben consultar el sitio web [dev.java](https://dev.java/) y los usuarios profesionales, oracle.com/java para obtener más información.



JavaScript es un lenguaje de programación que los desarrolladores utilizan para hacer páginas web interactivas. Desde actualizar fuentes de redes sociales a mostrar animaciones y mapas interactivos, las funciones de JavaScript pueden mejorar la experiencia del usuario de un sitio web. Como lenguaje de scripting del lado del servidor, se trata de una de las principales tecnologías de la World Wide Web. Por ejemplo, al navegar por Internet, en cualquier momento en el que vea un carrusel de imágenes, un menú desplegable “click-to-show” (clic para mostrar), o cambien de manera dinámica los elementos de color en una página web, estará viendo los efectos de JavaScript.

**EDITOR DE CODIGO**. –



IntelliJ IDEA, el IDE insignia de JetBrains, está diseñado para el desarrollo productivo de JVM. Con su conocimiento profundo de lenguajes y tecnologías, y su interfaz de usuario ergonómica, IntelliJ IDEA hace que el desarrollo sea una experiencia agradable.

**MANEJADOR DE DEPENDENCIAS**. –



**Apache Maven** es una potente herramienta de gestión de proyectos que se utiliza para gestión de dependencias, como herramienta de compilación e incluso como herramienta de documentación. Es de código abierto y gratuita.

Aunque se puede utilizar con diversos lenguajes (C#, Ruby, Scala...) se usa principalmente en proyectos **Java**, donde es muy común ya que está escrita en este lenguaje. De hecho, frameworks Java como **Spring** y **Spring Boot** la utilizan por defecto, por lo que conviene conocerla si trabajas en la plataforma Java y, desde luego, con Spring.

8.- **LISTADO DE SERVICIOS. -**