



VETERIWEB

BY

 WELASIGN

Versión: 0100

Fecha: 13/07/2018

INTRODUCCION

OBJETO

El propósito de este documento es el dar a conocer los avances de nuestro proyecto “Veteriweb” el cual es un sitio web de comercio electrónico en el cual personas pueden hacer citas para llevar a sus mascotas a la veterinaria de la que sea la página en específico.

ALCANCE

Se busca con este proyecto el mejorar las ventas de la veterinaria agilizando las reservaciones mediante el uso de una plataforma web para que los clientes tengan otra opción que no sea solo por teléfono y presencial

OBJETIVOS ESPECIFICOS

- Entrevistar a la dueña para conocer sus necesidades.
- Mostrar la visión de la empresa al público en la página web.
- Mostrar los servicios que ofrece la empresa.
- Mostrar testimonios de clientes satisfechos.
- Facilitar el negocio tanto para el usuario como para el propietario.
- Hacer una interfaz que sea fácil de usar para todo tipo de personas.

CONCEPTOS APLICABLES EN NUESTRO PROYECTO

Base de datos

Una base de datos o banco de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.

UI

El diseño UI podría definirse como una nueva tendencia en el diseño web que se enfoca en crear unas interfaces entre las personas y los diferentes dispositivos digitales.

UX

Lo que el cliente percibe al usar o probar un producto y una forma de ver estos productos de manera integral desde el punto de vista de un usuario que probablemente no le importa cómo se hacen esos productos, sólo el producto en sí mismo

TIPOS DE DIAGRAMAS UML

DIAGRAMAS DE CASOS DE USO

Los Casos de Uso no forma parte de la llamada Fase de Diseño, sino parte de la fase de Análisis, respondiendo el interrogante ¿Qué? De forma que al ser parte del análisis ayuda a describir que es lo que el sistema debe hacer.

Estos diagramas muestran operaciones que se esperan de una aplicación o sistema y como se relaciona con su entorno, es por ello que se ve desde el punto de vista del usuario. Describen un uso del sistema y como éste interactúa con el usuario.

Los casos de usos se representan en el diagrama por una elipse la cual denota un requerimiento solucionado por el sistema.

El conjunto de casos de usos representa la totalidad de operaciones que va a desarrollar el sistema. Por último, a estas elipses lo acompaña un nombre significativo de manera de rótulo.

Otro elemento fundamental de estos diagramas son los actores la cual representa a un usuario del sistema, que necesita o interactúa con algún caso de uso, la que también es acompañado por un nombre. Por último tenemos los flujos de eventos que corresponde a la ejecución normal y exitosa del caso de uso.

DIAGRAMA DE CLASES

En UML el diagrama de clases es uno de los tipos de diagramas o símbolo estático y tiene como fin describir la estructura de un sistema mostrando sus clases, atributos y relaciones entre ellos.

Estos diagramas son utilizados durante el proceso de análisis y diseño de los sistemas informáticos, en donde se intentan conformar el diagrama conceptual de la información que se manejará en el sistema

Los diagramas de clases tienen las siguientes características:

- Las clases define el ámbito de definición de un conjunto de objetos.
- Cada objeto pertenece a una clase.
- Los objetos se crean por instanciación de las clases.

DIAGRAMA DE OBJETOS

Forma parte de la vista estática del sistema. En este diagrama se modelan las instancias de las clases del Diagrama de Clases. Este diagrama cabe aclarar que cuenta con objetos y enlaces. En estos diagramas también es posible encontrar las clases para tomar como referencia su instanciación.

En otras palabras, el Diagrama de Objetos muestra un conjunto de objetos y sus relaciones en un momento concreto. Los Diagramas de Objetos son realmente útiles para modelar estructuras de datos complejas

DIAGRAMAS DE COMPORTAMIENTOS

Diagrama de Estados

Un estado es una condición durante la vida de un objeto, de forma que cuando dicha condición se satisface se lleva a cabo alguna acción o se espera por un evento.

El estado de un objeto se puede caracterizar por el valor de uno o varios de los atributos de su clase, además, el estado de un objeto también se puede caracterizar por la existencia de un enlace con otro objeto.

El diagrama de estados engloba todos los mensajes que un objeto puede enviar o recibir, en otras palabras, es un escenario que representa un camino dentro de un diagrama.

Como característica de estos diagramas siempre cuentan con dos estados especiales, el inicial y el final, con la particularidad que este diagrama puede tener solo un estado inicial pero varios estados finales.

Una transición entre estados representa un cambio de un estado origen a un estado sucesor destino que podría ser el mismo que el estado origen, dicho cambio de estado puede estar

aparejado con alguna acción. Además, las acciones se asocian a las transiciones y se consideran que ocurre de forma rápida e interrumpible.

Los elementos que componen estos diagramas son:

- Círculo lleno, apuntando el estado inicial.
- Círculo hueco que contiene un círculo lleno más pequeño en el interior, indicando el estado final.
- Rectángulo redondeado dividido por una línea horizontal, indicado los estados, en la parte de arriba se encuentra el nombre del estado y abajo se indica la actividad que realiza.
- Flecha, la cual denota la transición, el nombre del evento que causa esta transición etiqueta el cuerpo de la flecha.

Diagrama de actividad

Un Diagrama de Actividades representa un flujo de trabajo paso a paso de negocio y operacionales de los componentes en un sistema.

En UML 1, un diagrama de actividades es una variación del Diagrama de Estados UML donde los estados representan operaciones y las transiciones representan las actividades que ocurren cuando la operación es completa.

En la actualidad, el diagrama de actividades en UML 2.0 es similar al aspecto del diagrama en UML 1, solo que ahora la semántica está basada en lo que se conoce como Redes de Petri. En UML 2.0, el diagrama general de interacción está basado en el diagrama de Actividad.

Componentes:

- Inicio: el inicio de un diagrama de actividades es representado por un círculo de color negro sólido.
- Actividad: Una actividad representa la acción que será realizada por el sistema la cual representa dentro de un óvalo.
- Transición: Una transición ocurre cuando se lleva acabo el cambio de una actividad a otra, la transición es representada simplemente por una línea con una flecha en su terminación para indicar su dirección.

DIAGRAMA DE INTERACCION

Diagrama de Secuencia

Un Diagrama de Secuencias muestra una interacción ordenada según la secuencia temporal de eventos y el intercambio de mensajes. Los diagramas de secuencia ponen especial énfasis en el orden y el momento en el que se envían los mensajes a los objetos.

En los diagramas de Secuencias los elementos están representados por líneas intermitentes verticales, con el nombre del objeto en la parte más alta.

Los mensajes pueden ser o bien síncronos, el tipo normal de llamada del mensaje donde se pasa el control a objeto llamado hasta que el método finalice, o asíncronos donde se devuelve el control directamente al objeto que realiza la llamada.

Los mensajes síncronos tienen una caja vertical en un lateral del objeto que muestra el flujo del control del programa.

Diagrama de Colaboración

Un diagrama de colaboración, se puede decir que es una forma alternativa al diagrama de secuencias a la hora de mostrar un escenario.

Este tipo de diagrama muestra las interacciones que ocurren entre los objetos que participan en una situación determinada.

A diferencia del diagrama de secuencia, el diagrama de colaboración se enfoca en la relación entre los objetos y su topología de comunicación.

En estos diagramas los mensajes enviados de un objeto a otro se representan mediante flechas, acompañado del nombre del mensaje, los parámetros y la secuencia del mensaje.

Estos diagramas están indicados para mostrar una situación o flujo de programa específico y son considerados uno de los mejores diagramas para mostrar o explicar rápidamente un proceso dentro de la lógica del programa

DIAGRAMA DE IMPLEMENTACION

Diagrama de componentes

Lo que distingue el Diagrama de Componentes de otro tipo de diagramas es sin duda su contenido. Normalmente contiene componentes, interfaces y relaciones entre ellos. Los componentes pertenecen a un mundo físico, es decir, representan a un bloque de construcción al modelar aspectos físicos de un sistema.

Cada componente debe tener un nombre que lo distinga de los demás. Al igual que las clases los componentes pueden enriquecerse con compartimientos adicionales que muestran sus detalles.

Diagrama de Despliegue

Básicamente este tipo de diagrama se utiliza para modelar el Hardware utilizado en la implementación del sistema y la relaciones entre sus componentes.

Los elementos usados por este tipo de diagrama son nodos, componentes y asociaciones. En el UML 2.0 los componentes ya no están dentro de nodos, en cambio puede haber artefactos (archivo, un programa, una biblioteca o Base de datos) u otros nodos dentro de nodos.

Además, los Diagramas de Despliegue muestran la configuración en funcionamiento del sistema incluyendo su software y su hardware. Para cada componente de un diagrama es necesario que se deba documentar las características técnicas requeridas, el tráfico de red, el tiempo de respuesta, etc.

BASE DE DATOS

```
1 • CREATE DATABASE veteriweb;
2 • USE veteriweb;
3
4 • CREATE TABLE perroTipo
5   (
6     IdPerroTipo INT(11) NOT NULL AUTO_INCREMENT,
7     Descrpccion VARCHAR(45) NOT NULL,
8     PRIMARY KEY(IdPerroTipo)
9   );
10
11 • CREATE TABLE clientes
12   (
13     IdClientes INT(11) NOT NULL AUTO_INCREMENT,
14     Nombre VARCHAR(11) NOT NULL,
15     Apaterno VARCHAR(11) NOT NULL,
16     Amaterno VARCHAR(11) NULL,
17     Correo VARCHAR(45) NOT NULL,
18     Telefono INT(14) NOT NULL,
19     Clave VARCHAR(45) NOT NULL,
20     PRIMARY KEY(IdClientes)
21   );
22 • CREATE TABLE reservacion
23   (
24     IdReservacion INT(11) NOT NULL AUTO_INCREMENT,
25     Cliente INT(11) NOT NULL,
26     Perro INT(11) NOT NULL,
27     PRIMARY KEY(IdReservacion),
28     FOREIGN KEY(Cliente) REFERENCES clientes (IdClientes)
29     ON UPDATE CASCADE
30     ON DELETE CASCADE,
31     FOREIGN KEY(Perro) REFERENCES perro (IdPerroTipo)
32     ON UPDATE CASCADE
33     ON DELETE CASCADE
34   );
```