

Proyecto: Aplicativo web para veterinarias

Contexto

Su equipo ha sido contratado por una clínica veterinaria que desea poder comenzar a digitalizar su negocio. El cliente desea una página web que promocióne sus servicios, permita a clientes estar atentos al estado de sus mascotas y al personal de la veterinaria registrar las atenciones médicas que realizan.

Negocio

La clínica veterinaria se especializa en gatos que necesitan ser hospitalizados, estando en reposo y tratamiento en las instalaciones físicas del negocio. Los clientes, son los dueños de las mascotas los cuales al llegar al establecimiento por primera vez dan algunos datos personales¹ (cedula, nombre, correo, celular), posteriormente uno de los veterinarios toma otras medidas de la mascota (nombre, raza, edad, peso, enfermedad y una foto) y la registra en el sistema. Tras esto la mascota se queda algunos días en el hospital donde los veterinarios la cuidan y administran tratamientos hasta que se recupera.

Lo ideal es que el personal veterinario a través de la misma página web pueda ir registrando las atenciones que realizan. Se espera que estos puedan tener un usuario con los siguientes datos:

- Cedula
- Contraseña
- Especialidad
- Foto (URL)
- Numero de atenciones
- Nombre

Los médicos veterinarios administran tratamientos a las mascotas que están hospitalizadas. Cada tratamiento consta de una droga que un veterinario da a una mascota en una fecha en específico. **Toda droga consta de nombre, precio de compra, precio de venta, unidades disponibles y unidades vendidas** (esta base de datos actualmente está en un Excel). Todos los tratamientos/atenciones deben quedar guardados en el sistema.

¹ Es el veterinario quien registra tanto a la mascota como al cliente en el sistema.

Finalmente, el dueño del negocio desea un portal de administrador donde pueda analizar de manera sencilla los datos de su negocio: cantidad de mascotas en tratamiento, cantidad de atenciones que se han realizado, precio total de los tratamientos. Además, a través de este portal podrá agregar, eliminar o modificar a los médicos veterinarios.

Adicionales

- Cuando una mascota llega a la veterinaria su enfermedad puede ser “vacío”, mientras es atendida por algún médico.
- Un veterinario puede tratar a varias mascotas y una mascota puede ser tratada por diferentes veterinarios. La relación entre un veterinario y una mascota está data por los tratamientos suministrados.
- Un dueño puede tener varias mascotas registradas en el hospital.
- Cuando una mascota se cura y el dueño la recoge la información de esta no debe ser borrada del sistema. La mascota debe pasar a un estado “inactivo”.
- Puede suponer que la mascota solo sufre una enfermedad a la vez.
- No usar herencia.
- Inicialmente no es necesario guardar un historial de cómo ha variado el estado de la mascota (peso, enfermedad) sin embargo si es necesario que se guarde el historial de tratamientos realizados.

Requerimientos

- **Feature 01:** Diagrama de Clases, entidad relación y mockups + navegación.

AC1	Given la información del sistema Then generar un diagrama de clases y entidad relación para cada uno de los componentes
AC2	Given la información del sistema Then generar un diagrama entidad-relación. Este debe mostrar PK, FK, campos que pueden ser nulos, cardinalidad, campos únicos.
AC3	Given la información del sistema Then generar los mockups del diseño de la página. And Cada mockup debe estar asociado a una URL And los mockups se deben poder navegar de manera interactiva o mostrar cómo sería la navegación. And diferenciar a cuáles pantallas puede acceder cada uno de los tipos de usuario
AC4	Given la información del negocio Then crear un logo And crear un nombre
AC5	Given la información del negocio The definir una paleta de colores

	https://www.youtube.com/watch?v=yYwEnLYT55c (No es necesario usar HSB, ni tener una paleta compleja, lo importante es definir los colores de su aplicación, variaciones, y colores neutrales)
--	---

- **Feature 02:** Repositorio – Github

AC6	Given código generado durante el desarrollo Then crear un repositorio en Github con código de backend y frontend. And generar una rama de “desarrollo” y otra “producción” o “master”.
AC7	Given los entregables del feature 1 Then cargar los entregables en el repositorio.
AC8	Given los entregables del feature 1 Then personalizar el repositorio con información del negocio y del desarrollo

- **Feature 03:** Base de datos

AC9	Given la información de cada componente Then generar las entidades en el sistema y en la BD.
AC10	Given cada una de las entidades Then generar repositorios
AC11	Given la información de cada componente Then generar datos de prueba que permitan simular el funcionamiento real de la página. ²

- **Feature 04:** Landing page.

AC12	Given la información del negocio y los diseños generados Then generar una landing page para promocionar la veterinaria And permitir navegar hacia el resto de las funcionalidades o portales.
------	--

- **Feature 05(dueño/cliente):** Mostrar mascotas en tratamiento.

AC13	Given la cedula de un cliente. When el cliente está registrado. Then este podrá ver el estado de sus mascotas registradas.
------	---

² Para la entrega 1, solo es necesario cargar datos relacionados a mascotas y dueños.

Se desea un portal donde los veterinarios puedan registrar y actualizar mascotas, clientes y los tratamientos suministrados a las mascotas.

- **Feature 06(portal veterinario):** CRUD dueño

AC14 CREATE	Given la información del dueño When en dueño no está registrado Then pedir los datos And crear la información en la base de datos
AC15 READ	Given la cedula del dueño When el dueño está registrado Then mostrar datos del dueño And mostrar información de mascotas del dueño
AC16 UPDATE	Given la cedula de un dueño Then poder actualizar los datos de este en el sistema

- **Feature 07(portal veterinario):** CRUD mascota

AC17 UPDATE	Given el nombre de una mascota Then poder actualizar los datos de esta en el sistema
AC18 SEARCH	Given el nombre de una mascota Then poder buscar una mascota and mostrar su información and (optional) mostrar información del dueño
AC19 READ	Given la información de la base de datos Then mostrar todas las mascotas de la veterinaria
AC20 CREATE	Given la información de la mascota y el dueño When mascota no ha sido registrada Then usar esta información para crear una nueva mascota

Portal de administrador

- **Feature 08:** Registrar un veterinario(ADMINISTRADOR)

AC21 CREATE	Given la información del veterinario Then registrarlo en la base de datos
AC22 UPDATE	Given la cedula de un veterinario Then poder actualizar los datos de este en el sistema
AC23 UPDATE	Given la cedula de un veterinario Then cambiar el estado del veterinario a desactivado

- **Feature 09:** dashabord/tablero de datos(ADMINISTRADOR)

AC24	Given los datos de la página Then Crear un dashboard que presente diferentes KPIs <ul style="list-style-type: none"> • Cantidad total de tratamiento administrados en el último mes
------	---

	<ul style="list-style-type: none"> • Cantidad de tratamientos por tipo de medicamento administrado en el último mes (tabla medicamento - cantidad) • Cantidad de veterinarios activos en la plataforma • Cantidad de veterinarios inactivos en la plataforma • Cantidad de mascotas totales en la veterinaria • Cantidad de mascotas activas (en tratamiento) en la veterinaria • Ventas totales de la veterinaria • Ganancias totales de la veterinaria • Top 3 tratamientos con más unidades vendidas
--	---

• **Feature 09:** Cargar medicamentos(INICIALIZACIÓN)

AC24	Given la información de los medicamentos desde un archivo Then cargar la información de los medicamentos cuando inicia el programa
------	---

• **Feature 10:** Dar tratamiento a mascota (VETERINARIO)

AC25 CREATE	Given una mascota, un veterinario, y un medicamento desde una lista desplegable And la fecha actual When Unidades disponibles > 0 Then crear un tratamiento para la mascota And modificar unidades disponibles y unidades vendidas
----------------	--

• **Feature 11:** Ver historial medico(VETERINARIO)

AC26 READ	Given el nombre de una mascota Then mostrar la lista de tratamientos dados
AC27 DELETE	Given la lista de tratamientos Then eliminar un tratamiento

• **Feature 12:** Crear roles

AC28	Given la información del usuario administrador Then crear usuario de tipo veterinario
AC29	Given la información del usuario veterinario Then crear usuario de tipo dueño
AC30	Given la información del usuario veterinario Then crear mascota

• **Feature 13:** Implementar control de acceso basado en roles

Roles/Funcionalidades	Registrar Mascota	Registrar Veterinario	Dar tratamiento a mascota	Ver historial médico con permisos de lectura y escritura	Ver historial médico con permisos de lectura	Ver listado de todas las mascotas	Ver listado de mascotas de un dueño
ADMIN	X	X	X	X	X	X	X
VETERINARIO	X		X	X	X	X	X
DUEÑO					X		X

AC31	Given la información de roles y funcionalidades Then implementar el control de acceso para las funcionalidades
------	---

Entregas

- Primera Entrega
 - Features: 1 al 7
 - Video explicativo de la página a nivel funcional mostrando persistencia de los datos. (10 minutos máximo)
- Segunda Entrega
 - Features: 8 a 11
 - SPA
 - Video explicativo de cada una de las features mostrando la persistencia en la BD. (10 minutos máximo)
- Tercera Entrega
 - Features: 12 a 13
 - Pruebas automatizadas a nivel del cliente web con al menos un 50% de coverage
 - Pruebas automatizadas a nivel del servidor con al menos un 50% de coverage
 - Las pruebas automáticas deben cargar datos de prueba en la base de datos, los cuales deben removerse después de cada prueba
 - Video explicativo de cada una de las features y como fueron implementadas

Calificación

La nota de los grupos se dividirá en 2 partes (2.5 unidades C/U). La primera será la calificación del cliente, que se calculará a partir de la entrega de cada Sprint. La

segunda será la calificación del product manager el cuál revisa que todos los requerimientos a nivel funcional de la página se cumplan.

Calificación del cliente:

Cada semana se realizarán entre una y dos entregas en formato de sprint donde los grupos de trabajo deben mostrar adelantos FUNCIONALES de su trabajo. Al final de cada Sprint los grupos deben enviar un video donde se muestre en menos de 5 minutos los avances funcionales del proyecto (no es necesario que se muestre código excepto que se pida).

El cliente se fija de manera general en temas funcionales, UX, UI y CX, para dar su calificación por sprint.

Al final de semestre la calificación del cliente aporta para distinguir cuál ha sido el mejor proyecto durante todo el curso.

Calificación del Product owner:

Este revisará al final de cada corte a nivel funcional y de código como va el producto general a entregar. Se usará un enfoque de calificación “substractiva” donde cada uno de los grupos empieza con las 2.5 unidades y por cada error cometido se van substrayendo decimas de la nota final.

Algunas de las penalizaciones son:

- Código no comentado
- Malos nombres de variables
- No hay validación de los formularios
- Problemas de rendimiento de la página
- No se cumplen con los requisitos funcionales propuestos
- Código duplicado (puede ser por mal uso de funciones o de componentes)
- No usar los patrones de diseño mostrados en clase
- Mal diseño (de base de datos, diagrama de clases, navegación)
- Usar paquetes para organizar el proyecto de manera coherente
- Usar recursos de alta calidad donde no es necesario