INFORME DE CUMPLIMIENTO DE LA PRÁCTICA DE INGENIERÍA DE SOFTWARE I

VETTRACK

REQUISITOS QUE SI SE HAN LLEGADO A IMPLEMENTAR

1. Las mascotas deben estar asociadas a un dueño
2. Los clientes pueden ver su historial de citas y compras
3. Los administradores pueden dar nuevas citas a los clientes
4. Tener un listado de productos a la venta
5. Se puede eliminar a un cliente de la base de datos
6. Los administradores pueden registrar más de una mascota bajo el nombre de un propietario
7. Interfaz de inicio de sesión
8. Interfaz distinta cliente/administrador
9. Conexión de la aplicación con la base de datos
10. Una base de datos en la nube
11. Lenguaje de programación elegido “Java”
12. Una conexión entre base de datos usando otro lenguaje de programación

REQUISITOS QUE NO HEMOS IMPLEMENTADO

1. Los productos tienen un stock y al gestionar ventas se actualiza el inventario
2. Se puede cambiar los datos de la mascota
3. Botón para registrar clientes en la interfaz de inicio de sesión
4. No se podrá suplantar usuarios
5. No se podrá crear un usuario sin mascotas

REQUISITOS QUE NO HEMOS IMPLEMENTADO JUSTIFICACIÓN

1. Los productos tienen un stock y al gestionar ventas se actualiza el inventario

No hemos podido formar una idea clara para poder implementar un sistema de stock sin tener que implementar las transacciones, lo cual debido al “poco” tiempo del trabajo que tenemos no hemos podido meternos con ello

2. Se puede cambiar los datos de la mascota

No hemos podido manejar además de crear una mascota a un usuario en particular poder modificar en cualquier momento la información

3. Botón para registrar clientes en la interfaz de inicio de sesión

En los requisitos de la práctica pone que solamente los administradores pueden crear usuarios (entre ellos clientes)

4. No se podrá suplantar usuarios

No pudimos hacer un sistema de 2FA (doble factor de autenticación) porque tendríamos que estar manejando alguna manera de enviar el código a través del email o un teléfono y es bastante más complejo de lo que tenemos nivel en esta asignatura

5. No se podrá crear un usuario sin mascotas

Pensamos que sería mejor que existiesen usuarios sin mascotas porque nosotros primero creamos el cliente y luego vamos añadiendo mascotas, no lo hacemos todo a la vez

NUEVOS REQUISITOS

1. Una vez se elimina un cliente se eliminan las mascotas que tengan a su nombre
2. El administrador puede crear servicios y ventas a parte de crear artículos
3. El cliente puede ver cada mascota a través de un extensible para poder elegir que información desplegar
4. Tanto el cliente como el Administrador tienen un botón para ver la información del usuario, consta de nombre de usuario, contraseña (oculta al menos que le des al botón) y el rol (además de que si eres cliente tienes información adicional)

CAMBIOS EN ARTEFACTOS

Modificaciones en el diagrama de casos de uso debido a distribución diferente de implementación y cambio de requisitos.

Modificaciones en el diagrama relacional de la base de datos debido a que han cambiado varias de las tablas y asociaciones, además de arreglar varios problemas estéticos.

Todos estos cambios han sido actualizados tanto en los editables (\etc\EDITABLES) como en las versiones guardadas en PDF (\etc\PDF)

PATRONES DE DISEÑO GOF QUE SE HAYAN UTILIZADO EN LA REALIZACIÓN DE LA PRÁCTICA

Usamos patrones como Singleton, Fachada y Builder.

- Singleton: Usamos el patrón Singleton en 2 momentos de la aplicación, el primero al crear una conexión con la base de datos, solamente puede existir una instancia; y la otra seria cuando verificamos que el usuario y la contraseña sean correctas y luego creamos con la clase Usuario el usuario para más tarde usar sus métodos

- Fachada: Usamos el patrón fachada para la conexión que tenemos con la base de datos ya que muchas de las cosas que gestionamos en la aplicación (venta, servicio, artículo etc.) las gestionamos a través de la base de datos

- Builder: al instanciar un Usuario introduciendo las credenciales en la interfaz del login, se relaciona ese Usuario con un Administrador o un Cliente