

Alternativas pensadas TPA1

Requerimientos generales

1. **Se debe permitir el registro de mascotas teniendo en cuenta que una persona puede tener más de una.**

Se toma la decisión de crear una Clase Persona y una Clase Mascota. Siendo que la clase Persona será instanciada con los datos brindados por el documento sin mascotas. Éstas serán creadas y asociadas al mismo tiempo, una vez creada la Persona "Dueña" de las mismas.

Se modela a las mascotas de un dueño como una lista de instancias de Mascota. Se consideró encapsular la abstracción de Mascota en una clase con fines cohesivos.

2. **Se debe permitir que un administrador agregue posibles características de mascotas con facilidad.**

En principio se diseñó como un Enum. Dado que el administrador puede agregar características en tiempo de ejecución, se tuvo que optar por la solución actual de modelarlas como clases según su tipo implementando una interfaz Característica y agrupando todas en un Singleton Repositorio Características.

También se consideró utilizar una estructura de datos HashMap. Se descartó por el fuerte tipado estático de la tecnología. Utilizar esta solución impedía tener distintos tipos de Característica

3. **Se debe permitir que una persona informe que encontró una mascota perdida con chapita para que un voluntario le notifique al dueño de esta situación.**

Se creó una clase Rescatista quien es la única entidad capaz de crear una MascotaPerdida, pudiendo tener solo 1 dado que la alimentación de estos datos proviene de un formulario.

Consideramos que no es necesario referenciar al Rescatista con un ID, usuario u otro identificador ya que el requisito no lo enuncia. Con sus datos personales basta para que el Voluntario pueda contactarlo.

La "Chapita" no fue considerada un dato necesario en el sistema por el momento.

4. **Se debe permitir listar las mascotas encontradas en los últimos 10 días para que los voluntarios puedan contactar a sus respectivos dueños**

Se diseñó e implementó un método que calcule estas mascotas encontradas en los últimos 10 días sin tener en cuenta la notificación al voluntario. Se considera que ésta se maneja de manera informal ajena al dominio de este sistema. Se implementó un método que nos informe si la mascota fue perdida en tal día y luego mediante un filter en la lista de rescatistas se obtienen las mascotas perdidas en los últimos X días.

Requerimientos de seguridad

- 5. Se le debe otorgar la posibilidad de generar un usuario a la persona que está registrando a su mascota en la plataforma. También se debe poder generar usuarios Administradores.**

Creamos una clase abstracta usuario para encapsular el sistema de credenciales y no usar la clase Persona o Administrador. Estas dos clases heredan de la clase Usuario ya que ambas comparten el comportamiento de Login.

- 6. Por el momento, sólo se requiere guardar usuario y contraseña.**

En la clase usuario se modelaron dos atributos User y Password.

- 7. Siguiendo las recomendaciones del OWASP (Proyecto Abierto de Seguridad en Aplicaciones Web)¹, que se ha constituido en un estándar de facto para la seguridad, se pide: Implemente controles contra contraseñas débiles. Cuando el usuario ingrese una nueva clave, la misma ha de verificarse contra la lista del Top 10.000 de peores contraseñas. Alinear la política de longitud, complejidad y rotación de contraseñas con las recomendaciones de la Sección 5.1.1 para Secretos Memorizados de la Guía NIST² 800-63^{3,4}.**

En primer lugar se utilizó un protocolo de Hash MD5 para no almacenar la contraseña tal cual la ingreso el usuario.

De acuerdo con las recomendaciones del OWASP se limitó la longitud de la contraseña y se comparó con un listado de contraseñas comunes para reducir la vulnerabilidad de la misma. Para esto se creó una interfaz de ChequeoDeCalidadDeClave que será utilizada por las distintas clases encargadas de validar la contraseña.

Se decidió no hacer más chequeos por el momento para no caer en un sobrediseño, en un futuro se pueden agregar más medidas de seguridad.

¹ [OWASP Top 10 - 2017](#)

² El NIST es el National Institute of Standards and Technology, de Estados Unidos de América.

³ <https://pages.nist.gov/800-63-3/sp800-63b.html#memsecret>

⁴ Nota: tratar con su ayudante cuáles ítems de esta política de seguridad deben ser implementados

Decisiones en las estructuras

Contacto: Esta información se encapsuló en una nueva clase Contacto, sin comportamiento, con fines cohesivos. Se buscó evitar redundancias entre datos personales y datos de contacto teniendo en cuenta que un Dueño podría dejar un contacto el cual es uno mismo.

Tipo y sexo de Mascota: Se diseñó el tipo de mascota como un Enum para limitar el dominio solamente a Perros y Gatos. Lo mismo para el sexo, teniendo solo dos posibilidades.

Fotos: Para las fotos decidimos recibirlo con una lista de strings, textos. Se presume por el momento que son URLs. Decidimos ir por un diseño minimalista que sea flexible en caso de obtener mas información de “cómo” llegan esas fotos al sistema