Escuela Técnica Superior de Ingeniería Informática Departamento de Lenguajes y Sistemas Informáticos	**Introducción a la Ingeniería del Software y a los Sistemas de Información**	
	Prueba de modelado conceptual **IC G1** 7 nov 2019	Tiempo 1h

Un centro de Acogida de Animales (CAA) licita la modelación conceptual de un Sistema de Información para llevar a cabo su cometido. A continuación se detallan los objetivos y requisitos (**RI**: Requisitos de Información, **RF**: Requisitos Funcionales y **RN**: Reglas de Negocio).

Obj
Como: Responsable del CAA
Necesito: Gestionar los ingresos y adopciones de distintas especies y razas de animales en el Centro de Recogida.
Para: evitar que los animales deambulen por la vía pública.

Obj-1
Como: Responsable del CAA
Necesito: Llevar a cabo el ingreso de un animal, ***abandonado*** en la vía pública o ***entregado*** por una persona, en cuyo caso habría que identificarla, clasificándolo por razas y especies.
Para: Mantener los animales custodiados y ofrecerlos en adopción a personas interesadas.

Obj-2
Como: Responsable del CAA
Necesito: Gestionar consultas de animales disponibles y adjudicar adopciones.
Para: Adjudicar cuanto antes los animales existentes y disminuir costes de mantenimiento del CAD.

RI-1.1
Como: Operario de Recogida de Animales
Necesito: Conocer el nombre de cada especie y nombre de cada raza dentro de una especie.
Un animal ingresado puede tener un nombre y, opcionalmente, un microchip. Siempre se especificará la fecha y hora del ingreso y una descripción adicional del animal.
Si lo ha entregado una persona se la identificará (especificando nombre, dirección y email); en caso contrario se especificará como abandono del animal en la vía pública, describiendo, obligatoriamente, el lugar y hora del hallazgo.
Para: Ingresar cada mascota con la clasificación adecuada y/o conocer datos sobre el hallazgo.

RF-2.1
Como: Operario de Adopciones
Necesito: Facilitar consultas de animales disponibles, estableciendo criterios de selección por razas y, opcionalmente, por términos incluidos en la descripción de cada ingreso.
Para: Facilitar la adopción de animales.

RI-2.2
Como: Operario de Adopciones
Necesito: Registrar adopciones especificando persona que realiza la adopción de un animal disponible y fecha y hora en que se produce.
Para: Dejar constancia de las adopciones realizadas por el CAD.

RN-01
Como: Responsable del CAA
Necesito: Que una persona no pueda adoptar más de dos animales abandonados en un mes.
Para: Poder evitar que se concentren adopciones en personas.

RN-02
Como: Responsable de la Oficina
Necesito: Si un animal es hallado "**abandonado**" en la vía pública, entonces es obligatorio especificar el "**lugar, fecha y hora de encuentro**", en caso contrario se trata de una "**entrega**" y hay que identificar la persona que la realiza (nombre, dirección y email).
Para: Facilitar las búsquedas de sus dueños.

Elabore un modelo conceptual usando tantos diagramas de clases UML como considere necesario. Si identifica algún problema en el enunciado, indíquelo junto con el modelo desarrollado.

Prueba de modelado conceptual. IC G1
7 nov 2019

Solución

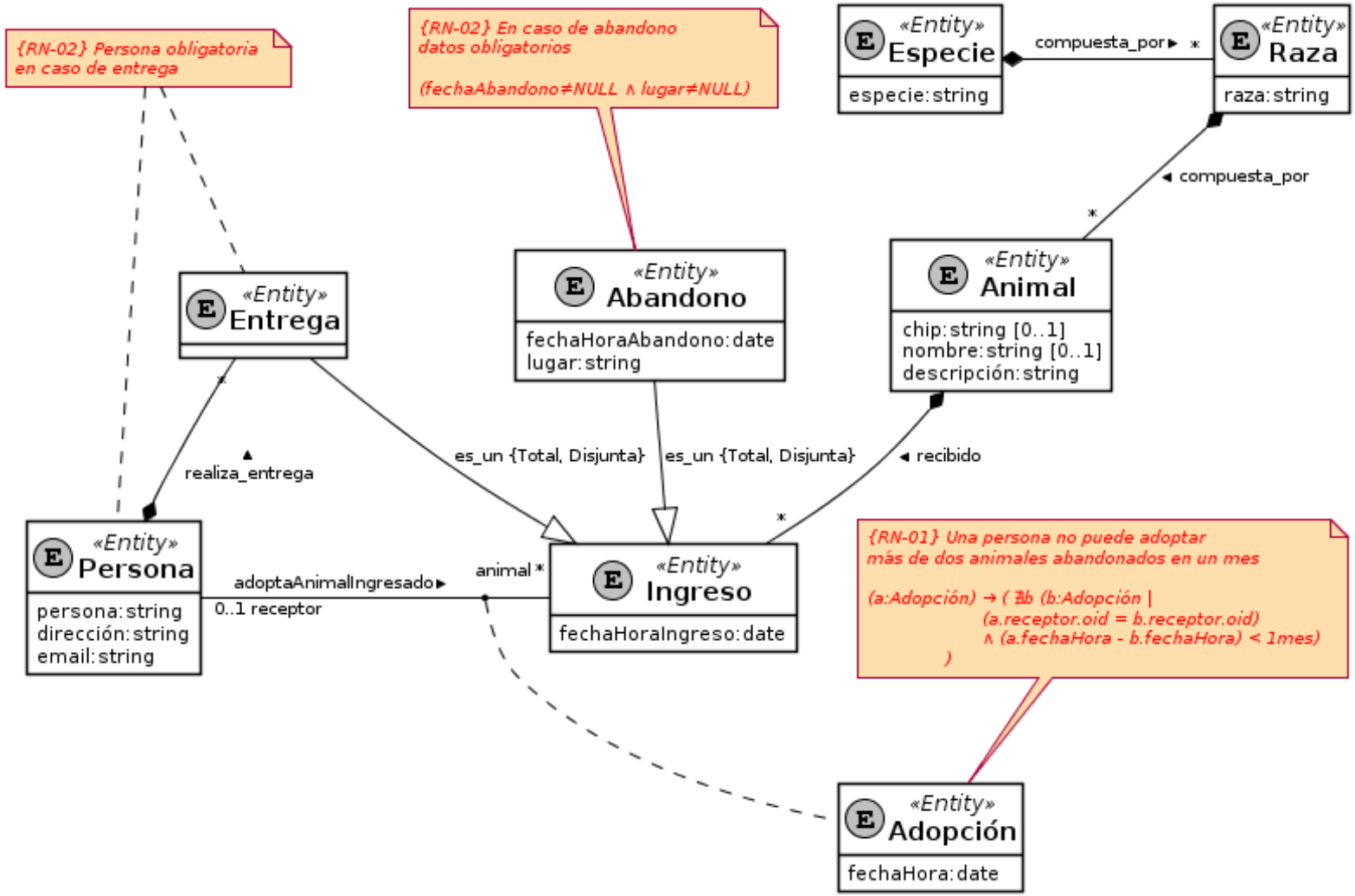

Fig.1. Diagrama de clases

PlantUml

```
@startuml
'Acogida de animales
'Opciones gráficas
skinparam class {
   FontSize 15
   FontStyle bold
   BackgroundColor transparent
   borderColor  black
}
skinparam arrow {
   FontSize 10
   Color  black
}
'skinparam linetype ortho
skinparam note {
   FontSize 10
   FontStyle italic
   FontColor Red
   BackgroundColor NavajoWhite
}

skinparam shadowing false
hide methods

'Entidades
'---------
class Especie << (E,silver) Entity >> {
especie:string
}

class Raza << (E,silver) Entity >> {
raza:string
}
class Animal << (E,silver) Entity >> {
chip:string [0..1]
nombre:string [0..1]
descripción:string
}
class Ingreso << (E,silver) Entity >> {
fechaHoraIngreso:date
}
class Abandono << (E,silver) Entity >> {
fechaHoraAbandono:date
lugar:string
}
class Entrega << (E,silver) Entity >> {
}

class Persona << (E,silver) Entity >> {
persona:string
dirección:string
email:string
}

class Adopción << (E,silver) Entity >> {
fechaHora:date
}

'Asociaciones
'------------
Especie *-right- "*" Raza : > compuesta_por
Raza *-down- "*" Animal : > compuesta_por
Animal *-down- "*" Ingreso : recibido >
Persona *-up-  "*" Entrega  : > realiza_entrega
Persona "1 receptor" - "animal *  " Ingreso : adoptaAnimalIngresado >
(Persona, Ingreso) .. Adopción
Abandono -down-|> Ingreso: "es_un {Total, Disjunta}"
Entrega -right-|> Ingreso: "es_un {Total, Disjunta}"

' Restricciones
'--------------
note as Nx
{RN-02} Persona obligatoria
en caso de entrega
end note
Nx..Entrega
Nx..Persona

note top of Abandono
{RN-02} En caso de abandono
datos obligatorios

(fechaAbandono≠NULL ∧ lugar≠NULL)
end note
note as Ny
   {RN-01} Una persona no puede adoptar
   más de dos animales abandonados en un mes

   (a:Adopción) ⟶ ( ∄b (b:Adopción |
                  (a.receptor.oid = b.receptor.oid)
                  ∧ (a.fechaHora - b.fechaHora) < 1mes)
               )
end note
Ny..Adopción
```

Código PlantUML (probar en https://www.planttext.com/)