



## PET HERO



**CARRERA:** Tecnicatura Universitaria en Programación (TUP)

**MATERIA:** Metodología de Sistemas I

**INTEGRANTES:** Agustín Álvarez, Matías de Andrade, Daniel Diaz

**COMISIONES:** 4 y 5

**DOCENTES:** Fernando Castañeda y Cristian Halm

**GITHUB DEL PROYECTO:** [PET HERO](#)

## CONTENIDO

1.	INTRODUCCIÓN.....	3
1.1.	Propósito.....	3
1.2.	Alcance.....	3
1.3.	Definiciones.....	3
2.	FUNCIONALIDADES DEL SISTEMA.....	4
3.	DEFINICIÓN DE REQUISITOS DEL SISTEMA.....	5
3.1.	Definición de Requisitos Funcionales.....	5
3.2.	Definición de Requisitos No Funcionales.....	7
4.	DIAGRAMA DE CASO DE USO.....	8
4.1.	Especificación Caso de Uso.....	8
4.2.	Camino de Éxito.....	9
4.3.	Alternativas y Excepciones.....	10
5.	DIAGRAMA DE MODELO CONCEPTUAL.....	11
6.	DIAGRAMA DE ENTIDADES Y RELACIONES.....	12

## INTRODUCCIÓN

En el siguiente documento se presentan las especificaciones de requisitos del proyecto “PET HERO” , una aplicación mediante la cual los usuarios podrán solicitar el cuidado de sus mascotas para una fecha determinada.

En esta sección se desarrollan los propósitos, alcance, definiciones y referencias. Luego se describirá al sistema, se definirán los requisitos y se expondrán los diagramas vinculados a la aplicación.

### PROPÓSITO

El propósito de la aplicación es que un usuario pueda registrarse como un **Dueño** (Owner) donde podrá gestionar a sus mascotas (Perro o Gato) y solicitar reservas para su cuidado, o **Cuidador** (Keeper) en la cual podrá gestionar estas solicitudes, aceptarlas, rechazarlas y gestionar sus días de trabajo o tipos de mascotas a cuidar. Estos usuarios también podrán interactuar entre sí a través de la implementación de un chat de mensajes.

### ALCANCE

Esta aplicación está dirigida a todo tipo de persona que tenga alguna mascota que necesite ser cuidada o cualquier persona que necesite una salida laboral y quiera dedicarse al cuidado de mascotas por un tiempo determinado.

### DEFINICIONES

Aplicación web: tipo de software que se codifica en un lenguaje soportado por los navegadores web y cuya ejecución es llevada a cabo por el navegador en Internet o en una intranet.

Owner: Dueño de una o más mascotas.

Keeper: Cuidador de mascotas.

Admin: Administrador del sistema.

Requisito funcional: condición que describe qué debe hacer el sistema respecto a su entorno. Especifica los servicios que se espera que el sistema proveerá, sus entradas y salidas, excepciones, etc.

Requisito no funcional: propiedades emergentes del sistema. Ejemplos: la fiabilidad, el tiempo de respuesta, la capacidad de almacenamiento, la capacidad de los dispositivos de entrada/salida, y la representación de datos que se utiliza en las interfaces del sistema.

Requisito: condición que debe cumplir un sistema para satisfacer un contrato, una norma o una especificación.

## **FUNCIONALIDADES DEL SISTEMA**

Las funcionalidades del sistema varían según el tipo de usuario que lo esté utilizando.

Funcionalidades generales:

- Alta / Registro de usuario.
- Login / Ingreso de usuario.
- Update / Modificación de datos personales del usuario.
- Logout / Cerrar sesión.
- Ver perfil propio.

Funcionalidades Owner:

- Agregar una mascota (Perro o Gato).
- Consultar listado de sus mascotas.
- Dar de baja una mascota.
- Consultar lista de Keepers disponibles según rango de fechas.
- Solicitar una reserva indicando el keeper, el día y la mascota a cuidar.
- Ver y realizar los pagos de reservas pendientes
- Chatear con un keeper.

Funcionalidades Keeper:

- Consultar lista de reservas pendientes.
- Aceptar o rechazar una solicitud de reserva.
- Consultar el historial de reservas.
- Modificar la disponibilidad.
- Chatear con un owner.

Funcionalidades Admin:

- Consultar lista de Owners registrados.
- Consultar lista de Keepers registrados.
- Consultar lista de todos los usuarios registrados.
- Eliminar un usuario.
- Agregar nuevos tipos de usuarios al sistema.
- Listar los tipos de usuarios utilizables.
- Ver todas las mascotas registradas de todos los Owners.

## DEFINICIÓN DE REQUISITOS DEL SISTEMA

En esta sección se presentan los requisitos funcionales y no funcionales que deberán ser cumplidos por el sistema.

### DEFINICIÓN DE REQUISITOS FUNCIONALES

A continuación, se describen los requisitos funcionales del sistema por tipo de usuario. Se ha elegido esta clasificación debido a que el sistema presenta diversas funcionalidades según se trate del usuario Owner, Keeper o Admin.

ID	Nombre	Descripción	Usuario
RF-001	Registrar usuario	El sistema deberá permitir que los usuarios no registrados se registren a través de su correo y contraseña. Además, deberá elegir el tipo de usuario.	Owner Keeper Admin
RF-002	Iniciar sesión	El sistema deberá permitir que los usuarios registrados ingresen través de su correo y contraseña.	Owner Keeper Admin
RF-003	Cerrar sesión	El sistema deberá permitir al usuario cerrar sesión y dirigir a la pantalla de inicio	Owner Keeper Admin
RF-004	Visualizar perfil	El sistema deberá proporcionar a los usuarios su información personal según su tipo.	Owner Keeper Admin
RF-005	Modificar datos personales	El sistema permitirá a los usuarios poder modificar sus datos personales	Owner Keeper Admin
RF-006	Consultar lista de usuarios registrados	El sistema le permitirá al administrador ver una lista con todos los usuarios registrados en la aplicación	Admin
RF-007	Eliminar usuario	El sistema le permitirá al administrador eliminar usuarios.	Admin
RF-008	Listar tipos de usuario	El sistema le mostrará al administrador los tipos de usuarios registrados para utilizar	Admin

RF-009	Agregar tipos de usuario	El administrador tiene la opción de agregar nuevos tipos de usuarios	Admin
RF-010	Agregar una mascota	El sistema le permitirá al owner registrado poder agregar una mascota a su perfil ingresando algunos datos como nombre, fecha de nacimiento, tamaño, raza y multimedia.	Owner
RF-011	Dar de baja una mascota	El owner tendrá la capacidad de dar de baja alguna mascota que tenga registrada	Owner
RF-012	Consultar lista de keepers disponibles	El owner debe poder buscar a todos los keepers disponibles según un rango de fechas ingresado	Owner
RF-013	Generar una solicitud de reserva	El sistema le permitirá al owner generar una solicitud de reserva a un keeper seleccionado indicando la fecha a reservar, y las mascota a cuidar	Owner
RF-014	Visualizar facturas pendientes de pago	El owner podrá visualizar una lista de aquellas facturas pendientes de pago emitidas por los keepers reservados	Owner
RF-015	Realizar pago de factura	El sistema le brindará al owner un formulario de pago donde deberá ingresar la tarjeta a debitar, su número y código.	Owner
RF-016	Modificar disponibilidad	El sistema le permitirá al keeper poder modificar los días y otros datos como el tipo de mascota, el tamaño o el precio de las reservas.	Keeper
RF-017	Consultar lista de solicitudes de reserva pendientes	El keeper podrá visualizar una lista de todas las solicitudes pendientes donde le aparecerá el dueño emisor de la solicitud, su mascota, tipo, tamaño y fecha solicitada.	Keeper
RF-018	Aceptar o rechazar solicitudes	El keeper podrá, desde la lista anterior, aceptar o cancelar cualquiera de las reservas listadas.	Keeper

RF-019	Enviar factura por mail	El sistema le enviará por mail al owner una factura con los detalles de la reserva aceptada por el keeper.	Keeper
RF-020	Listar historial de reservas	El keeper podrá ver una lista de todas las reservas históricas aceptadas y finalizadas.	Keeper
RF-021	Chat entre owner y keeper	El owner podrá iniciar un chat con cualquier keeper antes de generar una solicitud de reserva. Un keeper solo podrá conversar con aquellos owners que hayan iniciado un chat con él.	Owner Keeper

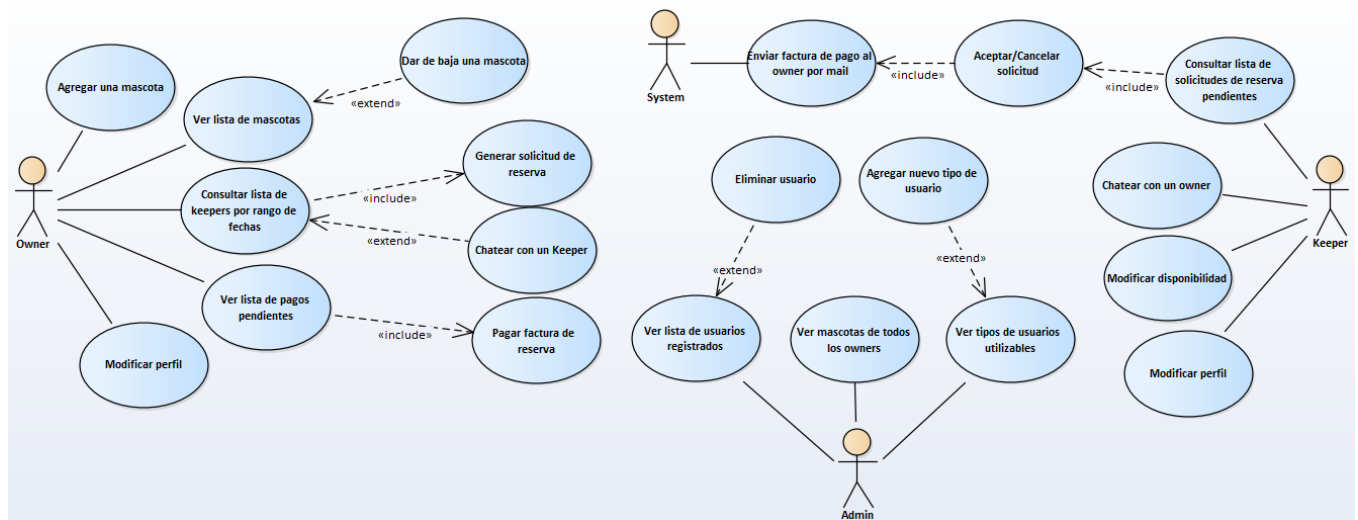
## DEFINICIÓN DE REQUISITOS NO FUNCIONALES

A continuación, se presentan los requisitos no funcionales del sistema

ID	Descripción
RF-001	El sistema deberá ser implementado como una aplicación web
RF-002	Cada usuario que desee ingresar deberá introducir en la página principal su email y una contraseña, la cual será validada por el sistema, dándole acceso o enviándole un mensaje para que introduzca nuevamente sus datos
RF-003	Cada usuario tendrá asignado un perfil, que servirá para activar los servicios u opciones que puede realizar dentro del sistema
RF-004	El sistema deberá tener una interfaz gráfica sencilla, amigable, y con buena legibilidad. Estará basada en menús, ventanas, listas desplegables y botones de acción
RF-005	El sistema deberá ser desarrollado bajo software libre, utilizando el lenguaje de programación PHP y el estándar HTML para el diseño de las páginas web. Se hará uso de la herramienta Bootstrap para brindar estilos. De esta forma se garantizará que el código HTML generado pueda ser interpretado por cualquiera de los navegadores comerciales existentes en el mercado

RF-006	El sistema deberá ser diseñado según la arquitectura MVC de tres capas
RF-007	La organización, manipulación, consulta y almacenamiento de los datos serán en una base de datos de forma local.

### DIAGRAMA DE CASOS DE USO



### ESPECIFICACIÓN DE CASO DE USO: Generando una solicitud de reserva

Identificador	RF-012 y RF-013
Nombre	Generando una solicitud de reserva
Analistas	Agustín Alvarez Matias de Andrade Daniel Diaz
Actor	Owner
Prioridad	Alta
Objetivos	El objetivo es que el owner pueda realizarle una solicitud de reserva a un keeper en una fecha elegida.
Precondiciones	<ul style="list-style-type: none"> <li>Deberán existir keepers que estén disponibles dentro del rango de fechas ingresado.</li> <li>El owner en cuestión deberá tener al menos una mascota activa y registrada a su nombre.</li> </ul>
Postcondiciones	<ul style="list-style-type: none"> <li>El keeper recibirá una notificación de solicitud de reserva</li> </ul>



## Interfaz de usuario: Selección del keeper

PET HERO									
Name	Last Name	Email	Phone Number	Adress	Pet Size that i can handle :)	Since	To	Price	
Agustin	Alvarez	agus@gmail.com	351235134	Keeper street 1	small	2022-11-22	2022-12-30	U\$S1000	REQUEST RESERVATION
Matias	de Andrade	matute@gmail.com	352364352	keeper street 2	medium	2022-11-29	2022-12-16	U\$S1500	REQUEST RESERVATION

## Completando datos de la solicitud:

PET HERO									
Loading Reserve									
Choose a day			Pet Size Available			Choose my pet			
2022-11-29			medium			Tom Michi			
GENERATE									

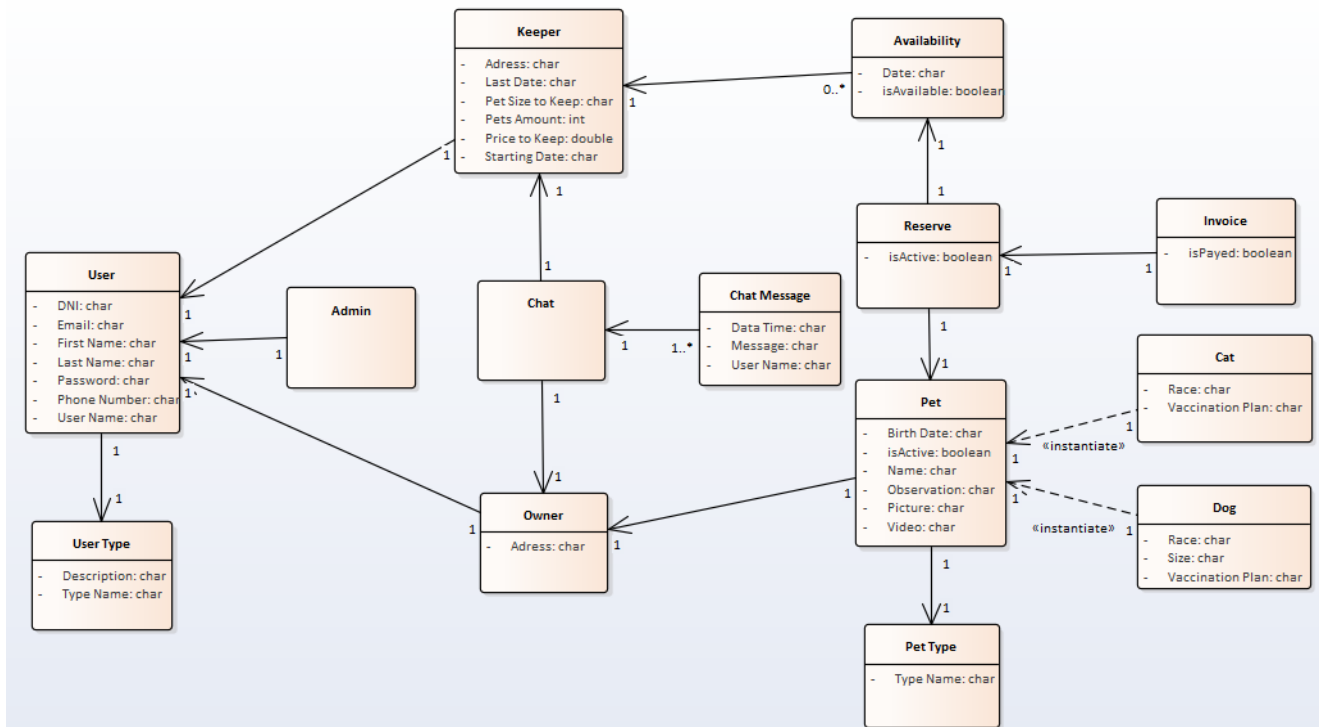
## CAMINO DE ÉXITO

Paso	Actor	Sistema
1	El owner ingresa un rango de fechas para filtrar los keepers disponibles	El sistema lista los keepers disponibles en esas fechas
2	El owner selecciona al keeper que quiere reservar	El sistema muestra un formulario para completar los datos de la solicitud
3	El owner ingresa el día y la mascota que quiere reservar, por último, genera la solicitud	El sistema verifica la disponibilidad del keeper para el día elegido y en caso correcto envía la solicitud.

**ALTERNATIVAS Y EXCEPCIONES**

<b>Paso</b>	<b>Actor</b>	<b>Sistema</b>
1	El owner ingresa un rango de fechas en las que ningún keeper está disponible	El sistema le informa al usuario que no hay keepers disponibles en ese rango indicado.
2	El owner ingresa una o más mascotas en un día donde el keeper ya tenía el cupo de reservas lleno.	El sistema pide reingresar el día a solicitar.
3	El owner ingresa más de una mascota que son de diferentes tipos.	El sistema notifica al usuario que no pueden reservarse 2 tipos de mascotas diferentes el mismo día.
4	El owner intenta reservar una mascota en un día donde el keeper ya tenía una reserva con una mascota de otro tipo	El sistema hace la verificación y notifica al usuario que no se puede realizar la solicitud
5	El owner ingresa una mascota cuyo tamaño no coincide con el tamaño especificado por el keeper	El sistema le notifica al usuario que no es posible realizar la solicitud
6	El owner no tiene ninguna mascota registrada o activa.	El sistema le sugiere al usuario registrar una nueva mascota para empezar a solicitar reservas.
7	El owner no selecciona ninguna mascota en la carga de la solicitud.	El sistema no permite generar la solicitud y pide seleccionar una mascota para generarla.

## DIAGRAMA DE MODELO CONCEPTUAL



## DIAGRAMA DE ENTIDAD RELACIÓN (DER)

