



Universidad de Santiago de Chile
FACULTAD DE CIENCIA
DEPARTAMENTO DE MATEMÁTICA Y CIENCIA DE LA COMPUTACIÓN

INGENIERÍA DE SOFTWARE II
INFORME N°5. “PROGRAMACIÓN ORIENTADA A OBJETOS - IMPLEMENTACIÓN FINAL”



Integrantes:

Sebastián Holloway I.
Pablo Maertens M.
Sergio Muñoz C.
Katerine Muñoz T.
Ignacio Olave R.

Profesor:

Dino Araya S.

Fecha de entrega:

21 de enero de 2019

Resumen

En este informe se presenta el análisis, desarrollo y programación del sistema *Buscatdog*, el cual permite buscar y crear publicaciones de mascotas perdidas¹ y abandonadas².

Para esto utilizamos una Metodología de Desarrollo Ágil SCRUM³ en conjunto con la versión simplificada de OMT++⁴, que nos permite diseñar y modelar las distintas funcionalidades necesarias para cumplir con el propósito del software antes mencionado.

Con la puesta en marcha del software esperamos que a aquellos dueños que pierden sus mascotas, al utilizar la plataforma, puedan reencontrarse con estas. Por esto mismo que es de gran importancia la correcta implementación del sistema.

Finalmente, se conocerán las conclusiones de cada capítulo del informe, que representan las distintas etapas que se debieron trabajar para poner en marcha el software, así como también la conclusión general del proyecto y el proceso realizado para entregar un sistema terminado que cumpla con lo estipulado en este documento.

PALABRAS CLAVES:

¹Parte de las funcionalidades principales del sistema Buscatdog.

²Parte de las funcionalidades principales del sistema Buscatdog.

³Metodología de desarrollo de software que nace como alternativa a las metodologías tradicionales y que se basa en lo escrito en el Manifiesto Ágil.

⁴Metodología Orientada a Objetos que permite modelar las distintas funcionalidades del software.

Índice

1	PRESENTACIÓN Y PLANIFICACIÓN DEL PROYECTO ÁGIL.	7
1.1	Introducción.	7
1.2	Objetivos del proyecto.	7
1.2.1	Objetivo general.	7
1.2.2	Objetivos específicos.	7
1.3	Descripción de la problemática.	8
1.3.1	Motivación.	8
1.3.2	Definición del problema.	8
1.3.3	Estado del arte.	8
1.3.3.1	Sistema <i>WOF</i> ⁽³⁾	8
1.3.3.2	Sistema <i>BuscoMascota</i> ⁽⁴⁾	8
1.3.3.3	Grupos de Facebook.	8
1.4	Descripción de la solución propuesta.	9
1.4.1	Características de la solución.	10
1.4.2	Propósitos de la solución.	10
1.4.3	Alcances y limitaciones de la solución.	10
1.5	Metodología, herramientas y ambiente de desarrollo.	10
1.5.1	Metodología a usar.	10
1.5.2	Herramientas de desarrollo.	11
1.5.3	Ambiente de desarrollo.	11
1.6	Plan de trabajo.	12
1.6.1	Planificación de entrevistas.	12
1.6.2	Planificación del proyecto.	12
1.7	Conclusiones.	12
2	ANÁLISIS DE REQUERIMIENTOS Y CASOS DE USO.	13
2.1	Introducción.	13
2.2	Análisis de requerimientos.	13
2.2.1	Requerimientos funcionales	13
2.2.2	Requerimientos no funcionales	14
2.2.3	Requerimientos de implementación	14
2.3	Casos de uso.	14
2.3.1	Diagrama de casos de uso.	14
2.3.2	Especificación de casos de uso.	16
2.4	Conclusiones.	20
3	ANÁLISIS ORIENTADO A OBJETOS OMT++.	21
3.1	Introducción.	21
3.2	Análisis OO.	21
3.2.1	Análisis de objetos	21
3.2.1.1	Modelo de objetos del análisis.	21
3.2.2	Análisis del comportamiento.	26
3.2.2.1	Especificación de operaciones.	26
3.2.3	Especificación de la interfaz de usuario.	27
3.2.3.1	Diagramas de diálogo.	28
3.2.3.2	Especificación de componente.	28
3.3	Conclusiones.	38

4	DISEÑO ORIENTADO A OBJETOS OMT++.	39
4.1	Introducción.	39
4.2	Diseño OO.	39
4.2.1	Diseño de objetos.	39
4.2.2	Diseño del comportamiento.	42
4.3	Conclusiones.	48
5	PROGRAMACIÓN ORIENTADA A OBJETOS OMT++.	48
5.1	Introducción.	48
5.2	Clases Modelo.	48
5.3	Clases Vista.	50
5.4	Clases Controlador.	52
6	CONCLUSIONES DEL PROYECTO.	54
7	REFERENCIAS BIBLIOGRÁFICAS.	55
8	APÉNDICE A. MANUAL DE USUARIO	55
8.1	Crear publicación mascotas perdidas	55
8.2	Crear publicación mascotas abandonadas	57
8.3	Registrar usuario	59
8.4	Alojar mascota abandonada	60
8.5	Actualizar ubicación de mascota	62
8.6	Iniciar sesión	63
8.7	Listar publicaciones	63
8.8	Actualizar estado de mascota perdida	65
9	APÉNDICE B. MANUAL DE EXPLOTACIÓN	65
9.1	Licencia.	65
9.2	Instalación.	65
9.2.1	PostgreSQL 11.1	65
9.2.2	XAMPP 7.3.0	71
10	APÉNDICE C. GRUPOS DE FACEBOOK DE BÚSQUEDA DE MASCOTAS PERDIDAS.	76
11	APÉNDICE D. CARTA GANTT	77
12	APÉNDICE E. DETALLE DE REUNIONES.	79

Índice de tablas

1	Herramientas de desarrollo.	11
2	Hardware de desarrollo.	11
3	Consultorías	12
4	Especificación de caso de uso n°1 - Gestión Publicación.	16
5	Especificación de caso de uso n°2 - Publicar mascota perdida.	16
6	Especificación de caso de uso n°3 - Publicar mascota abandonada.	17
7	Especificación de caso de uso n°4 - Alojar mascota abandonada.	17
8	Especificación de caso de uso n°5 - Actualizar publicación mascota.	18
9	Especificación de caso de uso n°6 - Actualizar estado de mascota perdida.	18

10	Especificación de caso de uso n°7 - Actualizar ubicación mascota perdida	19
11	Especificación de caso de uso n°8 - Listar publicaciones mascotas.	19
12	Especificación de caso de uso n°9 - Registrar usuario.	20
13	Clase Persona.	22
14	Clase Administrador.	23
15	Clase Invitado.	23
16	Clase Usuario.	23
17	Clase Mascota.	23
18	Clase Perro.	24
19	Clase Gato.	24
20	Clase Publicación.	24
21	Clase Perdida.	25
22	Clase Abandonada.	25
23	Clase Alojamiento.	25
24	Lista de operaciones del sistema Buscatdog.	27
25	Descripción de comportamiento Inicio Sesión	29
26	Descripción de comportamiento Crear publicación mascota abandonada	30
27	Descripción de comportamiento Crear publicación mascota abandonada con alojamiento	31
28	Descripción de comportamiento Crear publicación mascota perdida	32
29	Descripción de comportamiento registrar usuario	33
30	Descripción de comportamiento error	33
31	Descripción de comportamiento Buscatdog	34
32	Descripción de comportamiento Panel usuario	35
33	Descripción de comportamiento Visualizar publicación	36
34	Descripción de comportamiento Actualizar estado publicación	37
35	Descripción de comportamiento Cambiar contraseña	38
36	Clase Modelo - Usuario.	48
37	Clase Modelo - Invitado.	49
38	Clase Modelo - Mascota.	49
39	Clase Modelo - Perdida	49
40	Clase Modelo - Abandonada	50
41	Clase Modelo - Alojamiento	50
42	Clase Vista - Principal	50
43	Clase Vista - Crear Publicación Perdida	51
44	Clase Vista - Crear Publicación Abandonada	51
45	Clase Vista - Crear Publicación Alojamiento	51
46	Clase Vista - Registrar Usuario	51
47	Clase Vista - Cambio Contraseña	52
48	Clase Vista - Inicio Sesión	52
49	Clase Controlador - Principal	52
50	Clase Controlador - Crear Publicación Perdida	53
51	Clase Controlador - Crear Publicación Abandonada	53
52	Clase Controlador - Crear Publicación Alojamiento	53
53	Clase Controlador - Registro Usuario	53
54	Clase Controlador - Cambio Contraseña	54
55	Clase Controlador - Inicio Sesión	54

Índice de figuras

1	Metáfora	9
2	Diagrama de casos de uso del sistema Buscatdog.	15
3	Modelo de objetos del análisis del sistema Buscatdog	22
4	Diagrama de diálogo del sistema Buscatdog.	28
5	Diagrama de control Inicio Sesión.	29
6	Diagrama de control Crear publicación mascota abandonada.	29
7	Diagrama de control Crear publicación mascota abandonada con alojamiento.	30
8	Diagrama de control Crear publicación mascota perdida.	31
9	Diagrama de control registrar usuario.	32
10	Diagrama de control error.	33
11	Diagrama de control Buscatdog.	34
12	Diagrama de control Panel usuario.	35
13	Diagrama de control Visualizar publicación.	36
14	Diagrama de control Actualizar estado publicación.	37
15	Diagrama de control Cambiar contraseña.	38
16	Modelo Vista Controlador <i>MVC</i> , parte izquierda, del sistema <i>Buscatdog</i> . Parte 1	40
17	Modelo Vista Controlador <i>MVC</i> , parte derecha, del sistema <i>Buscatdog</i> . Parte 2	41
18	Traza Crear publicación de mascota perdida.	42
19	Traza Crear publicación de mascota abandonada.	43
20	Traza Crear publicación de mascota abandonada con alojamiento.	44
21	Traza Registrar Usuario.	45
22	Traza Cambio Contraseña.	46
23	Traza Inicio Sesión.	47
24	Crear publicación mascotas perdidas. Paso 1.	56
25	Crear publicación mascotas perdidas. Paso 2.	57
26	Crear publicación mascotas abandonadas. Paso 1.	58
27	Crear publicación mascotas abandonadas. Paso 2.	58
28	Registrar usuario. Paso 1.	59
29	Registrar usuario. Paso 2.	59
30	Registrar usuario. Paso 3.	60
31	Alojar mascota abandonada. Paso 1.	60
32	Alojar mascota abandonada. Paso 2.	61
33	Alojar mascota abandonada. Paso 3.	61
34	Actualizar ubicación de mascota. Paso 1.	62
35	Actualizar ubicación de mascota. Paso 2.	62
36	Iniciar sesión. Paso 1.	63
37	Iniciar sesión. Paso 2.	63
38	Iniciar sesión. Paso 3.	63
39	Listar publicaciones. Paso 1.	64
40	Listar publicaciones. Paso 2.	64
41	Listar publicaciones. Paso 3.	64
42	Actualizar estado de mascota perdida. Paso 1.	65
43	Paso 1 - Instalación PostgreSQL	66
44	Paso 2 - Instalación PostgreSQL	66
45	Paso 3 - Instalación PostgreSQL	67
46	Paso 4 - Instalación PostgreSQL	67
47	Paso 5 - Instalación PostgreSQL	68

48	Paso 6 - Instalación PostgreSQL	68
49	Paso 7 - Instalación PostgreSQL	69
50	Paso 8 - Instalación PostgreSQL	69
51	Paso 9 - Instalación PostgreSQL	70
52	Paso 10 - Instalación PostgreSQL	70
53	Creación de la Base de Datos	71
54	Paso 1 - Instalación XAMPP	71
55	Paso 2 - Instalación XAMPP	71
56	Paso 3 - Instalación XAMPP	72
57	Paso 4 - Instalación XAMPP	72
58	Paso 5 - Instalación XAMPP	73
59	Paso 6 - Instalación XAMPP	73
60	Paso 7 - Instalación XAMPP	74
61	Paso 8 - Instalación XAMPP	74
62	XAMPP - Panel de control	75
63	XAMPP - Panel de control (módulos)	75
64	Grupos de facebook para buscar mascotas perdidas.	76
65	Carta Gantt Sistema Buscatdog: Sprint 1.	77
66	Carta Gantt Sistema Buscatdog: Sprint 2.	78

Capítulo 1.

1. PRESENTACIÓN Y PLANIFICACIÓN DEL PROYECTO ÁGIL.

1.1. Introducción.

El presente informe introduce el proyecto que se desarrollará durante el semestre, el sistema Buscatdog. Gran cantidad de mascotas se pierden a diario en Chile, consecuencia de esto comienza una etapa de búsqueda por parte de sus dueños utilizando diversas estrategias para dar con la ubicación de su mascota. Motivado por esto surge Buscatdog, una aplicación web dedicada a encontrar mascotas perdidas.

Las siguientes secciones especificarán los objetivos principales y secundarios de este proyecto, de dónde surge la problemática y cómo se planea solucionarla, también se describirá la metodología a usar, las herramientas y ambientes de desarrollo, así como el plan de trabajo que se adoptará para llevar a cabo el proyecto. La gestión de este proyecto se basará en la metodología de desarrollo ágil SCRUM, mientras que la parte técnica se basará en una versión simplificada de OMT++.

1.2. Objetivos del proyecto.

A continuación se presentarán los objetivos del sistema *Buscatdog*.

1.2.1. Objetivo general.

Diseño e implementación del sistema *Buscatdog*.

1.2.2. Objetivos específicos.

- Planificación de sprint de desarrollo en ambiente ágil.
- Analizar requerimientos y casos de uso.
- Especificación de las historias de usuario.
- Análisis orientado a objetos.
- Especificación de la interfaz de usuario.
- Diseño orientado a objetos.
- Implementación de la solución en base a programación orientada a objetos.
- Plan de pruebas unitarias.
- Plan de pruebas funcionales de la solución para verificación de objetivos mediante las historias de usuario.
- Validación del sistema por parte del cliente.
- Puesta en producción.

1.3. Descripción de la problemática.

El siguiente párrafo muestra la motivación y la definición del problema, a considerar en el sistema .

1.3.1. Motivación.

Cada vez que se pierde alguna mascota, el primer instinto del dueño/a es pegar panfletos con la información de la mascota perdida por los alrededores. Generalmente estos avisos no duran mucho tiempo puestos en las calles, o simplemente la gente no les presta atención. Esto nos lleva, por ejemplo, al caso de los perros, que son la mascota más común en Chile⁽¹⁾, solamente el 25 % de los que se encuentran en las calles tienen dueño/a o domicilio, lo que quiere decir que el 75 % restante son perros perdidos o ambulantes⁽²⁾.

1.3.2. Definición del problema.

El problema a resolver con este proyecto es que las mascotas sean reunidas con sus dueños/as, ya que con los métodos actuales de búsqueda, como lo son pegar panfletos en las calles o publicar en grupos de Facebook específicos de búsqueda de mascotas que abarcan áreas tan grandes como lo son comunas completas, no están siendo totalmente efectivos puesto a que la mayoría de estas siguen estando en las calles.

Por otro lado, el utilizar grupos de Facebook no asegura la entrega completa de información a otros ya que al cubrir grandes áreas de población, las publicaciones pierden relevancia rápidamente, quedando en un segundo plano las que son más antiguas.

1.3.3. Estado del arte.

A continuación se muestran algunas soluciones existentes relacionadas con el problema.

1.3.3.1 Sistema *WOF*⁽³⁾.

Este es un sistema que permite encontrar y reportar casos de mascotas perdidas, en adopción y emergencias tales como atropellos o situaciones de maltrato. Este sistema está disponible para plataforma móvil, iOS y Android.

1.3.3.2 Sistema *BuscoMascota*⁽⁴⁾.

Este sistema permite a los usuario encontrar mascotas desaparecidas, buscar parejas para sus mascotas, comprar o vender mascotas y adoptar mascotas. Funciona en una plataforma web donde es posible crearse una cuenta de usuario con un E-Mail, o bien, conectarse con una cuenta de Facebook. Para poder publicar una mascota perdida o encontrada es necesario tener una cuenta de usuario.

1.3.3.3 Grupos de Facebook.

En la red social Facebook mucha gente suele crear grupos de mascotas perdidas por comunas donde realizan publicaciones especificando la foto, el lugar y algunos detalles relevantes para encontrar la

mascota perdida. El detalle se puede ver en *apéndice: Sección C*.

Como se puede apreciar, las 3 soluciones presentadas cubren el problema principal que es crear publicaciones en los casos en que el usuario haya perdido o encontrado una mascota. La gran diferencia entre el sistema Buscatdog y estas soluciones es que la solución propuesta permite obtener un rango estimado de localización de la mascota, la cual se calcula de acuerdo a las características de esta (edad, carácter, raza, entre otras).

1.4. Descripción de la solución propuesta.

En seguida se presentarán las características, propósitos, alcances y limitaciones de la solución a considerar para el proyecto.

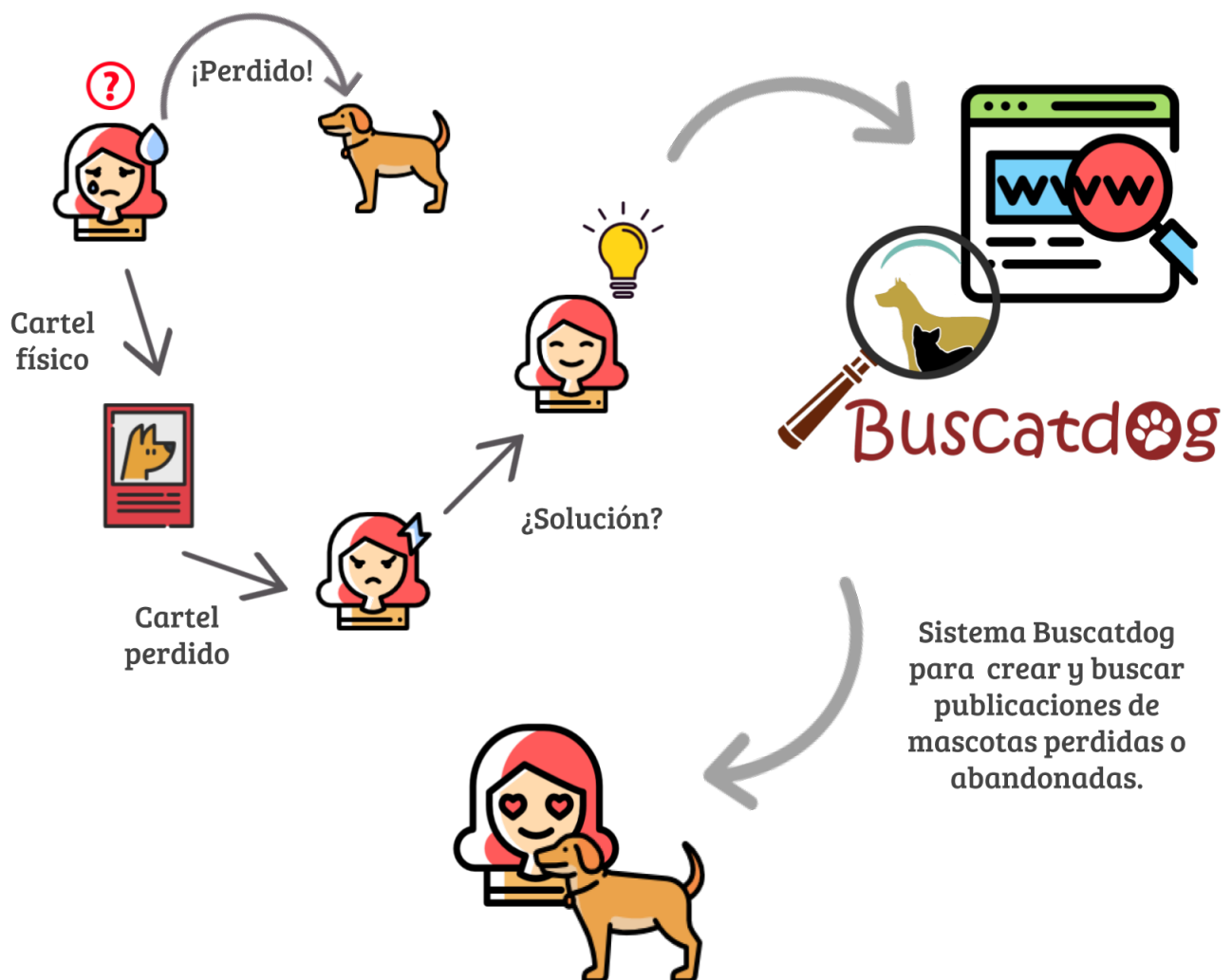


Figura 1: Metáfora

1.4.1. Características de la solución.

El sistema debe ser eficiente y fácil de utilizar, además de contar con las siguientes características:

- Control de acceso/registro a la plataforma a través de correo electrónico.
- Crear, editar y borrar publicaciones de mascotas perdidas.
- Crear, editar y borrar publicaciones de mascotas abandonadas
- Crear, editar y borrar el perfil del usuario.
- Acceder a mapa con visualización de área donde posiblemente se encuentre la mascota.
- Listar mascotas perdidas bajo parámetros definidos por el usuario.

1.4.2. Propósitos de la solución.

Los propósitos de la solución consisten en suministrar una plataforma en línea de fácil uso, que permita la búsqueda de mascotas perdidas gracias al uso de tecnologías informáticas que incluyen la geolocalización suministrada por la Plataforma de Google Maps⁽⁵⁾ y las distintas API's que allí se encuentran.

1.4.3. Alcances y limitaciones de la solución.

- No incluye poblamiento de datos.
- El software es de uso libre y gratuito.
- Incluye el mantenimiento del software por el tiempo definido para el cliente, pero solo en caso de errores.
- El software es solo web, por lo que necesita de conexión a internet.
- La información y manipulación de datos por parte de los usuarios es exclusiva responsabilidad de ellos mismos.

1.5. Metodología, herramientas y ambiente de desarrollo.

En esta sección se presenta la metodología a usar, las herramientas y ambientes de desarrollo para *Buscatdog*.

1.5.1. Metodología a usar.

La metodología ágil a utilizar será SCRUM, ya que consideramos que, entre SCRUM y XP, SCRUM se acomoda mas a nuestra formación como equipo, ya que para XP se requiere que el equipo tenga programadores experimentados en el tema a tratar.

Cabe destacar que en este anteproyecto se han logrado identificar una basta cantidad de tareas (las que serán detalladas en etapas siguientes) que deben ser realizadas para poder llevar a cabo el proyecto y que su funcionalidad sea la correcta, es por esto que se estima conveniente hacer uso de SCRUM, ya que en XP la programación debe ser en parejas, lo que podría ralentizar las fechas definidas y

en consecuencia el retraso del proyecto como tal. En cambio, SCRUM establece que la programación debe ser de manera individual, lo que permite repartir más tareas simultáneamente, siendo favorable considerando que solo se cuenta con un equipo de 5 integrantes.

El desarrollo técnico se basará en OMT++, una metodología de análisis y diseño orientado a objetos.

1.5.2. Herramientas de desarrollo.

En la Tabla 1 se nombran las herramientas de desarrollo preliminares con las que se trabajará en el proyecto:

Tabla 1: Herramientas de desarrollo.	
Entorno	Descripción de Herramienta
Sistema Operativo	<ul style="list-style-type: none"> ▪ Microsoft Windows 10 ▪ GNU Linux Ubuntu 16.04 ▪ GNU Linux Fedora 28
IDE Desarrollo	<ul style="list-style-type: none"> ▪ Sublime Text 3 ▪ Visual Studio 1.27
Software de programación	<ul style="list-style-type: none"> ▪ PHP 7.3.1
Base de datos y servidor Web	<ul style="list-style-type: none"> ▪ PostgreSQL 10 ▪ XAMPP 7.3.0
Arquitectura y diseño	<ul style="list-style-type: none"> ▪ pgAdmin 4 v3.4

La Tabla 2 resumen el hardware que se utilizará para desarrollar el proyecto.

Tabla 2: Hardware de desarrollo.			
Computador	Procesador	RAM	Almacenamiento
Notebook	Intel Core i7-7700 HQ	16 GB	1 TB + 256GB SSD
Notebook	Intel Core i7-7700 HQ	12 GB	1 TB + 128GB SSD
Notebook	Intel Core i5-6700 HQ	12 GB	1 TB + 256GB SSD
Notebook	Intel Core i5-6700 HQ	8 GB	1 TB + 128GB SSD
Notebook	Intel Core i5-6700 HQ	8 GB	1 TB

1.5.3. Ambiente de desarrollo.

Dado que se utilizará una metodología de desarrollo ágil, se trabajará principalmente en lugares que brinden estancia adecuada para todo el equipo de desarrollo. Se escogió la biblioteca central para

reuniones, puesto que posee mesas de trabajo colaborativo, pizarras y acceso a internet. En ocasiones en que se junte parte del equipo también se eligió la biblioteca del Departamento de Matemáticas y Ciencia de la Computación, que posee salas privadas de distracción aptas para realizar trabajo más focalizado.

Para mantener el trabajo sincronizado y organizado, se utilizarán múltiples herramientas de trabajo colaborativo online, como Trello, que permite establecer listas ToDo del proyecto y Slack que permite mantener al equipo comunicado, organizado y actualizado en lo que respecta al proyecto.

1.6. Plan de trabajo.

A continuación se presenta las planificaciones a realizar para *Buscatdog*.

1.6.1. Planificación de entrevistas.

Se establece que, ya que el software no se está desarrollando para un cliente en particular, sino que para muchos posibles usuarios, es más conveniente y se alcanza a más personas si es que se realizan encuestas online para establecer los requerimientos próximos.

Por otro lado, es necesario tener reuniones tipo consultorías en las herramientas de desarrollo antes mencionadas, para así hacer un correcto uso de estas.

Tabla 3: Consultorías

Fecha	Tema de consultoría	Participantes
12/11/18	API Google Maps	Francisco Moreno
26/11/18	Retroalimentación sobre desarrollo orientado a objetos	ScrumMaster Darío Palminio

1.6.2. Planificación del proyecto.

La planificación del proyecto se realiza en base a la metodología SCRUM. El detalle de esta se encuentra en *apéndice: sección D*.

1.7. Conclusiones.

El presente informe describe los aspectos fundamentales del anteproyecto del sistema Buscatdog, el cual es un software que permite encontrar mascotas perdidas y abandonadas. Esto se logra al crear publicaciones de mascotas perdidas y abandonadas en una plataforma web, de manera que otros usuarios puedan ver estas publicaciones.

El sistema además puede mostrar la posible ubicación de la mascota en un mapa. La finalidad del sistema es que los dueños de mascotas encuentren a sus más leales compañeros.

Para el desarrollo del proyecto se optará por una metodología ágil, en particular SCRUM, donde se han planificado dos sprints. La duración del proyecto está comprendida entre el 21 de octubre y 21 de enero. El desarrollo técnico se basará en una versión simplificada de OMT++.

Se debe destacar que al adoptar una metodología ágil, el equipo de desarrollo estará preparado para enfrentar cambios en cualquier instancia del proyecto, por lo mismo la carta Gantt no será rígida, y sufrirá cambios dependiendo de las prioridades que el equipo estime.

Es fundamental además mencionar que como equipo se abogará por una mentalidad ágil, donde cada miembro se comprometerá a participar activamente en el desarrollo del proyecto y además la comunicación simétrica prime por sobre todo, pues cada miembro del equipo será fundamental para el desarrollo exitoso del proyecto de cara a la etapa de análisis de requerimientos y casos de uso.

Capítulo 2.

2. ANÁLISIS DE REQUERIMIENTOS Y CASOS DE USO.

2.1. Introducción.

Esta fase de conceptualización tiene como objetivo obtener los requisitos esenciales del sistema con el propósito de identificar y documentar lo necesario a fin de que este sea de fácil lectura tanto para clientes y equipo de desarrollo.

Es por esto que se presentan los requisitos de implementación, funcionales y no funcionales del sistema *Buscatdog*. Los requisitos funcionales explican el funcionamiento del sistema a través de los casos de usos, mientras que los requisitos no funcionales dan a conocer los parámetros que limitan un software.

Así, en este capítulo se detallará el análisis de requerimientos realizado, la especificación de casos de usos y las conclusiones pertinentes a lo que corresponde esta etapa.

2.2. Análisis de requerimientos.

Esta sección da a conocer el análisis de requerimientos del sistema, categorizados en funcionales, no funcionales y de implementación.

2.2.1. Requerimientos funcionales

- Crear publicación mascotas perdidas.
- Crear publicación mascotas abandonadas.
- Registrar usuario.
- Actualizar publicaciones de mascotas perdidas.
- Listar publicaciones según estado (perdido, encontrado).
- Listar publicaciones según geolocalización.
- Visualizar estimación de la posición de la mascota.

2.2.2. Requerimientos no funcionales

- El sistema opera 24/7.
- El software se mantendrá como código abierto.
- El diseño de la página será responsive.
- Se realizará mantenimiento de tablas básicas.

2.2.3. Requerimientos de implementación

- El sistema debe funcionar en cualquier navegador web.
- El software necesita conexión a internet.
- El sistema debe ocupar el motor de base de datos PostgreSQL de forma objeto-relacional.

2.3. Casos de uso.

Esta sección da a conocer el Diagrama de casos de uso del sistema *Buscatdog*, además de una especificación detallada de los casos de uso.

2.3.1. Diagrama de casos de uso.

En la figura 2 se muestra el diagrama de casos de uso del sistema *Buscatdog*.

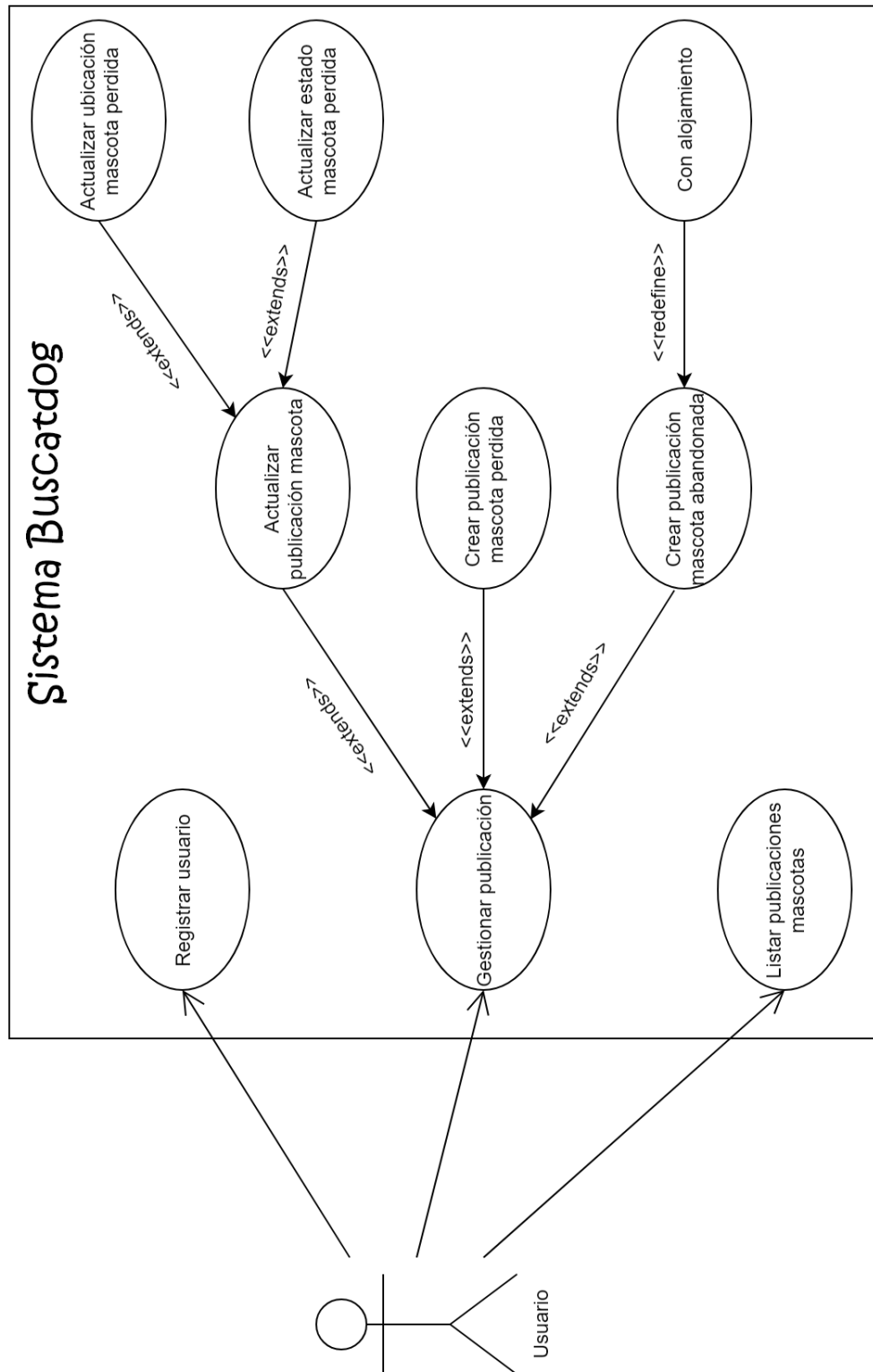


Figura 2: Diagrama de casos de uso del sistema Buscatdog.

2.3.2. Especificación de casos de uso.

A continuación, de las tablas 4 a la 12 se muestran las especificaciones de casos de uso del sistema *Buscatdog*.

Tabla 4: Especificación de caso de uso n°1 - Gestión Publicación.

Caso de uso	Gestión Publicación.
Resumen	Gestiona las publicaciones definidas en los casos de uso n°2, 3 y 4.
Frecuencia	Ilimitada.
Actores	Usuario.
Precondiciones	No tiene.
Descripción Flujo normal (escenario exitoso)	
Responsabilidad del actor: 2. Selecciona la opción crear publicación mascota perdida o crear publicación mascota abandonada.	Responsabilidad del sistema: 1. Habilita opción crear publicación mascota perdida, crear publicación mascota abandonada. 3. Redirige al caso de uso n°2, 3 o 4. 4. Fin caso de uso.
Excepciones	No tiene.
Postcondiciones	Publicación gestionada.

Tabla 5: Especificación de caso de uso n°2 - Publicar mascota perdida.

Caso de uso	Publicar mascota perdida.
Resumen	El usuario ingresa una publicación de mascota perdida.
Frecuencia	Ilimitada.
Actores	Usuario.
Precondiciones	El usuario debe estar registrado en el sistema.
Descripción Flujo normal (escenario exitoso)	
Responsabilidad del actor: 2. Ingresa los atributos dirección de pérdida, datos de la mascota (nombre mascota, raza, color, tamaño, personalidad, accesorios) e imagen de la mascota [EX01]. 3. Confirma ingreso de publicación de mascota perdida.	Responsabilidad del sistema: 1. Habilita formulario de ingreso de publicación de mascota perdida. 4.Fin caso de uso.
Excepciones	[EX01] El usuario no ingresa los campos obligatorios. El sistema despliega mensaje "Debe completar campos obligatorios".
Postcondiciones	Mascota perdida publicada.

Tabla 6: Especificación de caso de uso n°3 - Publicar mascota abandonada.

Caso de uso	Publicar mascota abandonada.
Resumen	El usuario ingresa una publicación de mascota abandonada.
Frecuencia	Ilimitada.
Actores	Usuario.
Precondiciones	No tiene.
Descripción Flujo normal (escenario exitoso)	
Responsabilidad del actor: 2. Ingresa los atributos dirección de pérdida, datos de la mascota (nombre mascota, raza, color, tamaño, personalidad, accesorios) e imagen de la mascota [EX01]. 3. Confirma ingreso de publicación de mascota abandonada.	Responsabilidad del sistema: 1. Habilita formulario de ingreso de publicación de mascota abandonada. 4. Fin caso de uso.
Excepciones	[EX01] El usuario no ingresa los campos obligatorios. El sistema despliega mensaje "Debe completar campos obligatorios".
Postcondiciones	PMascota abandonada publicada.

Tabla 7: Especificación de caso de uso n°4 - Alojar mascota abandonada.

Caso de uso	Alojar mascota abandonada.
Resumen	El usuario determina si aloja a la mascota abandonada.
Frecuencia	Ilimitada.
Actores	Usuario.
Precondiciones	Seleccionar opción de alojamiento en caso de uso n°3. El usuario debe haber iniciado sesión.
Descripción Flujo normal (escenario exitoso)	
Responsabilidad del actor: 2. Selecciona opción de alojamiento 4. Ingresa los atributos dirección y duración del alojamiento [EX01]. 5. Confirma ingreso de publicación de mascota abandonada.	Responsabilidad del sistema: 1. Habilitar opción de alojamiento mascota abandonada. 3. Habilita formulario de alojamiento. 6. Fin caso de uso
Excepciones	[EX01] El usuario no ingresa campos obligatorios. El sistema despliega mensaje "Debe completar campos obligatorios"
Postcondiciones	Mascota abandonada alojada.

Tabla 8: Especificación de caso de uso n°5 - Actualizar publicación mascota.

Caso de uso	Actualizar publicación mascota.	
Resumen	Actualiza las publicaciones de mascotas definidas en los casos de uso n°6 y n°7.	
Frecuencia	Ilimitada.	
Actores	Usuario.	
Precondiciones	No tiene.	
Descripción Flujo normal (escenario exitoso)		
Responsabilidad del actor: 2. Selecciona la opción actualizar ubicación de mascota perdida o actualizar estado de mascota perdida.		Responsabilidad del sistema: 1. Habilita opción actualizar ubicación de mascota perdida y actualizar estado de mascota perdida. 3. Redirige al caso de uso n°6 o 7. 4. Fin del caso de uso
Excepciones	No tiene.	
Postcondiciones	Publicación actualizada.	

Tabla 9: Especificación de caso de uso n°6 - Actualizar estado de mascota perdida.

Caso de uso	Actualizar estado mascota perdida.
Resumen	Actualiza el estado de una publicación de mascota perdida.
Frecuencia	Ilimitada.
Actores	Usuario.
Precondiciones	La publicación debe estar creada. La publicación debe estar activa. El usuario debe haber iniciado sesión.
Descripción Flujo normal (escenario exitoso)	
Responsabilidad del actor: 2. Selecciona opción mascota encontrada. 3. Confirma la actualización del estado de publicación de una mascota perdida.	Responsabilidad del sistema: 1. Habilita opción actualizar estado de mascota perdida. 4. Fin caso de uso.
Excepciones	No tiene.
Postcondiciones	Estado de la publicación actualizado.

Tabla 10: Especificación de caso de uso n°7 - Actualizar ubicación mascota perdida

Caso de uso	Actualizar ubicación mascota perdida.
Resumen	Se actualiza la ubicación de la mascota según el último avistamiento.
Frecuencia	Ilimitada.
Actores	Usuario.
Precondiciones	La publicación debe estar creada. La publicación debe estar activa.

Descripción Flujo normal (escenario exitoso)

Responsabilidad del actor:	Responsabilidad del sistema:
2. Selecciona opción actualizar ubicación de mascota perdida.	1. Habilita opción actualizar ubicación de mascota perdida.
3. Ingresa nueva ubicación.	
4. Confirma la actualización de la ubicación de la publicación de la mascota perdida.	
	5. Fin de caso de uso.
Excepciones	No tiene.
Postcondiciones	Ubicación de la publicación actualizado.

Tabla 11: Especificación de caso de uso n°8 - Listar publicaciones mascotas.

Caso de uso	Listar publicaciones mascotas.
Resumen	Lista publicaciones.
Frecuencia	Ilimitada.
Actores	Usuario.
Precondiciones	No tiene.

Descripción Flujo normal (escenario exitoso)

Responsabilidad del actor:	Responsabilidad del sistema:
2. Selecciona la opción especie o ubicación.	1. Habilita opciones por especie o ubicación.
	3. Despliega lista de publicaciones según opción seleccionada.
	4. Fin caso de uso.
Excepciones	No tiene.
Postcondiciones	Publicaciones de mascotas listadas.

Tabla 12: Especificación de caso de uso n°9 - Registrar usuario.

Caso de uso	Registrar usuario.
Resumen	Registra usuario en el sistema.
Frecuencia	Ilimitada.
Actores	Usuario.
Precondiciones	No tiene.
Descripción Flujo normal (escenario exitoso)	
Responsabilidad del actor: 2. Ingresa los atributos nombre, apellido, email, contraseña y teléfono [EXC01]. 3. Selecciona opción registrar.	Responsabilidad del sistema: 1. Habilita formulario de registro de usuario. 3. Fin del caso de uso.
Excepciones	[Exc01]: El usuario no ingresa los campos obligatorios. El sistema despliega mensaje "Debe completar campos obligatorios".
Postcondiciones	Usuario registrado.

2.4. Conclusiones.

La especificación de requerimientos es un proceso vital en el desarrollo de un Software, pues en él se plasman los cimientos de todo lo que viene en las etapas posteriores.

En particular en OMT++ esta etapa corresponde a la conceptualización o captura de requerimientos. Aquí es donde se enuncia el problema y se capturan los requerimientos tanto funcionales como no funcionales. La importancia del trabajo colaborativo es fundamental, pues otorga diferentes puntos de vista los cuales se analizan para llegar a un consenso y así establecer requerimientos de manera certera, sin caer en la ambigüedad, además de enriquecer el hecho de que todos los miembros del equipo comprendan los requerimientos en su totalidad.

Con esto ya planteado se procedió a identificar los roles que interactúan con el sistema y así plantear los principales casos de uso del sistema Buscatdog para luego plasmarlos en el diagrama de casos de uso principal. Además, se mencionaron las especificaciones de cada uno de los casos de uso identificados, de manera que sean consistentes con el diagrama construido.

En cuanto al desarrollo ágil, el trabajo se ha desarrollado como se planteó previamente, en reuniones semanales donde todo el equipo trabaja en lo mismo, sin repartir tareas de manera individual, sin embargo aún se siguen algunas prácticas de la metodología tradicional, como la falta de iniciativa a la hora de comenzar a trabajar, o romper el esquema de jerarquía vertical al momento de enfrentar labores de desarrollo, todo esto se tratará de corregir en la siguiente etapa que corresponde a análisis orientado a objetos.

Capítulo 3.

3. ANÁLISIS ORIENTADO A OBJETOS OMT++.

3.1. Introducción.

En este informe se presenta el Análisis Orientado a Objetos del sistema *Buscatdog*.

Específicamente, se da a conocer el Análisis de Objetos, a través del Modelo de Objetos del análisis; el Análisis de Comportamiento, a través de la Lista de Operaciones; y por último, la Especificación de la interfaz de usuario, especificando los Diagramas de Diálogo y la Especificación de componentes.

Los artefactos anteriormente señalados, corresponden a aquellas funcionalidades que se implementarán en el Sprint 1, la cuales corresponden a la creación de publicación de mascotas perdidas, creación de publicación de mascotas encontradas e inicio de sesión con el registro de usuario.

3.2. Análisis OO.

Esta sección se presentan los artefactos del análisis de objetos, comportamiento y la especificación de la interfaz de usuario.

3.2.1. Análisis de objetos

A continuación se detalla el modelo de objetos del análisis del sistema *Buscatdog*. Para esto se utilizó los casos de uso del segundo avance del proyecto.

3.2.1.1 Modelo de objetos del análisis.

En la figura 3 se presenta el modelo de objetos del sistema *Buscatdog*. Posterior a ello se mostrarán las respectivas tablas que representan el diccionario de datos de cada una de las clases que lo componen.

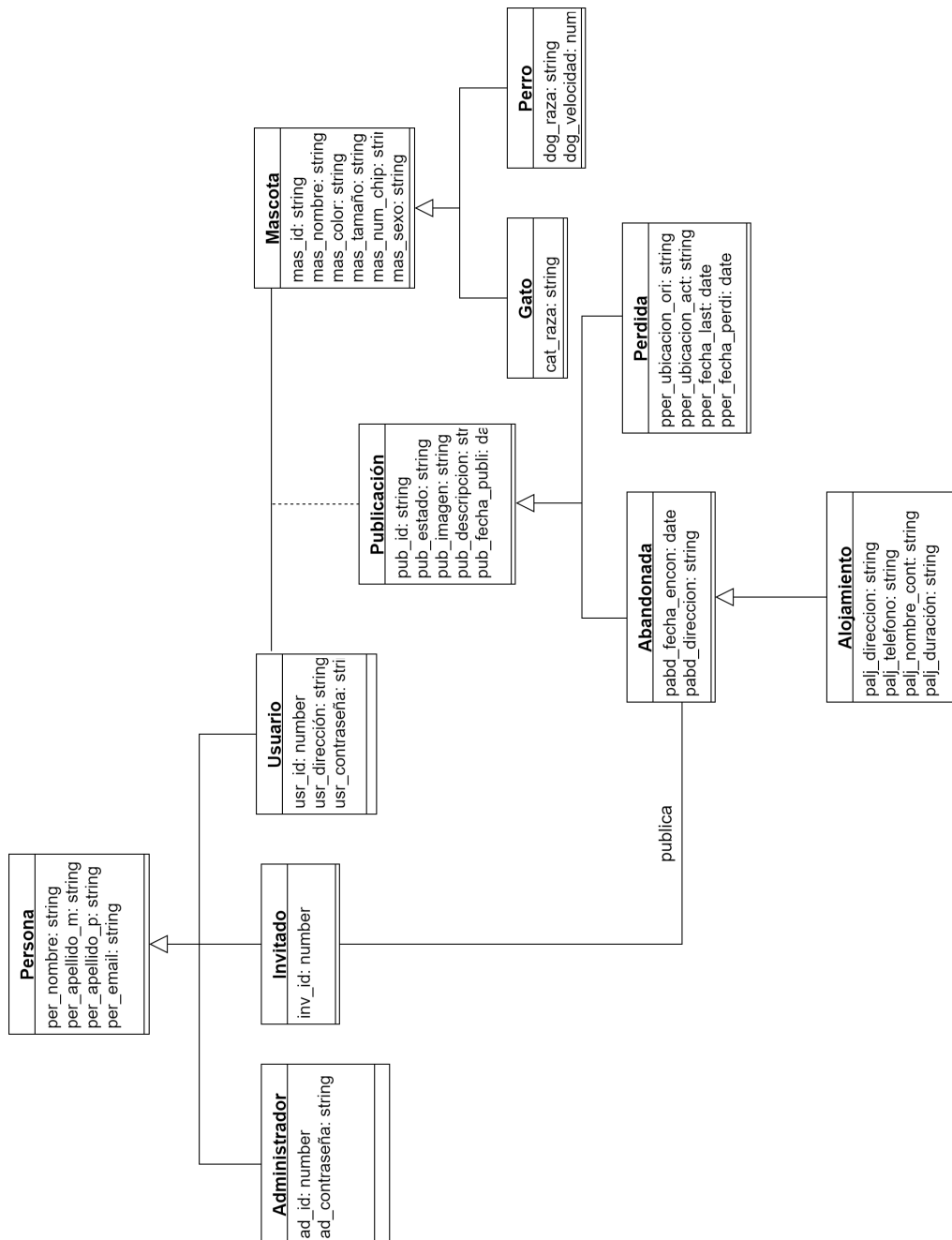


Figura 3: Modelo de objetos del análisis del sistema Buscatdog

Las tablas 13 a la 23 muestran el diccionario de datos del modelo de análisis de *Buscatdog* para cada una de las clases presentadas. Además se presentan ejemplos de instancias de las clases mencionadas.

Tabla 13: Clase Persona.

Clase	Descripción	Atributos
Persona	Persona genérica	per_nombre: Nombre per_apellido_m: Apellido materno per_apellido_p: Apellido paterno per_email: E-mail

Ejemplos de instancias:

- Andrea - Soto - Castañeda - ascasta@gmail.com
- Mauricio - Molina - Fuentes - mmfuentes101@gmail.com

Tabla 14: Clase Administrador.

Clase	Descripción	Atributos
Administrador	Administrador del sistema	ad_id: Id. de administrador ad_contraseña: Contraseña

Ejemplos de instancias:

- Óscar - Campos - Quezada - soyadmin@gmail.com - 01 - lkj258
- Federico - Sánchez - Molina - fede1018@hotmail.com - 02 - ytrew88

Tabla 15: Clase Invitado.

Clase	Descripción	Atributos
Invitado	Usuario que requiere registro corto	inv_id: Id invitado

Ejemplos de instancias:

- Carolina - Herrera - Castañeda - caroanimalista@gmail.com - 2536
- Felipe - Soto - Fuentes - felipesoyyo@gmail.com - 1568

Tabla 16: Clase Usuario.

Clase	Descripción	Atributos
Usuario	Usuario del sistema que requiere registro	usr_id: Id usuario usr_dirección: Dirección usr_contraseña: Contraseña

Ejemplos de instancias:

- Sofia - Navarro - Castañeda - sofi101@gmail.com - 363 - Los Rosales 789 - 456lop5
- Mauricio - Baeza - Fuentes - mauricat@gmail.com - 442 - Uvas del viento 154 - supersegura37

Tabla 17: Clase Mascota.

Clase	Descripción	Atributos
Mascota	Datos de una mascota genérica.	mas_id: Id de la mascota. mas_nombre: nombre de la mascota. mas_color: color del pelaje de la mascota. mas_tamaño: tamaño de la mascota. mas_num_chip: numero de chip de la mascota. massexo: Sexo de la mascota. mas_caracter: Personalidad de la mascota.

Ejemplos de instancias:

- 321 - Spike - Negro - Mediano - 5258965 - Macho - Manso
- 653 - Chiki - Café claro - Chico - 8858915 - Hembra - Inquieto

Tabla 18: Clase Perro.

Clase	Descripción	Atributos
Perro	Clase tipo perro que hereda de mascota.	dog_raza: Raza del perro. dog_velocidad: Velocidad a la que corre tal raza de perro.

Ejemplos de instancias:

- 321 - Spike - Blanco - Mediano - 5258965 - Macho - Manso - Labrador - 6
- 653 - Chiki - Café claro - Chico - 8858915 - Hembra - Inquieto - Mestizo - 4

Tabla 19: Clase Gato.

Clase	Descripción	Atributos
Gato	Clase tipo gato hereda de mascota.	cat_raza: Raza del gato.

Ejemplos de instancias:

- 543 - Lucía - Negra - Chica - 5252065 - Hembra - Violenta - Bombay
- 653 - Perro - Café claro - Chico - 8123915 - Macho - Manso - Bengali

Tabla 20: Clase Publicación.

Clase	Descripción	Atributos
Publicación	Publicación genérica	pub_id: Id de la publicación pub_estado: Estado de mascota asociada a la publicación pub_imagen: Imágen de la mascota pub_descripcion: Descripción de la mascota asociada a la publicación pub_fecha_publici: Fecha de creación de publicación usr_id: Usuario que crea publicación mas_id: Mascota asociada a la publicación

Ejemplos de instancias:

- 365 - Perdido - “/img/365.jpg” - “Perrita amigable, con herida en la pata” - 02/12/2018 - 368 - 579
- 125 - Encontrado - “/img/125.jpg” - “Cola cortada, con herida en la oreja” - 11/11/2018 - 864 - 282

Tabla 21: Clase Perdida.

Clase	Descripción	Atributos
Perdida	Publicación de mascota perdida.	pper_ubicación_ori : Ubicación en donde la mascota fue extraviada. pper_ubicación_act : Estimación de la ubicación actual de la mascota. pper_fecha_last : Fecha última vista de mascota. pper_fecha_perdi : Fecha en que se perdió la mascota. usr_id : Usuario que crea publicación mas_id : Mascota asociada a la publicación

Ejemplos de instancias:

- 653 - Perdido - “/img/36445.jpg” - “Perrita amigable, con herida en la pata” - 02/12/2018 - Perpetua Freire 1223 - Vicuña Mackenna 1550 - 02/12/2018 - 02/12/2018 - 495 - 475
- 887 - Perdido - “/img/12245.jpg” - “Cola cortada, con herida en la oreja” - 11/11/2018 - 09/11/2018 - Vitacura - MT - Vitacura 1358 - 05/01/2019 - 09/11/2018 - 987 - 12254

Tabla 22: Clase Abandonada.

Clase	Descripción	Atributos
Abandonada	Publicación de mascota abandonada.	pabd_fecha_encon : Fecha en que la mascota fue encontrada. pabd_dirección : Dirección en donde la mascota fue encontrada. usr_id : Usuario que crea publicación mas_id : Mascota asociada a la publicación

Ejemplos de instancias:

- 365 - Abandonada - “/img/365.jpg” - “Perrita amigable, con herida en la pata” - 01/12/2018 - Gran Avenida 1256 - 764 - 215
- 125 - Abandonada - “/img/125.jpg” - “Cola cortada, con herida en la oreja” - 11/11/2018 - Padre Hurtado 6325 - 79756 - 1243

Tabla 23: Clase Alojamiento.

Clase	Descripción	Atributos
Alojamiento	Alojamiento para publicación de mascota abandonada.	palj_dirección : Dirección del alojamiento. palj_teléfono : Teléfono de contacto. palj_nombre_cont : Nombre del contacto. palj_duración : Duración del alojamiento. usr_id : Usuario que crea publicación mas_id : Mascota asociada a la publicación

Ejemplos de instancias:

- 774 - Alojado - “/img/653.jpg” - “Perrita tierna, con herida en el cuello” - 10/10/2018 - Alonso de Ercilla 8899 - +5636512569 - Carmen - 2 semanas

- 885 - Alojado - “/img/447.jpg” - “Cola chueca, mancha en el ojo derecho” - 08/05/2018 - Las Sóforas 336 - +569986652 - Julio - 4 semanas

3.2.2. Análisis del comportamiento.

A continuación, se da a conocer el análisis de comportamiento donde se mostrará la especificación de operaciones del sistema, así como también la especificación de interfaz de usuario.

3.2.2.1 Especificación de operaciones.

A continuación, en la tabla 24 se mostrará el listado de las operaciones del sistema *Buscatdog*.

Tabla 24: Lista de operaciones del sistema Buscatdog.

Nro	Operación
1	Seleccionar crear publicación mascota perdida.
2	Seleccionar crear publicación mascota abandonada.
3	Ingresar nombre de mascota.
4	Ingresar raza de mascota.
5	Ingresar color de mascota.
6	Ingresar tamaño de mascota.
7	Ingresar carácter de mascota.
8	Ingresar descripción de mascota.
9	Ingresar dirección.
10	Adjuntar una imagen de mascota.
11	Aceptar mensaje excepción.
12	Seleccionar crear publicación mascota abandonada con alojamiento.
13	Ingresar duración de alojamiento.
14	Seleccionar actualizar publicación mascota perdida.
15	Seleccionar actualizar publicación mascota abandonada.
16	Seleccionar modificar estado de mascota perdida.
17	Ingresar ubicación de mascota.
18	Seleccionar opción especie.
19	Seleccionar opción ubicación.
20	Seleccionar iniciar sesión.
21	Ingresar E-mail.
22	Ingresar contraseña.
23	Seleccionar crear usuario.
24	Ingresar nombre de usuario.
25	Ingresar apellido paterno de usuario.
26	Ingresar apellido materno de usuario.
27	Ingresar teléfono.
28	Ingresar nombre contacto.
29	Salir del sistema.
30	Ingresar sexo de mascota.
31	Aceptar mensaje.
32	Ingresar contraseña nueva.
33	Seleccionar estado publicación.
34	Seleccionar visualizar publicación.
35	Seleccionar opción cambio de contraseña.
36	Seleccionar filtrar publicaciones.
37	Seleccionar actualizar estado publicación.
38	Seleccionar actualizar ubicación publicación.
39	Listar publicación usuario.

3.2.3. Especificación de la interfaz de usuario.

En esta sección se presentará una primera aproximación de la interfaz de usuario, mediante los artefactos: construcción de diagramas de diálogo y especificación de componentes.

3.2.3.1 Diagramas de diálogo.

A continuación, en la figura 4 se muestra el diagrama de diálogo del sistema *Buscatdog*.

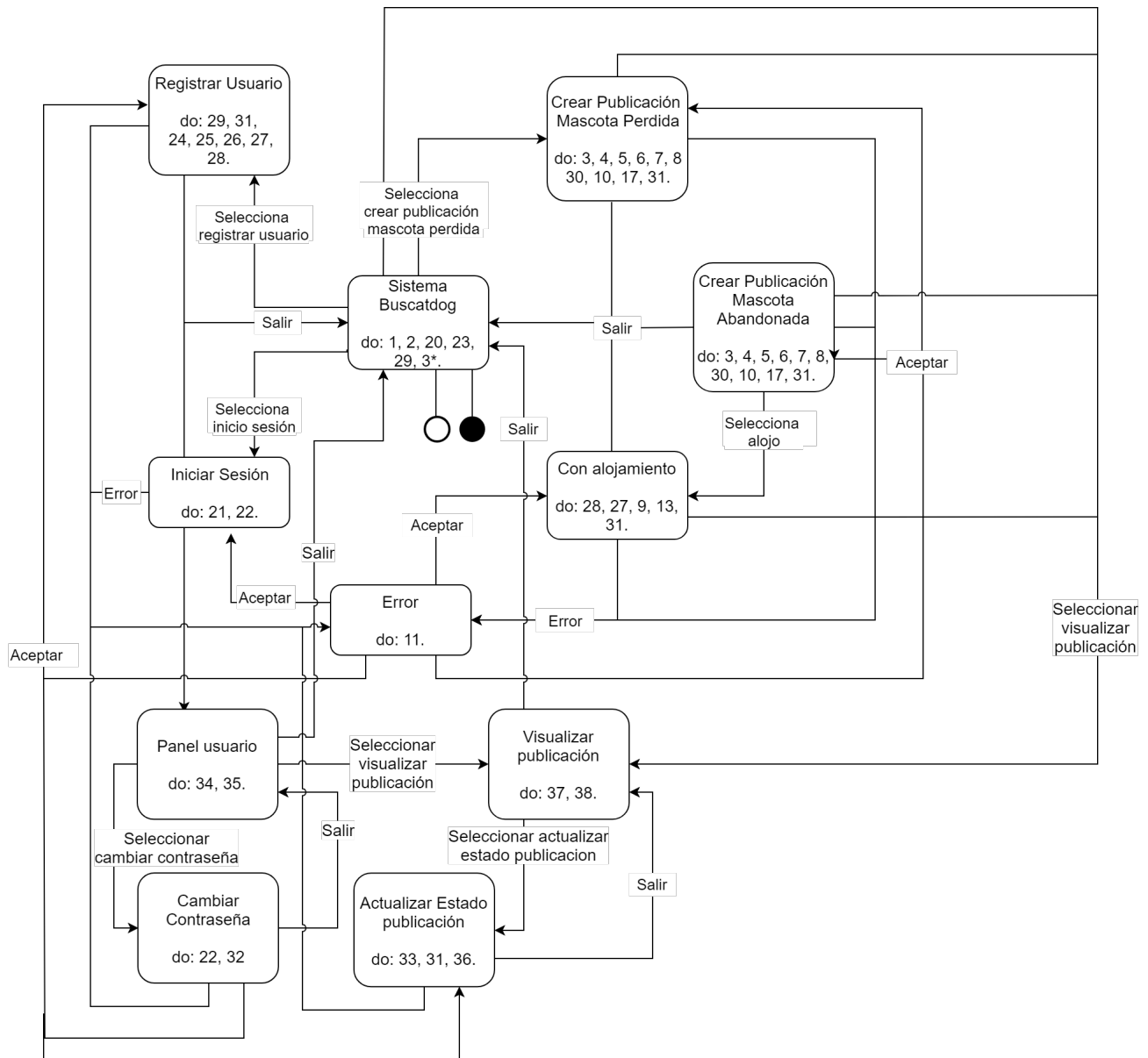


Figura 4: Diagrama de diálogo del sistema Buscatdog.

3.2.3.2 Especificación de componente.

A continuación, desde la figura 5 a la 15 se describen los diagramas de componentes junto a sus respectivas descripciones de comportamiento encontradas desde la tabla 16 a la 35.

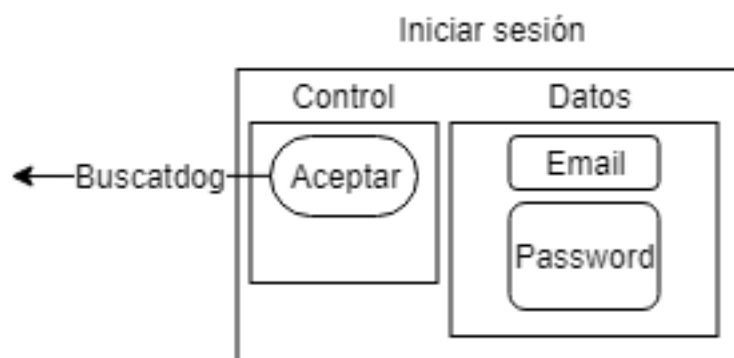


Figura 5: Diagrama de control Inicio Sesión.

Tabla 25: Descripción de comportamiento Inicio Sesión

Componente	Manipulación	Retroalimentación
Iniciar sesión	<ul style="list-style-type: none"> Seleccionar opción "aceptar" 	Entrada: <ul style="list-style-type: none"> E-mail Contraseña

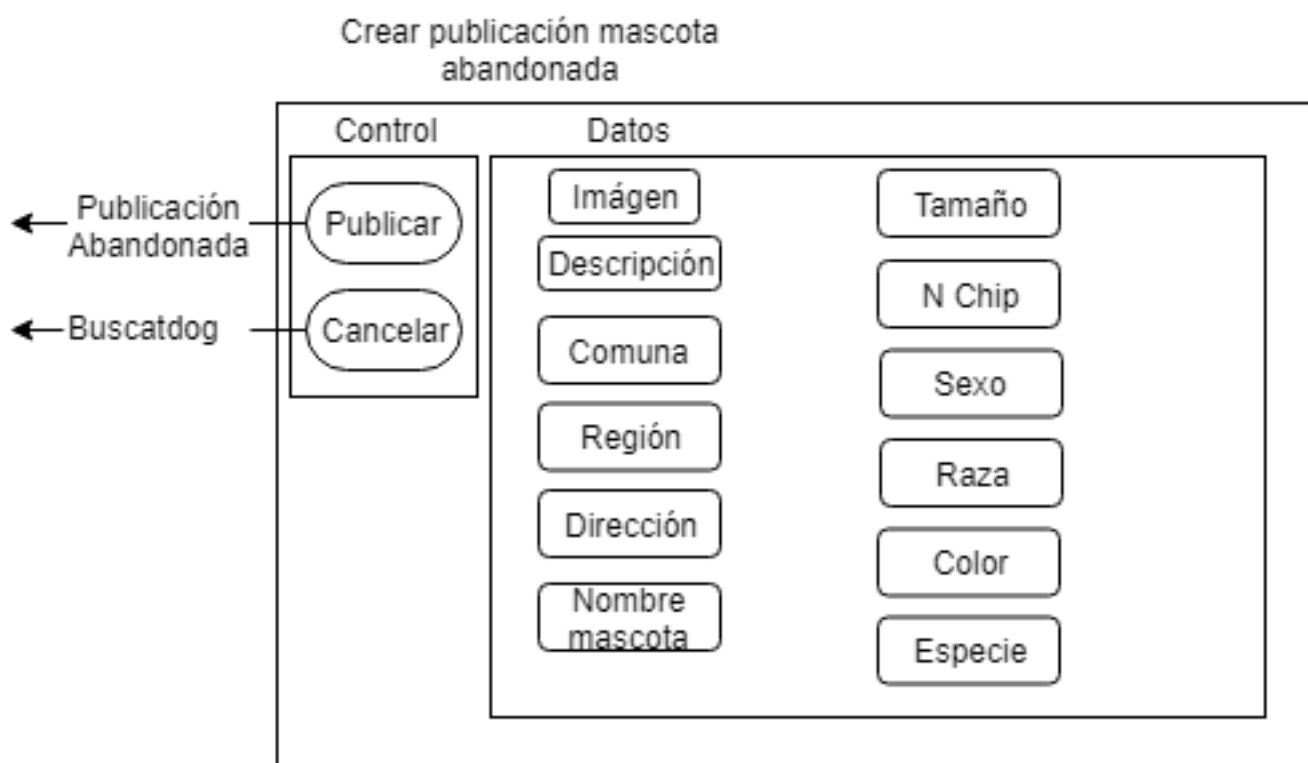


Figura 6: Diagrama de control Crear publicación mascota abandonada.

Tabla 26: Descripción de comportamiento Crear publicación mascota abandonada

Componente	Manipulación	Retroalimentación
Crear publicación mascota abandonada	<ul style="list-style-type: none"> ■ Seleccionar opción “Publicar” ■ Seleccionar opción “Cancelar” 	<p>Entrada:</p> <ul style="list-style-type: none"> ■ Imagen ■ Descripción ■ Nombre Mascota ■ Comuna ■ Región ■ Dirección ■ Tamaño ■ N Chip ■ Sexo ■ Raza ■ Color ■ Especie



Figura 7: Diagrama de control Crear publicación mascota abandonada con alojamiento.

Tabla 27: Descripción de comportamiento Crear publicación mascota abandonada con alojamiento

Componente	Manipulación	Retroalimentación
Crear publicación mascota abandonada con alojamiento	<ul style="list-style-type: none"> ■ Seleccionar opción “Publicar” ■ Seleccionar opción “Cancelar” 	<p>Entrada:</p> <ul style="list-style-type: none"> ■ Imagen ■ Descripción ■ Fecha mascota encontrada ■ Comuna ■ Región ■ Dirección ■ Teléfono de contacto ■ Duración de alojamiento ■ Dirección de alojamiento ■ Nombre de contacto ■ Nombre mascota ■ Tamaño ■ N Chip ■ Sexo ■ Raza ■ Color ■ Especie

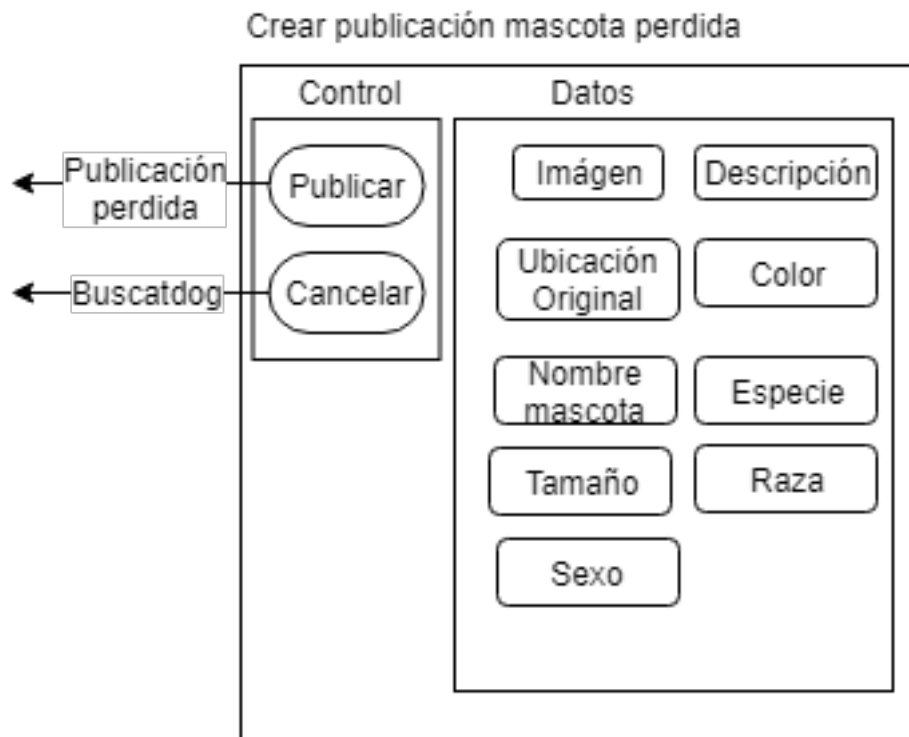


Figura 8: Diagrama de control Crear publicación mascota perdida.

Tabla 28: Descripción de comportamiento Crear publicación mascota perdida

Componente	Manipulación	Retroalimentación
Crear publicación mascota perdida	<ul style="list-style-type: none"> ■ Seleccionar opción “Publicar” ■ Seleccionar opción “Cancelar” 	<p>Entrada:</p> <ul style="list-style-type: none"> ■ Imagen ■ Descripción ■ Ubicación ■ Nombre mascota ■ Tamaño ■ N Chip ■ Sexo ■ Raza ■ Color ■ Especie

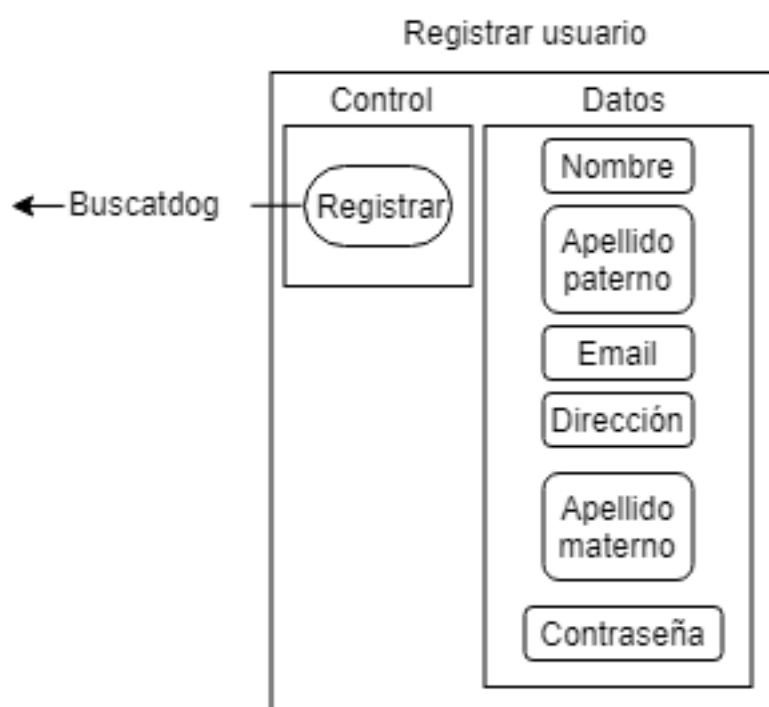


Figura 9: Diagrama de control registrar usuario.

Tabla 29: Descripción de comportamiento registrar usuario

Componente	Manipulación	Retroalimentación
Registrar usuario	<ul style="list-style-type: none"> ■ Seleccionar opción “Registrar” 	<p>Entrada:</p> <ul style="list-style-type: none"> ■ Nombre ■ Apellido paterno ■ Apellido materno ■ E-mail ■ Dirección ■ Contraseña

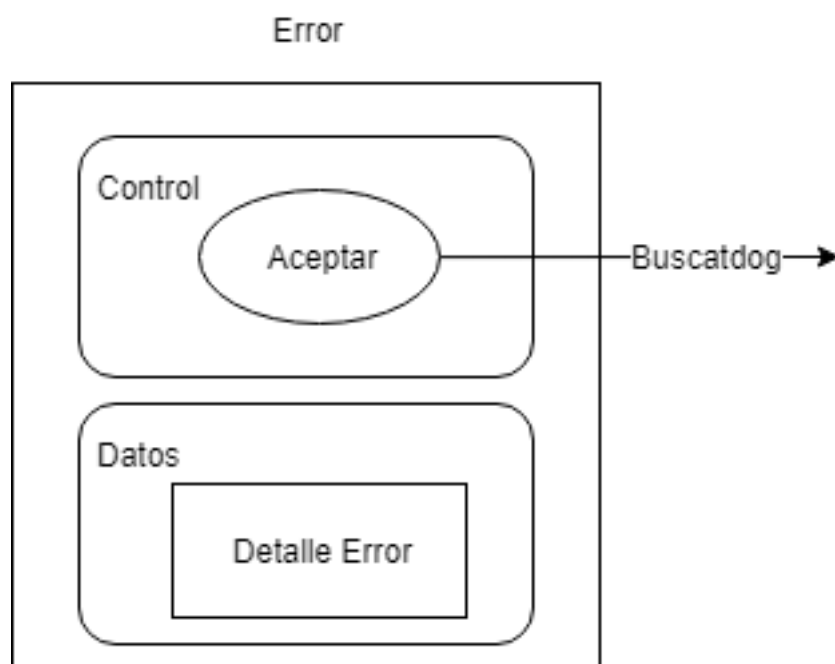


Figura 10: Diagrama de control error.

Tabla 30: Descripción de comportamiento error

Componente	Manipulación	Retroalimentación
Error	<ul style="list-style-type: none"> ■ Seleccionar “Aceptar” 	<p>Salida:</p> <ul style="list-style-type: none"> ■ Detalle Error

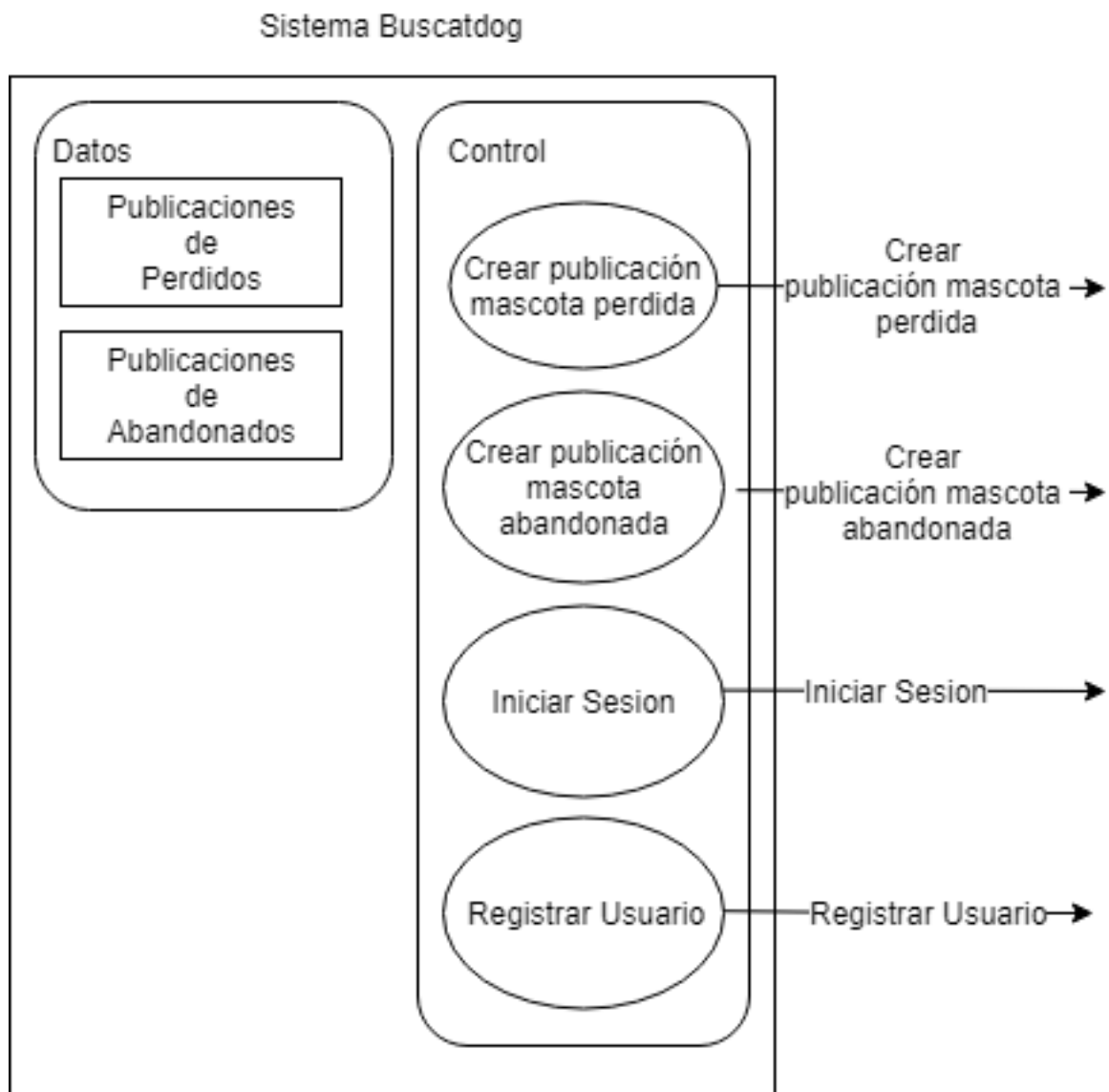


Figura 11: Diagrama de control Buscatdog.

Tabla 31: Descripción de comportamiento Buscatdog

Componente	Manipulación	Retroalimentación
Buscatdog	<ul style="list-style-type: none"> ■ Crear publicación mascota perdida ■ Crear publicación mascota abandonada ■ Iniciar sesión ■ Registrar usuario 	Salida: <ul style="list-style-type: none"> ■ Publicaciones mascotas perdidas ■ Publicaciones mascotas abandonadas

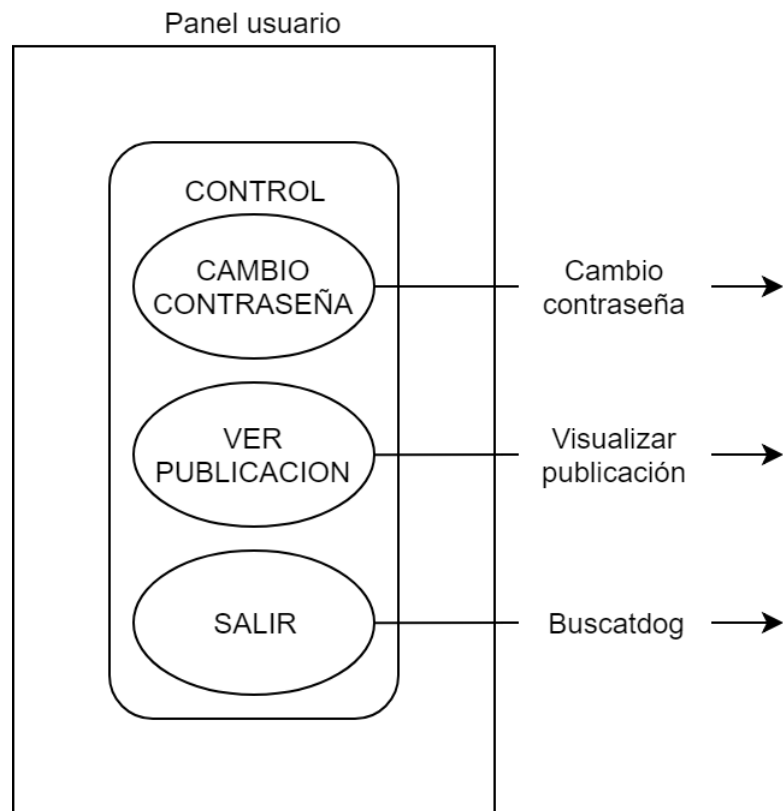


Figura 12: Diagrama de control Panel usuario.

Tabla 32: Descripción de comportamiento Panel usuario

Componente	Manipulación	Retroalimentación
Panel usuario	<ul style="list-style-type: none"> ■ Seleccionar "Cambio contraseña" ■ Seleccionar "Ver publicación" ■ Seleccionar "Salir" 	Salida: <ul style="list-style-type: none"> ■ Buscatdog ■ Cambio contraseña ■ Visualizar publicación

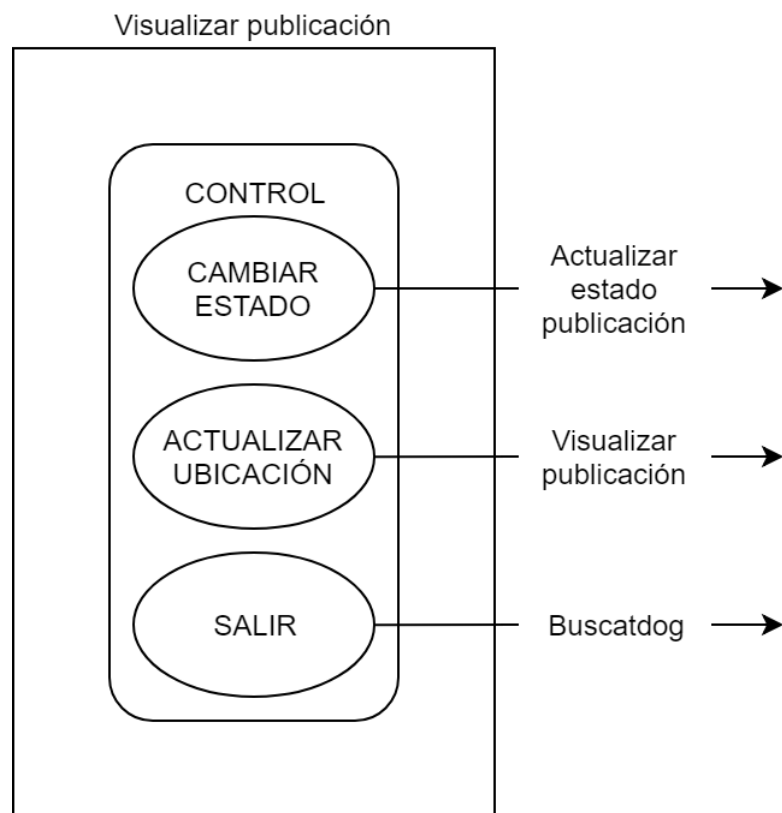


Figura 13: Diagrama de control Visualizar publicación.

Tabla 33: Descripción de comportamiento Visualizar publicación

Componente	Manipulación	Retroalimentación
Visualizar publicación	<ul style="list-style-type: none"> ■ Seleccionar "Cambiar estado" ■ Seleccionar "Actualizar ubicación" ■ Seleccionar "Salir" 	Salida: <ul style="list-style-type: none"> ■ Buscatdog ■ Actualizar estado publicación ■ Visualizar publicación

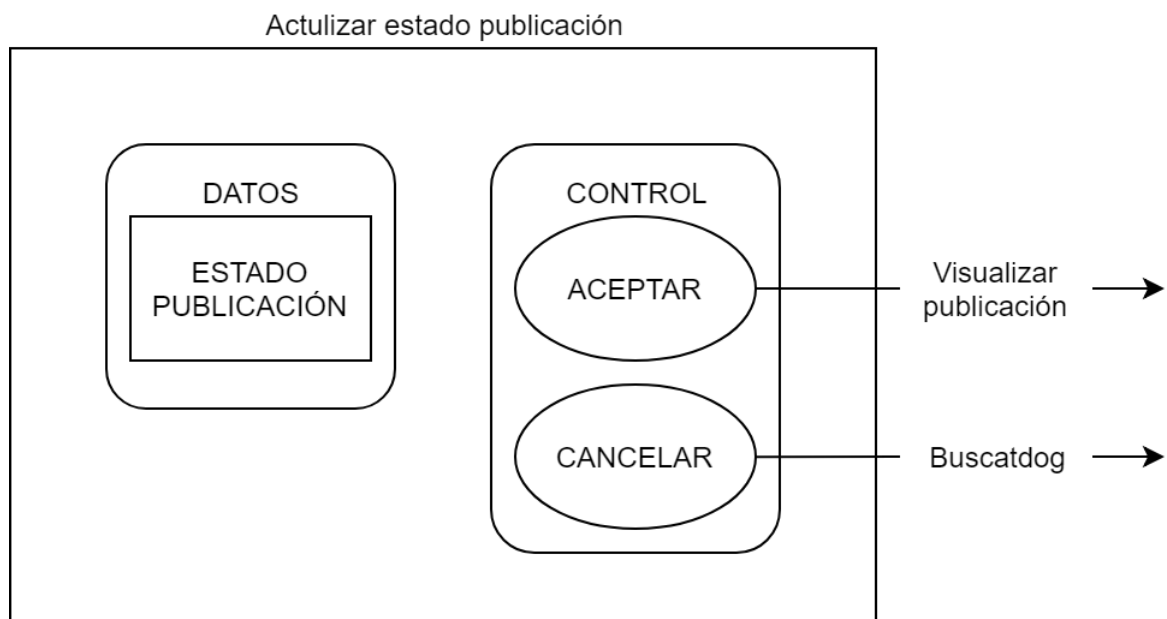


Figura 14: Diagrama de control Actualizar estado publicación.

Tabla 34: Descripción de comportamiento Actualizar estado publicación

Componente	Manipulación	Retroalimentación
Actualizar estado publicación	<ul style="list-style-type: none"> ▪ Seleccionar "aceptar" ▪ Seleccionar "cancelar" 	<p>Entrada:</p> <ul style="list-style-type: none"> ▪ Estado publicación <p>Salida:</p> <ul style="list-style-type: none"> ▪ Visualizar publicación ▪ Buscatdog

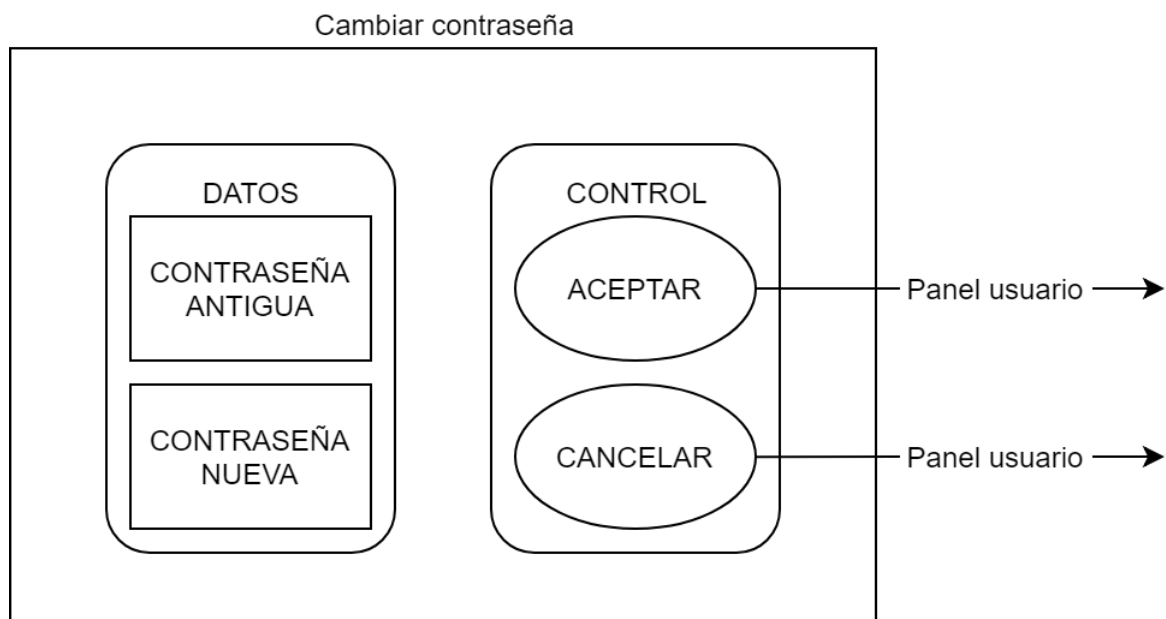


Figura 15: Diagrama de control Cambiar contraseña.

Tabla 35: Descripción de comportamiento Cambiar contraseña

Componente	Manipulación	Retroalimentación
Cambiar contraseña	<ul style="list-style-type: none"> ▪ Seleccionar "aceptar" ▪ Seleccionar "cancelar" 	Entrada: <ul style="list-style-type: none"> ▪ Contraseña antigua ▪ Contraseña nueva Salida: <ul style="list-style-type: none"> ▪ Panel usuario

3.3. Conclusiones.

Como se pudo apreciar, en este informe se presentaron tres artefactos correspondientes al Análisis Orientado a Objetos con sus definiciones correspondientes.

Estos artefactos son el Modelo de Objetos del Análisis del sistema completo, ya que es necesario dar un contexto a la solución general del problema. Luego se crea una lista de especificación de operaciones, la cual se genera en base a los casos de uso desarrollados con anterioridad a esta etapa, y por último, se determinan los diagramas de diálogos y especificaciones de componentes correspondientes al Sprint 1.

Al finalizar la etapa, se concluye como equipo que esta permite comprender mejor el dominio del problema al especificar las interacciones entre usuario y sistema de los requerimientos que conforman el Sprint 1 a través de los artefactos anteriormente mencionados. Estos sirven como base de cara a la siguiente etapa, Diseño Orientado a Objetos.

Capítulo 4.

4. DISEÑO ORIENTADO A OBJETOS OMT++.

4.1. Introducción.

En este informe se presenta el Diseño Orientado a Objetos del sistema *Buscatdog*.

Se transforma lo desarrollado en la fase anterior, Análisis Orientado a Objetos, para definir los métodos de las clases que serán utilizadas en la etapa de implementación del sistema.

Para esto diseñamos dos actividades principales, el diseño de objetos y el diseño de comportamiento, que produce el modelo de objetos de diseño y las trazas de evento respectivas, siendo éstos los artefactos de la fase de diseño.

4.2. Diseño OO.

Esta sección se presentan los artefactos del diseño de objetos y del diseño del comportamiento generados durante la etapa de diseño orientado a objetos utilizando OMT++.

4.2.1. Diseño de objetos.

A continuación, las figura 16 y 17 presentan el modelo de objeto del diseño del sistema Buscatdog utilizando el Modelo Vista Controlador *MVC*.

VISTA

CONTROLADOR

MODELO

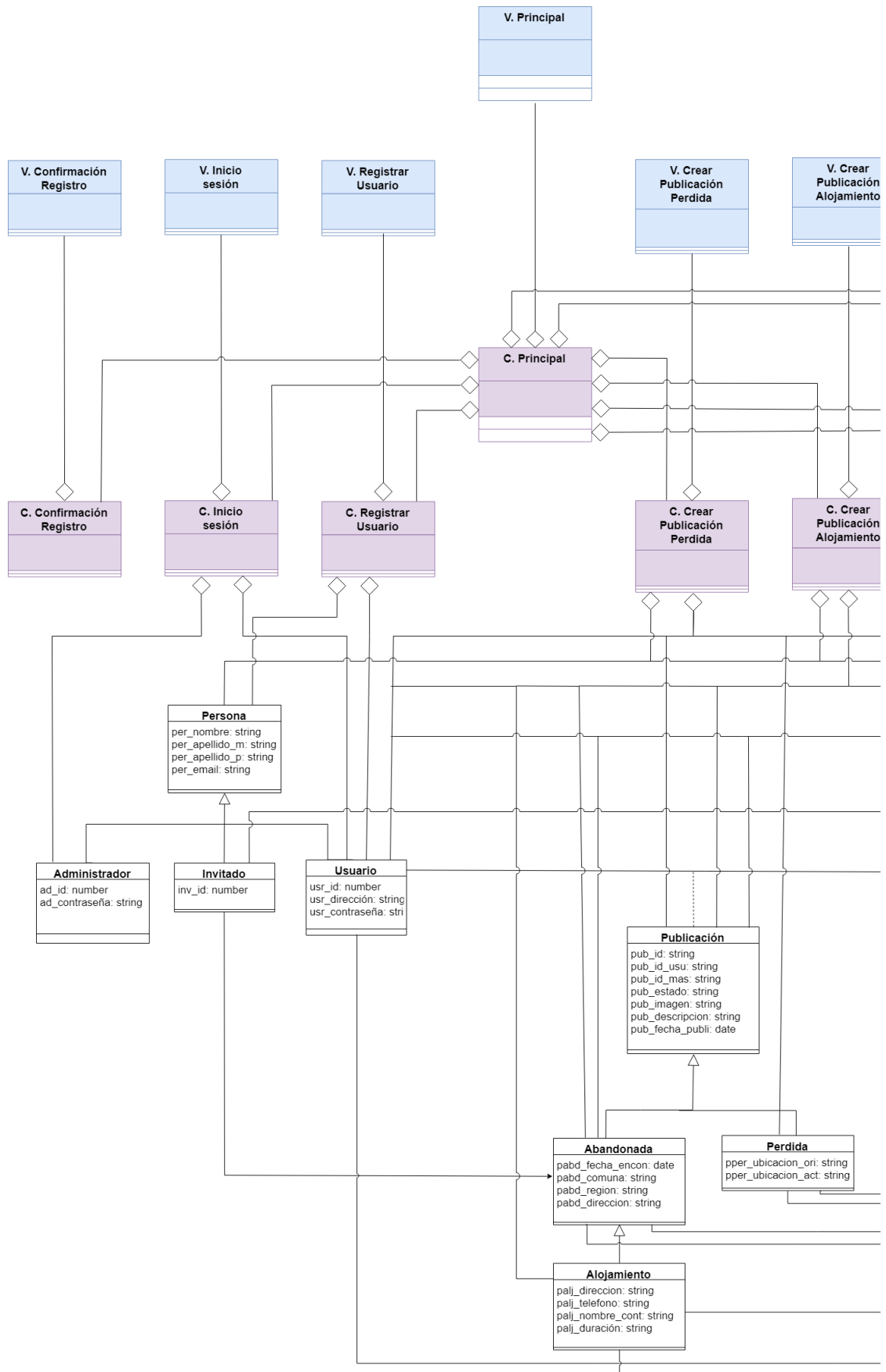


Figura 16: Modelo Vista Controlador *MVC*, parte izquierda, del sistema *Buscatdog*. Parte 1

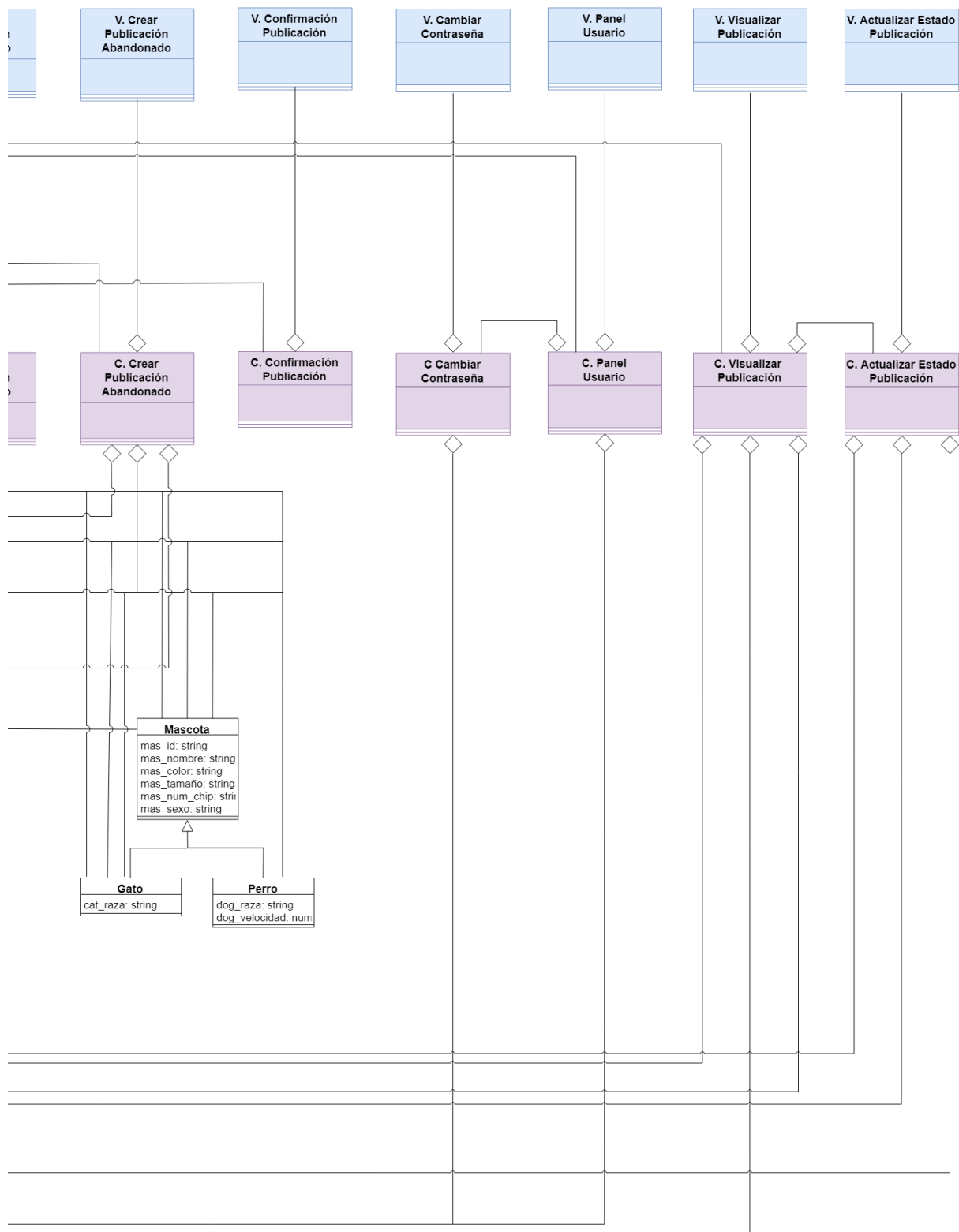


Figura 17: Modelo Vista Controlador *MVC*, parte derecha, del sistema *Buscatdog*. Parte 2

4.2.2. Diseño del comportamiento.

En esta sección, se presentan las trazas de eventos de las operaciones del sistema *Buscatdog*. Dichas operaciones fueron previamente definidas durante la etapa de Análisis Orientado a Objetos.

A continuación, las figuras 18, 19 y 20 presentan las trazas de los eventos "Crear publicación de mascota perdida", "Crear publicación de mascota abandonada" y "Crear publicación de mascota abandonada con alojamiento".

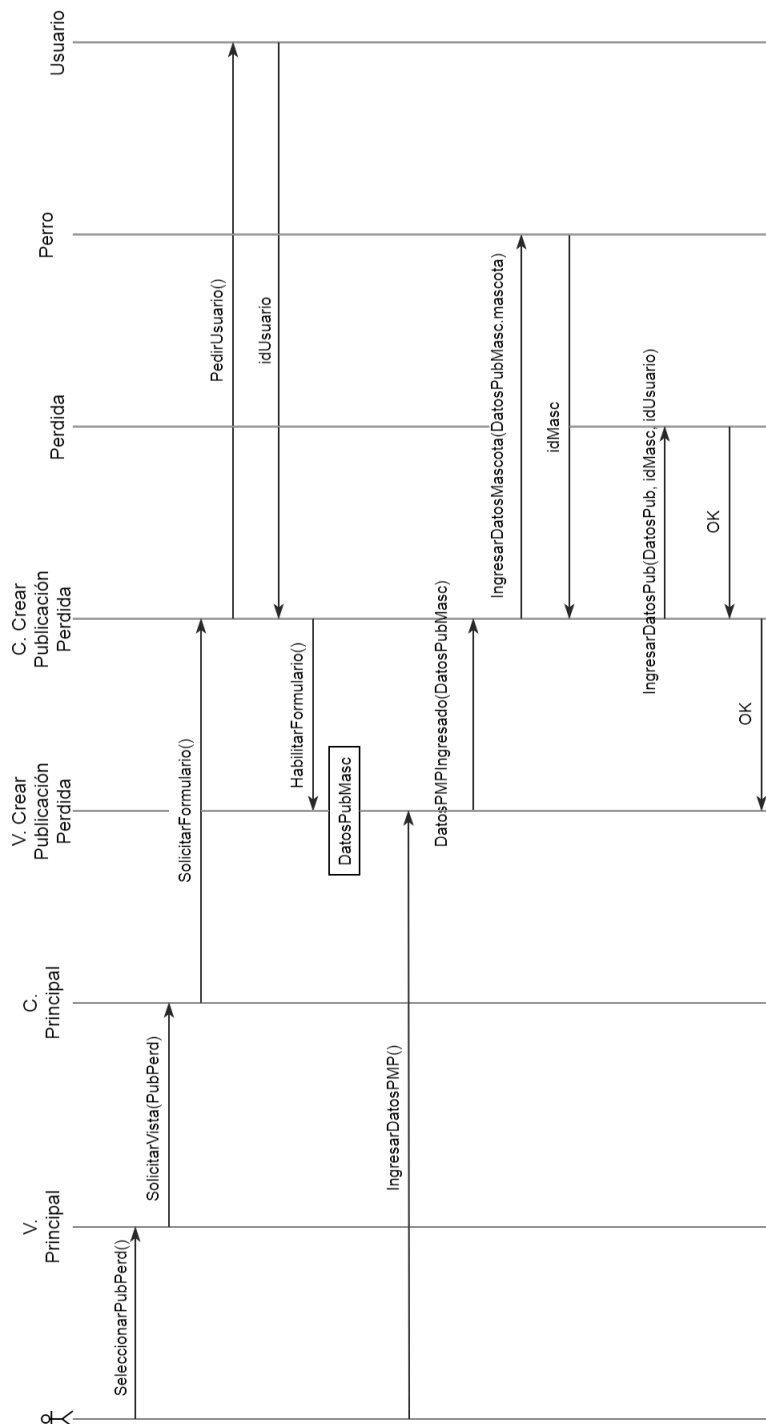


Figura 18: Traza Crear publicación de mascota perdida.

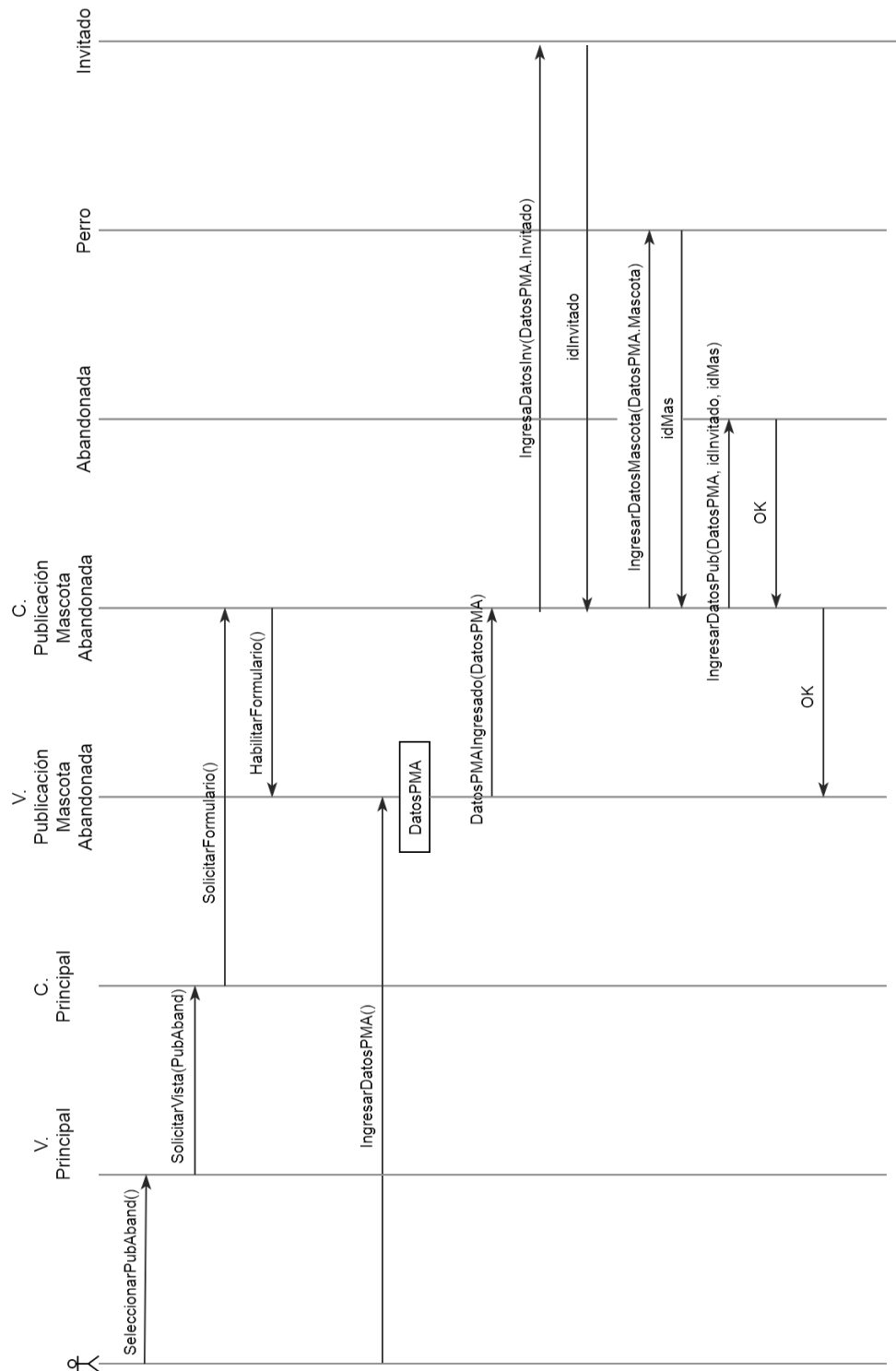


Figura 19: Traza Crear publicación de mascota abandonada.

A continuación, la figura 21 muestra la traza "creación de publicación de mascota abandonada con alojamiento". Esta se crea a partir de la traza en la figura 4, cuando el usuario ingresa los datos de la publicación de mascota abandonada, es decir, asumimos que el controlador ya ha habilitado el formulario y el usuario ya ha ingresado los datos de la publicación.

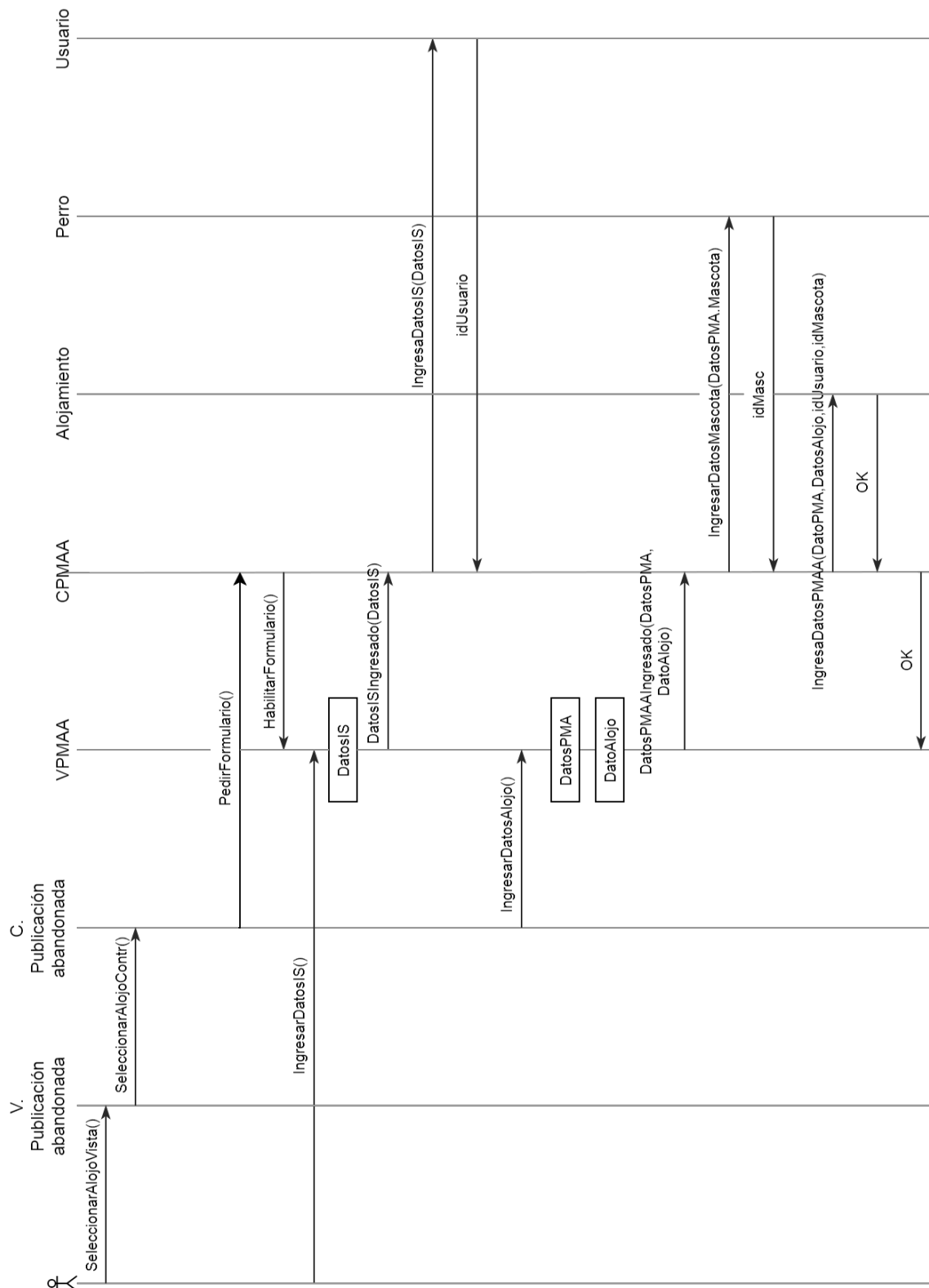


Figura 20: Traza Crear publicación de mascota abandonada con alojamiento.

A continuación, las figura 21, 22 y 23 muestran las trazas correspondiente a "Registrar Usuario", "Cambio de Contraseña" e "Inicio Sesión" respectivamente.

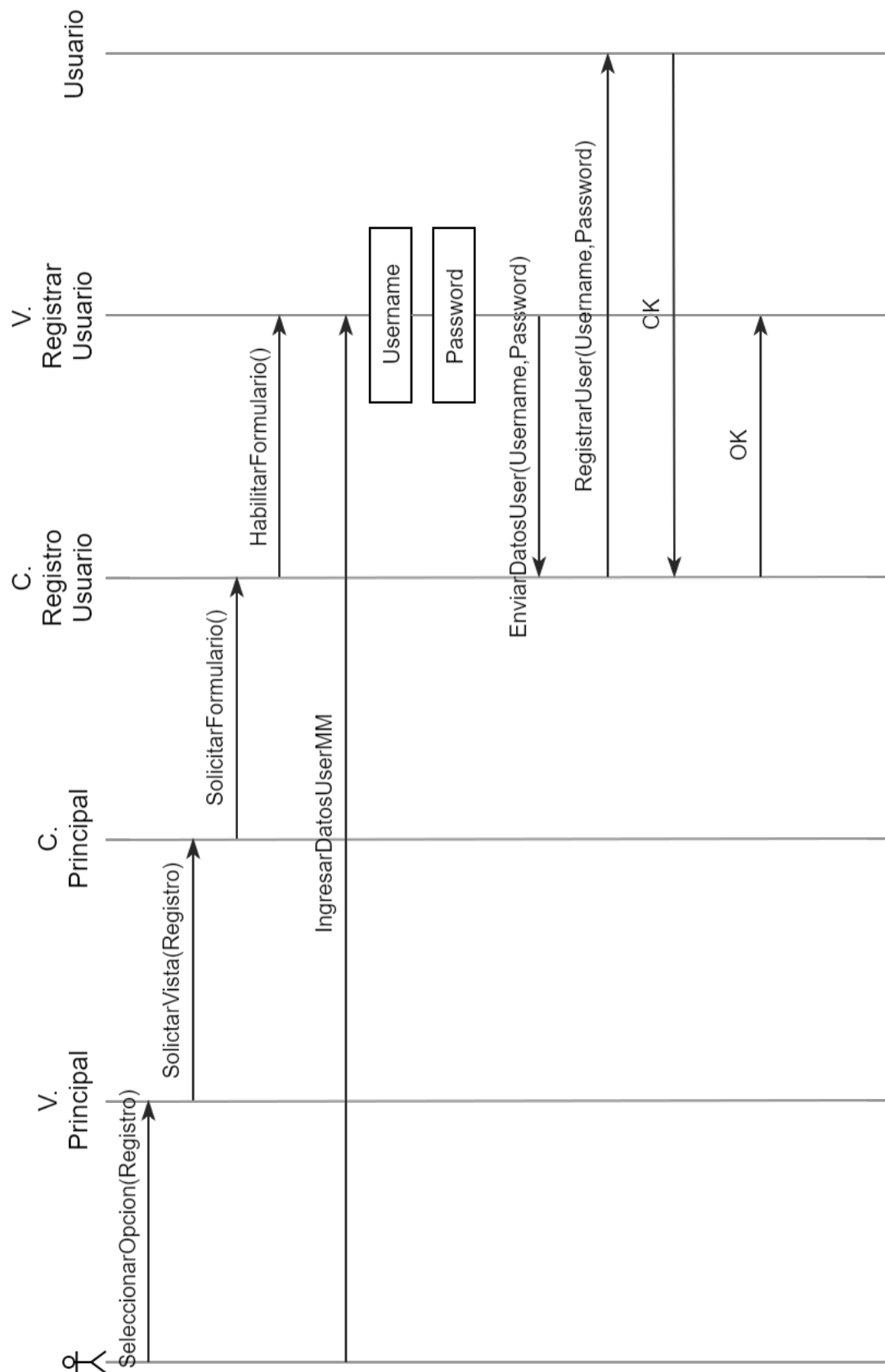


Figura 21: Traza Registrar Usuario.

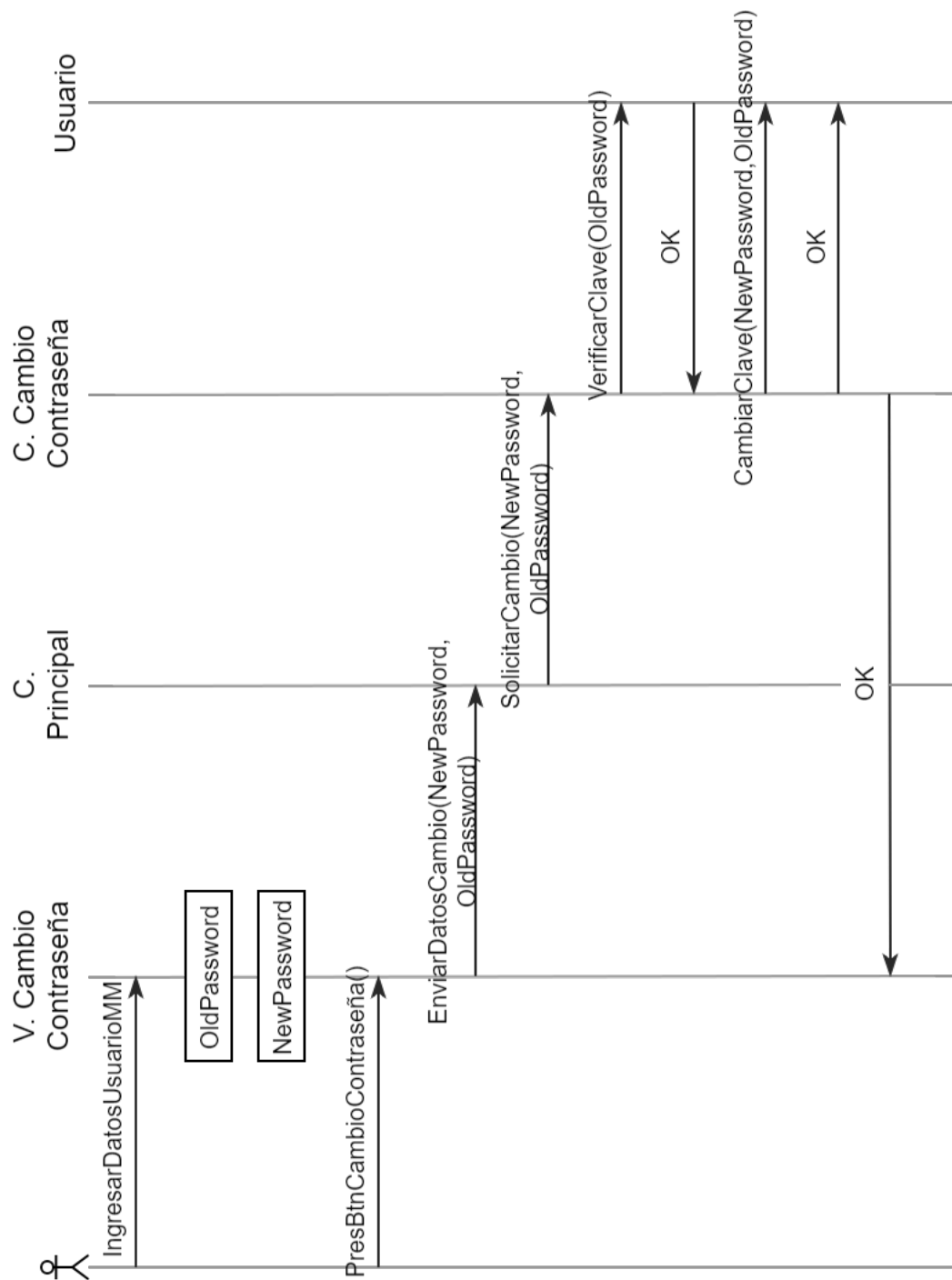


Figura 22: Trazo Cambio Contraseña.

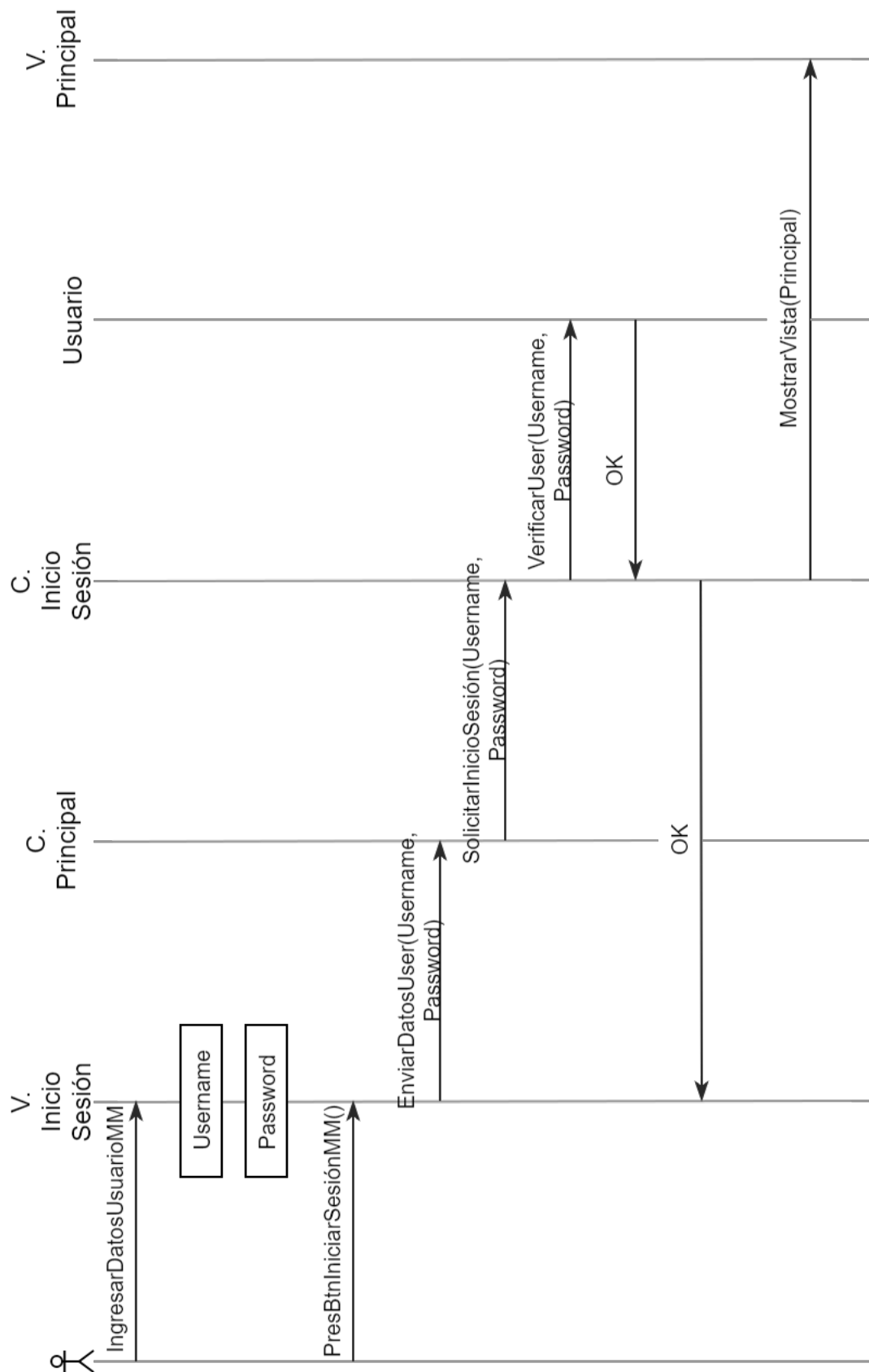


Figura 23: Traza Inicio Sesión.

4.3. Conclusiones.

Como se pudo apreciar, en este informe se presentaron dos artefactos correspondientes al Diseño Orientado a Objetos con sus definiciones correspondientes.

Estos artefactos son el Modelo de Objetos del Diseño del sistema completo, ya que es necesario dar un contexto a la solución general del problema, agregando las vistas y los controladores que definirán cómo se interactuará con el sistema de parte del actor y como se comportará el sistema. Luego se crea una traza de eventos, la cual se genera para describir puntualmente aquellas responsabilidades pertinentes al sistema, estas responsabilidades se traducen como métodos, los cuales se agregan al Modelo de Objetos del Diseño en las capas Vista, Controlador o Modelo según corresponda.

Al finalizar la etapa, se concluye como equipo que esta permite comprender mejor el dominio del problema al especificar las responsabilidades del sistema a través de los artefactos anteriormente mencionados. Estos sirven como base de cara a la siguiente etapa, Programación Orientada a Objetos.

Capítulo 5.

5. PROGRAMACIÓN ORIENTADA A OBJETOS OMT++.

5.1. Introducción.

En este informe se presenta la Programación Orientada a Objetos del sistema *Buscatdog*. Se transforma lo desarrollado en la fase anterior, Diseño Orientado a Objetos, tomando el modelo de objetos del diseño para implementarlo en un lenguaje de programación. Para esto se separan las clases que pertenecen tanto al modelo, como a la vista y al controlador que se explican a continuación.

5.2. Clases Modelo.

A continuación se presenta la especificación de las clases que forman la capa modelo del diseño de objetos.

Tabla 36: Clase Modelo - Usuario.

Clase Modelo - Usuario		
Atributos	usr_id	Identificador del usuario
	usr_dirección	Dirección
	usr_contraseña	Contraseña
Métodos	PedirUsuario()	Obtiene los valores de un usuario
	IngresarDatos()	Ingresa datos para inicio de sesión
	RegistrarUser()	Registra usuario
	VerificarClave()	Comprueba validez de la contraseña
	CambiarClave()	Cambia la contraseña por otra nueva
	VerificarUser()	Comprueba validez de los datos de inicio de sesión

Tabla 37: Clase Modelo - Invitado.

Clase Modelo - Invitado		
Atributos	inv_id	Identificador de invitado
Métodos	IngresarDatosInv()	Registra invitado

Tabla 38: Clase Modelo - Mascota.

Clase Modelo - Mascota		
Atributos	mas_id	Identificador de mascota
	mas_nombre	Nombre
	mas_color	Color pelaje
	mas_tamaño	Tamaño
	mas_num_chip	Número chip
	mas_sexo	Sexo
	mas_caracter	Carácter
Métodos	IngresarDatosMascota()	Registra mascota

Tabla 39: Clase Modelo - Perdida

Clase Modelo - Perdida		
Atributos	pper_ubicacion_ori	Ubicación donde la mascota fue extraviada
	pper_ubicacion_act	Estimación de la ubicación actual de la mascota
	usr_id	Identificador de usuario
	mas_id	Identificador de mascota
Métodos	IngresarDatosPub()	Registra una publicación de mascota perdida

Tabla 40: Clase Modelo - Abandonada

Clase Modelo - Abandonada		
Atributos	pabd_fecha_encon	Fecha en que la mascota fue encontrada
	pabd_comuna	Comuna donde la mascota fue encontrada
	pabd_region	Región donde la mascota fue encontrada
	pabd_direccion	Dirección en donde la mascota fue encontrada
	usr_id	Usuario que crea publicación
	mas_id	Mascota asociada a la publicación
Métodos	IngresarDatosPub()	Registra publicación de mascota abandonada

Tabla 41: Clase Modelo - Alojamiento

Clase Modelo - Alojamiento		
Atributos	palj_direccion	Dirección del alojo
	palj_telefono	Teléfono de contacto
	palj_nombre_cont	Nombre del contacto
	palj_duracion	Duración del alojo
	usr_id	Usuario que crea publicación
	mas_id	Mascota asociada a la publicación
Métodos	IngresarDatosPMAA()	Registra publicación de mascota abandonada con alojamiento

5.3. Clases Vista.

A continuación se presenta la especificación de las clases que forman la capa vista del diseño de objetos.

Tabla 42: Clase Vista - Principal

Clase Vista Principal		
Atributos	TPublicacion	Tipo de publicación a mostrar
	Opcion	Tipo de operación a realizar
Métodos	SeleccionarPubPerd()	Selecciona crear publicación mascota perdida
	SeleccionarOpcion()	Selecciona opción de vista
	MostrarVista()	Muestra vista seleccionada
	SeleccionarPubAband()	Selecciona crear publicación mascota abandonada

Tabla 43: Clase Vista - Crear Publicación Perdida

Clase Vista Crear Publicación Perdida		
Atributos		
Métodos	IngresarDatosPMP()	Ingresa datos de publicación para una mascota perdida
	HabilitarFormulario()	Permite visualizar formato de formulario para ingresar datos

Tabla 44: Clase Vista - Crear Publicación Abandonada

Clase Vista Crear Publicación Abandonada		
Atributos		
Métodos	IngresarDatosPMa()	Ingresa datos de publicación para una mascota abandonada
	SeleccionarAlojoVista()	Despliega formulario para publicación con alojo
	HabilitarFormulario()	Permite visualizar formato de formulario para ingresar datos

Tabla 45: Clase Vista - Crear Publicación Alojamiento

Clase Vista Crear Publicación Alojamiento		
Atributos		
Métodos	IngresarDatosAlojo()	Ingresa datos de alojo
	IngresarDatosIS()	Ingresa datos de inicio de sesión
	HabilitarFormulario()	Permite visualizar formato de formulario para ingresar datos

Tabla 46: Clase Vista - Registrar Usuario

Clase Vista Registrar Usuario		
Atributos		
Métodos	IngresarDatosUserMM()	Ingresa datos de usuario
	HabilitarFormulario()	Permite visualizar formato de formulario para ingresar datos

Tabla 47: Clase Vista - Cambio Contraseña

Clase Vista Cambio Contraseña		
Atributos		
Métodos	IngresarDatosUsuarioMM()	Ingresa datos de usuario
	PressBtnCambioContraseña()	Acepta cambiar contraseña

Tabla 48: Clase Vista - Inicio Sesión

Clase Vista Inicio Sesión		
Atributos		
Métodos	IngresarDatosUsuarioMM()	Ingresa datos de usuario
	PressBtnIniciarSesiónMM()	Acepta inicio de sesión

5.4. Clases Controlador.

A continuación se presenta la especificación de las clases que forman la capa controlador del diseño de objetos.

Tabla 49: Clase Controlador - Principal

Clase Controlador Principal		
Atributos	TPublicacion	Tipo de publicación de vista seleccionada
	NewPassword	Nueva contraseña
	OldPassword	Contraseña antigua
	Username	Nombre de usuario
	Password	Contraseña
Métodos	SolicitarVista()	Solicita vista al controlador pertinente
	EnviarDatosUser()	Recibe datos de usuario
	EnviarDatosCambio()	Recibe datos de cambio de contraseña

Tabla 50: Clase Controlador - Crear Publicación Perdida

Clase Controlador Crear Publicación Perdida		
Atributos	DatosPubMasc	Datos publicación ingresada
Métodos	SolicitarFormulario()	Solicita formulario
	MetadataIngresada()	Recibe datos de de publicación

Tabla 51: Clase Controlador - Crear Publicación Abandonada

Clase Controlador Crear Publicación Abandonada		
Atributos	DatosPMA	Datos publicación ingresada
Métodos	SolicitarFormulario()	Solicita formulario
	DatosPMAIngresado()	Recibe datos de la publicación
	SeleccionarAlojoContr()	Prepara recepción de datos de alojamiento

Tabla 52: Clase Controlador - Crear Publicación Alojamiento

Clase Controlador Crear Publicación Alojamiento		
Atributos	DatosIS	Datos de inicio de sesión ingresados
	DatosPMA	Datos publicación ingresada
	DatosAlojo	Datos de alojamiento ingresado
Métodos	PedirFormulario()	Solicita formulario
	DatosISIngresado()	Recibe datos de inicio de sesión
	DatosPMAIngresado()	Recibe datos de la publicación

Tabla 53: Clase Controlador - Registro Usuario

Clase Controlador Registro Usuario		
Atributos	Username	Nombre de usuario
	Password	Contraseña de usuario
Métodos	SolicitarFormulario()	Solicita formulario
	EnviarDatosUser()	Recibe datos de del usuario a registrar

Tabla 54: Clase Controlador - Cambio Contraseña

Clase Controlador Cambio Contraseña		
Atributos	NewPassword	Contraseña nueva
	OldPassword	Contraseña antigua
Métodos	SolicitarCambio()	Solicita cambio de contraseña con datos ingresados

Tabla 55: Clase Controlador - Inicio Sesión

Clase Controlador Inicio Sesión		
Atributos	Username	Nombre de usuario
	Password	Contraseña de usuario
Métodos	SolicitarInicioSesion()	Solicita inicio de sesión con datos ingresados

Capítulo 6.

6. CONCLUSIONES DEL PROYECTO.

A lo largo del proyecto se han realizado diversas actividades para que se cumpla el objetivo de implementar el sistema *Buscatdog*. Éstas consistieron en las fases de conceptualización, análisis, diseño y programación, orientados a objetos. En estas etapas del desarrollo se utilizó el lenguaje técnico obtenido de OMT++, en donde herramientas tales como los casos de uso, los modelos de objetos del análisis y del diseño, el diagrama de diálogos y las trazas de eventos fueron descritas en el informe.

De esta manera se pudo contemplar que el trabajo de equipo consiste en un punto fundamental dentro de todo el desarrollo de software, el cual permite construir estructuras de mayor complejidad teniendo límites de recursos y de tiempo de proporciones realistas. Las habilidades de cada integrante como individuo varían y conforman la posibilidad de repartir labores e idear soluciones que en cualquier trabajo individual podrían no ser alcanzables.

Al comenzar desde la fase de conceptualización, el equipo logró una clara visualización de cómo sería el crecimiento del sistema, pero en etapas posteriores se generaron ligeras confusiones y diferencias de ideas sobre los conceptos y técnicas que se deben utilizar para conseguir cumplir con los objetivos anteriormente descritos.

Se considera que los elementos suministrados por el curso de Ingeniería de Software II permiten guiar estructuradamente actividades que de otra manera serían caóticas, y confiere valor a aquellos descubrimientos que la humanidad ha desarrollado a lo largo de la historia.

En conclusión, tanto las herramientas OMT++ utilizadas como las experiencias que se obtuvieron en el trabajo en equipo en la totalidad de la implementación de sistema *Buscatdog* suman competencias profesionales muy valiosas. Las complicaciones encontradas fueron atravesadas y en los casos necesarios, se reasignaron tareas, horas de trabajo y responsabilidades para cumplir con las entregas (cuyo detalle se puede ver en *apéndice: sección E*, lo que resultó en una alta cohesión de los integrantes, logrando completar la implementación del software.

7. REFERENCIAS BIBLIOGRÁFICAS.

- ⁽¹⁾ AdimarkGfK - Estudios públicos (2014) (<http://www.adimark.cl/es/estudios/index.asp?id=264>)
- ⁽²⁾ Soto Parraguez, Alejandra Pamela; Análisis de un problema público no abordado, El caso de los perros vagabundos y callejeros en Chile (2013) (<http://repositorio.uchile.cl/handle/2250/113119>)
- ⁽³⁾ WOF - Aplicaciones en Google Play (2018)
(<https://play.google.com/store/apps/details?id=com.ibex.wof>)
- ⁽⁴⁾ BuscoMascota.cl - Mascota Desaparecida - Encontrada (2018)
(<https://buscomascota.cl/mascotas-desaparecidas>).
- ⁽⁵⁾ Google Maps Platform - Google Developers (2019)
(<https://developers.google.com/maps/documentation/?hl=es>).
- ⁽⁶⁾ The GNU General Public License v3.0 - GNU Project - Free Software Foundation (2007)
(<https://www.gnu.org/licenses/gpl.html>)

8. APÉNDICE A. MANUAL DE USUARIO

En esta sección se muestra el manual de usuario, que corresponde a una guía funcional que pueden utilizar los usuarios para encontrar, conocer y aplicar las principales funcionalidades del software.

8.1. Crear publicación mascotas perdidas

A continuación, las figuras 25 y 26 muestran el flujo respectivo para la creación de una publicación de mascota perdida.

CREAR PUBLICACIÓN

Crear publicación de mascota perdida o abandonada.

Paso 1: Seleccionar con el mouse el botón "Publicar mascota perdida".

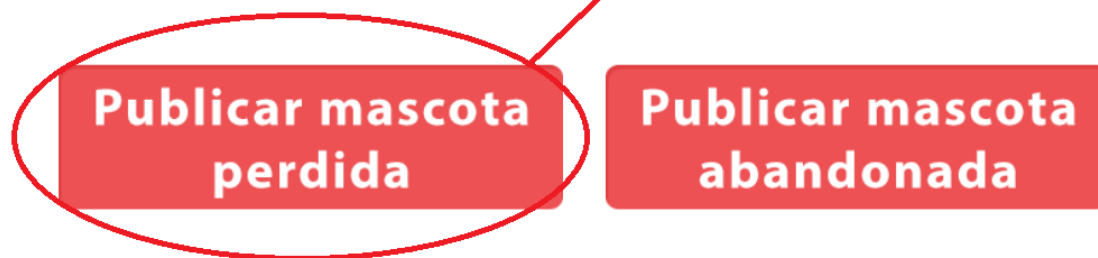


Figura 24: Crear publicación mascotas perdidas. Paso 1.

En la figura 24 se aprecian las opciones disponibles al ingresar a la plataforma. Para crear una publicación de mascota perdida, se debe seleccionar el botón **Publicar mascota perdida**, tal como lo muestra el paso 1.

CREAR PUBLICACIÓN DE MASCOTA PERDIDA

Los siguientes campos corresponden a información de tu mascota:

Nombre
Ej.: Copito

Especie Raza
Ej.: Labrador

Tamaño Color
Ej.: Negro

Sexo Número de Chip
Ej.: 123456789123456

Dirección de pérdida
Ej.: Ecuador 8981, Estacion Central, Santiago

Descripción adicional

Subir fotografía
 No file chosen

Paso 2: Rellenar el formulario con los datos de la mascota perdida, incluyendo una fotografía de referencia.

Figura 25: Crear publicación mascotas perdidas. Paso 2.

El paso 2 se ilustra en la figura 25, que contiene el formulario en donde se deben ingresar los datos de la publicación. La publicación será creada una vez se seleccione el botón **Publicar**.

8.2. Crear publicación mascotas abandonadas

A continuación, las figuras 26 y 27 muestran como crear una publicación de mascota abandonada.



CREAR PUBLICACIÓN

Crear publicación de mascota perdida o abandonada.

Paso 1: Seleccionar con el mouse el botón "Publicar mascota abandonada."



Figura 26: Crear publicación mascotas abandonadas. Paso 1.

CREAR PUBLICACIÓN DE MASCOTA ABANDONADA

Los siguientes campos corresponden a información de la mascota vista:

Especie

Raza

Ej.: Labrador

Tamaño

Color

Ej.: Negro

Sexo

Dirección donde se vió a la mascota

Ej.: Ecuador 8981, Estacion Central, Santiago

Fecha en que se vió a la mascota

AAAA-MM-DD

Descripción adicional

Subir fotografía

No file chosen

Paso 2: Rellenar formulario con los datos de la mascota abandonada, incluyendo una foto de referencia

Figura 27: Crear publicación mascotas abandonadas. Paso 2.

Siguiendo los pasos 1 y 2 de la creación de publicaciones de mascotas abandonadas se tendrá la publicación creada.

8.3. Registrar usuario

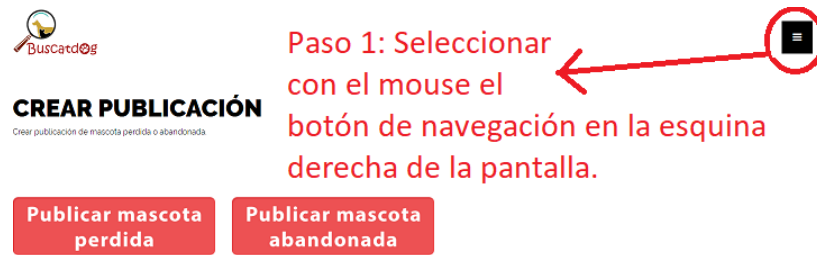


Figura 28: Registrar usuario. Paso 1.

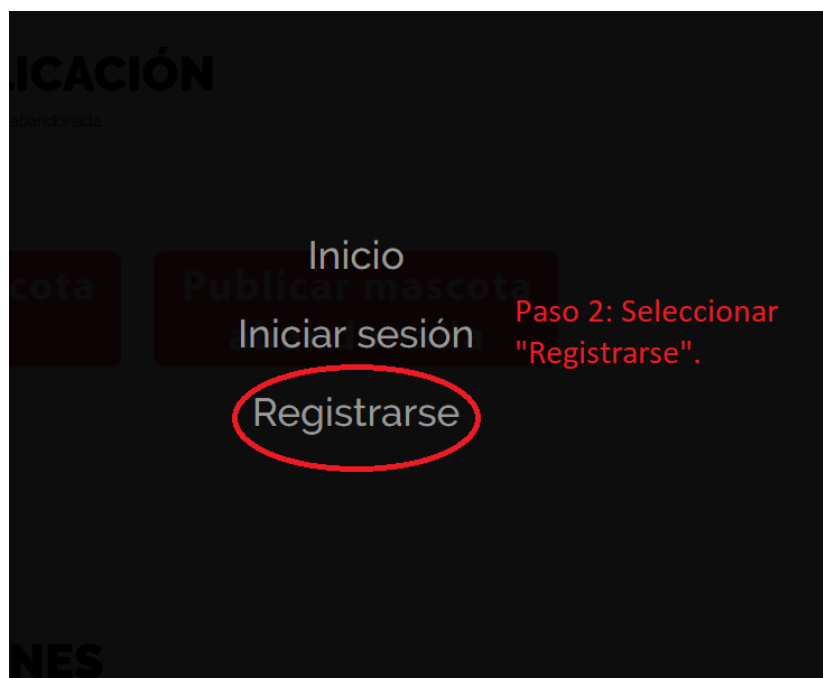


Figura 29: Registrar usuario. Paso 2.

REGISTRAR USUARIO

Completa los siguientes campos con tu información personal:

Nombre

Ej.: Lucas

Apellido paterno

Ej.: Saavedra

Apellido materno

Ej.: Morales

Correo electrónico

Ej.: usuario@gmail.com

Contraseña

Dirección

Ej.: Uvas del viento 1313, Ñuñoa, Santiago

Registrar

Paso 3: Rellenar el
formulario con los datos
del usuario a registrar.

Figura 30: Registrar usuario. Paso 3.

Siguiendo todos los pasos del registro de usuario se tendrá un usuario nuevo creado.

8.4. Alojar mascota abandonada



CREAR PUBLICACIÓN

Crear publicación de mascota perdida o abandonada.

Publicar mascota
perdida

Publicar mascota
abandonada

Paso 1: Seleccionar con el
mouse el botón "Publicar
mascota abandonada.

Figura 31: Alojar mascota abandonada. Paso 1.



Paso 2: Seleccionar
botón "Incluir
alojamiento".



Figura 32: Alojar mascota abandonada. Paso 2.

ABANDONADA CON ALOJAMIENTO

Los siguientes campos corresponden a información de la mascota vista:

Especie | Gato
Sexo | Hembra
Tamaño | Chico
Color

Ej.: Negro

Raza

Ej.: Labrador

Dirección donde se encontró a la mascota

Ej.: Ecuador 8981, Estacion Central, Santiago

Fecha en que se rescató a la mascota

AAAA-MM-DD

Descripción adicional

Subir fotografía

Choose File No file chosen

Los siguientes campos corresponden a información del alojamiento:

Dirección de alojamiento

Ej.: Las peras 8981, Estacion Central, Santiago

Teléfono de contacto

Ej.: +56985632551

Nombre encargado

Ej.: Sofía Soto

Puedo dar alojamiento hasta:

AAAA-MM-DD

Publicar

Paso 3:
Rellenar
formulario
con los datos
de la mascota
abandonada y
los del
alojamiento,
incluyendo
foto de
referencia de
la mascota

Figura 33: Alojar mascota abandonada. Paso 3.

8.5. Actualizar ubicación de mascota



Figura 34: Actualizar ubicación de mascota. Paso 1.

LO VI EN:

Ubicación

Ej.: Los Vilos 551, La Granja, Región Metropolitana

Actualizar ubicación

Paso 2: Desplazarse hacia el final de la página e ingresar la nueva ubicación de la mascota. Luego presione el botón "Actualizar ubicación".

Figura 35: Actualizar ubicación de mascota. Paso 2.

8.6. Iniciar sesión

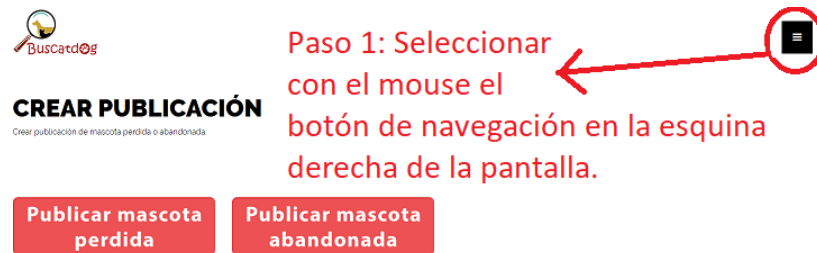


Figura 36: Iniciar sesión. Paso 1.

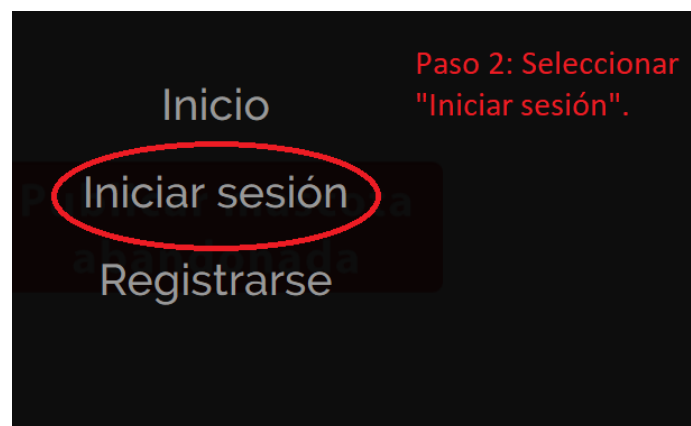


Figura 37: Iniciar sesión. Paso 2.

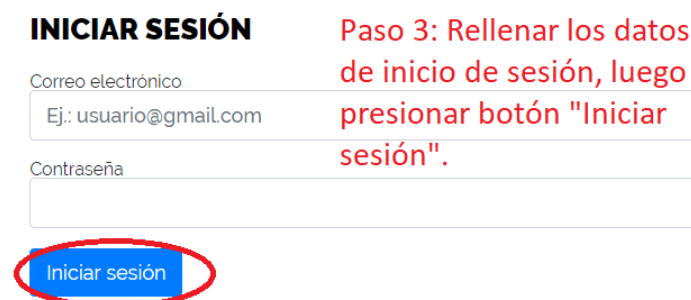


Figura 38: Iniciar sesión. Paso 3.

8.7. Listar publicaciones

Con la sesión de un usuario iniciada, seguir los pasos mostrados a continuación.

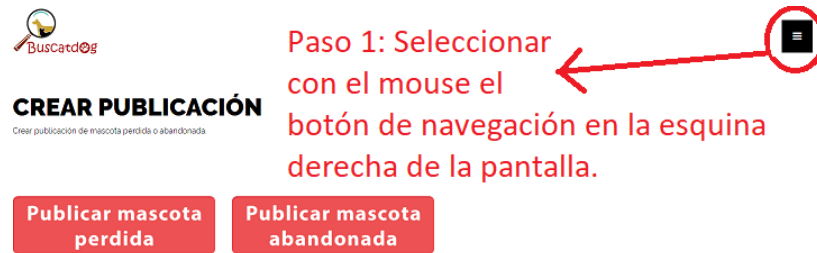


Figura 39: Listar publicaciones. Paso 1.



Figura 40: Listar publicaciones. Paso 2.

HOLA JUAN

ACTUALIZAR ESTADO PUBLICACIÓN

Mascota	Publicación	Estado	Acción
Canela	04/08/2018	Perdido	Lo encuentre
Moka	03/02/2018	Perdido	Lo encuentre

Paso 3: En el rectángulo se puede apreciar la lista de publicaciones creadas por el usuario con la sesión iniciada.

Figura 41: Listar publicaciones. Paso 3.

8.8. Actualizar estado de mascota perdida

Mientras el usuario se encuentre en el panel, visualizando sus publicaciones listadas, puede realizar la actualización del estado de sus mascotas perdidas.

HOLA JUAN

ACTUALIZAR ESTADO PUBLICACIÓN

Mascota	Publicación	Estado	Acción
Canela	04/08/2018	Perdido	Lo encuentre
Moka	03/02/2018	Perdido	Lo encuentre

Paso 1: Seleccionar botón "Lo encontré". Luego, el estado de la mascota se actualizará a "Encontrada".

Figura 42: Actualizar estado de mascota perdida. Paso 1.

9. APÉNDICE B. MANUAL DE EXPLOTACIÓN

9.1. Licencia.

El sistema Buscatdog (el "Software") se rige por la licencia GNU General Public License v3.0⁽⁶⁾, la cual se puede encontrar con el nombre LICENSE dentro del código del Software.

9.2. Instalación.

A continuación, se presentan las tecnologías necesarias para la instalación y correcta ejecución del Software.

9.2.1. PostgreSQL 11.1

El primer paso es el levantamiento de la base de datos con el programa PostgreSQL 11.1, el que puede ser descargado del sitio oficial <https://www.postgresql.org>. Para su instalación, debe ser realizada de la siguiente forma:

Una vez descargado el archivo de instalación lo ejecutamos y si tenemos activado el control de cuentas de usuario nos mostrará una advertencia con el texto "¿Desea permitir que este programa realice cambios en el equipo?", pulsaremos "Sí" para continuar con la instalación de PostgreSQL:

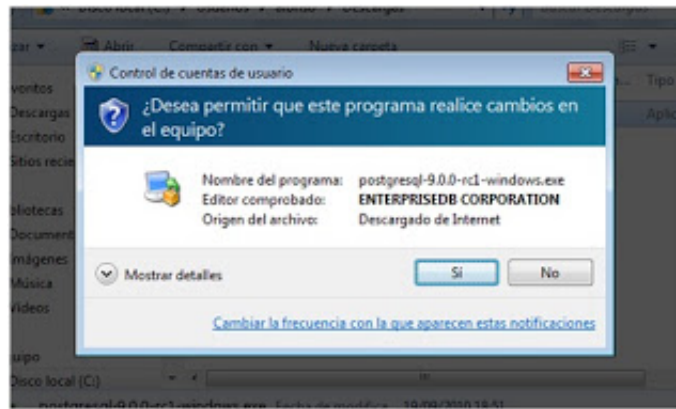


Figura 43: Paso 1 - Instalación PostgreSQL

Se iniciará el asistente para instalar PostgreSQL, pulsaremos "Siguiente":

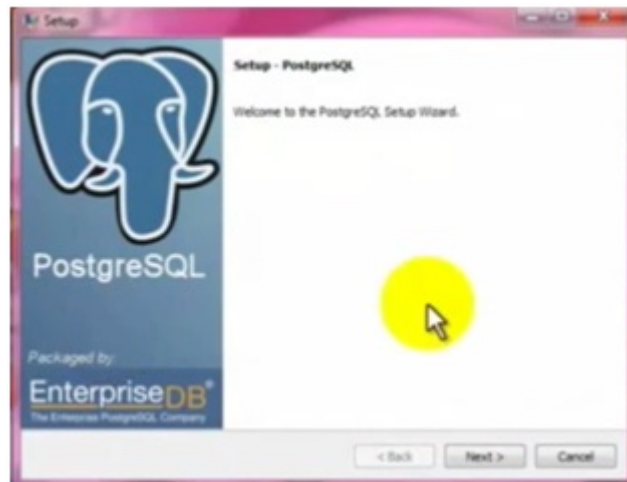


Figura 44: Paso 2 - Instalación PostgreSQL

Indicaremos la carpeta de instalación de PostgreSQL, donde se guardarán los ejecutables, librerías y ficheros de configuración de PostgreSQL:



Figura 45: Paso 3 - Instalación PostgreSQL

Indicaremos también la carpeta donde se guardarán los datos por defecto de PostgreSQL:



Figura 46: Paso 4 - Instalación PostgreSQL

Introduciremos la contraseña para el súper usuario "postgres" que será con el que iniciemos sesión para administrar la base de datos, si tenemos clave de cuenta en Windows introduciremos nuestra clave.

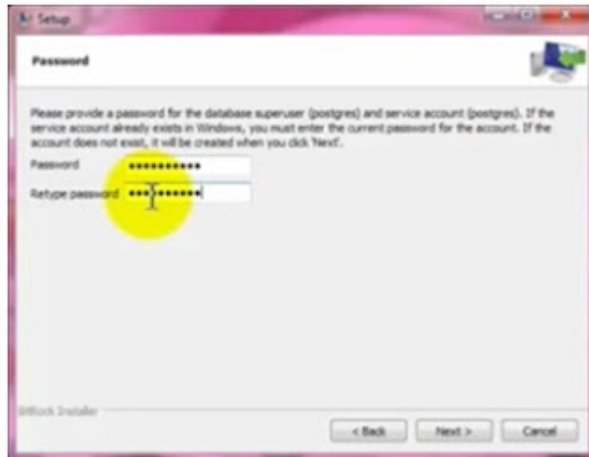


Figura 47: Paso 5 - Instalación PostgreSQL

Introduciremos el puerto de escucha para la conexión con el servidor PostgreSQL, por defecto el 5432:

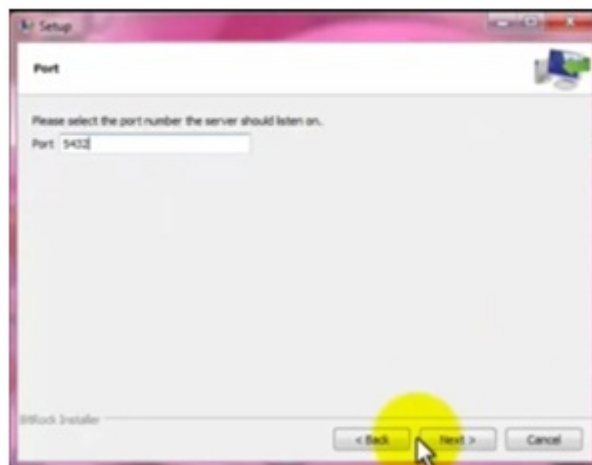


Figura 48: Paso 6 - Instalación PostgreSQL

Seleccionaremos la configuración regional:

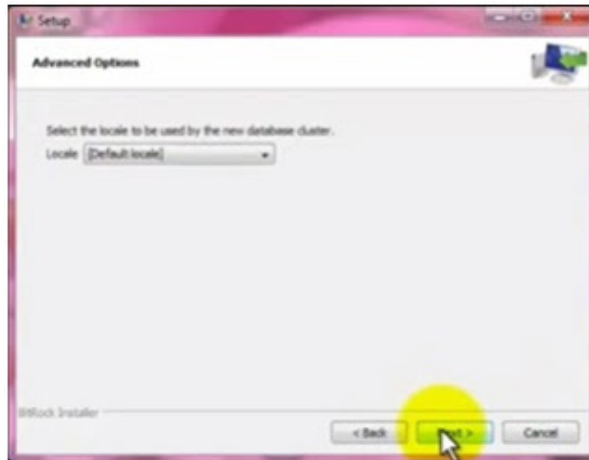


Figura 49: Paso 7 - Instalación PostgreSQL

Pulsaremos "Siguiente" para iniciar la instalación definitiva.



Figura 50: Paso 8 - Instalación PostgreSQL

Se iniciará el asistente para instalar el motor de base de datos PostgreSQL, que creará las carpetas oportunas, copiará los ficheros necesarios y creará el servicio Windows para iniciar de forma automática el motor de base de datos:

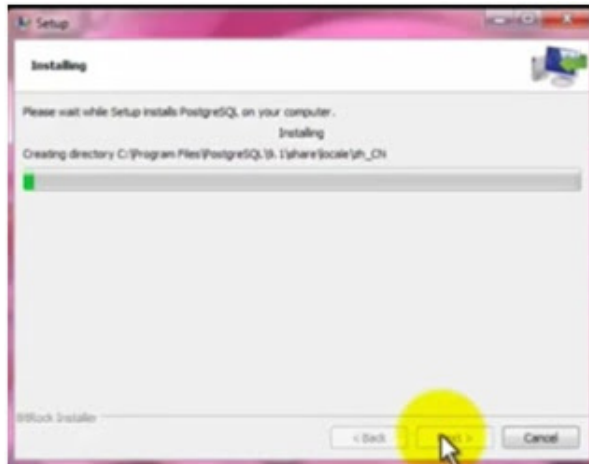


Figura 51: Paso 9 - Instalación PostgreSQL

Una vez finalizada la instalación el asistente nos dará la posibilidad de ejecutar Stack Builder, aplicación que nos permitirá instalar otros componentes y herramientas para PostgreSQL:



Figura 52: Paso 10 - Instalación PostgreSQL

Luego se procede a montar las tablas del software en la base de datos, mediante un archivo llamado *export* en el cual se encuentran todas las tablas y los datos de estas. Para crear las tablas en la base de datos, solamente se tiene que copiar el código del archivo en una plantilla SQL del frontal. Para finalizar seleccionar todo el código y compilarlo, como se muestra en la Figura 54.

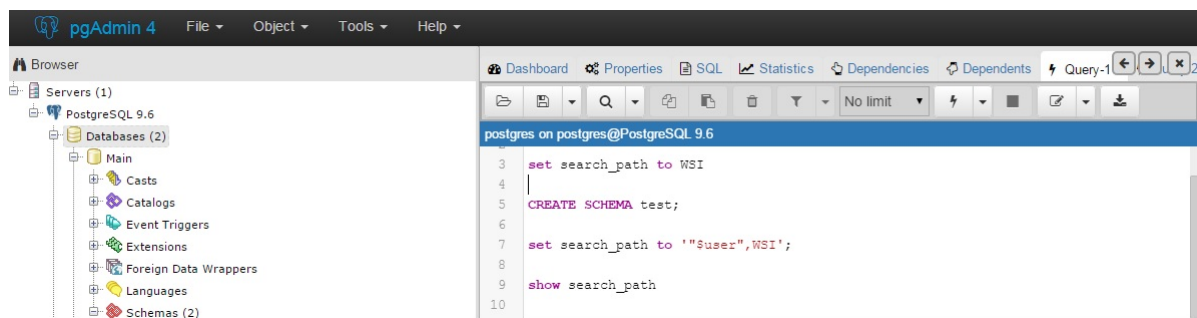


Figura 53: Creación de la Base de Datos

9.2.2. XAMPP 7.3.0

A continuación se presenta la guía de instalación de XAMPP, que permite montar el sitio web. XAMPP se puede descargar del sitio web <https://www.apachefriends.org>.

Una vez descargado el paquete, puedes ejecutar el archivo .exe haciendo doble clic en él (Se recomienda desactivar el programa antivirus hasta que todos los componentes estén instalados, ya que puede obstaculizar el proceso de instalación).

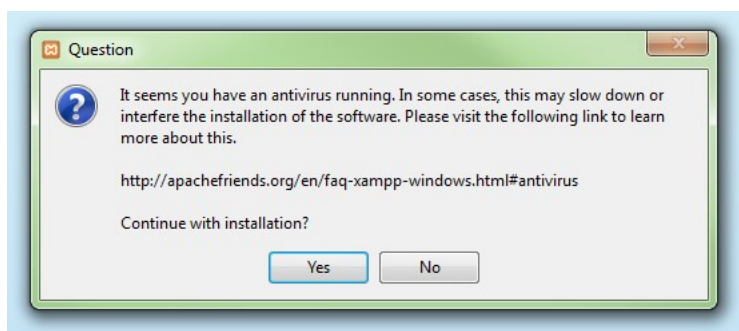


Figura 54: Paso 1 - Instalación XAMPP

También el control de cuentas de usuario (User Account Control, UAC) puede interferir en la instalación, ya que limita los derechos de escritura en la unidad de disco C:. Para saber cómo desactivar temporalmente el UAC puedes dirigirte a las páginas de soporte de Microsoft.

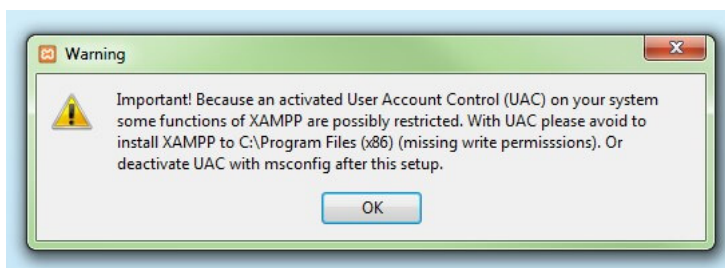


Figura 55: Paso 2 - Instalación XAMPP

Una vez superados estos pasos, aparece la pantalla de inicio del asistente para instalar XAMPP. Para ajustar las configuraciones de la instalación se hace clic en "Next".

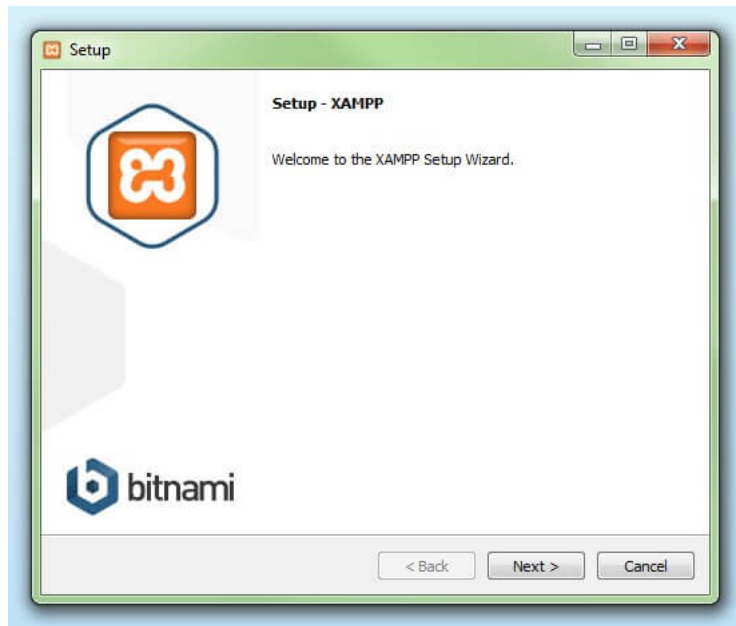


Figura 56: Paso 3 - Instalación XAMPP

En la rúbrica “Select components” se pueden excluir de la instalación componentes aislados del paquete de software de XAMPP. Se recomienda la configuración estándar para un servidor de prueba local, con la cual se instalan todos los componentes disponibles. Confirma la selección haciendo clic en “Next”.

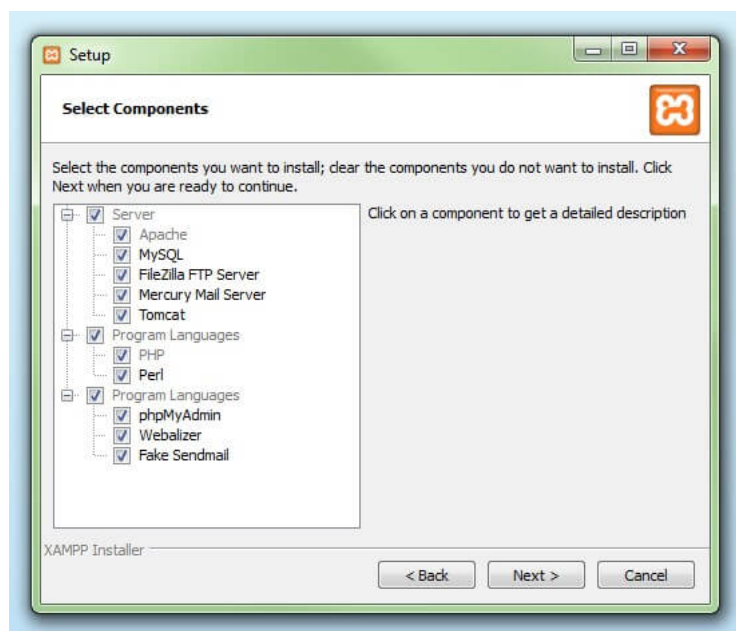


Figura 57: Paso 4 - Instalación XAMPP

En este paso se escoge el directorio donde se instalará el paquete. Si se ha escogido la configuración estándar se creará una carpeta con el nombre XAMPP en C:.

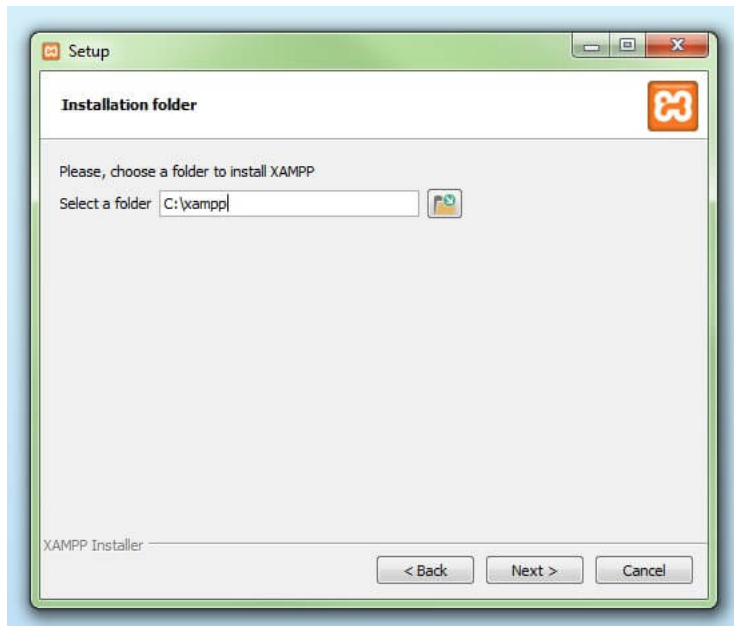


Figura 58: Paso 5 - Instalación XAMPP

El asistente extrae los componentes seleccionados y los guarda en el directorio escogido en un proceso que puede durar algunos minutos. El avance de la instalación se muestra como una barra de carga de color verde.



Figura 59: Paso 6 - Instalación XAMPP

Durante el proceso de instalación es frecuente que el asistente avise del bloqueo de Firewall. En la ventana de diálogo puedes marcar las casillas correspondientes para permitir la comunicación del servidor Apache en una red privada o en una red de trabajo. Recuerda que no se recomienda usarlo en una red pública.

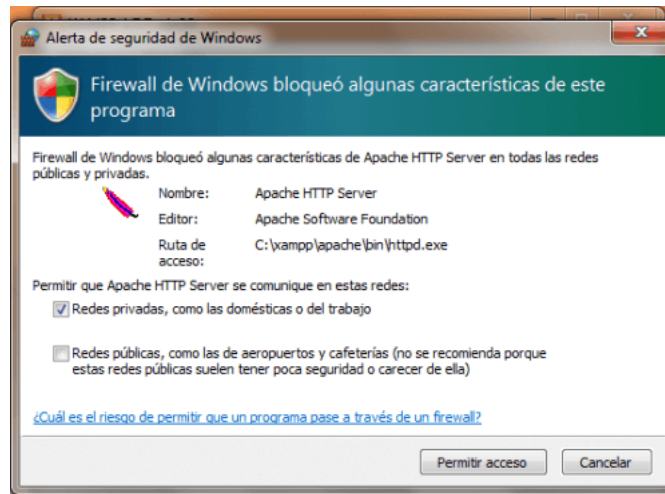


Figura 60: Paso 7 - Instalación XAMPP

Una vez extraídos e instalados todos los componentes puedes cerrar el asistente con la tecla “Finish”. Para acceder inmediatamente al panel de control solo es necesario marcar la casilla que pregunta si deseamos hacerlo.

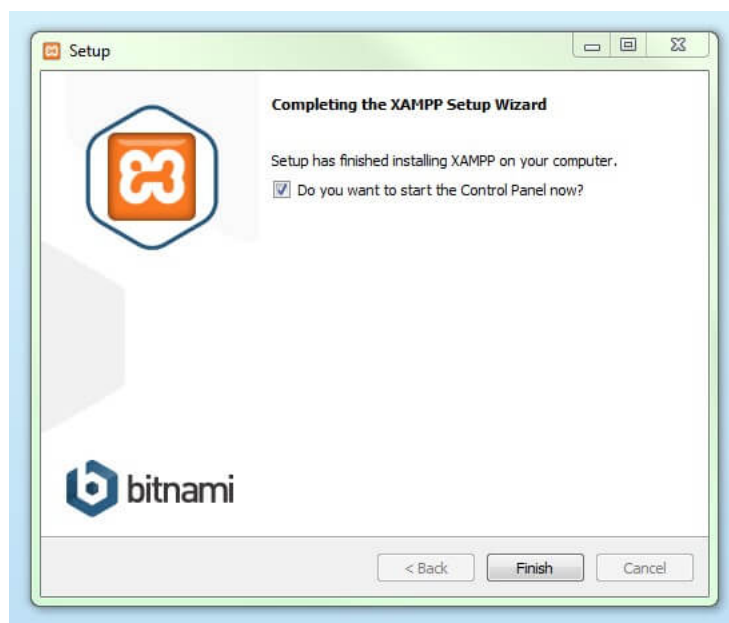


Figura 61: Paso 8 - Instalación XAMPP

En la diáfana interfaz de usuario del panel de control se protocolan todas las acciones y es posible activar o desactivar los módulos por separado con un simple clic. Además, se dispone de diversas utilidades como: Config: para configurar XAMPP así como otros componentes aislados. Netstart: muestra todos los procesos en funcionamiento en el ordenador local Shell: lanza una ventana de comandos UNIX Explorer: abre la carpeta XAMPP en el explorador de Windows Services: muestra todos los servicios en funcionamiento Help: incluye enlaces a foros de usuarios Quit: se usar para salir del panel de control

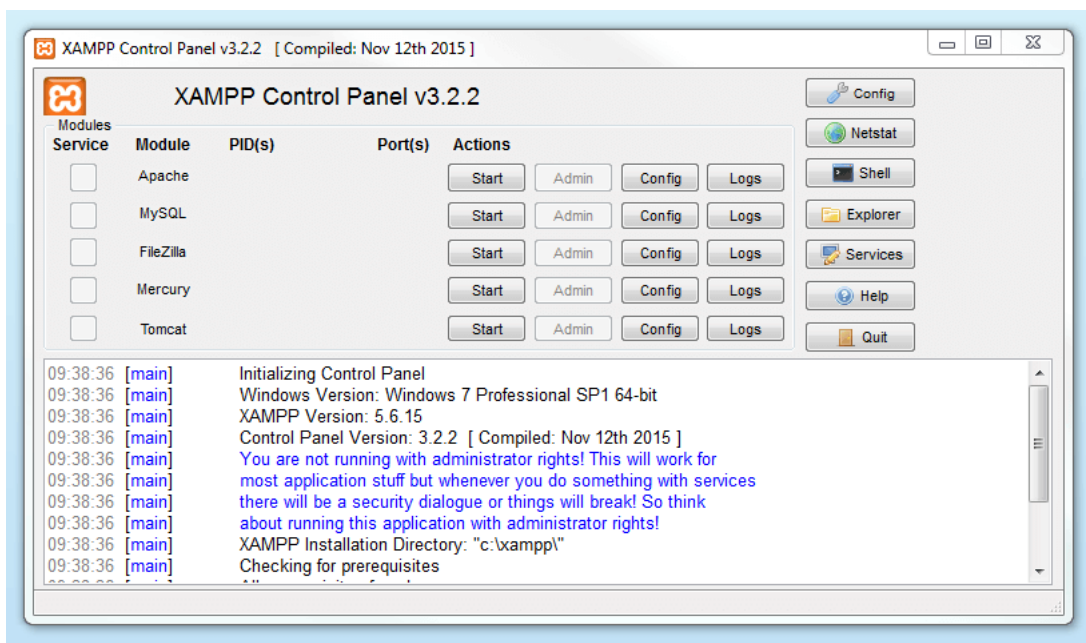


Figura 62: XAMPP - Panel de control

En la parte superior se pueden iniciar o interrumpir los módulos de XAMPP por separado mediante los comandos “Start” y “Stop” bajo “Actions”. Los módulos que se activaron aparecen marcados en verde.

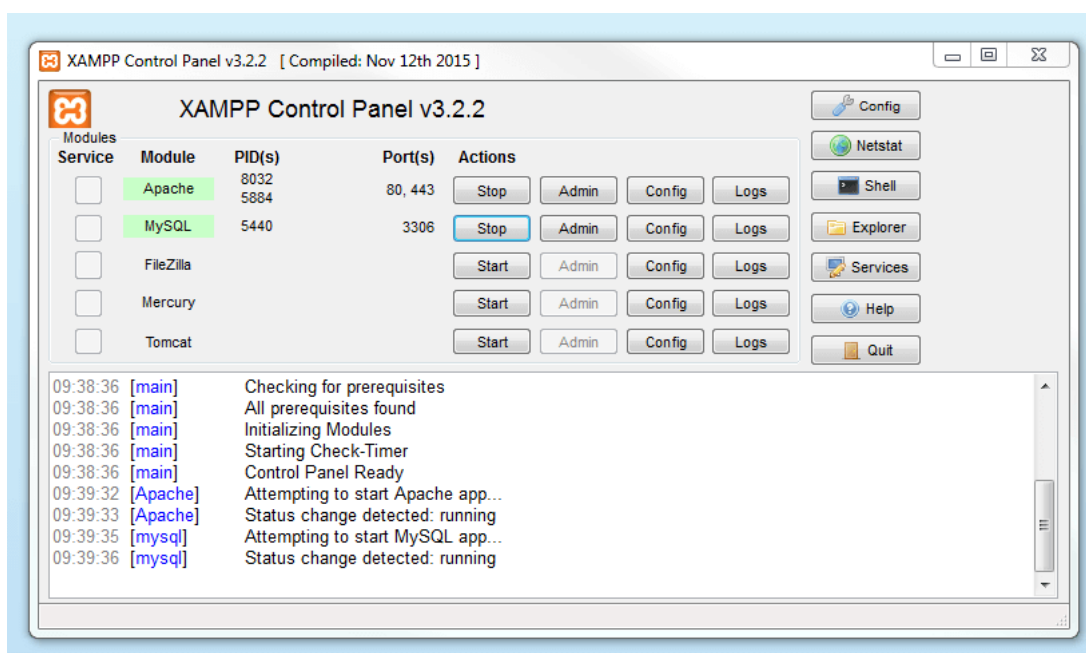


Figura 63: XAMPP - Panel de control (módulos)

Finalmente se debe copiar el directorio la aplicación web *Buscatdog* dentro del directorio raíz del sitio web de XAMPP.

10. APÉNDICE C. GRUPOS DE FACEBOOK DE BÚSQUEDA DE MASCOTAS PERDIDAS.



Figura 64: Grupos de facebook para buscar mascotas perdidas.

11. APÉNDICE D. CARTA GANTT

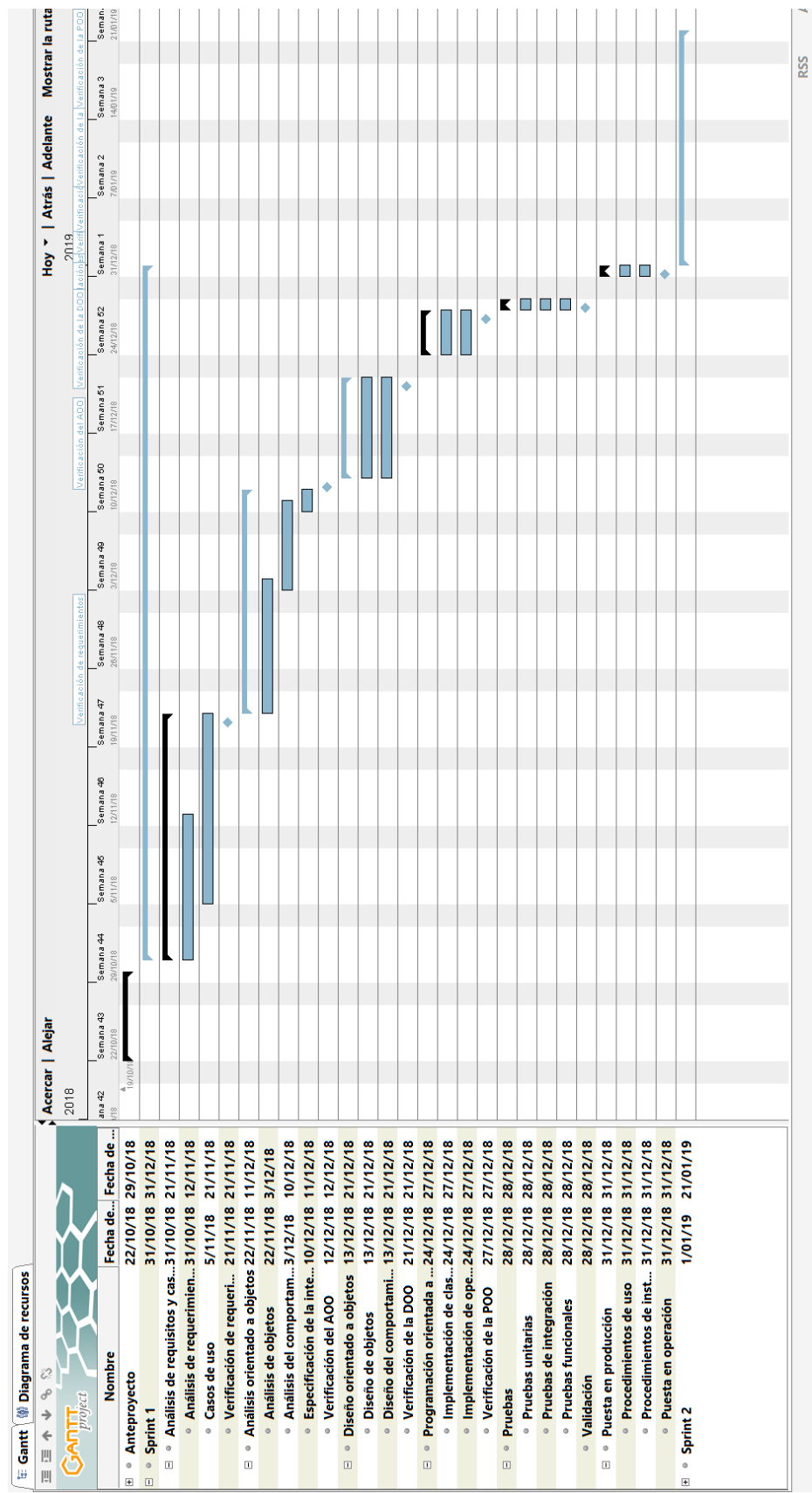


Figura 65: Carta Gantt Sistema Buscatdog: Sprint 1.

12. APÉNDICE E. DETALLE DE REUNIONES.

Este documento contabiliza las asistencias a las reuniones realizadas durante el semestre.

✓: asistido

O: inasistencia avisada

✗: inasistencia no avisada

Presentaciones de avances

	Katerine Muñoz	Sebastián Holloway	Sergio Muñoz	Pablo Maertens	Ignacio Olave	Duración en horas
01 Viernes 19.Oct.2018	✓	✓	✓	✓	✗	1,5
02 Lunes 22.Oct.2018	✓	✓	✓	O	✗	1,5
03 Jueves 25.Oct.2018	✓	✓	✓	✓	✓	1,5
04 Sábado 27.Oct.2018	✓	✓	✓	✓	✓	4,5
05 Lunes 29.Oct.2018	✓	✓	✓	✓	✓	1,5
06 Miércoles 31.Oct.2018	✓	✓	✓	✓	✗	1,5
07 Lunes 05.Nov.2018	✓	✓	O	✓	✓	1,5
08 Lunes 12.Nov.2018	✓	✓	✓	✓	✓	1,5
09 Miércoles 14.Nov.2018	✓	✓	✓	✓	✓	1,5
09 Lunes 19.Nov-2018	✓	✓	✓	✓	✗	1,5
10 Miércoles 21.Nov-2018	✓	✓	✓	✓	✓	1,5
11 Lunes 03.Dic-2018	✓	✓	✓	✓	✓	1,5
12 Lunes 10.Dic-2018	✓	✓	✓	✓	✓	1,5
13 Martes 11.Dic-2018	✓	✓	✓	✓	✓	3,0
14 Miércoles 12.Dic-2018	✓	✓	✓	✓	✓	1,5
15 Lunes 17.Dic-2018	✓	✓	✓	✓	✓	1,5

16 Miércoles 26.Dic-2018	✓	✓	✓	✓	O	1,5
17 Miércoles 02.Ene-2019	✓	✓	✓	O	O	3,0
18 Viernes 04.Ene-2019	✓	✓	✓	✓	✓	3,0
19 Miércoles 09.Ene-2019	✓	✓	✓	✓	O	1,5
20 Miércoles 16.Ene-2019	✓	✓	✓	✓	✓	3,0
21 Sábado 19.Ene-2019	✓	✓	✓	✓	O	12,0
22 Lunes 21.Ene-2019	✓	✓	✓	✓	✓	1,5

Se debe tomar en consideración, que las horas de trabajo especificadas en este documento sólo corresponden a las horas trabajadas durante las reuniones y no añaden las horas que se dedicaron individualmente por cada integrante del equipo en el proyecto.