|  |
| --- |
| Descripción de Diseño de Software |
| SDD de SnoutPoint |
| **SnoutPoint Network**  Camilo Oviedo Lizarazo  Esteban Hernández Losada  Fabiana Díaz Cedeño  Camilo Benavides Franco  Sebastián Jiménez Nieto  Alejandra Rocha Sabogal  David Suarez Guerrero |

# Historial de Cambios

|  |  |  |  |
| --- | --- | --- | --- |
| **Historial de Cambios** | | | |
| Versión | Cambios efectuados | Fecha de Actualización | Área(s)  Encargada(s) |
| **V0.1** | Documento estructurado para su desarrollo.  Agregado 5. Introducción. | 8 de Mayo de 2015 | EAD |
| **V0.2** | Agregada la arquitectura del sistema (Numeral 6) | 25 de Mayo de 2015 | EAD |
| **V0.3** | Agregado la representación y explicación de los diagramas de comportamiento del sistema (Numeral 7) | 28 de Mayo de 2015 | EAD |
| **V0.4** | Se agrega La estructura del sistema con su respectiva explicación detallada, además, se corrigen algunos diagramas del comportamiento del sistema (Numeral 7) | 30 de Mayo de 2015 | EAD |
| **V0.5** | Se agrega detalle en la Estructura del sistema, además de la sección de Persistencia. | 30 de Mayo de 2015 | EAD |
|  |  |  |  |

# 2. Tabla de Contenidos

Propósito: Que el lector encuentre rápidamente una sección específica del documento

# 3. Lista de Figuras

Propósito: Que el lector encuentre rápidamente una figura específica del documento

# 4. Lista de Tablas

Propósito: Que el lector encuentre rápidamente una tabla específica del documento

# 5. Introducción

El documento de Descripción de Diseño de Software (Software Design Description o SDD) es una representación del diseño del software que será utilizada para registrar la información de diseño y comunicar esa información para la construcción del código a partir del diseño (IEEE Standards Board, 2009).

Para el caso de SnoutPoint no será la excepción, pues este documento tendrá consignados todos los modelos, diagramas y artefactos de diseño relevantes para la construcción del software, con una correspondiente descripción para que el equipo de análisis y diseño, así como el equipo de desarrollo y pruebas y la gerencia puedan tener una abstracción sobre la composición del sistema en aspectos físicos, lógicos y funcionales, facilitando también la comprensión de como el sistema estará compuesto, que elementos de software y hardware requiere, cómo sería el comportamiento ideal del mismo, como se almacenará y manipulará la información y cuál será la interacción de SnoutPoint con el usuario final.

# 6. Arquitectura

Por medio de la arquitectura del sistema, se busca explicar y mostrar el comportamiento del sistema en términos generales, en un alto nivel de abstracción para poder desarrollar el diseño detallado ([ver Diseño Detallado](#_7._Diseño_Detallado)).

## Vista Lógica del Sistema

Por medio del Diagrama de componentes se mostrará los diferentes artefactos que representan elementos concretos del mundo real que son resultado de los procesos de desarrollo del equipo de análisis y diseño (UML Diagrams - OMG, 2014). Aquí se tienen en cuenta también aspectos de interfaces, puertos y relaciones entre componentes.

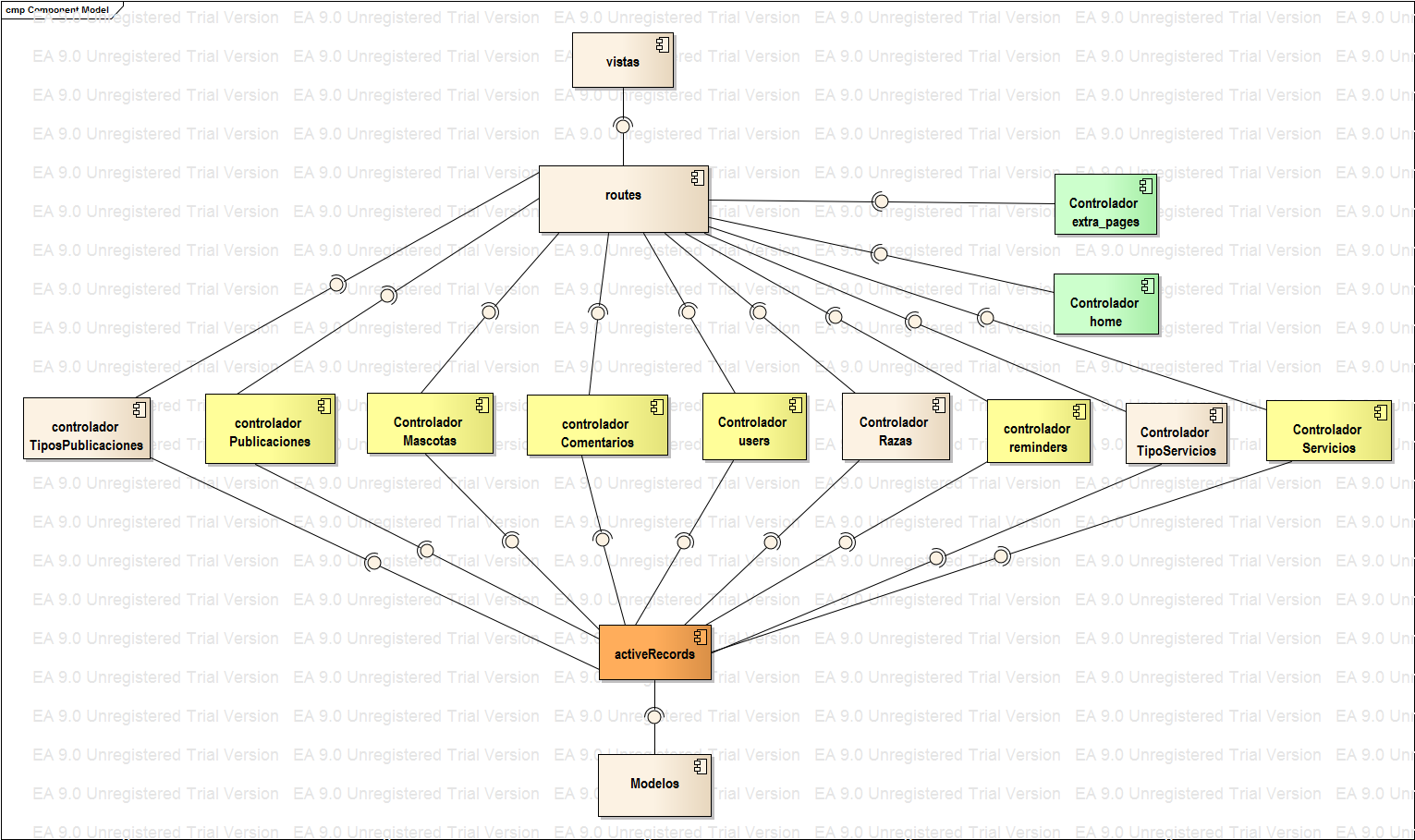
En este caso, por tratarse de una página web, se utiliza un patrón de Modelo-Vista-Controlador que facilita el desarrollo en Ruby on Rails (Rails Core Team), por lo que los componentes del sistema se encontrarán segmentados entre Modelos, Vistas y Controladores, según el comportamiento buscado y la información mínima que debe tener el sistema según los casos de uso y requerimientos levantados anteriormente.

Ilustración 1. Diagrama de Componentes

El Diagrama de componentes muestra que existe una serie de modelos, los cuales contienen todos los datos e información que sea necesaria guardar y buscar su persistencia para su reutilización en las operaciones del sistema, almacenándolo como registros en una base de datos.

Gracias al framework que ofrece Ruby on Rails, por medio del componente de Active Records que provee una interfaz para realizar consultas desde los modelos (Hansson, 2015), facilitándole a los controladores el obtener información cada que se les solicite y poder aplicar las modificaciones dentro de los modelos, asegurando la consistencia de la información.

Se determina que se debe tener un controlador para las entidades principales del modelo: Usuario, Mascota, Servicios, Publicaciones, Comentarios y Recordatorios. Cada controlador se encarga de realizar operaciones sobre la información de los modelos según las solicitudes que reciba por parte del cliente. También se tienen un controlador de home y uno de extra pages, los cuales se encargan de cargar la información de páginas que no requieran de entidades u operaciones lógicas, como la redirección a la página de Términos y Condiciones de SnoutPoint o la carga de la página inicial de SnoutPoint.

El componente de routes (David Heinemeier Hansson, 2015) es el que determina cuál es el controlador que se va a utilizar según lo que el usuario realice dentro de la página y cual vista se debe desplegar según la operación realizada dentro de la página. Este determina cual vista debe cargarse para el cliente y qué acciones de qué controlador se deben ejecutar para el despliegue de esta vista.

## Vista Física del Sistema

La vista física del sistema servirá para entender cuáles son los componentes físicos en los cuales el sistema va a ser instalado y realizará sus operaciones. (IEEE Standards Board, 2009) Para realizar esta vista, se utiliza un diagrama de despliegue que explica a grandes rasgos cual es la composición a nivel de hardware y archivos de SnoutPoint, para entender cómo será instalado y como debe ser su organización interna a nivel de carpetas, paquetes y archivos.

Dado que uno de los principales requerimientos del sistema es que debe cumplir una arquitectura de cliente-servidor (Pavlich & Torres, 2015), se mantendrá la aplicación en un servidor web que podrá ser accedido mediante HTTP sobre el protocolo TCP/IP por parte de los clientes que deseen utilizar nuestra red social.

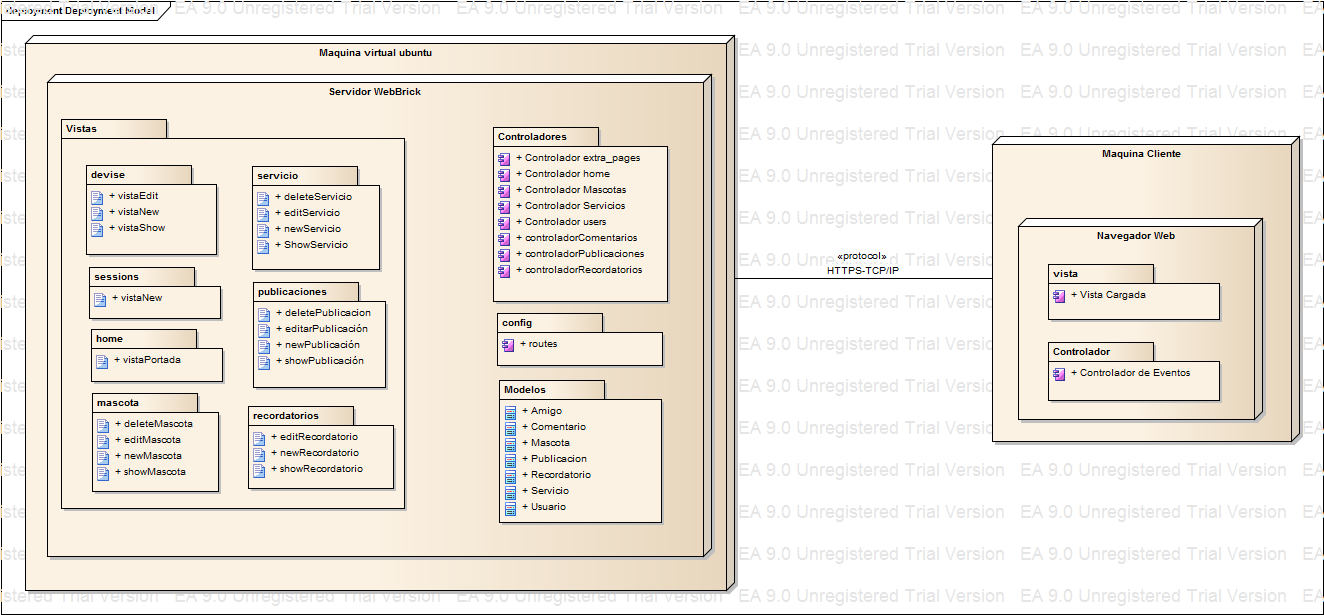


Ilustración . Diagrama de Despliegue

El diagrama de despliegue se cuentan con dos máquinas: una que será utilizada por el cliente y otra que será el servidor donde está alojado el sistema para su uso. El cliente debe tener un navegador web (Google Chrome recomendado), el cual despliega una vista y posee un controlador de eventos que recibe las acciones del usuario para mandárselas al servidor.

En cuanto al servidor, éste tendrá como se mencionó anteriormente el componente *routes,* que es el que se encarga de determinar cuál vista se va a desplegar al cliente y cuales controladores va a utilizar. Por otra parte, las vistas, los modelos y controladores estarán agrupados en sus paquetes respectivos, de tal manera que sea más fácil ubicar algunos de estos tres componentes mencionados del sistema.

Cabe resaltar que las vistas son unos conjuntos de archivos HTML por lo cual no poseen una mayor complejidad funcional: simplemente es texto. En contraste, los controladores si son componentes al realizar una serie de operaciones con los modelos. Adicionalmente, cada modelo es una clase que determina su relación con otras clases para mantener la correspondencia en el momento de hacer tablas de estas clases mencionadas, lo cual genera la dependencia de las tablas con respecto a otras ya que todas las clases del modelo heredan de ActiveRecords para que así puedan realizarse las consultas a las tablas (Hansson, 2015).

## Vista de Procesos del Sistema

En la vista de procesos de sistema se muestra de manera general, cuales son las acciones que el usuario puede realizar mientras se encuentra dentro del sistema (IEEE Standards Board, 2009). Esto es, mostrar que cuáles son las posibles acciones que se pueden realizar el usuario y cuál es la salida o resultado de esta acción. En este caso, se decide usar un esquema de procesos de manera que se muestren a grandes rasgos los procesos del sistema y posteriormente especificados dentro del comportamiento del sistema (ver el comportamiento del sistema).

Como se puede ver en la ilustración 3, se inicia el sistema ingresando a la página de SnoutPoint, posteriormente determina si inicia sesión o si desea registrarse en el sistema para luego iniciar sesión. Una vez que inicia sesión puede hacer uso de cualquiera de los servicios por la página, uno a la vez (No se pueden hace uso de más de un servicio al mismo tiempo). Adicionalmente, cuando se finaliza el proceso de cada servicio se puede seleccionar otro o simplemente cerrar sesión para dejar de usar el sistema.

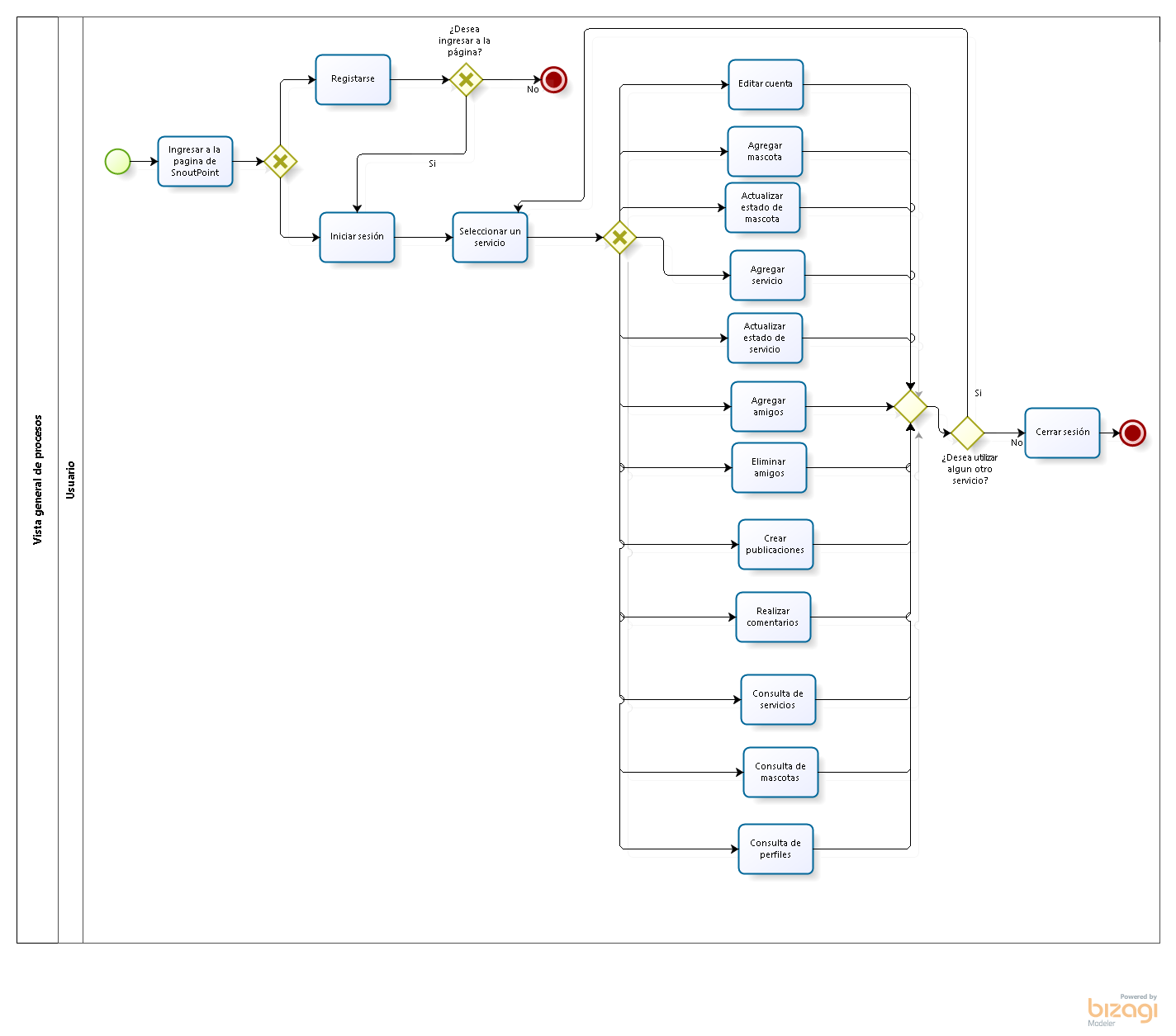


Ilustración . Vista General de Proceso

# 7. Diseño Detallado

Teniendo en cuenta la arquitectura del sistema realizado y detallado en el numeral anterior se especifican los aspectos relevantes para la implementación y entendimiento a bajo nivel del sistema de tal manera, que tanto para desarrollar el sistema como para que evolucione se puede entender cuál es la estructura del sistema, cómo se comporta, cómo se almacena la información y de qué manera el usuario interactúa con el sistema a través de la interfaz.

## Estructura del Sistema

En el caso de la estructura del sistema, se hace una especificación de las clases que hacen parte de los componentes mencionados en la vista lógica del sistema ( ver vista lógica del sistema), teniendo en cuenta que no todos poseen una representación de forma de clases.

Los modelos estructuran que datos se van a almacenar, operar, actualizar o eliminar dentro del sistema, además de la relación mutua que poseen estos datos. Los modelos pueden verse representados como un diagrama de clases, solo mostrando que datos contiene cada clase del modelo y cuales relaciones mantienen entre sí. Cabe resaltar que cada una de estas clases hereda de ActiveRecords para que tengan una tabla asociada en la base de datos del sistema (ver Persistencia).

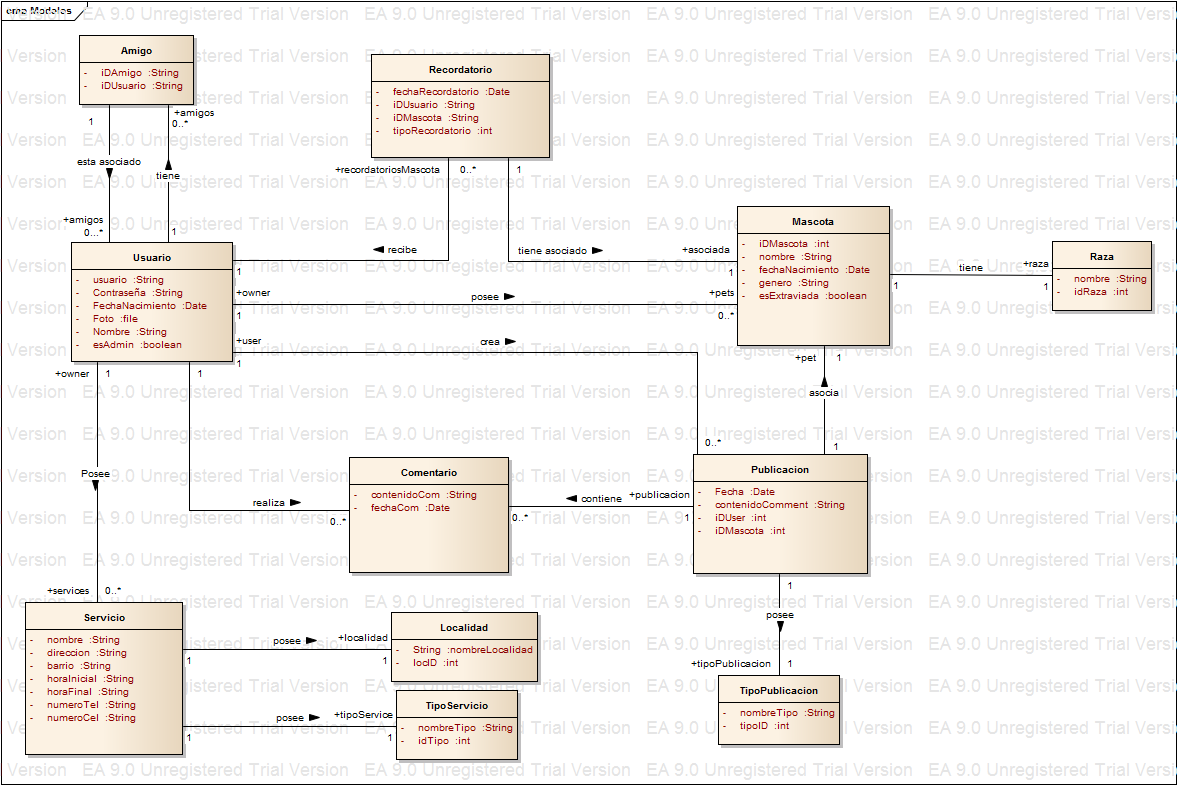


Ilustración . Diagrama de Clases de Modelos

|  |  |
| --- | --- |
| Elementos | Id |
| Usuario | C1 |
| Servicios | C2 |
| Amigo | C3 |
| Recordatorio | C4 |
| Mascota | C5 |
| Comentario | C6 |
| Publicación | C7 |
| Raza | C8 |
| Localidad | C9 |
| TipoServicio | C10 |
| TipoPublicacion | C11 |

Tabla . Nomenclatura de las clases de modelos

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | C1 | | **Elemento** | Usuario | |
| **Descripción** | El usuario es el encargado de interactuar en SnoutPonit | | | | |
| **Atributos** | | | | | |
| **Nombre** | | **Descripción** | | | **Tipo de Dato** |
| **ID** | | Este es un Id el cual es un único identificador del usuario. | | | String |
| **Contraseña** | | Es la clave que el usuario crea, la cual será solicitada cuando este ingrese. | | | String |
| **FechaNacimiento** | | Es la fecha de nacimiento del usuario. | | | Date |
| **Foto** | | Es una foto que identifica al usuario, si este desea poner. De lo contrario se pondrá una imagen establecida. | | | File |
| **Nombre** | | Es el nombre del usuario. | | | String |
| **esAdmin** | | Este indica si un usuario un usuario administrador o si es un usuario común. | | | boolean |
| **Relaciones** | | | | | |
| **Nombre** | | **Descripción** | | | **Tipo de relación** |
| **Amigo** | | Son las personas con las que el usuario se puede poner en contacto. | | | Un usuario tiene 0 o muchos amigos.  Un usuario está asociado a 0 o muchos amigos |
| **Servicio** | | Son los servicios que el usuario desea ofrecer en SnoutPoint. | | | Un usuario posee 0 o muchos servicios |
| **Comentario** | | Son los comentarios que hace el usuario acerca de una publicación. | | | Un usuario realizar 0 o muchos comentarios |
| **Recordatorio** | | Son los recordatorios asociados a las mascotas, que les notificaran a los usuarios las fechas importantes que este desee que le recuerden. | | | Un usuario recibe 0 o muchos recordatorios |
| **Mascota** | | Son las mascotas que desee tener el usuario en su cuenta. | | | Un usuario posee 0 o muchas mascotas |
| **Publicación** | | Son las publicaciones que el usuario desee hacer ya sea de sus mascotas o servicios asociados. | | | Un usuario crea 0 o muchas publicaciones |
| **Objetivo** | Interactuar con el fin de dar a conocer las mascotas o servicios que desee y así obtener beneficios de esto. | | | | |

Tabla . Descripción de la clase Usuario

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | C2 | | **Elemento** | Servicio | |
| **Descripción** | Son los servicios que un usuario desea ofrecer en SnoutPoint | | | | |
| **Atributos** | | | | | |
| **Nombre** | | **Descripción** | | | **Tipo de Dato** |
| **Nombre** | | Nombre del servicio que se desea prestar | | | String |
| **Dirección** | | Dirección en donde se encuentra ubicado dl servicio | | | String |
| **numeroTel** | | Número de teléfono del servicio | | | String |
| **numeroCel** | | Numero de celular del servicio | | | String |
| **Barrio** | | Barrio en donde se encuentra ubicado el servicio | | | String |
| **HoraInicial** | | Es la hora en la cual se abre el lugar del servicio | | | String |
| **HoraFinal** | | Es la hora en la cual se cierra el lugar del servicio | | | String |
| **Relaciones** | | | | | |
| **Nombre** | | **Descripción** | | | **Tipo de relación** |
| **Usuario** | | Es el usuario que maneja la cuenta del servicio | | | 0 o muchos servicios son asociados a un usuario |
| **Localidad** | | Es la localidad en la que se encuentra el servicio | | | Un servicio tiene una localidad |
| **TipoServicio** | | Es el tipo de servicio que se pude prestar en SnoutPoint | | | Un servicio puede ser de un tipo |
| **Objetivo** | Ofrecer distintos servicios para las mascotas, ya sea veterinarias, fundaciones, tiendas. | | | | |

Tabla . Descripción de la clase Servicio

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | C3 | | **Elemento** | Amigo | |
| **Descripción** | Son los demás usuario que se relacionan | | | | |
| **Atributos** | | | | | |
| **Nombre** | | **Descripción** | | | **Tipo de Dato** |
| **IdAmigo** | | El id que identifica a el amigo que está asociado al usuario | | | String |
| **IdUsuario** | | Es el id que identifica al usuario | | | String |
| **Relaciones** | | | | | |
| **Nombre** | | **Descripción** | | | **Tipo de relación** |
| **Usuario** | | Es el usuario encargado de agregar y tener a los amigos | | | 0 o muchos amigos están asociados a un usuario. |
| **Objetivo** | Interactuar con los demás usuarios | | | | |

Tabla . Descripción de la clase Amigo

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | C4 | | **Elemento** | Recordatorio | |
| **Descripción** | Son los eventos importantes de las mascotas | | | | |
| **Atributos** | | | | | |
| **Nombre** | | **Descripción** | | | **Tipo de Dato** |
| **fechaRecordatorio** | | Fecha del recordatorio | | | Date |
| **IdUsuario** | | Id del usuario al cual se le enviará el recordatorio. | | | String |
| **IdMascota** | | Id de la mascota de la cual se hará el recordatorio. | | | String |
| **tipoRecordatorio** | | Es el tipo de recordatorio que se va a notificar. | | | Int |
| **Relaciones** | | | | | |
| **Nombre** | | **Descripción** | | | **Tipo de relación** |
| **Mascota** | | Mascota de la cual se hará el recordatorio | | | Un recordatorio está asociado a una mascota |
| **Usuario** | | Usuario al cual se le enviara el recordatorio | | | 0 o muchos recordatorios son enviados a un usuario. |
| **Objetivo** | Notificar al usuario de los eventos que anteriormente éste solicito que le recordaran. | | | | |

Tabla . Descripción de clase Recordatorio

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | C5 | | **Elemento** | Mascota | |
| **Descripción** | Es la mascota que maneja el usuario | | | | |
| **Atributos** | | | | | |
| **Nombre** | | **Descripción** | | | **Tipo de Dato** |
| **IdOwner** | | Id que identifica al dueño de la mascota | | | String |
| **IdMascita** | | Id identificador de cada mascota | | | String |
| **Animal** | | El tipo de animal que sea la mascota, ya sea perro o gato | | | String |
| **Edad** | | Años o meses que tenga de vida la mascota | | | Int |
| **FechaNacimiento** | | Fecha de nacimiento de la mascota | | | Date |
| **Genero** | | Si la mascota es macho o hembra | | | String |
| **Nombre** | | Nombre de la mascota | | | String |
| **esExtraviada** | | Un identificador para saber si la mascota esta extraviada | | | String |
| **Relaciones** | | | | | |
| **Nombre** | | **Descripción** | | | **Tipo de relación** |
| **Recordatorio** | | Es el recordatorio que notifica sobre eventos de la mascota. | | | Una mascota está asociada a un recordatorio |
| **Publicación** | | Son las publicaciones que se realizan acerca de las mascotas. | | | Una publicación tiene asociada una mascota. |
| **Usuario** | | Es el usuario que tiene a las mascotas. | | | 0 o muchas mascotas están asociadas a un usuario. |
| **Raza** | | Es la raza que tiene cada mascota | | | Una mascota tiene una raza |
| **Objetivo** | Poder interactuar con otras mascotas y con los servicios ofrecidos | | | | |

Tabla . Descripción de clase Comentario

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | C6 | | **Elemento** | Comentario | |
| **Descripción** | Son los comentarios que se desean hacer en las publicaciones. | | | | |
| **Atributos** | | | | | |
| **Nombre** | | **Descripción** | | | **Tipo de Dato** |
| **contenidoCom** | | Es lo que el usuario escribe en el cometario. | | | String |
| **fechaCom** | | Fecha en la cual se realizó el comentario. | | | Date |
| **Relaciones** | | | | | |
| **Nombre** | | **Descripción** | | | **Tipo de relación** |
| **Usuario** | | Es el encargado de hacer el comentario | | | 0 o muchos comentarios son realizados por un usuario |
| **Publicación** | | Es donde se pone el comentario | | | 0 o muchos comentarios están asociados a una publicaciones |
| **Objetivo** | Interactuar con las publicaciones de las mascotas. | | | | |

Tabla . Descripción de la clase Mascota

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | C7 | | **Elemento** | Publicación | |
| **Descripción** | Fotos o acontecimientos importantes que desee publicar una persona sobre su mascota | | | | |
| **Atributos** | | | | | |
| **Nombre** | | **Descripción** | | | **Tipo de Dato** |
| **Contenido** | | Contenido que tenga la publicación | | | String |
| **Fecha** | | Fecha en la que se realizó la publicación | | | Date |
| **Relaciones** | | | | | |
| **Nombre** | | **Descripción** | | | **Tipo de relación** |
| **Mascota** | | Es la mascota de la cual se hace la publicación | | | Una publicación está asociada a una mascota |
| **Usuario** | | Es el usuario que realiza la publicación | | | 0 o muchas publicaciones es realizada por un usuario |
| **Comentario** | | Son los comentarios que se hacen a la publicación | | | Una publicación tiene 0 o muchos comentarios |
| **TipoPublicacion** | | Es el tipo de publicación que se puede hacer | | | Una publicación tiene un tipo de publicación |
| **Objetivo** | Dar a conocer información adicional sobre cada mascota de la persona | | | | |

Tabla . Descripción de la clase Publicación

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | C8 | | **Elemento** | Raza | |
| **Descripción** | Raza de la mascota | | | | |
| **Atributos** | | | | | |
| **Nombre** | | **Descripción** | | | **Tipo de Dato** |
| **Nombre** | | Nombre de la raza | | | String |
| **Id** | | Id identificador de cada raza | | | Int |
| **Animal** | | Tipo de animal (gato o perro) | | | String |
| **Relaciones** | | | | | |
| **Nombre** | | **Descripción** | | | **Tipo de relación** |
| **Mascota** | | Es la mascota de la cual se asigna la raza correspondiente | | | Una raza es asignada a una mascota |
| **Objetivo** | Dar consistencia en cuanto a la raza de cada tipo de mascota. | | | | |

Tabla . Descripción de clase Raza

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | C9 | | **Elemento** | Localidad | |
| **Descripción** | Localidad donde se encuentra el servicio | | | | |
| **Atributos** | | | | | |
| **Nombre** | | **Descripción** | | | **Tipo de Dato** |
| **nombreLocalidad** | | Nombre de la localidad | | | String |
| **Id** | | Id identificador de cada localidad | | | Int |
| **Relaciones** | | | | | |
| **Nombre** | | **Descripción** | | | **Tipo de relación** |
| **Servicio** | | Es el servicio prestado por SnoutPoint | | | Cada localidad pertenece a un servicio |
| **Objetivo** | Dar a conocer la lista de localidades que existen en la cuidad de Bogotá para que el usuario escoja en cual se encuentra su servicio ofrecido. | | | | |

Tabla . Descripción de clase Localidad

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | C10 | | **Elemento** | TipoServicio | |
| **Descripción** | Los diferentes tipos de servicio que hay en SnoutPoint | | | | |
| **Atributos** | | | | | |
| **Nombre** | | **Descripción** | | | **Tipo de Dato** |
| **nombreTipo** | | Nombre del tipo de servicio | | | String |
| **Id** | | Id identificador de tipo de servicio | | | Int |
| **Relaciones** | | | | | |
| **Nombre** | | **Descripción** | | | **Tipo de relación** |
| **Servicio** | | Es el servicio prestado por SnoutPoint | | | Un tipo de servicio es asignado a un servicio |
| **Objetivo** | Tener una lista de los servicios que se pueden ofrecer en SnoutPoint para que no existan inconsistencias. | | | | |

Tabla . Descripción de clase TipoServicio

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | C11 | | **Elemento** | TipoPublicacion | |
| **Descripción** | Los diferentes tipos de publicación que se pueden hacer | | | | |
| **Atributos** | | | | | |
| **Nombre** | | **Descripción** | | | **Tipo de Dato** |
| **nombreTipo** | | Nombre del tipo de publicación | | | String |
| **Id** | | Id identificador de tipo de publicacion | | | Int |
| **Relaciones** | | | | | |
| **Nombre** | | **Descripción** | | | **Tipo de relación** |
| **Publicacion** | | Son las publicaciones que se pueden hacer en SnoutPoint | | | Un tipo de publicación es asignado a una publicación |
| **Objetivo** | Tener una lista de los tipos de publicaciones que se pueden generar en SnoutPoint | | | | |

Tabla . Descripción de clase TipoPublicación

Por su parte, cada controlador es una clase que tiene relación con una clase de modelo determinada (el controlador de mascotas tiene relación con la clase de mascota y la tabla de mascotas, por ejemplo). Cada controlador se encargará de realizar operaciones de creación (New para registrar la información y Create generar el registro en la base de datos), lectura (Show), modificación de datos (edit para cambiar los datos y Update para actualizar los cambios) y eliminación (delete para determinar que registro se va a eliminar y destroy para eliminar de la base de datos).

Dado que todos los controladores realizan las mismas operaciones, solo que cambian la información que utilizan y que no tienen ninguna relación mutua como si tienen las clases de modelo, no es necesario realizar un diagrama de clases para estos controladores, pues lo relevante en estos es entender las operaciones de Show, New, Create, Edit, Update, Delete y Destroy que realizan los controladores.

## Comportamiento del Sistema

El objetivo de la sección de comportamiento del sistema es el de conocer los conceptos básicos para el modelado avanzado del comportamiento del sistema; además, el de conocer las características de los diagramas de cada uno de los diagramas realizados. [7]

Por tal motivo, a continuación se encuentran los diagramas que representan el comportamiento del sistema, con su respectiva descripción.

[7] http://www.ctr.unican.es/asignaturas/is1/is1-t12-trans.pdf

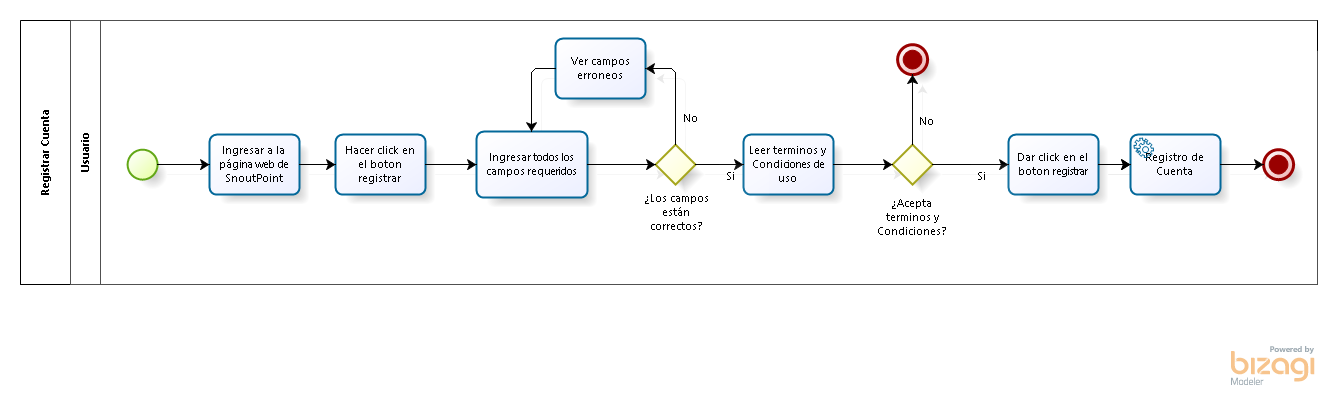


Ilustración . Proceso Registrar Cuenta

En el primer diagrama detallado en la ilustración anterior, se puede observar el proceso para Registrar una Cuenta en el sitio web de SnoutPoint. Cabe resaltar que se realizaron términos y condiciones y se deben tomar en cuenta en el proceso de registrar cuenta.

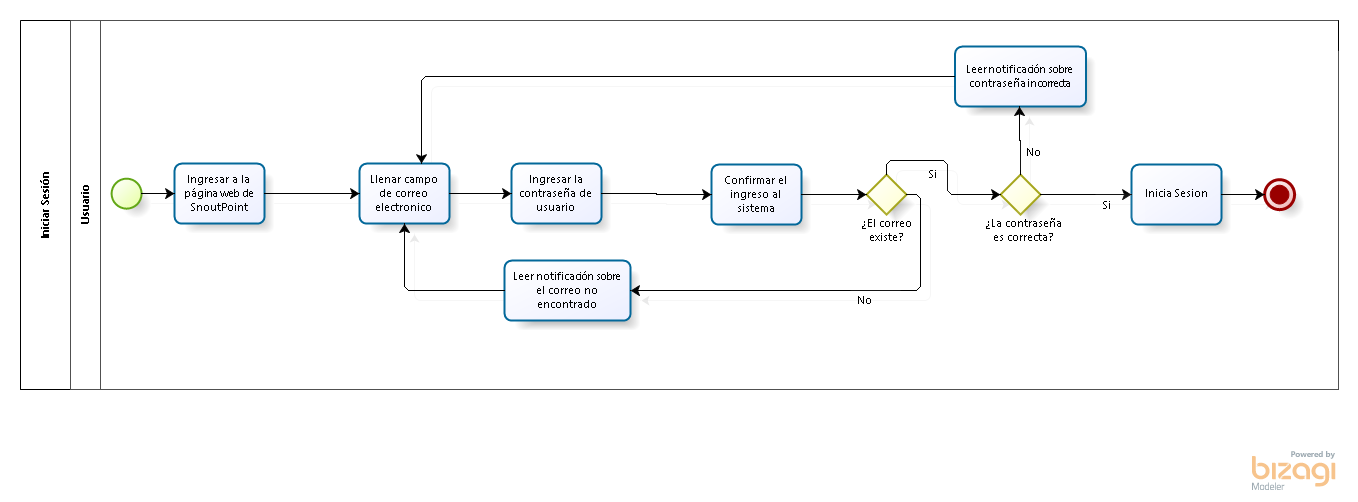


Ilustración . Proceso Iniciar Sesión

Para el diagrama anterior se detalla el proceso de iniciar sesión en el cual el usuario deberá completar los campos de correo y contraseña correctamente antes de poder continuar a la página principal de su cuenta.

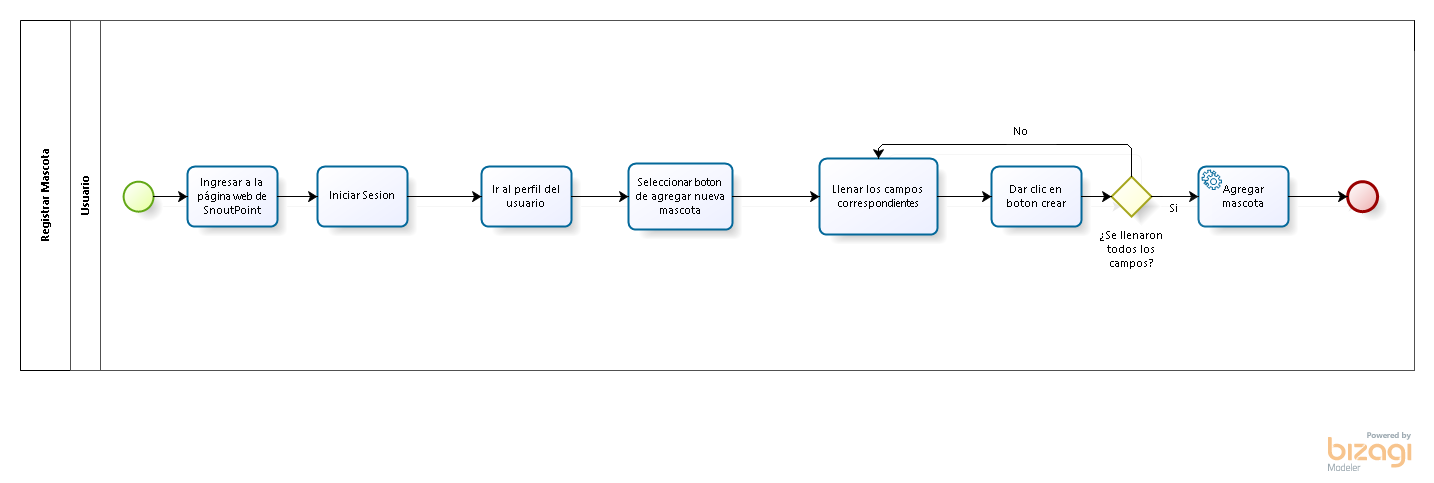


Ilustración . Proceso de Registrar Mascota

Para el diagrama anterior se detalla el proceso de registrar mascota en el cual, el usuario primero debe iniciar sesión en su cuenta para así poder agregar alguna mascota nueva a su cuenta.

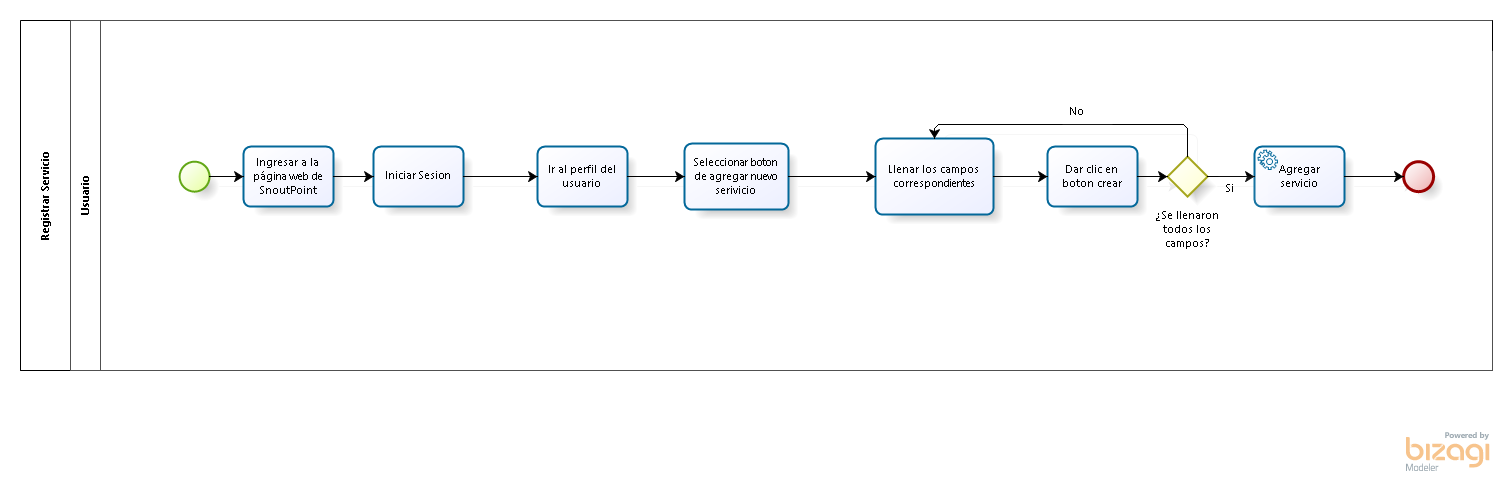


Ilustración . Proceso de Registrar Servicio

Para el diagrama anterior se detalla el proceso de registrar servicio, el cual el usuario debe Iniciar Sesión en primer lugar para luego llenar todos los campos correspondientes y finalmente poder agregar un servicio asociado a su cuenta.

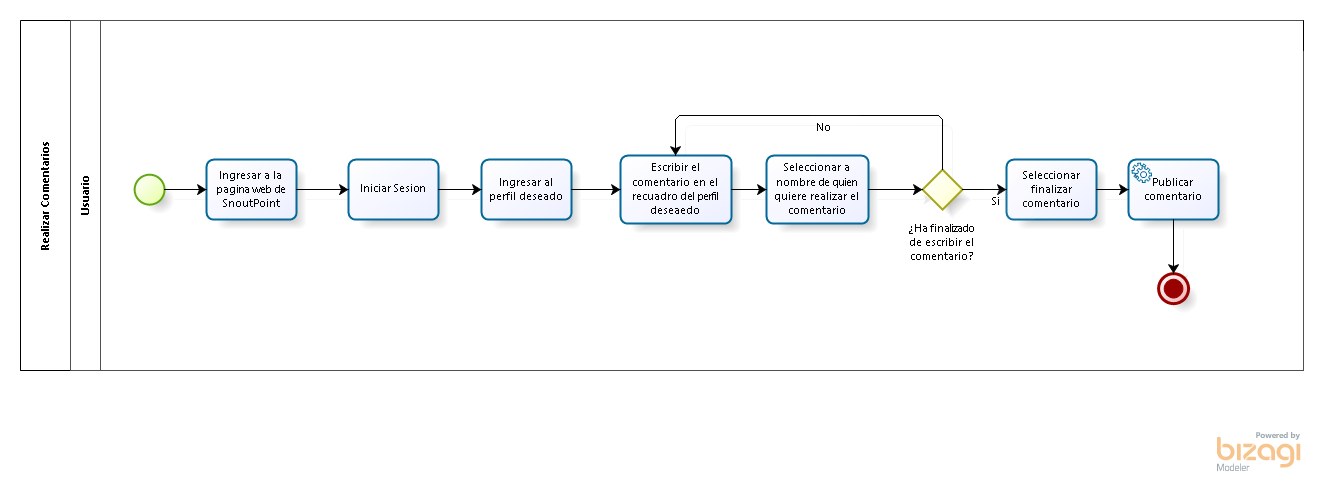


Ilustración . Proceso de Realizar Comentario

En el diagrama anterior se detalla el proceso de realizar comentario, el cual, el usuario debe iniciar sesión en su cuenta antes de poder publicar algún comentario en el perfil de cualquier otro usuario.

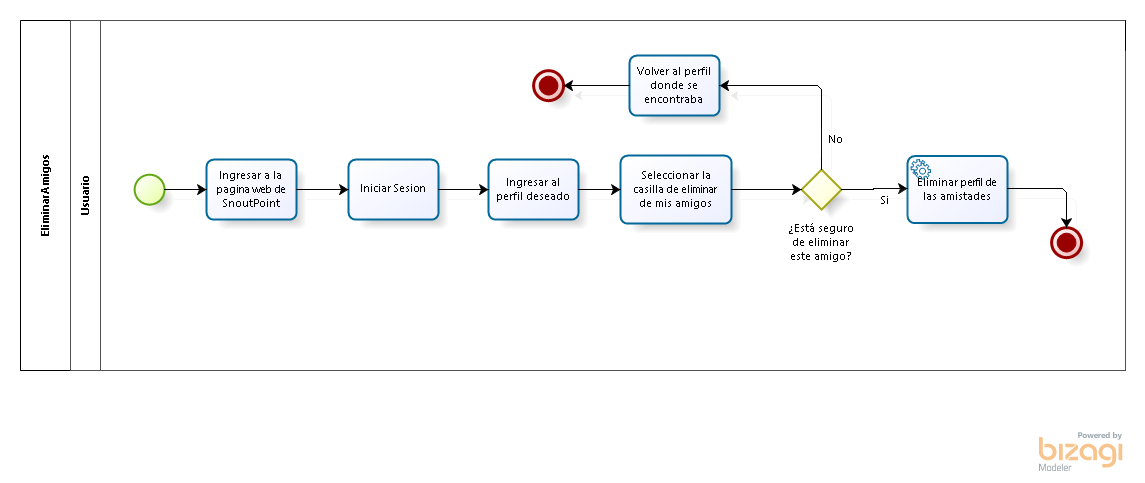


Ilustración . Proceso de Eliminar Amigo

En el diagrama anterior se detalla el proceso de eliminar amigos, el cual se hace un proceso de verificación en dos pasos para saber que el usuario está seguro de querer eliminar este amigo de su cuenta.

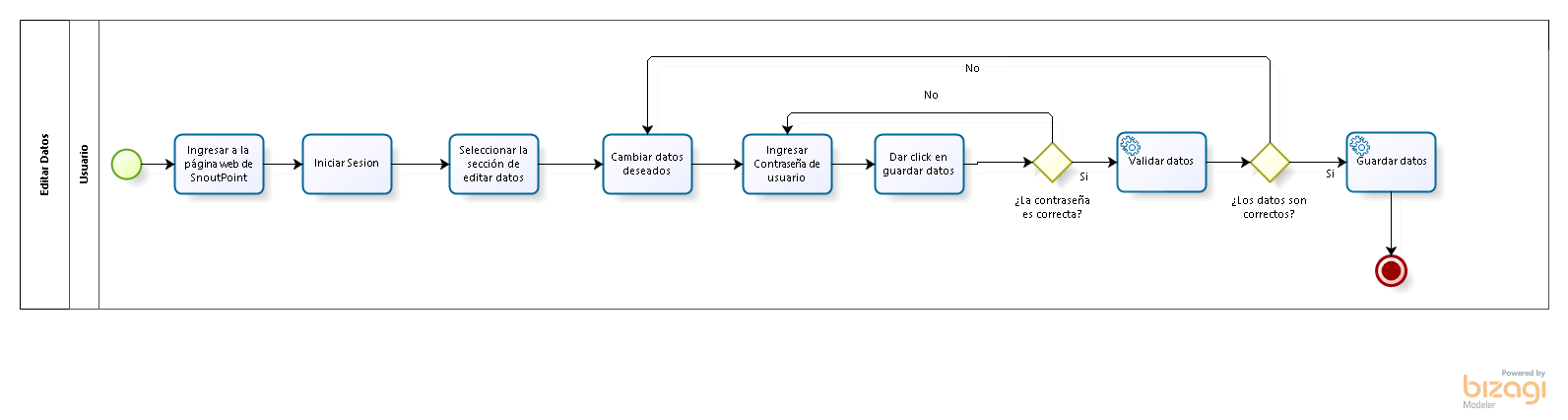


Ilustración . Proceso de Editar Datos

En el diagrama anterior se detalla el proceso de editar datos del usuario, en el que se puede modificar los datos de la cuenta del usuario y al finalizar se debe validar los cambios ingresando la contraseña de usuario.

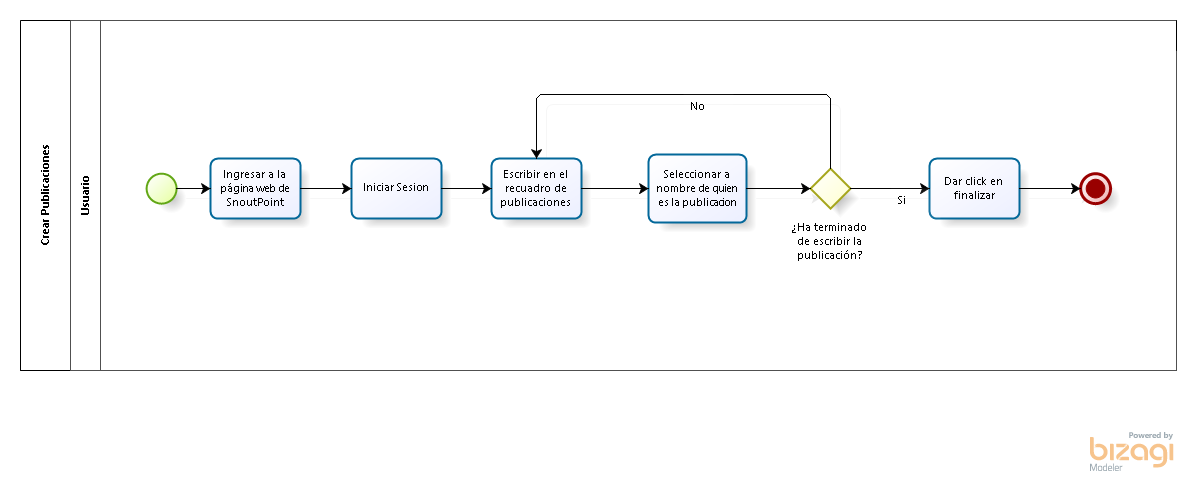


Ilustración . Proceso de Crear Publicaciones

En el diagrama anterior se detalla el proceso de crear publicaciones, en el cual, el usuario crea contenido en su propia cuenta de cosas que le interesan.

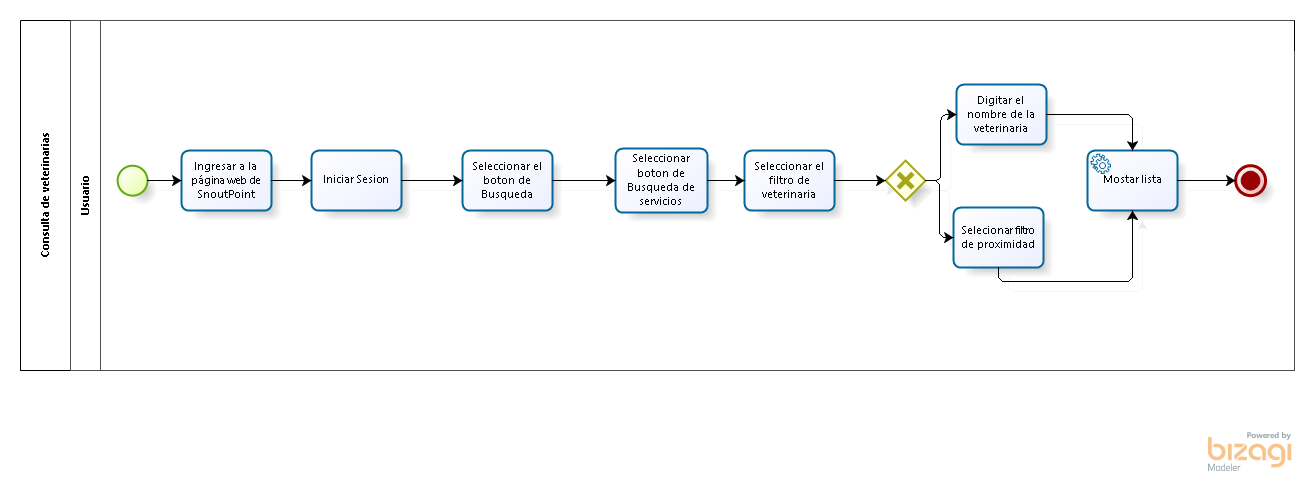


Ilustración . Proceso de Consulta de Veterinarias

En el anterior diagrama se detalla el proceso de consulta de veterinarias en el que se hace una consulta básica con filtros para encontrar rápidamente lo que el usuario desea.

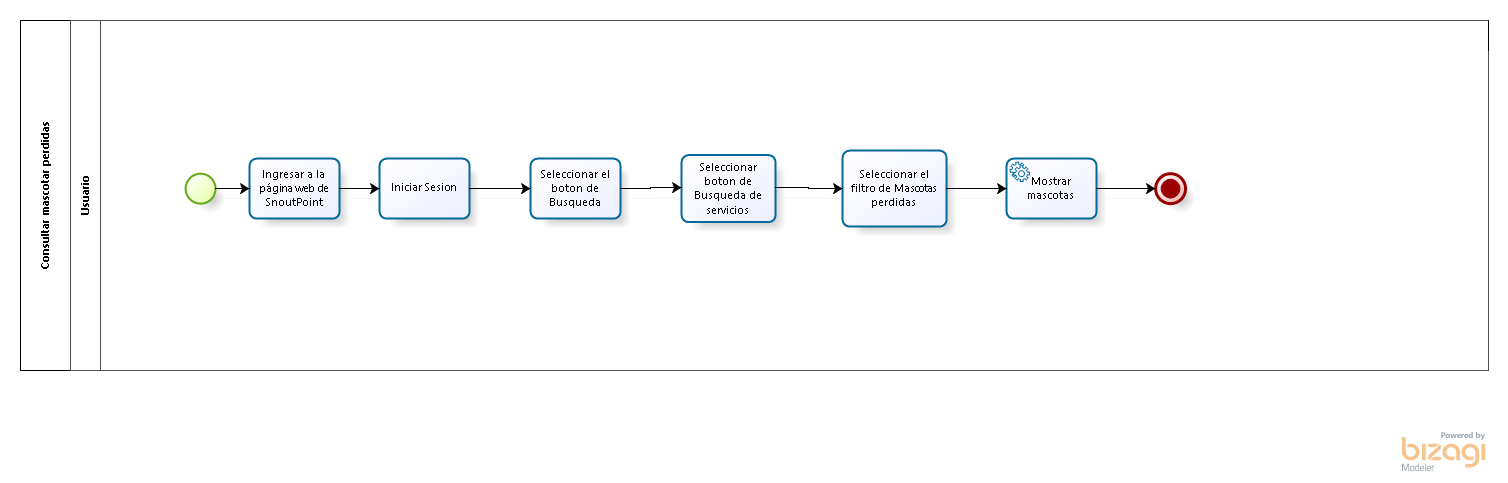


Ilustración . Proceso de Consulta de Mascotas Perdidas

En el anterior diagrama se detalla el proceso de consulta de mascotas perdidas, el cual también mediante filtros se realiza la consulta para encontrar cuales mascotas se han perdido recientemente.

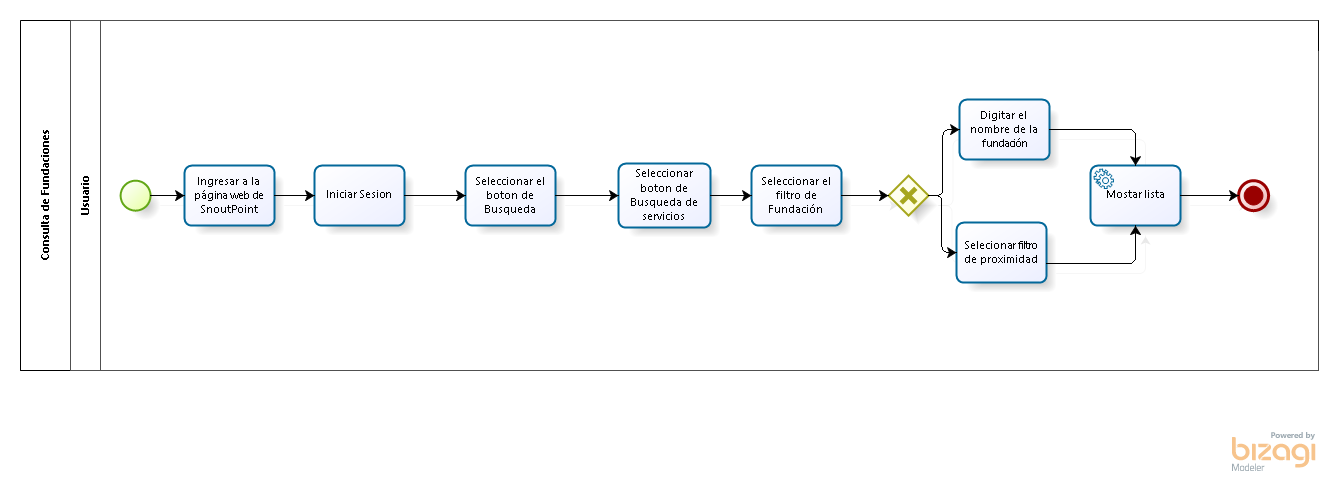


Ilustración . Proceso de Consulta de Fundaciones

En el anterior diagrama se detalla el proceso de consulta de fundaciones, el cual mediante filtros se detalla se puede buscar entre el nombre de la fundación más cercana o también el más próximo al usuario.

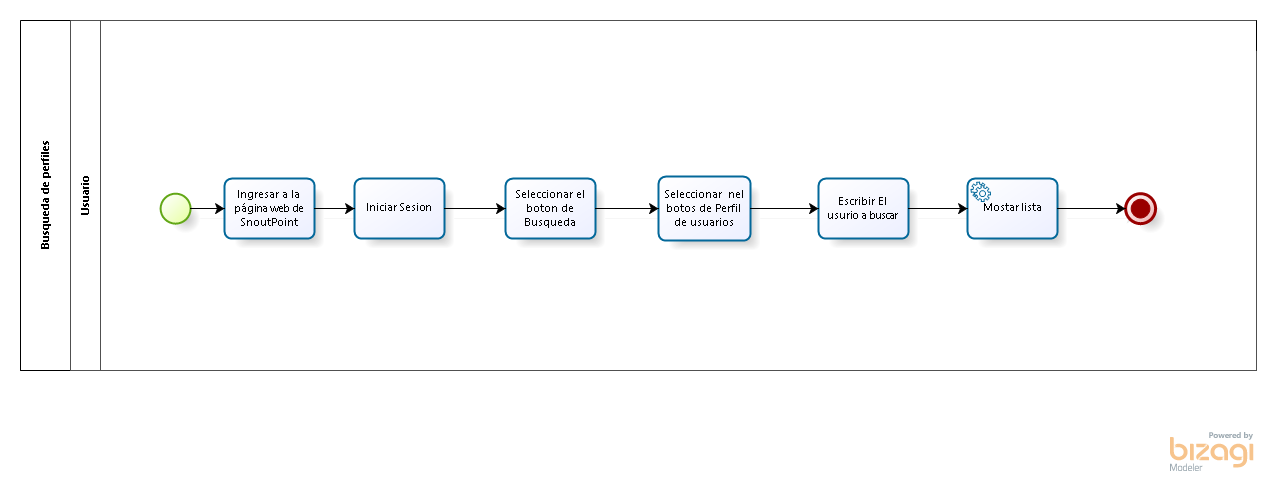


Ilustración . Proceso de Búsqueda de Perfiles

El anterior diagrama detalla el proceso de búsqueda de perfiles el cual existe un área de búsqueda en el que se puede realizar las búsquedas de perfiles.

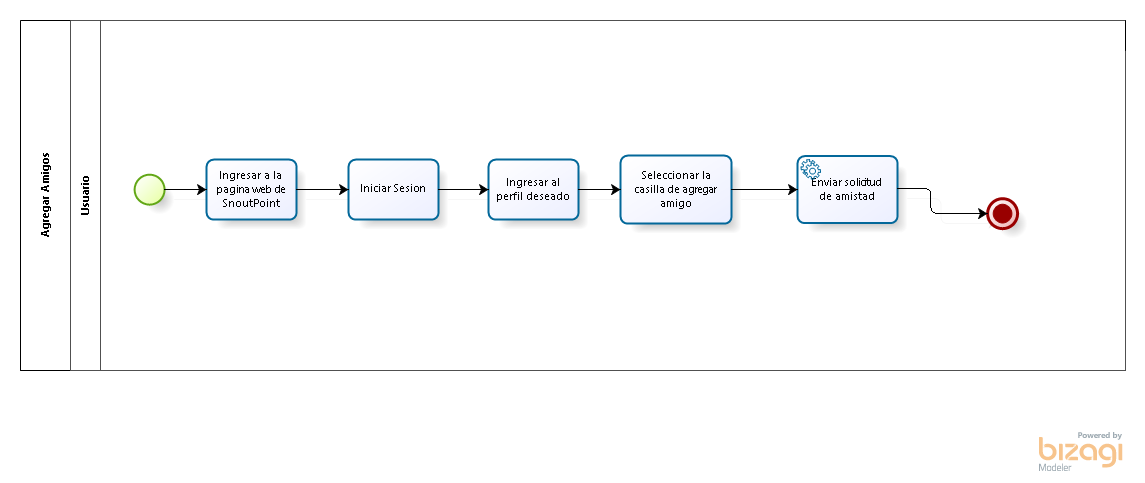


Ilustración . Proceso de Agregar Amigos

En el anterior diagrama se puede detallar el proceso agregar amigos, el cual se debe ingresar al perfil del amigo deseado y ahí aparecerá la opción de agregar amigo para enviar la solicitud de amistad.

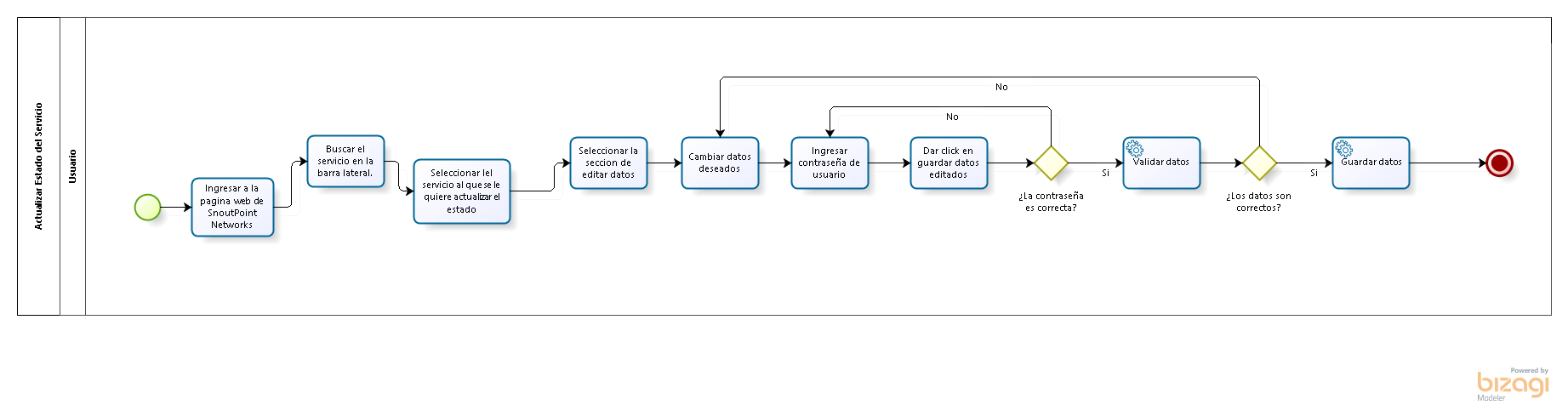


Ilustración . Proceso Actualizar Estado del Servicio

En el anterior diagrama se detalla el proceso de actualizar estado del servicio, que permite cambiar datos del servicio asociado a una cuenta de usuario y al final debe ingresar la contraseña de la cuenta para aceptar los cambios realizados.

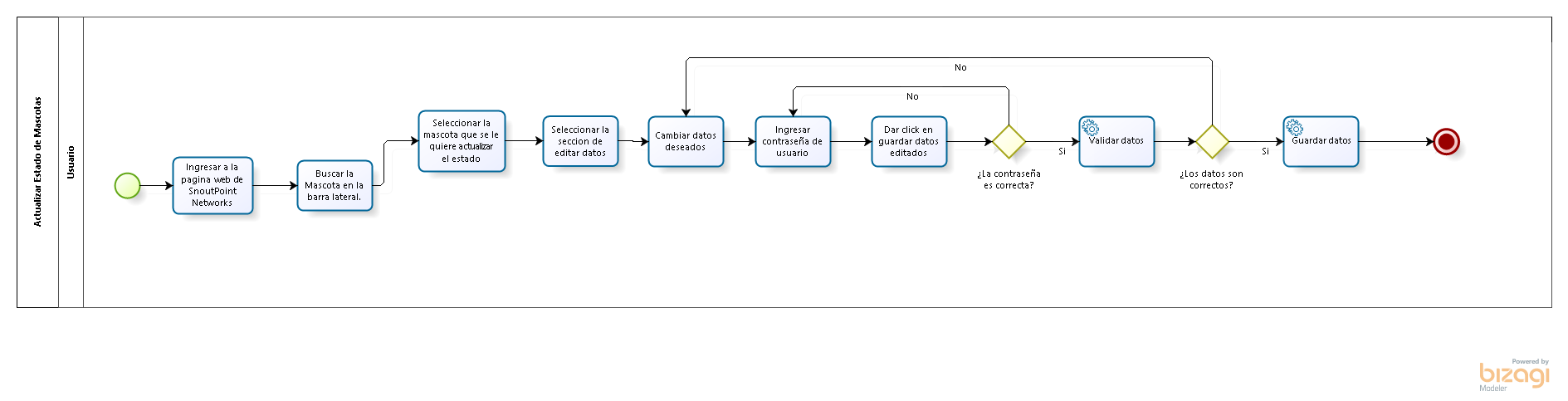


Ilustración . Proceso Actualizar Estado de Mascotas

En el anterior diagrama se detalla el proceso de actualizar el estado de mascota, en el cual el usuario debe estar seguro de cuál mascota modificará para así poder ingresar a la sección de edición y finalmente ingresar la contraseña para guardar los cambios realizados.

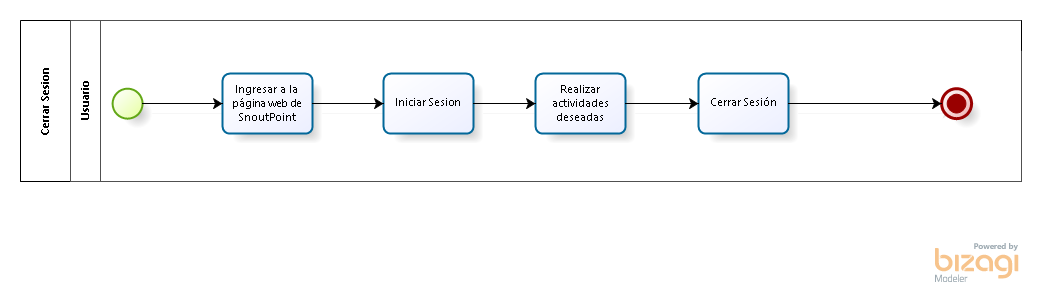


Ilustración . Proceso de Cerrar Sesión

En el anterior diagrama se detalla el proceso de cerrar sesión, el cual es bastante básico debido a que simplemente debe realizar sus actividades deseadas y finalmente dar en la opción cerrar sesión.

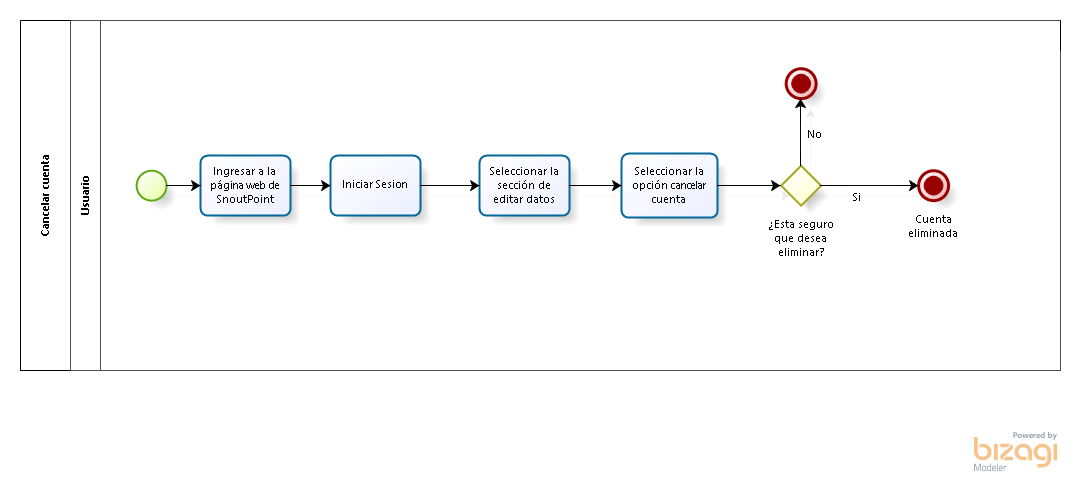


Ilustración . Proceso de Cancelar Cuenta

En el anterior diagrama se detalla el proceso de cancelar la cuenta, en el cual, el usuario cuando quiera eliminar su cuenta permanentemente debe seleccionar la opción de cancelar cuenta y hay doble verificación para que el usuario esté seguro de lo que está realizando.

A partir de este momento se detalla los procesos del SuperUsuario, aquellos que tienen permisos especiales sobre las cuentas en la red social SnoutPoint. Estos usuarios poseen tres procesos básicos que se detallan a continuación.

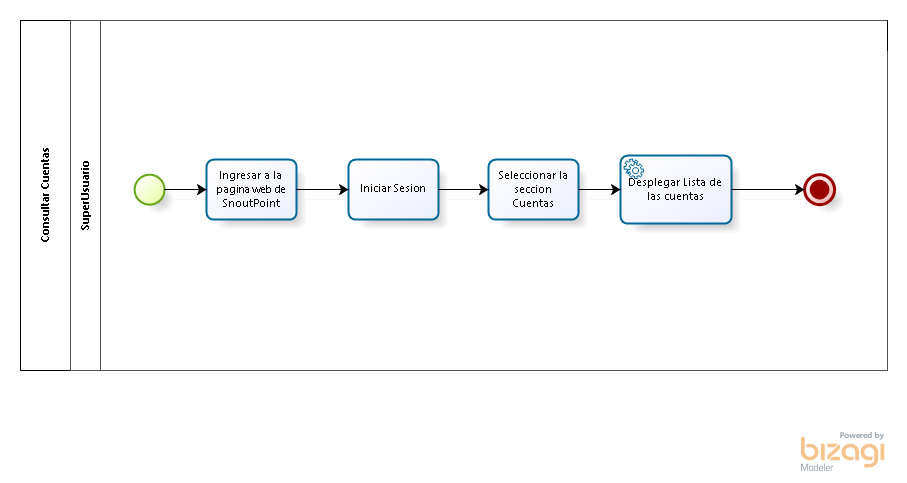


Ilustración . Proceso de Consultar Cuentas (SuperUsuario)

En el anterior diagrama se detalla el proceso de consultar cuentas en un SuperUsuario, el cual permite que el SuperUsuario vea una lista completa de todas las cuentas registradas en la red social SnoutPoint.

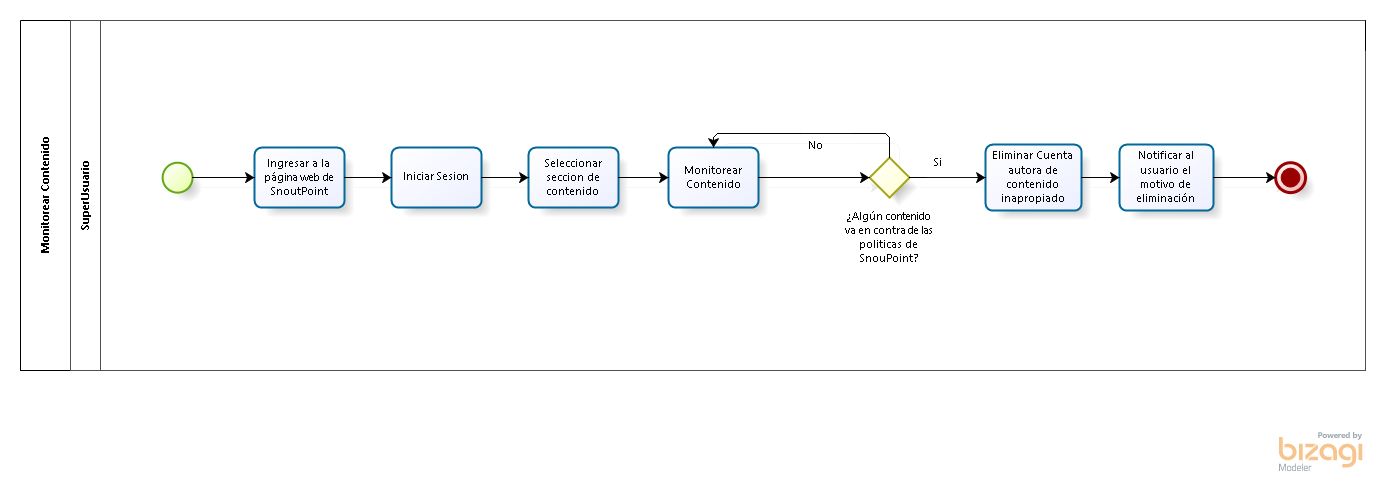


Ilustración . Proceso Monitorear Contenido (SuperUsuario)

En el anterior diagrama se detalla el proceso de monitorear contenido en un SuperUsuario, el cual permite que éste esté pendiente del contenido que se comparte en la red social evitando que se publiquen contenidos que van en contra de los principios de SnoutPoint.

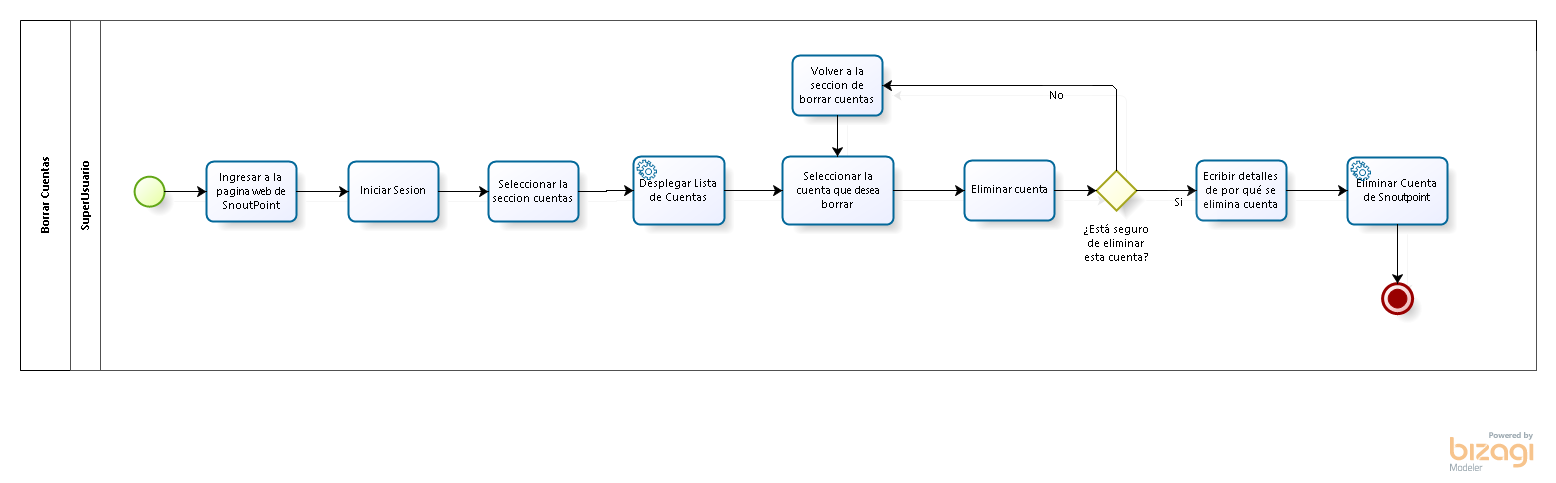


Ilustración . Proceso de Borrar Cuentas (SuperUsuario)

En el anterior diagrama se detalla el proceso de borrar cuentas en un SuperUsuario, el cual permite que con la lista que se le despliega de todas las cuentas elija cual cuenta desea borrar y por qué se le eliminó la cuenta.

## Persistencia

Para el sistema de SnoutPoint se hace necesario usar una base de datos que almacene toda la información definida en los modelos, además que el framework de Ruby on Rails requiere el uso de bases de datos al estar basado en Modelo-Vista-Controlador, requiriendo de tablas para almacenar los registros y datos que posteriormente serían utilizados por el controlador y desplegados en la vista correspondiente.

Es por esto que se requiere de una representación de Modelo Relacional para que se represente formalmente la relación entre entidades dentro del sistema, incluyendo las llaves de identificación primarias y foráneas en cada registro.

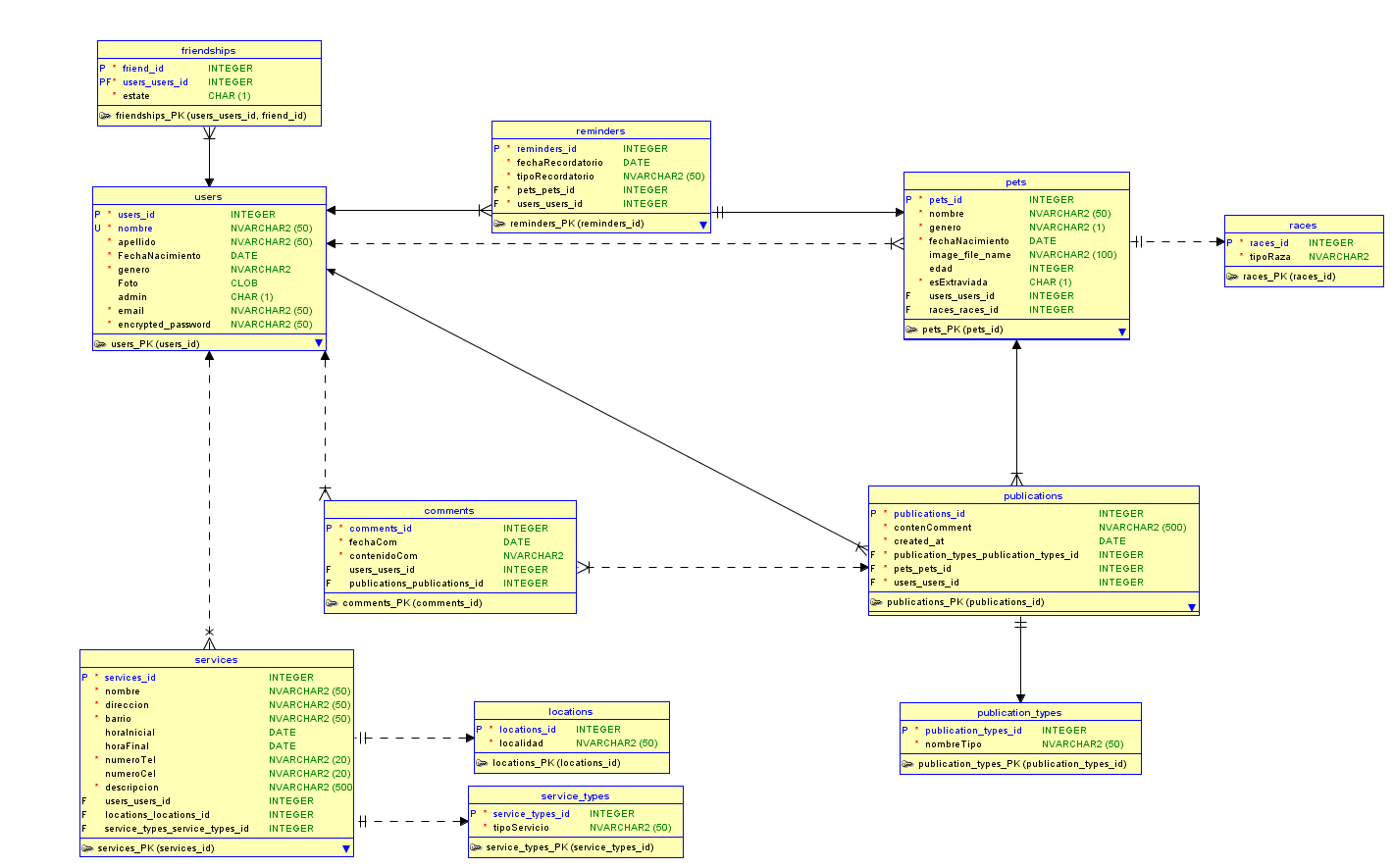


Ilustración . Modelo Relacional de Tablas

Como se puede ver en la ilustración del modelo relacional, las tablas se encuentran todas con nombres en inglés para guardar cierta consistencia con el framework de Rails al ser completamente desarrollado en inglés, además de resultar más sencillo y corto generar nombres plurales en ingles que en español.

|  |  |
| --- | --- |
| Entidad | Id |
| Friendships | C1 |
| Users | C2 |
| Reminders | C3 |
| Comments | C4 |
| Services | C5 |
| Locations | C6 |
| Service\_typ | C7 |
| Publications | C8 |
| Publications\_types | C9 |
| Pets | C10 |
| Races | C11 |

Tabla . Nomenclatura para Entidades

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | C1 | | **Entidad** | Friendships | |
| **Descripción** | Son los usuarios que se relacionan con otros | | | | |
| **Atributos** | | | | | |
| **Nombre** | | **Descripción** | | | **Tipo de Dato** |
| **Friend\_id** | | Llave primaria de la entidad, que se refiere al id único de cada amigo. | | | INTEGER |
| **Users\_users\_id** | | Es una llave foránea de la tabla de usuario, que identifica a cada usuario | | | INTEGER |
| **Estate** | | Es el estado en que se encuentra la solicitud de amistad. | | | CHAR |
| **Relaciones** | | | | | |
| **Nombre** | | **Descripción** | | | **Tipo de relación** |
| **Users** | | Son los usuarios que están en SnoutPoint | | | Esta entidad pertenece a la entidad amigos. Un usuario puede tener muchos amigos. |
| **Objetivo** | Interactuar con los demás usuarios | | | | |

Tabla . Descripción de la entidad Friendships

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | C2 | | **Entidad** | Users | |
| **Descripción** | El usuario es el encargado de interactuar en SnoutPonit | | | | |
| **Atributos** | | | | | |
| **Nombre** | | **Descripción** | | | **Tipo de Dato** |
| **Users\_id** | | Es la llave primaria de la entidad, que identifica al usuario como único | | | INTEGER |
| **Nombre** | | Es el nombre de cada usuario | | | NVARCHAR2 |
| **Apellido** | | Es el apellido del cada usuario | | | NVARCHAR2 |
| **FechaNacimiento** | | Es la fecha de nacimiento del usuario | | | Date |
| **Genero** | | Es el género del usuario | | | NVARCHAR2 |
| **Foto** | | Es la foto que desee poner el usuario para ser mostrada | | | CLOB |
| **Admin** | | Es un booleano que muestra si el usuario es administrador o no | | | BOOL |
| **Email** | | Es el email del usuario | | | NVARCHAR2 |
| **Encrypted­\_passworrd** | | Es la contraseña del usuario | | | NVARCHAR2 |
| **Relaciones** | | | | | |
| **Nombre** | | **Descripción** | | | **Tipo de relación** |
| **Reminders** | | Son los recordatorios asociados a las mascotas, que les notificaran a los usuarios las fechas importantes que este desee que le recuerden. | | | Un usuario puede recibir muchos recordatorios acerca de sus mascotas. |
| **Services** | | Son los servicios que el usuario desea ofrecer en SnoutPoint. | | | Un usuario puede tener muchos servicios. |
| **Comments** | | Son los comentarios que hace el usuario acerca de una publicación. | | | Un usuario puede generar muchos cometario. |
| **Publications** | | Son las publicaciones que el usuario desee hacer ya sea de sus mascotas o servicios asociados. | | | Un usuario puede realizar muchas publicaciones. |
| **Pets** | | Son las mascotas que desee tener el usuario en su cuenta. | | | Un usuario puede tener muchas mascotas. |
| **Objetivo** | Interactuar con el fin de dar a conocer las mascotas o servicios que desee y así obtener beneficios de esto. | | | | |

Tabla . Descripción de la entidad Users

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | C3 | | **Entidad** | Reminders | |
| **Descripción** | Son los eventos importantes de las mascotas | | | | |
| **Atributos** | | | | | |
| **Nombre** | | **Descripción** | | | **Tipo de Dato** |
| **Reminders\_id** | | Es la llave primaria que contiene el id de cada recordatorio. | | | INTEGER |
| **FechaRecordatorio** | | Fecha del recordatorio. | | | DATE |
| **TipoRecordatorio** | | Es el tipo de recordatorio que se va a notificar. | | | NVARCHAR2 |
| **Users\_users\_id** | | Es una llave foránea que contiene el id del usuario al cual se le será enviado el recordatorio. | | | INTEGER |
| **Pets\_pets\_id** | | Es la llave foránea que contiene el id de la mascota de la cual pertenece el recordatorio. | | | INTEGER |
| **Relaciones** | | | | | |
| **Nombre** | | **Descripción** | | | **Tipo de relación** |
| **Pets** | | Mascota de la cual se hará el recordatorio | | | Una mascota puede recibir recordatorios. |
| **Users** | | Usuario al cual se le enviara el recordatorio | | | Un usuario puede recibir recordatorios. |
| **Objetivo** | Notificar al usuario de los eventos que anteriormente éste solicito que le recordaran. | | | | |

Tabla . Descripción de la entidad Reminders

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | C4 | | **Entidad** | Comments | |
| **Descripción** | Son los comentarios que se desean hacer en las publicaciones. | | | | |
| **Atributos** | | | | | |
| **Nombre** | | **Descripción** | | | **Tipo de Dato** |
| **Comments\_id** | | Es la llave primaria que indica el id único de cada comentario. | | | INTEGER |
| **FechaCom** | | Es la fecha en la que se realiza el comentario. | | | DATE |
| **ContenidoCom** | | Es el contenido del comentario. | | | NVARCHAR |
| **Users\_users\_id** | | Es una llave foránea que indica el id del usuario que realiza el cometario. | | | INTEGER |
| **Publications\_publications\_id** | | Es una llave foránea que indica el id de la publicación a la cual se le asigna el cometario. | | | INTEGER |
| **Relaciones** | | | | | |
| **Nombre** | | **Descripción** | | | **Tipo de relación** |
| **Users** | | Es el encargado de hacer el comentario | | | Un usuario puede generar varias publicaciones. |
| **Publications** | | Es donde se pone el comentario | | | Una publicación puede tener muchos cometarios. |
| **Objetivo** | Interactuar con las publicaciones de las mascotas. | | | | |

Tabla . Descripción de la entidad Comments

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | C5 | | **Entidad** | Services | |
| **Descripción** | Son los servicios que un usuario desea ofrecer en SnoutPoint | | | | |
| **Atributos** | | | | | |
| **Nombre** | | **Descripción** | | | **Tipo de Dato** |
| **Services\_id** | | Es la llave primaria del servicio que representa el id. | | | INTEGER |
| **Nombre** | | Nombre del servicio que se desea prestar | | | NVARCHAR2 |
| **Dirección** | | Dirección en donde se encuentra ubicado dl servicio | | | NVARCHAR2 |
| **Barrio** | | Barrio en donde se encuentra ubicado el servicio | | | NVARCHAR2 |
| **HoraInicial** | | Es la hora en la cual se abre el lugar del servicio | | | DATE |
| **HoraFinal** | | Es la hora en la cual se cierra el lugar del servicio | | | DATE |
| **NumeroTel** | | Número de teléfono del servicio | | | NVARCHAR2 |
| **NumeroCel** | | Numero de celular del servicio | | | NVARCHAR2 |
| **Descripcion** | | Es la descripción del servicio | | | NVARCHAR2 |
| **Users\_users\_id** | | Es la llave foránea que indica el id del usuario al cual pertenece la publicación. | | | INTEGER |
| **Locations\_locations\_id** | | Es la llave foránea que indica el id de la localización de los servicios. | | | INTEGER |
| **Service\_types\_service\_types\_id** | | Es la llave foránea que indica el id del tipo de servicio. | | | INTEGER |
| **Relaciones** | | | | | |
| **Nombre** | | **Descripción** | | | **Tipo de relación** |
| **Users** | | Es el usuario dueño del servicio. | | | El servicio le pertenece al usuario. Un usuario puede tener varios servicios. |
| **Locations** | | Es la localidad en la cual se encuentra el servicio. | | | Un servicio puede tener una localidad. |
| **Services\_types** | | Es el tipo de servicio que se puede ofrecer. | | | Un servicio puede tener un tipo. |
| **Objetivo** | Ofrecer distintos servicios para las mascotas, ya sea veterinarias, fundaciones, tiendas. | | | | |

Tabla . Descripción de la entidad ServicesTabla

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | C6 | | **Entidad** | Locations | |
| **Descripción** | Localidad donde se encuentra el servicio | | | | |
| **Atributos** | | | | | |
| **Nombre** | | **Descripción** | | | **Tipo de Dato** |
| **Locations\_id** | | El la llave primaria que indica el id de la localidad | | | INTEGER |
| **Localidad** | | Es el nombre de la localidad | | | NVARCHAR2 |
| **Relaciones** | | | | | |
| **Nombre** | | **Descripción** | | | **Tipo de relación** |
| **Services** | | Es el servicio del cual se indicará la localidad | | | Un servicio tiene una localidad. |
| **Objetivo** | Dar a conocer la lista de localidades que existen en la cuidad de Bogotá para que el usuario escoja en cual se encuentra su servicio ofrecido. | | | | |

Tabla . Descripción de la entidad Locations

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | C7 | | **Entidad** | Service\_types | |
| **Descripción** | Los diferentes tipos de servicio que hay en SnoutPoint | | | | |
| **Atributos** | | | | | |
| **Nombre** | | **Descripción** | | | **Tipo de Dato** |
| **Service\_types\_id** | | Es la llave primaria que indica el id de los tipos de servicio | | | INTEGER |
| **tipoServicio** | | Es el tipo de servicio | | | NVARCHAR2 |
| **Relaciones** | | | | | |
| **Nombre** | | **Descripción** | | | **Tipo de relación** |
| **Services** | | Es el servicio del cual se indicará la localidad | | | Un servicio tiene una localidad |
| **Objetivo** | Tener una lista de los servicios que se pueden ofrecer en SnoutPoint para que no existan inconsistencias. | | | | |

Tabla . Descripción de la entidad Service\_Types

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | C8 | | **Entidad** | Publications | |
| **Descripción** | Fotos o acontecimientos importantes que desee publicar una persona sobre su mascota. | | | | |
| **Atributos** | | | | | |
| **Nombre** | | **Descripción** | | | **Tipo de Dato** |
| **Publications\_id** | | Es la llave primaria que indica el id de la publicación | | | INTEGER |
| **ContentComment** | | Es el contenido de la publicación | | | NVARCHAR2 |
| **Created\_at** | | Es la fecha en que se creó la publicación | | | DATE |
| **Publication\_types\_publications\_types\_id** | | Es la llave foránea que indica el id del tipo de publicación | | | INTEGER |
| **Pets\_pets\_id** | | Es la llave foránea que indica el id de la mascota | | | INTEGER |
| **Users\_users\_id** | | Es la llave foránea que indica el id del usuario | | | INTEGER |
| **Relaciones** | | | | | |
| **Nombre** | | **Descripción** | | | **Tipo de relación** |
| **Publication\_types** | | Son los tipos de publicaciones que existen. | | | Una publicación tiene un tipo de publicación |
| **Pets** | | Son las mascotas que desee tener el usuario en su cuenta. | | | Una mascota puede tener muchas publicaciones. |
| **Comments** | | Son los comentarios que hace el usuario acerca de una publicación. | | | Una publicación puede tener muchos cometarios. |
| **users** | | Es el usuario dueño del servicio. | | | Un usuario puede generar varias publicaciones. |
| **Objetivo** | Dar a conocer información adicional sobre cada mascota de la persona. | | | | |

Tabla . Descripción de la entidad Publications

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | C9 | | **Entidad** | Publications\_types | |
| **Descripción** | Los diferentes tipos de publicación que se pueden hacer | | | | |
| **Atributos** | | | | | |
| **Nombre** | | **Descripción** | | | **Tipo de Dato** |
| **Publication\_types\_id** | | Es la llave primaria del id del tipo de publicación | | | INTEGER |
| **Nombre Tipo** | | Es el tipo de la publicación. | | | NVARCHAR2 |
| **Relaciones** | | | | | |
| **Nombre** | | **Descripción** | | | **Tipo de relación** |
| **publications** | | Es la información adicional sobre cada mascota de la persona. | | | Una publicación tiene un tipo de publicación que lo caracteriza. |
| **Objetivo** | Tener una lista de los tipos de publicaciones que se pueden generar en SnoutPoint | | | | |

Tabla . Descripción de la entidad Publications\_types

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | C10 | | **Entidad** | pets | |
| **Descripción** | Es la mascota que maneja el usuario | | | | |
| **Atributos** | | | | | |
| **Nombre** | | **Descripción** | | | **Tipo de Dato** |
| **Pets\_id** | | Es la llave primaria de la mascota | | | INTEGER |
| **Nombre** | | Es el nombre de la mascota | | | NVARCHAR2 |
| **Genero** | | Es el género de la mascota | | | NVARCHAR2 |
| **fechaNacimiento** | | Es la fecha de nacimiento de la mascota | | | DATE |
| **Image\_file\_name** | | Es la imagen que representa a la mascota en la red social | | | NVARCHAR2 |
| **edad** | | Es la edad de la mascota | | | INTEGER |
| **esExtraviada** | | Es la opción para saber si esta extraviada la mascota o no | | | CHAR |
| **Users\_users\_id** | | Es la llave foránea que indica el id del usuario | | | INTEGER |
| **Races\_races\_id** | | Es la llave foránea que indica el id de la raza | | | INTEGER |
| **Relaciones** | | | | | |
| **Nombre** | | **Descripción** | | | **Tipo de relación** |
| **Reminders** | | Es el recordatorio que notifica sobre eventos de la mascota. | | | Una mascota está asociada a un recordatorio |
| **Publications** | | Son las publicaciones que se realizan acerca de las mascotas. | | | Una mascota está asociada a una publicación. |
| **Users** | | Es el usuario que tiene a las mascotas. | | | 0 o muchas cotas están asociadas a un usuario. |
| **Races** | | Es la raza que tiene cada mascota | | | Una mascota tiene una raza |
| **Objetivo** | Es la mascota asociada a la cuenta de usuario | | | | |

Tabla . Descripción de la entidad Pets

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | C11 | | **Entidad** | races | |
| **Descripción** | Raza de la mascota | | | | |
| **Atributos** | | | | | |
| **Nombre** | | **Descripción** | | | **Tipo de Dato** |
| **Races\_id** | | Es la llave primaria de las razas | | | INTEGER |
| **tipoRaza** | | Es el nombre del tipo de la raza especificado | | | NVARCHAR2 |
| **Relaciones** | | | | | |
| **Nombre** | | **Descripción** | | | **Tipo de relación** |
| **pets** | | Es la mascota que posee el usuario | | | Una mascota tiene una raza asociada. |
| **Objetivo** | Dar consistencia en cuanto a la raza de cada tipo de mascota. | | | | |

Tabla . Descripción de la entidad Races

## Interfaz de Usuario

Propósito: Que el desarrollador entienda todos los detalles de la interfaz de usuario. Contenido: Diagramas que expliquen las principales pantallas de la interfaz gráfica y la navegación entre ellas. Por ejemplo, diagramas de navegación, diagramas de estado, Diagramas de Flujo de Interfaz de Usuario [6]. Todos los diagramas deben ir acompañados de texto o tablas que expliquen sus componentes

Nota: El concepto de interfaz de usuario es más general que el de interfaz gráfica, por lo cual en una situación real podría ser necesario describir aspectos adicionales al de la interfaz gráfica. Para efectos de esta entrega sólo se le pide especificar esto último.

# 8. Referencias

Indique aquí todas las referencias bibliográficas utilizadas en el documento. Utilice formato IEEE o APA para definirlas. Para administrar automáticamente las referencias, se recomienda el uso de la herramienta Zotero (www.zotero.org).

----

Referencias usadas en el formato

[1] ISO/IEC/IEEE 42010

[2] IEEE Std 1016-2009

[3] RUP. Software Architecture Document <http://www.ts.mah.se/RUP/RationalUnifiedProcess/webtmpl/templates/a_and_d/rup_sad.htm>

[4] Philippe Kruchten, Architectural Blueprints—The “4+1” View Model of Software Architecture. <http://www.cs.ubc.ca/~gregor/teaching/papers/4+1view-architecture.pdf>

[5] Scott Ambler, UML Data Modeling Profile. <http://www.agiledata.org/essays/umlDataModelingProfile.html>

[6] Scott Ambler, User Interface Flow Diagrms. <http://www.agilemodeling.com/artifacts/uiFlowDiagram.htm>