

PROTOTIPO SOFTWARE QUE FACILITE LA REALIZACIÓN DE LOS  
PROCESOS LLEVADOS A CABO EN LOS CONJUNTOS RESIDENCIALES

CARLOS ANDRÉS BALAGUERA PEÑA  
JORGE LUIS RODRÍGUEZ VARGAS  
JESÚS MIGUEL SEPÚLVEDA SUÁREZ

UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS  
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA  
BUCARAMANGA

2019

PROTOTIPO SOFTWARE QUE FACILITE LA REALIZACIÓN DE LOS  
PROCESOS LLEVADOS A CABO EN LOS CONJUNTOS RESIDENCIALES

CARLOS ANDRÉS BALAGUERA PEÑA  
JORGE LUIS RODRÍGUEZ VARGAS  
JESÚS MIGUEL SEPÚLVEDA SUÁREZ

Trabajo de grado para optar por el título de Ingeniero de Sistemas

DIRECTOR  
FERNANDO ANTONIO ROJAS MORALES  
Magister en Ciencias Computacionales

UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS  
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA  
BUCARAMANGA

2019

## DEDICATORIA

El autor Carlos Andrés Balaguera Peña dedica este logro a:

*A mis padres Carlos Eduardo Balaguera Dávila y Clara Inés Peña Bautista, por su amor, trabajo y sacrificio durante estos años, gracias a ustedes he logrado llegar hasta aquí y convertirme en lo que soy.*

*A mi hermana Natalia Andrea Balaguera Peña que siempre estuvo acompañándome en esta etapa y dándome apoyo moral.*

*A mis compañeros de trabajo Jesús Miguel Sepúlveda Suárez y Jorge Luis Rodríguez Vargas.*

*A toda mi familia y mis amigos que me apoyaron durante el desarrollo del proyecto con sus consejos y palabras alentadoras.*

## DEDICATORIA

El autor Jorge Luis Rodríguez Vargas dedica este logro a:

*A mis padres Belkyeliza Vargas Triana y Juan De Los Reyes Rodríguez Rojas quienes me educaron con buen ejemplo y dedicación, y siempre me brindaron su apoyo incondicional en cada uno de los aspectos de mi vida.*

*A mi hermana Yessica Alejandra Rodríguez Vargas a mi sobrino Diego Alejandro López Rodríguez quien siempre fue mi ejemplo a seguir*

*A todos mis familiares quienes estuvieron en mi proceso de crecimiento tanto personal como profesional y estuvieron atentos de mi cuidado.*

*A mis compañeros Jesús Miguel Sepúlveda Suárez y Carlos Andrés Balaguera Peña quienes a pesar de todos los inconvenientes que se presentaron en la realización de este trabajo siempre se mantuvieron positivos acerca del mismo.*

## DEDICATORIA

El autor Jesús Miguel Sepúlveda Suárez dedica este logro a:

*A mis hermanos Leydee Teresa Villamizar Suárez y Luis Eduardo Suárez López quienes siempre estuvieron con su apoyo incondicional en la realización de este proyecto, ellos hicieron posible este logro.*

*A mis compañeros Carlos Andrés Balaguera Peña y Jorge Luis Rodríguez Vargas quienes me estuvieron acompañando en esta travesía que, aunque en ocasiones estuvimos a punto de rendirnos siempre creyeron en la realización de este proyecto.*

## **AGRADECIMIENTOS**

A la Universidad Industrial de Santander por brindarnos las herramientas y los espacios para avanzar en nuestro desarrollo profesional, y a su alta calidad de educación.

Al profesor Fernando Antonio Rojas Morales, por acogernos como estudiantes de pregrado y ofrecernos su dirección en parte del proceso.

A Jenny Paola Forero, porque fue una guía fundamental en el desarrollo de este proyecto, siempre nos brindó su ayuda y estuvo disponible para cualquier duda o inquietud que tuviéramos, con su experiencia nos guió durante el camino y fue también una fuente de inspiración para nosotros.

A nuestros padres y familiares quienes nunca dudaron en nuestras capacidades, nos apoyaron y alentaron a seguir a pesar de las adversidades presentadas en el desarrollo.

## CONTENIDO

|  | Pág. |
|--|------|
| INTRODUCCIÓN .....   | 17   |
| 1. GENERALIDADES .....   | 20   |
| 1.1 PLANTEAMIENTO Y JUSTIFICACIÓN DEL PROBLEMA.....                          | 20   |
| 1.2 OBJETIVOS.....   | 22   |
| 1.2.1 Objetivo general.....  | 22   |
| 1.2.2 Objetivos específicos.....   | 22   |
| 2. MARCO TEÓRICO Y METODOLÓGICO .....  | 24   |
| 2.1 MARCO TEÓRICO .....  | 24   |
| 2.1.1 Conceptos.....   | 24   |
| 2.1.2 Ley 675 de 2001.....   | 24   |
| 2.1.2.1 Régimen de Propiedad Horizontal.....                                 | 24   |
| 2.1.2.2 Reglamento de Propiedad Horizontal.....                              | 24   |
| 2.1.3 Estándar IEEE830.....  | 25   |
| 2.1.4 Modelo vista controlador.....  | 25   |
| 2.1.5 Modelado en procesos BPMN.....   | 26   |
| 2.1.6 Lenguaje unificado de modelado UML.....                                | 26   |
| 2.1.7 Herramientas software utilizadas para el desarrollo del prototipo..... | 27   |
| 2.1.8 Estado del arte.....   | 28   |
| 2.1.8.1 VIVOOK:.....   | 28   |
| 2.1.8.2 Miscondominios.com.....  | 29   |
| 2.2 MARCO METODOLÓGICO.....  | 30   |
| 2.2.1 Prototipado Evolutivo.....   | 30   |
| 2.2.2 Etapas del prototipado evolutivo.....                                  | 32   |
| 2.2.2.1 Planeación y análisis.....   | 32   |

|  |    |
|--|----|
| 2.2.2.2 Diseño.....  | 32 |
| 2.2.2.3 Construcción del prototipo. ....   | 33 |
| 2.2.2.4 Evaluación del prototipo.....  | 33 |
| 2.2.2.5 Refinamiento del prototipo. ....   | 33 |
| <br>   |    |
| 3. RESULTADOS DEL PROYECTO.....  | 34 |
| 3.1 PLANEACIÓN Y ANÁLISIS .....  | 34 |
| 3.2 DISEÑO .....   | 42 |
| 3.2.1 Casos de uso.....  | 43 |
| 3.2.2 Diagramas de caso de uso. ....   | 48 |
| 3.2.3 Diagrama de clases. ....   | 51 |
| 3.2.4 Modelo de Base de Datos.....   | 53 |
| 3.2.5 Diagrama de procesos.....  | 56 |
| 3.2.6 Diagrama de estados.....   | 58 |
| 3.2.4 Diagrama de secuencia. ....  | 60 |
| 3.3 ITERACIONES PARA LA CREACIÓN, EVALUACIÓN Y REFINAMIENTOS<br>DEL PROTOTIPO..... | 61 |
| 3.3.1 Primera Iteración.....   | 63 |
| 3.3.1.1 Construcción del Prototipo. ....   | 63 |
| 3.3.1.2 Evaluación del Prototipo y Refinamiento del Prototipo .....                | 64 |
| 3.3.2 Segunda Iteración.....   | 66 |
| 3.3.2.1 Construcción del Prototipo.....  | 66 |
| 3.3.2.2 Evaluación del Prototipo y Refinamiento del Prototipo .....                | 67 |
| 3.3.3 Tercera Iteración.....   | 69 |
| 3.3.3.1 Construcción del Prototipo.....  | 69 |
| 3.3.3.2 Evaluación del Prototipo y Refinamiento del Prototipo .....                | 69 |
| 3.3.4 Cuarta Iteración .....   | 71 |
| 3.3.4.1 Construcción del Prototipo.....  | 71 |
| 3.3.4.2 Evaluación del Prototipo y Refinamiento del Prototipo. ....                | 71 |

4. CONCLUSIONES .....74

5. RECOMENDACIONES.....75

BIBLIOGRAFIA.....76

## LISTA DE TABLAS

|  | Pág. |
|--|------|
| Tabla 1. Objetivos .....   | 22   |
| Tabla 2. Herramientas utilizadas en el desarrollo .....                      | 27   |
| Tabla 3. Actores.....  | 35   |
| Tabla 4. Procesos .....  | 36   |
| Tabla 5. Requisitos .....  | 38   |
| Tabla 6. Procesos asociados a requisitos.....                                | 40   |
| Tabla 7. Casos de uso .....  | 41   |
| Tabla 8. Caso de uso de solicitud de creación de cuenta.....                 | 43   |
| Tabla 9. Caso de uso Consultar solicitud de creación de cuenta.....          | 44   |
| Tabla 10. Caso de uso aceptar solicitud de creación de cuenta.....           | 44   |
| Tabla 11. Caso de uso creación de un inmueble .....                          | 45   |
| Tabla 12. Caso de uso agregar nueva zona social al conjunto.....             | 46   |
| Tabla 13. Caso de uso registrar información de integrante del conjunto ..... | 46   |
| Tabla 14. Caso de uso registrar nuevo visitante.....                         | 47   |
| Tabla 15. Requisitos funcionales primera iteración.....                      | 61   |
| Tabla 16. Requisitos funcionales segunda iteración .....                     | 61   |
| Tabla 17. Requisitos funcionales tercera iteración.....                      | 62   |
| Tabla 18. Requisito funcionales cuarta iteración.....                        | 63   |
| Tabla 19. Evaluación casos de uso primera iteración.....                     | 65   |
| Tabla 20. Evaluación casos de uso segunda iteración .....                    | 67   |
| Tabla 21. Evaluación casos de uso tercera iteración.....                     | 70   |
| Tabla 22. Evaluación casos de uso cuarta iteración .....                     | 72   |

## LISTA DE ILUSTRACIONES

|   | Pág. |
|---|------|
| Ilustración 1. Esquema de metodología del proyecto .....        | 31   |
| Ilustración 2. Relación usuarios, roles y personas.....         | 36   |
| Ilustración 3. Gestionar integrantes del conjunto .....         | 49   |
| Ilustración 4. Gestionar conjuntos.....                         | 49   |
| Ilustración 5. Gestionar reservas .....                         | 49   |
| Ilustración 6. Gestionar zonas sociales.....                    | 50   |
| Ilustración 7. Solicitud de creación de conjunto .....          | 51   |
| Ilustración 8. Diagrama de clases.....                          | 52   |
| Ilustración 9. Diagrama de clases parte 1 .....                 | 52   |
| Ilustración 10. Diagrama de clases parte 2.....                 | 53   |
| Ilustración 11. Modelo de Base de datos .....                   | 54   |
| Ilustración 12. Modelo de Base de datos parte 1 .....           | 55   |
| Ilustración 13. Modelo de Base de datos parte 2.....            | 56   |
| Ilustración 14. Proceso de acceso visitantes .....              | 57   |
| Ilustración 15. Proceso de reserva de una zona social.....      | 57   |
| Ilustración 16. Proceso de PQRS .....                           | 58   |
| Ilustración 17. Diagrama de estados PQRS .....                  | 59   |
| Ilustración 18. Diagrama de estados solicitud de conjunto ..... | 59   |
| Ilustración 19. Diagrama para estados de inmueble.....          | 60   |
| Ilustración 20. Diagrama de secuencia PQRS.....                 | 60   |

## LISTA DE ANEXOS

“Ver Anexos Adjuntos en el CD y pueden visualizarse en la Base de Datos UIS –  
Biblioteca”

Anexo A. Documento requisitos según norma IEEE 830

Anexo B. Casos de uso

Anexo C. Diagramas de casos de uso

Anexo D. Diagrama de clases

Anexo E. Diagramas de procesos

Anexo F. Diseño de interfaces

Anexo G. Entrevista

## RESUMEN

**TÍTULO:** PROTOTIPO SOFTWARE QUE FACILITE LA REALIZACIÓN DE LOS PROCESOS LLEVADOS A CABO EN LOS CONJUNTOS RESIDENCIALES.\*

**AUTORES:** CARLOS ANDRÉS BALAGUERA PEÑA, JORGE LUIS RODRÍGUEZ VARGAS, JESÚS MIGUEL SEPÚLVEDA SUÁREZ\*\*

**PALABRAS CLAVE:** PROTOTIPO, PROPIEDAD HORIZONTAL, ROL, PROCESOS

### DESCRIPCIÓN

Los conjuntos residenciales hoy en día son uno de los lugares preferidos para vivir por las comunidades urbanas, debido a que brindan un mayor nivel de seguridad, confianza y tienen a disposición de sus residentes diferentes servicios para su esparcimiento y comodidad, como reserva y préstamo de zonas sociales, piscina, gimnasios, parques para niños, etc.

Muchas de estas propiedades horizontales no cuentan con sistemas integrados para el manejo de información, por lo que la administración de los recursos de la unidad se limita a algunos procesos básicos, principalmente contables, y de una forma convencional, además no se tienen en cuenta los demás actores. Es por esto que se llevó a cabo el desarrollo de un prototipo software que facilita los procesos llevados a cabo en los conjuntos residenciales con el fin de brindar un soporte tecnológico en este ámbito, teniendo información más segura y confiable.

Este proyecto tiene como resultado la realización de la documentación en base a la norma IEEE 830 y un prototipo final que facilita los procesos llevados a cabo en los conjuntos residenciales identificados en el transcurso del mismo, con la inserción de este software para la operación de procesos en los conjuntos residenciales permitirá una mejor administración más ágil y eficiente de los recursos de la unidad residencial, dado que en su desarrollo e implementación se identificaron, estandarizaron y automatizaron estos para que se puedan adaptar a las diferentes necesidades de los diferentes tipos de conjuntos.

---

\* Trabajo de Grado

\*\* Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería de Sistemas e Informática. Director: Fernando Antonio Rojas Morales, Magister en Ciencias Computacionales.

## ABSTRACT

**TITLE:** SOFTWARE PROTOTYPE THAT FACILITATES THE REALIZATION OF THE PROCESSES CARRIED OUT IN THE RESIDENTIAL SETS.\*

**AUTHORS:** CARLOS ANDRÉS BALAGUERA PEÑA, JORGE LUIS RODRÍGUEZ VARGAS, JESÚS MIGUEL SEPÚLVEDA SUÁREZ\*\*

**KEY WORDS:** PROTOTYPE, HORIZONTAL PROPERTY, ROL, PROCESSES

### **DESCRIPTION:**

Nowadays residential sets are one of the preferred places to live in urban communities, since they provide a higher level of security and trust also, they offer to its residents different services for recreation and comfort, such as booking and lending social areas, swimming pool, gyms, parks for children, etc.

Many of these horizontal properties do not have integrated systems for information management, so their administration is limited to some basic processes, mainly accounting what is more they are made in a conventional way, they also don't consider the other actors. That is why a software development was necessary to facilitate the processes in the residential sets to provide technological support in this area, having safer and more reliable information.

This project has as a result the realization of the documentation based on the IEEE 830 and an final prototype that facilitates the realization of the processes carried out in the residential sets identified during this project, whit the insertion of this software for the operation of processes in the residential sets will allow an agile and better administration of the resources of the residential set, given that during the development and implementation this processes were identified, standardized and automated for the adaptation to the needs of different types of residential sets.

---

\* Bachelor Thesis.

\*\* Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería de Sistemas e Informática. Director: Fernando Antonio Rojas Morales, Magister en Ciencias Computacionales.

## INTRODUCCIÓN

Según Laudon & Laudon un sistema de información es definido desde una perspectiva técnica como un conjunto de componentes interrelacionados que recolectan, procesan, almacenan y distribuyen información proveniente del entorno de la organización para apoyar los procesos de tomas de decisiones y de control de esta. Los sistemas de información se encargan de recolectar los datos en bruto y relacionarlos de tal forma que se conviertan en información útil a través de tres actividades básicas: entrada, procesamiento y salida<sup>1</sup>.

Teniendo en cuenta lo descrito anteriormente es de gran importancia para las organizaciones tener un control preciso de la información que manejan, para ellos es necesario contar con un sistema de información que permita el óptimo manejo de esta. Con la llegada de la era digital muchas tareas, que antes se realizaban de manera manual o presencial, se optimizaron para que se realicen de forma digital, lo que facilita no solo a las organizaciones el control de estas, sino que también ayuda a los usuarios para que las realicen de forma rápida, permitiéndoles brindar un mejor servicio, además la información que se genera de la realización de estas tareas con un sistema digital es más fiable.

Actualmente, en las organizaciones es de gran importancia tener un control preciso de la información, para ello es recomendable contar con un sistema de información que permita el óptimo manejo de esta. Con la llegada de la era digital muchas tareas, que antes se realizaban de manera manual o presencial, se optimizaron para que se realicen de forma digital, lo que facilita no solo a las organizaciones el control de estas tareas, sino que también ayuda a los usuarios para que las realicen de forma rápida, permitiéndoles prestar un mejor servicio, además la información que se

---

<sup>1</sup> LAUDON, Kenneth y LAUDON, Jane. Sistemas de información gerencial (Decimosegunda ed.). [En línea]. México: Pearson (ROMERO, Alfonso trad.), 2012, p. 47. (Consultado el 12 de noviembre del 2019). Disponible en: <https://juanantonioleonlopez.files.wordpress.com/2017/08/sistemas-de-informacion-gerencial-12va-edicion-kenneth-c-laudon.pdf>

genera de la realización de estas tareas con un sistema digital es más fiable y es mucho más fácil tener un control de esta.

Los conjuntos residenciales se han constituido como uno de los sistemas de vivienda preferidos por ciudadanos. La idea de conjunto residencial hace referencia a un grupo de inmuebles cuyos bienes de dominio particular se encuentran destinados a la vivienda de personas, en donde se identifica un esquema de administración de recursos que corresponden a acuerdos comunes definidos por los habitantes y regidos por la normatividad urbanística vigente. Generalmente están ubicados en una calle, una cuadra o una manzana cerrada. En la perspectiva de Laudon & Laudon, los conjuntos residenciales no son propiamente organizaciones empresariales, pues no tienen como fin último la producción de un bien o servicio para la consecución de ganancias económicas. Sin embargo, debido a que en los conjuntos residenciales sí es necesario administrar información sobre recursos, usuarios, procesos, es posible concebirlos como organizaciones y representarlos como sistemas de información<sup>2</sup>.

Tradicionalmente en los conjuntos residenciales solo se usan herramientas software para la administración de algunos procesos como el registro contable, por ejemplo, pero quedan exentos los procesos que pueden administrarse mediante sistemas de información. Según las entrevistas realizadas a los administradores de los conjuntos residenciales visitados, algunas de las tecnologías usadas para llevar un manejo de la información son principalmente tablas en Excel y diferentes tipos de archivos que no se conectan de ninguna manera. Dicho esto, es indispensable la creación de una herramienta tecnológica que soporte y facilite la gestión de la información que se obtiene de los procesos identificados que son llevados a cabo en los conjuntos residenciales.

---

<sup>2</sup> COLOMBIA. CONGRESO DE LA REPÚBLICA. Ley 675 (3, agosto, 2001). por medio de la cual se expide el régimen de propiedad horizontal. [En línea]. Diario Oficial, Bogotá, 2001, No. 44.509, capítulo 1, artículo 1. (Consultado el 12 de noviembre del 2019). Disponible en: [http://www.sic.gov.co/sites/default/files/normatividad/Ley\\_675\\_2001.pdf](http://www.sic.gov.co/sites/default/files/normatividad/Ley_675_2001.pdf)

En el anexo G se puede encontrar el formato de entrevista realizado a los administradores de los conjuntos residenciales

En respuesta a esta necesidad se desarrolla el prototipo software “HOMIE” que pretende agilizar y automatizar los procesos identificados que se desarrollan en los conjuntos residenciales, lo que permitirá en ellos un óptimo manejo de la información. El prototipo tiene como fin crear un canal de comunicación entre las personas que están involucradas en el conjunto residencial, dando un mejor soporte a estas, así mismo pretende facilitar el trabajo del administrador al poder manejar la información del conjunto desde un mismo lugar.

## 1. GENERALIDADES

### 1.1 PLANTEAMIENTO Y JUSTIFICACIÓN DEL PROBLEMA

El crecimiento poblacional en el área metropolitana de Bucaramanga ha ido en aumento, con base en las cifras ofrecidas por el DANE desde el 2005 al 2015 se tuvo un incremento en de 11.453 personas aproximadamente lo que representa un 2.21%. Dicha situación ha generado la necesidad de aumentar la capacidad de vivienda en espacios reducidos, por lo tanto, se ha venido masificando la construcción de conjuntos residenciales por parte de las diferentes constructoras en busca de una solución para este crecimiento demográfico<sup>3</sup>.

Por otro lado, la seguridad es un punto a tener en cuenta ya que ha tomado mucha importancia en los últimos tres, según la encuesta “Bucaramanga como vamos” el índice de percepción de inseguridad, el cual mide que tan inseguro se siente el ciudadano en su ciudad ha aumentado del 14% al 24%. Por otro lado, el reporte también permite evidenciar que las victimas de delito han aumentado en 1% el último año, haciendo que la población busque que sus viviendas sean lugares más seguros, como lo son los conjuntos residenciales.<sup>4</sup>

Además de esto, en los conjuntos residenciales se debe cumplir internamente con una serie de normas y reglas, como lo son el manual de convivencia y el reglamento interno del conjunto las cuales velan por una convivencia sana entre los residentes, por la prevención de riesgos en zonas comunes, el cumplimiento de leyes en contratación de personal y servicios, entre otras cosas. Adicionalmente los

---

<sup>3</sup> DANE. Población por edad y sexo. [En línea]. Bogotá: Gobierno de Colombia, 2015. (Consultado el 12 de noviembre del 2019). Disponible en: <https://geoportal.dane.gov.co/midaneapp/pob.html>

<sup>4</sup> BUCARAMANGA METROPOLITANA CÓMO VAMOS. Encuesta de percepción ciudadana 2018. [En línea]. Bucaramanga: Universidad de Santander, et al., 2018. (Consultado el 12 de noviembre del 2019). Disponible en: [https://www.bucaramangacomovamos.org/post/descarga-la-9na-encuesta-de-percepci%C3%B3n-ciudadana?fbclid=IwAR3BJaXoFLnKHFEhunZDE7Tvb\\_Zm\\_kf3\\_EsCUHngoW7XrVi4EtiR0pKOoHLQ](https://www.bucaramangacomovamos.org/post/descarga-la-9na-encuesta-de-percepci%C3%B3n-ciudadana?fbclid=IwAR3BJaXoFLnKHFEhunZDE7Tvb_Zm_kf3_EsCUHngoW7XrVi4EtiR0pKOoHLQ)

conjuntos residenciales también están regidos por la ley 675 del 2001 la cual tiene como objetivo “regular la forma especial de dominio, denominada propiedad horizontal, en la que concurren derechos de propiedad exclusiva sobre bienes privados y derechos de copropiedad sobre el terreno y los demás bienes comunes, con el fin de garantizar la seguridad y la convivencia pacífica en los inmuebles sometidos a ella, así como la función social de la propiedad”<sup>5</sup>.

Se realizó un sondeo en ocho conjuntos residenciales para obtener información de estos y parte o casi toda la información es manejada de manera poco efectiva y puede ser propensa a pérdidas, manejando libros escritos a mano o formatos sin ninguna conexión, lo que genera redundancia de datos y poca confiabilidad con la información manejada, demora o atraso en los procesos que allí se realizan y la posibilidad de no cumplir efectivamente con los reglamentos mencionados.

Este proyecto nace de la constatación de que en el mercado actual existen pocas herramientas TIC<sup>6</sup> como solución a las situaciones anteriormente descritas. Como lo son las herramientas mencionadas en el estado del arte, por tanto, se hace necesario el desarrollo de un prototipo software con la capacidad de facilitar los procesos que se identificaron como los más importantes que se realizan en dichas instalaciones, brindando un mejor uso de la información, más confiabilidad y seguridad.

Los beneficios que conlleva este proyecto se centran, principalmente, en brindar una mayor confiabilidad a la información manejada en los conjuntos residenciales, y al mismo tiempo pretende optimizar la mayoría de los procesos realizados. Debido a que la información será manejada de manera digital y almacenada en bases de datos seguras, se reducirá en gran parte el uso de papel y los costos de administración, trayendo así también un beneficio para el medio ambiente.

---

<sup>5</sup> COLOMBIA. CONGRESO DE LA REPÚBLICA. Ley 675, Óp. Cit.

<sup>6</sup> Tecnologías de la información y la comunicación

El prototipo software principalmente tendrá la capacidad de facilitar procesos tales como el ingreso de personas al conjunto, la reserva para bienes comunes de uso general, estado de deudas del inmueble, publicación de noticias, envío de PQRS a administración, registro de mascotas, publicación de informes contables por parte del administrador y publicación de actividades programadas de diferentes intereses.

## 1.2 OBJETIVOS

**1.2.1 Objetivo general.** Desarrollar un prototipo software que facilite los procesos de administración y de servicios en conjuntos residenciales

**1.2.2 Objetivos específicos.** A continuación, se enuncian los objetivos específicos planteados en la propuesta de este proyecto, en relación con una descripción de la manera como se ha dado cumplimiento a los mismos, así como el lugar donde puede encontrarse evidencia de este cumplimiento dentro de este informe.

Tabla 1. Objetivos

| No. | OBJETIVO  | CUMPLIMIENTO   | UBICACIÓN   |
|-----|---|--|---|
| 1   | Revisar el estado del arte y aplicaciones similares que faciliten los procesos de los conjuntos residenciales.  | Para dar cumplimiento a este objetivo se realizó una consulta de aplicaciones similares y se realizó un estudio de los módulos y facilidades que ofrece.   | Esto se puede encontrar en la sección <a href="#">2.1.2 Estado del arte</a>   |
| 2   | Consultar los procesos que se llevan a cabo en los conjuntos residenciales.   | Para dar cumplimiento a este objetivo se realizaron visitas a conjuntos residenciales donde el equipo de trabajo se reunió con el administrador del conjunto residencial para establecer los procesos y tareas que se llevan a cabo en los conjuntos residenciales | Esto se puede encontrar en la sección <a href="#">3.1 Planeación y Análisis.</a>  |
| 3   | Analizar los pasos que intervienen en los procesos que se van a tener en cuenta para el desarrollo del prototipo software en los conjuntos residenciales y diseñar un | Para dar cumplimiento a este objetivo se realizaron los diagramas de procesos haciendo uso de la herramienta Bizagi Modeler  | Esto se puede encontrar en la sección <a href="#">3.2.4 Diagramas de procesos.</a><br>La totalidad de los diagramas de procesos se encuentran en el |

| No. | OBJETIVO   | CUMPLIMIENTO   | UBICACIÓN  |
|-----|--|--|--|
|     | esquema de estos para entender su funcionamiento.  |  | <a href="#">anexo E.</a>   |
| 4   | Especificar los requisitos funcionales y no funcionales para el desarrollo del prototipo, siguiendo la norma IEEE 830. | Para dar cumplimiento a este objetivo se realizó el documento de especificación de requisitos basándonos en la norma IEEE 830            | Esto se puede encontrar en la sección <a href="#">2.1.3 Estándar IEE830</a> y con profundización en el <a href="#">anexo A</a> |
| 5   | Crear una versión del prototipo software que abarque el alcance proyecto.  | Para dar cumplimiento a este objetivo se realizó el desarrollo del prototipo haciendo uso de las herramientas mencionadas en la tabla 2. | Esto se puede encontrar en la sección <a href="#">3 Resultados del proyecto</a>  |

## 2. MARCO TEÓRICO Y METODOLÓGICO

### 2.1 MARCO TEÓRICO

**2.1.1 Conceptos.** El ámbito de los conjuntos residenciales responde al Régimen de propiedad horizontal, el cual, junto con el Reglamento de propiedad horizontal y la Ley 675 de 2001, ofrece tanto las definiciones como la reglamentación que opera a nivel nacional. A continuación, se describe brevemente cada una de estas reglamentaciones.

**2.1.2 Ley 675 de 2001.** Según la secretaria del senado “La presente ley regula la forma especial de dominio, denominada propiedad horizontal, en la que concurren derechos de propiedad exclusiva sobre bienes privados y derechos de copropiedad sobre el terreno y los demás bienes comunes, con el fin de garantizar la seguridad y la convivencia pacífica en los inmuebles sometidos a ella, así como la función social de la propiedad”.

A partir de estos reglamentos, a continuación, se describen los conceptos principales en el ámbito de los conjuntos residenciales que vienen al caso para los intereses de este proyecto.

**2.1.2.1 Régimen de Propiedad Horizontal.** Sistema jurídico que regula el sometimiento a propiedad horizontal de un edificio o conjunto, construido o por construirse.

**2.1.2.2 Reglamento de Propiedad Horizontal.** Estatuto que regula los derechos y obligaciones específicas de los copropietarios de un edificio o conjunto sometido al régimen de propiedad horizontal.

**2.1.3 Estándar IEEE830<sup>7</sup>.** Este estándar creado por la IEEE<sup>8</sup> es un conjunto de recomendaciones para realizar la especificación de requisitos de software, el cual tiene como producto final la documentación entre el cliente y el grupo de desarrollo para así cumplir con la totalidad de las exigencias estipuladas.

El análisis de requisitos es quizás una de las tareas más importantes en el ciclo de vida del desarrollo del software, debido a que en este se determinan los “planos” de la nueva aplicación que se va a crear, en donde se describe específicamente lo que el nuevo sistema debe hacer y cómo hacerlo. Por tanto, no se trata de una actividad solo de análisis, sino también de síntesis, que permite desarrollar junto con el cliente, requisitos con un alto nivel de detalle, dejando muy en claro para ambas partes cómo se llevará a cabo el desarrollo y que se espera obtener del mismo. Para el actual desarrollo se decidió, entonces, realizar la especificación de los requisitos basados en la norma IEEE830; esta formulación completa se puede consultar en el anexo A.

**2.1.4 Modelo vista controlador.** Para el desarrollo de este prototipo se determinó usar la arquitectura de software MVC<sup>9</sup> una de las arquitecturas más comunes y recomendables para el desarrollo del software, puesto que separa los componentes de la aplicación, dependiendo de la responsabilidad que tienen, lo que significa que cuando se hace un cambio en alguna parte del código, este no afecta el resto del mismo.

Para separar la lógica del negocio con los datos manejados por la aplicación, MVC propone 3 componentes principales que son el modelo, la vista y el controlador, dichos componentes serán descritos a la brevedad a continuación.

---

<sup>7</sup> IEEE. 830-1998 - IEEE Recommended Practice for Software Requirements Specifications. [En línea]. 2009. (Consultado el 12 de noviembre del 2019). Disponible en: <https://ieeexplore.ieee.org/document/720574>

<sup>8</sup> The Institute of Electrical and Electronic Engineers, Inc.

<sup>9</sup> Modelo Vista Controlador

**Modelo:** se encarga principalmente del manejo de los datos, generalmente es el encargado de realizar las consultas a la base de datos.

**Controlador:** tienen como tarea principal recibir las peticiones del usuario, solicitar la información al modelo, y devolver esta información a la vista si es necesario.

**Vista:** es la encargada de representar los datos suministrados por el controlador de una forma visual que pueda ser vista de una manera fácil por el usuario que está utilizando a la aplicación.

**2.1.5 Modelado en procesos BPMN.** BPMN<sup>10</sup> es un estándar aceptado internacionalmente como notación gráfica para describir procesos. BPMN permite, mediante un formato de flujo de trabajo, utilizar una notación estandarizada, para modelar un proceso con el mayor detalle posible para que sea entendible en cualquier contexto en el que se utilice.

En el desarrollo de este proyecto fue utilizada la versión BPMN 2.0, que corresponde a la última especificación de OMG<sup>11</sup>, en el modelamiento de procesos de negocio, haciendo uso de la herramienta BIZAGI MODELER, el documento completo correspondiente a este proyecto puede ser visualizado en el anexo E.

**2.1.6 Lenguaje unificado de modelado UML.** UML<sup>12</sup> puede entenderse como un lenguaje visual que consta de un conjunto de normas y gráficos en el mundo del desarrollo, que no sea comprensible solamente por el programador del software, sino también por los usuarios de negocios y quien quiera que desee entender el sistema.

---

<sup>10</sup> BUSINESS PROCESS MODEL AND NOTATION. Home. [En línea]. Object Management Group. (Consultado el 12 de noviembre del 2019). Disponible en: <http://www.bpmn.org/>

<sup>11</sup> Ibid.

<sup>12</sup> UNIFIED MODELING LANGUAGE. Introduction to UML. [En línea]. Object Management Group, 2005. (Consultado el 12 de noviembre del 2019). Disponible en: <http://www.uml.org/what-is-uml.htm>

Generalmente los diagramas UML permiten describir los límites, estructura y comportamiento del sistema y los objetos que contiene. Cabe resaltar que no es un lenguaje de programación, pero existen diferentes herramientas software que permiten generar código a través de los diagramas generados.

Los diagramas que describen el diseño del prototipo software desarrollado en este proyecto fueron realizados usando UML como lenguaje estándar, mediante el software STARUML. Entre los diagramas elaborados se encuentran: diagramas de casos de uso, de procesos, de estados y de clases, los cuales pueden ser visualizados en los anexos C, D y E.

**2.1.7 Herramientas software utilizadas para el desarrollo del prototipo.** En la siguiente tabla se encuentran enlistadas las diferentes herramientas que se usaron en el proyecto con su respectiva descripción

Tabla 2. Herramientas utilizadas en el desarrollo

| Herramienta             | Descripción   |
|-------------------------|---|
| <b>C#</b>               | C# es un lenguaje de programación orientado a objetos de Microsoft, diseñado para trabajar con la plataforma .NET. Facilita el intercambio de información y servicios en la WEB permitiendo crear una gran variedad de aplicaciones, además su sintaxis es robusta y de fácil entendimiento.  |
| <b>.NET</b>             | .NET es un entorno de desarrollo creado por Microsoft que está compuesto de herramientas, lenguajes de programación y librerías que permiten la construcción de diferentes tipos de aplicaciones de manera fácil, organizada y sencilla.  |
| <b>ASP.NET</b>          | ASP.NET es un framework open source creado por Microsoft para la construcción de aplicaciones web y servicios con .NET  |
| <b>Entity Framework</b> | Entity Framework es un conjunto de tecnologías en ADO.NET que soporta el desarrollo de aplicaciones software orientadas a datos. Entity framework permite a los desarrolladores trabajar con datos en forma de objetos y propiedades específicos de dominio, como las direcciones de cliente / cliente, sin tener que preocuparse por las tablas y columnas de la base de datos subyacentes donde se almacenan estos datos. |

| Herramienta           | Descripción   |
|-----------------------|---|
| <b>Visual Studio</b>  | Es un software integrado desarrollado por Microsoft, capaz de soportar múltiples lenguajes de programación como c++, c#, Visual Basic, Java, Python; así mismo soporta entornos de desarrollo web como ASP.net y Django entre otros. Facilita la creación de aplicaciones y servicios web en cualquier entorno que soporte la plataforma .NET |
| <b>LINQ</b>           | LINQ por sus siglas en inglés <b>Language Integrated Query</b> es un componente de .NET que le da capacidades de consulta de datos a los lenguajes usados en .NET   |
| <b>SQL Server</b>     | Es un sistema de manejo de datos del modelo relacional desarrollado por Microsoft   |
| <b>Bizagi Modeler</b> | Es una herramienta software de modelamiento de procesos de negocio que permite crear diagramas de procesos mediante el estándar BPMN  |
| <b>Balsamiq</b>       | Balsamiq es una herramienta que permite diseñar las interfaces de usuario de forma rápida y sencilla concentrándose solo en la estructura y contenido   |
| <b>StarUML</b>        | Es una herramienta de MKLab que permite realizar diagramas UML de una forma muy sencilla e intuitiva  |

**2.1.8 Estado del arte.** En la actualidad existen muy pocos desarrollos software orientados a la optimización de procesos en conjuntos residenciales, y los que se pueden encontrar en el mercado están dirigidos principalmente a las tareas realizadas por el administrador y las actividades que este realiza

A continuación, se mencionan las plataformas encontradas y algunas de sus características más relevantes.

2.1.8.1 VIVOOK<sup>13</sup>:. Esta plataforma web divide sus funcionalidades en dos módulos principales: el módulo de comunicación y el módulo de administración. El módulo de comunicación se orienta principalmente a los residentes y propietarios del conjunto en donde se encuentra herramientas para gestionar la noticias y avisos,

<sup>13</sup> VIVOOK. Inicio. [En línea]. (Consultado el 12 de noviembre del 2019). Disponible en: <https://www.vivook.com/>

un espacio exclusivo para objetos perdidos, calendario para visualizar los eventos disponibles en el momento, acceso a documentación relacionada al conjunto, encuestas y votaciones, directorio contactos e emergencias, blog y solicitudes al administrador. En el módulo administrativo se tiene la posibilidad de consultar movimientos y saldos de caja, envío e impresión de avisos de cobro, visualizar cuotas y deudores, ver historial de ingresos y pagos, estado de cuenta de cada vivienda, visualizar las tareas del administrador, y revisar el presupuesto del conjunto residencial.

Este software mexicano es una de las herramientas más completas encontradas puesto que abarca muchos de los procesos que se realizan en los conjuntos residenciales tanto para los propietarios como los administradores y cuenta con una versión de prueba de 30 días con el cien por ciento de todas sus funciones habilitadas.

2.1.8.2 Miscondominios.com<sup>14</sup>. Esta es una aplicación SaaS que te permite gestionar todas las tareas administrativas asociadas a un edificio, y compartir la información con los propietarios asociados de una manera ágil e interactiva.

Este software online cuenta con herramientas tanto para administradores como para propietarios en los conjunto residenciales; en cuanto a la parte administrativa se disponen de módulos para el control de gastos e ingresos, gestión de tesorería, control de morosos, gestión documental y registro de incidentes; para los propietarios se cuenta con herramientas que permiten acceder a gestión documental, ver situaciones de incidencias y la posibilidad de visualizar con total transparencia la situación financiera del conjunto.

---

<sup>14</sup> MISCONDOMINIOS. Página de inicio. [En línea]. Onyx. (Consultado el 12 de noviembre del 2019). Disponible en: <http://www.miscondominios.com/>

Cabe resaltar que esta plataforma cuenta también con una aplicación disponible para Android e IOS exclusiva para los propietarios, pero como la mayoría de herramientas descritas esta se centra principalmente en tareas administrativas limitando al residente y propietario excluyendo de la misma los procesos que estos llevan a cabo.

## **2.2 MARCO METODOLÓGICO**

Una metodología de desarrollo software hace referencia a los métodos, procesos y herramientas que permiten planear, controlar y monitorear el avance del proceso de desarrollo de software. Con una metodología establecida se tiene una visión clara de lo que se va a realizar, como se va a realizar y qué resultados se espera obtener.

Actualmente existe una variedad de metodologías de desarrollo, todas se caracterizan por distribuir el trabajo por etapas lo cual permite tener un mayor control en el desarrollo del aplicativo; no es posible decir que una es mejor que otra por sí misma, ya que cada proyecto se ajusta a una metodología de desarrollo diferente.

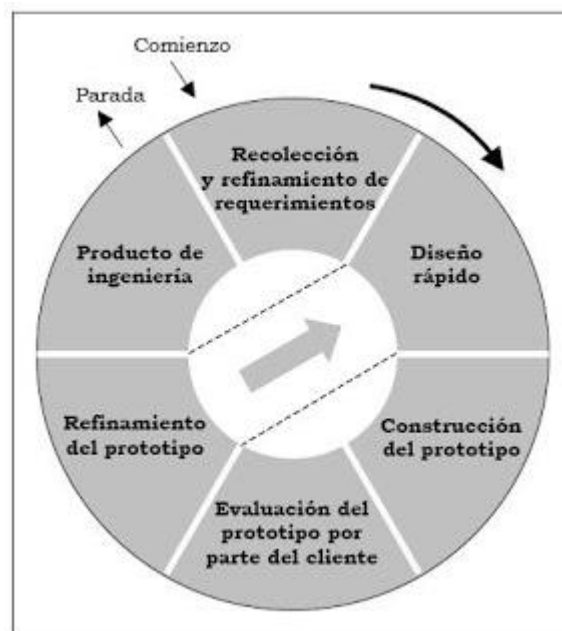
**2.2.1 Prototipado Evolutivo.** Para el desarrollo del proyecto se utilizó la metodología prototipado evolutivo en la cual se trabaja por fases o etapas, esto permite que se desarrolle el concepto del sistema a medida que avanza el proyecto y se tenga un mejor entendimiento de este de forma que pueda modificarse de manera inmediata respondiendo a la realimentación que se tiene por parte del cliente y del usuario final.

Esta metodología se caracteriza principalmente porque divide el desarrollo por etapas, generando así un prototipo el cual es sometido a varias iteraciones en donde se le realizan mejoras a dicho prototipo teniendo en cuenta las especificaciones del cliente y del equipo de desarrollo, de esta forma se genera en cada iteración un prototipo más estable al anterior permitiendo así tener una aplicación más robusta;

el número de iteraciones generadas no está especificado, se terminan las iteraciones del prototipo cuando tanto el cliente como el equipo de desarrollo estén conformes con el prototipo que se tiene en el momento.

En la Figura 1 se muestran las etapas de las que consta la metodología de prototipado evolutivo.

Ilustración 1. Esquema de metodología del proyecto<sup>15</sup>



### Ventajas

- Ideal para proyectos cuyos requisitos cambian con rapidez.
- Ideal para cuando el cliente no puede especificar el conjunto total de los requisitos.
- Ideal cuando no se logra identificar de forma apropiada el área de aplicación.
- Ideal cuando los desarrolladores no están seguros de la arquitectura o los algoritmos adecuados a utilizar.
- Reduce costos y aumenta la probabilidad de éxito en su consecución.

<sup>15</sup> PRESSMAN, Roger. Ingeniería del software: Un enfoque práctico. (Quinta ed.). Madrid, España: McGraw Hill, 2002. ISBN: 0-07-709677-0

## **Desventajas**

- Existe una imposibilidad de conocer al inicio del proyecto lo que se tardará en crear un producto aceptable.
- Esta aproximación puede convertirse fácilmente en una excusa para realizar el desarrollo con el modelo de codificar y corregir.

Debido a que los requisitos del proyecto estaban en constante cambio ya que no se tenía una visión global de la aplicación y algunos estaban mal definidos, no eran adecuados o hacían falta para la implementación, esto permitió valerse de esta metodología que facilita el cambio de requisitos en su desarrollo, debido a esto no se pudieron minimizar las desventajas de esta metodología ya que no se podían establecer tiempos fijos de entrega.

### **2.2.2 Etapas del prototipado evolutivo**

2.2.2.1 Planeación y análisis. En esta primera fase del desarrollo del proyecto se debe especificar los requisitos necesarios para el desarrollo del primer prototipo, por lo que se realizó un estudio a los procesos realizados en los conjuntos residenciales, además de comparar algunas otras herramientas existentes para automatizarlos.

2.2.2.2 Diseño. Con los requisitos especificados en lenguaje natural de la fase anterior se prosiguió a realizar un diseño inicial por medio del estándar de modelado UML, así como también se desarrolló un modelo de datos y una interfaz de usuario de fácil comprensión para su uso.

En esta fase se realizaron las siguientes actividades:

- Diagramas UML
- Diseño y modelado de la base de datos
- Diseño de las interfaces de usuario

2.2.2.3 Construcción del prototipo. En la fase de implementación se realizó la programación de los requisitos anteriormente mencionados. Se definen los módulos a desarrollar y la iteración a la que pertenece cada uno de ellos.

2.2.2.4 Evaluación del prototipo. Una vez realizado un prototipo estable en la fase anterior, se sometió a evaluación por parte del grupo de desarrollo, realizando las pruebas correspondientes establecidas por ellos mismos y documentándolas para que puedan usarse en la siguiente fase.

2.2.2.5 Refinamiento del prototipo. Con los resultados de la evaluación realizada en la fase anterior se prosigue a definir soluciones o mejoras al prototipo y se realiza su implementación inmediata para generar un prototipo nuevo que cubra las nuevas necesidades.

### **3. RESULTADOS DEL PROYECTO**

Siguiendo las etapas que tiene en cuenta la metodología seleccionada, se dio inicio con la planeación y análisis, seguida de la etapa de diseño y posteriormente se hizo la primera iteración construyendo el primer prototipo, evaluándolo y finalmente refinándolo, este proceso se realizó con 4 iteraciones para generar así un prototipo mejorado en cada una de ellas. A continuación, se muestra lo que se logró en cada una de estas etapas y los resultados que se obtuvieron.

#### **3.1 PLANEACIÓN Y ANÁLISIS**

Para el desarrollo de esta etapa se visitaron 8 conjuntos residenciales en el municipio de Floridablanca, y se tuvieron reuniones con sus administradores con el fin de identificar los procesos más relevantes o importantes con los que contaba cada uno de ellos que se identificaron por medio de una entrevista cuyo formato se encuentra en el anexo G.

Con base en estas declaraciones se prosiguió a estandarizar dichos procesos, teniendo en cuenta las tareas que tienen en común los conjuntos, y así generar el diagrama de estos incluyendo las tareas automáticas realizadas por el sistema y la participación de los usuarios. Los procesos se diseñaron haciendo uso del software Bizagi modeler, esto facilitó su entendimiento y permitió tener una visión más amplia de lo que cada uno de ellos implica. Los procesos diseñados para este proyecto pueden visualizarse en el anexo E.

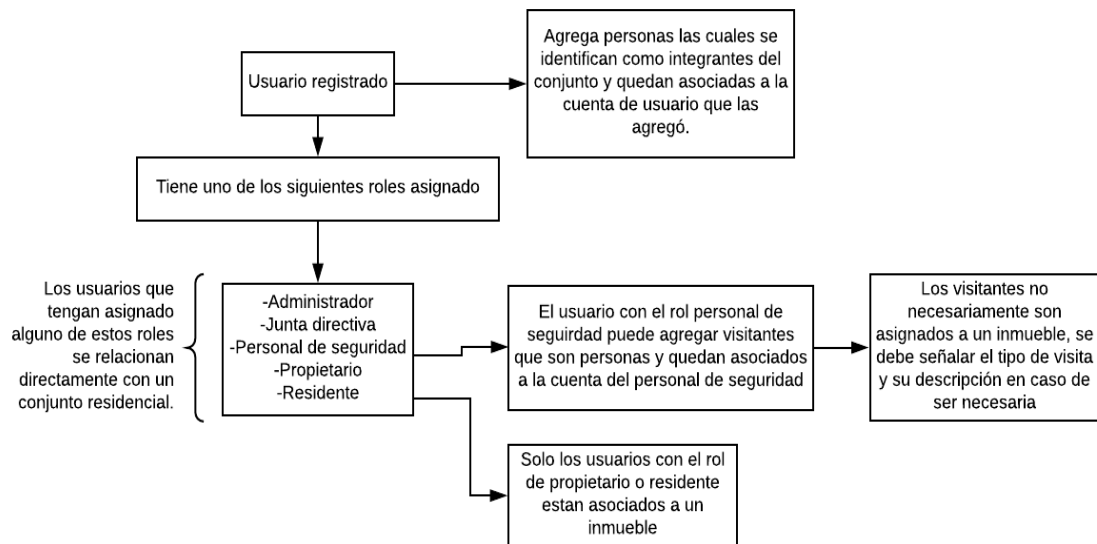
En la siguiente tabla se muestran los actores participantes del prototipo con su respectiva descripción.

Tabla 3. Actores

| <b>Actor</b>                 | <b>Descripción</b>   |
|------------------------------|--|
| <b>Superusuario</b>          | Es el usuario que tiene acceso a la plataforma y se encarga de hacer mantenimiento a esta.   |
| <b>Administrador</b>         | Encargado de administrar la propiedad horizontal, elegido por la junta directiva. Tiene acceso a la mayoría de las funcionalidades dentro de un conjunto. Se denomina el representante legal de la persona jurídica en cuestión. |
| <b>Residente</b>             | Persona titular o arrendatario del inmueble específico que reside en dichas instalaciones y tiene acceso a algunas funcionalidades.  |
| <b>Personal de seguridad</b> | Vigilante o encargado de velar por la seguridad de los residentes, así como el acceso de visitantes. Tiene acceso a algunas funcionalidades.   |
| <b>Junta directiva</b>       | Grupo de personas encargadas de la toma de decisiones relevantes en la propiedad horizontal. Tienen una cuenta que les permite el acceso a ciertas funcionalidades.  |
| <b>Propietario</b>           | Persona titular de un inmueble específico que no reside en el conjunto residencial, tiene acceso a algunas funcionalidades.  |
| <b>Visitante</b>             | Persona que no reside en el conjunto residencial e ingresa a este ya sea a un inmueble, a administración, a hacer reparaciones entre otras. No tienen cuenta de usuario.   |

En la siguiente figura se evidencia la relación de los usuarios con los diferentes roles participantes en el sistema, así mismo se muestra brevemente como están relacionadas las personas (integrantes del conjunto) con los usuarios. Cabe resalta que el usuario con el rol de superusuario no está contemplado en el diagrama.

Ilustración 2. Relación usuarios, roles y personas.



La siguiente tabla muestra los procesos tenidos en cuenta para el desarrollo del prototipo software.

Tabla 4. Procesos

| ID  | Nombre             | Actores asociados  | Descripción  |
|-----|--------------------|--|--|
| N01 | Gestionar conjunto | Administrador, Superusuario                                      | El sistema permite a los actores el control del conjunto residencial; en el caso del superusuario la parte de la creación, solicitudes de creación, deshabilitar, habilitar, etc. En cambio, al administrador solo le permite la visualización de su conjunto. |
| N02 | Gestionar inmueble | Administrador, Superusuario                                      | El sistema permite a los actores realizar todas las tareas involucradas en los inmuebles ya sea crearlos, editarlos, habilitarlos, deshabilitarlos y visualizarlos.  |
| N03 | Gestionar personas | Administrador, Personal de seguridad, Junta directiva, Residente | El sistema permite a los actores dependiendo del rol que tengan agregar personas, vehículos o mascotas a sus respectivas cuentas de usuario.   |
| N04 | Gestionar usuarios | Administrador, Superusuario                                      | El sistema permite a los actores, dependiendo del rol, la creación, habilitación y   |

| ID         | Nombre                       | Actores asociados  | Descripción   |
|------------|------------------------------|--|---|
|            |                              |  | deshabilitación de las cuentas.   |
| <b>N05</b> | Gestionar roles              | Superusuario   | El sistema permite al superusuario crear, editar y quitar los roles de las cuentas que limitan lo que el usuario puede hacer dependiendo de su rol.                                 |
| <b>N07</b> | Gestionar PQRS               | Administrador, Personal de seguridad, Junta directiva, Residente | El sistema permite a los actores dependiendo de su rol la gestión de las PQRS, que conlleva todo el proceso que se tiene que tener en cuenta al tener una PQRS.                     |
| <b>N08</b> | Gestionar informes contables | Administrador, Personal de seguridad, Junta directiva, Residente | El sistema permite a los actores dependiendo de su rol la gestión de los informes contables como lo son crear, editar, visualizar y archivar los informes.                          |
| <b>N09</b> | Gestionar reservas           | Administrador, Personal de seguridad, Junta directiva, Residente | El sistema permite a los actores dependiendo de su rol realizar una reserva de una zona social en una fecha específica.   |
| <b>N10</b> | Gestionar pagos o deudas     | Administrador, Personal de seguridad, Residente                  | El sistema permite al administrador asignar deudas a los residentes, así mismo permite cambiar el estado de las deudas, los residentes podrán visualizar las deudas de su inmueble. |
| <b>N11</b> | Gestionar calendario         | Administrador, Personal de seguridad, Junta directiva, Residente | El sistema permite al administrador publicar información en el calendario, asimismo a los demás actores les permite visualizar los eventos.   |
| <b>N12</b> | Gestionar visitas            | Administrador, Personal de seguridad, Residente                  | El sistema permite dependiendo del rol registrar una nueva visita, cambiar el estado de visitante frecuente y ver el historial de visitas   |
| <b>N13</b> | Gestionar cartelera          | Administrador, Personal de seguridad, Junta directiva, Residente | El sistema permite a los actores dependiendo de los permisos la publicación de noticias, así como la visualización de esta.   |
| <b>N14</b> | Gestionar foro               | Administrador, Personal de seguridad, Junta directiva, Residente | El sistema permite a los actores la gestión de un foro donde se puede discutir acerca de un tema en específico.   |
| <b>N15</b> | Gestionar zonas sociales     | Administrador, Personal de seguridad, Junta                      | El sistema permite dependiendo del rol crear, editar, cambiar el estado de una zona social (habilitada o inhabilitada), visualizar la   |

| ID         | Nombre           | Actores asociados    | Descripción  |
|------------|------------------|----------------------|--|
|            |                  | directiva, Residente | información de esta.   |
| <b>N16</b> | Gestionar diseño | Administrador        | Permite al administrador cambiar el color de fondo, agregar una imagen del conjunto. |

Con base en lo descrito anteriormente se inició el proceso de análisis de requisitos y su redacción con la norma IEEE 830 como guía, la cual estandariza esta tarea y brinda el nivel de detalle necesario para el desarrollo del presente trabajo.

En la siguiente tabla se pueden encontrar todos los requisitos funcionales que se identificaron en el proyecto, además estos se pueden encontrar en el anexo A: Documento de requisitos según norma IEEE830.

Tabla 5. Requisitos

| N    | Nombre del requisito  |
|------|---|
| RF01 | Solicitud de creación de un conjunto                                    |
| RF02 | Creación de un inmueble   |
| RF03 | Personalización de la visualización de la interfaz visual del prototipo |
| RF04 | Editar información de un integrante del conjunto                        |
| RF05 | Registrar información de nuevo integrante a la cuenta                   |
| RF06 | Inhabilitar integrante del conjunto                                     |
| RF07 | Consultar información del integrante del conjunto                       |
| RF08 | Realizar cambio de contraseña   |
| RF09 | Crear PQRS  |
| RF10 | Consultar PQRS  |
| RF11 | Responder PQRS  |
| RF12 | Cargar informes contables   |
| RF13 | Consultar informes contables  |
| RF14 | Descargar informe de contabilidad                                       |
| RF15 | Editar informes de contabilidad   |
| RF16 | Inhabilitar informes de contabilidad                                    |
| RF17 | Realizar reserva de la zona social                                      |
| RF18 | Consultar reserva   |
| RF19 | Generar deuda de reserva  |
| RF20 | Editar reserva  |
| RF21 | Cancelar reserva  |
| RF22 | Agregar evento al calendario  |
| RF23 | Consultar evento en el calendario                                       |
| RF24 | Modificar evento en el calendario                                       |
| RF25 | Archivar evento en el calendario  |
| RF26 | Registrar nuevo visitante   |

| N    | Nombre del requisito  |
|------|---|
| RF27 | Modificar información del visitante   |
| RF28 | Consultar información del visitante   |
| RF29 | Asignar visitante frecuente   |
| RF30 | Inhabilitar visitante frecuente   |
| RF31 | Registrar visita  |
| RF32 | Consultar historial de visitas  |
| RF33 | Agregar información a la cartelera  |
| RF34 | Consultar información de la cartelera   |
| RF35 | Editar información de la cartelera  |
| RF36 | Archivar información de la cartelera  |
| RF37 | Agregar nueva discusión del foro  |
| RF38 | Consultar discusiones del foro  |
| RF39 | Cerrar discusión del foro   |
| RF40 | Editar tema de discusión en el foro   |
| RF41 | Agregar comentario  |
| RF42 | Editar comentario   |
| RF43 | Archivar comentario   |
| RF44 | Asignar deuda de administración   |
| RF45 | Asignar deuda   |
| RF46 | Registrar pago de deuda   |
| RF47 | Consultar pago  |
| RF48 | Visualizar conjunto residencial   |
| RF49 | Habilitar conjunto residencial  |
| RF50 | Deshabilitar conjunto residencial   |
| RF51 | Aceptar o rechazar solicitud de creación de cuenta de un conjunto residencial |
| RF52 | Editar conjunto residencial   |
| RF53 | Concluir PQRS   |
| RF54 | Habilitar inmueble  |
| RF55 | Inhabilitar inmueble  |
| RF56 | Editar inmueble   |
| RF57 | Iniciar sesión  |
| RF58 | Agregar zona social   |
| RF59 | Editar zona social  |
| RF60 | Agregar rol   |
| RF61 | Editar rol  |
| RF62 | Quitar rol  |
| RF63 | Inhabilitar zona social   |
| RF64 | Habilitar zona social   |
| RF65 | Habilitar informes de contabilidad  |
| RF66 | Agregar nueva mascota   |
| RF67 | Quitar mascota  |
| RF68 | Editar mascota  |
| RF69 | Visualizar mascota  |
| RF70 | Agregar nuevo vehículo  |
| RF71 | Quitar vehículo   |
| RF72 | Editar vehículo   |
| RF73 | Visualizar vehículo   |

En la siguiente tabla se relacionan los requisitos funcionales que están asociados a los procesos de los conjuntos residenciales que tuvimos en cuenta para el desarrollo de este prototipo.

Tabla 6. Procesos asociados a requisitos

| ID  | Nombre del proceso                       | Requisitos               |
|-----|--|--------------------------|
| N01 | Gestionar conjunto                       | RF01, RF48 - RF52        |
| N02 | Gestionar inmueble                       | RF02, RF54 – RF56        |
| N03 | Gestionar personas, mascotas y vehículos | RF04 - RF07, RF66 – RF73 |
| N04 | Gestionar usuarios                       | RF08, RF57               |
| N05 | Gestionar roles                          | RF60 - RF62              |
| N07 | Gestionar PQRS                           | RF09 - RF11, RF53        |
| N08 | Gestionar informes contables             | RF12 - RF16, RF-65       |
| N09 | Gestionar reservas                       | RF17 - RF21              |
| N10 | Gestionar pagos o deudas                 | RF44 - RF47              |
| N11 | Gestionar calendario                     | RF22 - RF25              |
| N12 | Gestionar visitas                        | RF26 - RF32              |
| N13 | Gestionar cartelera                      | RF33 - RF 36             |
| N14 | Gestionar foro                           | RF36 - RF43              |
| N15 | Gestionar zonas sociales                 | RF58 – RF59, RF63 – RF64 |
| N16 | Gestionar diseño                         | RF03                     |

Teniendo esto como base se describen los casos de uso para el proyecto los cuales pueden evidenciarse en la siguiente tabla.

Tabla 7. Casos de uso

| ID    | Nombre del caso de uso  |
|-------|---|
| CU-01 | Iniciar sesión  |
| CU-02 | Cerrar sesión   |
| CU-03 | Solicitud de creación de cuenta                               |
| CU-04 | Consultar solicitud de creación de cuenta                     |
| CU-05 | Aceptar o rechazar solicitud de creación de cuenta            |
| CU-06 | Habilitar conjunto residencial                                |
| CU-07 | Deshabilitar conjunto residencial                             |
| CU-08 | Visualizar conjuntos residenciales                            |
| CU-09 | Editar información de un conjunto residencial                 |
| CU-10 | Creación de un inmueble                                       |
| CU-11 | Visualización de un inmueble                                  |
| CU-12 | Modificar información del inmueble                            |
| CU-13 | Habilitar inmueble  |
| CU-14 | Deshabilitar inmueble   |
| CU-15 | Agregar nueva zona social al conjunto                         |
| CU-16 | Visualizar una zona social                                    |
| CU-17 | Editar información de zona social del conjunto                |
| CU-18 | Deshabilitar zona social del conjunto                         |
| CU-19 | Habilitar zona social del conjunto                            |
| CU-20 | Personalización de la interfaz visual del prototipo           |
| CU-21 | Visualización de la información de un integrante del conjunto |
| CU-22 | Registrar información de integrante del conjunto              |
| CU-23 | Editar información de integrante del conjunto                 |
| CU-24 | Inhabilitar integrante del conjunto                           |
| CU-25 | Realizar cambio de contraseña                                 |
| CU-26 | Crear PQRS  |
| CU-27 | Consultar PQRS  |
| CU-28 | Responder PQRS  |
| CU-29 | Cargar informes de contabilidad                               |
| CU-30 | Consultar informes de contabilidad                            |
| CU-31 | Descargar informes de contabilidad                            |
| CU-32 | Inhabilitar informes de contabilidad                          |
| CU-33 | Editar informes de contabilidad                               |
| CU-34 | Realizar reserva de zona social                               |
| CU-35 | Consultar reservas  |
| CU-36 | Cancelar reserva  |
| CU-37 | Agregar evento al calendario                                  |
| CU-38 | Consultar evento en el calendario                             |
| CU-39 | Modificar evento en el calendario                             |
| CU-40 | Archivar evento en el calendario                              |
| CU-41 | Registrar nuevo visitante                                     |
| CU-42 | Consultar información del visitante                           |
| CU-43 | Modificar información del visitante                           |
| CU-44 | Asignar visitante frecuente                                   |
| CU-45 | Inhabilitar visitante frecuente                               |
| CU-46 | Registrar visita  |
| CU-47 | Consultar historial de visitas                                |
| CU-48 | Agregar información a la cartelera                            |
| CU-49 | Consultar información de la cartelera.                        |

| ID    | Nombre del caso de uso                |
|-------|---------------------------------------|
| CU-50 | Editar información de la cartelera.   |
| CU-51 | Archivar información de la cartelera. |
| CU-52 | Agregar nueva discusión al foro       |
| CU-53 | Consultar discusiones del foro        |
| CU-54 | Editar tema de discusión en el foro   |
| CU-55 | Cerrar discusión del foro             |
| CU-56 | Agregar comentario                    |
| CU-57 | Editar comentario                     |
| CU-58 | Archivar comentario                   |
| CU-59 | Asignar deuda                         |
| CU-60 | Visualizar historial de deudas        |
| CU-61 | Registrar pago deuda                  |
| CU-62 | Concluir PQRS                         |
| CU-63 | Crear un rol                          |
| CU-64 | Editar un rol                         |
| CU-65 | Quitar un rol                         |

### 3.2 DISEÑO

Esta etapa del desarrollo se enfocó en diseñar el sistema previamente, para seguido de esto, construir un primer prototipo de este, así que con los requisitos especificados en lenguaje natural y regidos por la norma IEEE830 se prosiguió a realizar los diagramas que más ayudarían en el proceso, utilizando el lenguaje estándar de modelado UML, con la herramienta StarUML. Se definieron entonces los casos de uso, los diagramas de casos de uso, los diagramas de procesos, los diagramas de clases y los diagramas de estado; todos los diagramas mencionados pueden verse al detalle en los anexos B, C, D y E.

Principalmente, con base en el diagrama de clases, se identificaron las tablas para ser tenidas en cuenta en la base de datos y a partir de esto diseñar su modelo con el diagrama entidad relación correspondiente que se puede observar en la sección 3.2.3 Modelo de base de datos.

Por otra parte, se realizó el diseño de la interfaz gráfica de usuario con la ayuda de la herramienta “Balsamiq” que permite de una forma muy sencilla crear una vista

específica de la aplicación. Para este proyecto se crearon las interfaces que se consideraron más relevantes y pueden encontrarse en el anexo F.

**3.2.1 Casos de uso.** Se define por caso de uso la narración o plantilla que describe una función o rasgo de un sistema desde el punto de vista de usuario, permitiendo así la creación de un modelo de requisitos más completo. A continuación, se mostrarán algunos de los casos de uso más relevantes del proyecto. Estos pueden visualizarse en su totalidad en el anexo B.

Tabla 8. Caso de uso de solicitud de creación de cuenta

|                          |  |   |
|--------------------------|--|---|
| <b>ID</b>                | CU-03  |   |
| <b>Nombre</b>            | Solicitud de creación de cuenta de conjunto residencial  |   |
| <b>Descripción</b>       | Permite enviar una solicitud con la respectiva información para crear una nueva cuenta de un conjunto residencial  |   |
| <b>Disparador</b>        | Se desea crear una nueva cuenta para el conjunto residencial   |   |
| <b>Actores</b>           | Administrador  |   |
| <b>Precondiciones</b>    | El conjunto residencial no está registrado en el sistema   |   |
| <b>Flujo Normal</b>      | <b>Usuario: Administrador</b><br>1. El administrador ingresa a la plataforma<br>3. El administrador hace clic en el botón para registrarse.<br>5. El administrador llena el formulario con la información requerida<br>7. El administrador hace clic en el botón para crear cuenta | <b>Sistema</b><br>2. El sistema muestra la página de inicio<br>4. El sistema muestra un formulario con los datos necesarios para enviar la solicitud<br>6. El sistema verifica que todos los campos estén diligenciados correctamente<br>8. El sistema pregunta si está seguro que la información es correcta |
| <b>Flujo Alternativo</b> | <b>Usuario: Administrador</b><br>6.2. El administrador hace clic en aceptar  | <b>Sistema</b><br>6.1. Si los datos no están diligenciados de forma correcta el sistema muestra un mensaje con la información correspondiente.<br>6.3. El sistema indica los campos con la información faltante o inválida.<br>6.4. Se dirige a 5.  |
| <b>Postcondiciones</b>   | Se envía la solicitud de creación de la cuenta y el sistema notifica al Superusuario de que tiene una nueva solicitud de creación  |   |

Tabla 9. Caso de uso Consultar solicitud de creación de cuenta

|                          |   |   |
|--------------------------|---|---|
| <b>ID</b>                | CU-04   |   |
| <b>Nombre</b>            | Consultar solicitud de creación de cuenta   |   |
| <b>Descripción</b>       | Permite consultar una solicitud de creación de cuenta de un conjunto residencial  |   |
| <b>Disparador</b>        | Se desea visualizar la solicitud de creación de cuenta de un conjunto residencial   |   |
| <b>Actores</b>           | Superusuario  |   |
| <b>Precondiciones</b>    | <ul style="list-style-type: none"> <li>• Tener una solicitud de creación de cuenta de un conjunto residencial</li> <li>• Haber iniciado sesión como Superusuario</li> </ul> |   |
| <b>Flujo Normal</b>      | <b>Usuario: Superusuario</b><br>1. El Superusuario accede a el módulo de solicitudes<br>3. El Superusuario hace clic en la solicitud deseada                                | <b>Sistema</b><br>2. El sistema muestra el módulo de solicitudes y una lista de estas<br>4. El sistema muestra la información detallada de la solicitud específica. |
| <b>Flujo Alternativo</b> |   |   |
| <b>Postcondiciones</b>   | Se muestra la información de la solicitud seleccionada  |   |

Tabla 10. Caso de uso aceptar solicitud de creación de cuenta

|                          |   |  |
|--------------------------|---|--|
| <b>ID</b>                | CU-05   |  |
| <b>Nombre</b>            | Aceptar solicitud de creación de cuenta   |  |
| <b>Descripción</b>       | Permite aceptar una solicitud de creación de cuenta de un conjunto residencial  |  |
| <b>Disparador</b>        | <ul style="list-style-type: none"> <li>• Se desea aceptar la solicitud de creación</li> </ul>   |  |
| <b>Actores</b>           | Superusuario  |  |
| <b>Precondiciones</b>    | <ul style="list-style-type: none"> <li>• Tener una solicitud de creación de cuenta de un conjunto residencial</li> <li>• Haber iniciado sesión como Superusuario</li> </ul>         |  |
| <b>Flujo Normal</b>      | <b>Usuario: SU</b><br>1. Ir a el caso de uso CU-04<br>2. El Superusuario hace clic en el botón para aceptar la solicitud<br>4. El Superusuario hace clic en el botón para confirmar | <b>Sistema</b><br>3. El sistema muestra un mensaje para la confirmación de dicha acción.<br>5. El sistema envía un correo electrónico al administrador indicando la novedad. |
| <b>Flujo Alternativo</b> |   |  |

|                        |  |
|------------------------|--|
| <b>Postcondiciones</b> | Se crea la cuenta del conjunto residencial y se envía un correo notificando la creación de la cuenta al administrador, Personal de seguridad y Junta directiva |
|------------------------|--|

Tabla 11. Caso de uso creación de un inmueble

|                          |  |  |
|--------------------------|--|--|
| <b>ID</b>                | CU-10  |  |
| <b>Nombre</b>            | Creación de un inmueble  |  |
| <b>Descripción</b>       | Se desea crear un nuevo inmueble asociado a la cuenta de un conjunto residencial   |  |
| <b>Disparador</b>        | Se desea registrar un inmueble que no se encuentra registrado  |  |
| <b>Actores</b>           | Administrador  |  |
| <b>Precondiciones</b>    | <ul style="list-style-type: none"> <li>• El inmueble no debe estar registrado previamente en el sistema.</li> <li>• Haber iniciado sesión como Administrador</li> </ul>  |  |
| <b>Flujo Normal</b>      | <b>Usuario: Administrador</b><br>1. El Administrador accede al módulo de virtualización<br>3. El Administrador da clic en el botón de agregar nuevo inmueble<br>5. El Administrador diligencia el formulario<br>6. El Administrador hace clic en el botón para aceptar<br>9. El Administrador hace clic en el botón para confirmar | <b>Sistema</b><br>2. El sistema muestra todos los inmuebles existentes en el sistema relacionados al conjunto residencial específico.<br>4. El sistema muestra el formulario correspondiente<br>7. El sistema verifica si los datos diligenciados son válidos<br>8. El sistema muestra un mensaje de confirmación y el botón correspondiente |
| <b>Flujo Alternativo</b> | <b>Usuario: Administrador</b><br>7.2. El Administrador hace clic en aceptar  | <b>Sistema</b><br>7.1. Si los datos no están diligenciados de forma correcta el sistema muestra un mensaje con la información correspondiente.<br>7.3. El sistema indica los campos con la información faltante o inválida.<br>7.4. Se dirige a 5.   |
| <b>Postcondiciones</b>   | Se virtualiza el nuevo inmueble a la cuenta del conjunto residencial y se guarda en la base de datos   |  |

Tabla 12. Caso de uso agregar nueva zona social al conjunto

|                          |  |   |
|--------------------------|--|---|
| <b>ID</b>                | CU-15  |   |
| <b>Nombre</b>            | Agregar nueva zona social al conjunto  |   |
| <b>Descripción</b>       | Se quiere agregar una nueva zona social al conjunto  |   |
| <b>Disparador</b>        | Se desea agregar una zona social que no está virtualizada  |   |
| <b>Actores</b>           | Administrador  |   |
| <b>Precondiciones</b>    | <ul style="list-style-type: none"> <li>• No debe estar virtualizada la zona social</li> <li>• Haber iniciado sesión como Administrador</li> </ul>  |   |
| <b>Flujo Normal</b>      | <b>Usuario: Administrador</b><br>1. El Administrador accede a el módulo correspondiente de las zonas sociales<br>3. El Administrador hace clic en agregar una nueva zona social<br>5. El Administrador llena los datos de la zona social<br>6. El Administrador hace clic en el botón para aceptar<br>9. El Administrador hace clic en el botón para la confirmación | <b>Sistema</b><br>2. El sistema muestra el módulo de las zonas sociales con una lista de las mismas<br>4. El sistema muestra el formulario de datos para agregar una nueva zona social<br>7. El sistema verifica que los datos ingresados sean válidos<br>8. El sistema muestra un mensaje de confirmación con el botón correspondiente para la aceptación o la cancelación de dicha acción.<br>10. El sistema realiza la actualización en base de datos. |
| <b>Flujo Alternativo</b> | <b>Actor: Administrador</b><br>7.2. El Administrador hace clic en aceptar<br>9.1. El Administrador hace clic en cancelar.  | <b>Sistema</b><br>7.1. Si los datos no están diligenciados de forma correcta El sistema muestra un mensaje con la información correspondiente.<br>7.3. El sistema indica los campos con la información faltante o inválida.<br>7.4. Se dirige a 5.<br>9.2. Se va a 5.   |
| <b>Postcondiciones</b>   | Se virtualiza la nueva zona social a la cuenta del conjunto residencial y se guarda en la base de datos  |   |

Tabla 13. Caso de uso registrar información de integrante del conjunto

|                    |   |
|--------------------|---|
| <b>ID</b>          | CU-22   |
| <b>Nombre</b>      | Registrar información de integrante del conjunto          |
| <b>Descripción</b> | Se quiere registrar la información de un nuevo integrante |
| <b>Disparador</b>  | No está registrada la información del integrante          |

|                          |  |   |
|--------------------------|--|---|
| <b>Actores</b>           | PS, Residente, Administrador, Junta Directiva  |   |
| <b>Precondiciones</b>    | <ul style="list-style-type: none"> <li>• Haber iniciado sesión como uno de los actores mencionados</li> <li>• No debe estar guardada la información del integrante en el sistema</li> </ul>  |   |
| <b>Flujo Normal</b>      | <b>Usuario: PS, Residente, Administrador, Junta Directiva</b><br>1. El usuario accede a el módulo correspondiente al directorio<br>3. El usuario hace clic en agregar nuevo integrante<br>5. El usuario llena los datos del integrante<br>6. El usuario hace clic en el botón para aceptar<br>9. El usuario hace clic en el botón para la confirmación | <b>Sistema</b><br>2. El sistema muestra el módulo de directorio con una lista de los integrantes existentes<br>4. El sistema muestra el formulario de datos para agregar un nuevo integrante<br>7. El sistema verifica que los datos ingresados sean válidos<br>8. El sistema muestra un mensaje de confirmación con el botón correspondiente para la aceptación o la cancelación de dicha acción. .<br>10. El sistema realiza la actualización en base de datos. |
| <b>Flujo Alternativo</b> | <b>Usuario: PS, Residente, Administrador, Junta Directiva</b><br>7.2. El usuario hace clic en aceptar<br>9.1. El usuario hace clic en cancelar.  | <b>Sistema</b><br>7.1. Si los datos no están diligenciados de forma correcta el sistema muestra un mensaje con la información correspondiente.<br>7.3. El sistema indica los campos con la información faltante o inválida.<br>7.4. Se dirige a 5.<br>9.2. Se va a 5.   |
| <b>Postcondiciones</b>   | Se registra la información del nuevo integrante a el conjunto y se guarda en la base de datos  |   |

Tabla 14. Caso de uso registrar nuevo visitante

|                       |   |   |
|-----------------------|---|---|
| <b>ID</b>             | CU-46   |   |
| <b>Nombre</b>         | Registrar nuevo visitante   |   |
| <b>Descripción</b>    | Permite registrar un visitante que no esté previamente registrado                               |   |
| <b>Disparador</b>     | No existen los datos del visitante en el sistema  |   |
| <b>Actores</b>        | PS  |   |
| <b>Precondiciones</b> | Haber iniciado sesión como PS   |   |
| <b>Flujo Normal</b>   | <b>Usuario: PS</b><br>1. El PS accede al módulo de visitantes<br>3. El PS hace clic en el botón | <b>Sistema</b><br>2. El sistema muestra el módulo de visitante con una lista de los existentes<br>4. El sistema muestra el formulario |

|                          |   |   |
|--------------------------|---|---|
|                          | para agregar nuevo visitante<br>5. El PS llena el formulario con los datos correspondientes<br>6. El PS hace clic en guardar<br>9. El PS hace clic en el botón para la confirmación | correspondiente<br>7. El sistema verifica que los campos diligenciados son válidos.<br>8. El sistema muestra un mensaje de confirmación con botones para la confirmación o cancelación de dicha acción.<br>10. El sistema actualiza la información en base de datos.      |
| <b>Flujo Alternativo</b> | <b>Usuario: PS</b><br>7.3. El PS hace clic en aceptar<br>9.1. El PS hace clic en cancelar   | <b>Sistema</b><br>7.1. Si los datos no están diligenciados de forma correcta el sistema muestra un mensaje con la información correspondiente.<br>7.2. El sistema indica los campos con la información faltante o inválida.<br>7.4. Se dirige a 5.<br>9.2. Se dirige a 5. |
| <b>Postcondiciones</b>   | Se registra el nuevo visitante guardando su información en base de datos.   |   |

**3.2.2 Diagramas de caso de uso.** Un diagrama de caso de uso permite tener una representación visual de las relaciones existentes entre los casos de uso y la participación de los usuarios que intervienen en ellos.

Se realizaron los siguientes diagramas de casos de uso, los cuales fueron graficados con la herramienta StarUML.

- Gestionar inmuebles.
- Gestionar integrantes del conjunto.
- Gestionar pagos.
- Registrar visitantes.
- Gestionar calendario.
- Gestionar cartelera.
- Gestionar conjuntos.
- Gestionar foro.
- Gestionar informes contables.
- Gestionar PQRS.
- Gestionar reservas.

- Gestionar zonas sociales.
- Iniciar sesión.
- Solicitud de creación.

En las siguientes figuras pueden observarse algunos de ellos. Pueden detallarse en su totalidad todos los diagramas de casos de uso en el anexo C.

Ilustración 3. Gestionar integrantes del conjunto

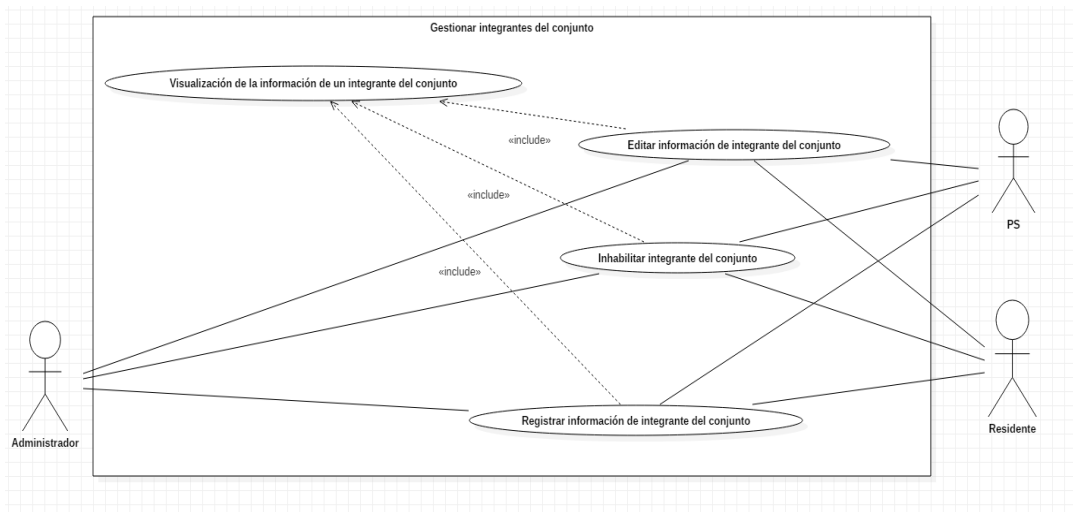


Ilustración 4. Gestionar conjuntos

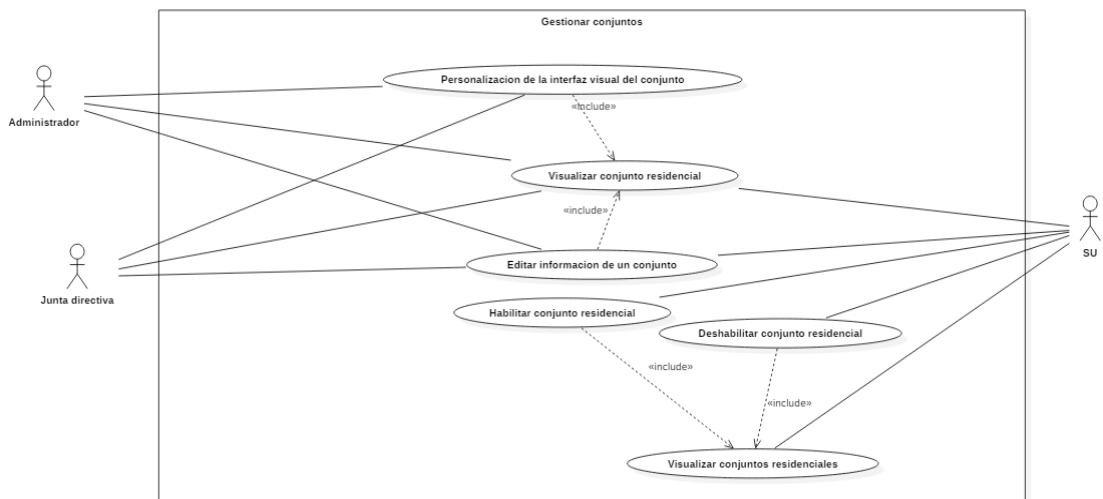


Ilustración 5. Gestionar reservas

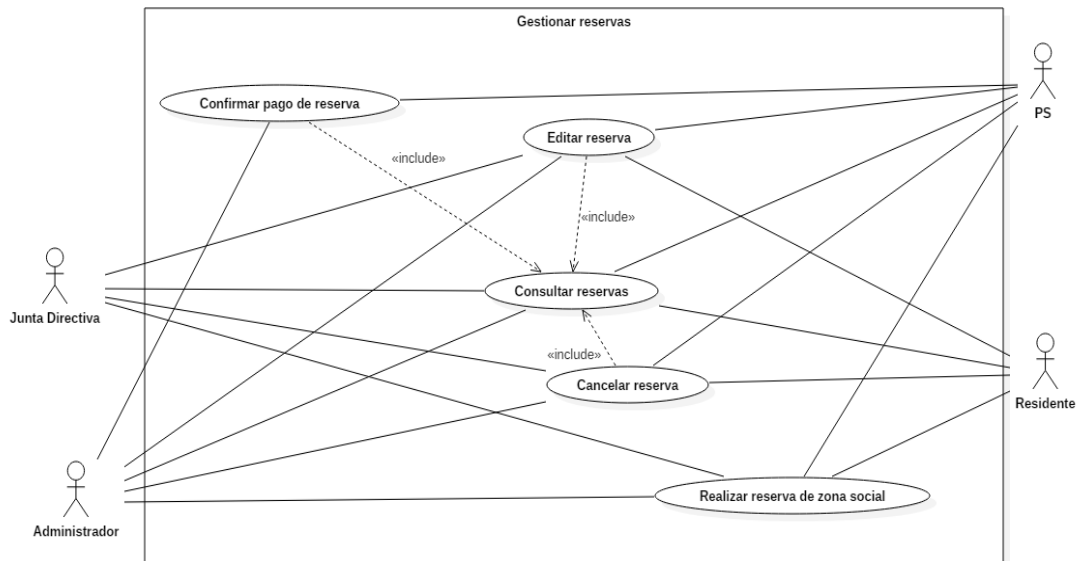


Ilustración 6. Gestionar zonas sociales

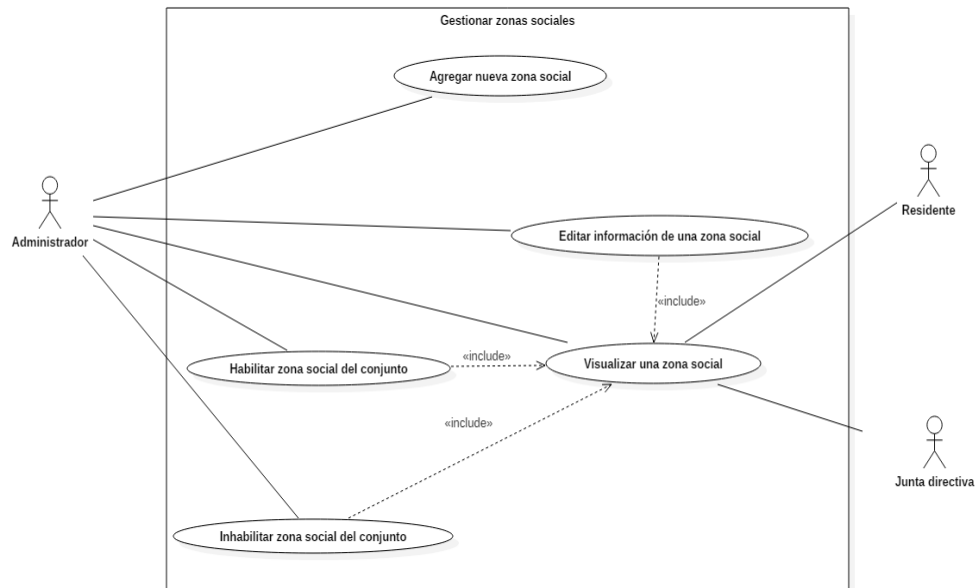
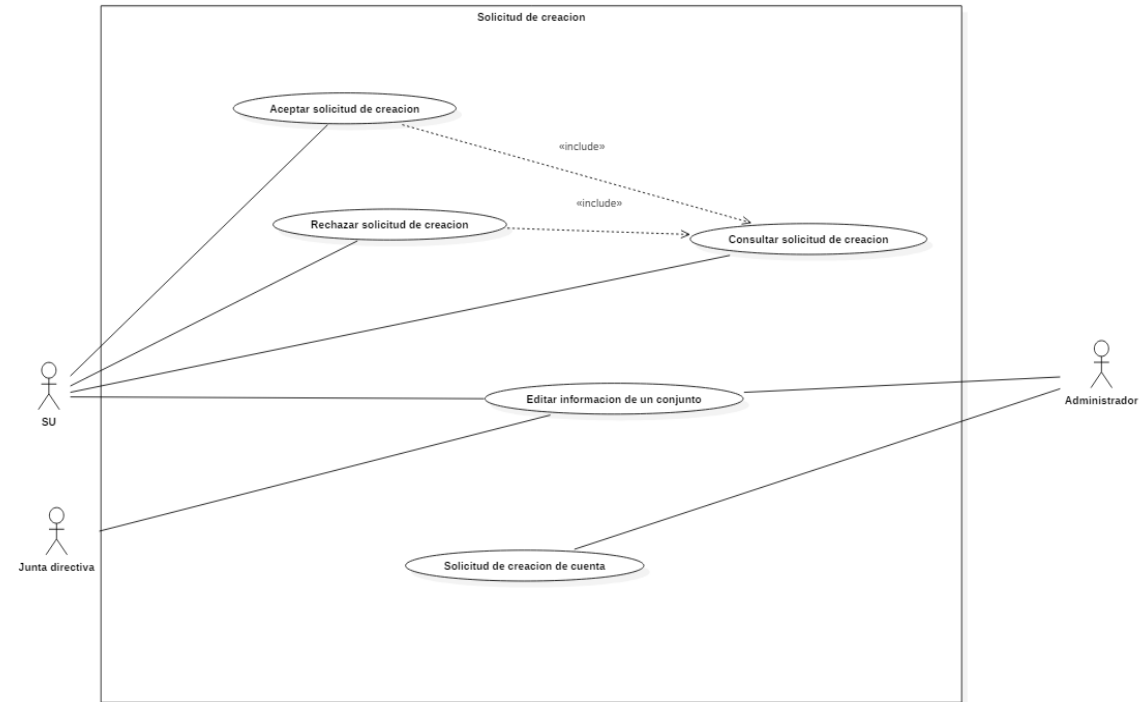


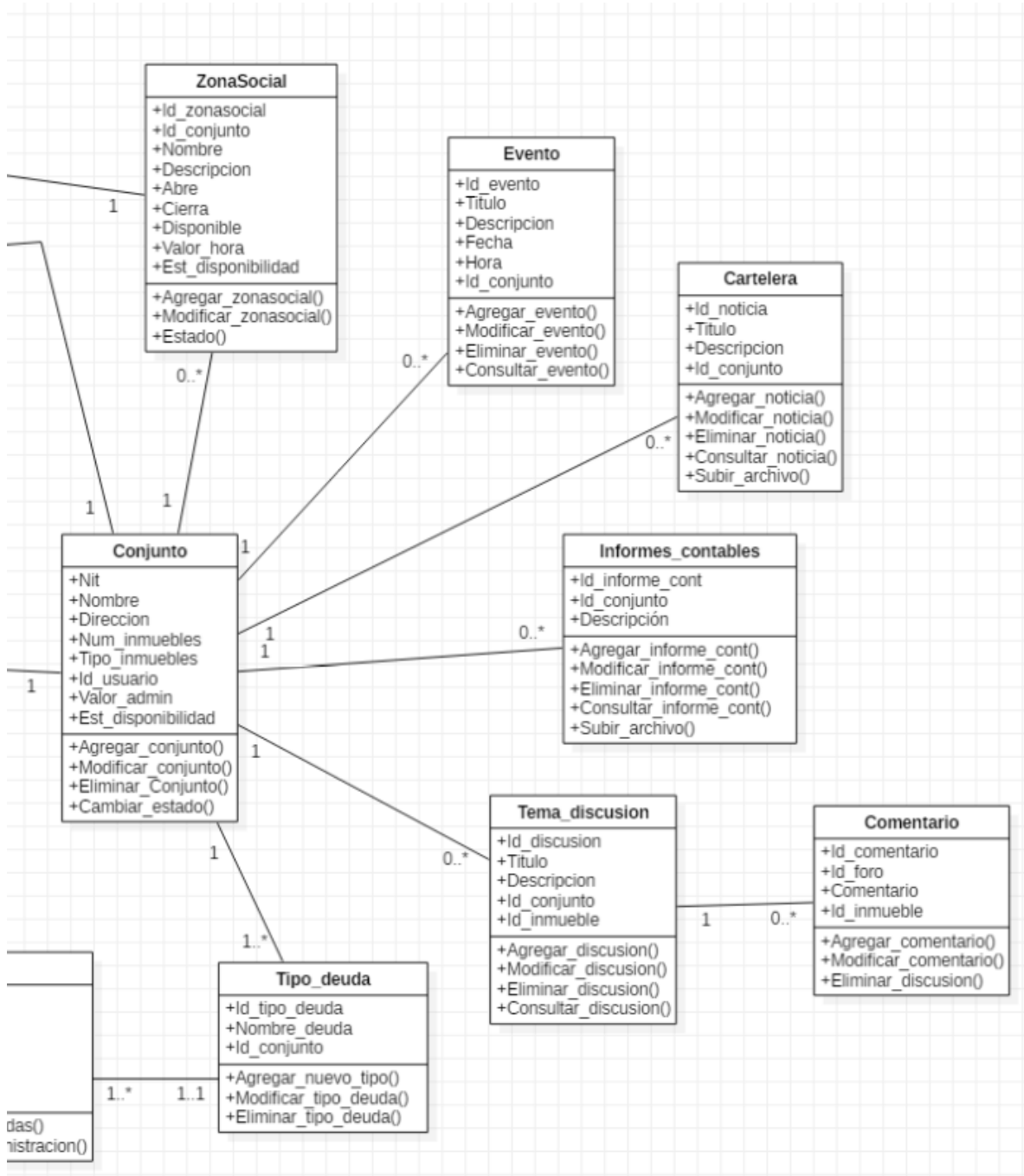
Ilustración 7. Solicitud de creación de conjunto



**3.2.3 Diagrama de clases.** Este tipo de diagrama permite modelar clases incluyendo sus atributos, operaciones, relaciones y asociaciones. Los elementos principales de los diagramas de clases son cajas las cuales representan clases o interfaces. Cada caja se divide en partes horizontales. La parte superior contiene el nombre de la clase, la parte del medio los atributos correspondientes a la misma y finalmente en la parte inferior se pueden visualizar los métodos que para dicha clase aplican. En la siguiente figura puede visualizarse el diagrama de clases creado para este proyecto en particular.



Ilustración 10. Diagrama de clases parte 2



**3.2.4 Modelo de Base de Datos.** Un modelo de base de datos muestra la estructura lógica de la base, incluidas las relaciones y limitaciones que determinan cómo se almacenan los datos y cómo se accede a ellos.

Ilustración 11. Modelo de Base de datos

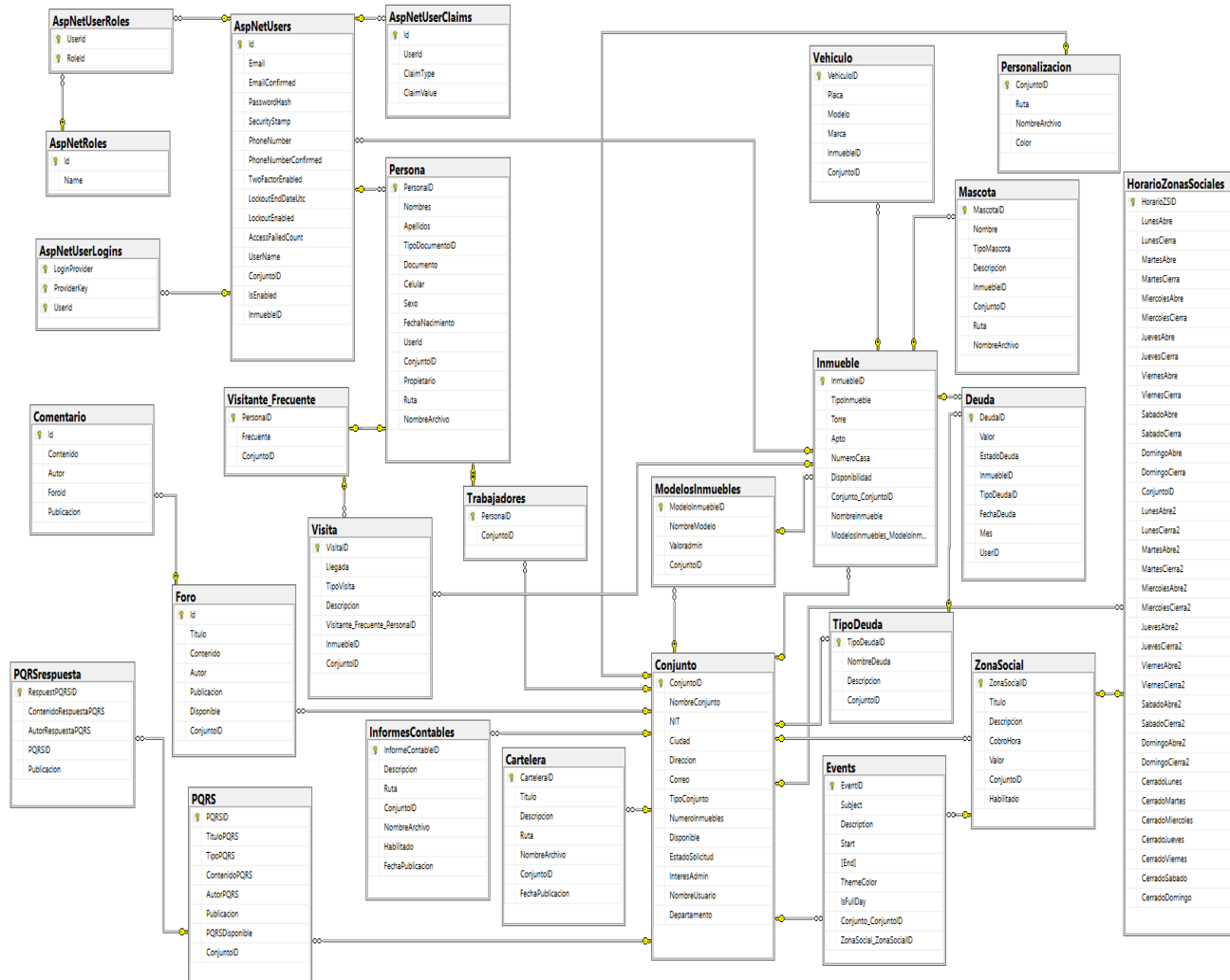


Ilustración 12. Modelo de Base de datos parte 1

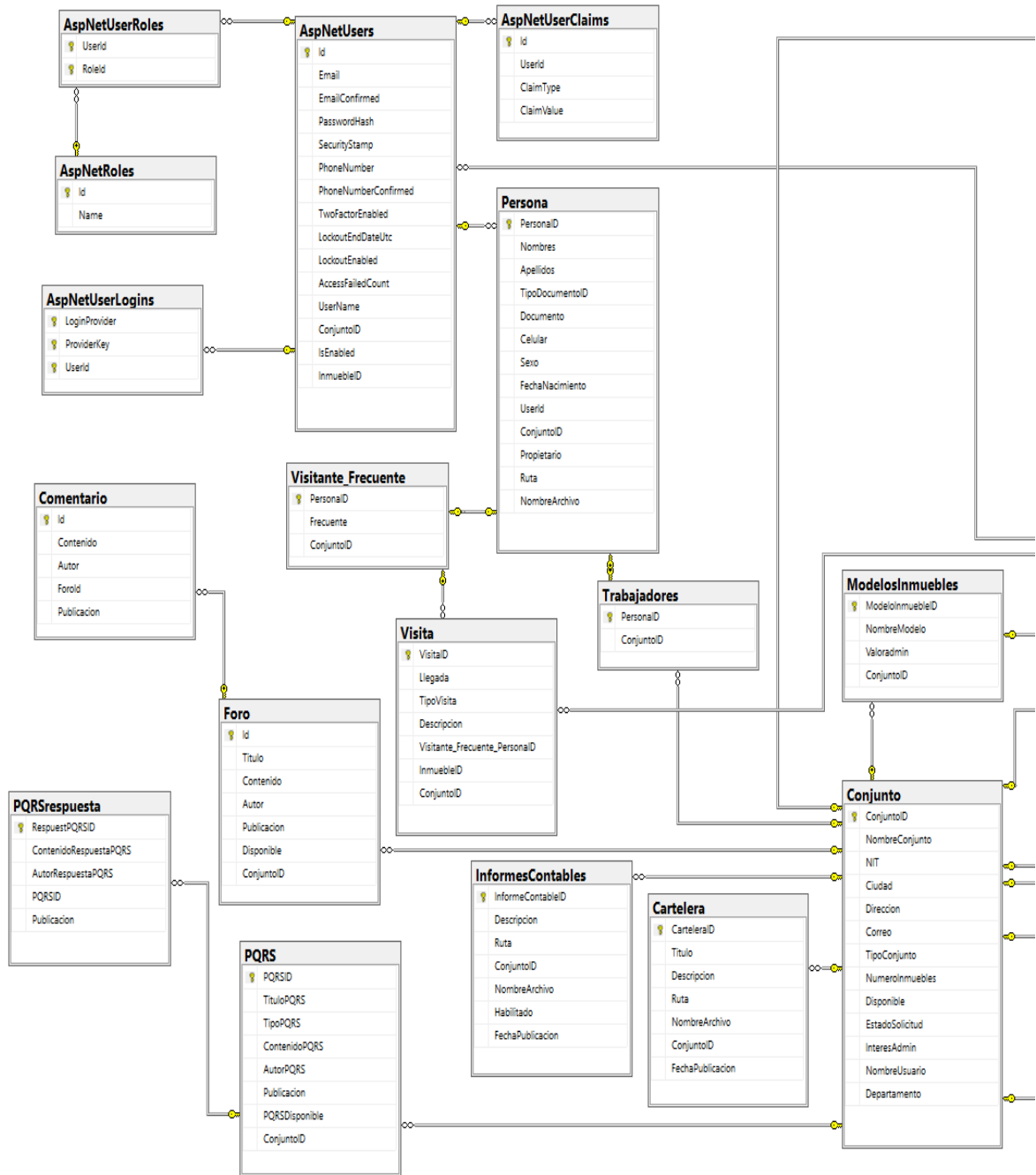
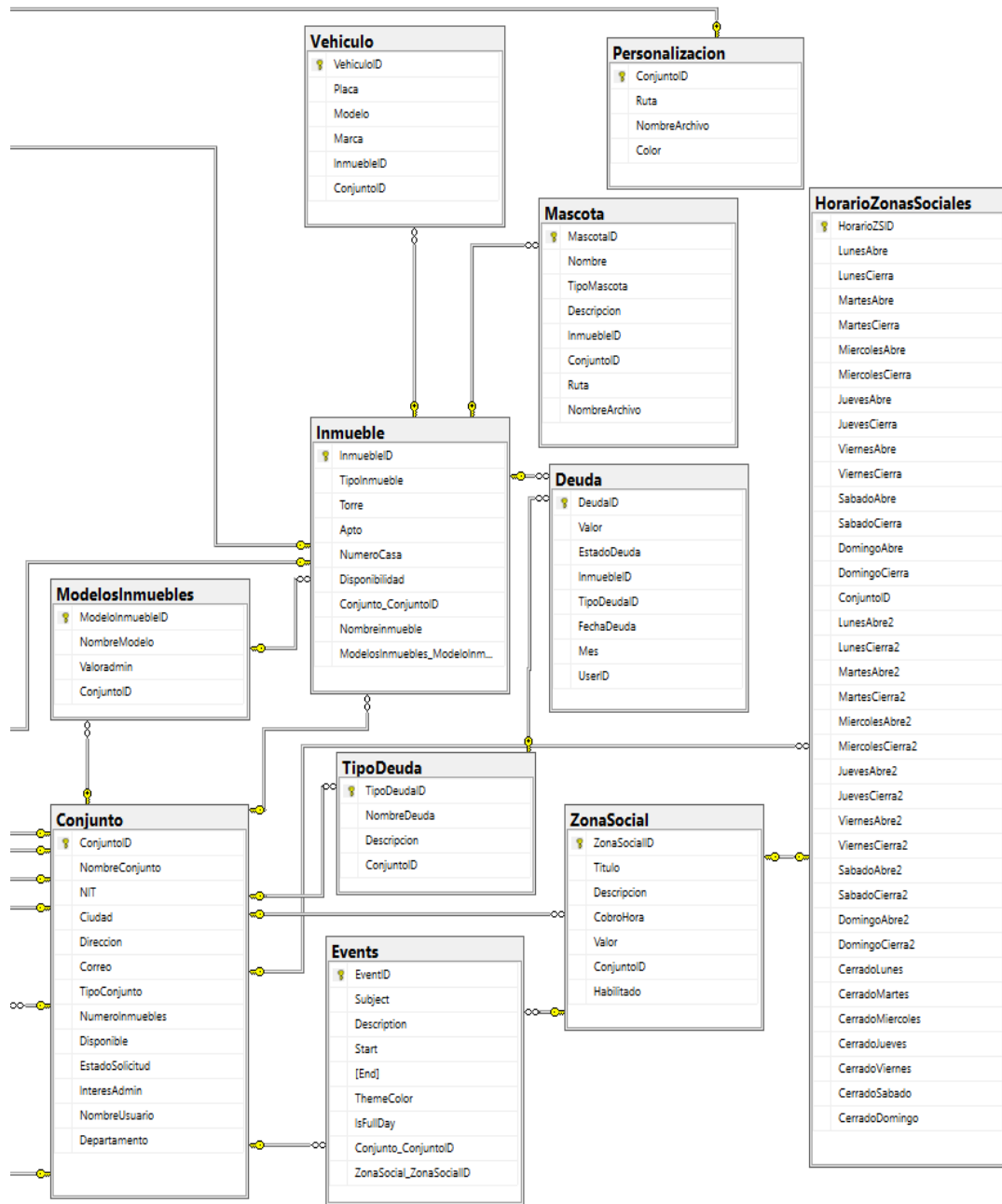


Ilustración 13. Modelo de Base de datos parte 2



**3.2.5 Diagrama de procesos.** Los diagramas de proceso mostrados a continuación representan los procesos con sus respectivas tareas que se realizan en los

conjuntos residenciales. Para ver todos los diagramas de proceso realizados se pueden encontrar en el anexo E.

Ilustración 14. Proceso de acceso visitantes

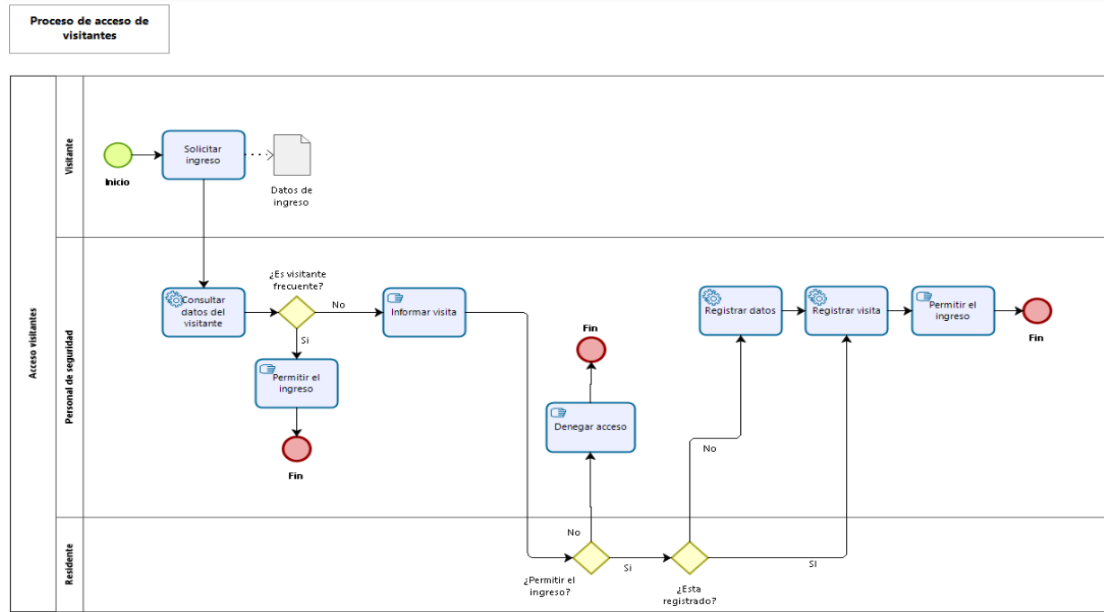


Ilustración 15. Proceso de reserva de una zona social

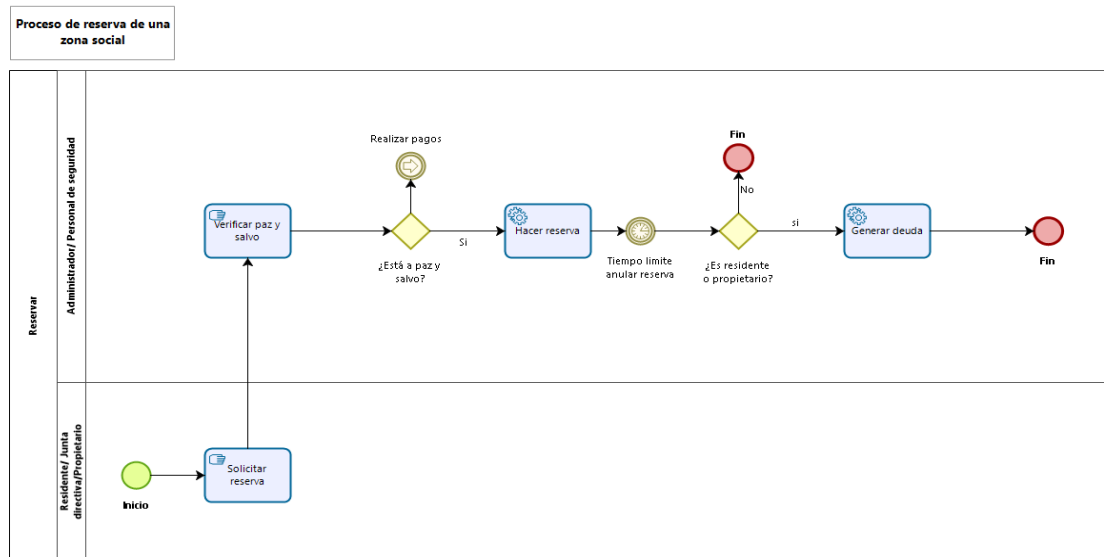
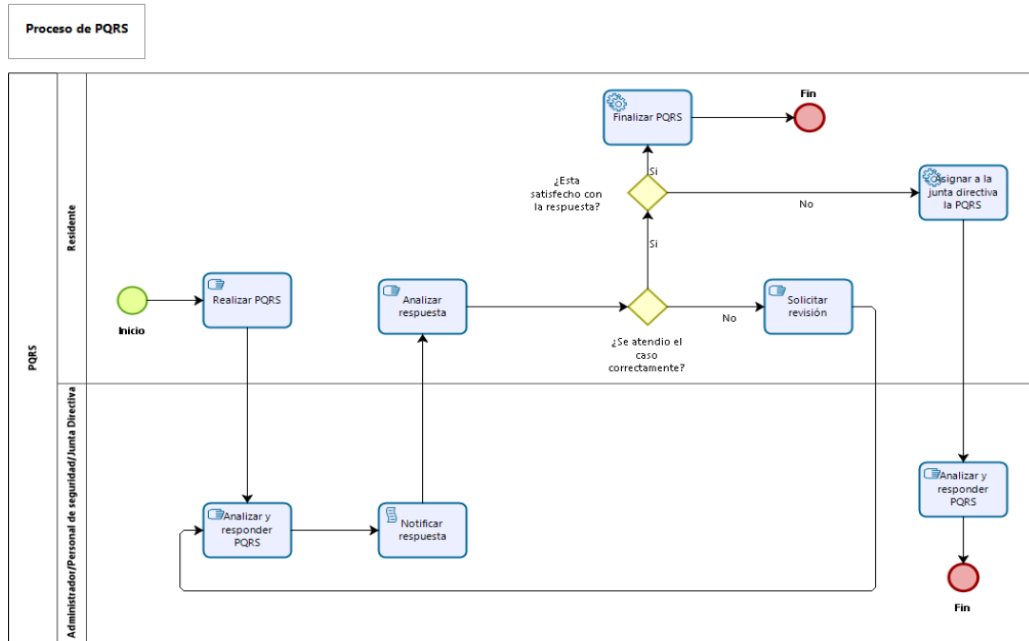


Ilustración 16. Proceso de PQRS



**3.2.6 Diagrama de estados.** Este tipo de diagrama permite de una manera muy sencilla visualizar y entender los estados posibles en los que puede entrar un objeto en particular y las transiciones correspondientes a cada uno, las cuales son acciones o eventos que sirven como detonantes para que el cambio entre estados sea posible.

A continuación, se evidencian los diagramas de estado para los procesos que lo ameriten permitiendo así un entendimiento absoluto por el lector, sin dejar lugar a dudas o ambigüedades.

Ilustración 17. Diagrama de estados PQRS

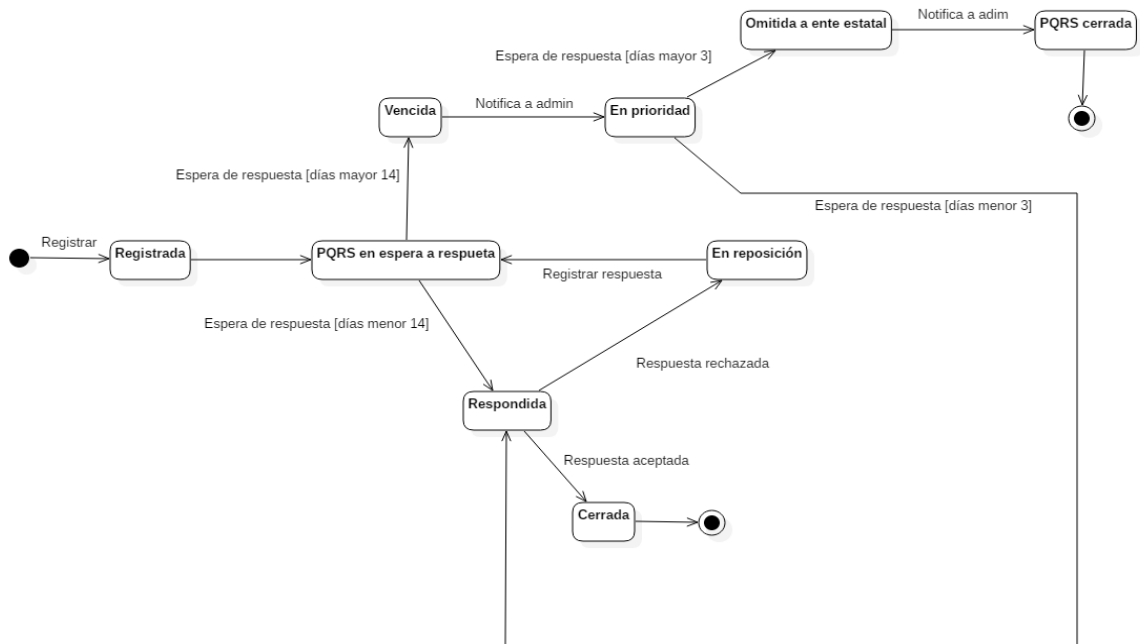


Ilustración 18. Diagrama de estados solicitud de conjunto

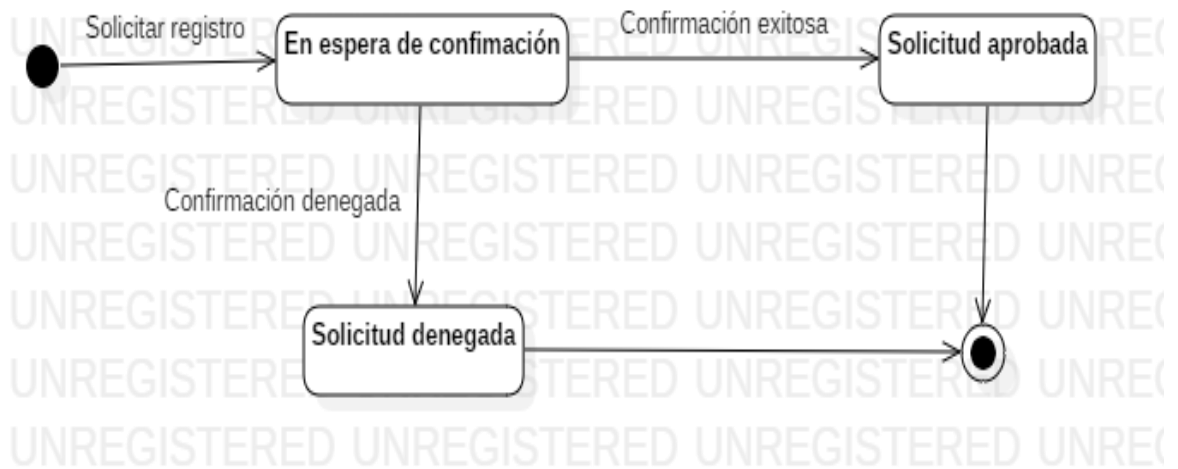
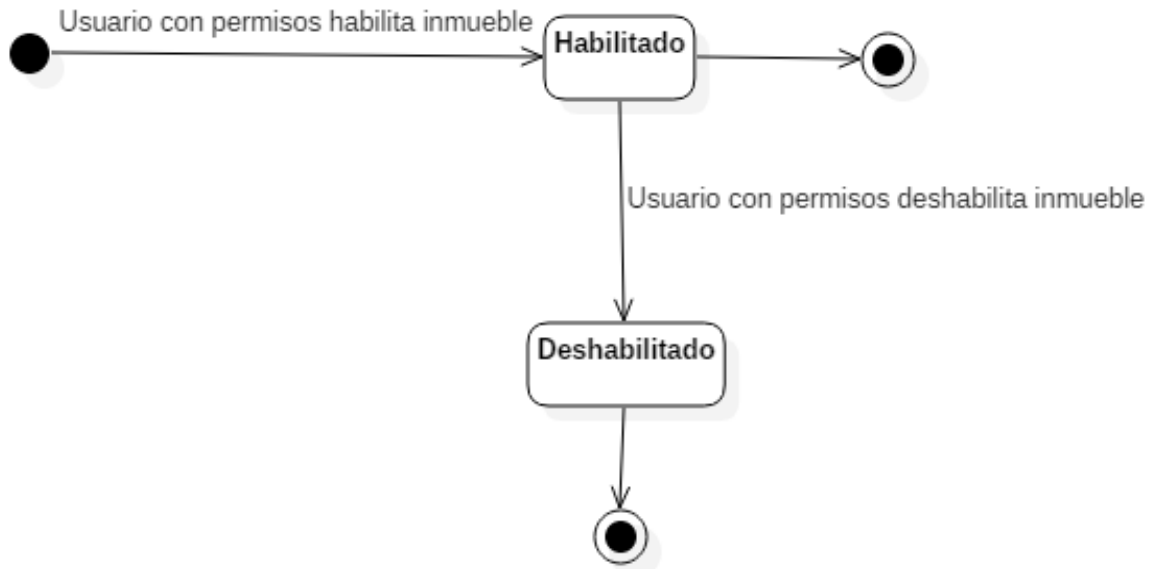
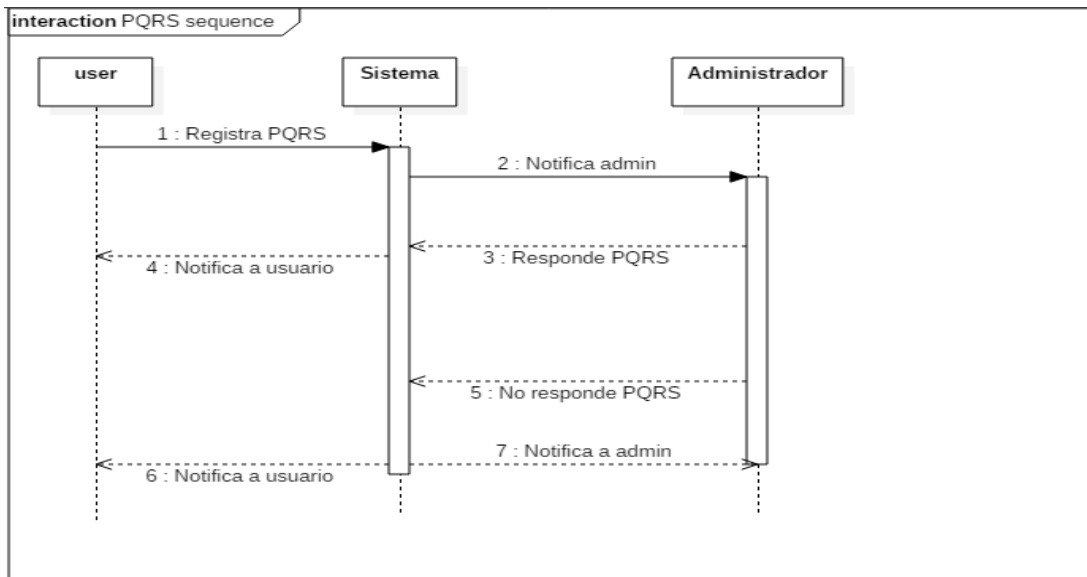


Ilustración 19. Diagrama para estados de inmueble



**3.2.4 Diagrama de secuencia.** Los diagramas de secuencia permiten evidenciar de manera gráfica la forma en la que se comunican los objetos entre sí a lo largo del tiempo.

Ilustración 20. Diagrama de secuencia PQRS



### 3.3 ITERACIONES PARA LA CREACIÓN, EVALUACIÓN Y REFINAMIENTOS DEL PROTOTIPO

Para el desarrollo de este proyecto según la metodología correspondiente se realizaron 4 iteraciones por lo que se decidió dividir el desarrollo por requisitos según las siguientes tablas.

Tabla 15. Requisitos funcionales primera iteración

| <b>Requisito funcional</b> |   |
|----------------------------|---|
| <b>Código</b>              | <b>Nombre</b>   |
| RF01                       | Solicitud de creación de un conjunto  |
| RF51                       | Aceptar o rechazar solicitud de creación de cuenta de un conjunto residencial |
| RF02                       | Creación de un inmueble   |
| RF56                       | Editar inmueble   |
| RF55                       | Inhabilitar inmueble  |
| RF54                       | Habilitar inmueble  |
| RF49                       | Habilitar conjunto residencial  |
| RF50                       | Inhabilitar conjunto residencial  |
| RF05                       | Registrar información de nuevo integrante a la cuenta                         |
| RF04                       | Editar información de un integrante del conjunto                              |
| RF06                       | Inhabilitar integrante del conjunto   |
| RF07                       | Consultar información del integrante del conjunto                             |
| RF57                       | Iniciar sesión  |
| RF08                       | Cambio de contraseña  |
| RF60                       | Agregar rol   |
| RF61                       | Editar rol  |
| RF62                       | Quitar rol  |

Tabla 16. Requisitos funcionales segunda iteración

| <b>Requisito funcional</b> |                           |
|----------------------------|---------------------------|
| <b>Código</b>              | <b>Nombre</b>             |
| RF09                       | Crear PQRS                |
| RF10                       | Consultar PQRS            |
| RF11                       | Responder PQRS            |
| RF26                       | Registrar nuevo visitante |

| <b>Requisito funcional</b> |                                      |
|----------------------------|--------------------------------------|
| <b>Código</b>              | <b>Nombre</b>                        |
| RF27                       | Modificar información del visitante  |
| RF28                       | Consultar información del visitante  |
| RF29                       | Asignar visitante frecuente          |
| RF30                       | Inhabilitar visitante frecuente      |
| RF31                       | Registrar visita                     |
| RF32                       | Consultar historial de visitas       |
| RF37                       | Agregar nueva discusión del foro     |
| RF38                       | Consultar discusiones del foro       |
| RF39                       | Cerrar discusión del foro            |
| RF40                       | Editar tema de discusión en el foro  |
| RF41                       | Agregar comentario                   |
| RF42                       | Editar comentario                    |
| RF43                       | Archivar comentario                  |
| RF12                       | Cargar informes contables            |
| RF13                       | Consultar informes contables         |
| RF14                       | Descargar informe de contabilidad    |
| RF15                       | Editar informes de contabilidad      |
| RF16                       | Inhabilitar informes de contabilidad |
| RF65                       | Habilitar informes de contabilidad   |
| RF66                       | Agregar nueva mascota                |
| RF67                       | Quitar mascota                       |
| RF68                       | Editar mascota                       |
| RF69                       | Visualizar mascota                   |
| RF70                       | Agregar nuevo vehículo               |
| RF71                       | Quitar vehículo                      |
| RF72                       | Editar vehículo                      |
| RF73                       | Visualizar vehículo                  |

Tabla 17. Requisitos funcionales tercera iteración

| <b>Requisito funcional</b> |                                    |
|----------------------------|------------------------------------|
| <b>Código</b>              | <b>Nombre</b>                      |
| RF17                       | Realizar reserva de la zona social |
| RF18                       | Consultar reserva                  |
| RF19                       | Generar deuda de reserva           |
| RF20                       | Editar reserva                     |
| RF21                       | Cancelar reserva                   |
| RF22                       | Agregar evento al calendario       |

| <b>Requisito funcional</b> |                                       |
|----------------------------|---------------------------------------|
| <b>Código</b>              | <b>Nombre</b>                         |
| RF23                       | Consultar evento en el calendario     |
| RF24                       | Modificar evento en el calendario     |
| RF25                       | Archivar evento en el calendario      |
| RF33                       | Agregar información a la cartelera    |
| RF34                       | Consultar información de la cartelera |
| RF35                       | Editar información de la cartelera    |
| RF36                       | Archivar información de la cartelera  |
| RF58                       | Agregar zona social                   |
| RF59                       | Editar zona social                    |
| RF63                       | Inhabilitar zona social               |
| RF64                       | Habilitar zona social                 |

Tabla 18. Requisito funcionales cuarta iteración

| <b>Requisito funcional</b> |   |
|----------------------------|---|
| <b>Código</b>              | <b>Nombre</b>   |
| RF03                       | Personalización de la visualización de la interfaz visual del prototipo |
| RF44                       | Asignar deuda de administración   |
| RF45                       | Asignar deuda   |
| RF46                       | Registrar pago de deuda   |
| RF47                       | Consultar pago  |
| RF48                       | Visualizar conjunto residencial   |
| RF52                       | Editar conjunto residencial   |
| RF53                       | Concluir PQRS   |

En esta etapa se realizaron pruebas para verificar que cada versión del prototipo cumpliera con los requisitos establecidos de cada iteración, la evaluación se realizaba a criterio del grupo de desarrollo puesto que no existe un cliente. En cada evaluación se generó un reporte de las fallas que se daban en los escenarios de pruebas para los casos de uso.

### **3.3.1 Primera Iteración**

3.3.1.1 Construcción del Prototipo. La codificación en Visual Studio inició con la definición de los contextos de trabajo, para esto se dejó el contexto que viene por

defecto para el manejo de los usuarios y los roles y se creó otro donde se encuentra la base de datos del proyecto con los modelos de la base de datos. Debido a que se trabajó usando Code First, en esta primera iteración se crearon los modelos de Conjunto, Inmueble, Persona, UserView y RolesView, además se modificaron los modelos de AccountViewModel e IdentityModels que vienen por defecto para el manejo de los usuarios, ya teniendo el modelo definido en clases, se continuó activando las migraciones para ambos contextos y creando carpetas para ambos contextos con las respectivas migraciones, se continuó haciendo una nueva migración en ambos contextos para que se actualizará la base de datos con estos modelos y sus modificaciones, luego de esto se prosiguió a crear los controladores de cada modelo haciendo uso de la herramienta de Scaffolding de Visual Studio que nos permitía crear los controladores y las vistas con las funciones básicas de Index, Crear, Editar, Eliminar y Detalles, los cuales eran previamente modificados para adecuarlos a las necesidades del prototipo, así mismo se modificó el controlador AccountController junto con la vista Register y Login para que funcionaran de acuerdo a lo especificado en los requisitos funcionales.

Hasta este punto solo se ha tenido en cuenta la funcionalidad del prototipo más no su diseño.

3.3.1.2 Evaluación del Prototipo y Refinamiento del Prototipo. Para la evaluación de esta iteración se usaron los casos de uso que se encuentran en la tabla 19, ya que solo se tuvo en cuenta la parte funcional y no la de diseño en esta iteración, se inició a hacer los casos de prueba ingresando datos de entrada y verificando si los datos de salida eran los esperados, y así sucesivamente con todos los escenarios de los casos de uso que se tomaron en cuenta, también se iba haciendo un reporte de los errores para el refinamiento del prototipo.

En la siguiente tabla se puede ver en detalle el caso de uso y su resultado correspondiente al someterlo a la prueba, se puede ver si tuvo un error, si se aprobó y las observaciones necesarias si en su caso aplican.

Tabla 19. Evaluación casos de uso primera iteración

| Casos de uso |  |        |  |
|--------------|--|--------|--|
| Código       | Nombre   | Aprobó | Observaciones  |
| CU-01        | Iniciar sesión                                     | Si     | Ajustar para iniciar sesión con el nombre de usuario                       |
| CU-02        | Cerrar sesión                                      | Si     |  |
| CU-03        | Solicitud de creación de cuenta                    | Si     | Ajustar para que se envíe un correo al SuperUsuario con la nueva solicitud |
| CU-04        | Consultar solicitud de creación de cuenta          | Si     | Agregar filtro de búsqueda   |
| CU-05        | Aceptar o rechazar solicitud de creación de cuenta | Si     | Agregar correo de rechazo  |
| CU-06        | Habilitar conjunto residencial                     | No     | Cambiar la propiedad enabled de los usuarios asociados al conjunto         |
| CU-07        | Deshabilitar conjunto residencial                  | No     | Cambiar la propiedad enabled de los usuarios asociados al conjunto         |
| CU-08        | Visualizar conjuntos residenciales                 | Si     |  |
| CU-09        | Editar información de un conjunto residencial      | Si     |  |
| CU-10        | Creación de un inmueble                            | Si     |  |
| CU-11        | Visualización de un inmueble                       | Si     |  |
| CU-12        | Modificar información del inmueble                 | Si     | Actualizar el nombre de usuario al modificar el nombre del inmueble        |
| CU-13        | Habilitar inmueble                                 | No     | Cambiar la propiedad enabled de los usuarios asociados al inmueble         |
| CU-14        | Deshabilitar inmueble                              | No     | Cambiar la propiedad enabled de los usuarios asociados al inmueble         |

| Casos de uso |   |        |                                 |
|--------------|---|--------|---------------------------------|
| Código       | Nombre  | Aprobó | Observaciones                   |
| CU-21        | Visualización de la información de un integrante del conjunto | Si     | Modificar el filtro de búsqueda |
| CU-22        | Registrar información de integrante del conjunto              | Si     |                                 |
| CU-23        | Editar información de integrante del conjunto                 | Si     |                                 |
| CU-24        | Inhabilitar integrante del conjunto                           | Si     |                                 |
| CU-25        | Realizar cambio de contraseña                                 | Si     |                                 |
| CU-63        | Crear un rol  | Si     |                                 |
| CU-64        | Editar un rol   | Si     |                                 |
| CU-65        | Quitar un rol   | Si     |                                 |

Teniendo en cuenta el reporte de errores con sus debidas observaciones de la tabla anterior, se procedió a hacer las respectivas correcciones o mejoras del prototipo y así continuar con la segunda iteración.

### 3.3.2 Segunda Iteración

3.3.2.1 Construcción del Prototipo. En esta segunda iteración se desarrollaron los módulos de Visitante\_Frecuente, Visitantes, Foro, PQRS, Mascota y Vehículo, para esto se inició creando los modelos necesarios para los respectivos módulos los cuales fueron los siguientes: Visitante\_Frecuente, Visita, Foro, Comentario, PQRS, PQRSrespuesta, Mascota y Vehiculo, ya teniendo los modelos se realizaron las respectivas migraciones lo que creo las respectivas tablas en la base de datos, luego con ayuda de Scaffolding se crearon los controladores y sus vistas correspondientes para cada modelo, estos se modificaron de manera que los controladores y las vistas se adecuarán a los requisitos del prototipo. Así mismo se

realizaron cambios para permitir el Inicio de sesión con el nombre de usuario y no con el correo, el envío de correos al SuperUsuario al crearse una nueva solicitud de creación de cuenta, se agregó un filtro de búsqueda en el módulo de conjuntos que permitiera realizar la búsqueda mediante las siguientes opciones: “Nombre del conjunto”, “Disponibilidad”, “Estado solicitud”, Se agregó el envío de correos al rechazar una solicitud, se realizó un cambio en el AccountController para que los usuarios asociados a un conjunto inhabilitado no puedan iniciar sesión, igualmente con un inmueble inhabilitado, se realizó un cambio en InmueblesController que permite que se cambie el nombre de usuario automáticamente al realizar el cambio del nombre del inmueble, se modificaron los filtros de búsqueda para poder realizar la búsqueda de nombres incompletos.

3.3.2.2 Evaluación del Prototipo y Refinamiento del Prototipo. Para la evaluación de esta iteración se usaron los casos de uso reflejados en la tabla 20, ya que solo se tuvo en cuenta la parte funcional y no la de diseño en esta iteración, se inició a hacer los casos de prueba ingresando datos de entrada y verificando si los datos de salida eran los esperados, y así sucesivamente con todos los escenarios de los casos de uso que se tomaron en cuenta, también se iba haciendo un reporte de los errores para el refinamiento del prototipo.

En la siguiente tabla se puede ver en detalle el caso de uso y su resultado correspondiente al someterlo a la prueba, se puede ver si tuvo un error, si se aprobó y las observaciones necesarias si en su caso aplican.

Tabla 20. Evaluación casos de uso segunda iteración

| Casos de uso |               |        |                   |
|--------------|---------------|--------|-------------------|
| Código       | Nombre        | Aprobó | Observaciones     |
| CU-26        | Crear PQRS    | Si     |                   |
| CU-27        | Consulta PQRS | Si     | Agregar búsqueda. |

| Casos de uso |                                      |        |                                    |
|--------------|--------------------------------------|--------|------------------------------------|
| Código       | Nombre                               | Aprobó | Observaciones                      |
| CU-28        | Responder una PQRS                   | Si     |                                    |
| CU-29        | Cargar informes de contabilidad      | Si     |                                    |
| CU-30        | Consultar informes de contabilidad   | Si     |                                    |
| CU-31        | Descargar informes de contabilidad   | Si     |                                    |
| CU-32        | Inhabilitar informes de contabilidad | Si     |                                    |
| CU-33        | Editar informes de contabilidad      | No     | No actualiza el archivo que carga. |
| CU-41        | Registrar nuevo visitante            | Si     |                                    |
| CU-42        | Consultar información del visitante  | Si     |                                    |
| CU-43        | Modificar información del visitante  | Si     |                                    |
| CU-44        | Asignar visitante frecuente          | Si     |                                    |
| CU-45        | Inhabilitar visitante frecuente      | Si     |                                    |
| CU-46        | Registrar visita                     | Si     |                                    |
| CU-49        | Consultar historial de visitas       | Si     |                                    |
| CU-52        | Agregar nueva discusión al foro      | Si     |                                    |
| CU-53        | Consultar discusiones del foro       | Si     |                                    |
| CU-54        | Editar tema de discusión en el foro  | Si     |                                    |
| CU-55        | Cerrar discusión del foro            | Si     |                                    |
| CU-56        | Agregar comentario                   | Si     |                                    |

| Casos de uso |                     |        |  |
|--------------|---------------------|--------|--|
| Código       | Nombre              | Aprobó | Observaciones  |
| CU-57        | Editar comentario   | No     | Arreglarlo para que solo lo pueda editar quien lo creo   |
| CU-58        | Archivar comentario | No     | Arreglarlo para que solo lo pueda archivar quien lo creo |

Teniendo en cuenta el reporte de errores con sus debidas observaciones de la tabla anterior, se procedió a hacer las respectivas correcciones o mejoras del prototipo y así continuar con la tercera iteración.

### 3.3.3 Tercera Iteración

3.3.3.1 Construcción del Prototipo. En esta tercera iteración se desarrollaron los módulos de Reserva, Events, Cartelera y Zonas Sociales, para esto se comenzó creando los modelos necesarios para los respectivos módulos, los modelos generados fueron los siguientes: Events que contiene lo necesario para el módulo de reservas y eventos, Cartelera, ZonaSocial y HorarioZonasSociales, teniendo los modelos listos se realizaron las migraciones en la base de datos para que se crearan las respectivas tablas, luego con ayuda de Scaffolding se crearon los controladores y las vistas correspondientes a cada modelo, estos se modificaron para que cumplieran con los requisitos del prototipo.

3.3.3.2 Evaluación del Prototipo y Refinamiento del Prototipo. Para la evaluación de esta iteración se usaron los casos de uso reflejados en la tabla 21, ya que solo se tuvo en cuenta la parte funcional y no la de diseño en esta iteración, se inició a hacer los casos de prueba ingresando datos de entrada y verificando si los datos de salida eran los esperados, y así sucesivamente con todos los escenarios de los casos de uso que se tomaron en cuenta, también se iba haciendo un reporte de los errores para el refinamiento del prototipo.

En la siguiente tabla se puede ver en detalle el caso de uso y su resultado correspondiente al someterlo a la prueba, se puede ver si tuvo un error, si se aprobó y las observaciones necesarias si en su caso aplican.

Tabla 21. Evaluación casos de uso tercera iteración

| Casos de uso |  |        |               |
|--------------|--|--------|---------------|
| Código       | Nombre   | Aprobó | Observaciones |
| CU-34        | Realizar reserva de zona social                | Si     |               |
| CU-35        | Consultar reservas                             | Si     |               |
| CU-36        | Cancelar reserva                               | Si     |               |
| CU-37        | Agregar evento al calendario                   | Si     |               |
| CU-38        | Consultar evento en el calendario              | Si     |               |
| CU-39        | Modificar evento en el calendario              | Si     |               |
| CU-40        | Archivar evento en el calendario               | Si     |               |
| CU-48        | Agregar información a la cartelera             | Si     |               |
| CU-49        | Consultar información de la cartelera          | Si     |               |
| CU-50        | Editar información de la cartelera             | Si     |               |
| CU-51        | Archivar información de la cartelera           | Si     |               |
| CU-15        | Agregar nueva zona social al conjunto          | Si     |               |
| CU-16        | Visualizar una zona social                     | Si     |               |
| CU-17        | Editar información de zona social del conjunto | Si     |               |

| Casos de uso |                                       |        |               |
|--------------|---------------------------------------|--------|---------------|
| Código       | Nombre                                | Aprobó | Observaciones |
| CU-18        | Deshabilitar zona social del conjunto | Si     |               |
| CU-19        | Habilitar zona social del conjunto    | SI     |               |

Teniendo en cuenta el reporte de errores con sus debidas observaciones de la tabla anterior, se procedió a hacer las respectivas correcciones o mejoras del prototipo y así continuar con la tercera iteración.

### 3.3.4 Cuarta Iteración

3.3.4.1 Construcción del Prototipo. En la cuarta iteración se desarrollaron los módulos de Deudas, Modelos Inmuebles y Trabajadores, para esto se inició creando los modelos necesarios para los respectivos módulos, los modelos generados en esta iteración fueron los siguientes: Deuda, TipoDeuda, ModelosInmuebles, Trabajadores, teniendo los modelos listos se continuo haciendo las migraciones en la base de datos para que se crearan las respectivas tablas, luego con ayuda de Scaffolding se crearon los controladores y las vistas correspondientes a cada modelo, estos se modificaron para que cumplieran con los requisitos del prototipo.

Así mismo se modificó el EventsController para añadir la propiedad de generar deudas al realizar reservas, se añadió un controlador extra asociado al modelo de usuarios el que permite crear una cuenta para un propietario que no sea residente. Se añadió la propiedad de verificar correo para poder realizar el cambio de contraseña.

3.3.4.2 Evaluación del Prototipo y Refinamiento del Prototipo. Para la evaluación de esta iteración se usaron los casos de uso reflejados en la tabla 22, ya que solo se

tuvo en cuenta la parte funcional y no la de diseño en esta iteración, se inició a hacer los casos de prueba ingresando datos de entrada y verificando si los datos de salida eran los esperados, y así sucesivamente con todos los escenarios de los casos de uso que se tomaron en cuenta, también se iba haciendo un reporte de los errores para el refinamiento del prototipo.

En la siguiente tabla se puede ver en detalle el caso de uso y su resultado correspondiente al someterlo a la prueba, se puede ver si tuvo un error, si se aprobó y las observaciones necesarias si en su caso aplican.

Tabla 22. Evaluación casos de uso cuarta iteración

| Casos de uso |   |        |               |
|--------------|---|--------|---------------|
| Código       | Nombre  | Aprobó | Observaciones |
| CU-59        | Asignar deuda                                       | Si     |               |
| CU-60        | Visualizar historial de pagos                       | Si     |               |
| CU-61        | Registrar pago deuda                                | Si     |               |
| CU-62        | Concluir PQRS                                       | Si     |               |
| CU-08        | Visualizar conjuntos residenciales                  | Si     |               |
| CU-09        | Editar información de un conjunto residencial       | Si     |               |
| CU-20        | Personalización de la interfaz visual del prototipo | Si     |               |

Teniendo en cuenta el reporte de errores con sus debidas observaciones de la tabla anterior, se procedió a hacer las respectivas correcciones o mejoras del prototipo y así continuar con la tercera iteración. Finalmente se contaba con un prototipo funcional que cumpliera con los casos de uso de esta y las iteraciones anteriores al

finalizar esta iteración por lo que con esta última se culminó el desarrollo del prototipo y el proyecto.

#### **4. CONCLUSIONES**

Dado a que Homie es una plataforma para uso de conjuntos residenciales fue indispensable identificar y generalizar sus procesos con el fin de tener homogeneidad en los mismos y poder brindar una solución viable para su desarrollo que pudiera adecuarse a cualquier conjunto residencial teniendo en cuenta la opinión de los administradores de algunos conjuntos del área metropolitana de Bucaramanga.

Es necesario entender y tener claras las leyes que regulan la propiedad horizontal y la jurisdicción que en ellas aplica, adicionalmente se debe tener conocimiento de cómo funciona la administración de los mismos y la relación que existe entre todos sus integrantes para de esta forma crear un entorno virtual que tenga en cuenta a toda la comunidad.

## 5. RECOMENDACIONES

Una vez finalizado el proyecto y teniendo en cuenta que como todo desarrollo software debe estar en constante evolución y mantenimiento se plantean las siguientes recomendaciones:

1. Es recomendable capacitar al administrador, personal de seguridad y junta directiva sobre la correcta utilización del prototipo.
2. Se recomienda realizar un ajuste a la base de datos para que un propietario pueda tener asignado varios conjuntos residenciales en caso de ser necesario, de la misma forma los propietarios pueden poseer inmuebles en diferentes conjuntos residenciales, lo cual no fue concebido en este prototipo.

## BIBLIOGRAFIA

BALSAMIQ STUDIOS. Balsamiq Wireframes. [En línea]. (Consultado el 12 de noviembre del 2019). Disponible en: <https://balsamiq.com/wireframes/>

BIZAGI. Estándares Bizagi modelamiento de procesos. [En línea]. (Consultado el 12 de noviembre del 2019). Disponible en: <https://www.bizagi.com/es/productos/beneficios/estandares>

BUCARAMANGA METROPOLITANA CÓMO VAMOS. Encuesta de percepción ciudadana 2018. [En línea]. Bucaramanga: Universidad de Santander, et al., 2018. (Consultado el 12 de noviembre del 2019). Disponible en: [https://www.bucaramangacomovamos.org/post/descarga-la-9na-encuesta-de-percepci%C3%B3n-ciudadana?fbclid=IwAR3BJaXoFLnKHFEhunZDE7TvbZm\\_kf3\\_EsCUHngoW7XrVi4EtiR0pKOoHLQ](https://www.bucaramangacomovamos.org/post/descarga-la-9na-encuesta-de-percepci%C3%B3n-ciudadana?fbclid=IwAR3BJaXoFLnKHFEhunZDE7TvbZm_kf3_EsCUHngoW7XrVi4EtiR0pKOoHLQ)

BUSINESS PROCESS MODEL AND NOTATION. *Home*. [En línea]. Object Management Group. (Consultado el 12 de noviembre del 2019). Disponible en: <http://www.bpmn.org/>

COLABORADORES DE WIKIPEDIA. Language Integrated Query. [En línea]. Wikipedia, 2019. (Consultado el 12 de noviembre del 2019). Disponible en: [https://es.wikipedia.org/wiki/Language\\_Integrated\\_Query](https://es.wikipedia.org/wiki/Language_Integrated_Query)

COLOMBIA. CONGRESO DE LA REPÚBLICA. Ley 675 (3, agosto, 2001). por medio de la cual se expide el régimen de propiedad horizontal. [En línea]. Diario Oficial, Bogotá, 2001, No. 44.509, capítulo 1, artículo 1. (Consultado el 12 de noviembre del 2019). Disponible en: [http://www.sic.gov.co/sites/default/files/normatividad/Ley\\_675\\_2001.pdf](http://www.sic.gov.co/sites/default/files/normatividad/Ley_675_2001.pdf)

DANE. Población por edad y sexo. [En línea]. Bogotá: Gobierno de Colombia, 2015. (Consultado el 12 de noviembre del 2019). Disponible en: <https://geoportal.dane.gov.co/midaneapp/pob.html>

GUTHRIE, Scott; BOYD, Eric; LIUSON, Julia; y HANSELMAN, Scott. What is ASP.NET? [En línea]. Microsoft, 2019. (Consultado el 12 de noviembre del 2019). Disponible en: <https://dotnet.microsoft.com/learn/web/what-is-aspnet>

IEEE. 830-1998 - IEEE Recommended Practice for Software Requirements Specifications. [En línea]. 2009. (Consultado el 12 de noviembre del 2019). Disponible en: <https://ieeexplore.ieee.org/document/720574>

JAYMAX. Coding Made Easy: C#. [En línea]. Microsoft, 2018. (Consultado el 12 de noviembre del 2019). Disponible en: <https://www.microsoft.com/es-co/p/coding-made-easy-c/9nblggh4pzql?activetab=pivot:overviewtab>

LAUDON, Kenneth y LAUDON, Jane. Sistemas de información gerencial (Decimosegunda ed.). [En línea]. México: Pearson (ROMERO, Alfonso trad.), 2012, p. 47. (Consultado el 12 de noviembre del 2019). Disponible en: <https://juanantonioleonlopez.files.wordpress.com/2017/08/sistemas-de-informacic3b3n-gerencial-12va-edicic3b3n-kenneth-c-laudon.pdf>

LUCIDCHART. Qué es el lenguaje unificado de modelado (UML). [En línea]. Lucid Software. (Consultado el 12 de noviembre del 2019). Disponible en: <https://www.lucidchart.com/pages/es/que-es-el-lenguaje-unificado-de-modelado-uml?a=0>

MAIDA, Esteban y PACIENZIA, Julián. Metodologías de desarrollo de software [en línea]. Tesis de Licenciatura en Sistemas y Computación. Facultad de Química e Ingeniería "Fray Rogelio Bacon". Universidad Católica Argentina, 2015. (Consultado

el 12 de noviembre del 2019). Disponible en: <https://repositorio.uca.edu.ar/handle/123456789/522>

MICROSOFT. Introducción al lenguaje C# y .NET Framework. [En línea]. 2015. (Consultado el 12 de noviembre del 2019). Disponible en: <https://docs.microsoft.com/es-es/dotnet/csharp/getting-started/introduction-to-the-csharp-language-and-the-net-framework>

MISCONDOMINIOS. Página de inicio. [En línea]. Onyx. (Consultado el 12 de noviembre del 2019). Disponible en: <http://www.miscondominios.com/>

MKLABS CO. Start UML software. [En línea]. (Consultado el 12 de noviembre del 2019). Disponible en: <http://staruml.io/>

PRESSMAN, Roger. *Ingeniería del software: Un enfoque práctico*. (Quinta ed.). Madrid, España: McGraw Hill, 2002. ISBN: 0-07-709677-0

UNIFIED MODELING LANGUAGE. Introduction to UML. [En línea]. Object Management Group, 2005. (Consultado el 12 de noviembre del 2019). Disponible en: <http://www.uml.org/what-is-uml.htm>

VIVOOK. Inicio. [En línea]. (Consultado el 12 de noviembre del 2019). Disponible en: <https://www.vivook.com/>

WARREN, Genevieve, et al. Introduction to LINQ Queries (C#). [En línea]. Microsoft, 2015. (Consultado el 12 de noviembre del 2019). Disponible en: <https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/concepts/linq/introduction-to-linq-queries>

WENZEL, Maira, et al. Entity Framework overview. [En línea]. Microsoft, 2018. (Consultado el 12 de noviembre del 2019). Disponible en: <https://docs.microsoft.com/en-us/dotnet/framework/data/adonet/ef/overview>