



UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO
CENTRO UNIVERSITARIO UAEM TEXCOCO

“SISTEMA DE INFORMACIÓN PARA LA ADMINISTRACIÓN
DEL COBRO DE USO DE LOS POZOS DE AGUA POTABLE
EN EL PUEBLO DE SAN MIGUEL TOCUILA.”

T E S I S

QUE PARA OBTENER EL TÍTULO DE:
INGENIERO EN COMPUTACIÓN

P R E S E N T A:

AARÓN PÉREZ GUERRERO

DIRECTOR DE TESIS:

M. EN ISC. IRENE AGUILAR JUÁREZ

REVISORES:

DR. JOEL AYALA DE LA VEGA
DR. ALFONSO ZARCO HIDALGO

TEXCOCO, MÉXICO
MARZO 2013



Índice

Agradecimientos	3
Dedicatorias	4
Índice	5
Índice de Tablas	10
I. Introducción	12
II. Planteamiento del problema	14
III. Objetivos	15
IV. Justificación	16
CAPÍTULO 1. Sistemas de información Y RAD	17
1.1. Concepto de un sistema de información.....	17
1.1.1 Que es un sistema	17
1.1.2. Que es información	18
1.2. Tipos de sistemas de información.....	18
1.3. Objetivo de un sistema de información	20
1.1 . Modelo del RAD.....	21
1.4.1. Características del RAD	24
1.4.2. Ventajas y desventajas del RAD	26
CAPÍTULO 2. Análisis y diseño de sistemas de información	27
2.1. Análisis de requerimientos	27
2.1.2 Requerimientos funcionales	27
2.1.2. Requerimientos no funcionales.....	29
2.2. Identificación de las clases de usuarios.....	29
2.2.1. Definición de los actores para generar casos de uso.....	30
2.2.2. Identificación de los casos de uso	31
2.2.3. Diagramas de casos de uso	32
2.2.4. Descripción de los casos de uso del usuario.	35
2.2.5. Descripción de los casos de uso del Administrador.....	41
2.3. Modelado de contenido del SI de agua potable.	59
2.3.1 Modelado de contenido para usuario (dueño).	59



2.3.2. Modelado de contenido para el administrador	61
2.4. Modelado funcional del sistema de información.....	64
2.4.1. Diagramas de actividades del sistema.....	65
2.5. Diseño de la base de datos.	74
2.5.1. Identificación de las entidades básicas.....	75
2.5.2. Diagrama entidad – relación.....	75
2.5.3 Creación de la entidades administrador / dueño.....	78
2.5.4 Creación de las entidades del sistema de pago.....	79
2.5.5 Modelo relacional.....	80
2.5.6. Diccionario de datos.....	84
2.6. Codificación del sistema	84
2.6.1. Tecnologías utilizadas	84
2.6.2. Java	85
2.6.3. Swing.....	85
2.6.5. iReport	86
2.7. Diseño de Interfaz	87
2.7.1. Diseño de la interfaz de usuario.....	87
2.7.3. Diseño arquitectónico.....	94
2.8. Diseño de navegación.....	95
2.8.1. Plan de navegación con símbolos de maquetación.....	96
2.8.2. Plan de navegación Usuario	99
2.8.3 Plan de navegación administración.....	100
2.8.4. Plan de navegación para pago.....	101
2.8.5 Plan de navegación para asamblea.....	102
2.8.6. Plan de navegación para generar reporte.....	103
2.9. Pruebas del sistema de Información.....	104
2.9.1. Pruebas de Contenido.....	105
2.9.2. Pruebas de la interfaz del sistema.....	106
2.9.3. Pruebas de configuración.....	107
2.9.4. Pruebas de seguridad	108



V. Resultados	109
VI. Conclusiones y Recomendaciones	117
VII Bibliografía	120
VIII. Anexo	122



Índice de figuras

Figura 1. Modelo de un sistema	19
Figura 2. Esquema de una metodología clásica de desarrollo de sistemas (Luis Castellanos, 2011).	20
Figura 3. Esquema de una metodología evolutiva incremental de desarrollo de sistemas (Luis Castellanos, 2011).	20
Figura 4. Modelo RAD de 60 a 90 Días.	23
Figura 5. Actores del sistema de información.	30
Figura 6. Diagrama de casos de uso para usuarios.	33
Figura 7. Diagrama de casos de uso para administradores.	34
Figura 8. “contenido principal”	59
Figura 9. “consulta de faltas”	60
Figura 10. Consultar retrasos	60
Figura 11. Generar Reporte	61
Figura 12. Ingresar al sistema	61
Figura 13. Administrar usuario.	62
Figura 15. Administrar administrador	63
Figura 16. Limpiar cuenta de usuario	64
Figura 17. Diagrama de actividad para “Ingresar al sistema de información”.	65
Figura 18. Diagrama de actividad para “Consulta de adeudo”.	66
Figura 19. Diagrama de actividad para “Generar reporte”.	67
Figura 20. Diagrama de actividad “Generar reporte” en el caso del administrador.	68
Figura 21. Diagrama de actividad para dar de alta a un usuario.	69
Figura 22. Diagrama de actividad para realizar la eliminación de un usuario.	70
Figura 23. Diagrama de actividad para modificar un usuario.	71
Figura 24. Diagrama de actividad para abrir nueva sesión	72
Figura 25. Diagrama de actividad para “registrar asistentes”.	73
Figura 26. Diagrama de actividad para cerrar o eliminar sesión de asamblea.	74
Figura 28. Estructura normalizada de las entidades administrador/ dueño.	78
Figura 29. Estructura del sistema de pago.	79
Figura 30. Diagrama relacional del SI de agua potable.	83
Figura 31. Pirámide de diseño de interfaz	87



Figura 32. Interfaz de usuario	88
Figura 33. Primer diseño de la interfaz (pantalla de inicio).	89
Figura 34. Bosquejo inicial de la interfaz del administrador.	89
Figura 35. Primer diseño de la interfaz administrador.	90
Figura 36. Bosquejo inicial de la interfaz de dueño de toma.	91
Figura 37. Primer diseño de la interfaz de dueño de toma.	92
Figura 38. Arquitectura sistema centralizado.	94
Figura 39. Símbolos de maquetación (Aguilar Juárez, 2011).	95
Figura 40. Plan de navegación maquetación.	96
Figura 41. Diseño de navegación de la ruta de usuario.	99
Figura 42. Plan de navegación para la administración.	100
Figura 43. Diseño de navegación, para realizar pago.	101
Figura 44. Plan de navegación para asamblea	102
Figura 45. Plan de navegación para generar reporte.	103
Figura 46. Proceso de pruebas.	104
Figura 47. Resultados acceso a usuarios, asambleas, atrasos, reporte.	113
Figura 48. Resultado de acceso a administración, ABC administración y usuarios.	114
Figura 49. Resultado realizar pago, administrar asamblea, generar reporte.	115
Figura 50. Pantallas del sistema completo final.	116



Índice de Tablas

Tabla 1. Requerimientos del usuario	28
Tabla 2. Requerimientos del administrador.	28
Tabla 3. Requerimientos del ingeniero.	28
Tabla 4. Requerimientos no funcionales	29
Tabla 5. Descripción del formato para los casos de uso.	32
Tabla 6. Caso de uso “ingresando al sistema”.	35
Tabla 7. Caso de uso “consulta de adeudos”	36
Tabla 8. Caso de Uso “Atrasos”	37
Tabla 9. Caso de Uso “Faltas”	38
Tabla 10. Caso de Uso “Generar Reporte”	39
Tabla 11. Caso de Uso “Imprime”	40
Tabla 12. Caso de uso “ingresar al sistema”.	41
Tabla 13. Caso de Uso “Consulta General.”	42
Tabla 14. Caso de Uso “Adeudo”	43
Tabla 15. Caso de Uso “Generar Reporte”	44
Tabla 16. Caso de Uso “Administra usuario”	45
Tabla 17. Caso de Uso “Generar nuevo usuario”	46
Tabla 18. Caso de Uso “Eliminar usuario”	47
Tabla 19. Caso de Uso “Modificar usuario”	48
Tabla 20. Caso de Uso “Imprimir”	49
Tabla 21. Caso de Uso “Asamblea”	50
Tabla 22. Caso de Uso “Nueva asamblea”	51
Tabla 23. Caso de Uso “Registrar asistentes”	52
Tabla 24. Caso de Uso “Eliminar sesión”	53
Tabla 25. Caso de Uso “Administrar administrador”	54
Tabla 26. Caso de Uso “Alta administrador”	55
Tabla 27. Caso de Uso “Eliminar administrador”	56
Tabla 28. Caso de Uso “Modificar administrador”	57
Tabla 29 Caso de uso. “Limpiar cuenta de usuario”	58
Tabla 30. Grados de normalización.	81
Tabla 31. Documentación de código	86



Tabla 32. Pruebas de contenido.	105
Tabla 33. Pruebas de la interfaz del sistema.	106
Tabla 34. Pruebas de configuración.	107
Tabla 35. Pruebas de Seguridad.	108
Tabla 36. Tabla de resultados por página.	110
Tabla 37. Comparación de tiempos.	112



I. Introducción

En el pueblo de San Miguel Tocuila existe un comité que lleva a cabo toda la administración de los pozos de agua potable, esto fue fundado aproximadamente desde los años 50s , en esos tiempos no existían las Tecnologías de la Información (TI) y mucho menos las nuevas herramientas tecnológicas, entonces esta administración desde sus inicios se ha hecho cargo de una forma manual de los registros que se llevaron a cabo a través de libros usados como bitácoras del día a día, esto hasta nuestros días aún sigue en operación y no por eso quiere decir que siga siendo funcional, ya que tomando de ejemplo la historia de nuestros ancestros cuando estábamos en las cavernas y éramos cavernícolas, entonces se realizaba la caza de animales para poder alimentar a toda la aldea, veíamos los inicios de la administración en donde los que realizaban la caza eran los hombres, y las mujeres esperaban la carne para poder prepararla, después cuando pudieron razonar más se dieron cuenta que si solo iban los hombres con mayor fortaleza física era más probable una caza exitosa, a esto le podemos mencionar que estaban haciendo uso de datos anteriores para poder tomar esta decisión, que usando estos conjuntos de datos se convierte en información para el patriarca o líder de esta aldea. Este caso sencillo lo podemos tomar como ejemplo de los fundamentos de la Administración, si bien este caso para el patriarca sería fácil de resolver, esto se complicaría cuando la aldea empezara a crecer y las familias fueran más a las que se les tuviera que dar de comer.

Pues bien viendo todos estos acontecimientos históricos nos podemos percatar que en la actualidad existe una poderosa herramienta como lo es la computadora en la cual nos podemos apoyar para realizar un sistema de información (SI), en la cual nos va a ayudar a realizar un sistema sistematizado de todos estos datos que usa nuestro comité de agua potable, veremos durante esta documentación que será más rápido, más fácil y más seguro el uso de este SI prototipo para la administración operacional que se lleva actualmente.

En el siguiente documento se pretende desarrollar el prototipo de un SI para automatizar el pago y cobro del servicio de agua potable, con la ayuda de una Interfaz Gráfica de Usuario (GUI) que se implementara con la poderosa plataforma de java, manejando programación orientada a objetos para realizar una estructura correcta de nuestro prototipo y que esta pueda ser una aplicación evolutiva en el cual, como objetivo inicial pretendemos realizar un borrador del sistema para poder ir creciendo conforme a los requerimientos del usuario (administrador / dueño de toma).

A si mismo se desarrollara una base de datos (BDs), con la ayuda del IDE(entorno de desarrollo) de WorkBench, el cual maneja un lenguaje de MySql Server y SQL, para poder tener el registro de todos los usuarios, así como el historial de los mismos, esto para agilizar el pago y cobro del servicio, más adelante veremos las ventajas de una BDs y como nos facilitara las consultas del historial del usuario y el registro de multas y faltas a asambleas que se realizan de manera aleatoria en el mes.



El ¿Por qué realizar este prototipo?, si bien como un ciudadano más de esta comunidad he detectado este problema de ya varios años, en donde la morosidad del usuario por hacer largas filas, y esperar horas para poder realizar un pago puntual de la mensualidad que le correspondería, ha causado un déficit del comité de agua potable , en donde no cuentan con los fondos suficientes para realizar el pago de luz que consumen las bombas para la extracción del agua del subsuelo del mismo pueblo, si es bien cierto que , nosotros no pagamos el agua sino mejor dicho pagamos el consumo de luz por la extracción de este preciado líquido. Llegue a una hipótesis en donde se pretende realizar un prototipo para agilizar y hasta cierto punto automatizar este proceso y evitar el engorroso tiempo de espera para hacer un pago de una mensualidad.

Durante la preparación de este tema, se tomaron en cuenta las anécdotas y diálogos de personas que participaron en el mismo comité en periodos anteriores, así como platicas personales con usuarios que me encontré, cuando yo mismo me vi en la necesidad de hacer este pago del servicio, y observe que la mayoría de usuarios ya estaban familiarizados con las tecnologías de información, haciendo uso de sus celulares, tabletas, gadgets, e incluso que muchos contaban con alguna tarjeta inteligente (crédito, débito, nomina, metro bus, metro), y como futuro ingeniero en computación llegue a la conclusión que con todas estas herramientas que cuento actualmente podría realizar una especie de quiosco o Cajero Automático (ATM) que puede ser implementado para mi SI para la administración del cobro del uso de los pozos de agua potable; caso específico: en el pueblo de San Miguel Tocuila, y realizar una primera versión de prueba para el usuario en donde, sea amigable e interactivo el uso, realizando una GUI touch-creen y realizar credenciales de identificación que con el uso de estas tarjetas de identificación puedan acezar y navegar en él.



II. Planteamiento del problema

Existe una población llamada San Miguel Tocuila localizada en el municipio de Texcoco del Estado de México, en donde desde sus principios fue fundada como un pueblo con gobierno autónomo mejor conocido como “usos y costumbre”, a que se refiere con esto ¿Que se gobierna solo? , falso, debido a que como un pueblo y parte de un municipio que está en un estado y este pertenece a los Estados Unidos Mexicanos, este se rige a través de una constitución política en la cual si se rompe alguna ley serán sancionados con alguna multa o cárcel.

Entonces usos y costumbres, es la toma de decisiones que afecten al pueblo, como en este caso es el uso racional y la administración de agua potable, que desde sus inicios Tocuila ha pertenecido al ya extinto lago de Texcoco en donde existía mucha zona fangosa y abundaba el agua, debido a esto el agua potable no es proporcionada por el municipio de Texcoco, vemos que el pueblo de Tocuila realizó pozos profundos para la extracción de agua potable para el consumo de sus habitantes, empezando con un solo pozo para el consumo de unos pocos habitantes y teniendo una administración muy básica llevando el registro en libros de todos los movimientos de este comité.

Todo esto parece ser funcional hasta cierto punto, hoy en día este pueblo ha dejado de ser un pueblo rural para dar paso a la urbanización y colonización de más barrios dentro del mismo pueblo, contando hoy en día con más de 25 000 habitantes y 5 pozos , entonces para realizar los pagos puntuales de este servicio vemos que el usuario ha llegado a ser moroso en sus pagos, pero teniendo hasta cierto punto justificación, debido a que es muy poco el tiempo que se le da para realizar este pago, y muy largas las filas para efectuarlo, teniendo una sobrecarga de trabajo para el comité de agua potable, revisando libro por libro todo el historial del usuario que pretende ponerse al corriente de sus pagos, y hacer la actualización en todos estos libros de que el usuario ya ha realizado sus pagos. Con esto detectamos que la parte de la administración del comité ha dejado de ser funcional y necesita de una mejora en su SI para agilizar este proceso.



III. Objetivos

Objetivo general

Demostrar que la implementación de un sistema de información para la administración del cobro, en el uso de los pozos de agua potable en el pueblo de San Miguel Tocuila facilitaría que los usuarios puedan realizar sus pagos de una forma más rápida y eficiente.

Objetivos específicos

- Obtener los requerimientos que se manejan para la administración del sistema actual para el cobro del agua potable.
- Desarrollar una base de datos digital con la ayuda de los sistemas gestores de base de datos (MySQL WorkBench) para complementar el sistema.
- Diseñar una interfaz gráfica con la ayuda del lenguaje de programación java, (Netbeans) para facilitar el manejo del sistema a los usuarios.
- Comprobar el desempeño del sistema.



IV. Justificación

La propuesta del uso de un sistema de información para el cobro del uso de los pozos de agua potable, permitiría un mayor interés de los usuarios por hacer este pago, porque ya no existirían esas largas colas interminables y tediosas y que al final de cuentas no llegasen a pasar por el tiempo de cobro tan limitado y el sistema de información tan ineficiente. Al implementarse un sistema de información (SI) estratégico ayudaría a agilizar este proceso y hacerlo más funcional y eficiente, se lograría que los usuarios no perdieran más su tiempo y que existieran dudas en cuanto a ¿Qué pasa con todo el dinero?, ¿Si los del comité del agua potable se roban el dinero?, ¿Si en verdad se paga o no la luz de los pozos?, ¿Quiénes son en verdad los que siempre pagan?, ¿Quiénes no pagan? , etc.

La implementación de este SI será muy útil , porque como ya se ha visto en las grandes y medianas empresas donde ya están implementados, en escuelas, en el gobierno incluso en los municipios y por qué no implementarlo en un pueblo en donde su población ha crecido y seguirá creciendo, la utilidad de este sistema no solo sería para que funcione por un periodo corto, más bien por un largo tiempo que ayudaría tanto a un comité, como los usuarios a administrar y realizar sus pagos puntualmente y sin contratiempos.



CAPÍTULO 1. Sistemas de información Y RAD

1.1. Concepto de un sistema de información

Los sistemas de información (SI) se pueden definir como un conjunto de componentes interrelacionados que almacenan, procesan y distribuyen información como apoyo en la toma de decisiones y el control de una organización, estos contienen información variada tal como datos sobre personas, lugares y cosas de interés para la organización o el entorno en el cual se desenvuelve (Laudon y Laudon, 2004).

1.1.1 Que es un sistema

Existen numerosas definiciones de sistema. Una definición genérica de sistema es la proporcionada por:

“Un sistema puede ser definido como un complejo de elementos interactuantes”

Otra definición de sistema, extraída de (Mcleod, 2000), es la que sigue:

“Un sistema es un grupo de elementos que se integran con el propósito común de lograr un objetivo”.

O bien (O’Brien, 1993) formula el concepto así:

“Grupo de componentes interrelacionados que trabajan juntos hacia un fin común aceptando inputs y produciendo outputs en un proceso de transformación organizado”.

En (Murdick Robert, 1988), el autor propone la siguiente definición:

“ Conjunto de elementos organizados que se encuentran en interacción, que buscan alguna meta o metas comunes, operando para ello sobre datos o información sobre energía o materia u organismos en una referencia temporal para producir como salida información, energía, materia u organismos”



Si bien existen numerosas definiciones de que es un sistema podemos concluir que todos estos conceptos son similares y podemos obtener un resumen de todos ellos diciendo que; un sistema es un conjunto de elementos organizados que interactúan. La interacción entre los elementos es vital para que un conjunto de elementos se pueda considerar un sistema. Un conjunto sin interacción entre ellos no puede ser considerado como un sistema, al menos a nuestros efectos. Un sistema es, pues, más que la simple suma de sus partes. Un sistema tiene metas como objetivos. Los sistemas tienen entradas y salidas. Los sistemas toman entradas, las procesan y generan una o varias salidas.

1.1.2. Que es información

Se entiende por información a los datos que han sido modelados de manera significativa para el entendimiento y uso del ser humano, en tanto que un dato son hechos o eventos dentro de una organización o en un entorno físico que aún no han sido modelados para que sean útiles a las personas (Laudon y Laudon, 2004).

Existen otras definiciones como:

- Es un conjunto organizado de datos, que constituye un mensaje sobre un cierto fenómeno o ente.
- Es un fenómeno que proporciona significado o sentido a las cosas, e indica mediante códigos y conjuntos de datos, los modelos del pensamiento humano.

Entonces tenemos que un sistema de información es:

- Es un conjunto de elementos que interactúan entre sí con el fin de apoyar las actividades de una empresa o negocio.
- Es un conjunto organizado de elementos que interactúan entre sí para procesar los datos y la información (incluyendo procesos manuales y automáticos) y distribuirla de la manera más adecuada posible en una determinada organización en función de sus objetivos.

1.2. Tipos de sistemas de información

Un aspecto que debe ser señalado, es que un sistema de información puede ser manual o puede ser automatizado. Existe la creencia generalizada, errónea por cierto, que un Sistema de Información debe ser automatizado. En la actualidad aún existen sistemas de información que se llevan de manera manual, y que aún son eficientes y que no tienen necesidad de ser automatizados, ya sea por el costo que representa o por la sencillez del mismo. Como sistema que es, un SI presenta el mismo modelo de Entrada-Proceso-Salida, Figura 1:

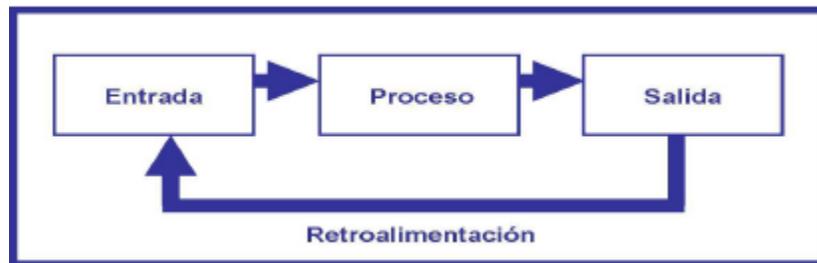


Figura 1. Modelo de un sistema

Un sistema de información presenta las siguientes características: generalidad, simplicidad, continuidad, consistencia, flexibilidad y dinamismo.

Algunos tipos de sistemas de información:

- Sistemas transaccionales: automatizan tareas operativas de la organización.
- Sistemas de apoyo de las decisiones: brindan información que sirve de apoyo a los mandos intermedios y a la alta administración en el proceso de toma de decisiones.
- Sistemas estratégicos: generan ventajas que los competidores no posean, tales como ventajas en costos y servicios diferenciados con clientes y proveedores.
- Sistema planificación de recursos (ERP – Enterprise Resource Planning): integran la información y los procesos de una organización en un solo sistema.

Los tipos de metodologías de desarrollo de sistemas son, Figura 2 (Luis Castellanos, 2011):

- Se maneja como proyecto
- Gran volumen de datos y transacciones
- Abarca varias áreas organizativas de la empresa
- Tiempo de desarrollo largo
- Requiere que se cumplan todas las etapas, para poder cumplir las siguientes (progresión lineal y secuencial de una fase a la otra)

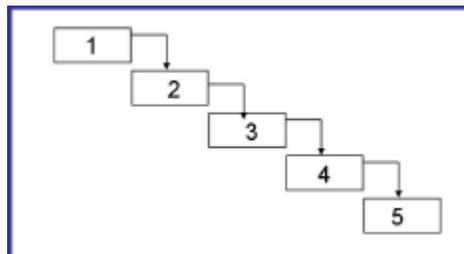


Figura 2. Esquema de una metodología clásica de desarrollo de sistemas (Luis Castellanos, 2011).

Evolutiva-Incremental, Figura 3 (Luis Castellanos, 2011):

- Se deriva de la estructurada
- Permite seguir secuencias ascendentes o descendentes en las etapas del desarrollo
- Permite cumplir etapas o fases en paralelo, por lo que es más flexible que la estructurada

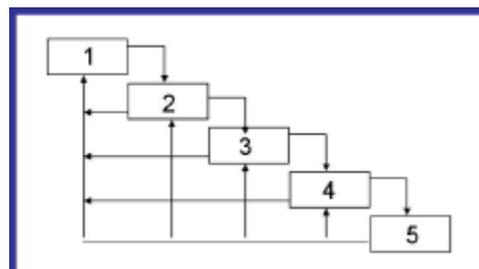


Figura 3, Esquema de una metodología evolutiva incremental de desarrollo de sistemas (Luis Castellanos, 2011).

1.3. Objetivo de un sistema de información

Los objetivos que persigue un sistema de información son:

- Automatizar los procesos operativos.
- Proporcionar información que sirva de apoyo al proceso de toma de decisiones.
- Lograr ventajas competitivas a través de su implantación y uso.



Prototipos

- Desarrolla modelo en escala del sistema propuesto (sistema objetivo), en otras palabras, desarrolla un “demo”, que de ser aprobado por el usuario, se culmina con todas las funcionalidades.
- Se prueba y refina hasta que usuarios quedan conformes
- Identifica necesidades de información
- Maneja principales procedimientos orientados a transacciones
- Produce informes críticos
- Permite consultas rudimentarias
- Complementa especificaciones imprecisas y/o incompletas de los usuarios

Orientada a Objetos

- No modela la realidad, sino la forma en que las personas comprenden y procesan la realidad
- Es un proceso ascendente basado en una abstracción de clases en aumento
- Se basa en identificación de objetos, definición y organización de librerías de clases, y creación de macros para aplicaciones específicas
- Utiliza menor cantidad de código
- Es más reutilizable

1.1 . Modelo del RAD

El desarrollo rápido de aplicaciones o RAD (Rapid Application Development) es un proceso de desarrollo de software, desarrollado inicialmente por James Martin en 1980. El método comprende el desarrollo iterativo, la construcción de prototipos y el uso de utilidades CASE. Tradicionalmente, el desarrollo rápido de aplicaciones tiende a englobar también la usabilidad, utilidad y la rapidez de ejecución.

El desarrollo rápido de aplicaciones, es un modelo de proceso del desarrollo del software lineal secuencial que enfatiza un ciclo de desarrollo extremadamente corto. RAD es una adaptación a "alta velocidad" en el que se logra el desarrollo rápido utilizando un enfoque de construcción basado en componentes. Si se comprenden bien los requisitos y se limita el ámbito del proyecto, el proceso RAD permite al equipo de desarrollo crear un "sistema completamente funcional" dentro de periodos cortos de tiempo. Cuando se utiliza principalmente para aplicaciones de sistemas de información, el enfoque RAD comprende las siguientes fases, Figura 4:



- **Modelado de gestión:** El flujo de información entre las funciones de gestión se modela de forma que responda a las siguientes preguntas: ¿Qué información conduce el proceso de gestión? ¿Qué información se genera? ¿Quién la genera? ¿A dónde va la información? ¿Quién la proceso?
- **Modelado de datos:** El flujo de información definido como parte de la fase de modelado de gestión se refina como un conjunto de objetos de datos necesarios para apoyar la empresa. Se definen las características (llamadas atributos) de cada uno de los objetos y las relaciones entre estos objetos.
- **Modelado de proceso:** Los objetos de datos definidos en la fase de modelado de datos quedan transformados para lograr el flujo de información necesario para implementar una función de gestión. Las descripciones del proceso se crean para añadir, modificar, suprimir, o recuperar un objeto de datos. Es la comunicación entre los objetos.
- **Generación de aplicaciones:** El RAD asume la utilización de técnicas de cuarta generación. En lugar de crear software con lenguajes de programación de tercera generación, el proceso RAD trabaja para volver a utilizar componentes de programas ya existentes (cuando es posible) o a crear componentes reutilizables (cuando sea necesario). En todos los casos se utilizan herramientas automáticas para facilitar la construcción del software.
- **Pruebas de entrega:** Como el proceso RAD enfatiza la reutilización, ya se han comprobado muchos de los componentes de los programas. Esto reduce tiempo de pruebas. Sin embargo, se deben probar todos los componentes nuevos y se deben ejercitar todas las interfaces a fondo.

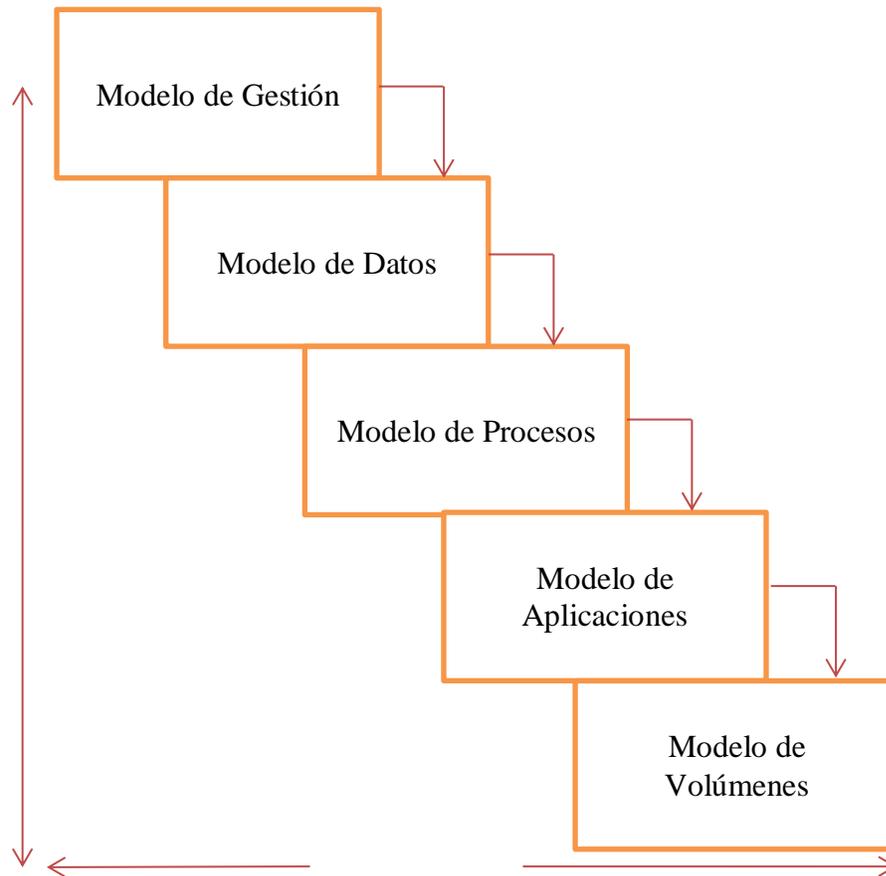


Figura 4. Modelo RAD de 60 a 90 Días.

Obviamente la limitación de tiempo impuesto en un proyecto RAD demanda "ámbito en escalas". Si una aplicación de gestión puede modularse se forma que permita completarse cada una de las funciones principales en menos de tres meses (utilizando el enfoque descrito anteriormente), es un candidato del RAD. Cada una de las funciones puede ser afrontada por un equipo RAD diferente y ser integradas en un solo conjunto.

Al igual que todos los modelos de proceso, el enfoque RAD tiene inconvenientes:

- Para proyectos grandes aunque por escalas, el RAD requiere recursos humanos suficientes como para crear el número correcto de equipos RAD.
- RAD requiere clientes y desarrolladores comprometidos en las rápidas actividades necesarias para completar un sistema en un marco de tiempo abreviado. Si no hay compromiso, por ninguna de las partes constituyentes, los proyectos RAD fracasaran.



No todos los tipos de aplicaciones son apropiados para RAD. Si un sistema no se puede modular adecuadamente. La construcción de los componentes necesarios para RAD será problemático. Si está en juego el alto rendimiento, y se va a conseguir el rendimiento convirtiendo interfaces en componentes de sistema, el enfoque RAD puede que no funcione. RAD no es adecuado cuando los riesgos técnicos son altos. Esto ocurre cuando una nueva aplicación hace uso de tecnologías nuevas, o cuando el nuevo software requiere un alto grado de interoperabilidad con programas de computadora ya existentes.

RAD enfatiza el desarrollo de componentes de programas reutilizables. La reutilización es la piedra angular de las tecnologías de objetos, y se encuentra en el modelo de proceso de ensamblaje.

1.4.1. Características del RAD

Entre las principales características del RAD tenemos:

1. Equipos híbridos

- Equipos compuestos por alrededor de seis personas, incluyendo desarrolladores y usuarios de tiempo completo del sistema así como aquellas personas involucradas con los requisitos.
- Los desarrolladores de RAD deben ser "renacentistas": analistas, diseñadores y programadores en uno.

2. Herramientas especializadas

- Desarrollo "visual"
- Creación de prototipos falsos (simulación pura)
- Creación de prototipos funcionales
- Múltiples lenguajes
- Calendario grupal
- Herramientas colaborativas y de trabajo en equipo
- Componentes reusables
- Interfaces estándares (API)
- Control de versiones

3. "Timeboxing"

4. Las funciones secundarias son eliminadas como sea necesario para cumplir con el calendario.

- Prototipos iterativos y evolucionarios
- Reunion JAD (Joint Application Development):
- Se reúnen los usuarios finales y los desarrolladores.
- Lluvia de ideas para obtener un borrador inicial de los requisitos.



Iterar hasta acabar:

- Los desarrolladores construyen y depuran el prototipo basado en los requisitos actuales.
- Los diseñadores revisan el prototipo.
- Los clientes prueban el prototipo, depuran los requisitos.
- Los clientes y desarrolladores se reúnen para revisar juntos el producto, refinar los requisitos y generar solicitudes de cambios.
- Los cambios para los que no hay tiempo no se realizan. Los requisitos secundarios se eliminan si es necesario para cumplir el calendario.
- Notas:

Cada iteración dura entre un día y tres semanas. Reuniones de 2 horas con facilitador que mantiene enfocado al grupo.

RAD tiende a funcionar cuando:

- La aplicación funcionará de manera independiente.
- Se pueden usar mayormente bibliotecas existentes.
- Desempeño no crítico.
- Distribución limitada, interna o vertical.
- Alcance del proyecto limitado.
- Confiabilidad no crítica.
- El sistema puede dividirse en muchos módulos independientes.
- El producto está dirigido a un mercado altamente especializado.
- El proyecto cuenta con fuertes limitantes de tiempos parciales (time boxes).
- La tecnología requerida tiene más de un año en el mercado.

RAD tiende a fallar cuando:

- La aplicación debe interpolar con sistemas existentes.
- Existen pocos componentes reutilizables.
- Alto desempeño crítico.
- El desarrollo no puede aprovechar herramientas de alto nivel.
- Distribución amplia, horizontal o masiva.
- RAD se convierta en QADAD (Quick and Dirty Application Development).
- Métodos RAD para desarrollar sistemas operativos (confiabilidad demasiado alta) o juegos (desempeño demasiado alto).
- Riesgos técnicos de tecnología de punta.



1.4.2. Ventajas y desventajas del RAD

Ventajas de RAD

- Comprar puede ahorrar dinero en comparación con construir.
- Los entregables pueden ser fácilmente trasladados a otra plataforma.
- El desarrollo se realiza a un nivel de abstracción mayor.
- Visibilidad temprana.
- Mayor flexibilidad.
- Menor codificación manual.
- Mayor involucramiento de los usuarios.
- Posiblemente menos fallas.
- Posiblemente menor costo.
- Ciclos de desarrollo más pequeños.
- Interfaz gráfica estándar.

Desventajas de RAD

- Comprar puede ser más caro que construir.
- Costo de herramientas integradas y equipo necesario.
- Progreso más difícil de medir.
- Menos eficiente.
- Menor precisión científica.
- Riesgo de revertirse a las prácticas sin control de antaño.
- Más fallas (por síndrome de "codificar a lo bestia").



CAPÍTULO 2. Análisis y diseño de sistemas de información

2.1. Análisis de requerimientos

En esta sección se mostrará el análisis del proyecto en cuanto a cuatro factores: análisis de contenido, análisis de interacción, análisis de funciones y un análisis de configuración. El modelado abarcará casos de uso, diagramas de actividades y otros elementos gráficos que ayuden a la comprensión del funcionamiento de la aplicación. Se comenzará con la identificación de los usuarios principales que serán los actores que participarán en todo el proceso y hasta finalizar con el sistema de información.

2.1.2 Requerimientos funcionales

Basándonos en el sistema que está implementando actualmente, dado el funcionamiento lento y problemático del sistema actual se genera la necesidad de analizar un nuevo sistema que se empezara a desarrollar empezando por identificar los nuevos requerimientos funcionales que necesitamos en nuestro nuevo sistema, los requerimientos funcionales son:

Declaraciones de los servicios que debe proporcionar el sistema, de la manera en que éste debe reaccionar a entradas singulares y de cómo se debe comportar en situaciones particulares. En algunos casos, los requerimientos funcionales de los sistemas también pueden declarar explícitamente lo que el sistema no debe hacer. Los requerimientos funcionales de un sistema describen lo que el sistema debe hacer.

Estos requerimientos dependen del tipo de software que se desarrolle, de los posibles usuarios del software y del enfoque general tomado por la organización al redactar requerimientos. Cuando se expresan como requerimientos del usuario, habitualmente se describen de una forma bastante abstracta. Sin embargo los requerimientos funcionales del sistema describen con detalle la función de éste, sus entradas y salidas, excepciones, etcétera. Los requerimientos funcionales para un sistema software se pueden expresar de diferentes formas. A continuación se presentan:



a) Requerimientos del sistema enfocados al usuario.

Tabla 1. Requerimientos del usuario

Usuario de toma de agua	
Identificador	Requerimientos funcionales
rq1	Leer tarjeta inteligente
rq2	Coloca su clave única
rq3	Muestra menú de opciones
rq4	Consultar adeudo, mostrando sus atrasos y asistencias a las asambleas
rq5	Imprimir Comprobante de Pago
rq6	Muestra menú de Salir o Cerrar sesión
rq7	Retira tarjeta

b) Requerimientos del sistema enfocados al administrador

Tabla 2. Requerimientos del administrador.

Administrador , Comité del agua potable	
Identificador	Requerimientos Funcionales
rq1	Leer tarjeta inteligente
rq2	Coloca su clave única
rq3	Genera Reporte mensual de Pagos e ingresos
rq4	Imprime comprobante
rq5	Muestra menú de Salir o Cerrar sesión
rq6	Retira tarjeta

c) Requerimientos del sistema para el ingeniero

Tabla 3. Requerimientos del ingeniero.

Ingeniero	
Identificador	Requerimientos funcionales
rq1	Inicia las interfaces graficas
rq2	Mantenimiento al sistema o la maquina
rq3	Soporte al desarrollo de nuevos requerimientos al sistema

Estos serían los principales requerimientos que necesitaran nuestro nuevo SI, y con esto nos damos cuenta de las necesidades y que es lo que va a realizar el sistema, pero también tenemos que tomar en cuenta los requerimientos no funcionales estos son:



2.1.2. Requerimientos no funcionales

Restricciones de los servicios o funciones ofrecidos por el sistema. Incluyen restricciones de tiempo, sobre el proceso de desarrollo y estándares. Los requerimientos no funcionales a menudo se aplican al sistema en su totalidad. Normalmente apenas se aplican a características o servicios individuales del sistema. Los requerimientos no funcionales, como su nombre sugiere, son aquellos requerimientos que no se refieren directamente a las funciones específicas que proporciona el sistema, sino a las propiedades emergentes de éste como la fiabilidad, el tiempo de respuesta y la capacidad de almacenamiento.

Tabla 4. Requerimientos no funcionales

Requerimientos NO Funcionales	Descripción
Usabilidad	El sistema será interactivo, rápido y eficiente, haciendo confortable el uso y la navegación, el sistema de información será implementado en forma de cajero automático con pantalla touch screen.
Seguridad	La información de los usuarios será confidencial, deben de identificarse plenamente para poder hacer uso del sistema.
Desempeño	El sistema será almacenado en un equipo con el solo propósito de procesar el sistema de información desempeñando un buen rendimiento.
Portabilidad	El sistema que se va a desarrollar será con plataforma de java por la cual al finalizar con las pruebas correspondientes se generara el ejecutable .jar, haciendo esto portable el SI.

2.2. Identificación de las clases de usuarios

Basándonos en los problemas que se encuentran actualmente en el sistema de cobro, que lleva a cabo el comité de agua potable, encontramos que los dueños de agua potable deben de hacer una larga fila para realizar su pago y cuando toca su turno esperar otro considerado tiempo para que encuentren su historial y así hacer el total de su adeudo.



Pues viendo esto vamos a realizar el sistema para el usuario/administrador y este sistema debe ser totalmente interactivo en el cual el usuario u administrador no necesariamente deba contar con conocimientos altos en el manejo de bases de datos, computación, o que realice cálculos de todos sus atrasos, entonces un usuario en nuestro sistema será capaz de realizar, Figura 5:

- Ingresar al sistema como usuario/ administrador.
- Consultar todos sus atrasos mensuales, así como sus pagos realizados
- Consultar sus faltas a asambleas realizadas en el mes que no a bonificado
- Consultar el total de su adeudo.
- Realizar el pago total.
- Realizar la impresión de su pago y comprobante.
- En el caso de la cuenta privilegiada (administrador), será capaz de realizar ABC (altas, eliminar, modificar o actualizar) del usuario o administrador.

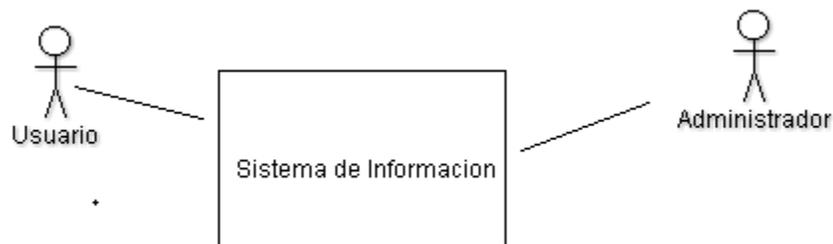


Figura 5. Actores del sistema de información.

2.2.1. Definición de los actores para generar casos de uso

Los actores son usuarios o elementos externos que tienen la necesidad de interactuar con el sistema con el fin de intercambiar información (Whitten y Bentley, 2008), los actores de este sistema son los mencionados en la sección anterior, los siguientes son la descripción de cada uno conforme a la interacción que tendrán con el sistema:

- a) Usuario(dueño de la toma de agua); el usuario tendrá acceso al sistema a una sección exclusiva en donde solo podrá realizar consultas e imprimir sus comprobantes de adeudo, se estima que cumpla con un perfil mínimo:

Perfil:

- Saber leer.
- Interesado en consultar su cuenta
- Interesado en realizar su pago o adelantarlos.
- Poder tocar una pantalla de cristal con el dedo.



- b) Administrador (comité de agua potable); el administrador será capaz de tener acceso al sistema, en forma privilegiada y poder realizar alta, baja o consulta de los usuarios (dueños), generar los reportes correspondientes para el uso de la administración del comité del agua potable, reportar los pagos de los usuarios, levantar y registrar asambleas:

Perfil:

- Saber leer.
- Ser parte del comité
- Conocimientos de las interfaces y navegación del sistema

2.2.2. Identificación de los casos de uso

Una vez identificado los actores del sistema se procede a realizar casos de uso para percibir cómo ve el usuario final la interacción de sí mismo con el sistema. Los siguientes puntos son un panorama general de cómo el usuario percibe la interacción con el sistema:

a) Casos de uso para el usuario.

- Ingresar al sistema como usuario.
- Consultar todos sus atrasos mensuales, así como sus pagos realizados
- Consultar sus faltas a asambleas del mes que no a bonificado
- Consultar el total de su adeudo.
- Realizar la impresión de su comprobante de adeudos.

b) Casos de uso para el administrador.

- Ingresar al sistema
- Generar reportes mensuales
- Generar reportes anuales
- Generar reportes de deudores.
- Imprimir comprobantes de los movimientos realizados durante su sesión.
- Iniciar sesión de asamblea
- Registrar la entrada de los usuarios a la asamblea.
- Dar de alta, eliminar o modificar el registro de un usuario.
- Dar de alta, eliminar o modificar el registro de un administrador.



2.2.3. Diagramas de casos de uso

Los diagramas de casos de uso son una secuencia de pasos relacionados que tiene como propósito completar una sola tarea del negocio (Whitten y Bentley, 2008).

El siguiente formato (Tabla 5) será el que se utilizara para representar los casos de uso, estos diagramas son una variación de los propuestos por (Whitten y Bentley, 2008).

Tabla 5. Descripción del formato para los casos de uso.

Nombre del caso de uso:	Nombre de un caso de uso específico.
ID del caso de uso:	Es un identificador único para cada caso de uso.
Fuente:	Define la entidad que origino la creación del caso de uso. Por ejemplo un documento, un requerimiento o un involucrado.
Actor Primario:	Es el principal beneficiado del manejo del caso de uso.
Descripción:	Es el resumen que define el propósito del caso de uso.
Precondición:	Restricción del estado del sistema antes de la ejecución del caso de uso.
Disparador:	Es el evento que genera la ejecución del caso de uso.
Curso típico de eventos:	Es la secuencia normal de actividades de actor y sistema para cumplir el objetivo del caso de uso.
Cursos alternos:	Documentan los comportamientos atípicos del caso de uso.
Conclusión:	Establece cuando termina la ejecución de un caso de uso.

En las Figuras 6 y 7 se muestran los diagramas de casos de uso para visitantes y administradores respectivamente, los actores son representados como individuos y los casos de uso como óvalos, estos casos de uso pueden ser creados por individuos (flechas continuas) o ser dependientes de otros casos de uso (flechas).

A continuación, se muestran estos diagramas:



Diagrama de casos de uso para usuarios, (Figura 6).

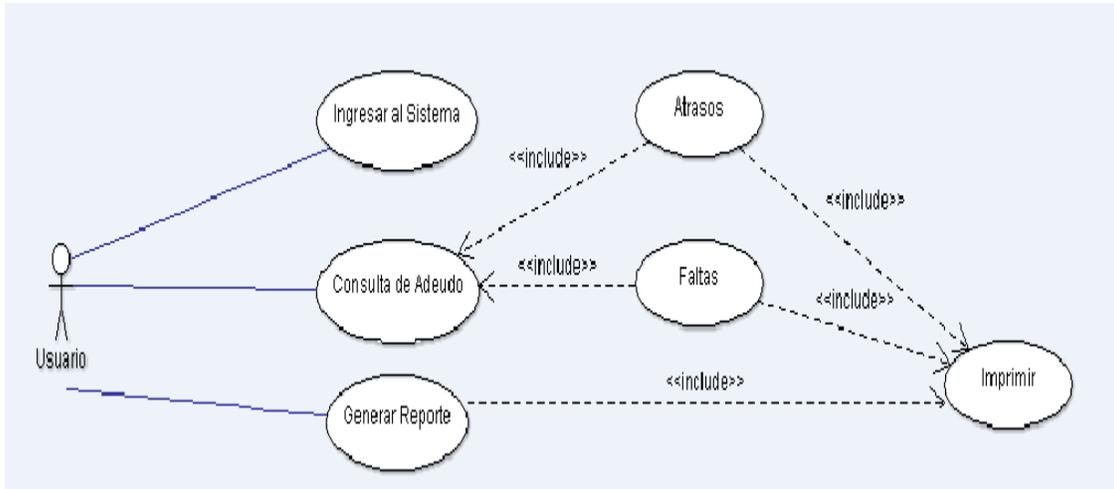
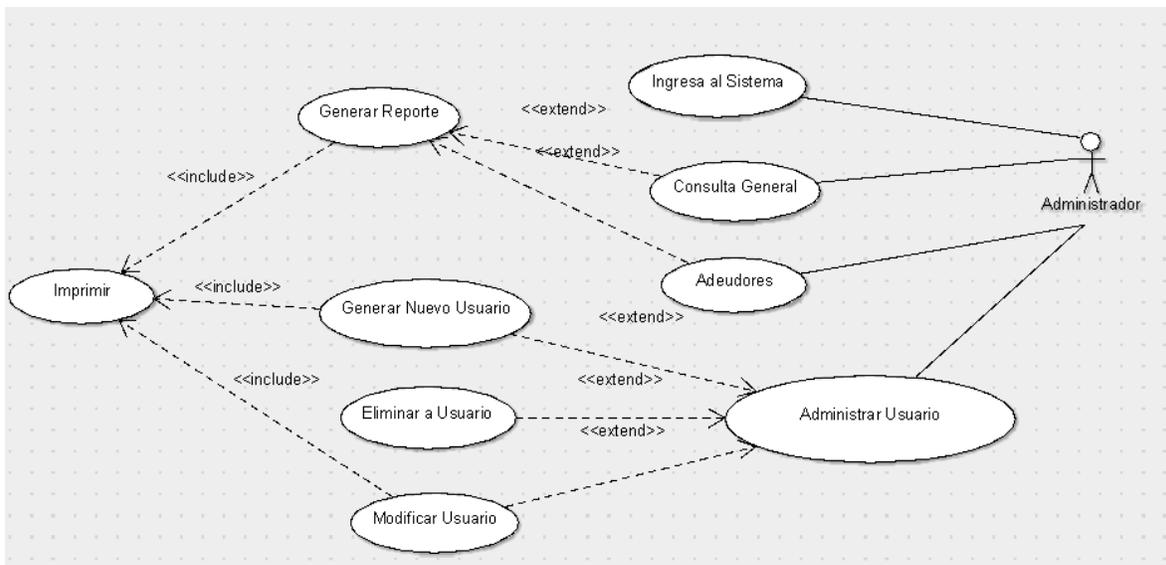


Figura 6. Diagrama de casos de uso para usuarios.

Diagrama de casos de uso para administradores, (Figura 7).



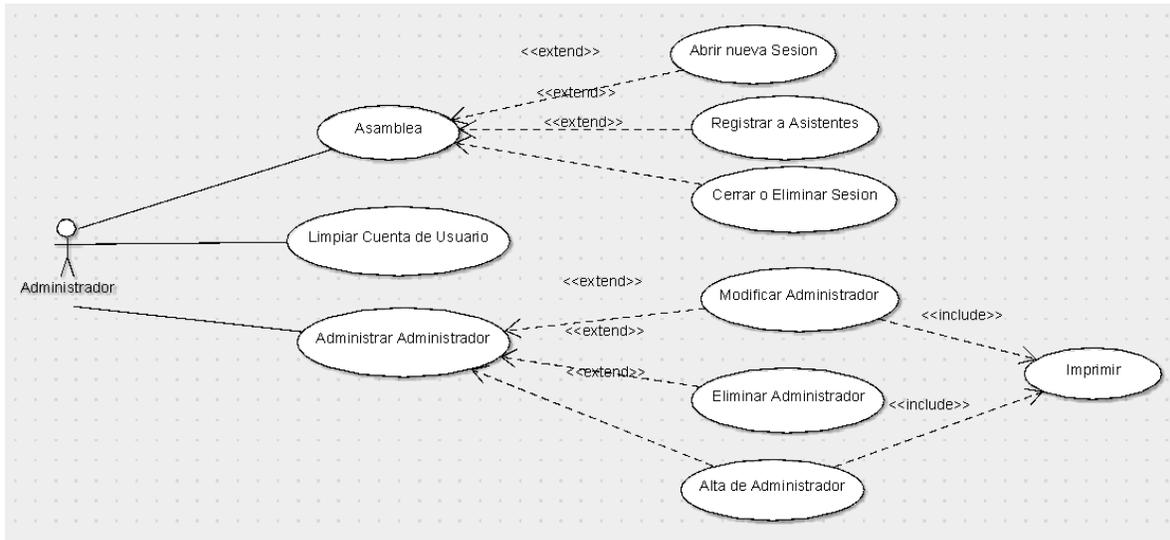


Figura 7. Diagrama de casos de uso para administradores.



2.2.4. Descripción de los casos de uso del usuario.

A continuación se describen los casos de uso del usuario.

Tabla 6. Caso de uso “ingresando al sistema”.

Nombre del caso de uso:	Ingresar al Sistema	
Id del caso de uso:	CU1	
Fuente:	APG	
Actor primario:	Usuario	
Descripción:	El usuario ingresa la credencial con código de barras y es activado el menú principal para el usuario.	
Precondición:	Estar en el menú de inicio	
Disparador:	Activar el botón de acceso	
Curso típico de eventos:	Acciones del actor	Respuesta del sistema
	1.- Activa el botón de login al sistema 2.- Pone en el lector la tarjeta inteligente.	3.- Reconoce la tarjeta. 4.- Busca en la base de datos el ID. 5.- Se le da acceso al sistema de usuario
Cursos alternos:	Tarjeta invalida solicitar una nueva en las oficinas, contraseña incorrecta	
Conclusión:	Acceso de sesión de usuario.	



Tabla 7. Caso de uso “consulta de adeudos”

Nombre del caso de uso:	Consulta de adeudos	
Id del caso de uso:	CU2	
Fuente:	APG.	
Actor primario:	Usuario	
Descripción:	Se muestra su saldo total de adeudo, en donde puede consultar, sus atrasos en los pagos del agua y las inasistencias por no estar presente en las juntas del comité del agua potable.	
Precondición:	Se muestra un menú de inicio	
Disparador:	Debió elegir la opción de Consulta	
Curso típico de eventos:	Acciones del actor	Respuesta del sistema
	1.- El usuario entra al menú de consulta.	2.- El sistema se conecta a la base de datos. 3.- El sistema suma el adeudo por atrasos y faltas. 4.- Se muestra el total de adeudo del usuario que está consultando.
Cursos alternos:	No hay adeudos, regresar al menú de inicio.	
Conclusión:	El usuario está comprobando su adeudo.	



Tabla 8. Caso de Uso "Atrasos"

Nombre del caso de uso:	Atrasos	
Id del caso de uso:	CU3	
Fuente:	APG	
Actor primario:	Usuario	
Descripción:	Se visualizan los pagos que no ha realizado y puede el usuario verificar su adeudo parcial, ya que aún faltan sus faltas de juntas, y puede regresar al menú inicial.	
Precondición:	El usuario debió elegir atrasos en el menú de consulta	
Disparador:	Elegir la opción de atrasos	
Curso típico de eventos:	Acciones del actor	Respuesta del sistema
	1.- El usuario elije la opción de atrasos. 4.- El usuario retrocede al menú de consulta	2.- El sistema hace conexión con la base de datos existente. 3.- Se muestran adeudos por año y mes en una nueva ventana y el total parcial de adeudos.
Cursos alternos:	No tienes atrasos de pagos	
Conclusión:	El usuario está comprobando su adeudo.	



Tabla 9. Caso de Uso “Faltas”

Nombre del caso de uso:	Faltas	
Id del caso de uso:	CU4	
Fuente:	APG	
Actor primario:	Usuario	
Descripción:	Se muestran las faltas y asistencias totales de las juntas organizadas por el comité de agua potable, el usuario puede ver el total de las juntas que se llevaron a cabo por día, mes y año	
Precondición:	El usuario está visualizando el menú de Consulta	
Disparador:	Se elige la opción de inasistencias	
Curso típico de eventos:	Acciones del Actor	Respuesta del Sistema
	1.- El usuario elije la opción de faltas.	2.- El sistema se conecta a la base de datos. 3.-Consulta las faltas del usuario. 4.- Se muestran las asistencias y faltas totales hasta ese día en forma de lista y la multa total.
Cursos alternos:	Uds. cuenta con todas las asistencias de las juntas	
Conclusión:	El usuario está comprobando su adeudo.	



Tabla 10. Caso de Uso “Generar Reporte”

Nombre del caso de uso:	Generar Reporte	
Id del caso de uso:	CU5	
Fuente:	APG	
Actor primario:	Usuario	
Descripción:	El usuario Puede visualizar en un solo reporte sus atrasos y faltas así como el monto total de su deuda.	
Precondición:	Estar en el menú de consulta	
Disparador:	Elegir la opción de generar reporte	
Curso típico de eventos:	Acciones del Actor	Respuesta del Sistema
	1.- El usuario presiona el botón de reporte.	2.-El sistema se conecta a la base de datos. 3.- El sistema lanza el reporte a través de nuestra herramienta iReport 4.- Se muestra el total de adeudo del usuario que está consultando. 5.- Se muestra en pantalla el reporte con la opción de poder imprimirlo
Cursos alternos:	Uds. Cuenta con todos sus pagos	
Conclusión:	El usuario es capaz de ver su monto total.	



Tabla 11. Caso de Uso “Imprime”

Nombre del caso de uso:	Imprime	
Id del caso de uso:	CU6	
Fuente:	APG	
Actor primario:	Usuario, Administrador	
Descripción:	El sistema imprime los comprobantes para el usuario en el caso que vaya a ser algún pago.	
Precondición:	Estar en las pantallas de consulta de atrasos, faltas o generar reporte.	
Disparador:	Dar la opción de imprimir	
Curso típico de eventos:	Acciones del Actor	Respuesta del Sistema
	1.-Estar en la pantalla con la consulta generada. 2.-Elegir la opción de imprimir.	3.- El sistema captura la ventana principal. 4.- Guarda la captura y la envía a la configuración de impresora para su impresión.
Cursos alternos:	Conecte la impresora	
Conclusión:	El usuario cuenta con un comprobante para validar sus movimientos dentro del sistema.	



2.2.5. Descripción de los casos de uso del Administrador

A continuación se describen los casos de uso del administrador.

Tabla 12. Caso de uso “ingresar al sistema”.

Nombre del caso de uso:	Ingresar al Sistema	
Id del caso de uso:	CU7	
Fuente:	APG	
Actor primario:	Administrador	
Descripción:	El usuario ingresa la credencial con código de barras y es activado el menú principal con privilegios de Administrador.	
Precondición:	Estar en el Menú de inicio	
Disparador:	Activar el botón de Acceso	
Curso típico de eventos:	Acciones del Actor	Respuesta del Sistema
	1.- Activa el botón de login al Sistema 2.- Pone en el lector la tarjeta inteligente.	3.- Reconoce la tarjeta. 4.- Busca en la base de datos el ID. 5.- Se le da acceso al sistema de usuario
Cursos alternos:	Tarjeta invalida solicitar una nueva en las oficinas, contraseña incorrecta	
Conclusión:	Acceso modo privilegiado.	



Tabla 13. Caso de Uso “Consulta General.”

Nombre del caso de uso:	Consulta General	
Id del caso de uso:	CU8	
Fuente:	APG	
Actor primario:	Administrador	
Descripción:	El administrador desea saber cuántos son los usuarios que aún no han pagado y los que ya realizaron sus pagos puntuales.	
Precondición:	Estar en el menú principal	
Disparador:	Elegir en el menú de administrador	
Curso típico de eventos:	Acciones del Actor	Respuesta del Sistema
	1.- Dar generar reporte general	2.- El sistema recibe la petición. 3.- Hace la conexión a la base de datos. 4.- Visualiza al administrador los resultados.
Cursos alternos:		
Conclusión:	El administrador será capaz de ver el estado de pagos en general.	



Tabla 14. Caso de Uso “Adeudo”

Nombre del caso de uso:	Adeudo	
Id del caso de uso:	CU9	
Fuente:	APG	
Actor primario:	Administrador	
Descripción:	El administrador consulta un usuario en específico.	
Precondición:	Estar en el menú de administrador	
Disparador:	Elegir la opción de Generar consulta Adeudo	
Curso típico de eventos:	Acciones del Actor	Respuesta del Sistema
	1.- El administrador selecciona ingreso mensual	2.- El sistema hace conexión con la base de datos. 3.-Suma todos los pagos realizados en el mes seleccionado. 4.- La suma total lo muestra en una ventana. 5.- Continúa la ejecución del sistema
Cursos alternos:	No hay cursos alternos	
Conclusión:	El generar reportes mensuales permite a la tesorería saber cuál es su presupuesto.	



Tabla 15. Caso de Uso “Generar Reporte”

Nombre del caso de uso:	Generar Reporte	
Id del caso de uso:	CU10	
Fuente:	APG	
Actor primario:	Administrador	
Descripción:	El administrador guardara o realizara la impresión de las consultas realizadas de forma general o de forma específica en un pdf.	
Precondición:	El administrador genero algún tipo de consulta	
Disparador:	Generar reporte	
Curso típico de eventos:	Acciones del Actor	Respuesta del Sistema
	1.- El administrador selecciona generar reporte	2.- El sistema hace conexión con la base de datos. 5.- Manda el reporte por medio de iReport
Cursos alternos:	Realice una consulta	
Conclusión:	El tener un archivo de las consultas permitirá dar a conocer al comité cuantos faltan por pagar	



Tabla 16. Caso de Uso “Administra usuario”

Nombre del caso de uso:	Administrar Usuario	
Id del caso de uso:	CU11	
Fuente:	APG	
Actor primario:	Administrador	
Descripción:	El administrador podrá dar de alta y realizar las credenciales de los usuarios nuevos, así como modificar su información o eliminar a algún usuario.	
Precondición:	El administrador está en el menú del administrador	
Disparador:	Se ejecuta la opción de administrar usuario	
Curso típico de eventos:	Acciones del Actor	Respuesta del Sistema
	1.- El administrador accede al menú de administrar	2.- El sistema reconoce la petición 3.- El sistema muestra la venta con el menú de administración de usuarios para poder realizar un alta, baja, consulta o eliminar a un usuario en específico.
Cursos alternos:	No hay cursos alternos	
Conclusión:	Permitir que el comité administre las cuentas de los nuevos usuarios y usuarios ya veteranos.	



Tabla 17. Caso de Uso “Generar nuevo usuario”

Nombre del caso de uso:	Generar Nuevo Usuario	
Id del caso de uso:	CU12	
Fuente:	APG	
Actor primario:	Administrador	
Descripción:	Dar de alta a nuevos usuarios de una toma de agua, y generar su registro al sistema.	
Precondición:	Ejecutar la opción de administrar usuario	
Disparador:	Activar botón de nuevo usuario	
Curso típico de eventos:	Acciones del Actor	Respuesta del Sistema
	1.- El administrador llena el formulario. 2.- Activa el botón de aceptar	3.- El sistema recibe la petición 4.- Realiza la conexión a la base de datos y guarda el registro.
Cursos alternos:	Cancelar petición	
Conclusión:	El administrador es capaz de realizar altas de nuevos usuarios al sistema.	



Tabla 18. Caso de Uso “Eliminar usuario”

Nombre del caso de uso:	Eliminar a Usuario	
Id del caso de uso:	CU13	
Fuente:	APG	
Actor primario:	Administrador	
Descripción:	El administrador será capaz de dar de baja a usuarios.	
Precondición:	Ejecutar la opción de administrar usuario	
Disparador:	Activar botón de eliminar usuario	
Curso típico de eventos:	Acciones del Actor	Respuesta del Sistema
	1.- El administrador Realiza la consulta de los usuarios. 5.- Selecciona al usuario que desea eliminar. 6.- Activa el botón de eliminar	3.- El sistema manda la petición a la base de datos. 4.- La base de datos muestra los resultados, el sistema manda esta información a pantalla. 7.- El sistema recibe la petición y realiza la conexión a la base de datos para eliminar el registro.
Cursos alternos:	Cancelar petición	
Conclusión:	Los usuarios morosos o finados serán capaces de limpiarlos del sistema para que no estén como archivo muerto.	



Tabla 19. Caso de Uso “Modificar usuario”

Nombre del caso de uso:	Modificar a usuario	
Id del caso de uso:	CU14	
Fuente:	APG	
Actor primario:	Administrador	
Descripción:	En el caso de que el administrador necesite modificar la información del usuario lo podrá realizar de dos formas distintas ya sea llenando de nuevo el formulario o realizando una consulta y modificar sus registros	
Precondición:	Ejecutar la opción de administrar usuario	
Disparador:	Activar botón de actualizar usuario	
Curso típico de eventos:	Acciones del Actor	Respuesta del Sistema
	1.- El administrador llena el formulario , y presiona el botón de actualizar	2.- El sistema recibe la petición 3.- Realiza la conexión a la base de datos 4.- La base de datos ejecuta las acciones y manda los resultados 5.- El sistema recibe los resultados de la BD y los muestra en pantalla.
Cursos alternos:	Cancelar petición	
Conclusión:	Los errores de registro son más comunes de lo que creemos.	



Tabla 20. Caso de Uso “Imprimir”

Nombre del caso de uso:	Imprimir	
Id del caso de uso:	CU15	
Fuente:	APG	
Actor primario:	Administrador	
Descripción:	Es necesario tener algún comprobante para validar los movimientos realizados dentro del sistema, para esto tenemos la opción de imprimirlos.	
Precondición:	Haber entrado al sistema como administrador	
Disparador:	Seleccionar la opción de generar reportes, generar o modificar algún usuario o Administrador	
Curso típico de eventos:	Acciones del Actor	Respuesta del Sistema
	1.- Activar la opción de Imprimir	2.- El sistema recibe la petición y manda la configuración de impresión.
Cursos alternos:	Cancelar petición, no existe impresora, no hay papel	
Conclusión:	Obtener los comprobantes de las consultas realizadas a si como sacar las nuevas credenciales para usuarios y administradores.	



Tabla 21. Caso de Uso "Asamblea"

Nombre del caso de uso:	Asamblea	
Id del caso de uso:	CU16	
Fuente:	APG	
Actor primario:	Administrador	
Descripción:	Menú de asamblea el administrador será capaz de levantar y cerrar una asamblea así como registrar el acceso de los usuarios a la misma.	
Precondición:	Haber entrado al sistema como administrador	
Disparador:	Acceder al menú de asamblea.	
Curso típico de eventos:	Acciones del Actor	Respuesta del Sistema
	1.- El administrador entra al menú de Asamblea	2.- El sistema recibe la petición y muestra en pantalla el menú de asamblea.
Cursos alternos:	No hay cursos alternos	
Conclusión:	El llevar un registro electrónico ahorrara mucho tiempo en el acceso de la misma así como un mejor control de las asambleas.	



Tabla 22. Caso de Uso “Nueva asamblea”

Nombre del caso de uso:	Nueva Asamblea	
Id del caso de uso:	CU17	
Fuente:	APG	
Actor primario:	Administrador	
Descripción:	El administrador será capaz de abrir una nueva sesión de asamblea dentro del menú de asamblea.	
Precondición:	Acceder al menú de asamblea.	
Disparador:	Seleccionar la opción nueva asamblea	
Curso típico de eventos:	Acciones del Actor	Respuesta del Sistema
	1.- El administrador da nueva asamblea	2.- El sistema recibe la petición y prepara todo para generar la nueva asamblea.
Cursos alternos:	Eliminar asamblea	
Conclusión:	Solo el administrador será capaz de administrar asambleas	



Tabla 23. Caso de Uso “Registrar asistentes”

Nombre del caso de uso:	Registrar asistentes	
Id del caso de uso:	CU18	
Fuente:	APG	
Actor primario:	Administrador	
Descripción:	El acceso a las asambleas será más organizado con un acceso electrónico por medio de sus tarjetas con código de barras.	
Precondición:	Acceder al menú de asamblea.	
Disparador:	Registrar asistencia	
Curso típico de eventos:	Acciones del Actor	Respuesta del Sistema
	1.- El administrador captura el código de barras y registra la asistencia.	2.- El sistema registra el código de barras. 3.- Envía la petición a la BD de nuevo registro de asistencia.
Cursos alternos:	El usuario no existe, conexión a la BD fallida	
Conclusión:	Este tipo de acceso a las asambleas será más ágil y se evitara la venta de contraseñas.	



Tabla 24. Caso de Uso “Eliminar sesión”

Nombre del caso de uso:	Eliminar sesión	
Id del caso de uso:	CU19	
Fuente:	APG	
Actor primario:	Administrador	
Descripción:	En muchas de las ocasiones las asambleas no pueden levantarse o iniciar debido al poco interés de los usuarios por asistir, en estos casos debemos de eliminar la asamblea	
Precondición:	Estar en el menú de asamblea y haber iniciado una.	
Disparador:	Activar el botón de eliminar	
Curso típico de eventos:	Acciones del Actor	Respuesta del Sistema
	1.- El administrador activa el botón de eliminar asamblea	2.- El sistema recibe la petición y prepara todas las órdenes para ejecutar las acciones. 3.- Se realiza la conexión a la BD 4.- La base de datos recibe las peticiones del sistema y ejecuta todas las ordenes.
Cursos alternos:	Cancelar petición	
Conclusión:	El administrador es capaz de eliminar una asamblea una vez que allá iniciado una, no importando si ya hubo registros levantados	



Tabla 25. Caso de Uso “Administrar administrador”

Nombre del caso de uso:	Administrar administrador	
Id del caso de uso:	CU20	
Fuente:	APG	
Actor primario:	Administrador	
Descripción:	El administrador podrá dar de alta y realizar las credenciales de los nuevos administradores del sistema, así como modificar su información o eliminar a alguno.	
Precondición:	El Administrador está en el menú del administrador	
Disparador:	Se ejecuta la opción de administrar administrador	
Curso típico de eventos:	Acciones del Actor	Respuesta del Sistema
	1.- El administrador accede al menú de administrar	2.- El sistema reconoce la petición 3.-El sistema muestra la venta con el menú de administración de Administrador para poder realizar un alta, baja, consulta o eliminar a un usuario en específico.
Cursos alternos:	No hay cursos alternos	
Conclusión:	Permitir que el comité administre sus propias cuentas y las del nuevo comité.	



Tabla 26. Caso de Uso "Alta administrador"

Nombre del caso de uso:	Alta administrador	
Id del caso de uso:	CU21	
Fuente:	APG	
Actor primario:	Administrador	
Descripción:	Dar de alta a nuevos usuarios del comité de agua, y generar su registro al sistema.	
Precondición:	Entrar al menú de administrar administrador	
Disparador:	Activar Botón de Nuevo	
Curso típico de eventos:	Acciones del Actor	Respuesta del Sistema
	1.- El administrador llena el formulario. 2.- Activa el botón de aceptar	3.- El sistema recibe la petición 4.- Realiza la conexión a la base de datos y guarda el registro.
Cursos alternos:	Cancelar petición	
Conclusión:	Permitir que el comité administre sus propias cuentas y las del nuevo comité.	



Tabla 27. Caso de Uso “Eliminar administrador”

Nombre del caso de uso:	Eliminar administrador	
Id del caso de uso:	CU22	
Fuente:	APG	
Actor primario:	Administrador	
Descripción:	El Administrador será capaz de dar de baja a otros administradores.	
Precondición:	Entrar al menú de administrar administrador	
Disparador:	Activar botón de eliminar	
Curso típico de eventos:	Acciones del Actor	Respuesta del Sistema
	1.- El administrador Realiza la consulta de los usuarios. 5.- Selecciona al usuario que desea eliminar. 6.- Activa el botón de eliminar	3.- El sistema manda la petición a la base de datos. 4.- La base de datos muestra los resultados, el sistema manda esta información a pantalla. 7.- El sistema recibe la petición y realiza la conexión a la base de datos para eliminar el registro.
Cursos alternos:	Cancelar petición	
Conclusión:	Cuando el comité cumple su periodo es necesario dar de baja sus cuentas y eliminar sus privilegios como administradores del sistema.	



Tabla 28. Caso de Uso “Modificar administrador”

Nombre del caso de uso:	Modificar administrador	
Id del caso de uso:	CU23	
Fuente:	APG	
Actor primario:	Administrador	
Descripción:	En el caso de que el administrador necesite modificar la información del el, lo podrá realizar de dos formas distintas ya sea llenando de nuevo el formulario o realizando una consulta y modificar sus registros	
Precondición:	Entrar al menú de administrar administrador	
Disparador:	Activar botón de actualizar	
Curso típico de eventos:	Acciones del Actor	Respuesta del Sistema
	1.- El administrador llena el formulario , y presiona el botón de actualizar	2.- El sistema recibe la petición 3.- Realiza la conexión a la base de datos 4.- La base de datos ejecuta las acciones y manda los resultados 5.- El sistema recibe los resultados de la BD y los muestra en pantalla.
Cursos alternos:	Cancelar petición	
Conclusión:	Los errores de registro son más comunes de lo que creemos.	



Tabla 29 Caso de uso. “Limpiar cuenta de usuario”

Nombre del caso de uso:	Limpiar cuenta de usuario	
Id del caso de uso:	CU24	
Fuente:	APG	
Actor primario:	Administrador	
Descripción:	El administrador será capaz de registrar los pagos del usuario a través del sistema, y limpiar su historial de adeudo y faltas de un usuario en específico.	
Precondición:	Estar en el menú del administrador	
Disparador:	Activar botón de realizar pagos	
Curso típico de eventos:	Acciones del Actor	Respuesta del Sistema
	1.- El administrador llena el formulario , y presiona el botón de pago	2.- El sistema recibe la petición 3.- Realiza la conexión a la base de datos 4.- La base de datos ejecuta las acciones y manda los resultados 5.- El sistema recibe los resultados de la BD y los muestra en pantalla.
Cursos alternos:	Cancelar petición, historial limpio	
Conclusión:	El sistema optimiza el tiempo de espera y limpia el historial del usuario con un solo clic.	



2.3. Modelado de contenido del SI de agua potable.

El modelo de contenido para el sistema de información se describe a partir de las páginas de aplicación, se representa a través de rectángulos cada página, estas se enlazan con flechas hacia los objetos de contenido (texto, gráficos, etc.) que son representados por medio de rectángulos (blancos), el modelo de contenido del sistema se presenta en las siguientes partes.

2.3.1 Modelado de contenido para usuario (dueño).

Contenido de la página principal.

Conforme a los casos de uso CU1, CU7 (Tabla 6 y 12) el usuario o administrador debe de contar con una interfaz que le permita entrar al sistema y sea fácil de entender y manejar , por tanto el usuario tendrá a su vista imágenes que simulen la acción a ejecutar y pueda desplegarse una nueva ventana para su acción. En este menú de inicio en un principio se verán varios botones desactivados hasta el momento de que accede al sistema por medio de su tarjeta con código de barras (Figura 8).

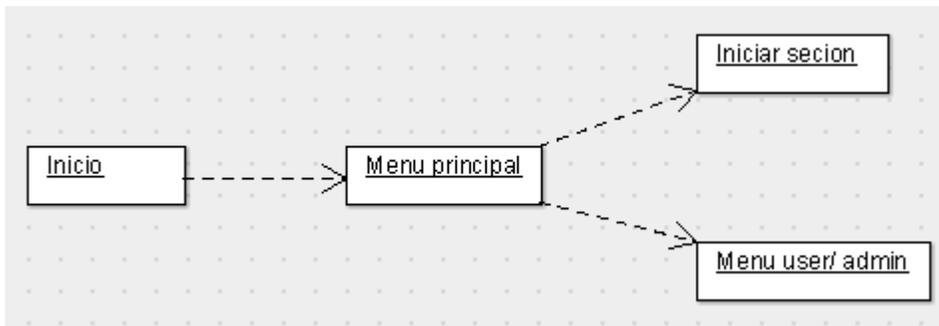


Figura 8. "contenido principal"

Contenido de página de consulta de faltas.

En base a los casos de uso CU3, CU2 en esta parte el usuario dispondrá de una interfaz en donde se le permitirá consultar el historial de sus faltas de todos los días que no ha pagado el total de sus multas, así como ver el monto de deuda por sus faltas que aún no ha liquidado, y teniendo en cuenta el CU6 el usuario podrá obtener un comprobante de esto por medio de una impresión de esta información (Figura 9).

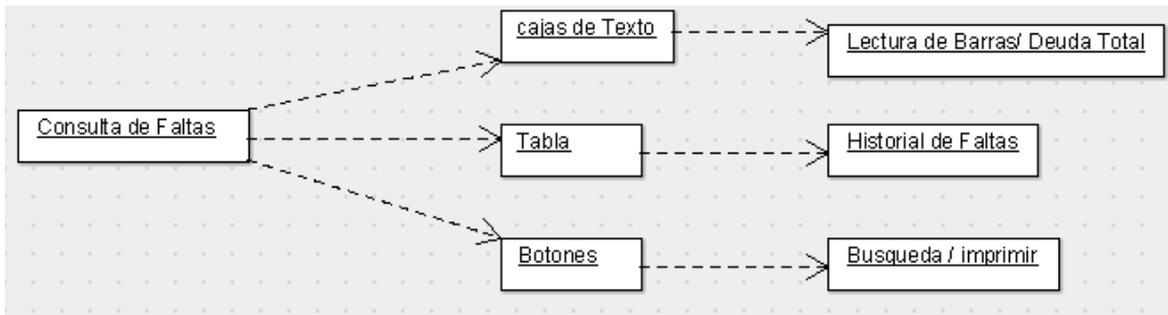


Figura 9. "consulta de faltas"

Contenido de Página, consultar retrasos

De acuerdo a los casos de uso CU2, CU4 y muy similar al contenido de página de consultar faltas el usuario será capaz de consultar sus atrasos totales no liquidados por medio de una tabla que mostrara el historial total de los pagos mensuales no realizados y ver adeudo total a través de las cajas de texto, así como obtener un comprobante de estos CU6 que será generado por medio de un botón, y como se pretende que el sistema sea de un fácil manejo y muy interactivo estos botones contarán con una imagen muy descriptiva de su acción (Figura 10).

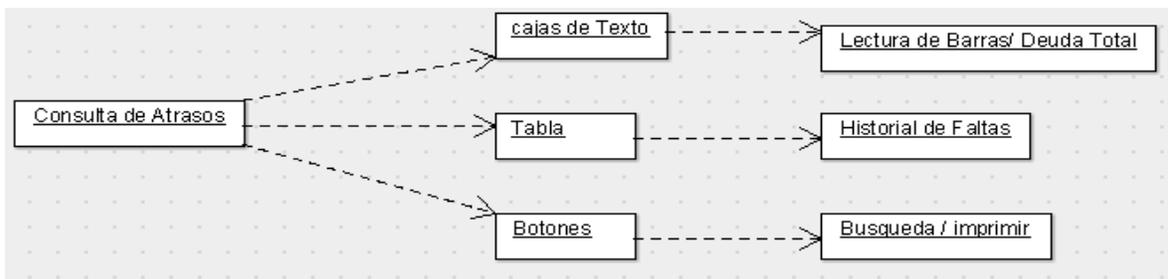


Figura 10. Consultar retrasos

Contenido de página, generar reporte

Esta página se basa en el caso de uso CU5 en el cual se genera el reporte total de los atrasos y faltas del historial del usuario que aún no han sido liquidados, en donde será necesario que registre sus código de barras haciendo uso del lector de barras para poder generar el reporte correspondiente, y a si mismo poder obtener el comprobante de este movimiento según el CU6 el cual es llevado la acción a través de un botón que generara el reporte y se mostrara al usuario finalmente en una nueva ventana (Figura 11).

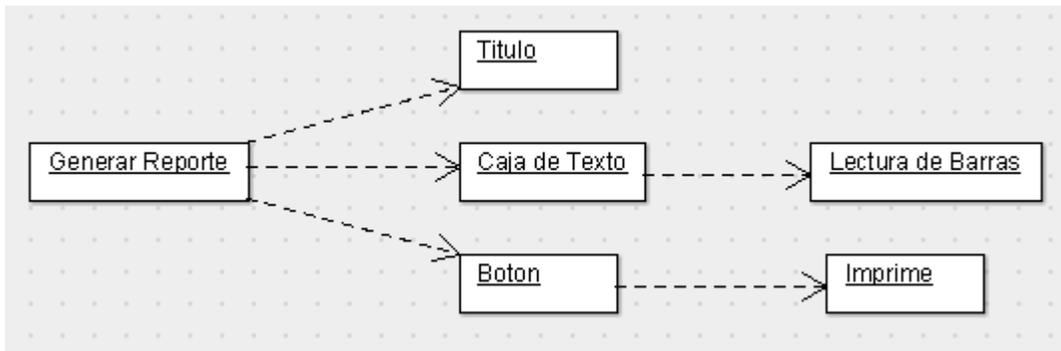


Figura 11. Generar Reporte

2.3.2. Modelado de contenido para el administrador.

El contenido mostrado para el administrador es considerado una sesión privilegiada debido a que goza de privilegios que el usuario no cuenta y que puede realizar más acciones.

Contenido de página, ingresar al sistema.

Para poder acceder como administrador en la cuenta privilegiada es necesario que compruebe que es un usuario esto será permitido a través de esta interfaz según nos lo indica el caso de uso CU7 en donde aremos uso del lector de barras para poder registrar el código de barras de nuestro sistema (Figura 12).



Figura 12. Ingresar al sistema

Contenido de página, generar reporte.

Los reportes que se generaran a partir de los casos de uso CU8, CU9, CU10 servirán como contenido de esta interfaz que será mostrada al administrador con un formulario para poder realizar una consulta en especifica de un usuario o de deudores que estén el sistema registrados todo esto será lanzado por medio de un botón el cual lanzara un reporte con la posibilidad de imprimirlo o guardarlo en archivo CU15 (Figura 12).

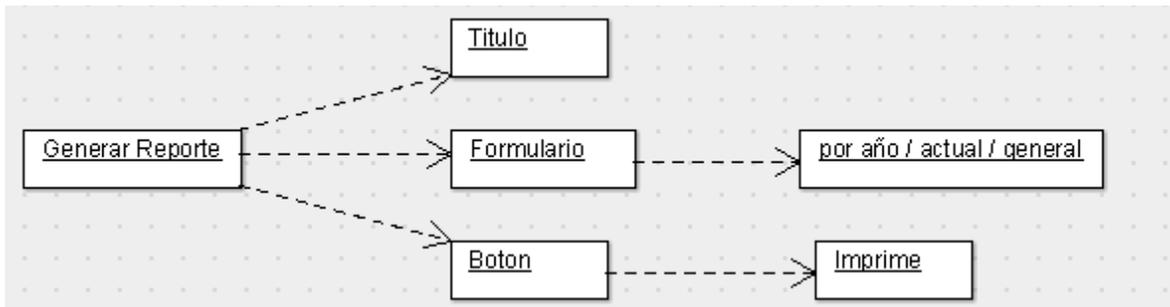


Figura 12. Generar reporte Ad.

Contenido de página, administrar usuario

Las funcionalidades para la gestión de los registros de nuevos usuarios en el sistema (CU11, CU12, CU13, CU14) para el administrador deben de ser lo más sencillas posibles, de fácil manejo, intuitivas y con una interfaz amigable, estas características se verán plasmadas en esta página de administración de usuario. En donde el administrador será capaz de realizar altas de usuarios con su código de barras correspondiente y generar las nuevas credenciales para el usuario (CU15) y así mismo corregir la información o eliminar al usuario del sistema si es necesario (Figura 13).

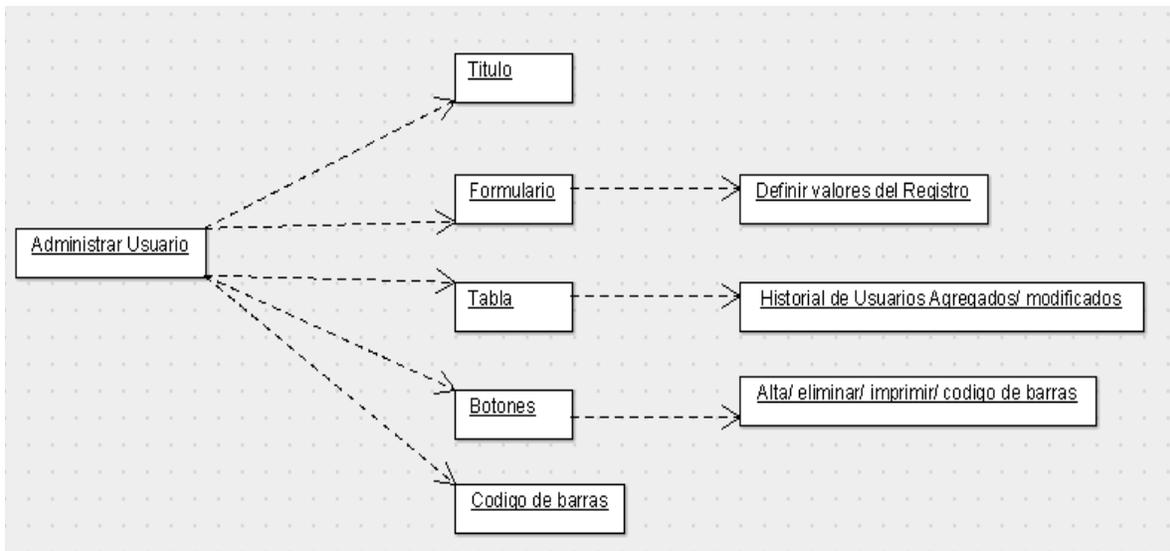


Figura 13. Administrar usuario.



Contenido de página, *asamblea*.

Esta página se basa en los casos de uso CU16, CU17, CU18, CU19 en la cual en esta ventana el administrador dispondrá de una interfaz en donde podrá realizar el levantamiento de una nueva sesión de *asamblea*, e ir registrando a los usuarios su acceso a la misma (CU18) y en un caso extremo donde deba de cancelarse la sesión levantada se dispone de una opción para cancelar la petición CU19 (Figura 14).



Figura 14. *Asamblea*

Contenido de la página *administrar administrador*.

Muy similar al contenido de *administrar usuario*, en esta ventana (CU16) se cumplen varios casos de uso (CU17, CU18, CU19) para hacer más simple, intuitivo y amigable para el administrador el uso de esta interfaz, podrá realizar las altas de los nuevos integrantes del comité con su respectiva credencial el cual contara con un código de barras para su fácil acceso al sistema, de igual forma podrán ser modificados o eliminados del sistema, esto para poder garantizar que estos administradores que han dejado el puesto ya no gocen de estos privilegios (Figura 15).

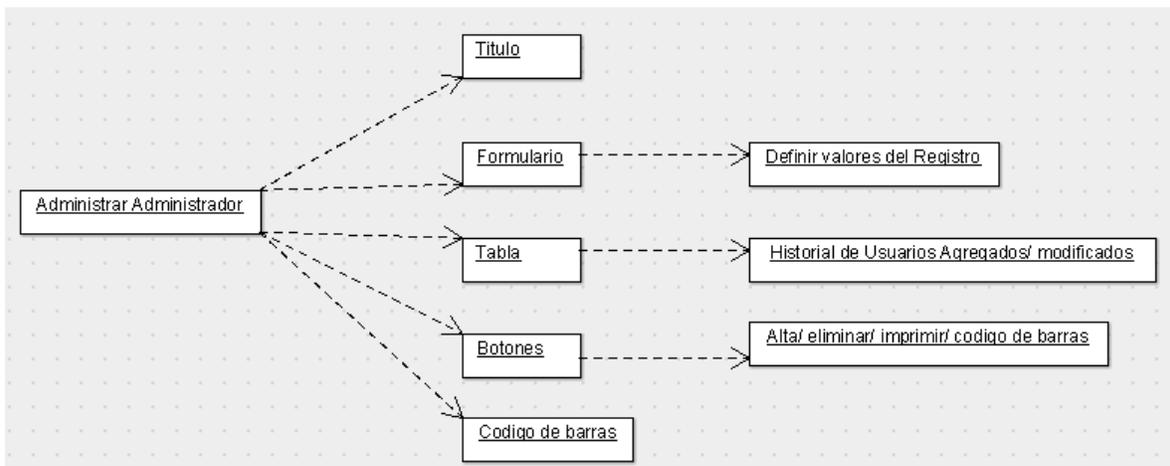


Figura 15. *Administrar administrador*



Contenido de la página limpiar cuenta de usuario.

En esta interfaz se cumple el caso de uso CU24 en donde el administrador cuenta con una sencilla y practica herramienta para poder poner en ceros los atrasos y faltas de un usuario en específico esto es que pueda realizar sus pagos correspondientes (Figura 16).

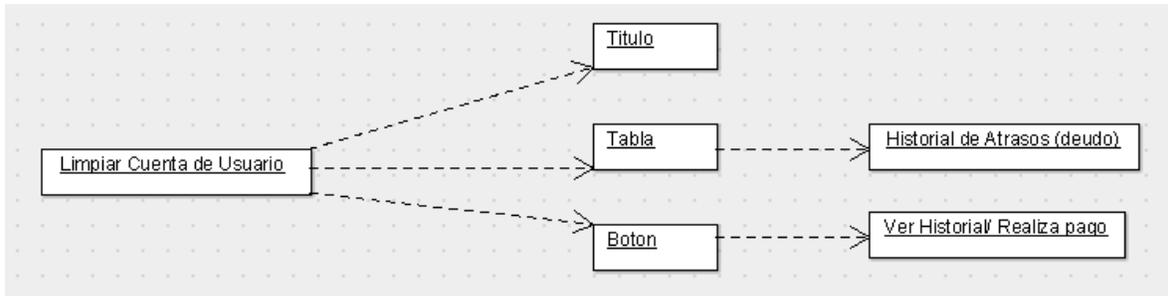


Figura 16. Limpiar cuenta de usuario

2.4. Modelado funcional del sistema de información

Para poder analizar y representar la funcionalidad del sistema se hará uso de los diagramas de actividad, estos complementan a los diagramas de caso de uso al proporcionar una representación gráfica del flujo de interacción.

El diagrama de actividad utiliza rectángulos para indicar una función específica del sistema, flechas para representar el flujo a través del sistema, rombos de decisión para mostrar una ramificación por decisión y líneas horizontales solidas de decisión para mostrar una ramificación por decisión y líneas horizontales sólidas para indicar que ocurren actividades paralelas (Presman Robert S, 2005).



2.4.1. Diagramas de actividades del sistema.

Diagrama de actividades para “Ingresar al sistema”.

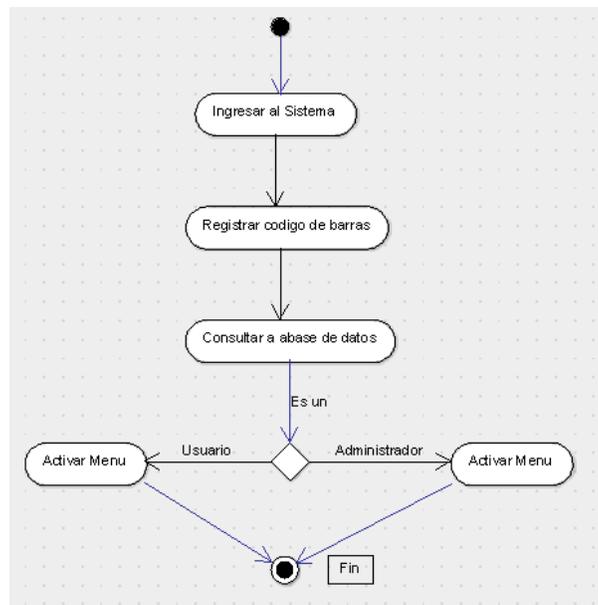


Figura 17. Diagrama de actividad para “Ingresar al sistema de información”.

El diagrama de la Figura 17 muestra las actividades que se realizan al tratar de acceder al sistema, en donde primero se tienen que identificar nuestros usuarios del sistema, ya sea como usuarios o como administradores (CU1, CU7), y dependiendo del tipo de usuario se activará un menú para cada sesión.



Diagrama de actividades para “consulta de adeudo”.

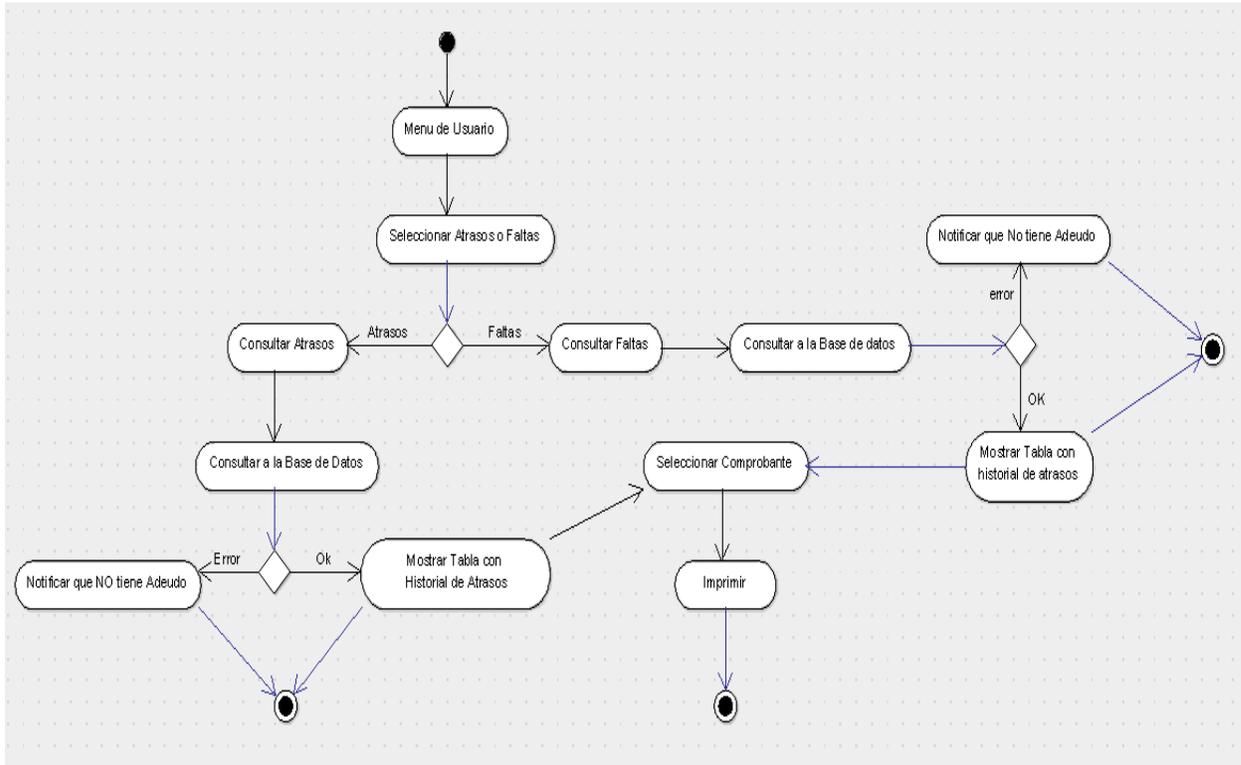


Figura 18. Diagrama de actividad para “Consulta de adeudo”.

Una vez que el usuario ha accedido a su menú principal puede realizar las siguientes actividades descritas en los casos de uso CU2, CU3, CU6 en donde será capaz de realizar las consultas de su historial de los pagos que no ha realizado y sus faltas totales que no ha pagado sus multas por tal motivo tenemos que, debe de ingresar al sistema interesarse por alguna opción, consultar atrasos o consultar faltas para esta acción debe el sistema conectarse a la base de datos del sistema y verificar que exista algún registro si no existe nada , el sistema no mostrara nada y mandara un mensaje de historial limpio, y existen registros estos se mostraran en una tabla con todos los registros totales y mostrando el monto total de adeudo, en caso de que el usuario necesite algún comprobante de este movimiento tiene la opción obtener al imprimir la pantalla de consulta (CU6) y finalizaría hasta este punto la actividad descrita en la Figura18.



Diagrama de actividad para “Generar reporte”.

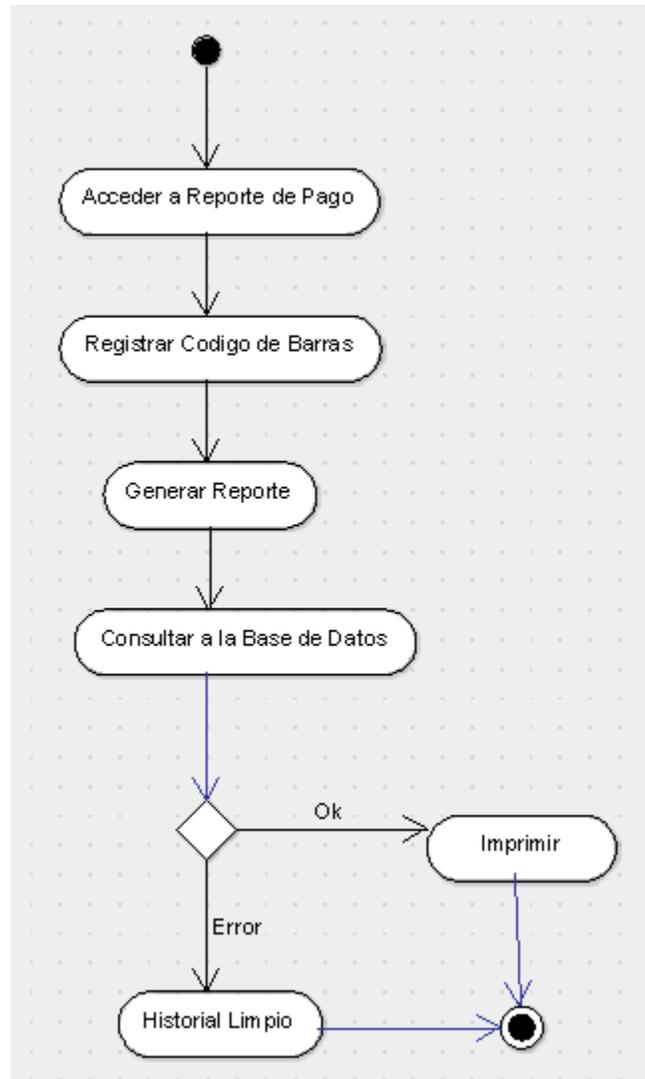


Figura 19. Diagrama de actividad para “Generar reporte”.

En el diagrama de actividad en la Figura 19 se describe el caso de uso CU5 en donde el usuario puede generar el reporte total de tanto de sus atrasos y faltas facilitando el cálculo del adeudo total sin necesidad de una búsqueda tediosa, utilizando la herramienta de iReport nos facilitan en mucho generar este tipo de reporte para poder mandarlo a impresión CU6 y tener el comprobante total de adeudo del usuario en caso de que su historial este limpio este reporte no se puede generar finalizando esta actividad.



Diagrama de actividad para “Generar reporte” en el caso del administrador.

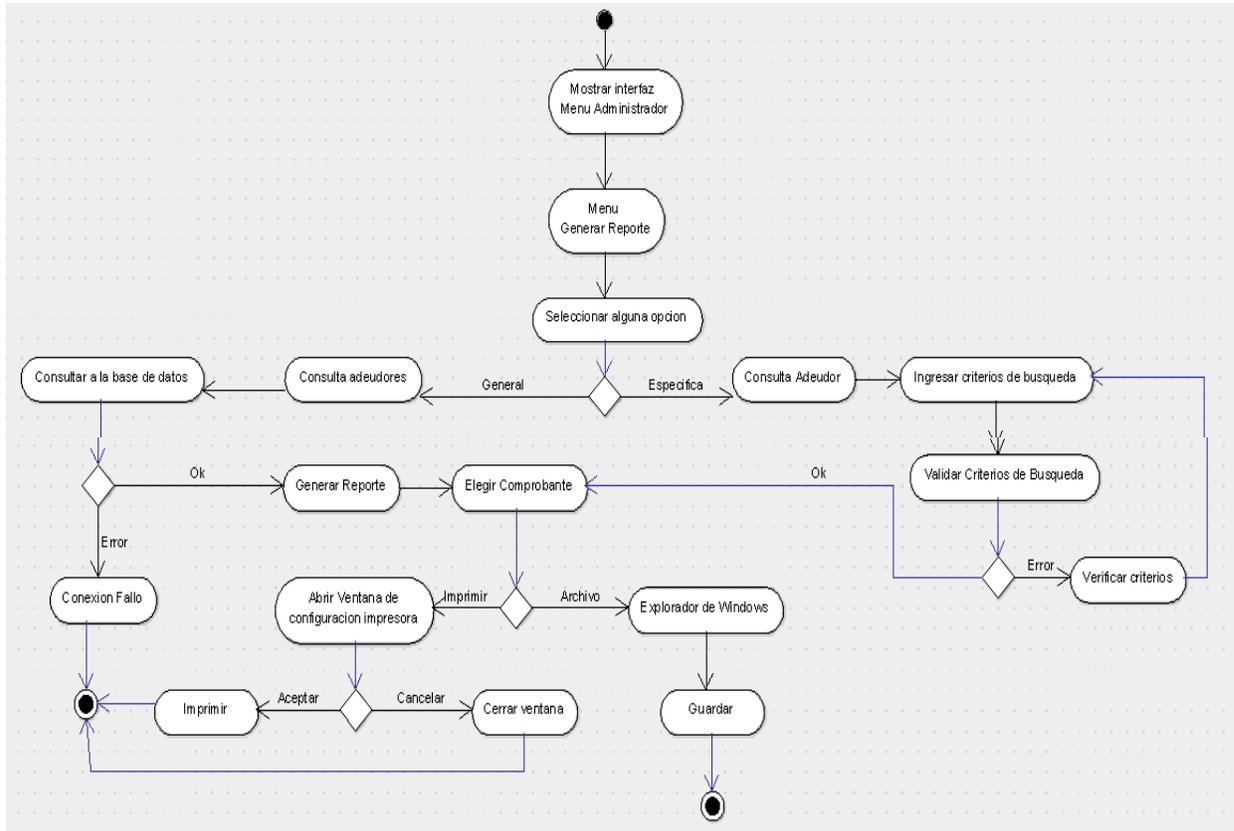


Figura 20. Diagrama de actividad “Generar reporte” en el caso del administrador.

Para poder generar el reporte en el caso del administrador vemos el diagrama de actividad representada en la Figura 20, en donde podemos ver descritos los casos de uso CU8, CU9 CU10, CU15 en donde el administrador puede realizar consultas a la base de datos de forma general de los usuarios que ya cuentan con sus pagos y los que aún faltan por realizar sus pagos mensuales a través de varios años, y en otra opción cuenta para ver el historial en específico de un usuario, proporcionando los criterios de búsqueda del usuario a buscar a sí mismo para poder tener un archivo de los historiales capturados cuenta con las opciones de tener un comprobante ya sea imprimiendo todo este historial o bien almacenándolo en forma de un archivo en el propio equipo o en cualquier unidad de almacenamiento masivo.



Diagrama de actividad para “Generar nuevo usuario”

Para realizar el alta de un nuevo usuario (CU12, CU11) en nuestro sistema se realizaran las siguientes actividades, se llevara a cabo el llenado de un formulario esto para registrar los datos personales del usuario así mismo como credencial generando su código de barras y procediendo a imprimirlo para que este comprobante (CU15), lo obtenga el usuario que será uso indispensable para poder hacer su navegación en el sistema, ene l siguiente diagrama se visualiza con más detalle estos paso Figura 21.

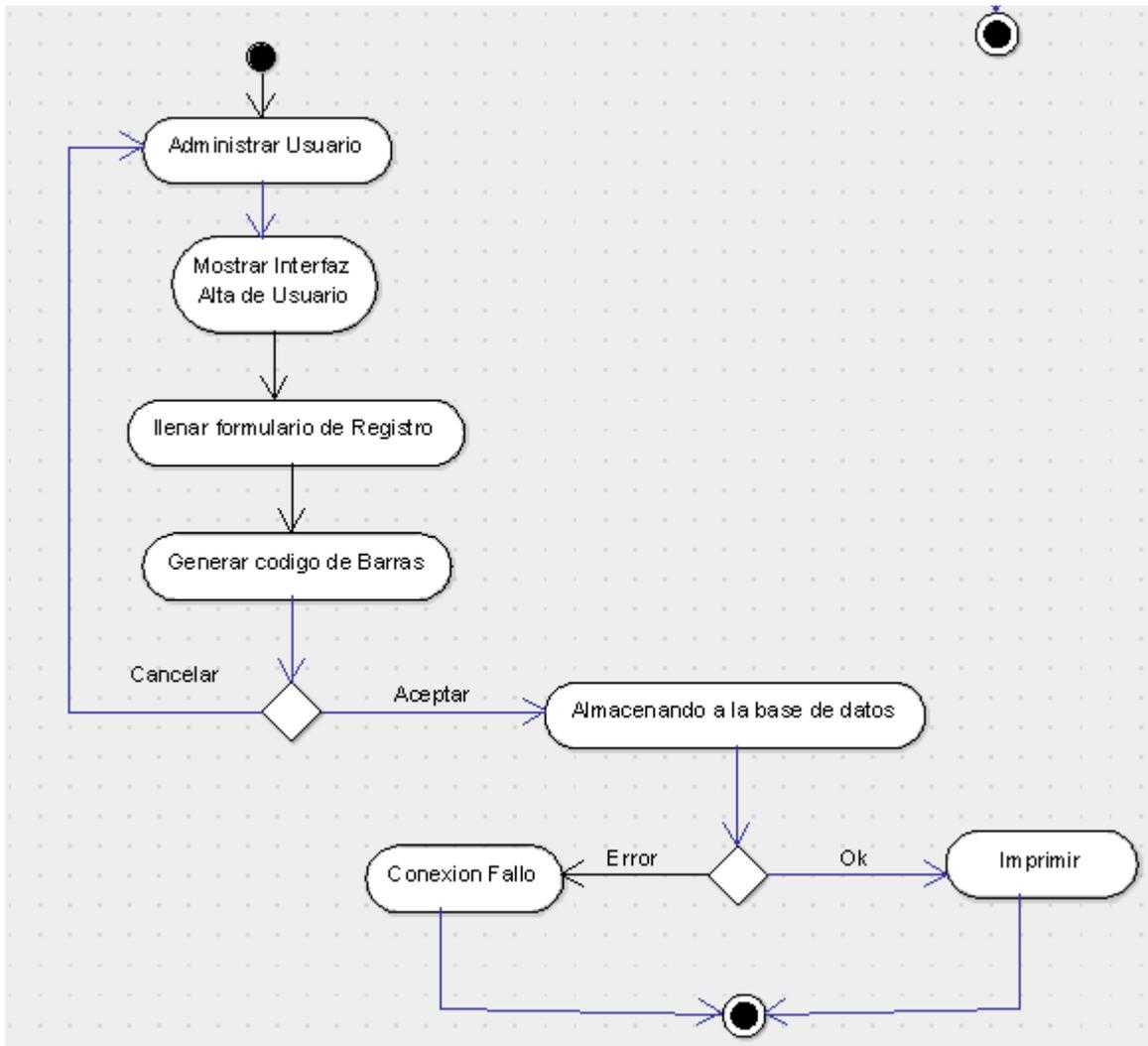


Figura 21. Diagrama de actividad para dar de alta a un usuario.



Diagrama de actividad para “Eliminar usuario”.

En la siguiente actividad el administrador goza del privilegio de poder eliminar a un usuario del sistema (CU13, CU11) , y para esta actividad esta descrita en la Figura 22, en donde podemos ver los pasos que se deben de seguir para poder realizar este paso, el primero es consultar si existe el usuario que deseamos eliminar, si es así podemos avanzar a realizar esta baja del usuario en el sistema y terminar hasta aquí la actividad, o bien corroborar que efectivamente se ha eliminado el usuario realizando una segunda consulta para verificar que el usuario ya no existe en el sistema.

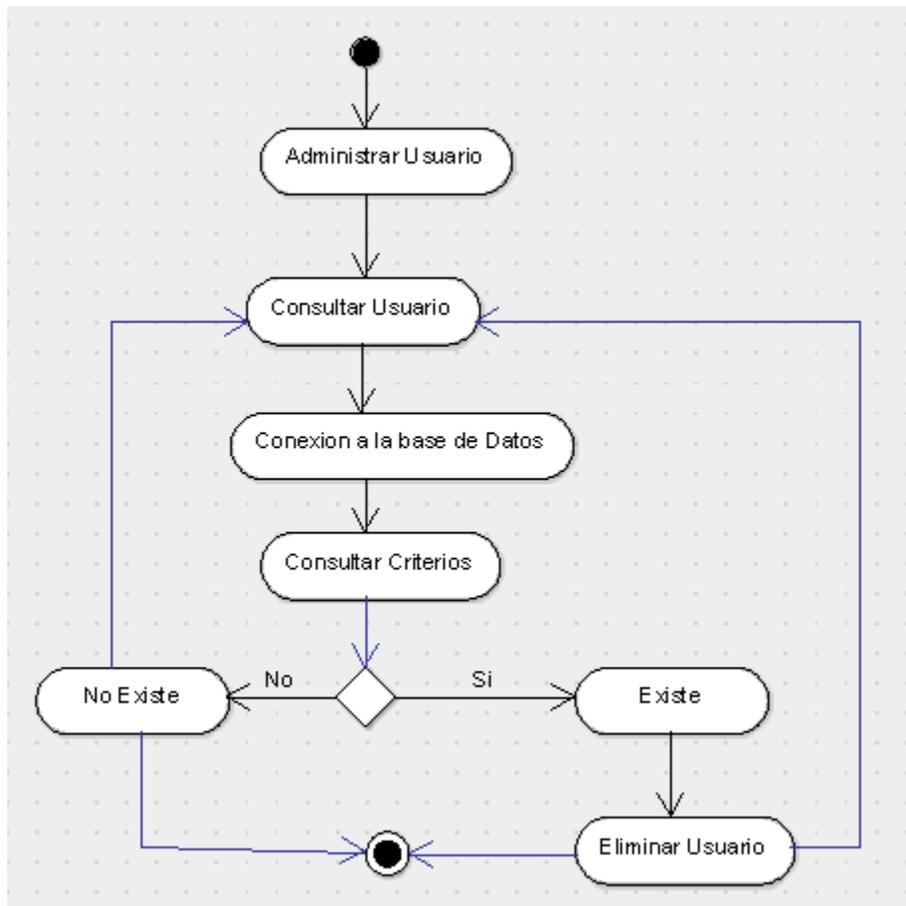


Figura 22. Diagrama de actividad para realizar la eliminación de un usuario.



Diagrama de actividad para “Modificar a un usuario”.

Para realizar la modificación de un usuario (CU14, CU11) es necesario saber si existe en el sistema para esto realizamos una consulta si existe procedemos a modificar su registro con nuevos parámetros y a almacenar esta nueva información en la base de datos del sistema y como estamos hablando de datos del usuario es necesario volver a imprimirle un comprobante (CU15) que será su credencial para poder comprobar que es un usuario del sistema (Figura 23).

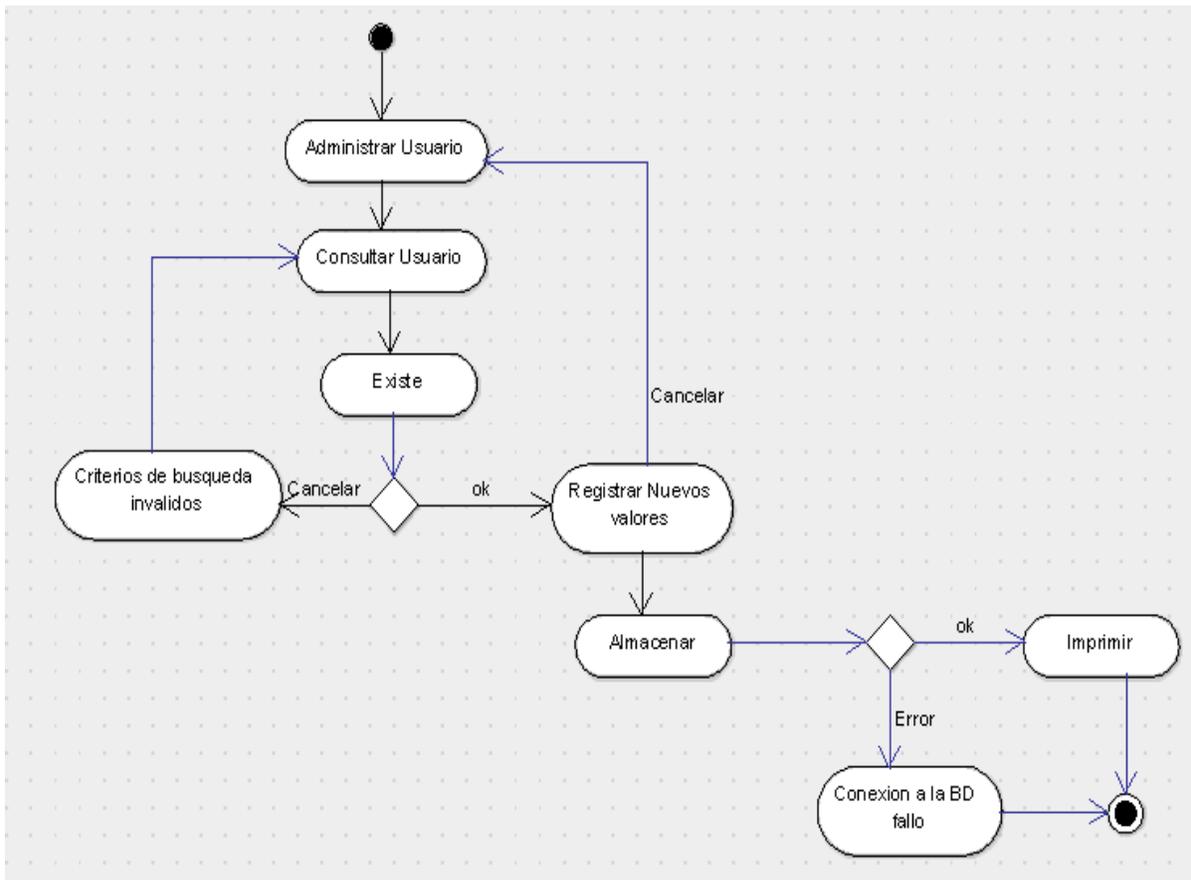


Figura 23. Diagrama de actividad para modificar un usuario.



Diagrama de actividad para realizar “Abrir nueva sesión”.

El administrador puede realizar las actividades de levantar una nueva asamblea y decidir los días de su apertura (CU16, CU17) en donde solo es necesario colocar la fecha del día y poder registrar el día en el sistema para obtener la nueva asamblea (Figura 24).

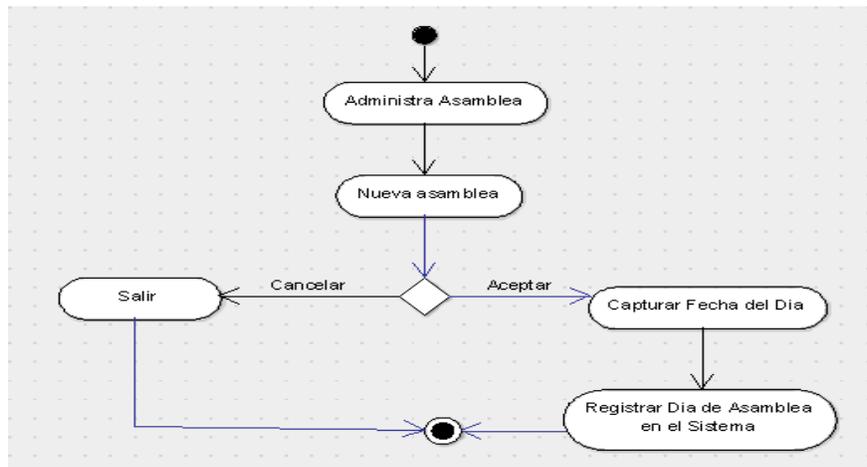


Figura 24. Diagrama de actividad para abrir nueva sesión



Diagrama de Actividad para “Registrar Asistentes”.

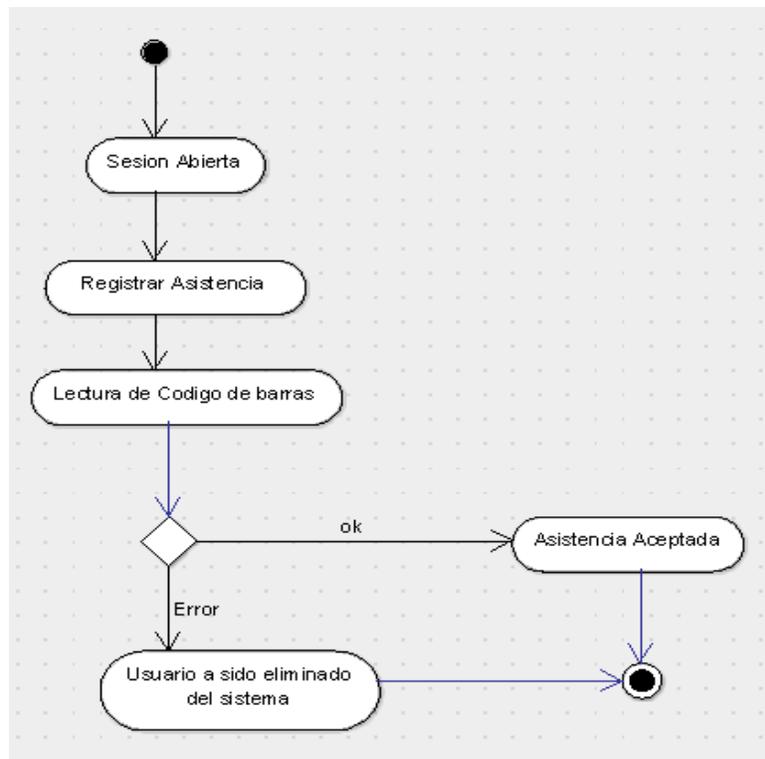


Figura 25. Diagrama de actividad para “registrar asistentes”.

Puesto que el administrador es el responsable de registrar la asistencia de los usuarios de las asambleas que se llevan a cabo, se representa en el diagrama de actividades (Figura 25) como se realiza la toma de asistencias (CU18) en el sistema de información, haciendo uso de la identificación de la tarjeta personal con el código de barras, solo bastara con escanearlo y tomar la asistencia del usuario dentro del sistema, en el caso que se llegase a presentar en donde ocasione un error el registro se deberá por que el usuario ya ha sido eliminado previamente o que la tarjeta este muy dañada.

Diagrama de actividad para “cerrar o eliminar sesión”.

Como lo mencionamos en el caso de uso CU19 es necesario cerrar la sesión en cierto momento para evitar seguir registrando a usuarios que no llegaron en este día, o en casos especiales cuando no se alcanza a un numero de asistencia en el cual se pueda abrir la



asamblea se debe de proceder a cancelar la misma para esto se describe en el diagrama de actividad representada en la Figura 26, en donde podemos apreciar en el caso de cancelar asamblea se limpian todos los registros que se vieron afectados en este día para así evitar el cobro a los que no llegaron.

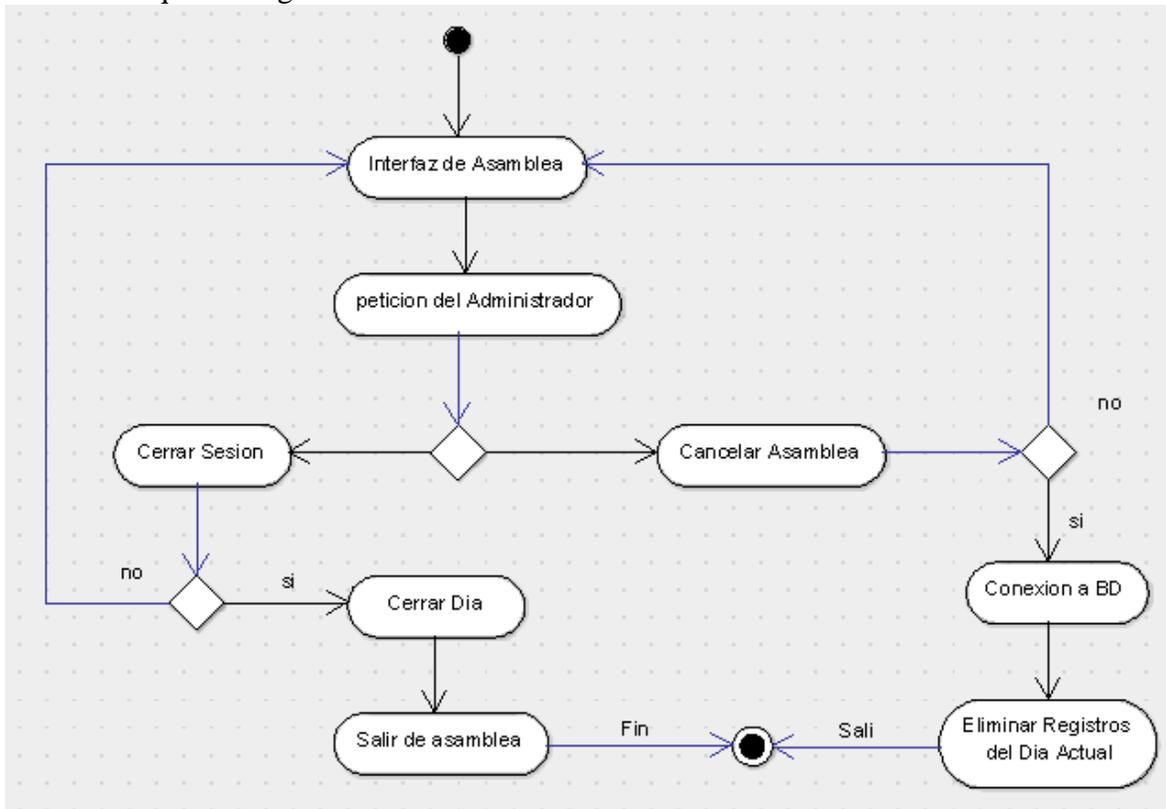


Figura 26. Diagrama de actividad para cerrar o eliminar sesión de asamblea.

2.5. Diseño de la base de datos.

Con los análisis que se han realizado con anterioridad, se puede realizar el diseño de la base de datos que contendrá la información de personal de los usuarios, así como el historial de sus pagos, adeudos, multas en el SI. El gestor de base de datos a utilizar será el IDE de WorkBench SQL, manejando el lenguaje de programación en BDs MySql.

MySql es una tecnología de libre distribución, lo que ahorrara los costos finales del prototipo, desarrollando sentencias primitivas esto es muy básicas, ya que no estamos hablando de un sistema distribuido ni de servidores, las sentencias SQL usadas serán primitivas para uso de este proyecto.



2.5.1. Identificación de las entidades básicas.

De acuerdo con el análisis previo de la información y con la información obtenida de las reuniones con el cliente se identificó la existencia de 2 sistemas generales las cuales son de acceso y de pagos, las cuales se divide cada una en dos entidades generales o principales estas son:

- Entidad administrador: almacena información del administrador.
- Entidad dueño: almacena información del dueño.
- Entidad pago mensual: almacena historial de pago
- Entidad asamblea: almacena historial de los días de sesión.

2.5.2. Diagrama entidad – relación.

El Modelo de Entidad Relación es un modelo de datos basado en una percepción del mundo real que consiste en un conjunto de objetos básicos llamados entidades y relaciones entre estos objetos, implementándose en forma gráfica a través del Diagrama Entidad Relación.

Se puede definir como Entidad a cualquier objeto, real o abstracto, que existe en un contexto determinado o puede llegar a existir y del cual deseamos guardar información (entidad = tabla). Hablamos de ejemplares cuando nos referimos a una clase de objetos con características similares (ejemplares = registros). Los atributos son características o propiedades asociadas a la entidad que toman valor en una instancia particular. Ejemplo; nombre, cédula, teléfono.

Se denomina clave principal o primaria al atributo o conjunto mínimo de atributos (uno o más campos) que permiten identificar en forma única cada instancia de la entidad, es decir, a cada registro de la tabla. Las claves principales se utilizan cuando se necesita hacer referencia a registros específicos de una tabla desde otra tabla. En un principio se puede identificar más de un atributo que cumpla las condiciones para ser clave, los mismos se denominan claves candidatas.

Si la clave primaria se determina mediante un solo atributo de la entidad, entonces se dice que la misma es una clave simple. En caso de estar conformada por más de un atributo, la misma se conoce como clave compuesta. La clave foránea (también llamada externa o secundaria) es un atributo que es clave primaria en otra entidad con la cual se relaciona.

Tipos de cardinalidad.



Se entiende por relación a la asociación entre 2 o más entidades.

- **Relación Uno a Uno:** Cuando un registro de una tabla sólo puede estar relacionado con un único registro de la otra tabla y viceversa. En este caso la clave foránea se ubica en alguna de las 2 tablas.
- **Relación Uno a Muchos:** Cuando un registro de una tabla (tabla secundaria) sólo puede estar relacionado con un único registro de la otra tabla (tabla principal) y un registro de la tabla principal puede tener más de un registro relacionado en la tabla secundaria. En este caso la clave foránea se ubica en la tabla secundaria.
- **Relación Muchos a Muchos:** Cuando un registro de una tabla puede estar relacionado con más de un registro de la otra tabla y viceversa. En este caso las dos tablas no pueden estar relacionadas directamente, se tiene que añadir una tabla entre las dos (Tabla débil o de vinculación) que incluya los pares de valores relacionados entre sí.
- El nombre de tabla débil deviene que con sus atributos propios no se puede encontrar la clave, por estar asociada a otra entidad. La clave de esta tabla se conforma por la unión de los campos claves de las tablas que relaciona.

Los pasos a seguir para obtener el diagrama de entidad relación son las siguientes:

1. Identificar las entidades.
2. Determinar las claves primarias.
3. Describir los atributos de las entidades.
4. Establecer relaciones entre las entidades.
5. Dibujar el modelo de datos.

Finalmente en la Figura 27 se muestran los resultados finales de estos pasos, en donde obtenemos el diagrama finalizado.

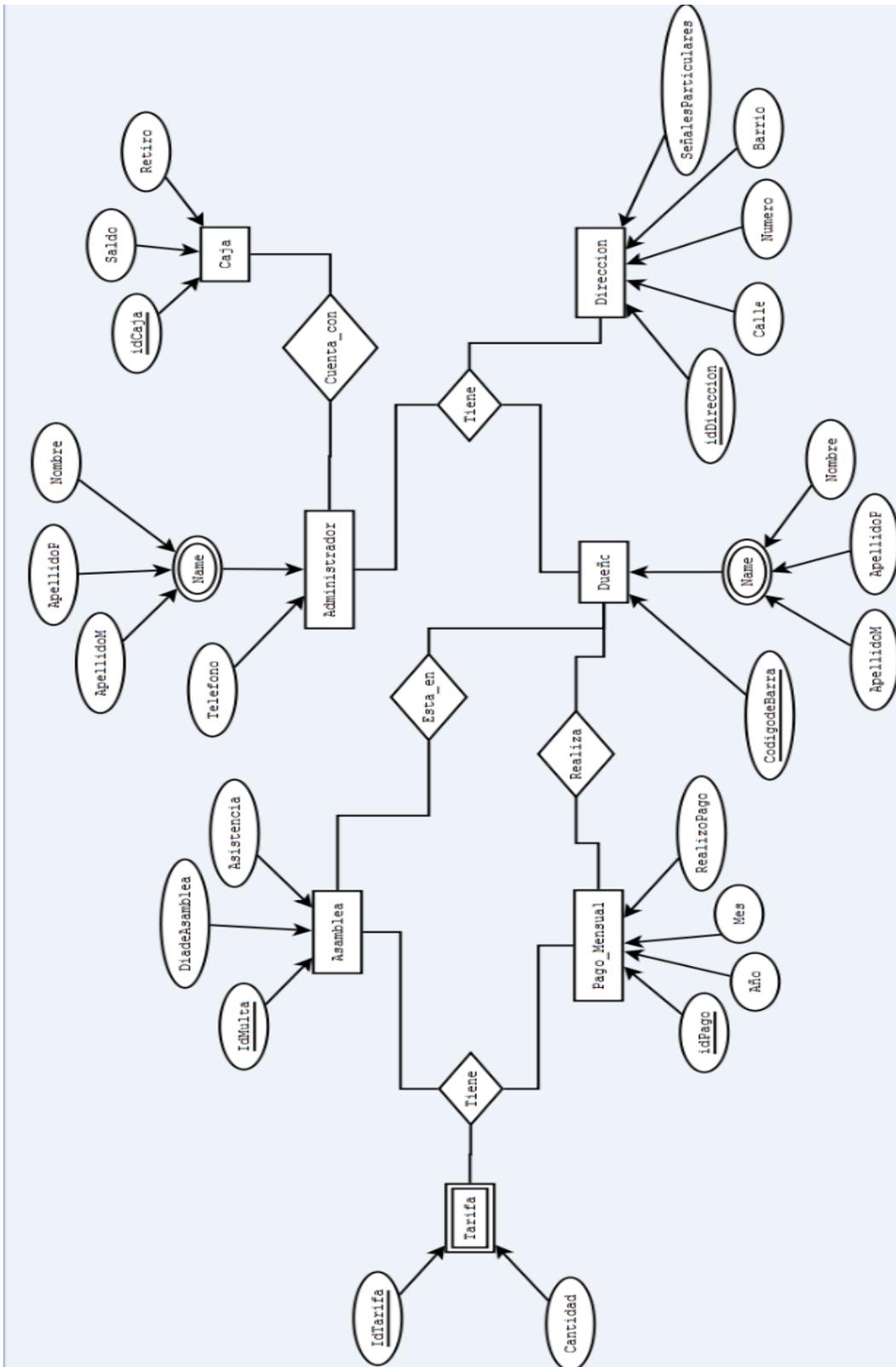


Figura 27. Diagrama entidad relación.



2.5.3 Creación de la entidades administrador / dueño

Analizando la información de estas dos entidades, y tomando en cuenta las reglas de normalización se encontró parámetros de tabla que podían permanecer a una nueva tabla para mayor manejo al momento de almacenar esta información, y en una nueva tabla que está directamente relacionado con la entidad de administrador. Por lo que se procedió a normalizar la tabla quedando de la siguiente forma.

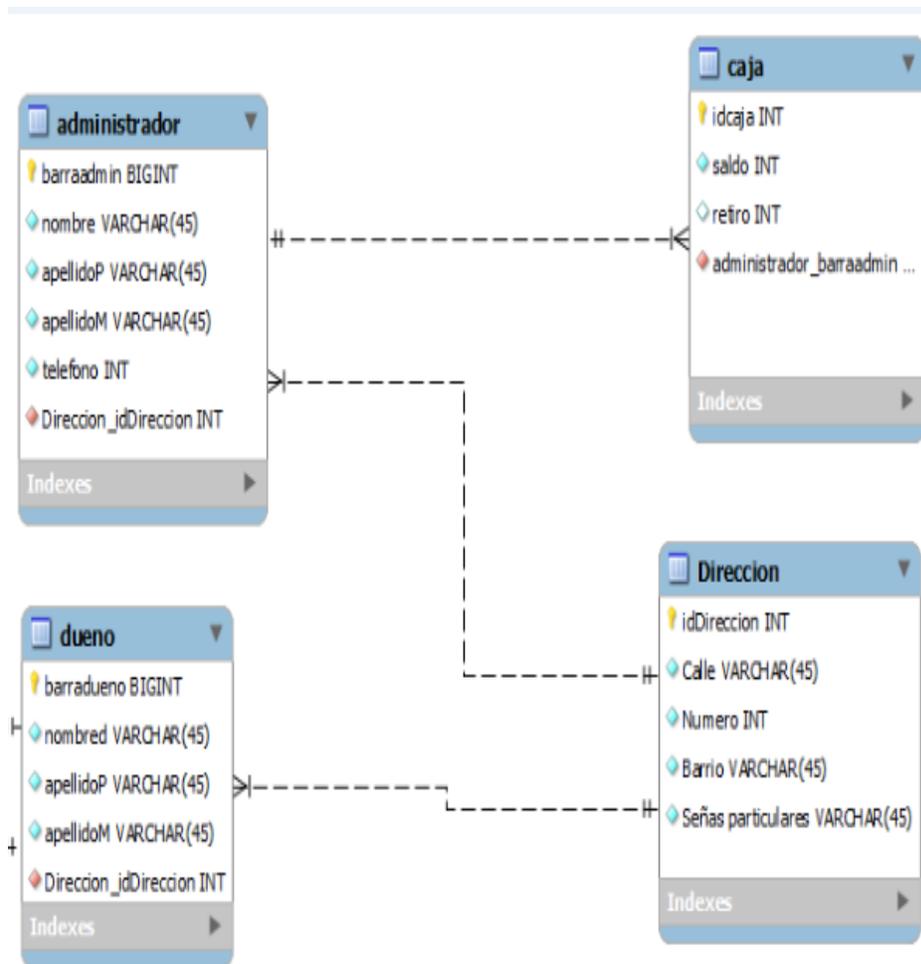


Figura 28. Estructura normalizada de las entidades administrador/ dueño.

En la Figura 28, se puede visualizar como se normalizo las entidades resultando una nueva tabla llamada dirección esto es para un almacenaje más descriptivo y a si mismo ayudar a la base de datos para ahorro del espacio de almacenamiento de la información, se puede apreciar una relación con las nuevas tablas en una relación M: 1 varios a uno.



2.5.4 Creación de las entidades del sistema de pago.

Las entidades presentadas anteriormente en el inciso 2.5.3 presentan las entidades localizadas dentro del sistema de acceso, a continuación veremos las entidades que describen al sistema de pago, en donde presentamos a dos principales tablas, que de igual forma se le ha aplicado las normas de normalización, estamos hablando de asamblea y pago mensual las cuales se detectó que no contaban con la 1FN, por lo cual se procedió a corregirlo quedando de la siguiente forma.

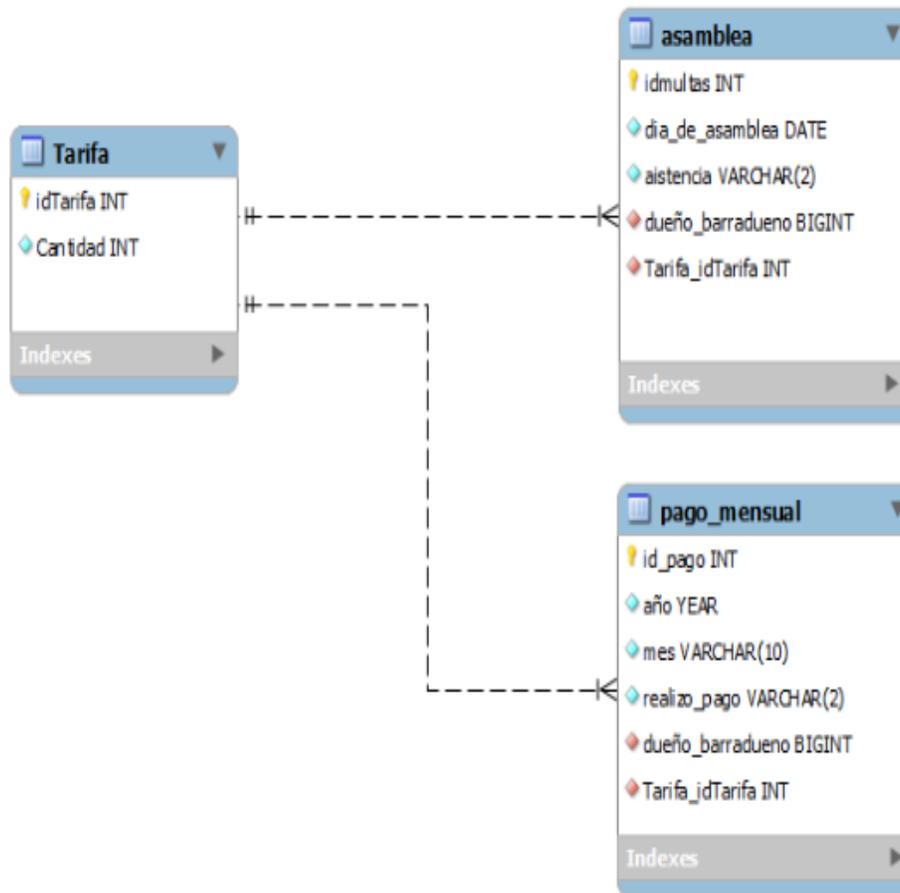


Figura 29. Estructura del sistema de pago.

Estas tablas son las encargadas de almacenar toda la información que al dueño de toma le interesara consultar, como lo son su historial negativo de multas y atrasos mensuales los cuales en el sistema se visualizaran por medio de la interfaz que se pretende diseñar y que ya se ha realizado un bosquejo de ello en capítulo anteriores.



La creación de una tabla tarifa se justifica porque esta tarifa es muy repetitiva en ambos casos que, en el caso de nuestro sistema solo existen dos tipos de tarifas, las de pago puntual y las multas por atrasos o faltas a asambleas (Figura 29).

2.5.5 Modelo relacional.

Los pasos a seguir para transformar nuestro diagrama E-R a un modelo relacional son varios, a continuación se describen (Elmasri, 2002):

- El modelo debe de estar normalizado.
- Las entidades y relaciones se convierten en relaciones.
- Con las relaciones resultantes se elabora el modelo relacional (metadatos).
- Se elabora el modelo relacional extendido (tuplas).

Al empezar el diseño de las tablas generales descritas anteriormente se encontró que, la base de datos no contaba con la normalización correspondiente por lo que, se procedió a realizarlo tomando en cuenta 3 niveles de normalización, que a continuación describiremos:

La normalización es el proceso mediante el cual se transforman datos complejos a un conjunto de estructuras de datos más pequeñas, que además de ser más simples y más estables, son más fáciles de mantener. También se puede entender la normalización como una serie de reglas que sirven para ayudar a los diseñadores de bases de datos a desarrollar un esquema que minimice los problemas de lógica. Cada regla está basada en la que le antecede. La normalización se adoptó porque el viejo estilo de poner todos los datos en un solo lugar, como un archivo o una tabla de la base de datos, era ineficiente y conducía a errores de lógica cuando se trataban de manipular los datos.

La normalización también hace las cosas fáciles de entender. Los seres humanos tenemos la tendencia de simplificar las cosas al máximo. Lo hacemos con casi todo, desde los animales hasta con los automóviles. Vemos una imagen de gran tamaño y la hacemos más simple agrupando cosas similares juntas. Las guías que la normalización provee crean el marco de referencia para simplificar una estructura de datos compleja.

Otra ventaja de la normalización de base de datos es el consumo de espacio. Una base de datos normalizada ocupa menos espacio en disco que una no normalizada. Hay menos repetición de datos, lo que tiene como consecuencia un mucho menor uso de espacio en disco. El proceso de normalización tiene un nombre y una serie de reglas para cada fase. Esto puede parecer un poco confuso al principio, pero poco a poco se va entendiendo el proceso, así como las razones para hacerlo de esta manera.



Grados de normalización

Existen básicamente tres niveles de normalización: primera forma normal (1NF), segunda forma normal (2NF) y tercera forma normal (3NF). Cada una de estas formas tiene sus propias reglas. Cuando una base de datos se conforma a un nivel, se considera normalizada a esa forma de normalización. No siempre es una buena idea tener una base de datos conformada en el nivel más alto de normalización, puede llevar a un nivel de complejidad que pudiera ser evitado si estuviera en un nivel más bajo de normalización.

En la tabla siguiente se describe brevemente en que consiste cada una de las reglas, y posteriormente se explican con más detalle.

Tabla 30. Grados de normalización.

Regla	Descripción
Primera Forma Normal (1FN)	Incluye la eliminación de todos los grupos repetidos.
Segunda Forma Normal (2FN)	Asegura que todas las columnas que no son llave sean completamente dependientes de la llave primaria (PK).
Tercera Forma Normal (3FN)	Elimina cualquier dependencia transitiva. Una dependencia transitiva es aquella en la cual las columnas que no son llave son dependientes de otras columnas que tampoco son llave.

Por lo mencionado anteriormente, se analizó y se encontraron en las entidades generales que aún no contaban con la normalización puesto que existían campos repetidos y no se contaban con llaves primarias, y no estaban identificados los 3 niveles en 1FN, 2FN, 3FN.

Como habíamos avanzado en su momento, un esquema conceptual basado en el modelo Entidad- Relación puede ser transformado, de acuerdo con unas sencillas reglas, en un esquema lógico, basado en el modelo relacional, y manipulable por un SGBD. A continuación veremos cuáles son esas reglas. El ejemplo de partida será el mismo que habíamos utilizado al presentar el modelo ER.

Reglas de transformación.

1. Por cada tipo de entidad fuerte E del esquema ER se crea una relación R que contenga todos los atributos simples y no multivaluados de E. Además, dado que el modelo relacional no admite los valores no atómicos, R contendrá también sólo los atributos simples que formen parte de cada atributo compuesto (no multivaluado) de E. Como clave primaria de R se escogerá el atributo o atributos simples que formen parte de la clave primaria de E. Los atributos derivados se ignoran.



2. Por cada tipo de entidad débil E del esquema ER se crea una relación R que contenga todos los atributos simples y no multivaluados de E. Además, dado que el modelo relacional no admite los valores no atómicos, R contendrá también sólo los atributos simples que formen parte de cada atributo compuesto (no multivaluado) de E. Como clave primaria de R se escogerá el atributo o atributos simples que formen parte del discriminante de E, además de la clave primaria del tipo de entidad fuerte E del que dependa E. Los atributos derivados se ignoran.
3. Por cada tipo de relación (de grado 2) R del esquema ER, de cardinalidad 1:1, se identifican a las relaciones S y T del esquema relacional que representan a los tipos de entidad participantes. Se escoge una de las dos relaciones (por ejemplo S) y se incluye como clave foránea de S la clave primaria de T. Además, se incluyen en S todos los atributos (no multivaluados) del tipo de relación R, incluidos aquellos que conformen un atributo compuesto. Los atributos derivados se ignoran.

Nota: para escoger la relación S en la que incluir los atributos, es mejor pensar en una que corresponda a un tipo de entidad con participación total en el tipo de relación a representar.
4. Por cada tipo de relación (de grado 2) R del esquema ER, de cardinalidad 1:N, se identifica a la relación S que representa al tipo de entidad participante del lado N, y a la relación T que representa al tipo de entidad participante del lado 1. Se incluye como clave foránea de S la clave primaria de T. Se incluyen también en S los atributos (no multivaluados) del tipo de relación, incluidos aquellos que conformen un atributo compuesto. Los atributos derivados se ignoran.
5. Por cada tipo de relación (de grado 2) R del esquema ER, de cardinalidad M: N, se crea una nueva relación S que tendrá como atributos de clave foránea los atributos que formen la clave primaria de los dos tipos de entidad participantes en R. Además, S incluirá los atributos simples (no multivaluados) del tipo de relación, incluidos aquellos que conformen un atributo compuesto. La clave primaria de S estará formada por los atributos de clave primaria de los tipos de entidad participantes en el tipo de relación. Los atributos derivados se ignoran.
6. Por cada atributo multivaluado correspondiente a un tipo de entidad o tipo de relación R del esquema ER, se crea una nueva relación S que tendrá como atributos de clave foránea los atributos de clave primaria de R. Además, S incluirá el atributo multivaluado; si el atributo es compuesto, se incluirán los atributos simples que lo integren. La clave primaria de S será la misma que la de R, unida al atributo multivaluado.



Con todo lo anterior una vez que nuestras tablas quedaron normalizadas y realizando todos los pasos antes mencionados, el diagrama de entidad relación quedó de la siguiente forma.

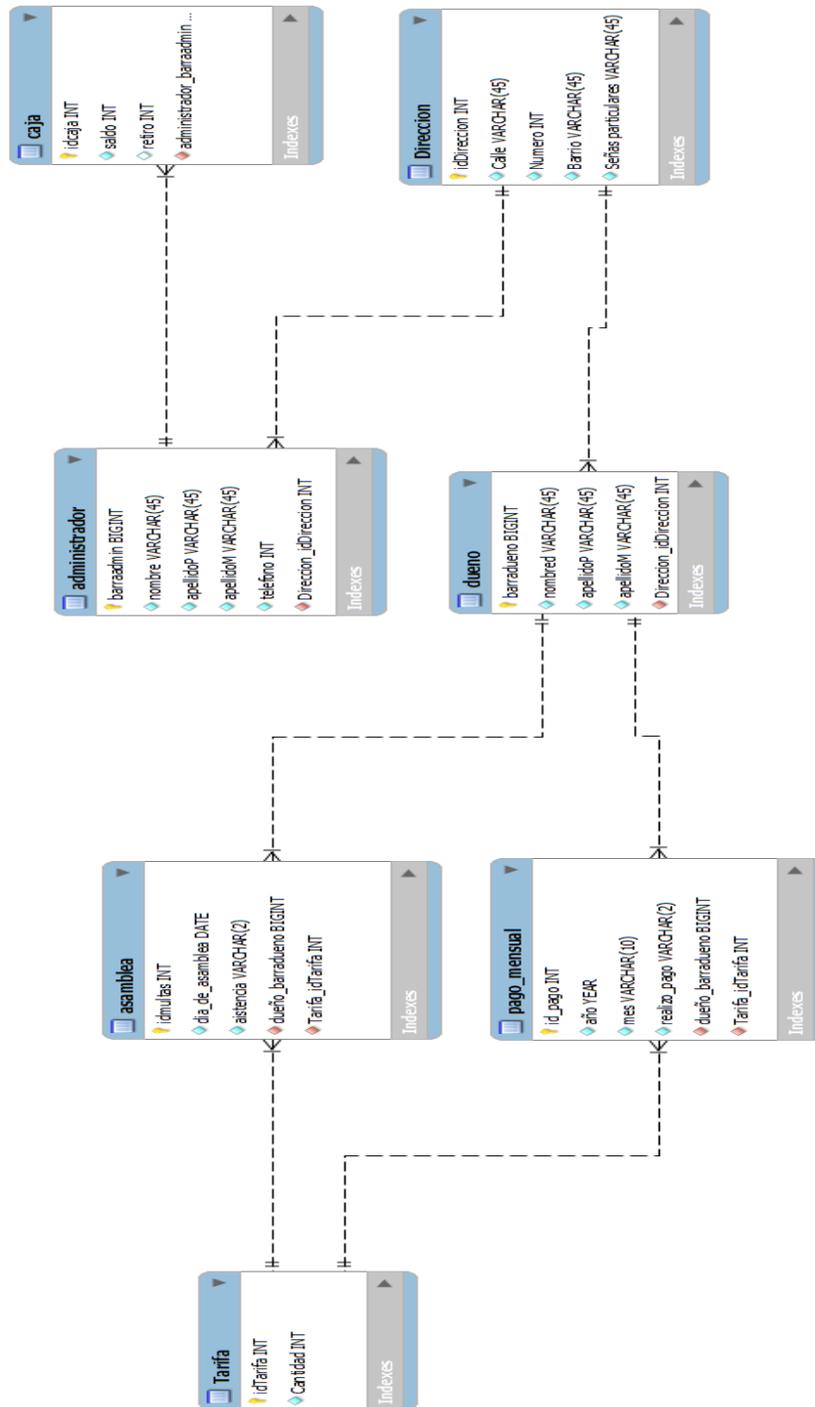


Figura 30. Diagrama relacional del SI de agua potable.



En la figura (30), se presenta el modelo final del diagrama relacional del SI, de nuestro sistema, el cual se puede visualizar con mayor apreciación las relaciones entre tablas principales con las entidades débiles, y las relación como patas de gallos de uno a varios (1:M), el cual se identifica por una pata de gallo de tres dedos o líneas en la cual se hereda una llave foránea a cada tabla que apunta esto es para uso práctico al momento de realizar las consultas en nuestra aplicación de escritorio.

2.5.6. Diccionario de datos.

El diccionario de datos permite implementar una base de datos en cualquier gestor de base de datos ya que incluye todo el esquema de una base de datos (metadatos).

El diccionario de datos del proyecto se encuentra en el anexo, en él se incluye todas las entidades y relaciones entre ellas así como los tipos y características de cada campo que conforman las entidades.

2.6. Codificación del sistema

La codificación del sistema se realizó con forme evolucionaba el proyecto, al terminar un primer módulo se procedía a verificar si cumplía o no con los requerimientos de usuario y se procedían hacer las pruebas necesarias para validar su funcionalidad, al terminar con esto se procedía a avanzar en la codificación del sistema con los demás módulos.

A continuación se presentan los aspectos generales de la codificación que incluye las características de la tecnología a utilizar y la documentación de código conformado por: el código de diseño, código de interacción y validación (cliente).

2.6.1. Tecnologías utilizadas

Las tecnologías a utilizar fueron definidas por el desarrollador del sistema, tomando en cuenta las tecnologías más eficientes para una aplicación nativa, estas tecnologías se acoplan a las necesidades del cliente y a los conocimientos técnicos del desarrollador. A continuación se describe cada tecnología utilizada.



2.6.2. Java

Es un lenguaje de programación de propósito general, concurrente, orientado a objetos y basado en clases que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible, las aplicaciones de Java son generalmente compiladas a bytecode (clase Java) que puede ejecutarse en cualquier máquina virtual Java (JVM) sin importar la arquitectura de la computadora subyacente.

2.6.3. Swing

Swing es una biblioteca gráfica para Java. Incluye widgets para interfaz gráfica de usuario tales como cajas de texto, botones, desplegables y tablas. Es un Framework MVC para desarrollar interfaces gráficas para Java con independencia de la plataforma. Sigue un simple modelo de programación por hilos, y posee las siguientes características principales:

Independencia de plataforma.

- Extensibilidad: es una arquitectura altamente particionada: los usuarios pueden proveer sus propias implementaciones modificadas para sobrescribir las implementaciones por defecto. Se puede extender clases existentes proveyendo alternativas de implementación para elementos esenciales.
- Personalizable: dado el modelo de representación programático del Framework de swing, el control permite representar diferentes estilos de apariencia "look and feel" (desde apariencia MacOS hasta apariencia Windows XP pasando por apariencia GTK+, IBM UNIX o HP UX entre otros). Además, los usuarios pueden proveer su propia implementación de apariencia, que permitirá cambios uniformes en la apariencia existente en las aplicaciones swing sin efectuar ningún cambio al código de aplicación.

2.6.4. MySql

MySql es un sistema gestor de base de datos que permite multiusuario, es un sistema que tiene amplia cobertura en internet lo cual provoca que el soporte sea igual de grande.

MySql permite la conexión entre él y algunos lenguajes de programación (Java) para producir aplicaciones con accesos a datos de manera rápida y sin errores.



2.6.5. iReport

iReport es el diseñador libre, abierta la fuente de información para JasperReports y JasperReports server. Crear diseños muy sofisticados que contienen gráficos, imágenes, sub informes, tablas de contingencia y mucho más. Acceda a sus datos a través de JDBC, TableModels, JavaBeans, XML, hibernate, CSV y fuentes personalizadas. Publica los informes en PDF, RTF, XML, XLS, CSV, HTML, XHTML, texto, DOCX, u Open Office.

2.6.6. Documentación de código

En la siguiente tabla se presenta la documentación del código fuente que mantiene el sistema de información.

Tabla 31. Documentación de código

Documentación de código						
CLASES	LÍNEA DE CÓDIGO					OBSERVACIONES
	JAVA	SWING	MYSQL	IREPORT	TOTAL	
Main	19	0	0	0	19	
Modelo	75	0	35	0	75	
Conexión	125	0	38	0	163	
Administrador Bean	44	0	0	0	44	Un bean es usado para la manipulación de registros BDs en java.
Usuario Bean	40	0	0	0	40	
Código de Barra	163	0	0	0	163	
Controlador	121	0	0	0	121	
Acerca de	71	34	0	0	105	
Administración	515	761	25	97	1398	Más funciones para el Adm.
Ventana Principal	96	219	0	0	315	
Usuario	331	257	13	19	620	swing es la parte visual
Acceso	84	71	0	0	155	
Total	1684	1342	111	116	3218	Aproximado

Las líneas mostradas en la tabla 31 solo son las contabilizadas desde cero, es decir las que forman parte del núcleo fundamental de la aplicación. En dicha estadística se contabiliza todas las líneas de programación desarrolladas en esta primera versión del prototipo sin tomar en cuenta las especificaciones de las librerías externas para mejoramiento visual de la aplicación (Edisoncorx), y las líneas de código para llevar a cabo la impresión de pantallas así como la importación de archivos.



2.7. Diseño de Interfaz

La parte del desarrollo grafico de la interfaz es tratado en esta sección, el diseño del sistema está dividido en seis apartados (Figura 28), comenzando por el diseño de la interfaz de usuario, siguiendo con el diseño estético y de contenido para pasar al diseño de navegación y arquitectónico de la aplicación de escritorio y finalizando con el diseño de componentes.

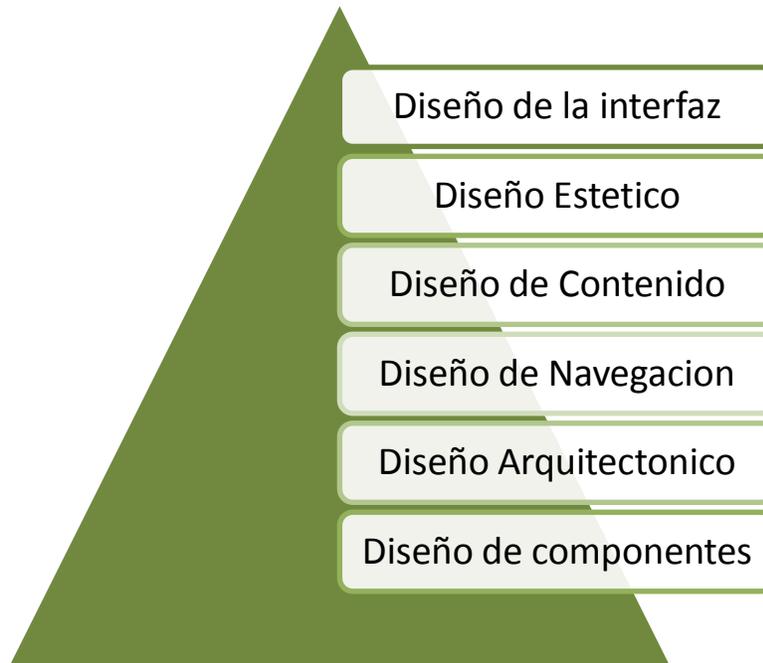


Figura 31. Pirámide de diseño de interfaz

2.7.1. Diseño de la interfaz de usuario

Para esta fase se comenzó realizado un bosquejo para representar la posible interfaz de la aplicación, en un esquema inicial de esta interfaz la podemos observar en la siguiente figura (32).



Figura 32. Interfaz de usuario

Para cumplir con los requisitos de la interfaz, por parte del usuario en donde deben de identificarse para poder acceder al sistema por medio de una identificación con código de barras, en esta pantalla de inicio se cumplirán los requerimientos (CU1, CU7), para esta primer pantalla de inicio dependerá de mucho el que tipo de usuario sea el que va a acceder al sistema (administrador, dueño de toma), ya que el menú para cada uno cambiara de acuerdo a sus privilegios en el sistema. Para los dos usuarios lo único que podrán ver será la cuestión de ayuda y acerca de, que son menús que siempre estarán activados.

En el esquema mostrado en la Figura 32, se muestra este primer borrador de la interfaz de inicio en donde, observamos que existe un menú principal, que será donde se localizaran los botones de acceso, cierre de sesión y acceso por parte del usuario, como estamos hablando de una aplicación de escritorio, podemos tener en cuenta que para el acceso al sistema se tiene que abrir una ventana emergente para identificarse y aprobar que pertenece el usuario al sistema, dependiendo el tipo de usuario vemos que también tenemos un menú para la parte de administración, que es donde podrá desempeñar todas las funciones y privilegios que le son otorgados al administrador en este sistema, y por ultimo hemos de mencionar que existe una imagen principal esto es para ser más amigable el sistema con el usuario y de igual forma pretendemos que los botones de acceso cuenten con una imagen relacionada con la función que desempeñara. Del esquema anterior se produjo el primer diseño de la interfaz de inicio (Figura 33).

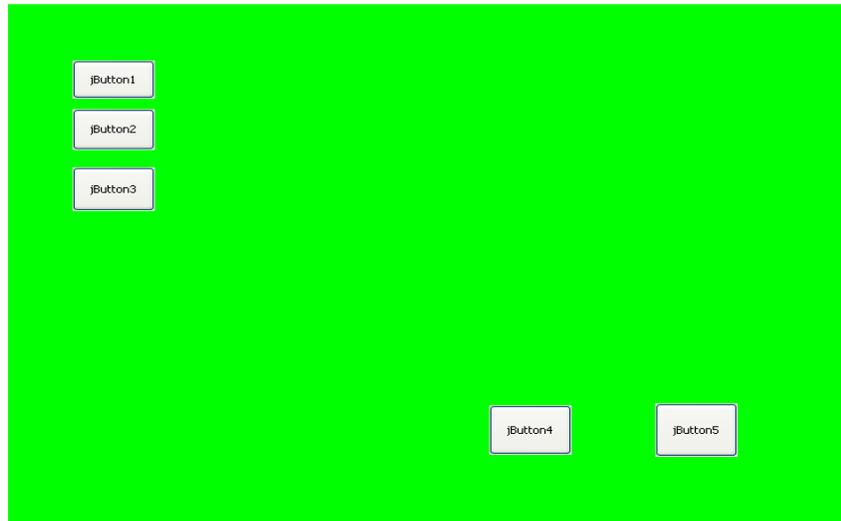


Figura 33. Primer diseño de la interfaz (pantalla de inicio).

Como hemos de saber el diseño de las interfaces varían de acuerdo al tipo de usuario esto es, que las interfaces estarán cambiando y mostrando nuevas ventanas y funciones según el sistema identifique al tipo de usuario, a continuación veremos un bosquejo inicial que representara la vista por parte del administrador en la siguiente Figura 34.



Figura 34. Bosquejo inicial de la interfaz del administrador.



Para cumplir con los requisitos plasmados en las descripciones de los casos de uso del administrador (CU8 – CU21), vemos en esta interfaz para el administrador , las funciones plasmadas en una barra de menú en donde se muestran varias ventanas internas que realizaran las funciones, como lo son la administración del usuario, y administración en donde se generaran las nuevas credenciales, modificaciones y eliminación de un usuario en específico (CU11, CU20), en la siguiente figura vemos una primera versión de la interfaz por parte del usuario administrador (Figura 35).

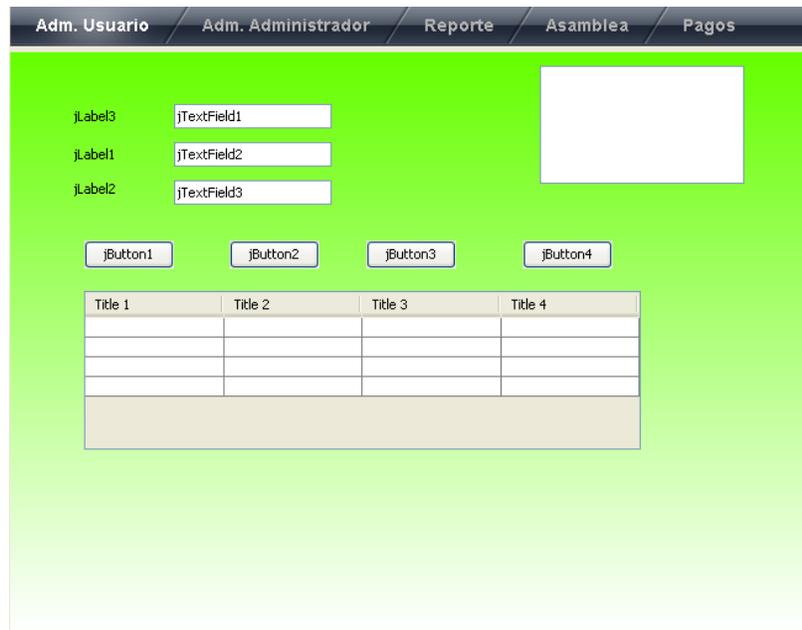


Figura 35. Primer diseño de la interfaz administrador.

Lo que apreciamos en la Figura 35, es la barra de menú que nos da la opción de navegar y seleccionar la función que quisiera desempeñarse, en el caso de administración de usuario vemos un borrador inicial de esta ventana en donde se muestra un formulario en donde se llenaran datos del usuario esto es para generar su registro y generar su credencial, igual se puede ver en la parte de abajo una tabla que es donde se visualizara el nuevo registro que se llevó a cabo o el historial de esta ventana, en la parte del recuadro blanco en la parte superior derecha, es donde se estará generando el código de barras.



Siguiendo con la representación de interfaces, se presenta a continuación la posible interfaz de usuario – dueño de toma de agua, en la siguiente Figura 36.



Figura 36. Bosquejo inicial de la interfaz de dueño de toma.

En esta interfaz se pretende cumplir con todos los requerimientos por parte del dueño de toma, en donde podemos apreciar una barra de menús en la cual nos ayudaran a navegar por ella pudiendo consultar los adeudos y atrasos de un usuario en específico (CU2, CU5), y esta interfaz será exclusiva para cada usuario que necesite acceder al sistema.

En la figura 34, se muestra un primer diseño (borrador), en el cual queda la barra de menús en la cual se implementan las funciones del usuario como lo son (CU3- CU5) generar reporte, consulta de atrasos, consulta de adeudo, en esta ventana se cumplen con todos los requerimientos por parte del dueño de toma de agua potable, y en la cual se pretende que la interfaz sea fácil, y amigable en su manejo, en donde no se requiera de sofisticados pasos o actividades a realizar para cualquier actividad, como ya hemos mencionado el sistema está pensado para realizarse en un ATM o algo similar a esto, y tomando en cuenta este punto , el sistema está pensado para un sistema touch creen, esto para ser más interactivo el sistema y de fácil uso , implementando imágenes en los botones que vayan relacionados con sus funciones, como ejemplo imprimir reporte , poner una impresora. Así de sencillo será el uso y navegación en el sistema.

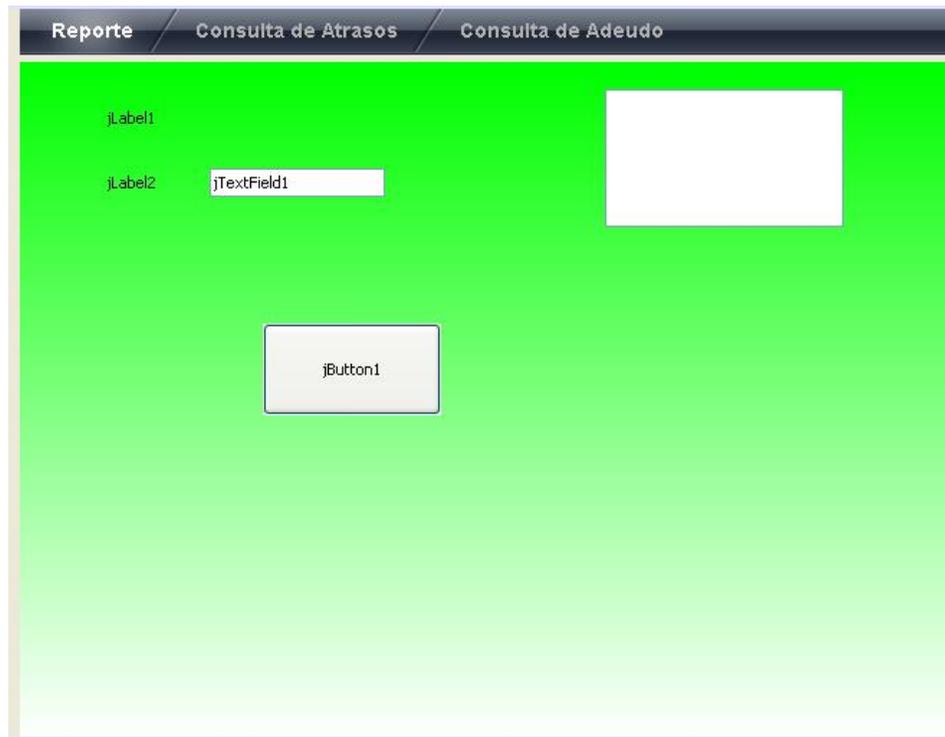


Figura 37. Primer diseño de la interfaz de dueño de toma.

2.7.2. Diseño estético o diseño gráfico del Sistema.

(Pressman, 2005), define dos aspectos importantes dentro del diseño estético:

- Plantilla, donde se tiene que:
 - Resaltar el contenido
 - Ordenar los elementos de arriba a la izquierda hacia abajo a la derecha.
 - No extender el contenido (en la medida de lo posible).
 - Considerar el tamaño y resolución de la pantalla.
- Diseño gráfico, donde se consideran:
 - Paleta de colores
 - Tamaños y fuentes
 - Elementos que complementen la plantilla (imágenes, audio, video, etc.)

En la realización de las interfaces se consideró la creación y seguimiento de un diseño estético en la aplicación, considerando las pantallas a mostrar y su orden en todas ellas creando plantillas propias puesto que estamos hablando de una aplicación de escritorio (nunca antes realizada) es difícil definir un estilo para todas ellas, por ello se dejaron valores por default en la mayoría de ellas.



Teniendo esto en consideración y tomando la estructura de la interfaz (Figuras 28- 33), la definición del diseño estético del sistema queda de la siguiente forma.

Botones de inicio principal.

Imagen: 128 px 128 px.
Tipo de fuente: Félix, Titling.
Tamaño fuente: 10px.
Color de fuente: [236, 233,216].

Menú de inicio administración.

Tipo de fuente: arial, bold.
Tamaño de fuente: 14px
Color de fuente: [255, 255,255].

Menú de inicio dueño de toma.

Tipo de fuente: arial, bold.
Tamaño de fuente: 14px
Color de fuente: [255, 255,255].

Tablas de consultas y registros.

Tipo fuente: tahoma plain.
Tamaño de fuente: 11px
Color de fuente: [0, 0,0].
Color de fondo: [255, 255,255].

Formularios, (etiquetas, botones, combos, cajas de texto).

Tipo fuente: arial, bold.
Tamaño de fuente: 14px
Color de fuente: [0, 0,0].
Color de fondo: [236, 233,216]
Bordes: xpsfillborder.



2.7.3. Diseño arquitectónico

La arquitectura que más se adapta a nuestros requerimientos es un sistema centralizado, en donde contamos con un equipo que va a realizar la función del servidor, que este a su vez cuenta con toda nuestra aplicación que estará en espera de recibir solicitudes, para que pueda procesarlas y enviar los resultados al usuario. En la siguiente Figura 38 muestra de manera clara esta arquitectura y los módulos que se manejan.

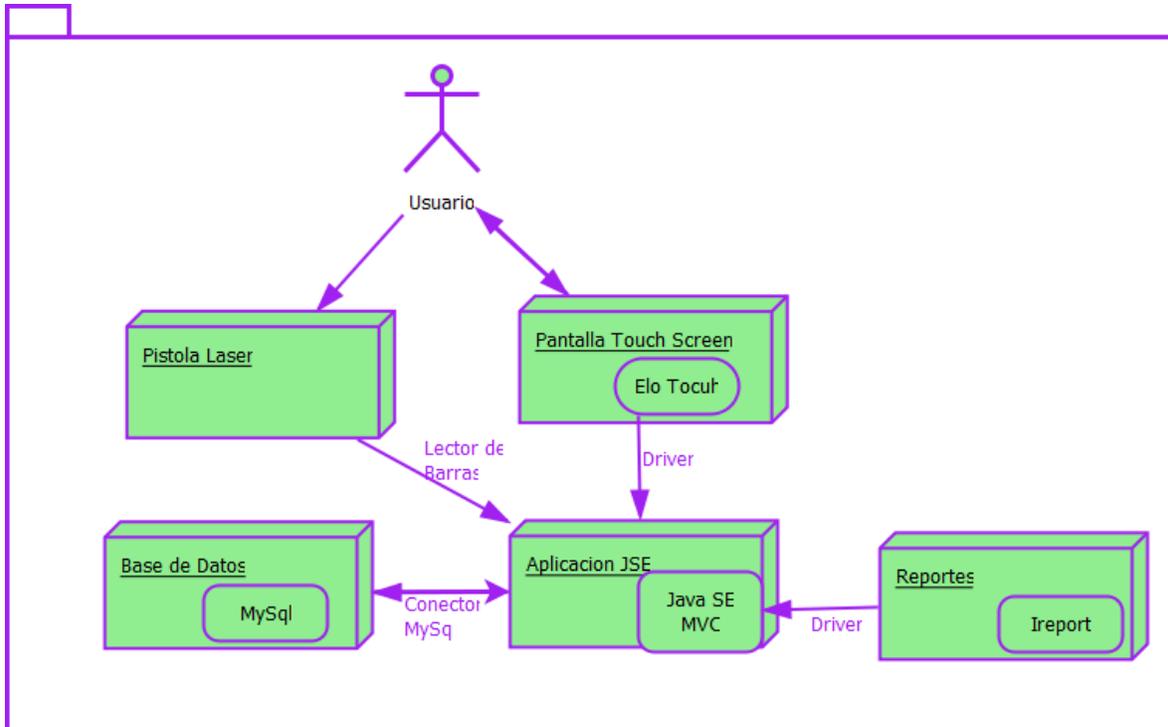


Figura 38. Arquitectura sistema centralizado.



2.8. Diseño de navegación

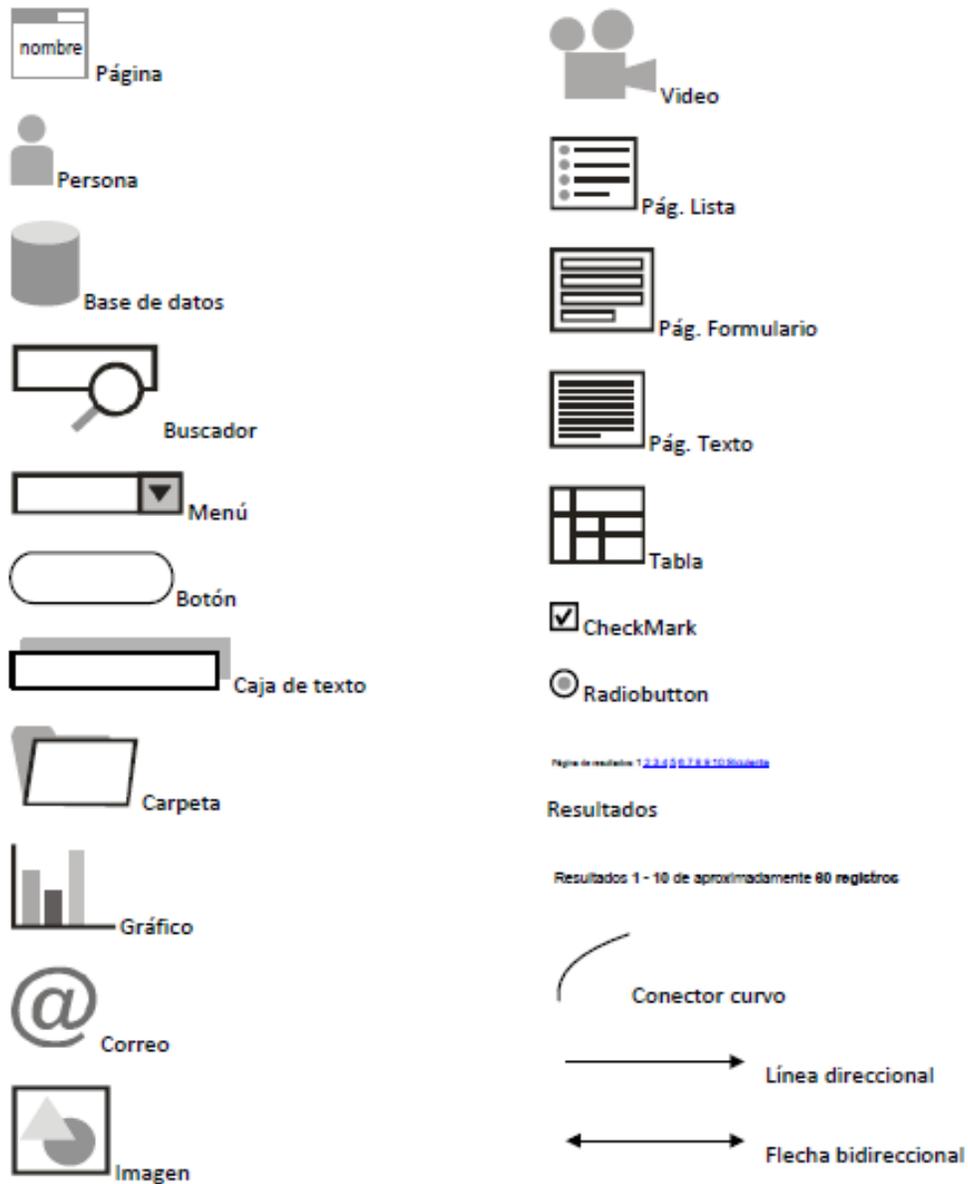


Figura 39. Símbolos de maquetación (Aguilar Juárez, 2011).

En la Figura 39, se muestra una lista de símbolos utilizados para la maquetación y la representación de la navegación.



2.8.1. Plan de navegación con símbolos de maquetación.

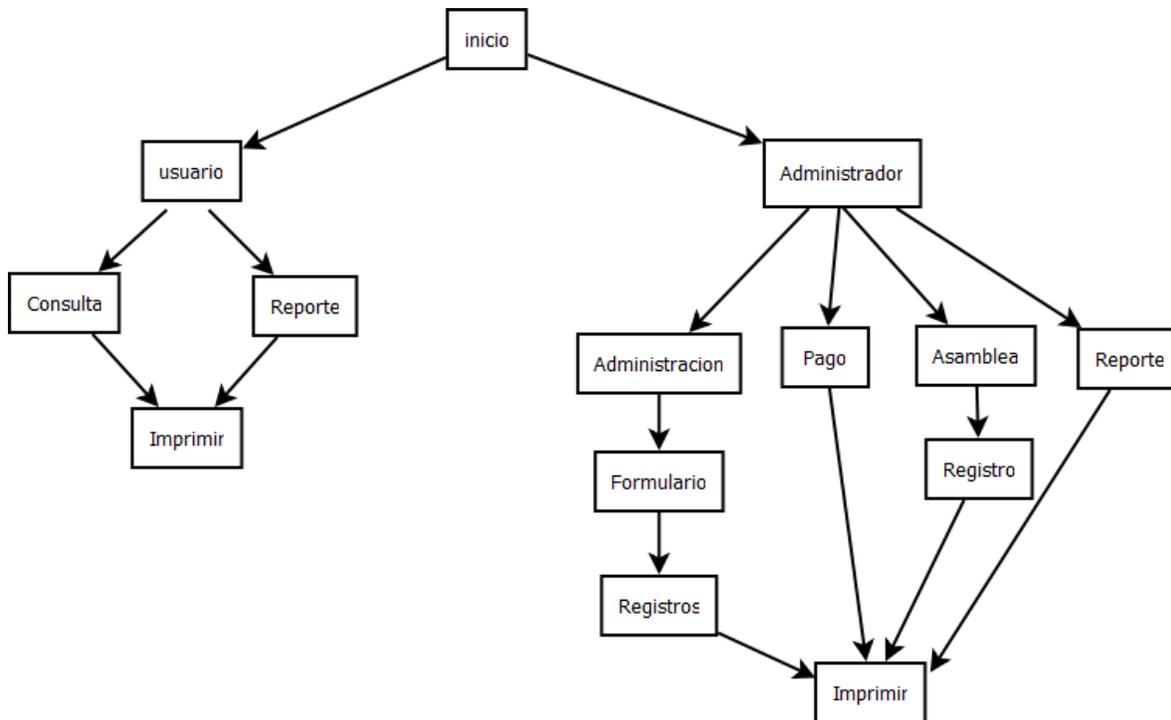


Figura 40. Plan de navegación maquetación.

Para efectos de análisis, el plan de navegación se dividió en dos posibles caminos principales, los cuales se definirán de acuerdo al tipo de usuario que acceda al SI esto es: Administrador o Usuario, los cuales como lo apreciamos en la siguiente Figura 40, el usuario cuenta con dos principales rutas y el administrador tiene tres principales rutas de navegación. Estos caminos son los siguientes:

En la pantalla de inicio de nuestro sistema contamos con un menú que se encontrara inactivo mostrando las opciones para poder acceder al menú de usuario o administrador, esto dependerá del acceso y privilegios que cuente su tarjeta de identificación, para validar el usuario contara con una ventana emergente para poder hacer lectura de su tarjeta y poder ingresar al sistema y desactivar los privilegios que tenga.

Ruta usuario

Al estar activo con una cuenta como usuario, este tendrá dos rutas diferentes para navegar en el sistema, estas son por medio de consultas, o por medio de genera reporte, las cuales son al acceder y entrar al menú de consultas el usuario podrá verificar su historial negativo que cuenta en sus atrasos mensuales mediante una tabla que contendrá todo este historial,



así como sus multas que cuente por no hacer acto de presencia en las juntas colectivas que organiza el comité de agua potable, contando con la opción de imprimir estos reportes por separado cada uno y ver el monto que debe por las razones separadas y este mismo comprobante funcionara como referencia al momento de hacer su liquidación al comité

La segunda ruta es donde el usuario contara con la opción mandar directamente en un solo reporte el historial de el mismo y contar con este comprobante e imprimirlo o bien guardar un respaldo de este en el equipo de cómputo.

Ruta administrador

El administrador es el que cuenta con más privilegios y funciones dentro de nuestro sistema contando con más de 3 rutas que puede seguir, que a continuación se describirán;

Administración

El administrador cuenta con la opción de registrar a nuevos usuarios al sistema ya sea tipo administrador o tipo dueño de toma, esto se realiza mediante el llenado de un formulario muy práctico, cabe mencionar que en esta parte si es necesario el uso del periférico del teclado para hacer llenado de este formulario, posteriormente a este llenado se procede a generar el código de barras para poder hacer el registro completo, y una vez realizado el proceso con éxito se deberá de hacer la generación de la credencial mediante una impresión del mismo.

En esta misma pantalla de navegación el administrador cuenta con las opciones de poder modificar los registros de un usuario en específico o bien hacer la eliminación por completo del sistema de alguien en específico.

Pagos

Por medio de los comprobantes que el usuario cuenta, por medio de las consultas o bien al generar su reporte, el administrador goza con el privilegio de hacer limpieza de este historial negativo del usuario, una vez que el usuario a liquidado su adeudo con el comité de agua potable y multas que cuente, mediante esta pantalla de navegación se cuenta con la opción de liquidación de cuenta de un usuario en específico haciendo uso de la pistola lectora de barras para poder entrar al historial del usuario y validar el adeudo actual y el adeudo una vez realizado los pagos correspondientes.



Asamblea

Esta navegación comienza cuando el administrador entra a su menú principal y accede a la opción de asamblea, el sistema cuenta con la innovación de llevar a cabo el registro de acceso a los usuarios a las diferentes asambleas que son llevadas a cabo de manera aleatoria, el sistema toma la fecha actual del día y realiza el alta de una nueva sesión de asamblea, el administrador es el único que puede llevar a cabo esta función.

Realizando el registro de las secciones por medio de un botón, pensando en el uso de nuestro sistema interactivo touch-creen, en esta parte no es necesario el uso de ningún otro dispositivo , solo bastara con la navegación del toque, el administrador cuenta con las opciones de poder cerrar la cesión del día, o bien poder cancelar la sesión, en los casos que no se llegase a empezar la asamblea y esta se cancele, el registro de los usuario es muy práctico y fácil de realizar solo bastara con hacer nuevamente uso de la pistola laser para registrar el acceso a las instalaciones y de esta forma tomar la asistencia correspondiente.

Reporte

La ultima ruta que formara la navegación completa del administrador y del sistema, es en la parte de generar reporte, esto es para ayudar a sacar datos estadísticos que ayuden a la administración que lleva el comité del agua potable, en esta parte de la navegación el administrador cuenta con un formulario en el caso que desee generar un reporte más específico, o bien puede hacer el reporte de todos los deudores pendientes que cuenta el sistema hasta ese día.



2.8.2. Plan de navegación Usuario



Figura 41. Diseño de navegación de la ruta de usuario.

En la Figura 41 se muestra el plan de navegación que puede seguir un usuario al entrar al sistema con sus privilegios de dueño de toma de agua potable en donde el usuario podrá realizar sus consultas de atrasos y faltas en el sistema con un solo clic o tocar la pantalla, y en el otro caso generar su reporte de ambos casos, cabe mencionar que en cualquiera de las dos rutas disponibles, se encuentra la opción de poder realizar la impresión del historial que se encuentre en pantalla visualmente, como ya se había mencionado en la Figura 39.



2.8.3 Plan de navegación administración.

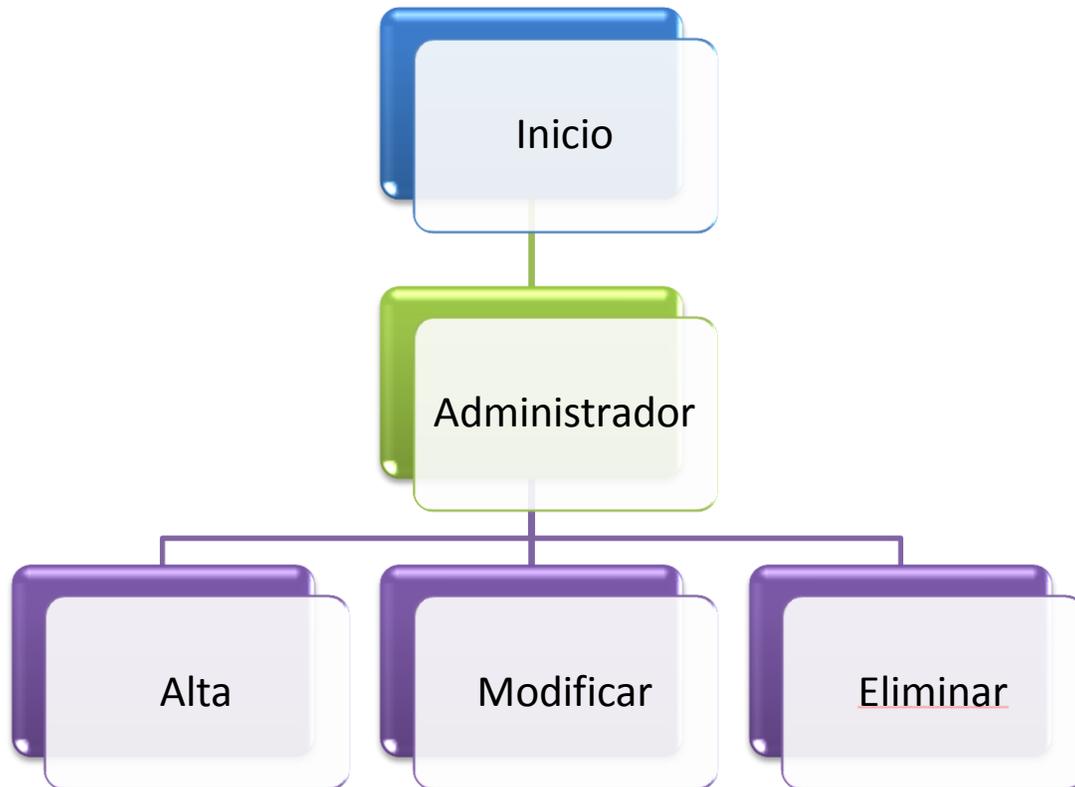


Figura 42. Plan de navegación para la administración.

La Figura 42, muestra el plan de navegación para la administración, una vez que el sistema ha detectado que el usuario que accedió es de tipo administrador, y que este a su vez acceda en el menú de administrador y necesite administrar a los usuarios del sistema, en donde se llevara a cabo las altas de nuevos usuarios y generar las credenciales para que puedan acceder al sistema, así como el poder realizar cambios o incluso eliminarlos del sistema como se explicaba en la representación de la Figura 39.



2.8.4. Plan de navegación para pago.



Figura 43. Diseño de navegación, para realizar pago.

La Figura 43 representa el diseño de navegación para realizar los pagos, por medio del administrador, cabe mencionar que para poder navegar en esta pantalla es necesario contar con un reporte generado anteriormente por parte del usuario en donde este reporte contara con los datos suficientes para que el administrador sepa qué hacer con esa información, esta es la representación de pago de la Figura 40.



2.8.5 Plan de navegación para asamblea.

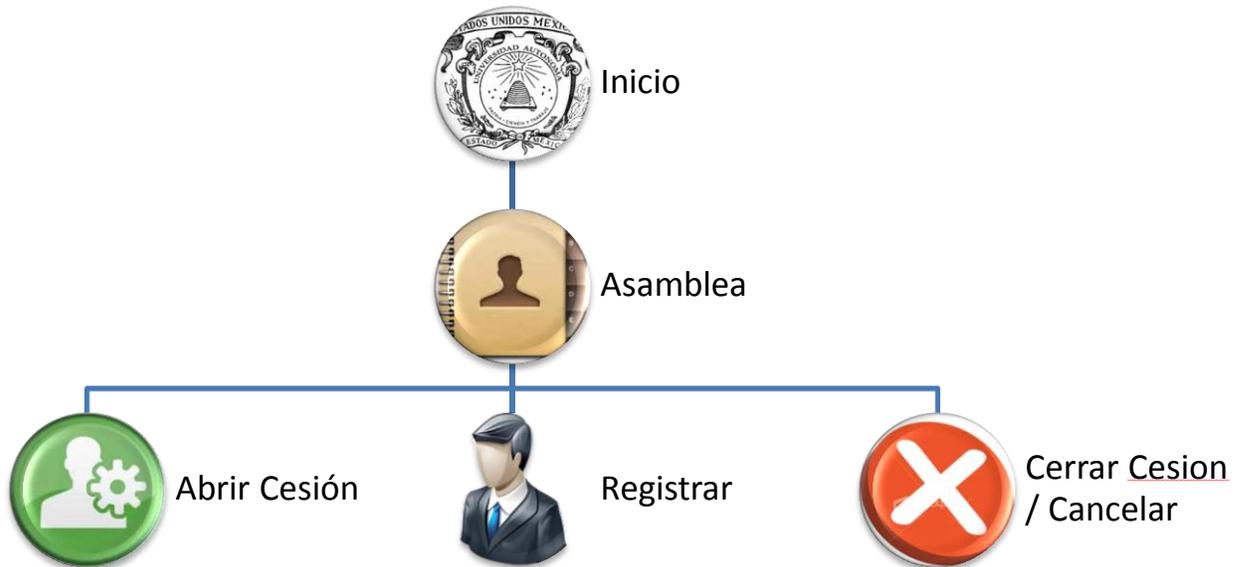


Figura 44. Plan de navegación para asamblea

La Figura 44, muestra el plan de navegación para asamblea, donde el administrador será capaz de llevar a cabo la apertura de una cesión, el registro de acceso y el cierre o cancelar la petición de apertura.

Este proceso de registro será muy práctico de realizar debido al uso de la pistola laser o lector laser el cual su función es facilitar el registro al sistema, de una forma más rápida y confiable, esta es la representación en pantallas de la representación de la Figura 39.



2.8.6. Plan de navegación para generar reporte.

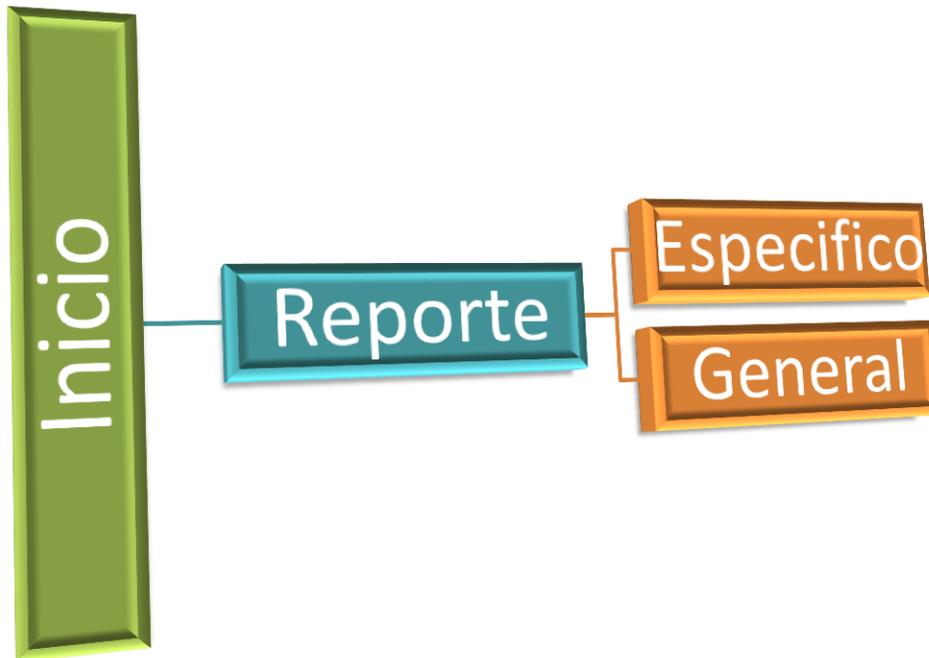


Figura 45. Plan de navegación para generar reporte.

La Figura 45 muestra el plan de navegación para generar reportes por parte del administrador el cual deberá de haber accedido como tal, y entrar al menú de reportes, en la opción de generar reporte específico el administrador contará con un formulario interactivo para poder realizar una búsqueda más específica, en cualquiera de estas dos rutas posibles al generar el reporte cuenta con la opción de poder mandarlo a imprimir para tener el historial estadístico para el comité de agua potable.



2.9. Pruebas del sistema de Información.

Las pruebas que se realizaran a continuación son en base a la siguiente figura.

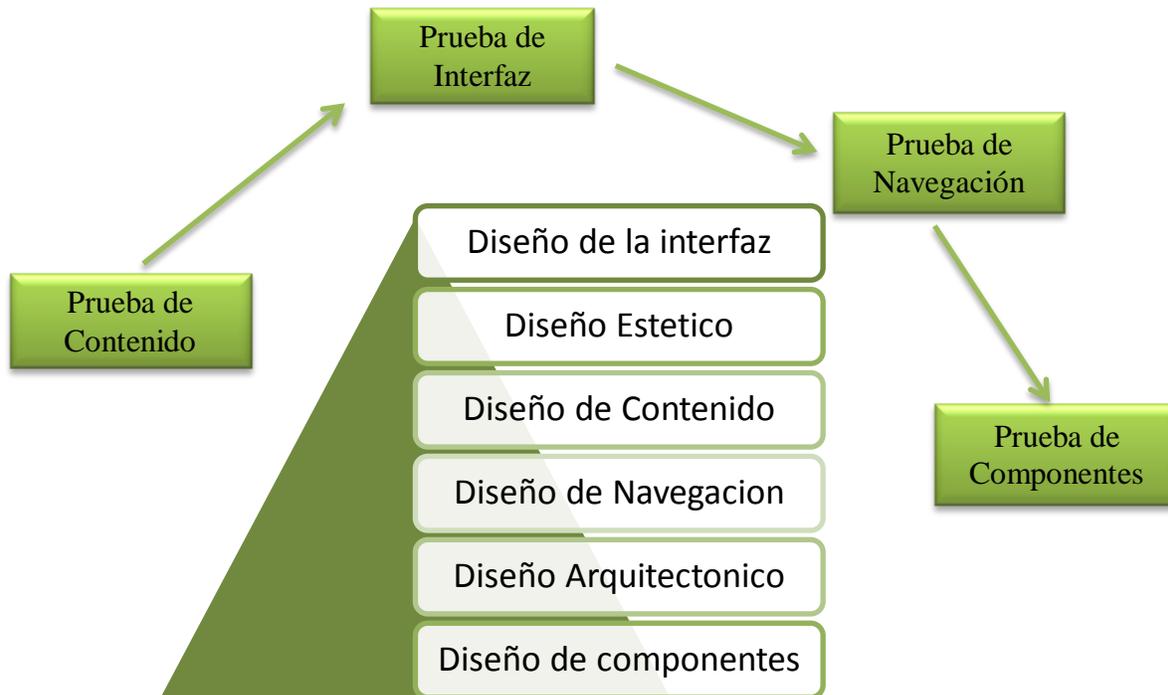


Figura 46. Proceso de pruebas.

Las pruebas de configuración, interfaz y contenido se realizaran en conjunto para identificar fallas de tipografía, gramática y ortografía tanto en la aplicación como en la base de datos. La codificación del sistema será probada y documentada en esta sección en busca de fallas y elementos que se puedan mejorar (Figura 46).

Las pruebas de navegación se realizaran para verificar el cumplimiento de los requisitos del cliente. Finalmente, la seguridad y desempeño se probará para optimizar el sistema y el acceso a la base de datos.

Cabe recordar que las pruebas se realizaron conforme se fueron desarrollando los incrementos para de esa manera refinarlos y ahorrar tiempo, para esta documentación se presentan las pruebas finales del sistema.



2.9.1. Pruebas de Contenido

Las pruebas realizadas al contenido resultaron favorables de acuerdo a la siguiente tabla, esta tabla está formada en base a los objetivos y preguntas formuladas por (Pressman 2005) para la realización de pruebas de contenido. A partir de este punto en cada prueba se marcara con una (*) para los aspectos que se cumplen, con una (X) los elementos que no se cumplen y con un guion (-) las situaciones que no se aplican en dichas pruebas.

Tabla 32. Pruebas de contenido.

PRUEBA DE CONTENIDO										
PRUEBA	SISTEMA		GRAFICA		IMPRESIÓN		EXPORTACIÓN		BASE DE DATOS	
	SI	NO	SI	NO	SI	NO	SI	NO	SI	NO
La gramática es correcta	*		*		*		*		*	
La ortografía es correcta	*		*		*		*		*	
La información es precisa	*		*		*		*		*	
Se violan derechos de autor		X		X		X		X		X
Existen conflictos estéticos		X		X		X		X		X
El contenido se muestra en el lugar preciso	*		*		*		*		-	
El contenido se muestra accesible	*		*		*		*		-	
La comunicación con la base de datos es correcta	*		*		-		*		*	
El texto obtenido de la base de datos es correcto	*		*		-		*		*	



2.9.2. Pruebas de la interfaz del sistema.

La revisión de la interfaz comenzó desde la formulación del proyecto en el análisis de los requisitos y al finalizar el desarrollo.

Las pruebas comprenden las características de la interfaz (diseño, estética, y contenido), mecanismos individuales (formatos) y otros, estas pruebas se presentan de manera detallada en la siguiente Tabla 33.

Tabla 33. Pruebas de la interfaz del sistema.

Pruebas de interfaz		
PRUEBA	SI	NO
El tipo de fuente es igual en todas las paginas	*	
Los colores (de texto, imágenes, botones, etc.) son iguales en todas las páginas.	*	
Los textos son comprensibles(tamaño y color adecuado)	*	
Los símbolos y acentos se muestran correctamente.	*	
los formularios (botones, cajas de texto, etc.) tienen el mismo tamaño	*	
Las tablas presentan la información de manera ordenada	*	
existe perdida de datos al imprimir comprobante		X
la interfaz es fácil de comprender	*	
Los botones realizan su función	*	
El sistema es totalmente touch		X



2.9.3. Pruebas de configuración.

Para la realización de esta prueba fue necesaria la exportación del proyecto hacia otro equipo de cómputo. El proyecto fue desarrollado en una Laptop de última generación de Sony teniendo las siguientes características:

Sistema operativo: Windows seven profesional
 Memoria ram: 4Gb
 Procesador: Intel core i3.
 Disco Duro: 500Gb.
 Netbeans: 7.3

Realizando la exportación a un equipo de escritorio personal, el cual cuenta con:

Sistema operativo: Windows XP
 Memoria ram: 1Gb
 Procesador: Intel core duo.
 Disco Duro: 320Gb.
 Netbeans: 7.3

Para lo cual los requerimientos mínimos para poder correr este SI es:

Sistema operativo: Windows XP sp3/ vista
 Memoria ram: 512 Mb
 Procesador: Intel Pentium III 800Mhz
 Disco Duro: 750Mb
 Netbeans: 7.3

Realizando las pruebas de configuración encontramos el siguiente resultado.

Tabla 34. Pruebas de configuración.

PRUEBAS DE CONFIGURACIÓN								
PRUEBA	SO		NETBEANS		WORKBENCH		IREPORT	
	SI	NO	SI	NO	SI	NO	SI	NO
EL SI es compatible	*		*		*		*	
EL SI es sensible a cambios		X		X		X		X
Los scripts funcionan	*		*		*		*	
Existe problema con el antivirus		X		X		X		X



2.9.4. Pruebas de seguridad

Las pruebas de seguridad abarcan dos áreas: el usuario y el administrador. La finalidad de las pruebas de seguridad es identificar agujeros (problemas) de seguridad en dichas partes, los resultados obtenidos de las pruebas realizadas pueden observarse en la tabla.

Los guiones indican que la prueba no se aplica para un tipo de usuario.

Tabla 35. Pruebas de Seguridad.

PRUEBAS	USUARIO		ADMINISTRADOR	
	SI	NO	SI	NO
Acceso al código fuente		X		X
Contraseñas encriptados	*		*	
ABC de usuario / administrador		X	*	
administrar asamblea		X	*	
imprimir reporte	*		*	
cuenta privilegiada		X	*	
ver historial de otros usuarios		X	*	

En general estas son las pruebas de seguridad del sistema, de manera más particular los elementos y situaciones que se revisaron fueron los siguientes:

- El prototipo está pensado para uso a nivel de interfaz, y en ningún momento se les permite modificar el código fuente del sistema, esto favorece a la seguridad del sistema y conservar la integridad.
- Como ya se ha mencionado anteriormente, se diseñaron dos tipos de cuentas para el acceso a nuestro sistema, para este paso el usuario debe de identificarse como tal, para poder acceder.
- El prototipo al no contar con un periférico como lo es el teclado, el usuario se ve limitado a realizar la navegación ya establecida.
- El usuario solo podrá realizar consultas de su estatus actual, y solo el administrador será capaz de realizar algún tipo de modificación a la cuenta.



V. Resultados

Se concluye con el prototipo de sistema de información para la administración del cobro, en el uso de los pozos de agua potable en el pueblo de san Miguel Tocuila, que facilite la obtención y comprensión de la misma fue el objetivo principal planteado para la realización del presente trabajo basado en la metodología RAD.

El objetivo general y los objetivos específicos se cumplieron conforme se ponía en práctica cada fase de la metodología RAD, el producto obtenido después de ejecutar estas etapas fue una aplicación de escritorio desarrollada con tecnologías libres como lo son Java, MySQL además de implementar una base de datos que almacena toda la información historia de los usuarios (dueños de la toma de agua) de la comunidad.

La interacción de la base de datos con el sistema se realiza por medio de código para realizar el ABC (Altas, Bajas, Consultas, Actualización), a continuación se presenta la tabla 36 donde se resume la cantidad de páginas, consultas por página y código fuente para obtener los resultados deseados, cabe aclarar que el sistema fue desarrollado con un patrón de diseño Modelo Vista Controlador (MVC), un patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario, (Trygve Reenskaug and James O. Coplien, 2013);

El modelo:

Es la representación de la información con la cual el sistema opera, por lo tanto gestiona todos los accesos a dicha información, tanto consultas como actualizaciones, implementando también los privilegios de acceso que se hayan descrito en las especificaciones de la aplicación (lógica de negocio). Envía a la vista aquella parte de la información que en cada momento se le solicita para que sea mostrada (típicamente a un usuario). Las peticiones de acceso o manipulación de información llegan al modelo a través del controlador.

El controlador:

Responde a eventos (usualmente acciones del usuario) e invoca peticiones al modelo cuando se hace alguna solicitud sobre la información (por ejemplo, editar un documento o un registro en una base de datos). También puede enviar comandos a su vista asociada si se solicita un cambio en la forma en que se presenta de modelo (por ejemplo, desplazamiento por un documento o por los diferentes registros de una base de datos), por tanto se podría decir que el controlador hace de intermediario entre la vista y el modelo.



La vista:

Presenta el modelo (información y lógica de negocio) en un formato adecuado para interactuar (usualmente la interfaz de usuario) por tanto requiere de dicho modelo la información que debe representar como salida.

De esta forma todas las consultas son representadas por la vista que son visibles para el usuario, pero existe una cantidad de código y consultas se toma en cuenta las paginas no visibles que las podemos apreciar en el (modelo, controlador), además de las consultas se implementó otros paquetes o librerías esto es para la realización de reportes con iReport y JasperReports el primero que es un entorno grafico ayuda a crear el reporte , mientras que el segundo se utilizan librerías que ayudan a compilar y ejecutar el reporte, otro paquete utilizado fue Javafx (Edisoncorx) esta librería fue más usada para la estética en la interfaz de usuario, pudiendo realizar mejoras que con la librería de Swing de java no son posibles.

Tabla 36. Tabla de resultados por página.

Pagina	Consultas	Reportes	Líneas Código	Resultado	Observaciones
Logueo	1	0	60	Haciendo uso del lector de barras, da acceso al sistema.	se contabilizo todo el código MVC
Alta Administración	3	0	65	Se muestra el formulario para ser llenado hacer un alta o cuando sea necesario modificar algún dato, se muestra la tabla de todos los usuarios registrados y en determinado caso eliminar algún registro	
Alta Usuarios	3	0	65	Se muestra el formulario para ser llenado hacer un alta o cuando sea necesario modificar algún dato, se muestra la tabla de todos los usuarios registrados y en determinado caso eliminar algún registro	las consultas se generan en la misma pagina



Reporte mensual	0	2	45	se generan dos diferentes reportes uno de ellos personal de un usuario y el otro muestra el estado actual general	
Asamblea	4	0	85	Se muestran los botones necesarios para gestionar un día de asamblea	
Pago	2	0	30	Realizando la liquidación de la cuenta en donde primero es necesario saber su estado actual para después hacer la liquidación	
Atraso	1	0	30	Muestra tabla con todos sus pendientes por pagar	Existe opción de poder imprimir pantalla si el usuario así lo desea
Faltas	1	0	25	Muestra tabla con todas sus faltas por no entrar a las asambleas	
Reporte mensual	0	1	30	Botón para generar todo el historial negativo del usuario. Y pueda ser manejado por la administración	El reporte es obligatorio que se mande a imprimir para analizar los datos

Por último la librería que se utilizo es SampleBarCode que nos permitió utilizar el driver del dispositivo pistola de láser, que es utilizada para la lectura del código de barras y facilitar varios pasos dentro de nuestro sistema así como evitar el periférico del teclado, ya que como hemos mencionado anteriormente este prototipo está pensado para todo público incluso aquellos que no tengan ningún conocimiento en el ramo de la computación.

El tener la información de manera clasificada minimiza los tiempos de búsqueda y por los tanto esfuerzo humano y recursos económicos, que pueden ser utilizado en otros aspectos, un ejemplo claro son los tiempos de búsqueda del historial de un usuario y el tiempo que se ahorra en encontrar estos datos se puede agilizar para realizar otras actividades.



Tabla 37. Comparación de tiempos.

	Tiempo	
	Prototipo	Sistema Actual
Registrar a un Usuario	5 minutos	20 minutos
Consulta Adeudo	5 minutos	20 minutos
Generar Reporte	2 minutos	10 minutos
Registrar Pagos	2 minutos	15 minutos
Obtener el historial Total	2 minutos	no existe
Comprobante de Asamblea	1 minuto	10 minutos
Consulta de Faltas	2 minutos	20 minutos

Como podemos apreciar en la tabla anterior, la realización de este prototipo permitirá agilizar el cobro del agua y evitar las engorrosas filas de espera, que tienen que padecer actualmente, así como obtener la información de forma casi inmediata sin preocuparse de algún intermediario que pueda alterar algún dato y asegurarnos de la integridad de los datos.

Los principales beneficiados con este proyecto son toda la comunidad en general tanto a los usuarios, como a los integrantes del comité de agua potable en donde la información cuanta con una base de datos organizada sistemáticamente y a los usuarios con la información al acceso de unos cuantos toques con el sistema touch. A continuación se muestran las pantallas del sistema final, en ellas se ven las pantallas principales y las opciones que existen tanto para navegación como para realizar las consultas mencionadas en la tabla anterior.

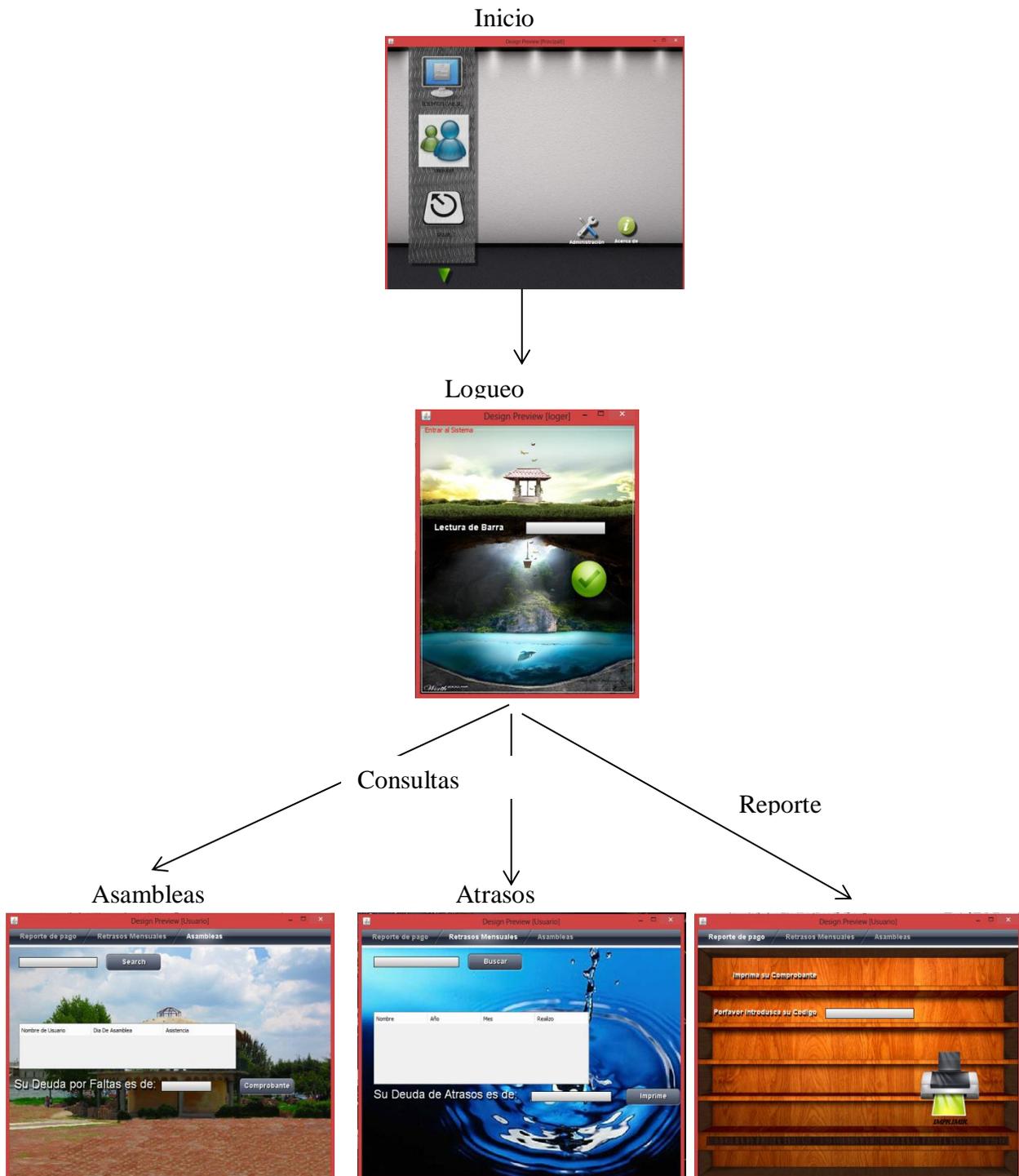


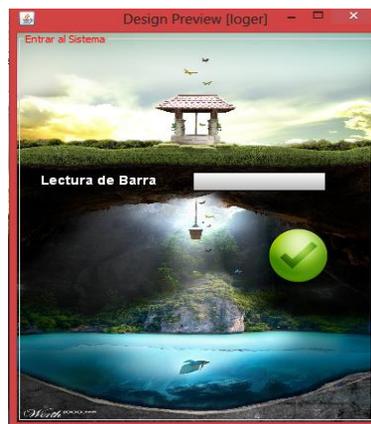
Figura 47. Resultados acceso a usuarios, asambleas, atrasos, reporte.



Inicio



Administración



ABC de Administración



ABC de Usuarios

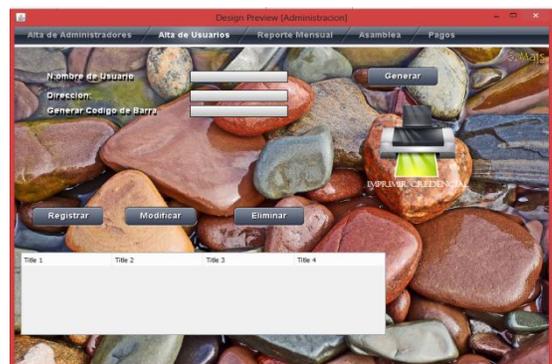


Figura 48. Resultado de acceso a administración, ABC administración y usuarios.



Administración



Reporte



Asamblea



Pago



Figura 49. Resultado realizar pago, administrar asamblea, generar reporte.

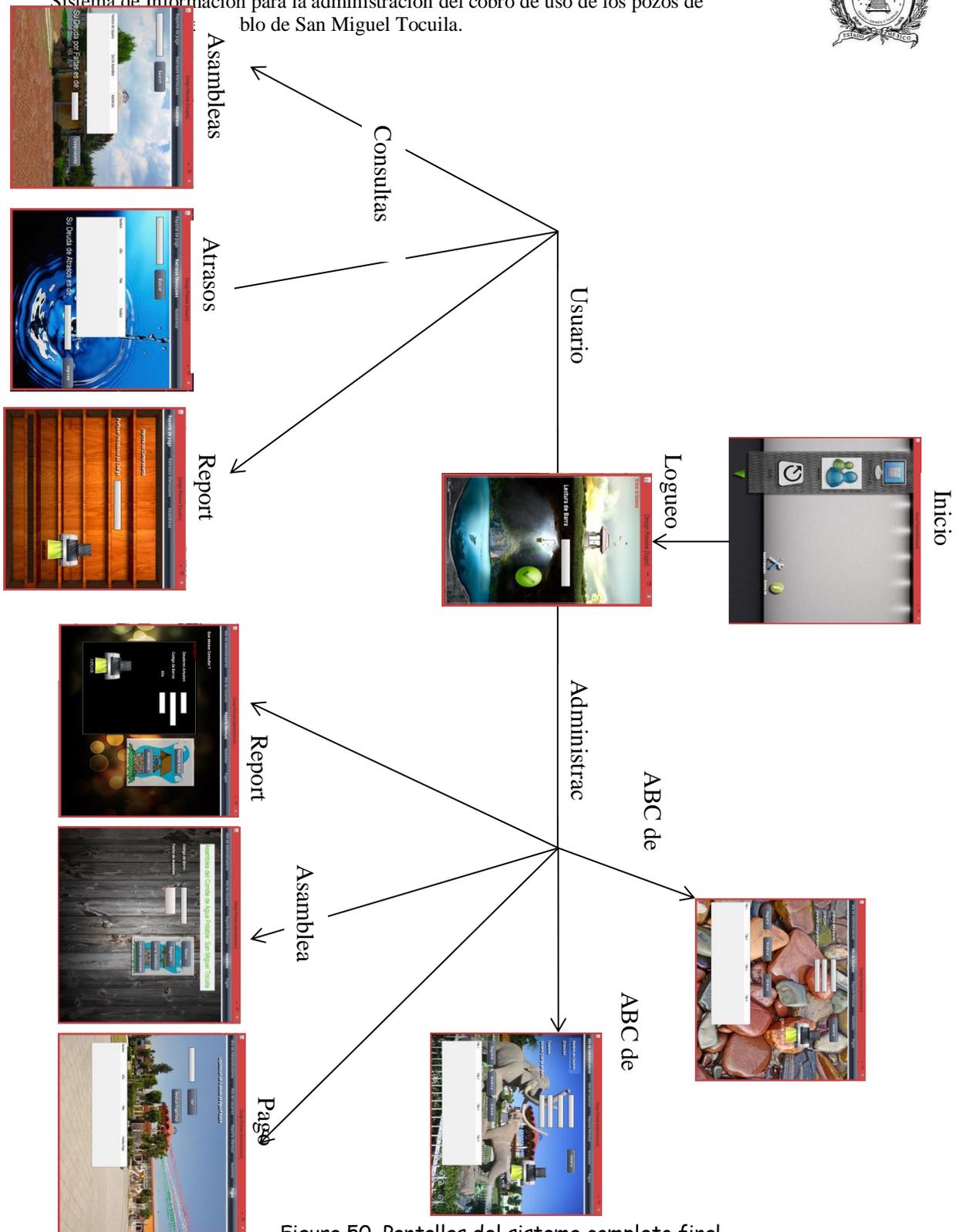


Figura 50. Pantallas del sistema completo final.



VI. Conclusiones y Recomendaciones

Realizar cualquier software o aplicación mediante una metodología de desarrollo proporciona muchas ventajas de entre las principales se podrían mencionar la organización y control tanto de recursos de tiempo, económicos y humanos, además de proporcionar una plataforma que ayuda en la gestión y documentación del todo el ciclo de desarrollo de un sistema.

La metodología utilizada en este sistema como bien sabemos es: desarrollo rápido de aplicaciones o RAD que es un proceso de desarrollo de software, desarrollado inicialmente por James Martin en 1980. El método comprende el desarrollo interactivo, la construcción de prototipos y el uso de utilidades CASE. Tradicionalmente, el desarrollo rápido de aplicaciones tiende a englobar también la usabilidad, utilidad y la rapidez de ejecución, la cual en su línea de vida desde el análisis hasta la implementación son un tanto diferente de los proyectos tradicionales, la metodología RAD concibe un desarrollo ágil ya que los clientes buscan tener sus productos funcionando en el menor tiempo posible como lo fue en este caso.

En ciertas situaciones, una solución utilizable al 80% puede producirse en el 20% de tiempo que se hubiera requerido para la solución completa. En ciertas situaciones, los requisitos de negocio de un sistema pueden satisfacerse aun cuando algunos de sus requisitos operacionales no se satisfagan. En ciertas situaciones, la aceptabilidad de un sistema puede determinarse en base a un conjunto mínimo de requisitos consensados en lugar de la totalidad de los requisitos. Con los métodos convencionales pasa un gran lapso de tiempo antes de que el cliente vea resultados. Con los métodos convencionales el desarrollo llega a tardar tanto que para cuando el sistema está listo para utilizarse los procesos del cliente han cambiado radicalmente. Con los métodos convencionales no hay nada hasta que el 100% del proceso de desarrollo se ha realizado, entonces se entrega el 100% del software.

La importancia de manejar una metodología de desarrollo y de apoyarse en ingenieros en computación en este tipo de proyectos es mayor si se desea obtener productos de calidad, que realicen sus funciones sin complicaciones y con un margen de error mínimo.

Proporcionar un sistema que disminuya los tiempos de búsqueda, significa que los interesados pueden aumentar sus tiempos para otras actividades, disminuyendo el esfuerzo mental y físico que producía buscar en hojas de cálculo o programas datos de manera secuencial, esto a su vez recorta los gastos económicos producidos por las mismas actividades de investigación y el tiempo que se llevaría para realizarlos.



Con lo anterior se puede concluir que haciendo uso de una metodología de desarrollo en los proyectos software que realiza un ingeniero en computación, se pueden lograr resultados satisfactorios que permitan a los usuarios obtener un beneficio, en este caso tener información disponible cualquier momento y lugar, liberándolo para que desempeñe otras actividades y aproveche su tiempo en distintas funciones.

Otro punto importante es que utilizar software libre abarata enormemente los costos finales ya que las licencias de los mismos no se toman o mínimamente se cotizan en los costos del mismo.

Por último, las recomendaciones para actividades o proyectos futuros que pueden integrarse a la aplicación para que este siga creciendo y solucione más problemas, y finalmente las acciones para mantener la aplicación funcionando en óptimas condiciones.

Los módulos pendientes y nuevos que se pueden agregar son los siguientes:

- ✚ Un contenedor para la recepción del pago de los usuarios, esto es que puedan consultar su deuda y en tiempo real hacer su depósito a su cuenta y que esta baya restándole a su saldo final y poder si desea obtener un comprobante de este movimientos. En vez de sacar comprobante y llevar a ser el pago al comité de agua potable. Esto es asemejándose más a un ATM. Esta característica pendiente no se pudo realizar debido a que el enfoque inicial era agilizar el cobro de agua potable, y que se necesitaba el apoyo de ingenieros en otra especialidad.
- ✚ Un módulo para que el usuario pueda decidir que pagos desea liquidación o incluso adelantar pagos de futuros meses o anualidades. Este módulo no se realizó debido a falta de tiempo.
- ✚ Llevar el control de tesorería por medio de nuestro sistema, esto es todos los ingresos y egresos que corren por parte de este comité de agua potable, para poder tener el corte de caja más rápido y sin mayores problemas, esto nos evitaría muchos problemas que existen actualmente. Estas mejoras pueden incluirse en las siguientes versiones del sistema esto es pensando a futuro.
- ✚ Migrar el sistema a un entorno de trabajo más robusto esto es a algún tipo de FrameWork, o realizar la migración a un sistema distribuido o web. Debido a que nos ofrecen más ventajas implementarlo en un sistema web, que en una aplicación de escritorio, este prototipo no fue posible realizarlo debido a la problemática de falta de recursos que se planteó desde un principio por lo cual se optó en una aplicación de escritorio local como primera versión.
- ✚ Además de estos módulos el sistema está abierto a cualquier otro que se quiera integrar para ser tomado como base del sistema de información.



Finalmente las actividades o recomendaciones para mantener el funcionamiento del sistema son las siguientes:

- ✚ Integrar un quiosco para el sistema de información y seguridad al hardware, la realización de este prototipo es básica y de bajo recursos hablando del hardware, al contar con más recursos, se podría realizar una caja que embace a todo nuestro equipo como es la impresora, la pantalla touch, el PC, y la pistola laser, contando con un lugar específico en un lugar adecuado para estas operaciones.
- ✚ Realizar constantes respaldos mensuales o quincenales de nuestra base de datos, esto siempre es necesario cuando estamos hablando de una aplicación SE, la realización de constantes copias de seguridad garantiza el poder recuperar esta información en caso de pérdida o daño al equipo.
- ✚ Mantener actualizado el software que se utiliza como lo es JDK de java, el IDE Netbeans, iReport, SampleBarCode, Edisoncorx, IDE WorkBench, al igual que las librerías como swing, iReport, conector MySql para la conexión a la base de datos.



VII Bibliografía

Kendall, Kenneth E., Kendall, Julie E. (2005). *Análisis y diseño de sistemas*. Pearson Education. 6ta Edición. México. Págs. 1-5.

Laudon, Kenneth C., Laudon, Jane P. (2004). *Sistemas de información gerencial*. Pearson Education. 8va. Edición. México. Págs. 8-31.

LAUDON, K.C., LAUDON, J. (1996). *Administración de los sistemas de información. Organización y tecnología*. Prentice-Hall. Madrid.

LAUDON, K.C., LAUDON, J. (1998). *Management information systems. New approaches to organization & technology*. Prentice-Hall. New Jersey. (5ª edición)

León, Serrano Gonzalo. (1999). *Ingeniería de sistemas de software*. ISDEFE. 1ra Edición. España.

Pressman, Robert S. (2005). *Ingeniería del software un enfoque práctico*. Mc. Graw Hill. 6ta edición. México. Págs. 48-77, 84-98, 501-634

Senn, James A. (1993). *Análisis y diseño de sistemas informáticos*. Mc. Graw Hill. 2da Edición. Pág. 21.

Sommerville, Ian. (2005). *Ingeniería del software*. Pearson Education. 7ma. Edición. México.

Whitten Jeffry, Bentley Lonnie, (2008). *Análisis de Sistemas: diseño y métodos*. Mc. Graw Hill. 7ma. Edición. México.

Quesada Laura, Vázquez Juan Carlos (1998), *Diagnostico sobre la aplicación de los sistemas de información automatizados en la administración pública*, Archivo Nacional de Costa Rica.

Rodríguez Marchan Mairu del Carmen (2005), *Elaboración de un sistema de información (base de datos) con el fin de ordenar y clasificar datos petrofísicos*, Universidad simon Bolivar, Bolivia.

Horacio Saroka Raúl (2002), *Sistema de información en la era digital*, fundación OSDE, Argentina la ley 11.723, ISBN: 987-9358-08-2

Tramulas, Jesús S. (1997). *Los sistemas de información: Una reflexión sobre información, sistema y documentación*. Revista general de información y documentación. Vol. 7, nº 1. Servicio de publicaciones Universidad Complutense. Madrid. Págs. 207-229.



Yourdon Edward. (1993). *Análisis estructurado moderno*. Prentice Hall. 1ra Edición. México. Pág. 26.

Idem: "Análisis y Diseño de Sistemas de Información". McGrawHill. México, 1992.

Yourdon, Edward: "Análisis Estructurado Moderno". Prentice Hall. México, 1993

Murdick, Robert: "Sistemas de Información Administrativa". Prentice Hall. México, 1988

Castellano Luis, "Desarrollo de sistemas de información bajo un enfoque incremental", universidad nacional experimental de la fuerza armada, Maracaibo, 2011.

Witten, C. Nevill - Manning, S. Cunningham. Building a Digital Library for Computer Science Research: Technical Issues. Proceedings of the 19th Australasian Computer Science Conference. 1996

J.M. Martínez, J.R.Hilera, J. Martínez, J.A. Gutiérrez. Orientación a Objetos en la Documentación Hipermedia. Universidad de Alcalá de Henares, Departamento de Ciencias de la Computación.

McLEOD, R. (2000). Sistemas de información gerencial. Séptima Edición. Pearson Educación. México

O'BRIEN, J.A. (1993). Management information systems: a managerial end user perspective. Irwin. Boston.

O'BRIEN, J.A. (2000). Sistemas de información gerencial. Irwin McGraw-Hill. Bogotá.

T. Reenskaug, "Thing-Model-View-Editor an Example from a planning system", http://folk.uio.no/trygver/2007/MVC_Originals.pdf, May 1979, acceso diciembre 2013.

T. Reenskaug, "Models-Views-Controllers", <http://heim.ifi.uio.no/~trygver/1979/mvc-2/1979-12-MVC.pdf>, December 1979, acceso diciembre 2013

R. Elmasri y S. Navathe. Fundamentos de los Sistemas de Bases de Datos (3ª edición). Addison Wesley, 2002.

A. Silberschatz, H. F. Korth y S. Sudarshan. Fundamentos de Bases de Datos (4ª edición). McGraw Hill, 2002



VIII. Anexo

Relación: Administrador
 Se relaciona con: dirección
 Llave primaria: barraadmin
 Llave foránea: direccion_idDireccion

Campo	Tipo	Cotejamiento	Atributos	Nulo
Barraadmin	bigint		UNSIGNED	no
nombre	varchar(45)	utf-8 generalCI		no
apellidoP	varchar(45)			no
apellidoM	varchar(45)			no
telefono	int(15)		unsigned	no
direccion_idDireccion	int(15)		unsigned	no

Relación: dueno
 Se relaciona con: dirección
 Llave primaria: barradueno
 Llave foránea: direccion_idDireccion

Campo	Tipo	Cotejamiento	Atributos	Nulo
barradueno	bigint		unsigned	no
nombred	varchar(45)	utf-8 generalCI		no
apellidoP	varchar(45)	utf-8 generalCI		no
apellidoM	varchar(45)	utf-8 generalCI		no
direccion_idDireccion	int(15)		unsigned	no

Relación: caja
 Se relaciona con: Administrador
 Llave primaria: idcaja
 Llave foránea: administrador_barraadmin



Campo	Tipo	Cotejamiento	Atributos	Nulo
idcaja	int(15)		unsigned	no
saldo	int(15)		unsigned	no
retiro	int(15)		unsigned	no
administrador_barradmin	bigint		unsigned	no

Relación: dirección
 Se relaciona con: Administrador, dueño
 Llave primaria: idDireccion
 Llave foránea: no

Campo	Tipo	Cotejamiento	Atributos	Nulo
idDireccion	int(15)		unsigned	no
calle	varchar(45)	utf-8 generalCI		no
numero	int(15)		unsigned	no
barrio	varchar(45)	utf-8 generalCI		no
señas_Particulares	varchar(45)	utf-8 generalCI		no

Relación: asamblea
 Se relaciona con: dueño, tarifa
 Llave primaria: idmultas
 Llave foránea: dueño_barradueno, Tarifa_idTarifa

Campo	Tipo	Cotejamiento	Atributos	Nulo
idmultas	int(15)		unsigned	no
dia_de_asamblea	Date			no
asistencia	varchar(45)	utf-8 generalCI		no
dueño_barradueno	bigint		unsigned	no
Tarifa_idTarifa	int(15)		unsigned	no

Relación: pago_mensual
 Se relaciona con: dueño, tarifa
 Llave primaria: id_pago
 Llave foránea: dueño_barradueno, Tarifa_idTarifa



Campo	Tipo	Cotejamiento	Atributos	Nulo
id_pago	int(15)		unsigned	no
año	year			no
mes	varchar(45)	utf-8 generalCI		no
realizo_pago	varchar(2)	utf-8 generalCI		no
dueño_barradueno	bigint		unsigned	no
Tarifa_idTarifa	int(15)		unsigned	no

Relación: tarifa

Se relaciona con: asamblea, pago_mensual

Llave primaria: idtarifa

Llave foránea: no

Campo	Tipo	Cotejamiento	Atributos	Nulo
idTarifa	int(15)		unsigned	no
cantidad	int(15)		unsigned	no