

INFORME FINAL DE ESTADÍA

ÁREA DE TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIÓN

PROGRAMA ACADÉMICO DE TECNOLOGÍAS DE LA INFORMACIÓN

APLICACIÓN WEB PARA CONTROL DE DOCUMENTOS

PARA LA EMPRESA



PRESENTA
ALDO OMAR GUAJARDO CHÁVEZ

PARA OBTENER EL TÍTULO DE
INGENIERO EN TECNOLOGÍAS DE LA INFORMACIÓN

ASESOR EMPRESARIAL: MARCO ANTONIO TREJO GÓMEZ
ASESOR ACADÉMICO: JORGE IVÁN RAMÍREZ PÉREZ

GENERACIÓN: 2015 - 2017
ACÁMBARO, GUANAJUATO. ABRIL 2017

RESUMEN

El presente documento detalla la información generada durante el desarrollo de la aplicación web del proyecto de estadía.

El desarrollo de la aplicación web, es una parte fundamental ya que permitirá a la Dirección del Registro Público de la Propiedad y el Comercio gestionar sus procedimientos en documentos. El resultado esperado con el desarrollo de la presente aplicación web se centra en controlar los documentos, dando un manejo de estos, permitiendo al usuario realizando diferentes acciones, como lo son dar de alta, dar de baja o actualizar la información referente a los documento, facilitando al usuario un manejo de esta información.

Esta aplicación web será de gran ayuda para hacer más eficiente y rápida el control de documentos de la dependencia, ya que fue creado para que el usuario pueda gestionar los documentos con los que cuenta la Dirección del Registro Público de la Propiedad y el Comercio.

AGRADECIMIENTOS

Mis agradecimientos son para aquellas personas que lograron la posibilidad de concluir este proyecto a la Dirección del Registro Público de la Propiedad y el Comercio, por haberme dado la oportunidad de desarrollar mi proyecto de estadía, por su apoyo y dedicación, por mostrarse siempre atentos y amables a lo largo de mi estancia en la dependencia, la cual me proporcionó experiencia y me permitió reafirmar los conocimientos adquiridos durante mi formación de igual manera a mi asesor empresarial Lic. Marco Antonio Trejo Gómez que en todo momento contribuyo para la realización de la aplicación web.

A mi asesor académico, el profesor Jorge Iván Ramírez Pérez por haberme señalado mis errores, así como guiarme para mejorarlos, por todo su tiempo, paciencia y apoyo brindado para poder realizar la aplicación web, por aclarar las dudas que en un momento se presentaron, pero sobre todo por ser un gran profesor.

A todos los que facilitaron e hicieron posible esta etapa tan importante en mi vida “Muchas Gracias”.

ÍNDICE

INTRODUCCIÓN

1. ANTECEDENTES.....	1
1.1. FILOSOFÍA EMPRESARIAL	1
1.2. MISIÓN.....	1
1.3. VISIÓN	2
1.4. VALORES	2
1.5. POLÍTICAS.....	2
1.5.1. Políticas de Calidad.....	2
1.5.2. Política de Seguridad de la Información.....	3
1.6. ORGANIGRAMA	4
1.7. HISTORIA	4
1.7.1. Antecedentes Nacionales.....	4
1.7.2. Antecedentes del Sistema Registral de México	5
1.7.3. Antecedentes Registro Público de la Propiedad y el Comercio del Estado de Querétaro.	6
2. PROBLEMÁTICA.....	7
3. JUSTIFICACIÓN.....	8
4. OBJETIVOS.....	9
4.1. OBJETIVO GENERAL.....	9
4.2. OBJETIVOS METODOLÓGICOS	9
4.3. ALCANCE	10
5. MARCO TEÓRICO	11
5.1. CICLOS DE VIDA.....	11
5.1.1. Modelo de ciclo de vida en Cascada.....	12
5.1.2. Modelo de ciclo de vida en V	13
5.1.3. Modelo de ciclo de vida en Espiral	14

5.1.4. Ciclo de vida en Iterativo	16
5.2. ESTRUCTURA DE DESCOMPOSICIÓN DEL TRABAJO	17
5.3. DEFINIR DIAGRAMA DE GANTT	18
5.3.1. Herramienta para generar diagrama de Gantt.	20
5.4. HERRAMIENTA PARA RECOLECTAR DATOS	23
5.4.1. Entrevista	23
5.4.2. Encuesta	24
5.4.3. Observación	25
5.5. ESPECIFICACIONES DE REQUERIMIENTOS	25
5.5.1. Requerimientos funcionales	26
5.5.2. Requerimientos no funcionales	27
5.6. HERRAMIENTAS DE MODELADO UML	28
5.6.1. Qué es UML	28
5.6.2. Qué diagramas lo componen	28
5.6.3. Herramientas para generar diagramas de UML	35
5.7. HERRAMIENTAS PARA LE DISEÑO DE BASE DE DATOS	36
5.7.1. Modelo Relacional.....	36
5.7.2. Software para diseñar el modelo relacional	37
5.8. HERRAMIENTA DE MAQUETADO	39
5.8.1. Diseño de bocetos.....	39
5.8.2. Frameworks de maquetado.....	42
5.9. ENTORNOS DE PROGRAMACIÓN	45
5.9.1. NetBeans	45
5.9.2. IntelliJ IDEA.....	46
5.9.3. Eclipse.....	47
5.9.4. Sublime Text	48
5.10. LENGUAJES DE PROGRAMACIÓN	49
5.10.1. JavaScript	49
5.10.2. PHP	51

5.10.3. HTML5	52
5.11. PRUEBAS	54
5.11.1. Pruebas unitarias	54
5.11.2. Pruebas de integración	54
6. METODOLOGÍA.....	55
6.1. PLANEACIÓN DE LAS ACTIVIDADES PARA LA CONSTRUCCIÓN DE APLICACIÓN.....	55
6.1.1. Ciclo de vida que se utilizó para el desarrollarlo	55
6.1.2. Herramientas que se utilizó para realizar el diagrama de Gantt....	56
6.2. RECOPIALAR EL MATERIAL NECESARIO	57
6.2.1. Herramientas seleccionadas para la recolectar material.....	57
6.2.2. Normas requeridas para especificar requerimientos	57
6.3. ANALIZAR INFORMACIÓN RECOPIILADA PARA DISEÑAR LOS DIAGRAMAS DE UML	58
6.4. DISEÑAR LOS DIAGRAMAS DE LA BASE DE DATOS.....	59
6.4.1. Herramienta seleccionada para el Diagrama Relacional.....	59
6.5. DISEÑAR LOS DIAGRAMAS DE INTERACCIÓN DE LA ETAPA DE DISEÑO.....	59
6.6. DISEÑAR LOS BOCETOS DEL SITIO WEB	59
6.7. CODIFICAR EL SITIO WEB.....	59
6.8. PROBAR EL SITIO WEB	60
7. RESULTADOS	61
7.1. ACTIVIDADES PARA LA CONSTRUCCIÓN DE LA APLICACIÓN	61
7.1.1. Ciclo de vida iterativo	61
7.1.2. Diagrama de Gantt.....	62
7.2. MATERIAL RECOPIILADO Y REQUERIMIENTOS	64
7.2.1. Entrevista	64
7.2.2. Especificación de Requerimientos de Software	61
7.3. ANÁLISIS DE LA INFORMACIÓN	62

7.3.1.	Casos de uso y secuencia	62
7.3.2.	Diagramas de clases.....	93
7.4.	BASE DE DATOS	93
7.4.1.	Diagrama Entidad Relación.....	93
7.4.2.	Diagrama Relacional.....	94
7.5.	MAPA DE NAVEGACIÓN	95
7.6.	BOCETOS.....	96
7.6.1.	Login	96
7.6.2.	Usuarios	96
7.6.3.	Documentos	99
7.7.	CODIFICACIÓN	99
7.7.1.	Base de datos	99
7.7.2.	Clases	111
7.8.	IMPLEMENTACIÓN	114
7.9.	VERIFICACIÓN.....	125
	CONCLUSIONES	127
	REFERENCIAS	128
	GLOSARIO	131
	ANEXOS	

INTRODUCCIÓN

Una aplicación web es un conjunto organizado y coherente de páginas web con una conexión a una base de datos, en el cual interactúen entre sí, realizando diferentes acciones. Su temática puede ser muy variada. Por medio de una aplicación web se puede ofrecer todo tipo de acciones para diferentes giros, informar, publicitar o vender contenidos, productos, administrar todo tipo de información.

Una aplicación web de calidad permitirá a la empresa o institución agilizar sus procesos, demostrando por encima de todo, el conocimiento y la aceptación de las nuevas tecnologías. Por eso, el no contar con una aplicación web es una gran desventaja competitiva con respecto a otras empresas.

En este documento se presenta el análisis que se llevó a cabo para el desarrollo de una aplicación web, además se muestra un resumen de las herramientas que se utilizaron para el desarrollo de dicha aplicación.

La aplicación web fue desarrollada de acuerdo a las necesidades de la dependencia, de manera que fuera de fácil de uso para el usuario, para esto se utilizaron diversas herramientas como fueron software de maquetados, editores de imágenes, un gestor de base de datos, una aplicación de edición y creación de contenidos web.

1. ANTECEDENTES

Dessler (1979) menciona que: “Los antecedentes reflejan los avances y el Estado actual del conocimiento en un área determinada y sirven de modelo o ejemplo para futuras investigaciones”.

1.1. FILOSOFÍA EMPRESARIAL

Según Dessler (1979) expresa que la filosofía empresarial:

Define los valores de la institución tanto de la perspectiva interna como de su relación con el entorno, es decir que la empresa debe ser congruente con lo que hace y con lo que ofrece a sus clientes investigar lo que ya está hecho.

El Registro Público de la Propiedad es una Institución Gubernamental que depende de la Secretaría de Gobierno en los Estados Mexicanos. El Registro tiene por objeto la publicidad oficial de la situación jurídica de los bienes inmuebles, siendo el objeto aquella cuestión sobre la cual va a versar el procedimiento, o lo que es igual, el acto jurídico que ha adquirido forma notarial y sin la cual no sería posible concebir su existencia para efectos registrales.

1.2. MISIÓN

Somos un organismo gubernamental encargado de dar certeza legal y seguridad jurídica a personas físicas y morales por medio de la inscripción y certificación de los actos jurídicos en materia inmobiliaria y mercantil así como la disponibilidad de acceso a los documentos registrados e información registral con que cuenta el Registro Público de la Propiedad y del Comercio.

1.3. VISIÓN

Aspiramos a ser un Registro Público que se distinga por su confiabilidad, modernidad y seguridad, así como por su personal altamente calificado, todo ello con el afán de contribuir a proporcionar la mayor certeza jurídica en los actos que se registren.

1.4. VALORES

- Eficiencia
- Honestidad
- Oportunidad
- Productividad
- Atención de calidad

1.5. POLÍTICAS

1.5.1. Políticas de Calidad

Brindar en la inscripción, publicidad y certificación de los actos y hechos jurídicos, un servicio honesto, oportuno y veraz que satisfaga las necesidades y expectativas de nuestros usuarios, bajo un sistema de calidad y mejora continua.



Imagen 1. ISO 9001: 2008.

1.5.2. Política de Seguridad de la Información

Mantener la confidencialidad, integridad y disponibilidad de la información necesaria para brindar los servicios a nuestros usuarios, controlando su acceso y uso únicamente para los propósitos autorizados dentro del marco legal y normativo aplicable.



IS 546793
ISO/IEC 27001:2005

Imagen 2. ISO/IEC 27001: 2005.

1.6. ORGANIGRAMA

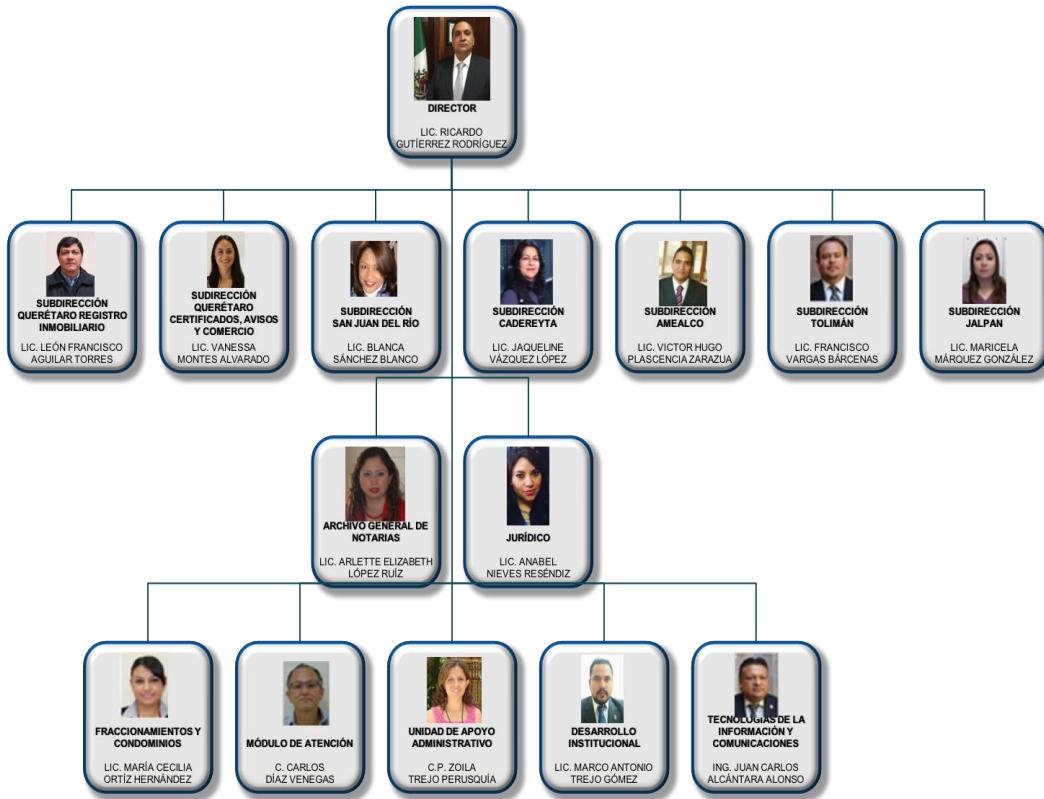


Figura 1. Organigrama.

1.7. HISTORIA

1.7.1. Antecedentes Nacionales

El régimen de la tenencia de la tierra en el derecho azteca tuvo un carácter público.

- La tarea primordial del tlacuilo era dejar constancia de todo tipo de acontecimiento por medio de signos ideográficos y pinturas.
- Durante la etapa de aculturación y colonización el oficio de escribano era específico de los españoles peninsulares.
- La función del escribano fedatario era asentar por escrito la fundación de las ciudades, la creación de instituciones, los asuntos tratados en los cabildos.

- El registro público de la propiedad surge como una finalidad administrativa., surge con la clandestinidad de las cargas y de los gravámenes de los bienes,
- El registro surgió como un medio de seguridad en el tráfico jurídico y esa finalidad la logra mediante la atribución de efectos a los asientos registrales relativos a la constitución, transmisión, modificación y extinción de los derechos reales sobre inmuebles.
- Su nombre surge de las inscripciones en los libros del registro y de los documentos relacionados con las inscripciones que estén archivados, poniendo a su disposición los libros con los asientos respectivos.

1.7.2. Antecedentes del Sistema Registral de México

- El régimen de la tenencia de la tierra en el derecho azteca tuvo un carácter público.
- La tarea primordial del tlacuilo era dejar constancia de todo tipo de acontecimiento por medio de signos ideográficos y pinturas.
- Durante la etapa de aculturación y colonización el oficio de escribano era específico de los españoles peninsulares.
- La función del escribano fedatario era asentar por escrito la fundación de las ciudades, la creación de instituciones, los asuntos tratados en los cabildos.
- El registro público de la propiedad surge como una finalidad administrativa, surge con la clandestinidad de las cargas y de los gravámenes de los bienes,
- El registro surgió como un medio de seguridad en el tráfico jurídico y esa finalidad la logra mediante la atribución de efectos a los asientos registrales relativos a la constitución, transmisión, modificación y extinción de los derechos reales sobre inmuebles.

- Su nombre surge de las inscripciones en los libros del registro y de los documentos relacionados con las inscripciones que estén archivados, poniendo a su disposición los libros con los asientos respectivos.

1.7.3. Antecedentes Registro Público de la Propiedad y el Comercio del Estado de Querétaro.

- En Querétaro (1897) el Gobernador Francisco G. De Cosio, se decreta la Ley Reglamentaria del Registro Público de la Propiedad.
- 1976 “Ley Reglamentaria de Registro de Registro Público” decretada por Antonio Calzada Gobernador.
- En 2003 se instituye el Programa de Desarrollo Institucional teniendo como objetivo principal objetivo el establecimiento de un programa de Calidad y mejora continua.
- 2007 Certificación de calidad ISO 9001-2000 (ISO 9001:2000 COPANT / ISO 9001-2000 NMX-CC-9001-IMNC-2000).
- 2010 Certificación de calidad ISO 9001-2008 (ISO 9001:2008 COPANT / ISO 9001-2000 NMX-CC-9001-IMNC-2008).
- Certificación de Seguridad de la Información bajo la norma ISO 27001:2005 otorgado por BSI.

2. PROBLEMÁTICA

Según (Arias, 1999) “La problemática consiste en describir de manera amplia la situación del objeto del estudio, ubicándola en un contexto que permita su origen, relaciones e incógnitas por descubrir”.

El Registro Público de la Propiedad y del Comercio del estado de Querétaro, hoy en día cuenta con software de gestión de documentos. En el contiene todos sus procesos documentados. En el cual gestiona sus documentos, permitiendo insertar, actualizar y modificar estos documentos.

La problemática se presenta en que el software no se muestra los documentos de una manera visible y accesible para su descarga del usuario. De igual manera no presenta de forma intuitiva al usuario como agregar, borrar y actualizar los documentos almacenados. Ya que el software no ofrece esto, no permite agilizar su utilización y dejar que su administración deje de ser complicada.

Es importante que se mantengan innovadas y/o actualizadas las herramientas para reducir los tiempos de su administración y de tal manera poder agregar y/o modificar los procesos documentados ofrecidos de parte de la dependencia, y es necesario que se continúe una gestión de sus documentos.

Puesto que en la dependencia gubernamental del registro público de la propiedad y del comercio, es de suma importancia el poder tener un control detallado de sus procesos, que les permiten agilizar sus labores como servidores públicos.

3. JUSTIFICACIÓN

La aplicación web fue pensada en la comodidad de gestionar los procesos documentados, y así permitirles una herramienta que cubra sus necesidades. Esta aplicación web ayudará a visualizar los documentos de manera que su descarga y consulta sea accesible.

La importancia de desarrollar una aplicación web le permitirá a el Registro Público de la Propiedad y del Comercio del estado de Querétaro, es administrar sus procesos capturados en documentos, de no hacerlo, el control de sus de sus documentos propiciara que se sigan demorando sus consultas de procesos.

La empresa ha enfocado sus acciones en aspectos de infraestructura, tal como seguir creciendo y al igual que le obliga a la institución brindar un mejor servicio.

Por todas las cuestiones mencionadas dentro de la misma problemática el departamento ha optado por incorporar una aplicación web de control de documentos, considerando una mejor productividad en sus consultas de documentos. Ya que dicha aplicación web permitara poder gestionar documentos a el Registro Público de la Propiedad y del Comercio. De esta manera el sistema le permitirá administrar de manera rápida sus procesos documentados.

4. OBJETIVOS

Para Arias (2012), un objetivo es el fin último al que se dirige una acción u operación. Es el resultado o sumatoria de una serie de metas y procesos.

4.1. OBJETIVO GENERAL

Un objetivo general expresa "el fin concreto de la investigación en correspondencia directa con la formulación del problema" (Arias, 2012).

Desarrollar una la aplicación web de control de documentos que gestione de manera eficiente todos sus procesos almacenados, que ejecutan en todo el Registro Público de la Propiedad y del Comercio del Estado de Querétaro, de tal manera que les permita agilizar sus actividades que conlleva la utilización de estos documentos.

4.2. OBJETIVOS METODOLÓGICOS

"Los objetivos metodológicos indican con precisión los conceptos, variables o dimensiones que serán objeto de estudio. Se derivan del objetivo general y contribuyen al logro de éste" (Chiavenato, 1992).

4.2.1. Planear las actividades, empleando en un ciclo de vida para la construcción de la aplicación a través un diagrama de Gantt.

4.2.2. Recolectar el material necesario, empleando técnicas de recopilación de datos y especificación de requerimientos.

4.2.3. Analizar información recopilada para diseñar los diagramas con herramientas Unified Modeling Language (UML).

4.2.4. Definir los requerimientos con base a la información recopilada.

4.2.5. Diseñar los diagramas de interacción de la etapa de diseño con herramientas UML.

4.2.6. Generar mapa de navegación con un programa de maquetado.

4.2.7. Realizar la programación utilizando un entorno de programación.

4.3. ALCANCE

Para esta definición, varios autores nos dicen que cuando se habla sobre el alcance de una investigación “no se debe pensar en una tipología, ya que más que una clasificación, lo único que indica dicho alcance es el resultado que se espera obtener del estudio” (Hernández, Fernández, & Baptista, 2010).

Desarrollar una aplicación web en la cual se lleve el control de los documentos con los que cuenta la Dirección de Registro Público de la Propiedad:

- La aplicación web cuenta con una conexión con una base de datos.
- Cuenta con el modulo administrador, en el cual permite accesos por medio de diferentes usuarios.
- Cuenta con el modulo documentos, en el cual permite agregar, borrar y actualizar documentos.
- Cuenta con formulario para agregar y editar documentos.
- Los documentos se muestran de manera visible y accesible para su descarga.
- La aplicación web muestra un apartado de cuantos documentos están almacenados.
- La aplicación web se implementó para al desarrollo institucional, para la administración de documentos.
- La aplicación web se implementara en un futuro en un servidor local.
- La aplicación web se entregara de manera digital.

5. MARCO TEÓRICO

Una aportación de Werther, Davis y Guzmán (2014) menciona que el marco teórico consiste en desarrollar la teoría que va a fundamentar el proyecto con base al planteamiento del problema que se ha realizado.

5.1. CICLOS DE VIDA

El propósito de los ciclos de vida es definir las distintas fases intermedias que se requieren para validar el desarrollo de la aplicación, es decir, para garantizar que el software cumpla los requisitos para la aplicación y verificación de los procedimientos de desarrollo.

Estos programas se originan en el hecho de que es muy costoso rectificar los errores que se detectan tarde dentro de la fase de implementación. El ciclo de vida permite que los errores se detecten lo antes posible y por lo tanto, permite a los desarrolladores concentrarse en la calidad del software, en los plazos de implementación y en los costos asociados.

Un modelo de ciclo de vida del software es una vista que ocurre durante el desarrollo del software, determina el orden de las etapas.

Tipos de vida de software:

- Ciclo de vida en Cascada.
- Ciclo de vida en V.
- Ciclo de vida en Espiral.
- Ciclo de vida Iterativo.

5.1.1. Modelo de ciclo de vida en Cascada

El ciclo de vida en cascada es un enfoque metodológico que ordena rigurosamente las etapas del ciclo de vida del software, de tal forma que el inicio de cada etapa debe esperar a la finalización de la inmediatamente anterior etapa.

A continuación se listan las etapas de este ciclo:

- Análisis de requisitos.
- Diseño del Sistema.
- Diseño del Programa.
- Codificación.
- Pruebas.
- Implementación.
- Mantenimiento.

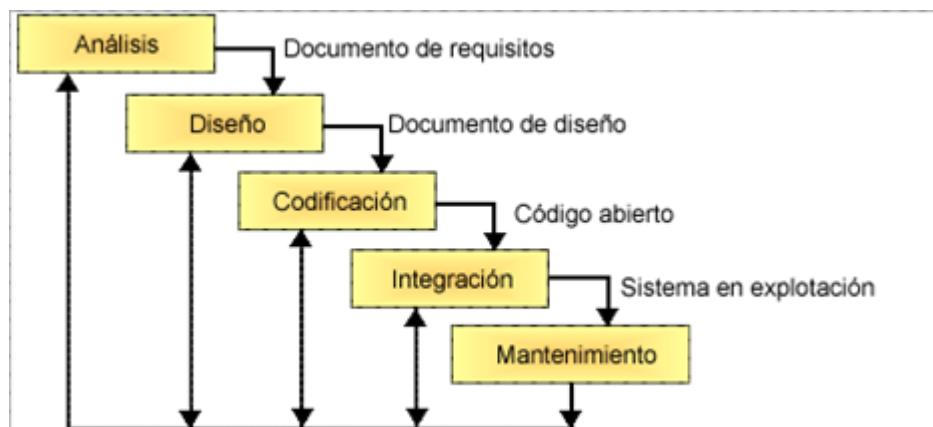


Imagen 3. Ciclo de vida cascada.

Ventajas:

- La planificación es sencilla.
- La calidad del producto resultante es alta.
- Permite trabajar con personal poco calificado.

Desventajas:

- Necesidad de tener todos los requisitos al principio.
- No se tienen indicadores fiables del progreso del trabajo.
- Es más lento que los demás y el costo es mayor también.

5.1.2. Modelo de ciclo de vida en V

El modelo de ciclo de vida V establece que los procedimientos utilizados para probar si la aplicación cumple las especificaciones ya deben haberse creado en la fase de diseño.

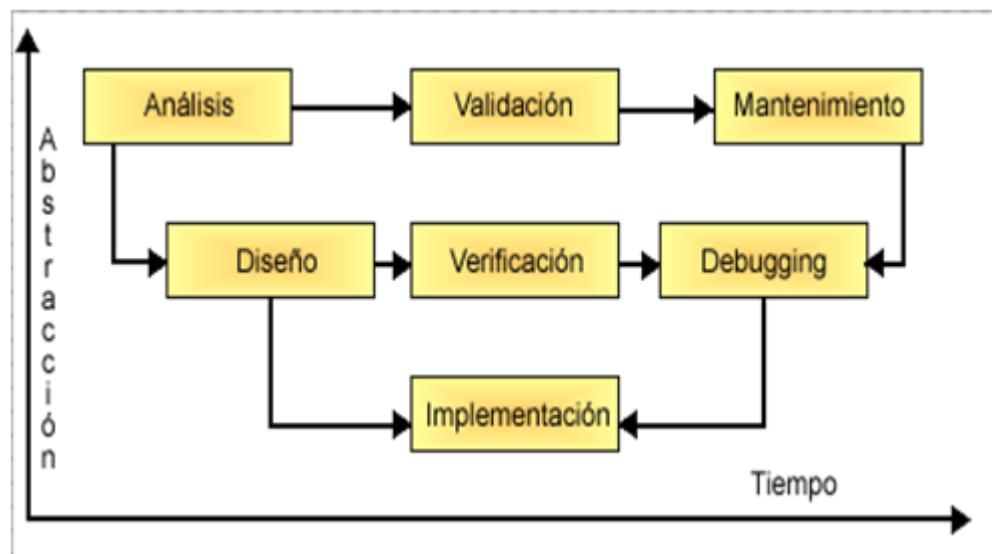


Imagen 4. Ciclo de vida en v.

Ventajas:

- Se utiliza correctamente para ciclos en los que se tiene una decisión estable del producto.
- Puede constituir una elección correcta para el desarrollo rápido.
- Ayuda a minimizar los gastos de la planificación porque permite realizarla sin problemas.

- Evita una fuente común de errores importantes.
- Presenta el proyecto con una estructura que ayuda a minimizar el esfuerzo inútil.

Desventajas:

- Dificultad para especificar claramente los requerimientos al comienzo del proyecto esto no permite flexibilidad los cambios.
- Para un proyecto de desarrollo rápido, el modelo de cascada puede suponer una cantidad excesiva de documentación.
- Si se intenta mantener la flexibilidad, la actualización de la especificación se puede convertir en un trabajo a tiempo completo.
- No es imposible volver atrás utilizando el modelo de cascada pura, pero si difícil.
- Genera pocos signos visibles de progreso hasta el final.

5.1.3. Modelo de ciclo de vida en Espiral

El ciclo consiste en una serie de ciclos que se repiten. Cada uno tiene las mismas fases y cuando termina da un producto ampliado con respecto al ciclo anterior. En este sentido es parecido al modelo incremental, la diferencia importante es que tiene en cuenta el concepto de riesgo.

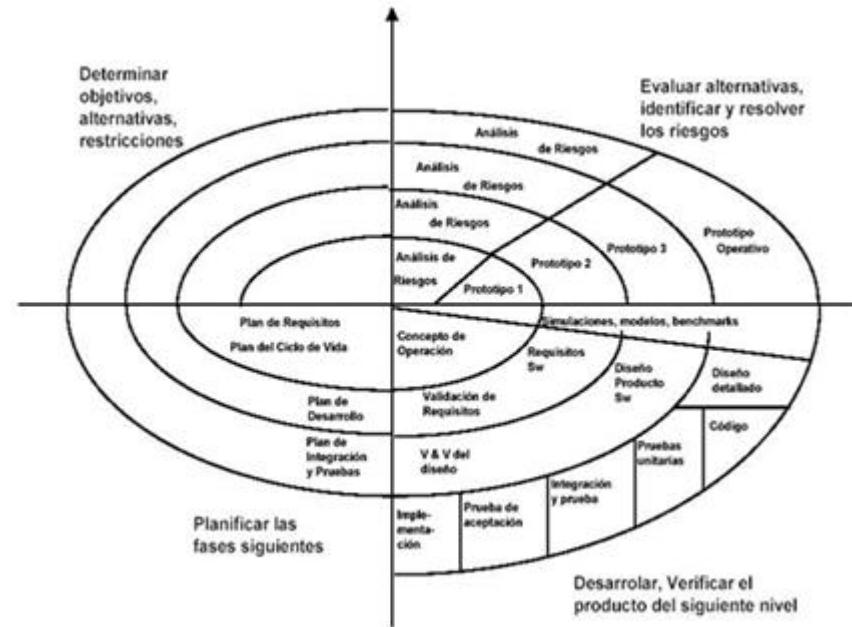


Imagen 5. Ciclo de vida en espiral.

Ventajas:

- No necesita una definición completa de los requisitos para empezar a funcionar.
- Al entregar productos desde el final de la primera iteración es más fácil validar los requisitos.
- El riesgo en general es menor, porque si todo se hace mal, solo se ha perdido el tiempo y recursos invertidos en una iteración.

Desventajas:

- Es difícil evaluar los riesgos.
- Necesita de la participación continua por parte del cliente.
- Cuando se subcontrata hay que producir previamente una especificación completa de lo que se necesita, y esto lleva tiempo.

5.1.4. Ciclo de vida en Iterativo

Este ciclo también se deriva del ciclo en cascada, este modelo busca reducir el riesgo que surge entre las necesidades del usuario y del producto final por malos entendidos durante la etapa de solicitud de requerimientos.

Al final de cada interacción se le entrega al cliente una versión mejorada o con mayores funcionalidades del producto. El cliente es el que después de cada interacción, evalúa y corrige o propone las mejoras para el producto. Esta interacción se repartirá hasta obtener un producto que satisfaga al cliente.

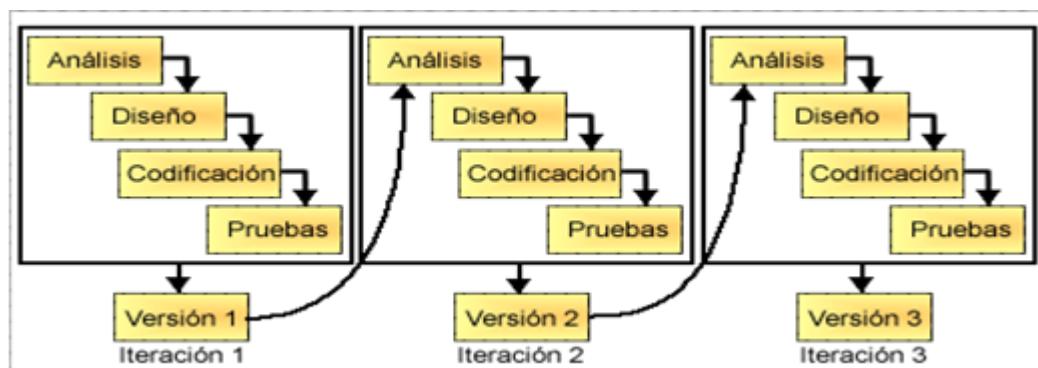


Imagen 6. Ciclo de vida en iterativo.

Ventajas:

- Adaptación del modelo a aplicaciones grandes, medianas.
- Creación de prototipos para presentarlos y conseguir la conformidad.

Desventajas:

- La mala planeación del proyecto en general.
- No tener comunicación.

5.2. ESTRUCTURA DE DESCOMPOSICIÓN DEL TRABAJO

Según La Guía del PMBOK®, “La Estructura de Desglose del Trabajo (EDT) es una descomposición jerárquica, orientada al producto entregable del trabajo que será ejecutado por el equipo del proyecto, para lograr los objetivos del proyecto y crear los productos entregables requeridos”. (Institute, 2013).

El logro de los objetivos del proyecto requiere de una EDT que defina todos los esfuerzos requeridos o necesarios, para la asignación de las responsabilidades al elemento definido de la organización y que a partir de la EDT se establezca un cronograma y presupuesto adecuado para la realización de los trabajos.

La EDT organiza y define el alcance total del proyecto y representa el trabajo especificado en la declaración del alcance del proyecto aprobada y vigente. El trabajo planificado está contenido en el nivel más bajo de los componentes de la EDT, denominados paquetes de trabajo.

Ventajas:

- La EDT define y organiza la estructura de trabajo total del proyecto.
- Cada actividad de la EDT tiene un entregable tangible.
- La EDT subdivide el trabajo del proyecto en porciones más pequeñas y fáciles de manejar.
- Cada nivel descendente representa una definición cada vez más detallada del trabajo del proyecto, tales componentes de más bajo nivel se denominan paquetes de trabajo.
- El trabajo planificado comprendido en los paquetes de trabajo puede ser programado, supervisado, controlado y sus costos estimados.

- La EDT es la representación de una estructura jerárquica.

Ejemplo:

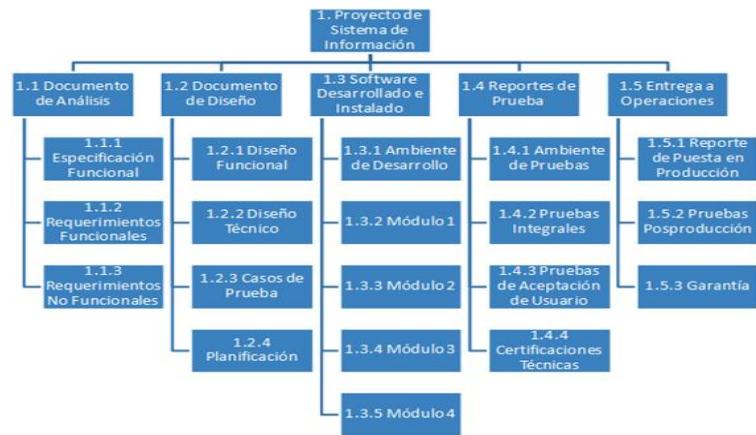


Figura 2. EDT.

5.3. DEFINIR DIAGRAMA DE GANTT

Un diagrama de Gantt es un tipo de gráfico de barras que ilustra un calendario del proyecto. Algunos gráficos de Gantt también muestran la dependencia, es decir una red que muestra las relaciones entre las actividades. Los diagramas de Gantt se pueden utilizar para mostrar en el calendario el estado de las actividades de cada trabajador.

Ventajas:

- Los diagramas de Gantt se han convertido en una técnica común para la representación de las fases y actividades de un proyecto como la estructura de desglose de trabajo (WBS), ya que estas pueden ser comprendidas por un amplio público.
- Un error común de los administradores de proyectos es tratar de definir la estructura de desglose de trabajo del proyecto, al mismo tiempo que se define el calendario de actividades. Esta práctica hace que sea muy difícil de seguir la regla al 100%. La WBS debe

ser plenamente definida, posteriormente, el calendario del proyecto puede ser diseñado.

- Aunque un diagrama de Gantt es fácil de entender para los pequeños proyectos que caben en una sola hoja, puede convertirse en algo bastante difícil de manejar para los proyectos con más de 30 actividades. Un diagrama de Gantt demasiado largo puede no ser adecuado para entender la secuencia de las actividades. Otra crítica es que estos diagramas comunican relativamente poca información; es decir, los proyectos son mucho más complejos de lo que puede comunicarse de manera efectiva con un diagrama de Gantt.
- Los diagramas de Gantt sólo representan una parte de la triple limitación de los proyectos, ya que se centran principalmente en el plan del programa. Además, los diagramas de Gantt no representan el tamaño de un proyecto o el tamaño relativo de los elementos de trabajo, por lo tanto, la magnitud de un programa es fácilmente mal entendida. Si dos proyectos tienen el mismo número de días de retraso, el proyecto más amplio tiene un mayor impacto en la utilización de los recursos, sin embargo, el diagrama de Gantt no representa a esta diferencia.
- Aunque la gestión de proyectos con el uso de software puede mostrar el cuadro de dependencias con líneas entre las actividades, mostrando un gran número de dependencias puede dar lugar a un cuadro saturado que no se puede leer.
- Debido a que las barras de un diagrama de Gantt tienen una altura fija, pueden tergiversar el tiempo de trabajo escalonado de un proyecto.

Ejemplo:

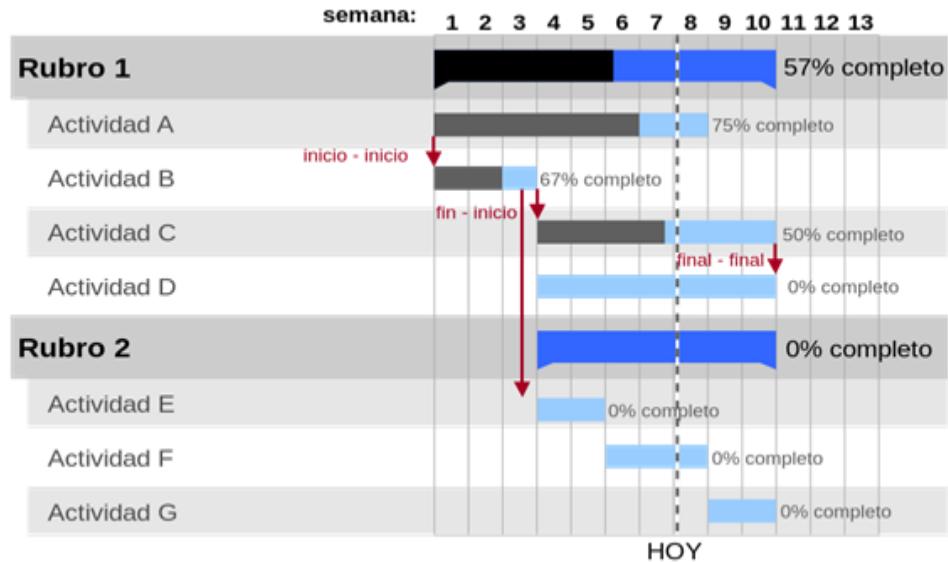


Imagen 7. Diagrama de Gantt.

5.3.1. Herramienta para generar diagrama de Gantt.

a) OpenProject

OpenProject es una aplicación de gestión de proyectos de escritorio de código abierto similar a Microsoft Project. OpenProject tiene una interfaz de usuario muy familiar e incluso abre archivos MS Project existentes (Source Forge, 2013).



Imagen 8. OpenProject logo.

Ejemplo:

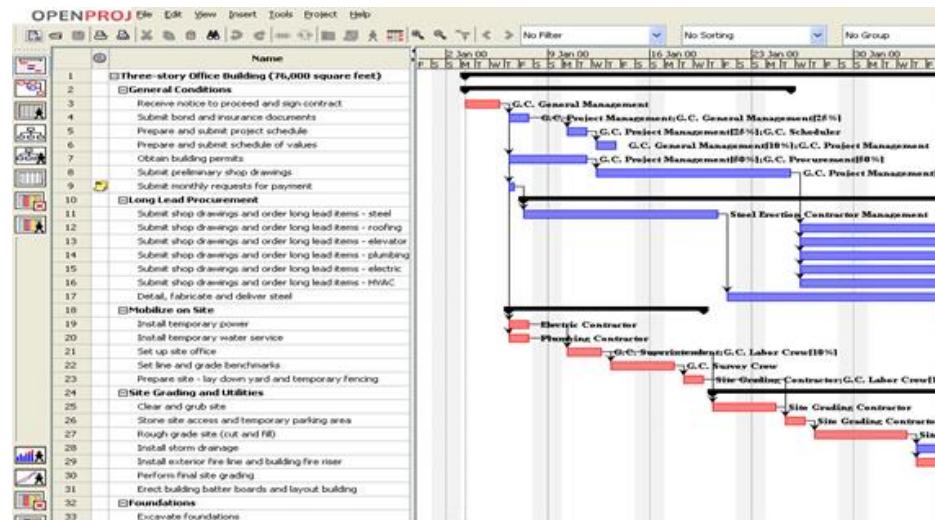


Imagen 9. OpenProject.

b) Gantt Project

Organizar tareas en una estructura de desglose de trabajo. Árbol jerárquico en el que se resumen los progresos, las fechas o los costos de las tareas de nivel inferior en los niveles superiores. Las tareas de resumen pueden colapsarse para ocultar tareas que no son importantes en este momento.

Crear tareas e hitos. Aparte de la fecha de inicio y la duración, cada tarea puede tener prioridad, color y patrón de relleno, notas de texto y campos personalizados definidos por el usuario.



Imagen 10. Gantt project logo.

Ejemplo:

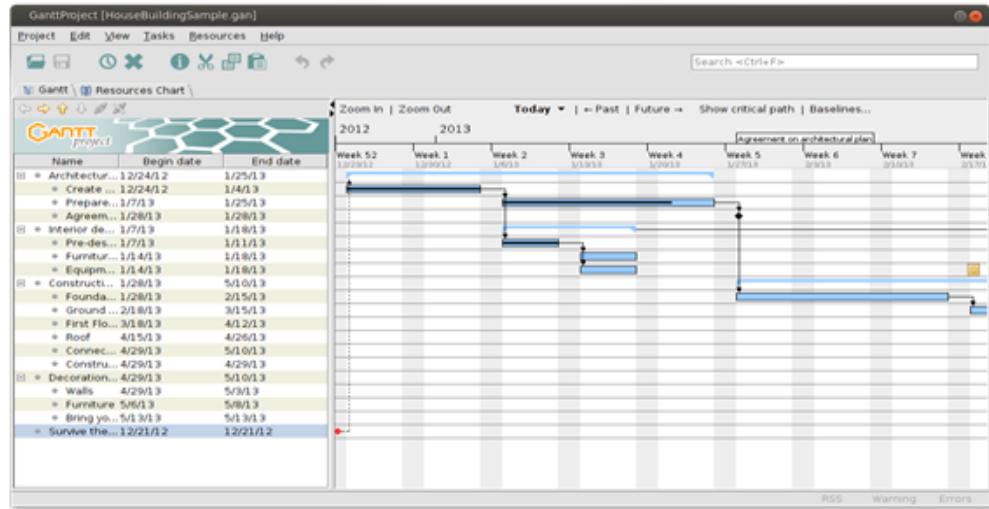


Imagen 11. Gantt Project.

c) Microsoft Project

Microsoft Project (MSP) es un software de administración de proyectos diseñado, desarrollado y comercializado por Microsoft para asistir a administradores de proyectos en el desarrollo de planes, asignación de recursos a tareas, dar seguimiento al progreso, administrar presupuesto y analizar cargas de trabajo.



Imagen 12. Microsoft Project logo.

Ejemplo:

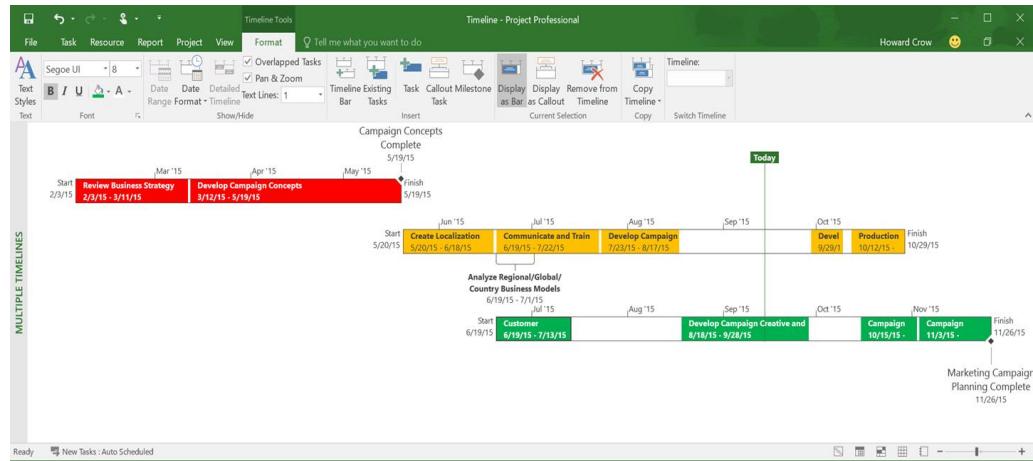


Imagen 13. Microsoft Project.

5.4. HERRAMIENTA PARA RECOLECTAR DATOS

Los analistas utilizan una variedad de métodos a fin de recopilar los datos sobre una situación existente, como entrevistas, cuestionarios, inspección de registros (revisión en el sitio) y observación. Cada uno tiene ventajas y desventajas. Generalmente, se utilizan dos o tres para complementar el trabajo de cada una y ayudar a asegurar una investigación completa.

5.4.1. Entrevista

Las entrevistas se utilizan para recabar información en forma verbal, a través de preguntas que propone el analista. Quienes responden pueden ser gerentes o empleados, los cuales son usuarios actuales del sistema existente, usuarios potenciales del sistema propuesto o aquellos que proporcionarán datos o serán afectados por la aplicación propuesta. El analista puede entrevistar al personal en forma individual o en grupos algunos analistas prefieren este método a las otras técnicas que se estudiarán más adelante. Sin embargo, las entrevistas no siempre son la mejor fuente de datos de aplicación.

Ventajas:

- Brinda la posibilidad de profundizar en cualquier tema.
- Evaluación más objetiva tanto de quienes responden como de las respuestas a las preguntas.

5.4.2. Encuesta

Hoy en día la palabra “encuesta” se usa más frecuentemente para describir un método de obtener información de una muestra de individuos. Esta “muestra” es usualmente sólo una fracción de la población bajo estudio.

Ejemplo:

El siguiente cuestionario será aplicado a los estudiantes o la muestra de la población, el cual consta de 10 ítems y con dos alternativas de tipo cerrado con la respuesta, **Sí** equivalente a: 1 o **NO** equivalente a: 0

FORMATO DE ENCUESTA PARA LOS ESTUDIANTES

PREGUNTAS	SI	NO
vives con tus padres		
realizas tus tareas escolares		
te gusta realizar dichas tareas		
tus padres te ayudan		
pasas la mayoría del tiempo sin tus padres		
eres hijo único		
tienes hermanos menos que tu		
alguno de tus padres u profesor ha tratado de pegarte		
te portas mal en clases		
tus padres o profesores te regañan mucho		

Otra técnica a utilizar sería La entrevista esta será realizada hacia los docentes y padres de familia para saber cómo es el comportamiento de dichos niños en el desarrollo de las clases y en el hogar y que métodos suelen utilizar ellos para sobrellevar la situación y como responden los dichos a dichos métodos. Para si poder medir las posibles causas en los niños y niñas de asumir tal conducta

Imagen 14. Encuesta.

5.4.3. Observación

La técnica de observación consiste en observar personas, fenómenos, hechos, casos, objetos, acciones, situaciones, etc., de los cuales se desea obtener la información necesaria para la investigación.

La técnica de observación se suele utilizar principalmente para observar el comportamiento de los consumidores en sus medios naturales.

Ventajas:

- Permite obtener información precisa que de otro modo no se podría obtener, o información que las personas no podrían o no quisieran brindar por diversos motivos, siendo una técnica fácil de aplicar y de bajo costo.

5.5. ESPECIFICACIONES DE REQUERIMIENTOS

Un requerimiento es la condición o capacidad que debe de tener un sistema, producto, servicio o componente para satisfacer un contrato, estándar, especificación, u otro documento formalmente interesados, dentro de los cuales se pueden tener alguna capacidad necesaria para el cliente o usuario que soluciona un problema o consigue un objetivo, como también puede ser alguna restricción impuesta por algún interesado (Ibañez, 2016).

Para realizar con éxito la definición de los requerimientos, es importante conseguir que sean claramente definidos para minimizar la ambigüedad, para esto es importante tener en cuenta lo siguiente:

- Definir los requerimientos teniendo en cuenta la información identificada con la perspectiva del usuario.
- Reutilizar requerimientos, revisando proyectos ya finalizados para ver si contienen material potencialmente reutilizable. La ventaja de esta reusabilidad es que, una vez que un requisito ha sido especificado

satisfactoriamente para un producto y que el producto ha tenido éxito, el requerimiento no tendrá que volverse a inventar, podrá ser utilizado las veces que se deseé teniendo en cuenta los derechos de autor.

- Se debe documentar los requerimientos de una forma clara y correcta. En la mayoría de los proyectos se observa que la documentación de los requerimientos puede parecer una tarea tediosa, pero es la única manera de asegurar que la esencia de los requisitos ha sido capturada correctamente, y que esto pueda ser probado.

5.5.1. Requerimientos funcionales

Estos requerimientos se utilizan para determinar qué hará el Software, definiendo las relaciones de su operación y su implementación, sin olvidar que deben ser explícitos también en lo que el sistema no debe hacer y qué validaciones se deben realizar, teniendo en cuenta cuál será el comportamiento del sistema.

Los requerimientos funcionales se pueden dividir en dos puntos de vista: El primero tiene relación con el usuario, donde se identifica la relación del usuario con el sistema desde el punto de vista del mismo; El segundo tiene relación con el sistema dando respuesta al usuario, es decir desde el punto de vista de lo que realiza el sistema.

Para un desarrollador de sistemas es natural dar interpretaciones de un requerimiento ambiguo con el fin de simplificar su implementación. Sin embargo, a menudo no es lo que el cliente desea. Se tienen que estipular nuevos requerimientos y se deben hacer cambios al sistema, retrasando la entrega de éste e incrementando el costo. En principio, la especificación de requerimientos funcionales de un sistema debe estar completa y ser consistente con lo solicitado por el usuario.

5.5.2. Requerimientos no funcionales

Estos requerimientos se basan en las restricciones de los servicios o funciones ofrecidos por el sistema. Incluyen restricciones de tiempo, sobre el proceso de desarrollo, estándares, usabilidad, portabilidad, entre otros.

Los Requerimientos funcionales son los requerimientos que no se refieren directamente a las funciones específicas que entrega el sistema, sino a las propiedades emergentes de éste como la fiabilidad, la respuesta en el tiempo y la capacidad de almacenamiento.

Los requerimientos no funcionales surgen de la necesidad del usuario, debido a las restricciones en el presupuesto, a las herramientas utilizadas, a las políticas de la organización, a la necesidad de interoperabilidad con otros sistemas de software o hardware o a factores externos como los reglamentos de seguridad, las políticas de privacidad, etcétera.

Los dos tipos de requerimientos especificados son de gran importancia para el desarrollo de una aplicación en software, por lo tanto siempre deben ser escritos con claridad, contener la mayor especificación de las necesidades expuestas por el cliente, esto con el fin de tener un soporte base desde el cual se trabajaran y no presentar ambigüedades en la definición y el resultado del producto. La figura a continuación muestra los inconvenientes que se pueden presentar cuando no se hace una identificación correcta de los requerimientos.

Según IEEE, un buen Documento de Requisitos, pese a no ser obligatorio que siga estrictamente la organización y el formato dados en el estándar 830, si debe incluir, de una forma o de otra, toda la información presentada en dicho estándar. El estándar de IEEE 830 no está libre de defectos ni de prejuicios, y por ello ha sido justamente criticado por múltiples

autores y desde múltiples puntos de vista, llegándose a cuestionar incluso si es realmente un estándar en el sentido habitual que tiene el término en otras ingenierías.

5.6. HERRAMIENTAS DE MODELADO UML

En todas las disciplinas de la ingeniería se hace evidente la importancia de los modelos ya que describen el aspecto y la conducta de "algo". Ese "algo" puede existir, estar en un estado de desarrollo o estar, todavía, en un estado de planeación. Es en este momento cuando los diseñadores del modelo deben investigar los requerimientos del producto terminado y dichos requerimientos pueden incluir áreas tales como funcionalidad, performance y confiabilidad. Además, a menudo, el modelo es dividido en un número de vistas, cada una de las cuales describe un aspecto específico del producto o sistema en construcción.

5.6.1. Qué es UML

UML es un lenguaje para hacer modelos y es independiente de los métodos de análisis y diseño. Existen diferencias importantes entre un método y un lenguaje de modelado. Un método es una manera explícita de estructurar el pensamiento y las acciones de cada individuo. Además, el método le dice al usuario qué hacer, cómo hacerlo, cuándo hacerlo y por qué hacerlo; mientras que el lenguaje de modelado carece de estas instrucciones. Los métodos contienen modelos y esos modelos son utilizados para describir algo y comunicar los resultados del uso del método.

5.6.2. Qué diagramas lo componen

a) Diagrama de Actividades

Los diagrama de actividades se usan para visualizar "la secuencia de actividades, el flujo de trabajo desde el punto de inicio hasta el punto final detallado muchas de las rutas de decisiones que existen en el progreso de

eventos contenidos en las actividades, son útiles para el modelado de negocios donde se usan para detallar el proceso involucrado en las actividades" (SYSTEMS, 2016). Los diagramas de actividades son de mayor utilidad para el modelo de negocios donde se usan para detallar el proceso involucrado en las actividades de negocio.

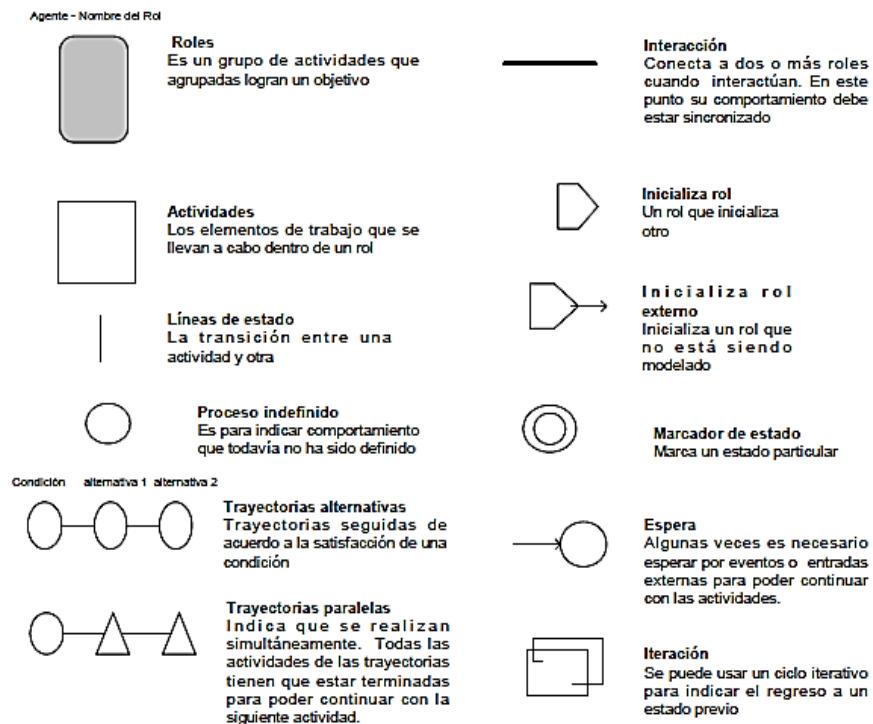


Imagen 15. Simbología diagrama de actividades.

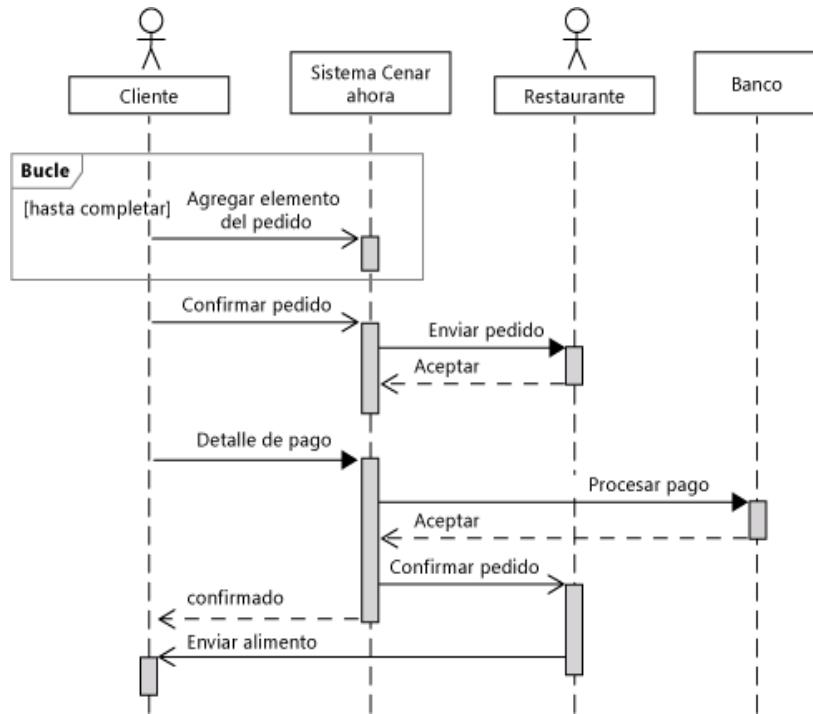


Imagen 16. Diagrama de actividades.

b) Diagrama de Casos de Uso

Un diagrama de casos de uso es una forma de diagrama de comportamiento UML mejorado (Teniente López, 2016). El Lenguaje de Modelado Unificado (UML), define una notación gráfica para representar casos de uso llamada modelo de casos de uso. UML no define estándares para que el formato escrito describa los casos de uso, y así mucha gente no entiende que esta notación gráfica define la naturaleza de un caso de uso; sin embargo una notación gráfica puede solo dar una vista general simple de un caso de uso o un conjunto de casos de uso.

SIMBOLOGIA

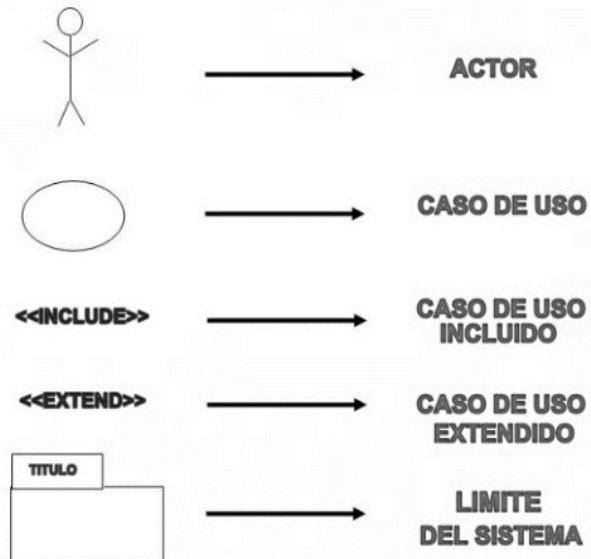


Imagen 17. Simbología diagrama de caso de uso.

Ejemplo:

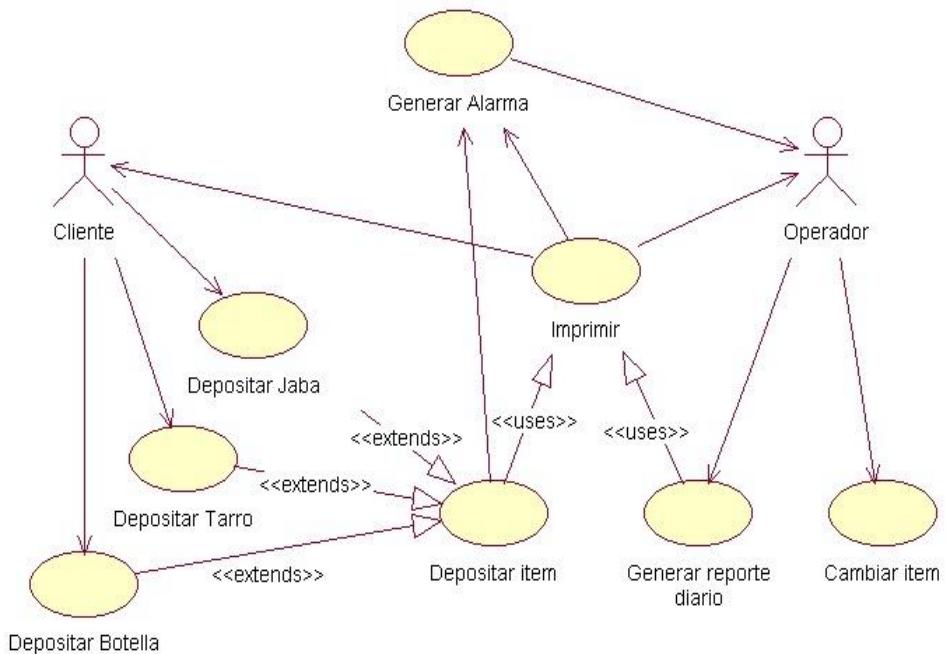


Imagen 18. Diagrama de caso de uso.

c) Diagrama de secuencia

Un diagrama de secuencia es una forma de diagrama de interacción que muestra los objetos como líneas de vida a lo largo de la página y con sus interacciones en el tiempo representadas como mensajes dibujados como flechas desde la línea de vida origen hasta la línea de vida destino. Los diagramas de secuencia son buenos para mostrar qué objetos se comunican con qué otros objetos y qué mensajes disparan esas comunicaciones. Los diagramas de secuencia no están pensados para mostrar lógicas de procedimientos complejos.

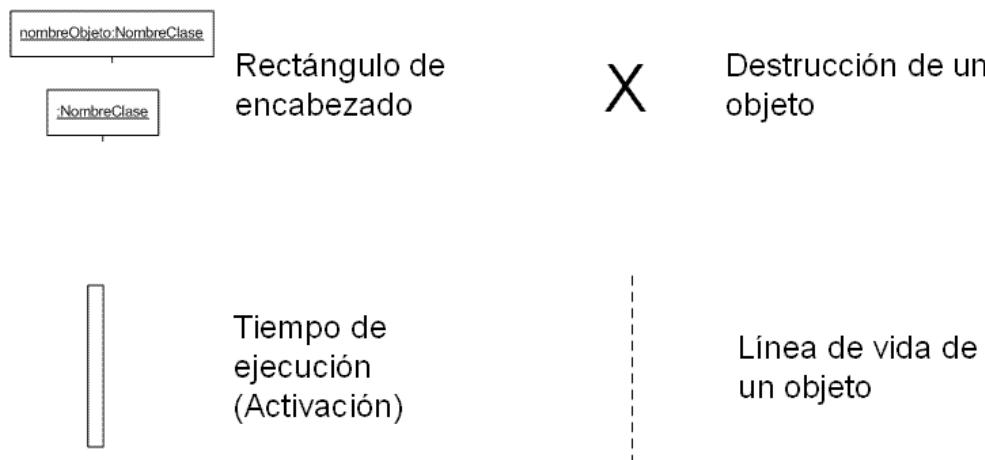


Imagen 19. Simbología diagrama de secuencia.

Ejemplo:

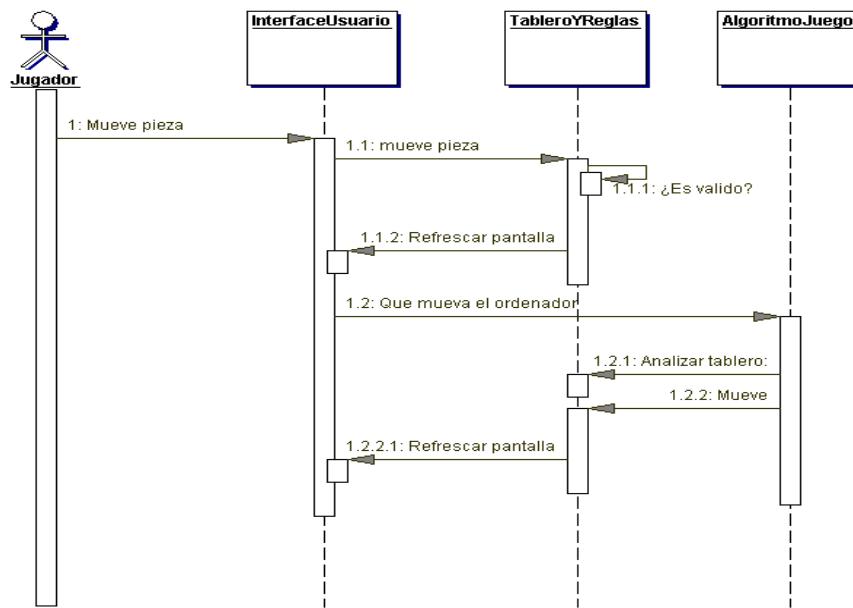


Imagen 20. Diagrama de secuencia.

d) Diagrama de Clases

Este tipo de diagramas muestra la estructura de un sistema, indicando las clases que lo integran, atributos de cada uno de ellas, cada uno de los métodos contenidos y cuales relaciones están establecidas para la clase con alguna otra de las mismas.



HERENCIA



AGREGACION



ASOCIACION



DEPENDENCIA O INSTANCIACION (USO)



Imagen 21. Simbología de diagrama de clases.

Ejemplo:

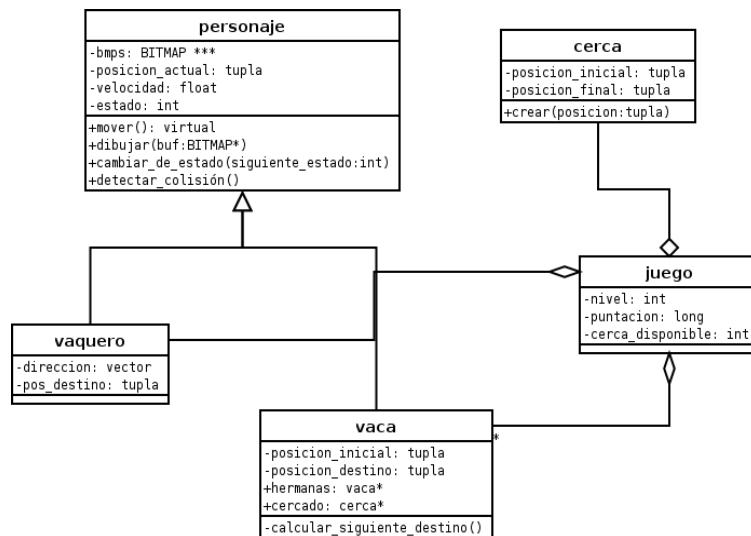


Imagen 22. Diagrama de clases.

5.6.3. Herramientas para generar diagramas de UML

a) StarUML

Es una herramienta de código abierto la cual contiene una gran diversidad de funciones para el desarrollo de diagramas UML, como los son: Diagramas de casos de uso, secuencia, clases entre algunos otros.



Imagen 23. StarUML.

b) Altova UModel

Es un software de paga para el diseño de diagramas UML, dentro de sus opciones tenemos: Diagrama de clase y objetos, estructura de un compuesto, componentes, implementación, paquetes y de perfil, además de incluir la característica de generar documentación del proyecto de forma automática.



Imagen 24. ALTOVA UModel.

c) draw.io

Es una herramienta para el diseño de diagramas UML, contando con la característica de ser ejecutada por medio de un navegador web, y permitir guardar los archivos en la nube o poder descargarlos a algún dispositivo de almacenamiento.

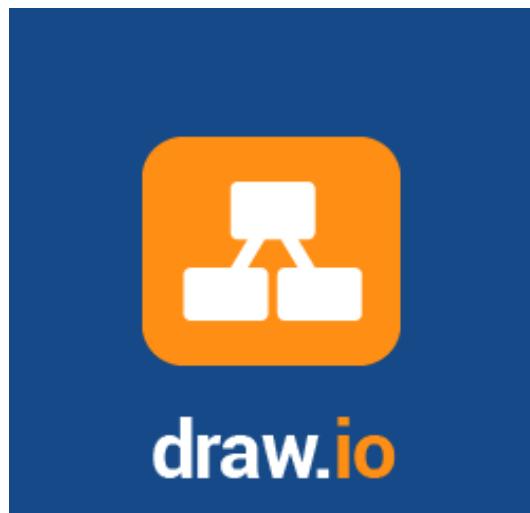


Imagen 25. draw.io.

5.7. HERRAMIENTAS PARA LE DISEÑO DE BASE DE DATOS

5.7.1. Modelo Relacional

Es un modelo de datos basado en la lógica de predicados y en la teoría de conjuntos. Es el modelo más utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente (Ferron, 2016).

Su idea fundamental es el uso de relaciones, las cuales podrían considerarse de forma lógica como conjuntos de datos llamados tuplas que es igual a que cada relación como si fuese una tabla que está compuesta por registros.

5.7.2. Software para diseñar el modelo relacional

a) PowerDesigner

Es un software que proporciona un único entorno de modelado que reúne las técnicas y notaciones de modelado de proceso de negocios y requerimientos, modelado de datos, modelado de la arquitectura de la empresa y el modelado de aplicaciones UML. Algunas de sus características son: generación de código, modelado de datos, generación de informes, análisis de requerimientos.



Imagen 26. PowerDesigner.

b) FIDIAS DBM

Es un software de modelador de bases de datos. Esta herramienta es muy útil a la hora de realizar un diseño de base de datos nuevos o preexistentes. La capacidad de realizar un diseño de modelo visualizándolo de forma gráfica, a través de los Diagramas de Entidad Relación, aumenta enormemente la eficiencia de nuestro trabajo reduciendo drásticamente el tiempo necesario con respecto a una realización manual de dichas tareas de diseño, análisis y estudio. Algunas de sus características básicas son: trabajo simultáneo en varios proyectos, generación automática del modelo relacional, generación automática de sentencias ANSI SQL.



Imagen 27. Fidias.

c) DB Designer

Es una herramienta para la construcción de bases de datos SQL, además de contar con una interfaz intuitiva y fácil de usar, en donde se tiene la representación virtual de las tablas y las relaciones contenidas en su proyecto, algunas de sus características son: cuenta con modos de interacción, cuenta con la de diseños en la que para crear y mantener el modelo de la base de datos es visual y el modo consulta se utiliza para trabajar con datos de la tabla y construir instrucciones de consulta SQL complejas para el uso de PHP, Kylix entre otros más.



Imagen 28. DB Designer.

d) MySQL Workbench

MySQL Workbench es una herramienta visual de diseño de bases de datos que integra desarrollo de software, Administración de bases de datos, diseño de bases de datos, creación y mantenimiento para el sistema de base de datos MySQL. Es el sucesor de DBDesigner 4 de fabFORCE.net, y reemplaza el anterior conjunto de software, MySQL GUI Tools Bundle.



Imagen 29. MySQL Workbench.

5.8. HERRAMIENTA DE MAQUETADO

5.8.1. Diseño de bocetos

a) Balsamiq

Balsamiq es un software de maquetado utilizando es una interfaz gráfica de usuario. Permite que el diseñador ordene o pre-construir la los widgets de manera dinámica arrastrando y soltando. La aplicación se ofrece en un escritorio de la versión, así como un plug-in para Google Drive, Confluencia y JIRA.



Imagen 30. Balsamiq.

Ventajas:

- La principal ventaja es la productividad, o dicho de otra manera, lo rápido que se puede hacer cualquier cosa. El hecho que cuente con tantos componentes ahorra mucho tiempo.
- Hay algunos componentes que están muy bien resueltos, y una vez que te metes a editarlos te dan un buen número de opciones.
- Como manejo general, permite algunas opciones sobre los controles como agruparlos, bloquearlos o posicionarlos en función de otros componentes.

Desventajas:

- Algunos componentes se quedan cortos en opciones.
- Aunque el ajuste automático se agradece al principio, cuando trabajas con muchos controles se echa en falta un sistema más preciso de ajuste (entrar coordenadas, reglas, etc).
- El canvas es dinámico, en función de lo que vas añadiendo utiliza scroll horizontal y vertical, pero no puedes fijar de entrada el tamaño del documento.

- El look and feel es cachondo y cumple con el cometido de evitar que nos centremos en el diseño, pero para algunos clientes puede resultar demasiado infantil o poco profesional, y no hay (por ahora) más alternativas.
- Es imposible hacer una navegación mínima ya que no permite ningún tipo de interactividad.

b) Axure

Es una herramienta orientada a diseñar wireframes y prototipos básicos o avanzados de forma fácil, incluso con aspecto de sketch y en la medida de los conocimientos y experiencia del arquitecto de información, incluirles más funcionalidades que simulen mejor las interacciones propuestas en el proyecto.



Imagen 31. Axure.

Ventajas:

- Interactividad que simula comportamiento de las interfaces gracias a su fortaleza como herramienta de prototipado.
- Funciona en cualquier navegador.
- Posibilidad de crear masters o documentos maestros, los cuales son muy útiles pues son reutilizables en todas las interfaces que se diseñen.

Desventajas:

- Una de las que podríamos llamar desventajas es su precio: 589 dólares.

c) UXPin

UXPin es un conjunto de herramientas de Experiencia de Usuario. Creado por diseñadores UX para cualquier persona que se preocupe por diseño que realmente importe.



Imagen 32. UXPin.

Ventajas:

- Creación rápida, fácil y de colaboración de wireframes y prototipos UX de forma ágil.
- UXPin permite crear iteraciones y control de versiones de documentos de nuestros archivos.
- Demostrar el proceso de diseño UX a compañeros y clientes.
- Colaborar con nuestros compañeros de equipo en todo el proceso de diseño.
- Recopilación de sugerencias y comentarios.

5.8.2. Frameworks de maquetado

a) YamL

YamL es un formato de serialización de datos legible por humanos, inspirado en lenguajes como XML, C, Python, Perl.

YamL fue creado bajo la creencia de que todos los datos pueden ser representados adecuadamente como combinaciones de listas, hashes (mapeos) y datos escalares (valores simples). La sintaxis es relativamente sencilla y fue diseñada teniendo en cuenta que fuera muy legible pero que a la vez fuese fácilmente mapeable a los tipos de datos más comunes en la mayoría de los lenguajes de alto nivel.

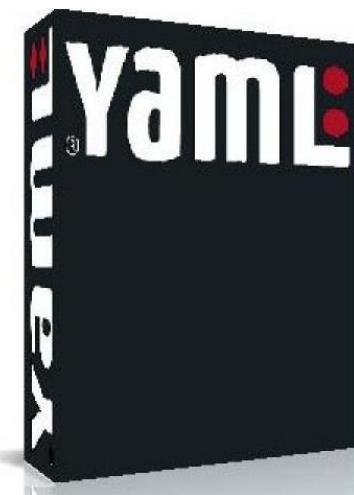


Imagen 33. YamL.

b) Pure

Pure es un Framework en desarrollo y mejora incluye por el momento tipografías, rejillas, formularios, botones, tablas y elementos de navegación. El código empleado en Pure CSS es muy ligero con un peso minimizado de 5.7KB. Sin embargo, es modular, por lo que es posible usar los estilos de cada módulo por separado.

Pure está basado en Normalize.CSS y no utiliza JavaScript, simplemente HTML-CSS por lo que es posible utilizarlo en un proyecto web cualquiera.



Imagen 34. Pure.

c) Boostrap

Es un conjunto de herramientas para el diseño de sitios y aplicaciones web, algunas de las plantillas que contiene son: tipografía, formularios, botones, cuadros, menú de navegación entre algunos otros elementos de diseño basado en HTML y CSS, además de extensiones de JavaScript adicionales, se encuentra bajo una licencia de uso libre.

Algunas de sus características principales son: compatibilidad con la gran mayoría de los navegadores, contiene diseños sensibles o responsivos lo que permite ajustar dinámicamente la página al dispositivo donde se esté visualizando como computadoras, tabletas o teléfonos inteligentes.



Imagen 35. Boostrap.

5.9. ENTORNOS DE PROGRAMACIÓN

5.9.1. NetBeans

NetBeans es un entorno de desarrollo integrado o entorno de desarrollo interactivo, en inglés Integrated Development Environment (IDE) que permite programar en distintos lenguajes, es ideal para trabajar con el lenguaje de desarrollo JAVA y todos sus derivados, además ofrece un excelente entorno para programar en PHP. También se puede descargar una vez instalado NetBeans, los complementos para programar en C++. La IDE de NetBeans es perfecta y muy cómoda para los programadores. Tiene un excelente balance entre una interfaz con múltiples opciones y un aceptable completamiento de código.



Imagen 36. NetBeans.

Ventajas:

- Admite distintos tipos de lenguaje.
- Posibilidad de agregar nuevos módulos para aumentar su funcionalidad.
- Disponibilidad en muchos idiomas.
- Desarrollo intuitivo y guiado.

Desventajas:

- Una de sus desventajas es el gran consumo de CPU y de RAM.

5.9.2. IntelliJ IDEA

IntelliJ IDEA es un entorno de desarrollo Java creado por Jet Brains del que existen dos distribuciones: Community Edition (open source) y Ultimate.



Imagen 37. IntelliJ IDEA.

Ventajas:

- Autocompletado de código. No se cómo lo hace, pero siempre te propone justo eso que necesitas.
- Integración con sistemas de control de versiones.
- Viene con todo instalado de serie.
- Existe un amplio set de plugins.
- Sensación de fiabilidad y robustez muy superior a otros entornos.
- Herramienta de refactorización extremadamente inteligente.

Desventajas:

- Es de pago.

5.9.3. Eclipse

Entorno de desarrollo integrado de código abierto multiplataforma para desarrollar proyectos. Esta plataforma ha sido usada para desarrollar entornos de desarrollo integrados, como el IDE de Java llamado Java Development Toolkit (JDT) y el compilador (ECJ) que se entrega como parte de Eclipse. En Eclipse se pueden usar diferentes lenguajes de programación como: Java, ANCI C, C++, JSP, sh, perl, php, sed.

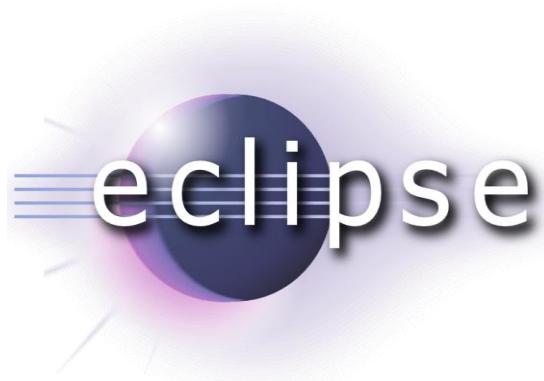


Imagen 38. Eclipse.

Ventajas:

- Dispone de un editor de texto con resaltador de sintaxis.
- La compilación es en tiempo real
- Tiene pruebas unitarias con (JUnit).
- Control de versiones con (CVS).
- Integración con (Ant),
- Asistentes (wizards) para creación de proyectos.

Desventajas:

- Su principalmente inconveniente, común a otros IDEs en mayor o menor medida, en el consumo de recursos del sistema.

- También que carece de mucho soporte para webapps (.war, jsp y servlets) tal y como lo hace netbeans. (los plugins por lo general no son ni tan potentes ni tan sencillos como el módulo que en Netbeans viene preinstalado).

5.9.4. Sublime Text

Sublime Text es un editor de texto pensado para escribir código en la mayoría de lenguajes de programación y formatos de texto más empleados en la actualidad como Java, Python, Perl, HTML, JavaScript, CSS, HTML, XML, PHP, C, C++, etc. Las principales características que tiene son la posibilidad de colorear el texto y corregir la sintaxis y ortografía del código; pero quizás su característica más apreciada por los desarrolladores es que admite el uso de abreviaturas conocidas como “snippets”. Sublime Text tiene un costo, pero permite su uso ilimitado en el tiempo de manera gratuita, sin embargo, es recomendable pagar su costo para impulsar su desarrollo.



Imagen 39. Sublime Text.

Ventajas:

- Su rapidez en la ejecución.
- Es muy ligero (7 Mbytes).
- Funcionalidades comparables a los entornos de desarrollo integrados.
- Multilenguaje y multiplataforma.
- Gestión de proyectos completos.

- Posibilidad de ejecutar y depurar código sin salir del editor.
- Comunidad de usuarios cada vez mayor.

5.10. LENGUAJES DE PROGRAMACIÓN

Un lenguaje de programación es un conjunto de instrucciones conformadas por símbolos, los cuales permiten manipular el comportamiento físico o lógico de una máquina, como puede ser una computadora.

Estos lenguajes por lo regular se componen de los siguientes pasos: se escribe, prueba, depura y se compila.

5.10.1. JavaScript

Es un lenguaje de programación interpretado, orientado a objetos “La POO nos dice que todo aquello que podemos manejar es un objeto, es decir, la ventana donde se carga una página web es un objeto, la página web en si también es un objeto” (Quijado J. L., 2007), basado en prototipos y dinámico. Es un lenguaje del lado del cliente, por lo que es interpretado por el navegador web, ofreciendo la posibilidad de mejorar la interfaz del usuario y páginas dinámicas, además de ser aplicado fuera de la web, como lo son dentro de widgets y documentos PDF principalmente, “Se emplea en una página web para dotarla de unas funcionalidades que el HTML no puede proporcionar por sí mismo” (Quijado J. L., 2007).

Este lenguaje se caracteriza por tener una sintaxis semejante a la de C y convenciones del lenguaje de programación Java, además de la implementación de AJAX “Asynchronous JavaScript And XML, hace referencia a un conjunto de tecnologías de uso común en la Web que, combinadas adecuadamente, permiten mejorar enormemente la experiencia del usuario con las aplicaciones de Internet” (Sierra, 2007) también se define como “Un

mecanismo de combinación de tecnologías y estándares de cliente, consistente en la solicitud asíncrona de datos al servidor desde una página Web y la utilización de estos para la actualización de una parte de la misma, sin obligar al navegador a realizar una recarga completa de toda la página” (Sierra, 2007), aunque este último no tenga ninguna relación directa con JavaScript.



Imagen 40. JavaScript.

Ventajas:

- Es compatible con la mayoría de la estructura de programación de C.
- Dispone de un tipado dinámico, lo que ofrece la posibilidad de usar una misma variable para almacenar diferentes tipos de valores.
- Se encuentra enfocado principalmente a la programación orientada a objetos.
- Se encuentra bajo la organización Mozilla Foundation.
- Permite realizar validación dentro de formularios, ejecutados del lado del cliente.
- Ofrece la posibilidad de crear animación, así como imágenes dinámicas o presentación de diapositivas.

Desventajas:

- No cuenta con un nivel de seguridad considerable.
- Requiere de introducir grandes cantidades de código dentro de un sitio web.
- Para acelerar la carga y ejecución de un código de este tipo, se requiere organizarlo en un archivo adjunto con la extensión js.

5.10.2. PHP

Es un lenguaje interpretado del lado del servidor, el cual ofrece la posibilidad de crear aplicaciones web con contenido dinámico, enviando al cliente una información determinada a diferencia del resto de los clientes conectados al mismo servidor.

Este lenguaje se puede combinar con otros para crear aplicaciones más complejas, algunas de estos lenguajes son: HTML 5 y JavaScript, dando así al programador una mayor libertad y flexibilidad del funcionamiento de las aplicaciones web.



Imagen 41. Php.

Ventajas:

- Ofrece la posibilidad de acceder de forma dinámica a información almacenada en una base de datos.

- Es considerado un lenguaje de programación relativamente fácil de aprender.
- Brinda un considerable nivel de seguridad, pues este código no es visible ni interpretado por el cliente (navegador web).
- Es de uso libre, ofreciendo una gran disponibilidad de documentación para su entendimiento.
- Cuenta con capacidad para manejo con diversos motores de bases de datos.
- Tiene la capacidad de aplicar técnicas de programación orientada a objetos.

Desventajas:

- Al ser un lenguaje interpretado, el código no puede ser ocultado.
- Debido a ser un lenguaje interpretado, éste es más lento al momento de ser ejecutado a diferencia de uno de bajo nivel.
- Ocasiona un cierto nivel de confusión al momento del uso de variables, en ciertos IDEs, pues las variables no se clasifican por tipos.

5.10.3. HTML5

Es un lenguaje de marcado, el cual funciona basado en etiquetas, diseñado para la elaboración de sitios web, éste define principalmente una estructura básica, además de un código dentro del cual se especifica el contenido de la página, como lo es texto, imágenes, videos, links entre otros más “HTML es el corazón de cualquier página Web” (Crowder, 2005).



Imagen 42. HTML5.

Ventajas:

- Reduce la cantidad de plugins necesarios.
- Tiene un mejor manejo de errores.
- El código es simple, permitiendo crear páginas ligeras, de carga rápida.
- Mayor compatibilidad con dispositivos móviles.
- Dispone de la capacidad para realizar diseños a distintos dispositivos, como lo son Smartphone, Tablet y pc.
- Es de uso público o libre.

Desventajas:

- HTML 5 no está completamente terminado, y aun pueden aplicarse cambios.
- Aun no es compatible con todos los navegadores.

5.11. PRUEBAS

5.11.1. Pruebas unitarias

Este tipo de pruebas son ejecutadas normalmente por el equipo de desarrollo, básicamente consisten en la ejecución de actividades que le permitan verificar al desarrollador que los componentes unitarios están codificados bajo condiciones de robustez, esto es, soportando el ingreso de datos erróneos o inesperados y demostrando así la capacidad de tratar errores de manera controlada.

5.11.2. Pruebas de integración

Este tipo de pruebas son ejecutadas por el equipo de desarrollo y consisten en la comprobación de que elementos del software que interactúan entre sí, funcionan de manera correcta.

Consiste en realizar pruebas para verificar que un gran conjunto de partes de software funciona junto, es la fase de pruebas de software en la cual módulos individuales de software son combinados y probados con un grupo.

6. METODOLOGÍA

Parte del proceso de investigación o método científico, que permite sistematizar los métodos y las técnicas necesarias para llevarla a cabo. (Hernández, Fernández, & Baptista, 2006).

6.1. PLANEACIÓN DE LAS ACTIVIDADES PARA LA CONSTRUCCIÓN DE APLICACIÓN

6.1.1. Ciclo de vida que se utilizó para el desarrollarlo

Teniendo en cuenta los diferentes tipos de ciclos de vida, se llegó a la elección del ciclo iterativo, ya que este permite que el trabajo como el equipo que lo desarrolla van evolucionando a lo largo de las etapas.

Una de las principales ventajas que nos ofrece este modelo o ciclo de vida es que no hace falta que los requisitos estén totalmente definidos al inicio del desarrollo, sino que se pueden ir mejorando en cada una de las iteraciones. Igual que otros modelos similares tiene las ventajas propias de realizar el desarrollo en pequeños ciclos, lo que permite gestionar mejor los riesgos, gestionar mejor las entregas.

La finalidad del ciclo de vida es al final de cada iteración se le entrega al cliente una versión mejorada o con mayores funcionalidades del producto. El cliente es quien después de cada iteración evalúa el producto y lo corrige o propone mejoras. Estas iteraciones se repetirán hasta obtener un producto que satisfaga las necesidades del cliente. En este caso el ciclo de vida tendrá variantes ya que se adaptará para su mejor uso. Teniendo en cuenta los diferentes tipos de ciclos de vida, se llegó a la elección de utilizar el ciclo iterativo, ya que este permite que el trabajo como el equipo que lo desarrolla van evolucionando a lo largo de las etapas.

Una de las principales ventajas que nos ofrece este modelo o ciclo de vida es que no hace falta que los requisitos estén totalmente definidos al inicio del desarrollo, sino que se pueden ir mejorando en cada una de las iteraciones. Igual que otros modelos similares tiene las ventajas propias de realizar el desarrollo en pequeños ciclos, lo que permite gestionar mejor los riesgos, gestionar mejor las entregas.

La finalidad del ciclo de vida es al final de cada iteración se le entrega al cliente una versión mejorada o con mayores funcionalidades del producto. El cliente es quien después de cada iteración evalúa el producto y lo corrige o propone mejoras. Estas iteraciones se repetirán hasta obtener un producto que satisfaga las necesidades del cliente. En este caso el ciclo de vida tendrá variantes ya que se adaptará para su mejor uso, dando como resultado la elaboración de módulos por cada uno de los ciclos desarrollados en el ciclo de vida. Dando el inicio del siguiente modulo con la terminación del anterior.

6.1.2. Herramientas que se utilizó para realizar el diagrama de Gantt

La herramienta que se utilizará para realizar el diagrama de Gantt será Microsoft Project. Ya que siendo el diagrama de Gantt una herramienta muy gráfica, en el que cualquier persona puede comprender fácilmente cuáles son las etapas del proceso Microsoft Project nos proporciona una visualización del diagrama de Gantt que nos permite situarlo en un lugar visible y en formato grande, donde todo usuario pueda ubicarlo de manera fácil.

Esto nos ayudará a que se recuerden los objetivos y se conozca cuándo van a tener lugar las acciones planificadas. Agregando que Microsoft Project uno del principal software para la gestión de proyectos, ya se tenía conocimiento de él y de su manejo, desarrollando y gestionando proyectos con anterioridad.

6.2. RECOPIALAR EL MATERIAL NECESARIO

6.2.1. Herramientas seleccionadas para la recolectar material

Para la recolección de datos se optó por implementar como herramienta de recolección, la entrevista ya que permite conocer en palabras claras y precisas del entrevistado una descripción detallada, de igual manera como puntualizar las observaciones que el entrevistado quiera dar a conocer.

Durante la entrevista, se llegó a la conclusión de que también fue empleada la observación, ya que permitió conocer detalles, para la recolección de datos.

6.2.2. Normas requeridas para especificar requerimientos

El estándar IEEE 830-1998 para el SRS (Software Requirements Specification) o ERS (Especificación de requerimientos de software) fue implementado para la especificación de los requerimientos o requisitos de software el cual tiene como producto final la documentación de los acuerdos entre el cliente y/o el grupo de desarrollo para así cumplir con la totalidad de exigencias estipuladas.

Definir los requerimientos de un software puede parecer algo sencillo, pero común que el usuario o el desarrollador cometan errores u omisiones importantes. Muchos problemas de los proyectos de software se deben a una adecuada especificación de requerimientos. Para definir los requerimientos de un software, podemos apoyarnos en la norma IEEE 830, que nos guía para hacer las preguntas pertinentes. Esta norma le sirve tanto al usuario como al desarrollador. El propósito principal de esta norma es ayudarnos a elaborar un documento muy útil, SRS (Software Requirements Specification) o ERS (Especificación de requerimientos de software).

Objetivos que permiten establecer los requerimientos establecidos sean correctos:

- Un cliente describa claramente lo que quiere.
- Un proveedor entienda claramente lo que el cliente quiere.
- Se establezcan bases para un contrato de desarrollo.
- Se reduzca el esfuerzo de análisis, diseño, y programación.
- Se mencionará el nombre al futuro sistema.
- Se explicará lo que el sistema hará y lo que no hará.
- Se describirán los beneficios, objetivos y metas que se espera alcanzar con el futuro sistema.

6.3. ANALIZAR INFORMACIÓN RECOLLECTADA PARA DISEÑAR LOS DIAGRAMAS DE UML

Para realizar el análisis de la información recabada se implementó los diagramas UML utilizando la herramienta StartUML.

Se implementaron los diagramas de caso de uso, este nos permite conocer las diferentes interacciones en un determinado escenario involucrando como los actores que interviene. De igual manera se implementaron diagramas de secuencia, estos en base a una línea temporal se muestra las acciones, así como el orden en que son ejecutadas. El diagrama de clases también se utilizó, este representa la interacción entre las diferentes clases, así como identificar los diferentes atributos y métodos que la componen.

6.4. DISEÑAR LOS DIAGRAMAS DE LA BASE DE DATOS

6.4.1. Herramienta seleccionada para el Diagrama Relacional

En la elaboración del diagrama relacional se utilizó la herramienta MySQL Workbench, esta herramienta cuenta con las características esenciales, de igual manera cuenta con una licencia de uso libre.

6.5. DISEÑAR LOS DIAGRAMAS DE INTERACCIÓN DE LA ETAPA DE DISEÑO

En la realización del diagrama de interacción se elegido por utilizar la herramienta Balsamiq Mockups, ya que nos permite diseñar de manera rápida, además de ofrecer una pre visualización de la interacción que contiene el diagrama.

6.6. DISEÑAR LOS BOCETOS DEL SITIO WEB

En la elaboración de los bocetos que se requieren para el desarrollo de la aplicación web se utilizó la herramienta Balsamiq Mockups, la cual nos permite crear de manera fácil gracias a su interface y herramientas con la que cuenta. Al poder realizar las visualizaciones de cada interface de la aplicación web, permite verificar la relación e interacción que tendrás interfaces.

6.7. CODIFICAR EL SITIO WEB

Se seleccionó dos tipos de lenguajes de programación. El primer lenguaje seleccionado para la codificación del sitio web fue PHP ya que cubre con las características para la construcción del proyecto, ofreciendo la posibilidad de conjuntarse con HTML5, siendo este lenguaje basado en etiquetas siendo así necesario para la parte grafica de la aplicación web.

En conjunto estos diferentes lenguajes de programación, hace una perfecta combinación que permiten cumplir con los alcances establecidos en el

proyecto, haciendo uso de los frameworks, como lo es Codeigniter a través del uso de PHP, de igual manera Bootstrap para las hojas de estilo en cascada, así como el correcto funcionamiento a través del servidor Apache desde donde se incorporara a la aplicación web.

6.8. PROBAR EL SITIO WEB

Para la implementación de las pruebas de funcionamiento de la aplicación web para su correcto funcionamiento, se implementan pruebas unitarias que a la par al ciclo de vida permiten hacer una evaluación y verificación por módulos, corroborando los resultados deseados con los obtenidos, para constatar su funcionalidad, de igual manera para realizar las modificaciones pertinentes.

7. RESULTADOS

Algunos autores hacen mención que “los resultados deben ofrecer una respuesta al planteamiento del problema y fundamentar las estrategias que se usaron para abordarlo, así como los datos recolectados, analizados e interpretados por el investigador” (Hernández, Fernández, & Baptista, 2010).

7.1. ACTIVIDADES PARA LA CONSTRUCCIÓN DE LA APLICACIÓN

7.1.1. Ciclo de vida iterativo

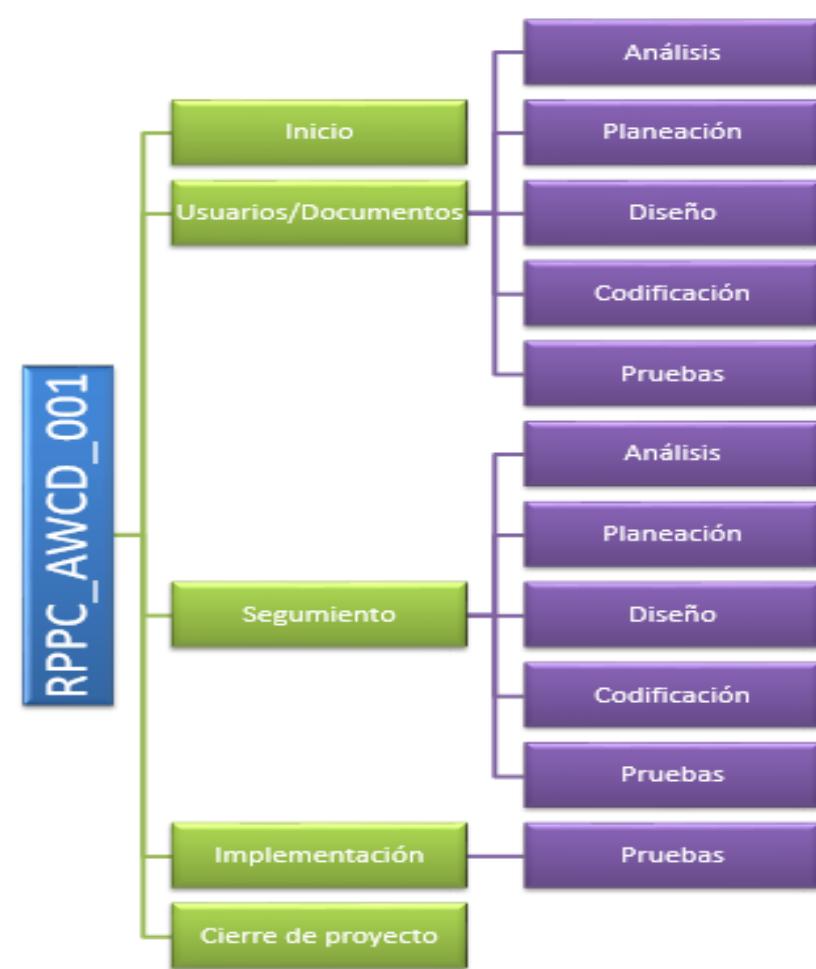


Figura 3. Ciclo de vida iterativo.

7.1.2. Diagrama de Gantt

	Mc de tar	EDT	Nombre de tarea	Duración	Comienzo	Fin
1	1	RPPC_APCT_001	39.28 días	lun 09/01/17	jun 27/02/17	
2	1.1	Formación del equipo	2 horas	lun 09/01/17	lun 09/01/17	
3	1.2	▲ Fase/Iteración 1	24.88 días	lun 09/01/17	jue 09/02/17	
4	1.2.1	▲ Análisis	5.38 días	lun 09/01/17	jun 16/01/17	
5	1.2.1.1	▲ Elicitación	2.38 días	lun 09/01/17	mié 11/01/17	
6	1.2.1.1.1	Definir estrategia de recolección de datos	1 hora	lun 09/01/17	lun 09/01/17	
7	1.2.1.1.2	Planificación de reuniones de licitación	2 horas	lun 09/01/17	lun 09/01/17	
8	1.2.1.1.3	Identificar procesos del sistema	8 horas	lun 09/01/17	mar 10/01/17	
9	1.2.1.1.4	Identificar requerimientos funcionales	6 horas	mar 10/01/17	mié 11/01/17	
10	1.2.1.1.5	Identificar requerimientos no funcionales	2 horas	mié 11/01/17	mié 11/01/17	
11	1.2.1.2	▲ Especificación	2.25 días	mié 11/01/17	vie 13/01/17	
12	1.2.1.2.1	Evaluación de requerimientos	2 horas	mié 11/01/17	mié 11/01/17	
13	1.2.1.2.2	SRS	2 días	mié 11/01/17	vie 13/01/17	
14	1.2.1.3	▲ Gestión de requerimientos	0.75 días	vie 13/01/17	jun 16/01/17	
15	1.2.1.3.1	Priorizar requerimientos	2 horas	vie 13/01/17	sáb 14/01/17	
16	1.2.1.3.2	Almacenar en repositorio	1 hora	sáb 14/01/17	sáb 14/01/17	
17	1.2.1.3.3	Asignar a desarrolladores de software	1 hora	sáb 14/01/17	sáb 14/01/17	
18	1.2.1.3.4	Administración de cambios de requerimientos	1 hora	sáb 14/01/17	sáb 14/01/17	
19	1.2.1.3.5	Administración de la trazabilidad de requerimientos	1 hora	lun 16/01/17	lun 16/01/17	
20	1.2.2	▲ Planificación	2.5 días	lun 16/01/17	mié 18/01/17	
21	1.2.2.1	Definición del ciclo de vida	2 horas	lun 16/01/17	lun 16/01/17	
22	1.2.2.2	Definición de políticas	1 hora	lun 16/01/17	lun 16/01/17	

Imagen 43. Diagrama de Gantt 1.

22	1.2.2.2	Definición de políticas	1 hora	lun 16/01/17	lun 16/01/17
23	1.2.2.3	Actualizar plan de proyecto	1 hora	lun 16/01/17	lun 16/01/17
24	1.2.2.4	▲ Estimación	2 días	lun 16/01/17	mié 18/01/17
25	1.2.2.4.1	Actualizar WBS/Calendario	1 día	lun 16/01/17	mar 17/01/17
26	1.2.2.4.2	Diccionario del WBS	1 día	mar 17/01/17	mié 18/01/17
27	1.2.3	▲ Diseño	9.75 días	mié 18/01/17	mar 31/01/17
28	1.2.3.1	Definir la arquitectura	3 horas	mié 18/01/17	mié 18/01/17
29	1.2.3.2	Realizar casos de uso	1 día	jue 19/01/17	jue 19/01/17
30	1.2.3.3	Realizar diagramas de secuencia	1 día	vie 20/01/17	vie 20/01/17
31	1.2.3.4	Realizar diagramas de clases	2 días	sáb 21/01/17	mar 24/01/17
32	1.2.3.5	Realizar diagrama de actividades	1 día	mar 24/01/17	mié 25/01/17
33	1.2.3.6	Realizar el mapa de navegación	3 horas	mié 25/01/17	mié 25/01/17
34	1.2.3.7	Realizar maquetado (bocetos)	2 días	mié 25/01/17	vie 27/01/17
35	1.2.3.8	▲ Diseño de diagramas de la base de datos	2 días	vie 27/01/17	mar 31/01/17
36	1.2.3.8.1	Realizar el diagrama E-R	1 día	vie 27/01/17	lun 30/01/17
37	1.2.3.8.2	Realizar el diagrama Relacional	1 día	lun 30/01/17	mar 31/01/17
38	1.2.4	▲ Codificación	5.25 días	mar 31/01/17	mar 07/02/17
39	1.2.4.1	Codificación de la base de datos	3 horas	mar 31/01/17	mar 31/01/17
40	1.2.4.2	▲ Modulo Usuarios	4.88 días	mar 31/01/17	mar 07/02/17
41	1.2.4.2.1	Codificar interfaz	2 días	mar 31/01/17	jue 02/02/17
42	1.2.4.2.2	Clases de Usuarios	2 días	jue 02/02/17	lun 06/02/17
43	1.2.4.2.3	Realizar conexión a la base de datos	1 hora	lun 06/02/17	lun 06/02/17
44	1.2.4.2.4	Alta de Usuario	1 hora	lun 06/02/17	lun 06/02/17
45	1.2.4.2.5	Baja de Usuario	1 hora	lun 06/02/17	lun 06/02/17
46	1.2.4.2.6	Actualización de Usuario	2 horas	lun 06/02/17	lun 06/02/17
47	1.2.4.2.7	Consultas de Usuario	2 horas	lun 06/02/17	mar 07/02/17
48	1.2.5	▲ Pruebas	2 días	mar 07/02/17	jue 09/02/17

Imagen 44. Diagrama de Gantt 2.

DIAGRAMA DE GANTT

31	1.2.3.4	Realizar diagramas de clases	2 días	sáb 21/01/17	mar 24/01/17
32	1.2.3.5	Realizar diagrama de actividades	1 día	mar 24/01/17	mié 25/01/17
33	1.2.3.6	Realizar el mapa de navegación	3 horas	mié 25/01/17	mié 25/01/17
34	1.2.3.7	Realizar maquetado (bocetos)	2 días	mié 25/01/17	vie 27/01/17
35	1.2.3.8	▷ Diseño de diagramas de la base de datos	2 días	vie 27/01/17	mar 31/01/17
36	1.2.3.8.1	Realizar el diagrama E-R	1 día	vie 27/01/17	lun 30/01/17
37	1.2.3.8.2	Realizar el diagrama Relacional	1 día	lun 30/01/17	mar 31/01/17
38	1.2.4	▷ Codificación	5.25 días	mar 31/01/17	mar 07/02/17
39	1.2.4.1	Codificación de la base de datos	3 horas	mar 31/01/17	mar 31/01/17
40	1.2.4.2	▷ Módulo Usuarios	4.88 días	mar 31/01/17	mar 07/02/17
41	1.2.4.2.1	Codificar interfaz	2 días	mar 31/01/17	jue 02/02/17
42	1.2.4.2.2	Clases de Usuarios	2 días	jue 02/02/17	lun 06/02/17
43	1.2.4.2.3	Realizar conexión a la base de datos	1 hora	lun 06/02/17	lun 06/02/17
44	1.2.4.2.4	Alta de Usuario	1 hora	lun 06/02/17	lun 06/02/17
45	1.2.4.2.5	Baja de Usuario	1 hora	lun 06/02/17	lun 06/02/17
46	1.2.4.2.6	Actualización de Usuario	2 horas	lun 06/02/17	lun 06/02/17
47	1.2.4.2.7	Consultas de Usuario	2 horas	lun 06/02/17	mar 07/02/17
48	1.2.5	▷ Pruebas	2 días	mar 07/02/17	jue 09/02/17
49	1.2.5.1	▷ Realizar pruebas de caja blanca	2 días	mar 07/02/17	jue 09/02/17
50	1.2.5.1.1	▷ Módulo de Usuarios	2 días	mar 07/02/17	jue 09/02/17
51	1.2.5.1.1.1	Pruebas unitarias	1 día	mar 07/02/17	mié 08/02/17
52	1.2.5.1.1.2	Pruebas de modulo	1 día	mié 08/02/17	jue 09/02/17
53	1.3	▷ Fase/Iteración 2	11.15 días	jue 09/02/17	jue 23/02/17
95	1.4	▷ Implementación	1.63 días	jue 23/02/17	vie 24/02/17
101	1.5	▷ Cierre de proyecto	1.38 días	vie 24/02/17	lun 27/02/17
109					

Imagen 45. Diagrama de Gantt 3.

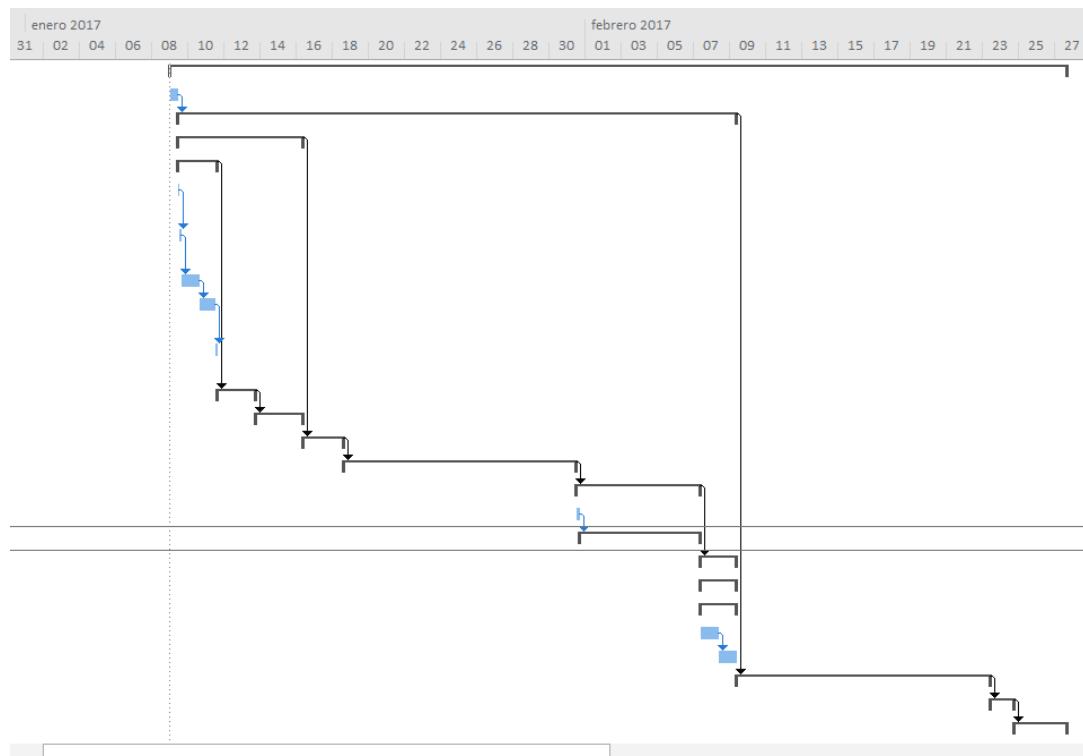


Imagen 46. Línea temporal de Diagrama de Gantt.

7.2. MATERIAL RECOPILADO Y REQUERIMIENTOS

7.2.1. Entrevista

Entrevista

Entrevistador: Aldo Omar Guajardo Chávez Entrevistado: Marco Antonio Trejo Gómez

Cargo: Desarrollo Institucional

Empresa: Registró Público de la Propiedad y del Comercio.

Lugar: Querétaro, Querétaro. Fecha: 07/02/2017

El objetivo de esta entrevista, es la recolección de información que servirá para el diseño del sistema.

1. ¿Cuál es el objetivo de la aplicación web?
|
2. ¿Qué tipo de información se manejará en la aplicación?
3. ¿Todas las personas acceden al mismo tipo de información?
4. ¿Piensa quienes pueden llegar a hacer uso del sistema?
5. ¿Cómo se pretende administrar la aplicación?
6. ¿Qué se le piden al administrador?
7. ¿Qué datos se le piden al usuario?
8. ¿Qué datos se le piden al analista?

Imagen 47. Entrevista.

7.2.2. Especificación de Requerimientos de Software

Base de Datos de Requerimientos								
Id	Nombre	Descripción	Stakeholders	Recepción y Evaluación			Seguimiento	
				Aceptado	Razones de Rechazo	Validado	Valoración del	Estatus
RC_001	Tabla de usuarios	La aplicación web proporcionará una cuadricula de líneas horizontales y verticales que proporcionen información de cada usuario registrado en el sistema.	Cliente, Usuario final	Sí		Sí	Alta	En proceso
RC_002	Tabla de documentos	La aplicación web mostrará los documentos dados de alta en el sistema.	Cliente, Usuario final	Sí		Sí	Alta	En proceso
RF_001	Login	La aplicación web brinda acceso a un usuario registrado.	Cliente	Sí		Sí	Alta	En proceso
RF_002	Baja de usuario	La aplicación web permite eliminar un usuario registrado del sistema.	Cliente	Sí		Sí	Alta	En proceso
RF_003	Alta de usuario	La aplicación web permite registrar un nuevo usuario en el sistema.	Cliente	Sí		Sí	Alta	En proceso
RF_004	Actualizar datos de usuario	La aplicación web permite actualizar datos de un usuario registrado en el sistema.	Cliente	Sí		Sí	Alta	En proceso
RF_005	Alta de documento	La aplicación web permite registrar y subir un nuevo documento en el sistema.	Cliente	Sí		Sí	Alta	En proceso
RF_006	Baja de documento	La aplicación web permite eliminar un documento registrado en el sistema.	Cliente	Sí		Sí	Alta	En proceso
RF_007	Actualizar datos de documento	La aplicación web permite actualizar los datos de un documento registrado en el sistema.	Cliente	Sí		Sí	Alta	En proceso
RNF_001	Usabilidad	La navegación deberá ser fácil en la aplicación web.	Cliente, Usuario final	Sí		Sí	Alta	En proceso
RNF_002	Usabilidad	La descarga de documento deberá ser de fácil en la aplicación web.	Cliente, Usuario final	Sí		Sí	Alta	En proceso
RNF_003	Fiabilidad	El sistema contendrá información verídica y actualizada.	Cliente	Sí		Sí	Media	En proceso
RNF_004	Responsabilidad	La aplicación será responsive, ajustándose a la resolución del monitor o navegador web.	Cliente	Sí		Sí	Media	En proceso
RNF_005	Tiempo de carga	El tiempo de carga del sitio estará entre 5 y 8 segundos.	Cliente	Sí		Sí	Media	En proceso
RCP_0001	Conexión	La aplicación web contendrá con una conexión con una base de datos.	Cliente	Sí		Sí	Alta	En proceso
RCP_0002	Bootstrap	La aplicación web se ejecutara con bootstrap.	Cliente	Sí		Sí	Alta	En proceso
RCP_0003	Programación	La aplicación web estará en lenguaje de programación PHP y HTML.	Cliente	Sí		Sí	Alta	En proceso

Imagen 48. Base de Datos de Requerimientos.

7.3. ANÁLISIS DE LA INFORMACIÓN

7.3.1. Casos de uso y secuencia

Tabla 1. CU_001 y DS_001

Diagrama del caso de uso	
*Id	CU_001
*Nombre	Log in.
*Req funcional	RF_001


```

sequenceDiagram
    actor Admin as Administrador
    actor Syst as Sistema
    participant System as System
    Admin->>System: Acceder a la aplicacion web.
    activate System
    Note over System: Muestra la pagina de login.
    deactivate System
    Admin->>System: Llena los campos "No.Empleado" y "Clave".
    activate System
    Note over System: Oprime el boton "Entrar" para inicia sesion.
    deactivate System
    Note over System: Validacion de formulario, no deja campos vacios.
    Note over System: Verifica los datos "No.Empleado" y "Clave".
    Note over System: Muestra pagina de inicio de administrador.
    Note over System: Oprime el enlace de "Cerrar sesion".
    Note over System: Cierra la sesion del administrador.
    Admin-->>Syst: 
  
```

The diagram illustrates the flow of the 'Log in.' use case. It starts with the 'Administrador' actor sending a message to the 'System' to 'Acceder a la aplicacion web.' The system then 'Muestra la pagina de login.' The administrator then 'Llena los campos "No.Empleado" y "Clave."', followed by pressing the 'Entrar' button to start the session. The system performs 'Validación de formulario, no deja campos vacios.' and 'Verifica los datos "No.Empleado" y "Clave.". It then 'Muestra pagina de inicio de administrador.' The administrator then 'Oprime el enlace de "Cerrar sesion". Finally, the system 'Cierra la sesion del administrador.'

Diagrama de secuencia	
*Id	DS_001
*Nombre	Log in.
*Req funcional	RF_001

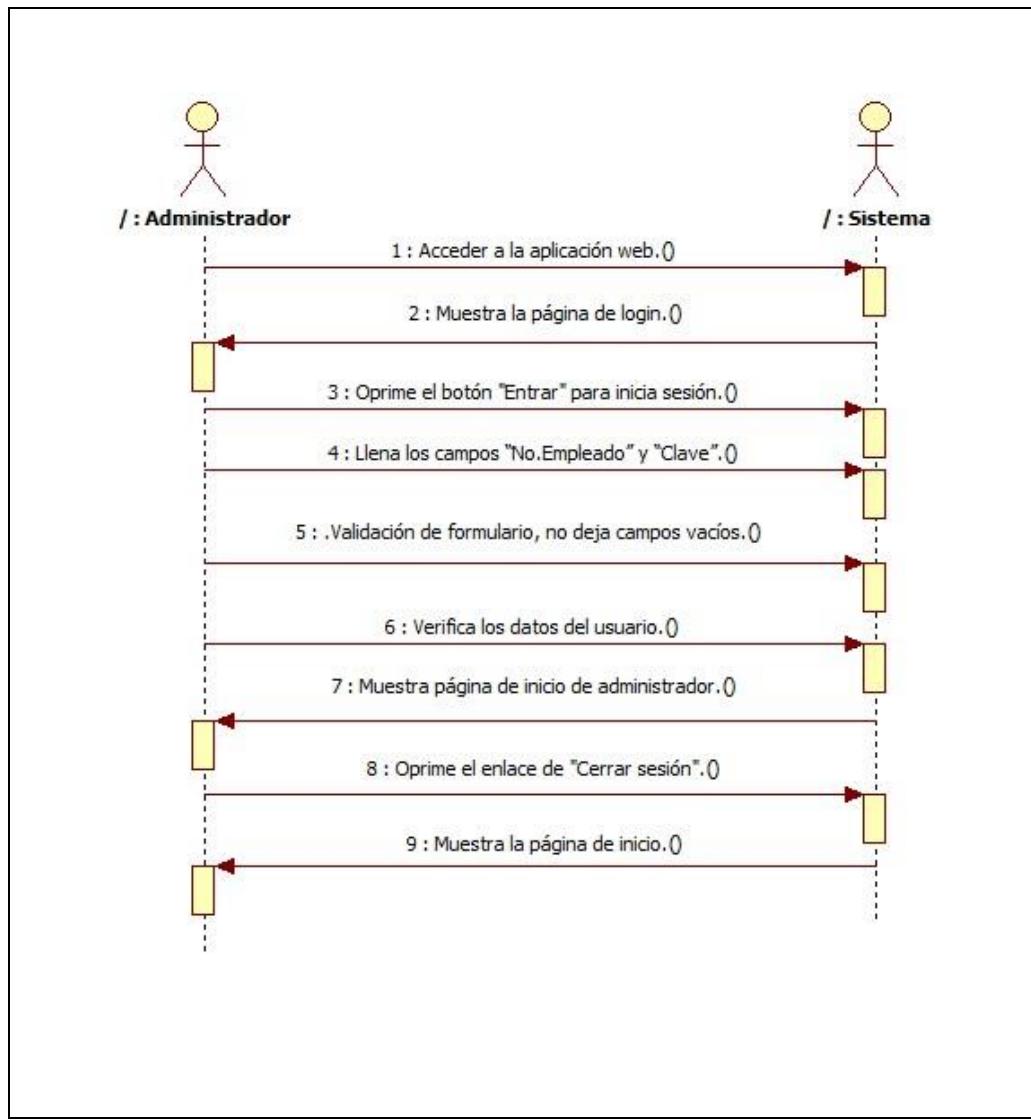


Tabla 2. Descripción CU_001.

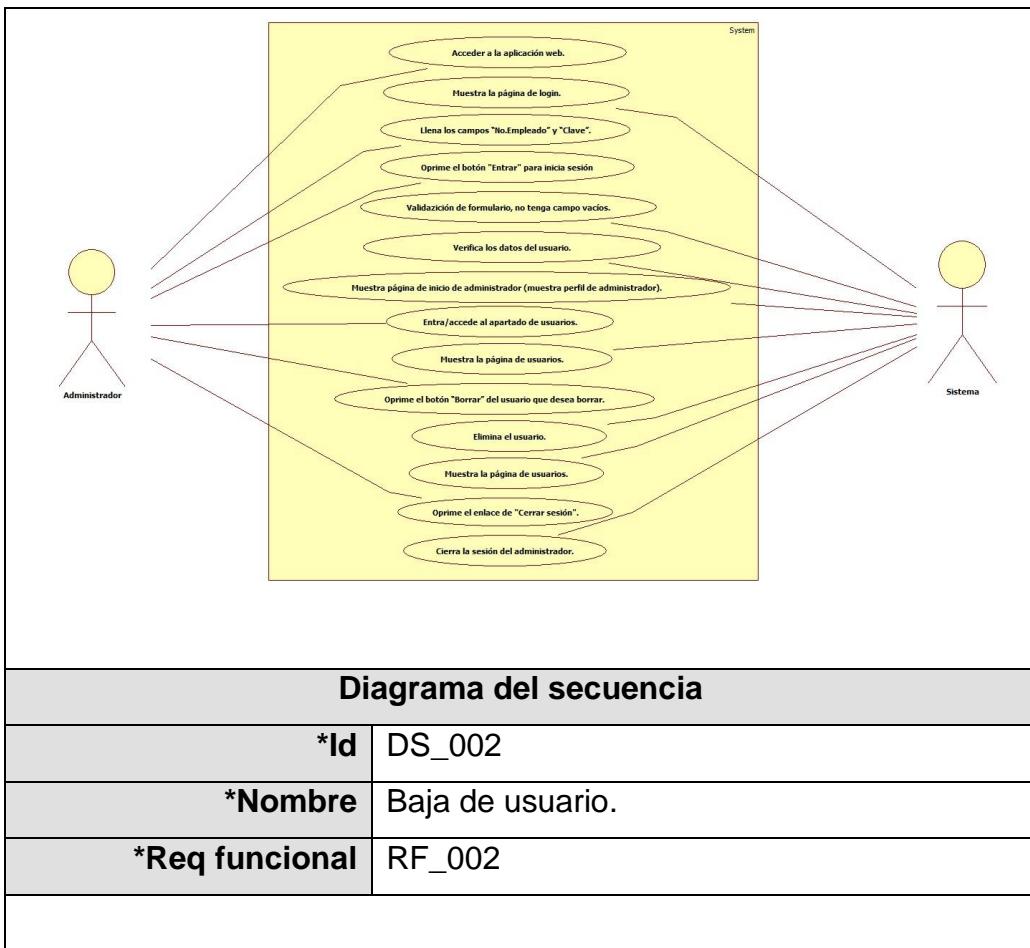
Descripción	
*Id	CU_001
*Nombre	Log in.
*Req. Funcional	RF_001
*Responsable	AOGC

Actualizado por	AOGC		
*Fecha de creación	19/02/2017		
Fecha de última actualización:	19/02/2016		
*Actores	1. Administrador 2. Sistema		
*Descripción	La aplicación web brinda acceso a un usuario registrado.		
*Evento activador	Oprimir el botón “Entrar”.		
*Precondiciones	Que el usuario este registrado.		
*Poscondiciones	El sistema muestra la página de inicio de administrador (muestra perfil de administrador).		
*Flujo normal	Paso	Actor	Acción
	1	Administrador	Acceder a la aplicación web.
	2	Sistema	Muestra la página de inicio.
	3	Administrador	Inicia sesión en la aplicación web.
	4	Administrador	Llena todos los campos “usuario” y “contraseña”.

	5	Sistema	Validación de formulario, no deja campos vacíos.
	6	Sistema	Verifica los datos del usuario y contraseña.
	7	Sistema	Muestra página de inicio de administrador (muestra perfil de administrador).
	8	Administrador	Cierra sesión.
	9	Sistema	Cierra la sesión del administrador.
Flujos alternativos	Pa so	Actor	Acción alternativa
	7	Sistema	Muestra la página de inicio.
	6	Sistema	Muestra la página de inicio.
Incluye			
*Frecuencia de Uso	Casi siempre		
Notas y Asuntos			

Tabla 3. CU_002 y DS_002.

Diagrama del caso de uso	
*Id	CU_002
*Nombre	Baja de usuario.
*Req funcional	RF_002



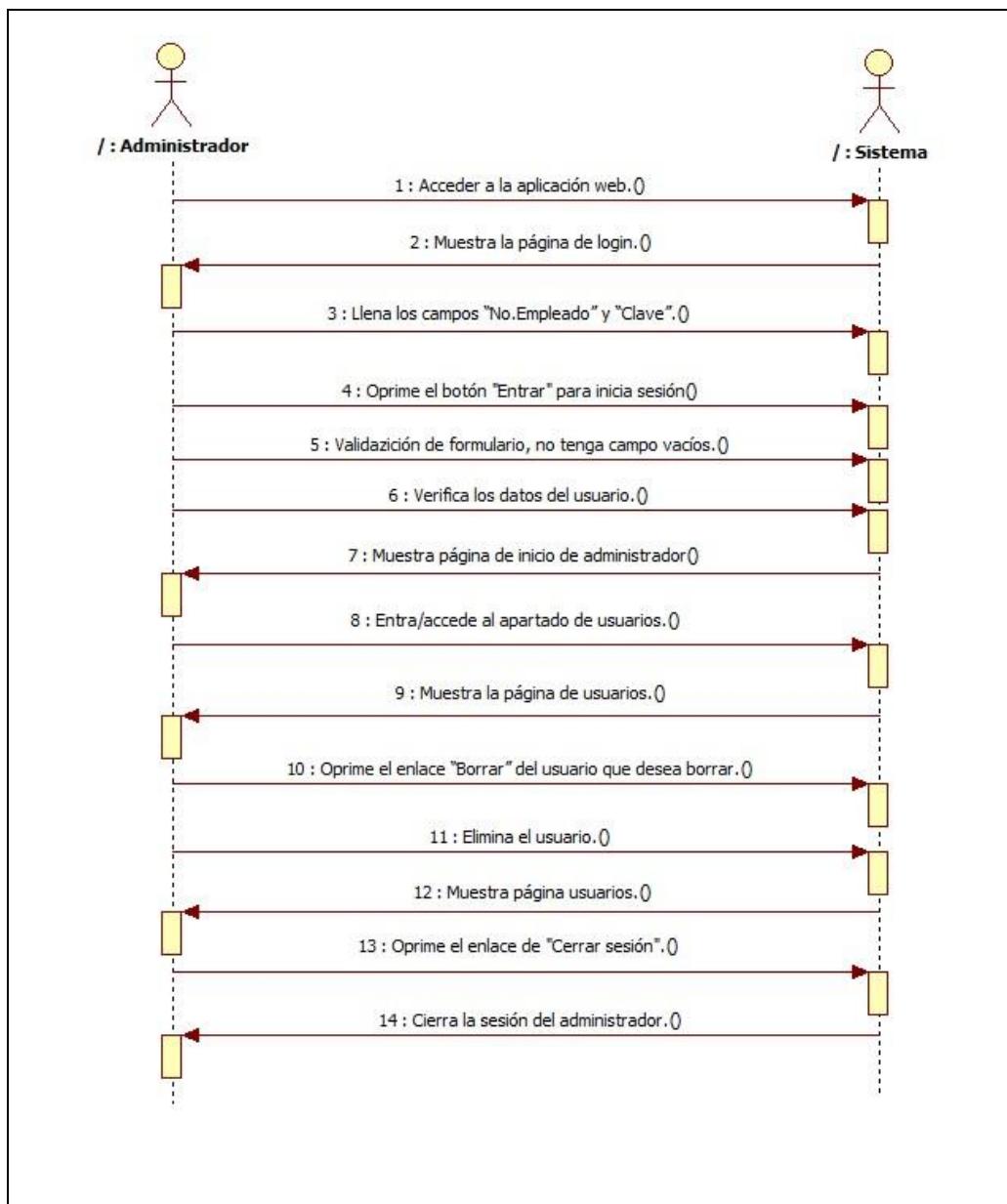


Tabla 4. CU_002 y DS_002.

Descripción	
*Id	CU_002
*Nombre	Baja de usuario.

*Req. Funcional	RF_002		
*Responsable	AOGC		
Actualizado por	AOGC		
*Fecha de creación	19/02/2017		
Fecha de última actualización:	19/02/2017		
*Actores	1. Administrador 2. Sistema		
*Descripción	La aplicación web permite eliminar un usuario registrado del sistema.		
*Evento activador	Oprimir el enlace “Borrar”.		
*Precondiciones	Que el usuario este registrado, identificar al usuario a borrar, tener sesión abierta en el sistema, contar con los privilegios necesarios.		
*Poscondiciones	El sistema muestra los usuarios.		
*Flujo normal	Paso	Actor	Acción

	1	Sistema	Muestra página de inicio de administrador (muestra perfil de administrador).
	2	Administrador	Entra/accede al apartado de usuarios.
	3	Sistema	Muestra la página de usuarios.
	4	Administrador	Oprime el enlace “Borrar” del usuario que desea borra.
	5	Sistema	Elimina el usuario.
	6	Sistema	Muestra la página de usuarios.
	7	Administrador	Oprime el enlace de Cerrar sesión.
	8	Sistema	Cierra la sesión del administrador.
Flujos alternativos	Paso	Actor	Acción alternativa
Incluye	RF_001 Log in.		
*Frecuencia de Uso	Casi siempre		
Notas y Asuntos			

Tabla 5. CU_003 y DS_003.

Diagrama del caso de uso	
*Id	CU_003
*Nombre	Alta de usuario.
*Req funcional	RF_003
<p>Acceder a la aplicación web. Hacerse la página de login. Llene los campos "NoEmpleado" y "Clave". Oprime el botón "Entrar" para iniciar sesión. Validación de formulario, no tenga campos vacíos. Verifica los datos del usuario. Hacerse página de administrador. Entra/escala al apartado de usuarios. Hacerse la página de usuarios. Oprime el enlace "Nuevo" para agregar nuevo usuario. Hacerse el formulario con los campos de Empleado, Proceso, Area, Puesto, Jefe, Clave, Status, Privilegio, IC, SATL, FE, UNITE. Hacerse el botón "Agregar". Llene todos los campos con la información del usuario. Oprime el botón "Agregar". Validación de formulario, no deje campos vacíos. Valida los datos automáticamente y visualmente. Registra los datos en el sistema. El sistema muestra los usuarios. Oprime el enlace de "Cerrar sesión". Cierre la sesión del administrador.</p>	
Diagrama del secuencia	
*Id	DS_003
*Nombre	Alta de usuario.
*Req funcional	RF_003

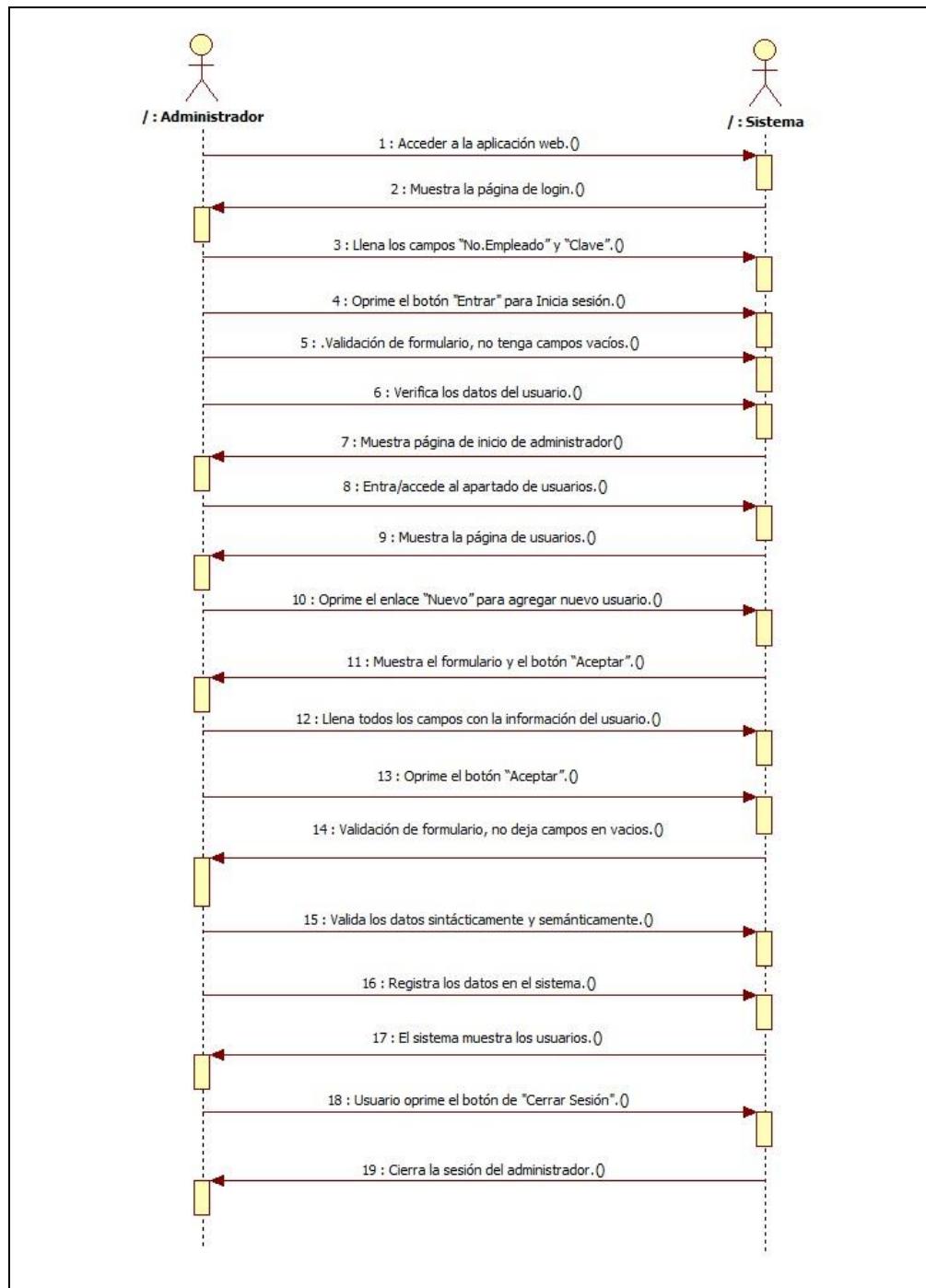


Tabla 6. Descripción de CU_003.

Descripción			
*Id	CU_003		
*Nombre	Alta de usuario.		
*Req. Funcional	RF_003		
*Responsable	AOGC		
Actualizado por	AOGC		
*Fecha de creación	19/02/2017		
Fecha de última actualización:	19/02/2017		
*Actores	1. Administrador 2. Sistema		
*Descripción	La aplicación web permite registrar un nuevo usuario en el sistema.		
*Evento activador	Oprimir el enlace “Nuevo”.		
*Precondiciones	Que el usuario este registrado, tener sesión abierta en el sistema, contar con los privilegios necesarios.		
*Poscondiciones	El sistema muestra los usuarios.		
*Flujo normal	Paso	Actor	Acción
	1	Sistema	Muestra página de inicio de administrador (muestra perfil de administrador).

Flujos	Paso	Actor	Acción alternativa
	2	Administrador	Entra/accede al apartado de usuarios.
	3	Sistema	Muestra la página de usuarios.
	4	Administrador	Oprime el enlace “Nuevo” para agregar nuevo usuario.
	5	Sistema	Muestra el formulario con los campos para llenar.
	6	Administrador	Llena todos los campos con la información del usuario seleccionado.
	7	Administrador	Oprime el botón “Aceptar”.
	8	Sistema	Validación de formulario, no deja campos vacíos.
	9	Sistema	Valida los datos sintácticamente y semánticamente.
	10	Sistema	Registra los datos en el sistema.
	11	Sistema	El sistema muestra los usuarios.
	12	Administrador	Cierra sesión.
	13	Sistema	Cierra la sesión del administrador.

alternativos			
Incluye	RF_001 Log in.		
*Frecuencia de Uso	Casi siempre		
Notas y Asuntos			

Tabla 7. CU_004 y DS_004.

Diagrama del caso de uso	
*Id	CU_004
*Nombre	Actualizar datos de usuario
*Req funcional	RF_004
Diagrama del secuencia	
*Id	DS_004
*Nombre	Actualizar datos de usuario
*Req funcional	RF_004

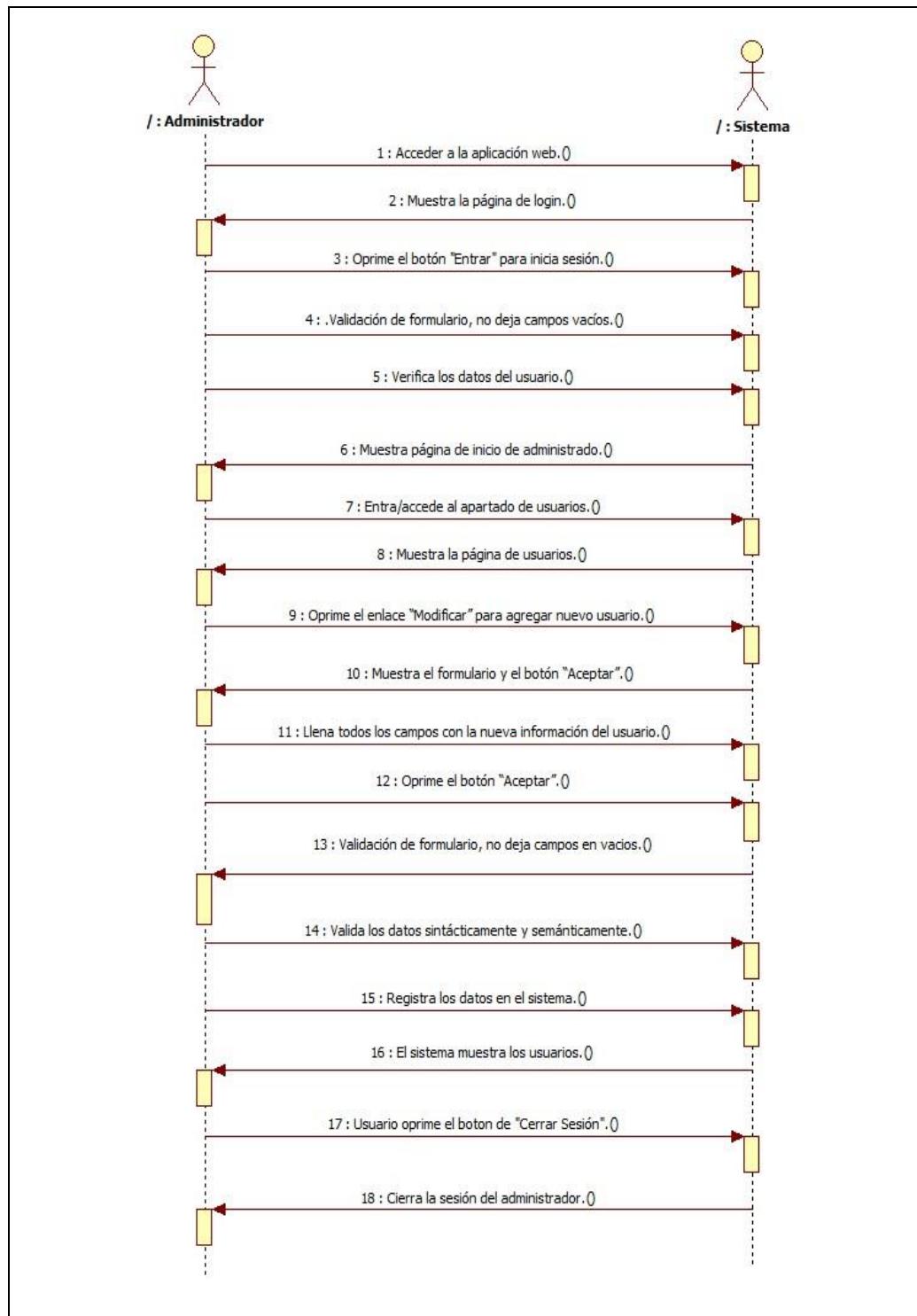


Tabla 8. Descripción CU_004.

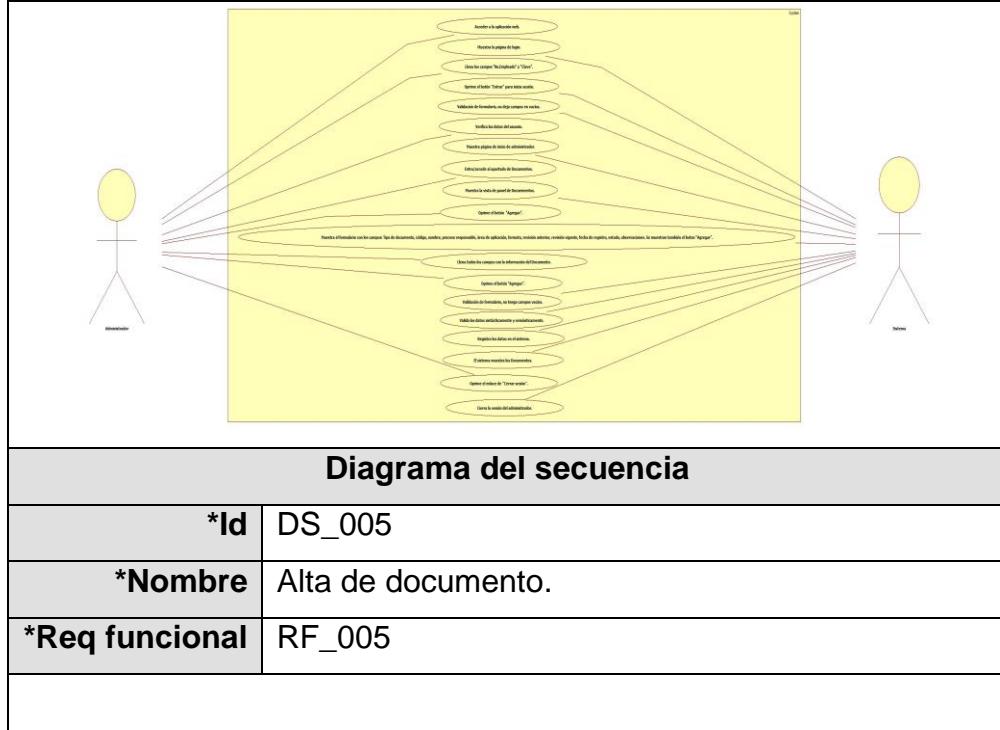
Descripción			
*Id	CU_004		
*Nombre	Actualizar datos de usuario.		
*Req. Funcional	RF_004		
*Responsable	AOGC		
Actualizado por	AOGC		
*Fecha de creación	19/02/2017		
Fecha de última actualización:	19/02/2017		
*Actores	1. Administrador 2. Sistema		
*Descripción	La aplicación web permite actualizar datos de un usuario registrado en el sistema.		
*Evento activador	Oprimir el enlace “Modificar”.		
*Precondiciones	1. Que el usuario este registrado, tener la nueva información, identificar al usuario a modificar, tener sesión abierta en el sistema, contar con los privilegios necesarios.		
*Poscondiciones	1. El sistema muestra los datos actualizados del usuario.		
*Flujo normal	Paso	Actor	Acción
	1	Sistema	Muestra el panel del administrador (Muestra perfil de administrador).
	2	Administrador	Entra/accede al apartado de usuarios.

	3	Sistema	Muestra la vista de panel de usuarios.
	4	Administrador	Oprimir el enlace “Modificar” del usuario que deseamos hacer los cambios.
	5	Sistema	Muestra el formulario con los datos del usuario precargados en los campos: nombre, usuario, contraseña, privilegios. Se muestran también los botones “Aceptar”.
	6	Administrador	Llena todos los campos con la nueva información del usuario seleccionado.
	7	Administrador	Oprime el botón “Aceptar”.
	8	Sistema	Validación de formulario, no deja campos vacíos.
	9	Sistema	Valida los datos sintácticamente y semánticamente.

	10	Sistema	Registra los datos en el sistema.
	11	Sistema	Muestra los usuarios y cierra el formulario.
	12	Administrador	Cierra sesión.
	13	Sistema	Cierra la sesión del administrador.
	Paso	Actor	Acción alternativa
Flujos alternativos			
Incluye	RF_001 Log in.		
*Frecuencia de Uso	Casi siempre		
Notas y Asuntos			

Tabla 9. CU_005 y DS_005.

Diagrama del caso de uso	
*Id	CU_005
*Nombre	Alta de documento.
*Req funcional	RF_005



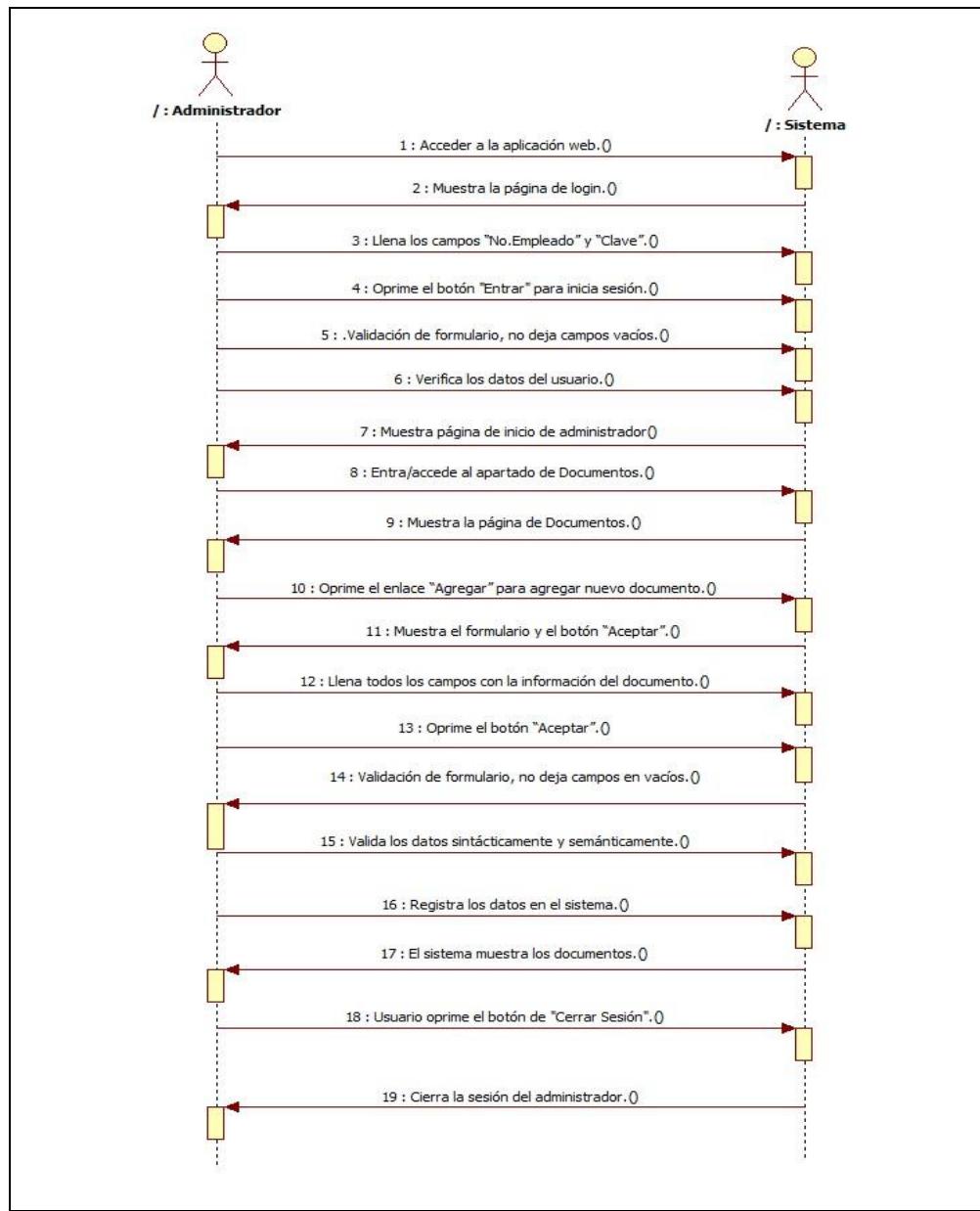


Tabla 10. Descripción CU_005.

Descripción	
*Id	CU_005
*Nombre	Alta de documento

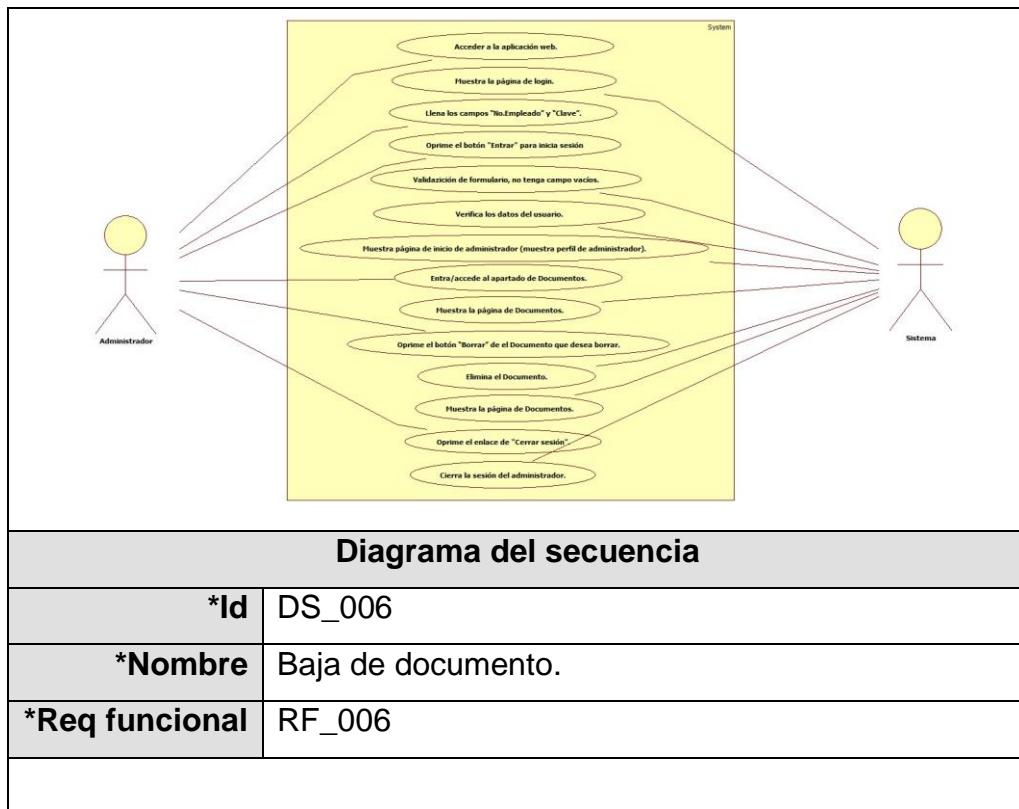
*Req. Funcional	RF_005		
*Responsable	AOGC		
Actualizado por	AOGC		
*Fecha de creación	19/02/2016		
Fecha de última actualización:	19/02/2016		
*Actores	1. Administrador 2. Sistema		
*Descripción	La aplicación web permite registrar un nuevo documento en el sistema.		
*Evento activador	Oprimir el enlace “Nuevo”.		
*Precondiciones	1. Que el usuario este registrado, tener información del documento, tener sesión abierta en el sistema, contar con los privilegios necesarios.		
*Poscondiciones	2. El sistema muestra los datos del documento.		
*Flujo	Paso	Actor	Acción

normal	1	Sistema	Muestra el panel del administrador (Muestra perfil de administrador).
	2	Administrador	Entra/accede al apartado de “Documentos”.
	3	Sistema	Muestra la vista de panel de “Documentos”.
	4	Administrador	Oprimir el enlace “Nuevo”.
	5	Sistema	Muestra el formulario con los campos: Tipo de documento, código, nombre, proceso responsable, área de aplicación, formato, revisión anterior, revisión vigente, fecha de registro, estado, observaciones. Se muestran también los botones “Aceptar”.
	6	Administrador	Llena todos los campos con la información del documento.
	7	Administrador	Oprime el botón “Aceptar”.
	8	Sistema	Validación de formulario, no deja campos vacíos.
	9	Sistema	Valida los datos sintácticamente y semánticamente.

	10	Sistema	Registra los datos en el sistema.
	11	Sistema	Cierra el formulario y muestra los documentos.
	12	Administrador	Cierra sesión.
	13	Sistema	Cierra la sesión del administrador.
	Paso	Actor	Acción alternativa
Flujos alternativos			
Incluye	RF_001 Log in.		
*Frecuencia de Uso	Casi siempre		
Notas y Asuntos			

Tabla 11. CU_006.

Diagrama del caso de uso	
*Id	CU_006
*Nombre	Baja de documento.
*Req funcional	RF_006



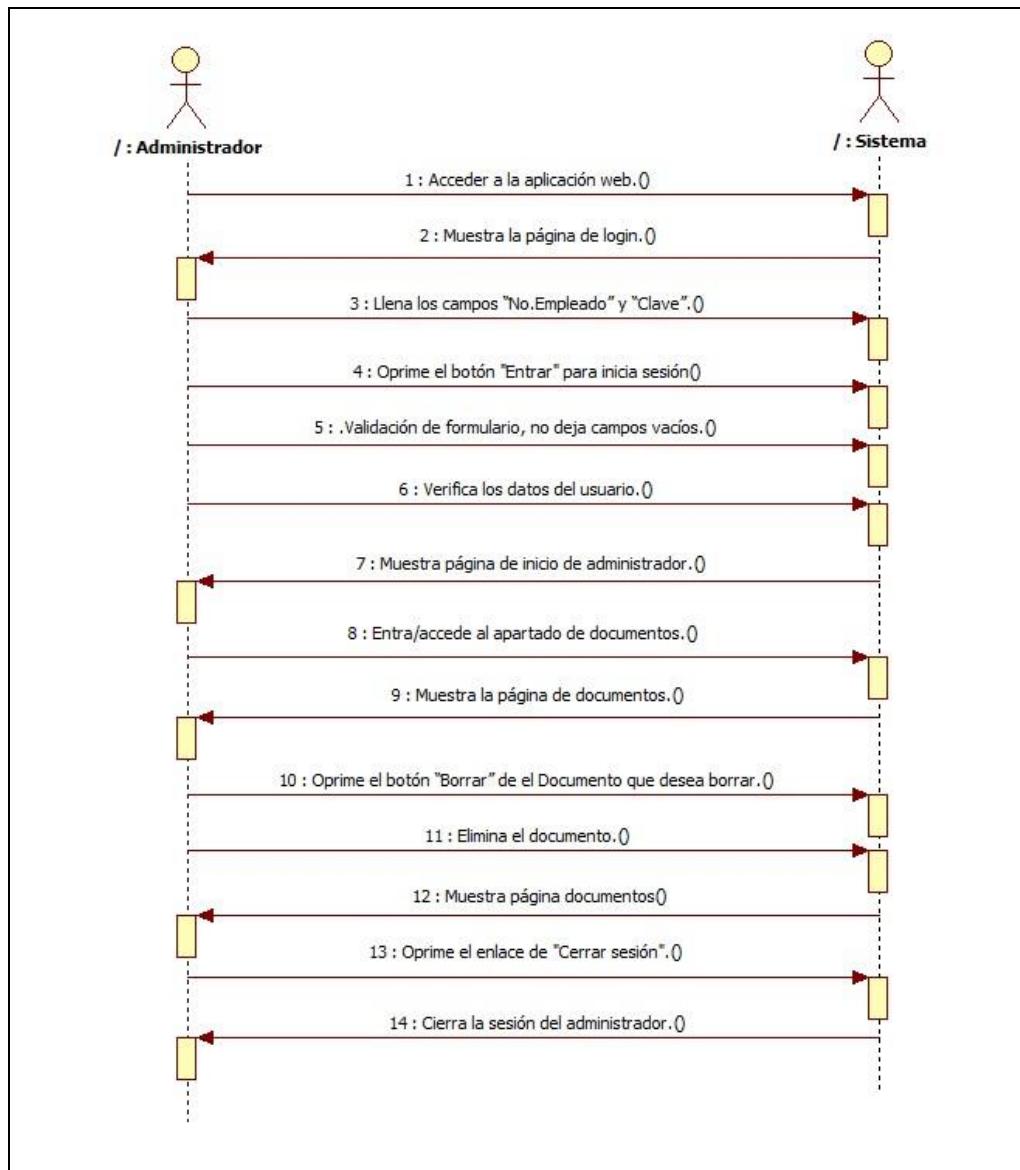


Tabla 12. Descripción CU_006.

Descripción	
*Id	CU_006
*Nombre	Baja de documentos.

*Req.	RF_006		
Funcional			
*Responsable	AOGC		
Actualizado por	AOGC		
*Fecha de creación	19/02/2017		
Fecha de última actualización:	19/02/2017		
*Actores	1. Administrador 2. Sistema		
*Descripción	La aplicación web permite eliminar un documento registrado en el sistema.		
*Evento activador	Oprimir el enlace “Eliminar”.		
*Precondiciones	1. Que el usuario este registrado, tener documentos registrados, tener sesión abierta en el sistema, contar con los privilegios necesarios.		
*Poscondiciones	3. El sistema muestra los datos del documento.		
*Flujo normal	Paso	Actor	Acción
	1	Sistema	Muestra el panel del administrador (Muestra perfil de administrador).

	2	Administrador	Entra/accede al apartado de “Documentos”.
	3	Sistema	Muestra la vista de panel de “Documentos”.
	4	Administrador	Oprimir el enlace “Eliminar” del documento a eliminar.
	5	Sistema	Elimina el documento.
	6	Sistema	Muestra la página de Documentos.
	7	Administrador	Cierra sesión.
	8	Sistema	Cierra la sesión del administrador.
	Paso	Actor	Acción alternativa
Flujos alternativos			
Incluye	RF_001 Log in.		
*Frecuencia de Uso	Casi siempre		
Notas y Asuntos			

Tabla 13. CU_007 y DS_007.

Diagrama del caso de uso	
*Id	CU_007
*Nombre	Actualizar datos de documento.
*Req	RF_007

funcional	
Diagrama del secuencia	
*Id	DS_007
*Nombre	Actualizar datos de documento.
*Req funcional	RF_007

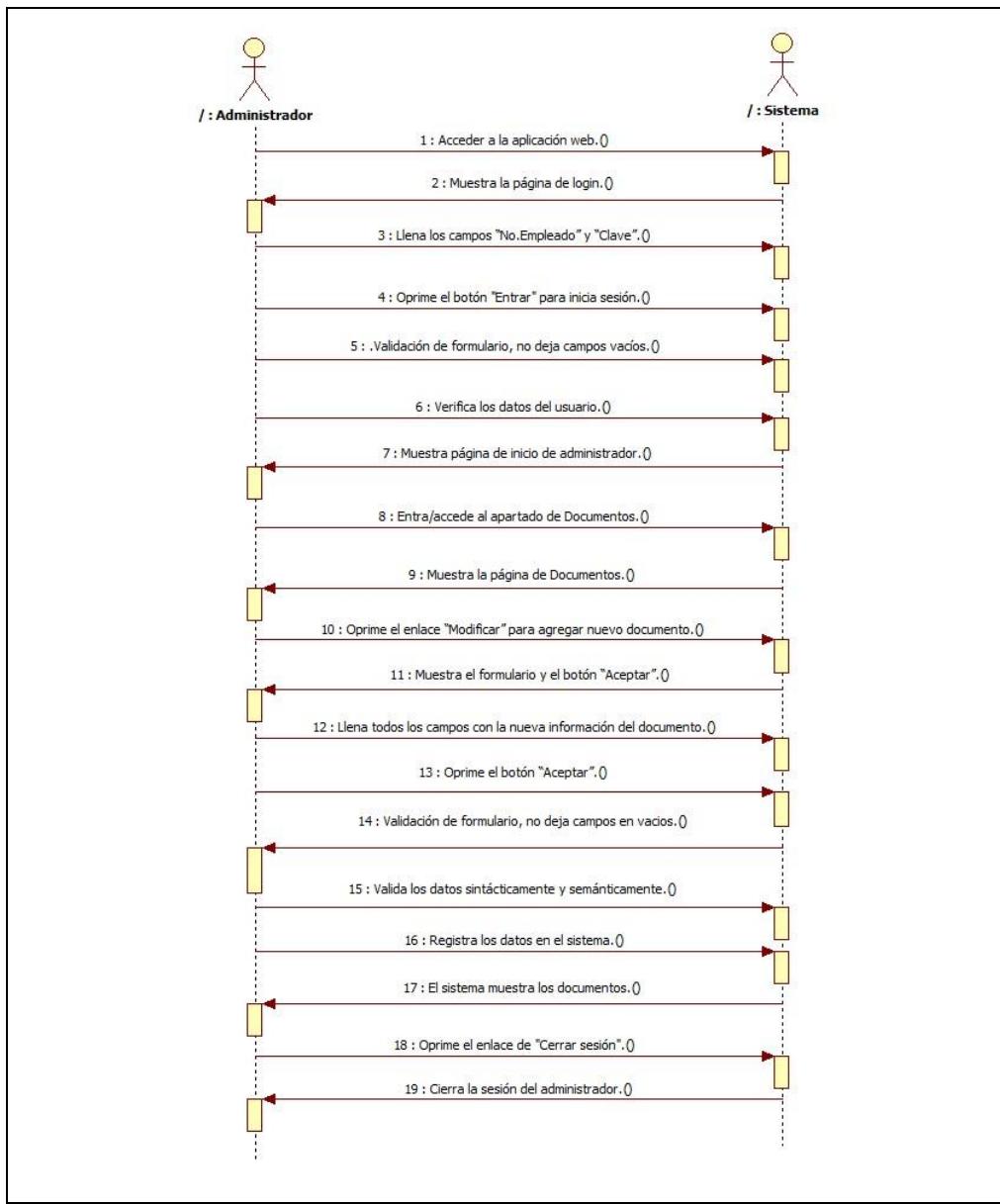


Tabla 14. Descripción CU_007.

Descripción	
*Id	CU_007
*Nombre	Actualizar datos de documentos.

*Req. Funcional	RF_007		
*Responsable	AOGC		
Actualizado por	AOGC		
*Fecha de creación	19/02/2017		
Fecha de última actualización :	19/02/2017		
*Actores	1. Administrador 2. Sistema		
*Descripción	La aplicación web permite actualizar los datos de un documento registrado en el sistema.		
*Evento activador	Oprimir el enlace “Modificar”.		
*Precondiciones	1. Que el usuario este registrado, tener producto registrado, tener información nueva de producto, tener sesión abierta en el sistema, contar con los privilegios necesarios.		
*Poscondiciones	4. El sistema muestra los datos del documento.		
*Flujo normal	Paso	Actor	Acción

	1	Sistema	Muestra el panel del administrador (Muestra perfil de administrador).
	2	Administrador	Entra/accede al apartado de “Documentos”.
	3	Sistema	Muestra la vista de panel de “Documentos”.
	4	Administrador	Oprimir el enlace “Modificar”.
	5	Sistema	Muestra el formulario con los datos del producto precargados en los campos: Tipo de documento, código, nombre, proceso responsable, área de aplicación, formato, revisión anterior, revisión vigente, fecha de registro, estado, observaciones. Se muestran también los botones “Aceptar”.
	6	Administrador	Llena todos los campos con la nueva información del documento seleccionado.
	7	Administrador	Oprime el botón “Aceptar”.
	8	Sistema	Validación de formulario, no deja campos vacíos.

	9	Sistema	Valida los datos sintácticamente y semánticamente.
	10	Sistema	Registra los datos en el sistema.
	11	Sistema	Cierra el formulario y muestra los documentos.
	12	Administrador	Cierra sesión.
	13	Sistema	Cierra la sesión del administrador.
	Paso	Actor	Acción alternativa
Flujos alternativos			
Incluye	RF_001 Log in.		
*Frecuencia de Uso	Casi siempre		
Notas y Asuntos			

7.3.2. Diagramas de clases

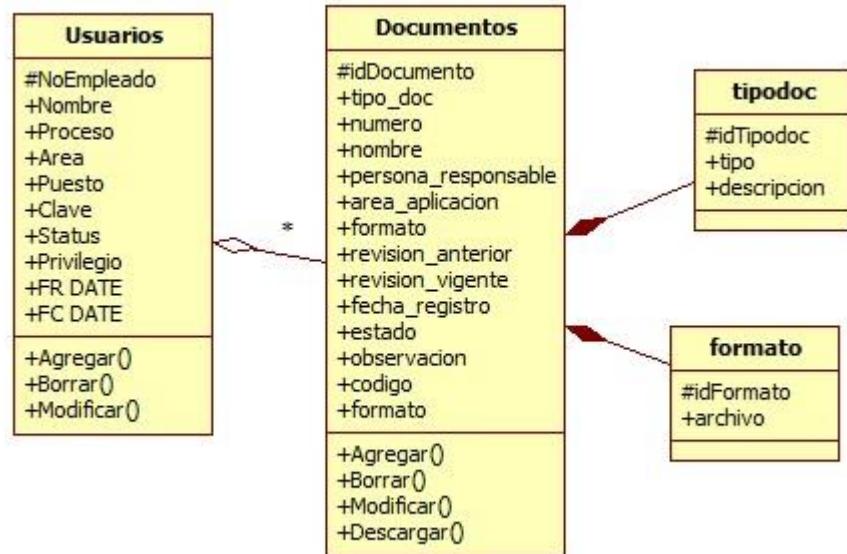


Imagen 49. Diagrama de clases.

7.4. BASE DE DATOS

7.4.1. Diagrama Entidad Relación

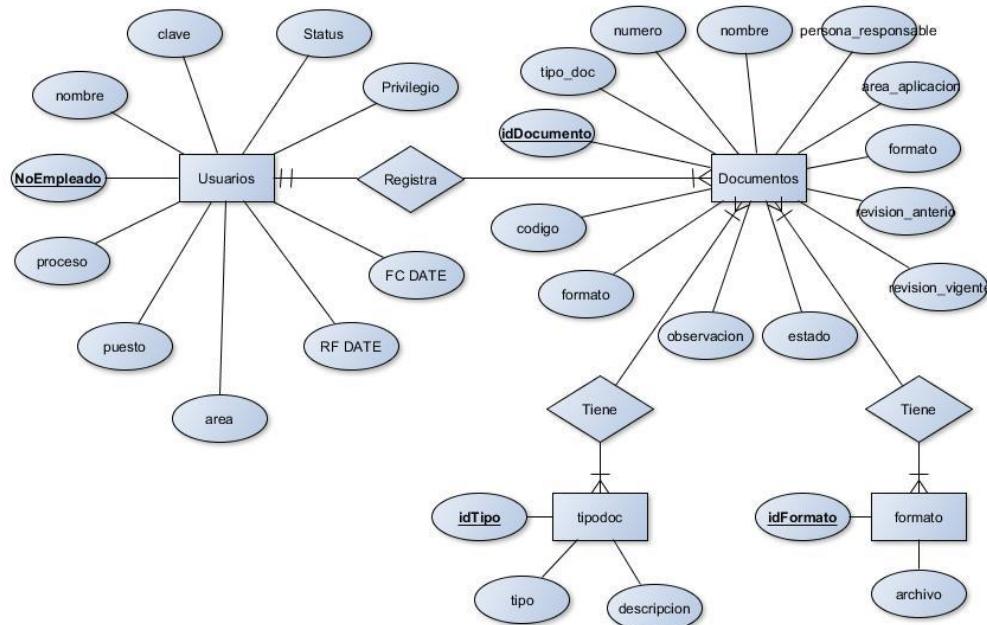


Figura 50. Diagrama Entidad Relación.

7.4.2. Diagrama Relacional

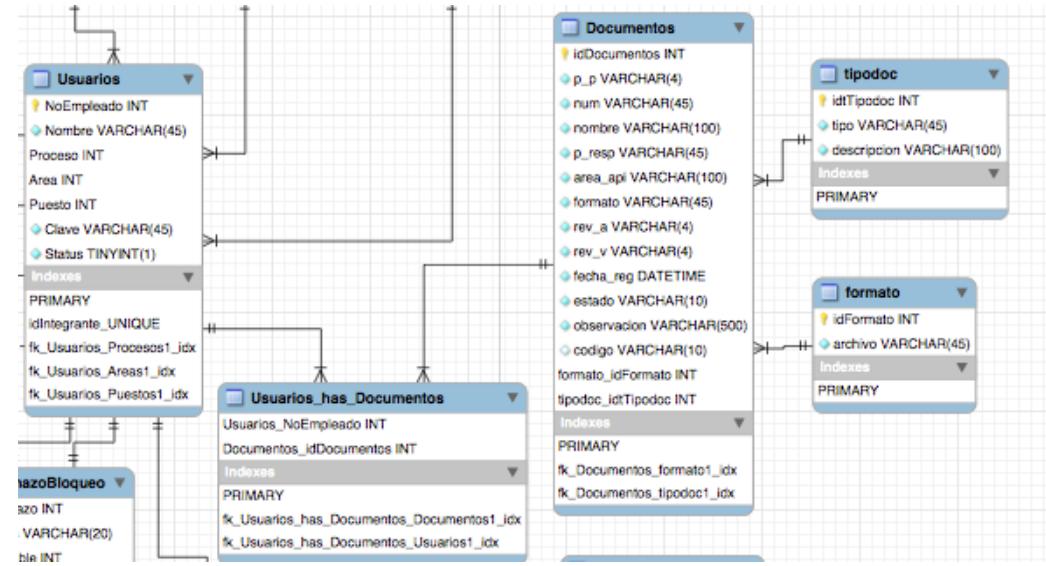


Imagen 51. Diagrama Relacional.

7.5. MAPA DE NAVEGACIÓN

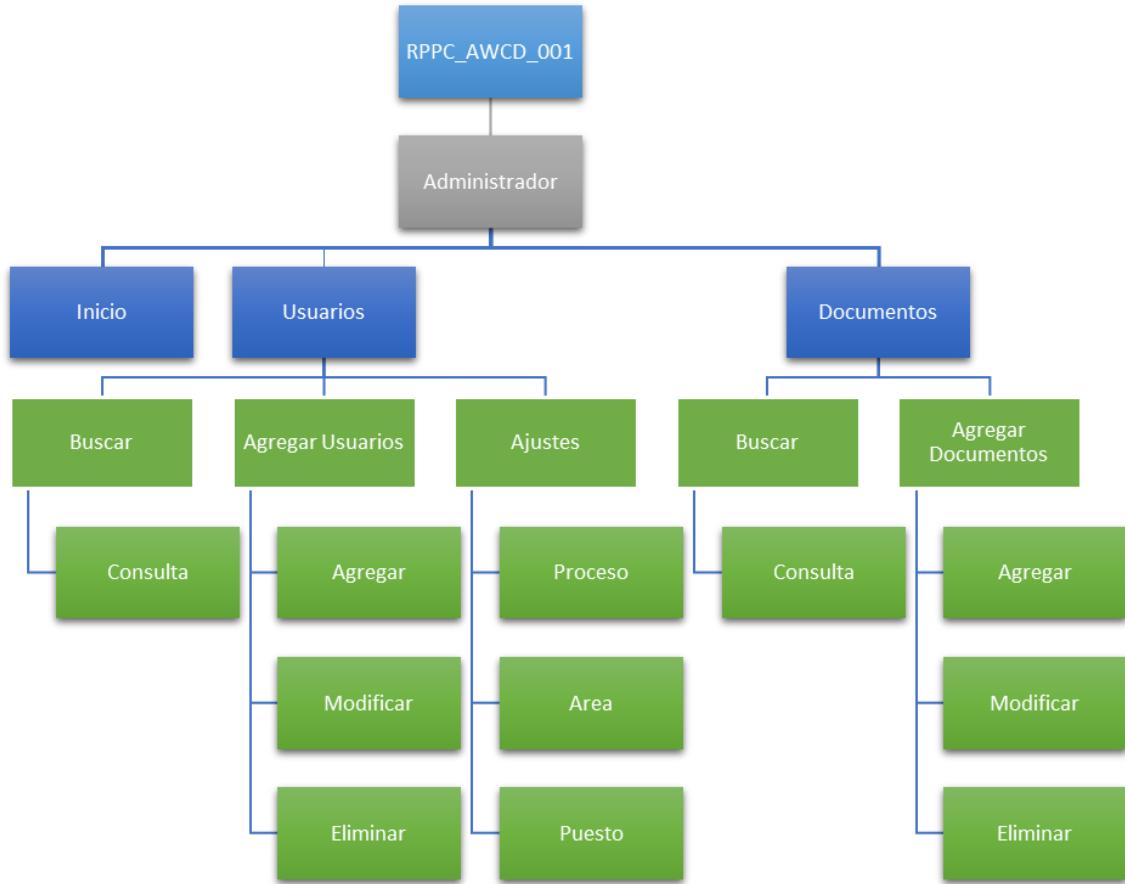


Figura 52. Mapa de navegación.

7.6. BOCETOS

7.6.1. Login

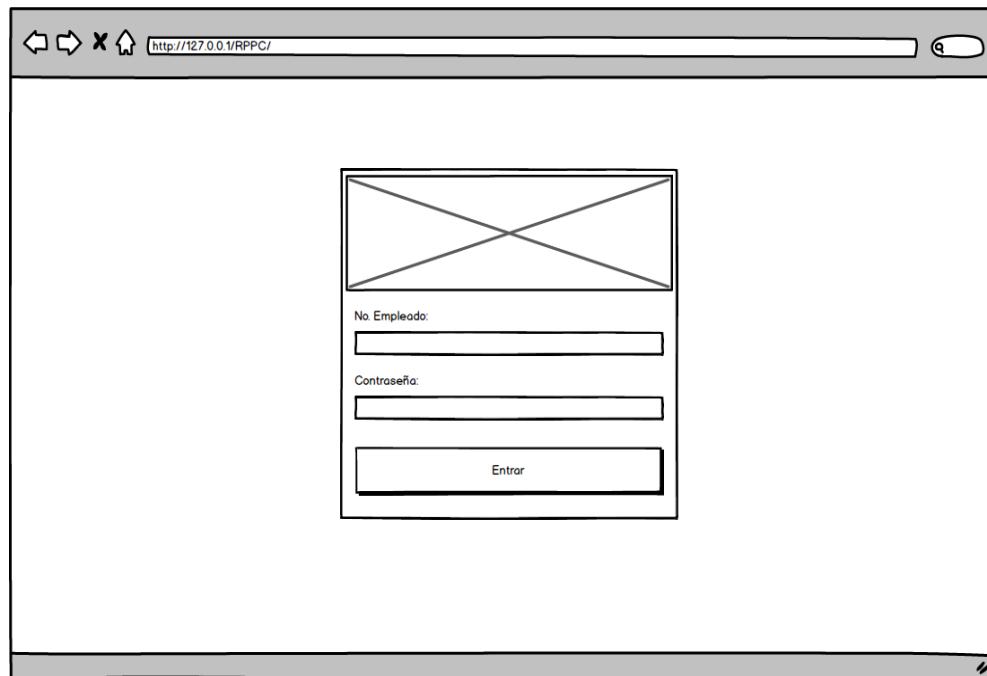


Imagen 53. Login.

7.6.2. Usuarios

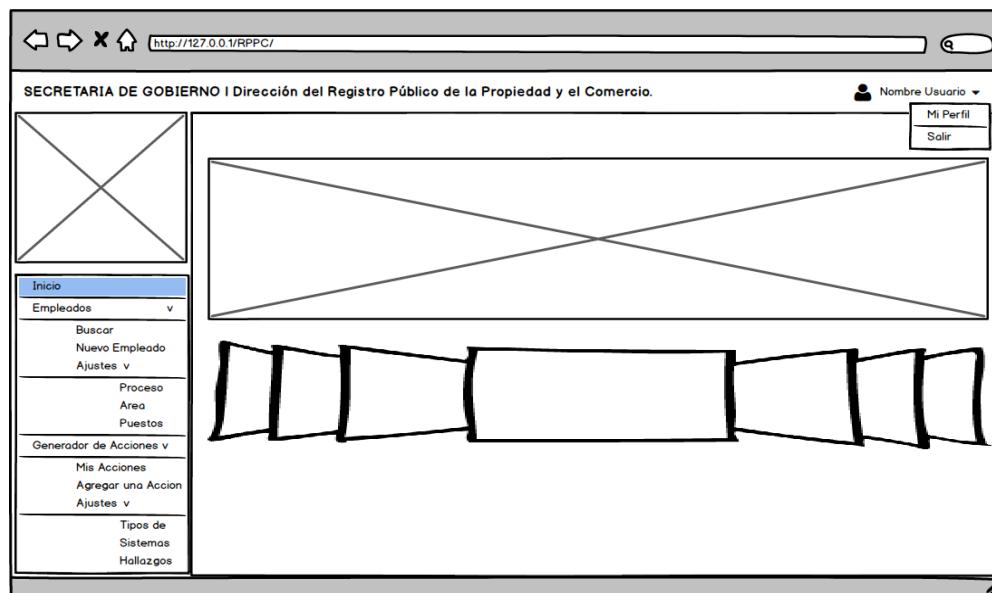


Imagen 54. Inicio.

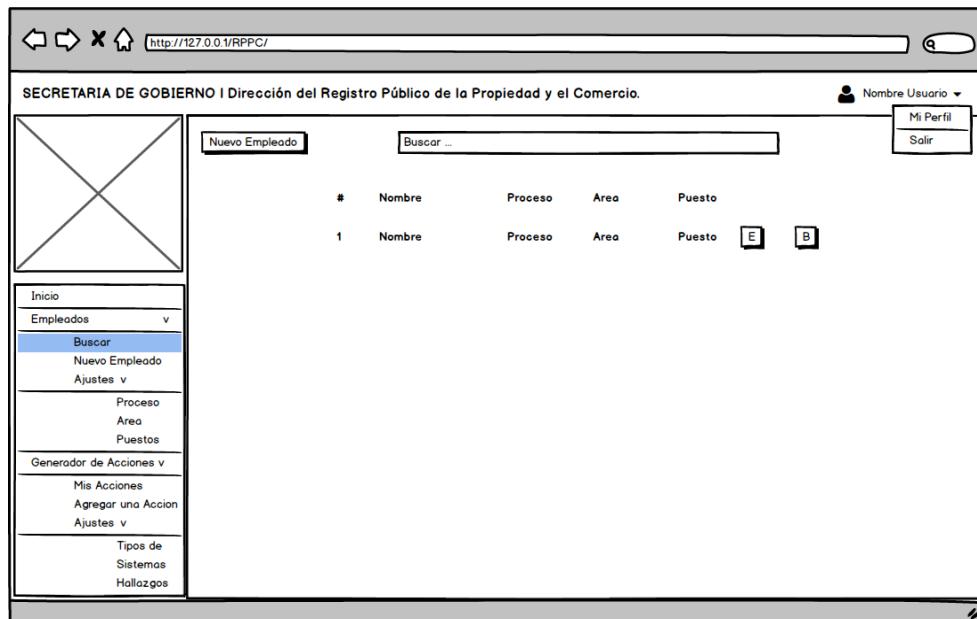


Imagen 55. Usuarios.

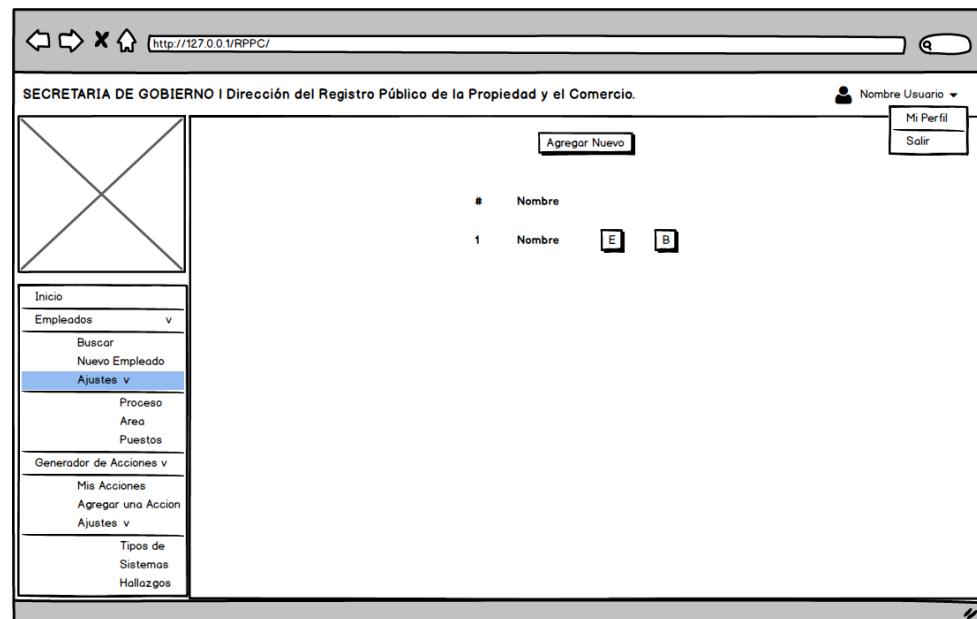


Imagen 56. Ajustes.

SECRETARIA DE GOBIERNO | Dirección del Registro Público de la Propiedad y el Comercio.

Nombre Usuario ▾

Mi Perfil
Salir

No. Empleado

Nombre

Proceso Selecciona...

Área Selecciona...

Puestos Selecciona...

Contraseña

Repetir Contraseña

Privilegios Selecciona...

Agregar

Imagen 57. Nuevo usuario.

SECRETARIA DE GOBIERNO | Dirección del Registro Público de la Propiedad y el Comercio.

Nombre Usuario ▾

Mi Perfil
Salir

Nombre

Guardar

Imagen 58. Agregar ajuste.

7.6.3. Documentos

The screenshot shows a web interface for managing documents. At the top right, there is a user profile icon with the text "Nombre Usuario" and dropdown arrows, along with "Mi Perfil" and "Salir" buttons. On the far right, there are two small square icons labeled "E" and "B". The main content area has a header "SECRETARIA DE GOBIERNO | Dirección del Registro Público de la Propiedad y el Comercio.". Below the header is a search bar with the placeholder "Buscar ...". To the right of the search bar is a "Nuevo Documento" button. A horizontal row of input fields is visible with labels: "# Nombre", "Tipo de documento", "Codigo", "Proceso Respon", "Área de aplicac", "Formato", "Revisión ante", "Revisión vige", "Fecha de regis", "Estado", and "Observaci". On the left side, there is a sidebar menu with the following items:

- Inicio
- Documentos
 - Buscar
 - Nuevo Documento
- Usuarios
 - Buscar
 - Nuevo Documento
 - Ajustes
- Proceso
 - Área
 - Puestos

Imagen 59. Documentos.

This screenshot shows the "Agregar Nuevo Documento" (Add New Document) form. At the top right, it features a user profile icon, "Nombre Usuario" with dropdown arrows, and "Mi Perfil" and "Salir" buttons. The main form area is titled "Agregar Nuevo Documento". It contains several input fields grouped into sections:

- Tipo de documento**: A dropdown menu.
- Nombre**: An input field.
- Código**: An input field.
- Proceso responsable**: A dropdown menu.
- Área de aplicación**: A dropdown menu.
- Formato**: An input field.
- Revisión Anterior**: An input field.
- Revisión Vigente**: An input field.
- Fecha de Registro**: An input field with a calendar icon.
- Estado**: A dropdown menu.
- Observaciones**: An input field.

At the bottom center of the form is a "Agregar" (Add) button.

Imagen 60. Agregar documento.

7.7. CODIFICACIÓN

7.7.1. Base de datos

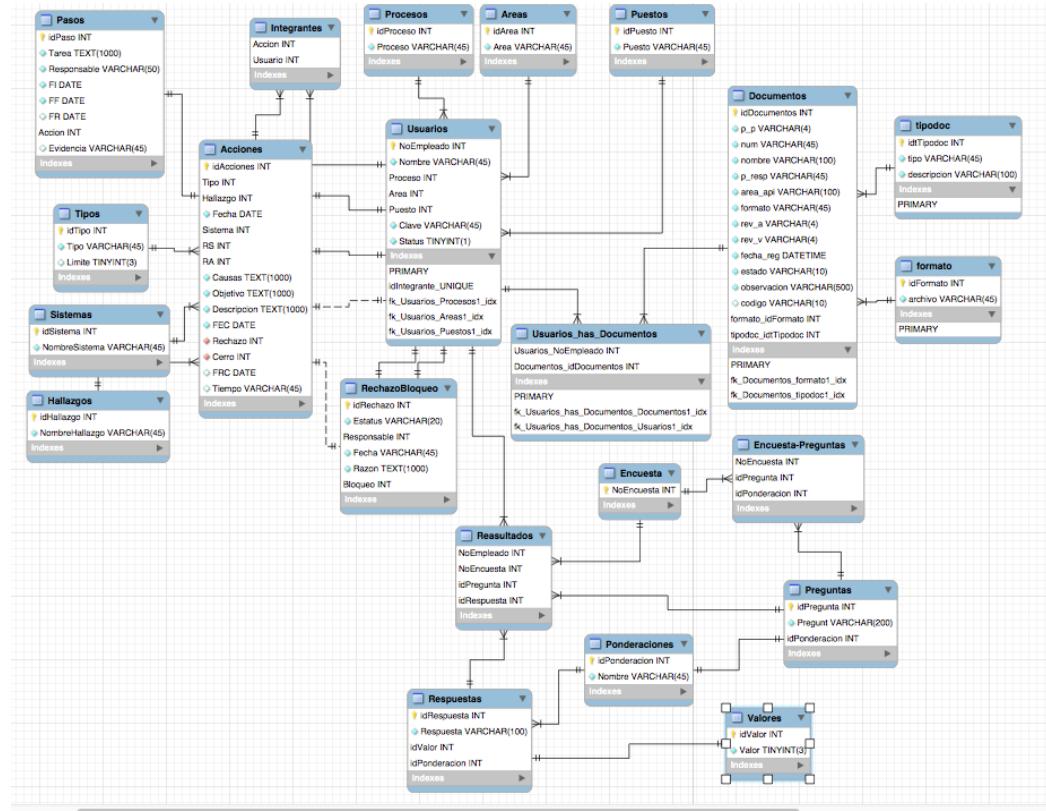


Imagen 61. Base de datos.

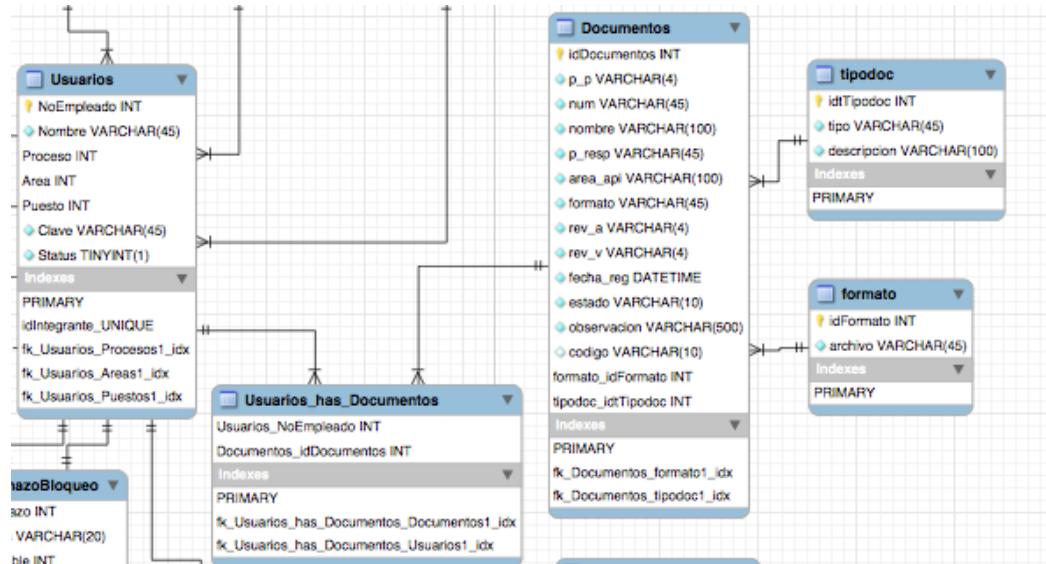


Imagen 62. Módulo documentos.

```

1 -- phpMyAdmin SQL Dump
2 -- version 4.3.11
3 -- http://www.phpmyadmin.net
4 --
5 -- Servidor: 127.0.0.1
6 -- Tiempo de generación: 08-03-2017 a las 18:30:48
7 -- Versión del servidor: 5.6.24
8 -- Versión de PHP: 5.6.8
9
10 SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
11 SET time_zone = "+00:00";
12
13
14 /*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
15 /*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
16 /*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
17 /*!40101 SET NAMES utf8 */;
18
19 --
20 -- Base de datos: `rppc`
21 --
22
23 DELIMITER $$*
24 --
25 -- Funciones
26 --
27 CREATE DEFINER=`root`@`localhost` FUNCTION `workdaydiff`(b date, a date) RETURNS int(11)
28 |   DETERMINISTIC
29 |   COMMENT 'working day difference for 2 dates'
30 BEGIN
31
32 DECLARE freedays int;
33
34 SET freedays = 0;
35 SET @x = DATEDIFF(b, a);
36 IF @x<0 THEN
37     SET @m = a;
38     SET a = b;
39     SET b = @m;
40     SET @m = -1;
41 ELSE
42     SET @m = 1;
43 END IF;

```

Imagen 63. Base de datos 1.

```

43 END IF;
44 SET @x = abs(@x) + 1;
45
46 SET @w1 = WEEKDAY(a)+1;
47 SET @wx1 = 8-@w1;
48 IF @w1>5 THEN
49     SET @w1 = 0;
50 ELSE
51     SET @w1 = 6-@w1;
52 END IF;
53
54 SET @wx2 = WEEKDAY(b)+1;
55 SET @w2 = @wx2;
56 IF @w2>5 THEN
57     SET @w2 = 5;
58 END IF;
59
60 SET @weeks = (@x-@wx1-@wx2)/7;
61 SET @noweekends = (@weeks*5)+@w1-@w2;
62
63 SET @result = @noweekends-freedays;
64 RETURN @result:@m;
65 END$$
66
67 DELIMITER ;
68
69 --
70
71 --
72 -- Estructura de tabla para la tabla `acciones`
73 --
74
75 CREATE TABLE IF NOT EXISTS `acciones` (
76     `idAcciones` int(11) NOT NULL,
77     `Tipo` int(11) NOT NULL,
78     `Hallazgo` int(11) NOT NULL,
79     `Fecha` date NOT NULL,
80     `Sistema` int(11) NOT NULL,
81     `RS` int(11) NOT NULL,
82     `RA` int(11) NOT NULL,
83     `Causas` text NOT NULL,
84     `Objetivo` text NOT NULL,
85     `Descripcion` text NOT NULL,

```

Imagen 64. Base de datos 2.

```

85   `Descripcion` text NOT NULL,
86   `FEC` date NOT NULL,
87   `Rechazo` int(11) DEFAULT NULL,
88   `Cerro` int(11) DEFAULT NULL,
89   `FRC` date DEFAULT NULL,
90   `Tiempo` varchar(45) DEFAULT NULL
91 ) ENGINE=InnoDB AUTO_INCREMENT=25 DEFAULT CHARSET=utf8;
92
93 --
94 -- Volcado de datos para la tabla `acciones`
95 --
96
97 INSERT INTO `acciones`(`idAcciones`, `Tipo`, `Hallazgo`, `Fecha`, `RS`, `RA`, `Causas`, `Objetivo`, `Descripcion`, `FEC`,
98 (2, 1, 1, '2017-03-15', 1, 1, 24, 'qq', 'qq', 'qq', '2017-03-23', NULL, NULL, NULL, NULL),
99 (3, 1, 1, '2017-03-06', 1, 11, 1, '1', '1', '1', '2017-03-06', NULL, NULL, NULL, NULL),
100 (4, 2, 1, '2017-03-06', 1, 1, 1, 'q', 'q', 'q', '2017-03-16', NULL, NULL, NULL, NULL),
101 (5, 2, 1, '2017-03-06', 1, 1, 1, 'q', 'q', 'q', '2017-03-16', NULL, NULL, NULL, NULL),
102 (6, 1, 1, '2017-03-06', 1, 1, 11, 'q', 'q', 'q', '2017-03-16', NULL, NULL, NULL, NULL),
103 (7, 1, 1, '2017-03-06', 1, 23, 25, 'Ejemplo', 'Ejemplo', 'Ejemplo', '2017-03-30', NULL, NULL, NULL, NULL),
104 (8, 1, 1, '2017-03-06', 1, 1, 11, 'q', 'q', 'q', '2017-03-30', NULL, NULL, NULL, NULL),
105 (9, 1, 1, '2017-03-06', 1, 1, 11, 'q', 'q', 'q', '2017-03-30', NULL, NULL, NULL, NULL),
106 (10, 1, 1, '2017-03-06', 1, 1, 11, 'qq', 'qq', 'qq', '2017-03-23', NULL, NULL, NULL, NULL),
107 (11, 1, 1, '2017-03-06', 1, 1, 11, 'qq', 'qq', 'qq', '2017-03-23', NULL, NULL, NULL, NULL),
108 (12, 1, 1, '2017-03-06', 1, 1, 11, 'qq', 'qq', 'qq', '2017-03-23', NULL, NULL, NULL, NULL),
109 (13, 1, 1, '2017-03-06', 1, 1, 11, 'q', 'w', 'e', '2017-03-29', NULL, NULL, '2017-03-31', NULL),
110 (14, 1, 1, '2017-03-06', 1, 23, 25, 'q', 'w', 'e', '2017-03-30', NULL, NULL, NULL, NULL),
111 (15, 3, 1, '2017-03-07', 1, 1, 23, 'qqq', 'www', 'eeee', '2017-03-30', NULL, NULL, NULL, NULL),
112 (16, 2, 1, '2017-03-07', 1, 1, 23, 'esta es una prueba de Causa preliminares', 'esta es una prueba de objetivo a lograr', 'esta es un
113 (17, 2, 1, '2017-03-07', 1, 1, 11, 'El Registro Pùblico de la Propiedad del Comercio del estado de Querétaro no cuenta con los recu
114 (19, 2, 1, '2017-03-08', 1, 1, 23, 'q', 'w', 'e', '2017-03-27', NULL, NULL, NULL, NULL),
115 (21, 1, 1, '2017-03-08', 1, 1, 11, 'q', 'w', 'e', '2017-03-24', NULL, NULL, NULL, NULL),
116 (22, 2, 1, '2017-03-08', 1, 24, 25, 'q', 'q', 'q', '2017-11-22', NULL, NULL, NULL, NULL),
117 (23, 1, 1, '2017-03-08', 1, 1, 1, 'asdfasdfasd', 'asdfasdfasdf', 'asdfasdfsda', '2017-03-31', NULL, NULL, NULL, NULL),
118 (24, 2, 1, '2017-03-08', 1, 23, 24, 'Kk', 'Ooo', 'Nnn', '2017-03-08', NULL, NULL, NULL, NULL);
119
120 -----
121
122 --
123 -- Estructura de tabla para la tabla `areas`
124 --
125
126 CREATE TABLE IF NOT EXISTS `areas` (
127   `idArea` int(11) NOT NULL,

```

Imagen 65. Base de datos 4.

```

127   `idArea` int(11) NOT NULL,
128   `Area` varchar(45) NOT NULL
129 ) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8;
130
131 --
132 -- Volcado de datos para la tabla `areas`
133 --
134
135 INSERT INTO `areas`(`idArea`, `Area`) VALUES
136 (1, 'Aa');
137
138
139 -----
140
141 --
142 -- Estructura de tabla para la tabla `documentos`
143 --
144
145 CREATE TABLE IF NOT EXISTS `documentos` (
146   `idDocumentos` int(11) NOT NULL,
147   `p_p` varchar(4) NOT NULL,
148   `num` varchar(45) NOT NULL,
149   `nombre` varchar(100) NOT NULL,
150   `p_resp` varchar(45) NOT NULL,
151   `area_api` varchar(100) NOT NULL,
152   `rev_a` varchar(4) NOT NULL,
153   `rev_v` varchar(4) NOT NULL,
154   `fecha_reg` datetime NOT NULL,
155   `estado` varchar(10) NOT NULL,
156   `observacion` varchar(500) NOT NULL,
157   `codigo` varchar(10) NOT NULL,
158   `formato_idFormato` int(11) NOT NULL,
159   `tipodoc_idTipodoc` int(11) NOT NULL
160 ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
161
162 -----
163
164 --
165 -- Estructura de tabla para la tabla `encuesta`
166 --
167
168 CREATE TABLE IF NOT EXISTS `encuesta` (
169   `NoEncuesta` int(11) NOT NULL

```

Imagen 66. Base de datos 5.

```

169 | `NoEncuesta` int(11) NOT NULL
170 ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
171 --
172 --
173 --
174 --
175 -- Estructura de tabla para la tabla `encuesta-preguntas`
176 --
177 --
178 CREATE TABLE IF NOT EXISTS `encuesta-preguntas` (
179 | `NoEncuesta` int(11) NOT NULL,
180 | `idPregunta` int(11) NOT NULL,
181 | `idPonderacion` int(11) NOT NULL
182 ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
183 --
184 --
185 --
186 --
187 -- Estructura de tabla para la tabla `formato`
188 --
189 --
190 CREATE TABLE IF NOT EXISTS `formato` (
191 | `idFormato` int(11) NOT NULL,
192 | `archivo` varchar(45) NOT NULL
193 ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
194 --
195 --
196 --
197 --
198 -- Estructura de tabla para la tabla `hallazgos`
199 --
200 --
201 CREATE TABLE IF NOT EXISTS `hallazgos` (
202 | `idHallazgo` int(11) NOT NULL,
203 | `NombreHallazgo` varchar(100) NOT NULL
204 ) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8;
205 --
206 --
207 -- Volcado de datos para la tabla `hallazgos`
208 --
209 --
210 INSERT INTO `hallazgos` (`idHallazgo`, `NombreHallazgo`) VALUES
211 (1, 'Interno');

```

Imagen 67. Base de datos 6.

```

211 (1, 'Interno');

212 --
213 --
214 --
215 --
216 -- Estructura de tabla para la tabla `integrantes`
217 --
218 --
219 CREATE TABLE IF NOT EXISTS `integrantes` (
220 | `Accion` int(11) NOT NULL,
221 | `Usuario` int(11) NOT NULL
222 ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
223 --
224 --
225 -- Volcado de datos para la tabla `integrantes`
226 --
227 --
228 INSERT INTO `integrantes` (`Accion`, `Usuario`) VALUES
229 (6, 1),
230 (7, 1),
231 (12, 1),
232 (13, 1),
233 (19, 1),
234 (21, 1),
235 (22, 1),
236 (23, 1),
237 (24, 1),
238 (6, 11),
239 (7, 11),
240 (10, 11),
241 (11, 11),
242 (15, 11),
243 (6, 23),
244 (8, 23),
245 (9, 23),
246 (10, 23),
247 (11, 23),
248 (17, 23),
249 (19, 23),
250 (23, 23),
251 (8, 25),
252 (9, 25),
253 (10, 25),

```

Imagen 68. Base de datos 7.

```

256 (16, 25),
257 (17, 25),
258 (22, 25);
259
260 --
261 --
262 --
263 -- Estructura de tabla para la tabla `pasos`
264 --
265
266 CREATE TABLE IF NOT EXISTS `pasos` (
267     `idPaso` int(11) NOT NULL,
268     `Tarea` text NOT NULL,
269     `Responsable` varchar(100) NOT NULL,
270     `FI` date NOT NULL,
271     `FF` date NOT NULL,
272     `FR` date DEFAULT NULL,
273     `Accion` int(11) NOT NULL,
274     `Evidencia` varchar(200) DEFAULT NULL
275 ) ENGINE=InnoDB AUTO_INCREMENT=14 DEFAULT CHARSET=utf8;
276
277 --
278 -- Volcado de datos para la tabla `pasos`
279 --
280
281 INSERT INTO `pasos` (`idPaso`, `Tarea`, `Responsable`, `FI`, `FF`, `Accion`, `Evidencia`) VALUES
282 (1, 'descripcion', 'responsable', '2017-03-06', '2017-03-06', NULL, 14, NULL),
283 (2, 'paso', 'responsable', '2017-03-16', '2017-03-31', NULL, 15, NULL),
284 (3, 'ejemplo de un paso', 'un responsable', '2017-03-16', '2017-03-22', NULL, 16, NULL),
285 (4, 'La titular de la Secretaría de Desarrollo Agrario, Territorial y Urbano expidió los Lineamientos del Programa de Modernización d
286 (5, 'Publicación de los Lineamientos del Programa de Modernización de los Registros Públicos de la Propiedad y Catastros en el Diario
287 (6, '11qqq', 'qqqq', '2017-03-23', '2017-03-20', NULL, 19, NULL),
288 (7, 'qq', 'qq', '2017-03-09', '2017-03-12', NULL, 21, NULL),
289 (8, 'qq', 'qq', '2017-03-27', '2017-03-12', NULL, 22, NULL),
290 (9, 'qqqq', 'qqq', '2017-03-20', '2017-03-30', NULL, 22, NULL),
291 (10, '1111', '1111', '2017-04-12', '2017-12-27', NULL, 22, NULL),
292 (11, 'sadfasdf', 'asdfasdf', '2017-03-08', '2017-03-10', NULL, 23, NULL),
293 (12, 'asdfsadf', 'asdfsdfafdfssdf', '2017-03-20', '2017-03-24', NULL, 23, NULL),
294 (13, 'Hyuinh', 'Oknu', '2017-03-08', '2017-03-10', NULL, 24, NULL);
295
296 --
297 --
298

```

Imagen 69. Base de datos 8.

```

298 --
299 -- Estructura de tabla para la tabla `ponderaciones`
300 --
301
302 ▼ CREATE TABLE IF NOT EXISTS `ponderaciones` (
303     `idPonderacion` int(11) NOT NULL,
304     `Nombre` varchar(45) NOT NULL
305 ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
306
307 --
308 --
309
310 -- Estructura de tabla para la tabla `preguntas`
311 --
312
313 ▼ CREATE TABLE IF NOT EXISTS `preguntas` (
314     `idPregunta` int(11) NOT NULL,
315     `Pregunt` varchar(200) NOT NULL,
316     `idPonderacion` int(11) NOT NULL
317 ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
318
319 --
320 --
321
322 -- Estructura de tabla para la tabla `procesos`
323 --
324
325 ▼ CREATE TABLE IF NOT EXISTS `procesos` (
326     `idProceso` int(11) NOT NULL,
327     `Proceso` varchar(45) NOT NULL
328 ) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8;
329
330 --
331 -- Volcado de datos para la tabla `procesos`
332 --
333
334 INSERT INTO `procesos` (`idProceso`, `Proceso`) VALUES
335 (1, 'aa');
336
337 --
338 --
339
340 -- Estructura de tabla para la tabla `puestos`

```

Imagen 70. Base de datos 9.

```

343 CREATE TABLE IF NOT EXISTS `puestos` (
344     `idPuesto` int(11) NOT NULL,
345     `Puesto` varchar(45) NOT NULL
346 ) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8;
347
348 --
349 -- Volcado de datos para la tabla `puestos`
350 --
351
352 INSERT INTO `puestos` (`idPuesto`, `Puesto`) VALUES
353 (1, 'aaaa');
354
355 --
356
357 --
358 -- Estructura de tabla para la tabla `reasultados`
359 --
360
361 CREATE TABLE IF NOT EXISTS `reasultados` (
362     `NoEmpleado` int(11) NOT NULL,
363     `NoEncuesta` int(11) NOT NULL,
364     `idPregunta` int(11) NOT NULL,
365     `idRespuesta` int(11) NOT NULL
366 ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
367
368 --
369
370 --
371 -- Estructura de tabla para la tabla `rechazobloqueo`
372 --
373
374 CREATE TABLE IF NOT EXISTS `rechazobloqueo` (
375     `idRechazo` int(11) NOT NULL,
376     `Estatus` varchar(20) NOT NULL,
377     `Responsable` int(11) NOT NULL,
378     `Fecha` varchar(45) NOT NULL,
379     `Razon` text NOT NULL,
380     `Bloqueo` int(11) NOT NULL
381 ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
382
383 --
384
385 --

```

Imagen 71. Base de datos 10.

```

388 CREATE TABLE IF NOT EXISTS `respuestas` (
389     `idRespuesta` int(11) NOT NULL,
390     `Respuesta` varchar(100) NOT NULL,
391     `idValor` int(11) NOT NULL,
392     `idPonderacion` int(11) NOT NULL
393 ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
394
395 --
396
397
398 --
399 -- Estructura de tabla para la tabla `sistemas`
400 --
401
402 CREATE TABLE IF NOT EXISTS `sistemas` (
403     `idSistema` int(11) NOT NULL,
404     `NombreSistema` varchar(100) NOT NULL
405 ) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8;
406
407 --
408 -- Volcado de datos para la tabla `sistemas`
409 --
410
411 INSERT INTO `sistemas` (`idSistema`, `NombreSistema`) VALUES
412 (1, 'Gestion');
413
414 --
415
416 --
417 -- Estructura de tabla para la tabla `tipodoc`
418 --
419
420 CREATE TABLE IF NOT EXISTS `tipodoc` (
421     `idTipodoc` int(11) NOT NULL,
422     `tipo` varchar(45) NOT NULL,
423     `descripcion` varchar(100) NOT NULL
424 ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
425
426 --
427
428 --
429 -- Estructura de tabla para la tabla `tipos`
430 --

```

Imagen 72. Base de datos 11.

```

430 --
431
432 CREATE TABLE IF NOT EXISTS `tipos` (
433     `idtipo` int(11) NOT NULL,
434     `Tipo` varchar(100) NOT NULL,
435     `Limite` int(11) DEFAULT NULL
436 ) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8;
437
438 --
439 -- Volcado de datos para la tabla `tipos`
440 --
441
442 INSERT INTO `tipos` (`idTipo`, `Tipo`, `Limite`) VALUES
443     (1, 'Mejora', NULL),
444     (2, 'Correctiva', NULL),
445     (3, 'Preventiva', NULL);
446
447 --
448
449 --
450 -- Estructura de tabla para la tabla `usuarios`
451 --
452
453 CREATE TABLE IF NOT EXISTS `usuarios` (
454     `NoEmpleado` int(11) NOT NULL,
455     `Nombre` varchar(45) NOT NULL,
456     `Proceso` int(11) NOT NULL,
457     `Area` int(11) NOT NULL,
458     `Puesto` int(11) NOT NULL,
459     `Clave` varchar(45) NOT NULL,
460     `Status` int(11) NOT NULL DEFAULT '1',
461     `Privilegio` int(11) NOT NULL DEFAULT '0',
462     `FR` date NOT NULL,
463     `FC` date DEFAULT NULL
464 ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
465
466 --
467 -- Volcado de datos para la tabla `usuarios`
468 --
469
470 INSERT INTO `usuarios` (`NoEmpleado`, `Nombre`, `Proceso`, `Area`, `Puesto`, `Clave`, `Status`, `Privilegio`, `FR`, `FC`) VALUES
471     (1, 'Miguel Angel Mejia Santiago', 1, 1, 1, '123456', 1, 1, '2017-03-05', NULL),
472     (11, 'Usuario Actualizado', 1, 1, 1, '123456', 1, 0, '2017-03-05', '2017-03-06'),

```

Imagen 73. Base de datos 12.

```

473     (23, 'Yesica Mejia Ramirez', 1, 1, 1, '123456', 1, 0, '2017-03-06', NULL),
474     (24, 'Estefani Garduño Alvarado', 1, 1, 1, '123456', 1, 0, '2017-03-06', '2017-03-06'),
475     (25, 'Maira Patricia Carmona Carmona', 1, 1, 1, '123456', 1, 1, '2017-03-06', NULL);
476
477 --
478
479 --
480 -- Estructura de tabla para la tabla `usuarios_has_documentos`
481 --
482
483 CREATE TABLE IF NOT EXISTS `usuarios_has_documentos` (
484     `Usuarios_NoEmpleado` int(11) NOT NULL,
485     `Documentos_idDocumentos` int(11) NOT NULL
486 ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
487
488 --
489
490 --
491 -- Estructura de tabla para la tabla `valores`
492 --
493
494 CREATE TABLE IF NOT EXISTS `valores` (
495     `idValor` int(11) NOT NULL,
496     `Valor` tinyint(3) NOT NULL
497 ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
498
499 --
500 -- Índices para tablas volcadas
501 --
502
503 --
504 -- Indices de la tabla `acciones`
505 --
506 ALTER TABLE `acciones`
507     ADD PRIMARY KEY (`idAcciones`,`Tipo`,`Hallazgo`,`Sistema`,`RS`,`RA`),
508     ADD KEY `fk_Acciones_Tipo_idx` (`Tipo`),
509     ADD KEY `fk_Acciones_idx` (`idAcciones`);
510
511 --
512 ALTER TABLE `areas`
513     ADD PRIMARY KEY (`idArea`);
514

```

Imagen 74. Base de datos 13.

```

514
515  -- Indices de la tabla `comentario`
516  --
517  ALTER TABLE `comentario`
518  | ADD PRIMARY KEY (`id_Comentario`,`Usuarios_NoEmpleado`,`Usuarios_Proceso`,`Usuarios_Area`,`Usuarios_Puesto`,`Tipo_id_Tipo`), ADD KE
520
521  --
522  -- Indices de la tabla `documentos`
523  --
524  ALTER TABLE `documentos`
525  | ADD PRIMARY KEY (`idDocumentos`,`formato_idFormato`,`tipodoc_idTipodoc`), ADD KEY `fk_Documentos_formato1_idx` (`formato_idFormato
526
527  --
528  -- Indices de la tabla `encuesta`
529  --
530  ALTER TABLE `encuesta`
531  | ADD PRIMARY KEY (`NoEncuesta`);
532
533  --
534  -- Indices de la tabla `encuesta-preguntas`
535  --
536  ALTER TABLE `encuesta-preguntas`
537  | ADD PRIMARY KEY (`NoEncuesta`,`idPregunta`,`idPonderacion`), ADD KEY `fk_Encuesta_has_Preguntas_Preguntas1_idx` (`idPregunta`,`idPo
538
539  --
540  -- Indices de la tabla `formato`
541  --
542  ALTER TABLE `formato`
543  | ADD PRIMARY KEY (`idFormato`);
544
545  --
546  -- Indices de la tabla `hallazgos`
547  --
548  ALTER TABLE `hallazgos`
549  | ADD PRIMARY KEY (`idHallazgo`);
550
551  --
552  -- Indices de la tabla `integrantes`
553  --
554  ALTER TABLE `integrantes`
555  | ADD PRIMARY KEY (`Accion`,`Usuario`), ADD KEY `fk_Acciones_has_Usuarios_Usuarios1_idx` (`Usuario`), ADD KEY `fk_Acciones_has_Usuar
556

```

Imagen 75. Base de datos 14.

```

514
515  -- Indices de la tabla `comentario`
516  --
517  ALTER TABLE `comentario`
518  | ADD PRIMARY KEY (`id_Comentario`,`Usuarios_NoEmpleado`,`Usuarios_Proceso`,`Usuarios_Area`,`Usuarios_Puesto`,`Tipo_id_Tipo`), ADD KE
520
521  --
522  -- Indices de la tabla `documentos`
523  --
524  ALTER TABLE `documentos`
525  | ADD PRIMARY KEY (`idDocumentos`,`formato_idFormato`,`tipodoc_idTipodoc`), ADD KEY `fk_Documentos_formato1_idx` (`formato_idFormato
526
527  --
528  -- Indices de la tabla `encuesta`
529  --
530  ALTER TABLE `encuesta`
531  | ADD PRIMARY KEY (`NoEncuesta`);
532
533  --
534  -- Indices de la tabla `encuesta-preguntas`
535  --
536  ALTER TABLE `encuesta-preguntas`
537  | ADD PRIMARY KEY (`NoEncuesta`,`idPregunta`,`idPonderacion`), ADD KEY `fk_Encuesta_has_Preguntas_Preguntas1_idx` (`idPregunta`,`idPo
538
539  --
540  -- Indices de la tabla `formato`
541  --
542  ALTER TABLE `formato`
543  | ADD PRIMARY KEY (`idFormato`);
544
545  --
546  -- Indices de la tabla `hallazgos`
547  --
548  ALTER TABLE `hallazgos`
549  | ADD PRIMARY KEY (`idHallazgo`);
550
551  --
552  -- Indices de la tabla `integrantes`
553  --
554  ALTER TABLE `integrantes`
555  | ADD PRIMARY KEY (`Accion`,`Usuario`), ADD KEY `fk_Acciones_has_Usuarios_Usuarios1_idx` (`Usuario`), ADD KEY `fk_Acciones_has_Usuar
556

```

Imagen 76. Base de datos 15.

```

559 --
560 ALTER TABLE `pasos`
561 | ADD PRIMARY KEY (`idPaso`,`Accion`), ADD KEY `fk_Pasos_Acciones1_idx` (`Accion`);
562 --
563 --
564 -- Indices de la tabla `ponderaciones`
565 --
566 ALTER TABLE `ponderaciones`
567 | ADD PRIMARY KEY (`idPonderacion`);
568 --
569 --
570 -- Indices de la tabla `preguntas`
571 --
572 ALTER TABLE `preguntas`
573 | ADD PRIMARY KEY (`idPregunta`,`idPonderacion`), ADD KEY `fk_Preguntas_Ponderaciones1_idx` (`idPonderacion`);
574 --
575 --
576 -- Indices de la tabla `procesos`
577 --
578 ALTER TABLE `procesos`
579 | ADD PRIMARY KEY (`idProceso`);
580 --
581 --
582 -- Indices de la tabla `puestos`
583 --
584 ALTER TABLE `puestos`
585 | ADD PRIMARY KEY (`idPuesto`);
586 --
587 --
588 -- Indices de la tabla `reasultados`
589 --
590 ALTER TABLE `reasultados`
591 | ADD PRIMARY KEY (`NoEmpleado`,`NoEncuesta`,`idPregunta`,`idRespuesta`), ADD KEY `fk_Usuarios_has_Encuesta_Encuesta1_idx` (`NoEncues
592 --
593 --
594 -- Indices de la tabla `rechazobloqueo`
595 --
596 ALTER TABLE `rechazobloqueo`
597 | ADD PRIMARY KEY (`idRechazo`,`Responsable`,`Bloqueo`), ADD KEY `fk_Rechazos_Usuarios1_idx` (`Responsable`), ADD KEY `fk_Rechazos_Us
598 --
599 --
600 -- Indices de la tabla `respuestas`
601 --
602 ALTER TABLE `respuestas`
603 | ADD PRIMARY KEY (`idRespuesta`,`idValor`,`idPonderacion`), ADD KEY `fk_Respuestas_Valor1_idx` (`idValor`), ADD KEY `fk_Respuestas_P
604 --
605 --
606 -- Indices de la tabla `sistemas`
607 --
608 ALTER TABLE `sistemas`
609 | ADD PRIMARY KEY (`idSistema`);
610 --
611 --
612 -- Indices de la tabla `tipo`
613 --
614 ALTER TABLE `tipo`
615 | ADD PRIMARY KEY (`id_Tipo`);
616 --
617 --
618 -- Indices de la tabla `tipodoc`
619 --
620 ALTER TABLE `tipodoc`
621 | ADD PRIMARY KEY (`idTipodoc`);
622 --
623 --
624 -- Indices de la tabla `tipos`
625 --
626 ALTER TABLE `tipos`
627 | ADD PRIMARY KEY (`idTipo`);
628 --
629 --
630 -- Indices de la tabla `usuarios`
631 --
632 ALTER TABLE `usuarios`
633 | ADD PRIMARY KEY (`NoEmpleado`,`Proceso`,`Area`,`Puesto`), ADD UNIQUE KEY `idIntegrante_UNIQUE` (`NoEmpleado`), ADD KEY `fk_Usuarios
634 --
635 --
636 -- Indices de la tabla `usuarios_has_documentos`
637 --
638 ALTER TABLE `usuarios_has_documentos`
639 | ADD PRIMARY KEY (`Usuarios_NoEmpleado`,`Documentos_idDocumentos`), ADD KEY `fk_Usuarios_has_Documentos_Documentos1_idx` (`Documento
640

```

Imagen 77. Base de datos 16.

```

598
599 --
600 -- Indices de la tabla `respuestas`
601 --
602 ALTER TABLE `respuestas`
603 | ADD PRIMARY KEY (`idRespuesta`,`idValor`,`idPonderacion`), ADD KEY `fk_Respuestas_Valor1_idx` (`idValor`), ADD KEY `fk_Respuestas_P
604 --
605 --
606 -- Indices de la tabla `sistemas`
607 --
608 ALTER TABLE `sistemas`
609 | ADD PRIMARY KEY (`idSistema`);
610 --
611 --
612 -- Indices de la tabla `tipo`
613 --
614 ALTER TABLE `tipo`
615 | ADD PRIMARY KEY (`id_Tipo`);
616 --
617 --
618 -- Indices de la tabla `tipodoc`
619 --
620 ALTER TABLE `tipodoc`
621 | ADD PRIMARY KEY (`idTipodoc`);
622 --
623 --
624 -- Indices de la tabla `tipos`
625 --
626 ALTER TABLE `tipos`
627 | ADD PRIMARY KEY (`idTipo`);
628 --
629 --
630 -- Indices de la tabla `usuarios`
631 --
632 ALTER TABLE `usuarios`
633 | ADD PRIMARY KEY (`NoEmpleado`,`Proceso`,`Area`,`Puesto`), ADD UNIQUE KEY `idIntegrante_UNIQUE` (`NoEmpleado`), ADD KEY `fk_Usuarios
634 --
635 --
636 -- Indices de la tabla `usuarios_has_documentos`
637 --
638 ALTER TABLE `usuarios_has_documentos`
639 | ADD PRIMARY KEY (`Usuarios_NoEmpleado`,`Documentos_idDocumentos`), ADD KEY `fk_Usuarios_has_Documentos_Documentos1_idx` (`Documento
640

```

Imagen 78. Base de datos 17.

```

640
641 -- Indices de la tabla `valores`
642 | ADD PRIMARY KEY (`idValor`);
643 |
644 ALTER TABLE `valores`
645 | MODIFY `idAcciones` int(11) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=25;
646 |
647 -- AUTO_INCREMENT de las tablas volcadas
648 |
649 |
650 |
651 -- AUTO_INCREMENT de la tabla `acciones`
652 |
653 |
654 ALTER TABLE `acciones`
655 | MODIFY `idAcciones` int(11) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=25;
656 |
657 -- AUTO_INCREMENT de la tabla `areas`
658 |
659 ALTER TABLE `areas`
660 | MODIFY `idArea` int(11) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=2;
661 |
662 -- AUTO_INCREMENT de la tabla `comentario`
663 |
664 ALTER TABLE `comentario`
665 | MODIFY `idComentario` int(11) NOT NULL AUTO_INCREMENT;
666 |
667 -- AUTO_INCREMENT de la tabla `documentos`
668 |
669 ALTER TABLE `documentos`
670 | MODIFY `idDocumentos` int(11) NOT NULL AUTO_INCREMENT;
671 |
672 -- AUTO_INCREMENT de la tabla `formato`
673 |
674 ALTER TABLE `formato`
675 | MODIFY `idFormato` int(11) NOT NULL AUTO_INCREMENT;
676 |
677 -- AUTO_INCREMENT de la tabla `hallazgos`
678 |
679 ALTER TABLE `hallazgos`
680 | MODIFY `idHallazo` int(11) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=2;
681 |
682 -- AUTO_INCREMENT de la tabla `pasos`

```

Imagen 79. Base de datos 18.

```

682 -- AUTO_INCREMENT de la tabla `pasos`
683 |
684 ALTER TABLE `pasos`
685 | MODIFY `idPaso` int(11) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=14;
686 |
687 -- AUTO_INCREMENT de la tabla `ponderaciones`
688 |
689 ALTER TABLE `ponderaciones`
690 | MODIFY `idPonderacion` int(11) NOT NULL AUTO_INCREMENT;
691 |
692 -- AUTO_INCREMENT de la tabla `preguntas`
693 |
694 ALTER TABLE `preguntas`
695 | MODIFY `idPregunta` int(11) NOT NULL AUTO_INCREMENT;
696 |
697 -- AUTO_INCREMENT de la tabla `procesos`
698 |
699 ALTER TABLE `procesos`
700 | MODIFY `idProceso` int(11) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=2;
701 |
702 -- AUTO_INCREMENT de la tabla `puestos`
703 |
704 ALTER TABLE `puestos`
705 | MODIFY `idPuesto` int(11) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=2;
706 |
707 -- AUTO_INCREMENT de la tabla `rechazobloqueo`
708 |
709 ALTER TABLE `rechazobloqueo`
710 | MODIFY `idRechazo` int(11) NOT NULL AUTO_INCREMENT;
711 |
712 -- AUTO_INCREMENT de la tabla `respuestas`
713 |
714 ALTER TABLE `respuestas`
715 | MODIFY `idRespuesta` int(11) NOT NULL AUTO_INCREMENT;
716 |
717 -- AUTO_INCREMENT de la tabla `sistemas`
718 |
719 ALTER TABLE `sistemas`
720 | MODIFY `idSistema` int(11) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=2;
721 |
722 -- AUTO_INCREMENT de la tabla `tipo`
723 |
724 ALTER TABLE `tipo`

```

Imagen 80. Base de datos 19.

```

724 ALTER TABLE `tipo`
725 | MODIFY `id_Tipo` int(11) NOT NULL AUTO_INCREMENT;
726 |
727 -- AUTO_INCREMENT de la tabla `tipodoc`
728 --
729 ALTER TABLE `tipodoc`
730 | MODIFY `idTipodoc` int(11) NOT NULL AUTO_INCREMENT;
731 --
732 -- AUTO_INCREMENT de la tabla `tipos`
733 --
734 ALTER TABLE `tipos`
735 | MODIFY `idTipo` int(11) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=4;
736 --
737 -- AUTO_INCREMENT de la tabla `valores`
738 --
739 ALTER TABLE `valores`
740 | MODIFY `idValor` int(11) NOT NULL AUTO_INCREMENT;
741 --
742 -- Restricciones para tablas volcadas
743 --
744 --
745 -- Filtros para la tabla `acciones`
746 --
747 --
748 ALTER TABLE `acciones`
749 ADD CONSTRAINT `fk_Acciones_Hallazgo1` FOREIGN KEY (`Hallazgo`) REFERENCES `hallazgos` (`idHallazgo`) ON DELETE NO ACTION ON UPDATE NO ACTION;
750 ADD CONSTRAINT `fk_Acciones_Rechazos1` FOREIGN KEY (`Rechazo`) REFERENCES `rechazobloqueo` (`idRechazo`) ON DELETE NO ACTION ON UPDATE NO ACTION;
751 ADD CONSTRAINT `fk_Acciones_Sistemas1` FOREIGN KEY (`Sistema`) REFERENCES `sistemas` (`idSistema`) ON DELETE NO ACTION ON UPDATE NO ACTION;
752 ADD CONSTRAINT `fk_Acciones_Tipo` FOREIGN KEY (`Tipo`) REFERENCES `tipos` (`idTipo`) ON DELETE NO ACTION ON UPDATE NO ACTION;
753 ADD CONSTRAINT `fk_Acciones_Usuarios1` FOREIGN KEY (`RS`) REFERENCES `usuarios` (`NoEmpleado`) ON DELETE NO ACTION ON UPDATE NO ACTION;
754 ADD CONSTRAINT `fk_Acciones_Usuarios2` FOREIGN KEY (`RA`) REFERENCES `usuarios` (`NoEmpleado`) ON DELETE NO ACTION ON UPDATE NO ACTION;
755 ADD CONSTRAINT `fk_Acciones_Usuarios3` FOREIGN KEY (`Cerro`) REFERENCES `usuarios` (`NoEmpleado`) ON DELETE NO ACTION ON UPDATE NO ACTION;
756 --
757 --
758 -- Filtros para la tabla `comentario`
759 --
760 ALTER TABLE `comentario`
761 ADD CONSTRAINT `fk_Comentario_Tipo1` FOREIGN KEY (`Tipo_id_Tipo`) REFERENCES `tipo` (`id_Tipo`) ON DELETE NO ACTION ON UPDATE NO ACTION;
762 ADD CONSTRAINT `fk_Comentario_Usuarios1` FOREIGN KEY (`Usuarios_NoEmpleado`, `Usuarios_Proceso`, `Usuarios_Area`, `Usuarios_Puesto`)
763 --
764 --
765 -- Filtros para la tabla `documentos`
766 --

```

Imagen 81. Base de datos 20.

```

766 --
767 ALTER TABLE `documentos`
768 ADD CONSTRAINT `fk_Documentos_formato1` FOREIGN KEY (`formato_idFormato`) REFERENCES `formato` (`idFormato`) ON DELETE NO ACTION ON UPDATE NO ACTION;
769 ADD CONSTRAINT `fk_Documentos_tipodoc1` FOREIGN KEY (`tipodoc_idTipodoc`) REFERENCES `tipodoc` (`idTipodoc`) ON DELETE NO ACTION ON UPDATE NO ACTION;
770 --
771 --
772 -- Filtros para la tabla `encuesta-preguntas`
773 --
774 ALTER TABLE `encuesta-preguntas`
775 ADD CONSTRAINT `fk_Encuesta_has_Preguntas_Encuesta1` FOREIGN KEY (`NoEncuesta`) REFERENCES `encuesta` (`NoEncuesta`) ON DELETE NO ACTION ON UPDATE NO ACTION;
776 ADD CONSTRAINT `fk_Encuesta_has_Preguntas_Preguntas1` FOREIGN KEY (`idPregunta`, `idPonderacion`) REFERENCES `preguntas` (`idPregunta`) ON DELETE NO ACTION ON UPDATE NO ACTION;
777 --
778 --
779 -- Filtros para la tabla `integrantes`
780 --
781 ALTER TABLE `integrantes`
782 ADD CONSTRAINT `fk_Acciones_has_Usuarios_Acciones1` FOREIGN KEY (`Accion`) REFERENCES `acciones` (`idAcciones`) ON DELETE NO ACTION ON UPDATE NO ACTION;
783 ADD CONSTRAINT `fk_Acciones_has_Usuarios_Usuarios1` FOREIGN KEY (`Usuario`) REFERENCES `usuarios` (`NoEmpleado`) ON DELETE NO ACTION ON UPDATE NO ACTION;
784 --
785 --
786 -- Filtros para la tabla `pasos`
787 --
788 ALTER TABLE `pasos`
789 ADD CONSTRAINT `fk_Pasos_Acciones1` FOREIGN KEY (`Accion`) REFERENCES `acciones` (`idAcciones`) ON DELETE NO ACTION ON UPDATE NO ACTION;
790 --
791 --
792 -- Filtros para la tabla `preguntas`
793 --
794 ALTER TABLE `preguntas`
795 ADD CONSTRAINT `fk_Preguntas_Ponderaciones1` FOREIGN KEY (`idPonderacion`) REFERENCES `ponderaciones` (`idPonderacion`) ON DELETE NO ACTION ON UPDATE NO ACTION;
796 --
797 --
798 -- Filtros para la tabla `reultados`
799 --
800 ALTER TABLE `reultados`
801 ADD CONSTRAINT `fk_Reultados_Preguntas1` FOREIGN KEY (`idPregunta`) REFERENCES `preguntas` (`idPregunta`) ON DELETE NO ACTION ON UPDATE NO ACTION;
802 ADD CONSTRAINT `fk_Reultados_Resuestas1` FOREIGN KEY (`idResuesta`) REFERENCES `respuestas` (`idResposta`) ON DELETE NO ACTION ON UPDATE NO ACTION;
803 ADD CONSTRAINT `fk_Usuarios_has_Encuesta_Encuestal` FOREIGN KEY (`NoEncuesta`) REFERENCES `encuesta` (`NoEncuesta`) ON DELETE NO ACTION ON UPDATE NO ACTION;
804 ADD CONSTRAINT `fk_Usuarios_has_Encuesta_Usuarios1` FOREIGN KEY (`NoEmpleado`) REFERENCES `usuarios` (`NoEmpleado`) ON DELETE NO ACTION ON UPDATE NO ACTION;
805 --
806 --
807 -- Filtros para la tabla `rechazobloqueo`
808 --

```

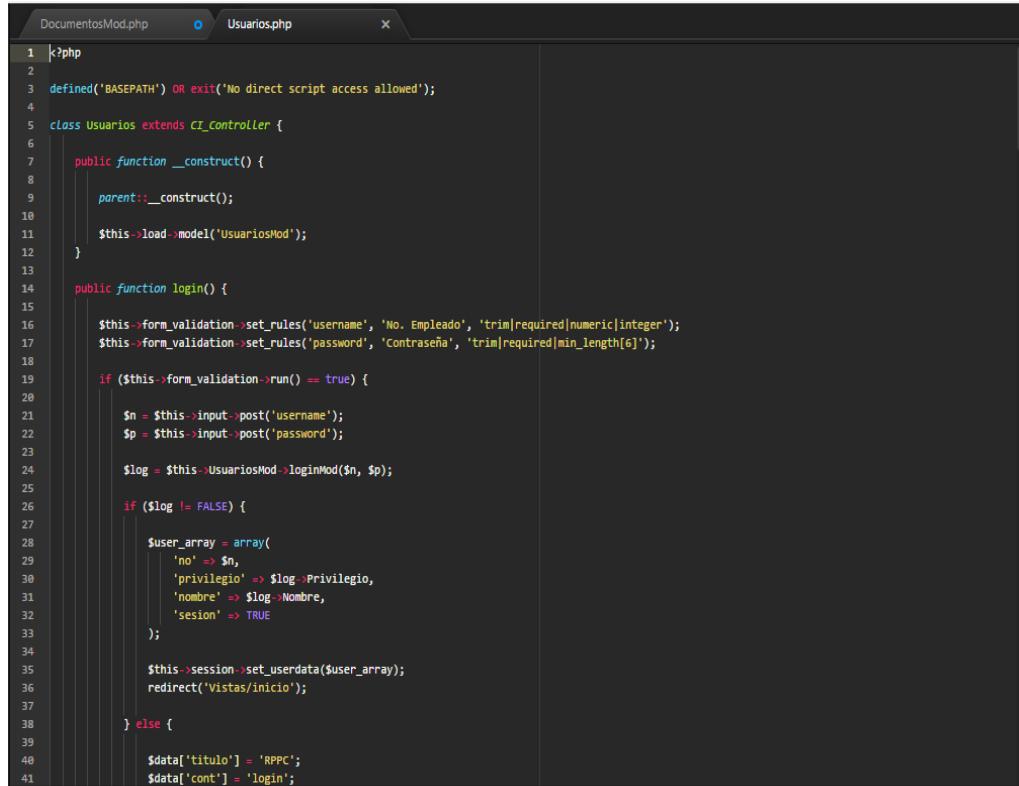
Imagen 82. Base de datos 21.

```

808 --
809 ALTER TABLE `rechazobloqueo`
810 ADD CONSTRAINT `fk_Rechazos_Usuarios1` FOREIGN KEY (`Responsable`) REFERENCES `usuarios` (`NoEmpleado`) ON DELETE NO ACTION ON UPDATE NO
811 ADD CONSTRAINT `fk_Rechazos_Usuarios2` FOREIGN KEY (`Bloqueo`) REFERENCES `usuarios` (`NoEmpleado`) ON DELETE NO ACTION ON UPDATE NO
812
813 --
814 -- Filtros para la tabla `respuestas`
815 --
816 ALTER TABLE `respuestas`
817 ADD CONSTRAINT `fk_Restuestas_Ponderaciones1` FOREIGN KEY (`idPonderacion`) REFERENCES `ponderaciones` (`idPonderacion`) ON DELETE NO
818 ADD CONSTRAINT `fk_Restuestas_Valori1` FOREIGN KEY (`idValor`) REFERENCES `valores` (`idValor`) ON DELETE NO ACTION ON UPDATE NO
819
820 --
821 -- Filtros para la tabla `usuarios`
822 --
823 ALTER TABLE `usuarios`
824 ADD CONSTRAINT `fk_Usuarios_Areas1` FOREIGN KEY (`Area`) REFERENCES `areas` (`idArea`) ON DELETE NO ACTION ON UPDATE NO ACTION,
825 ADD CONSTRAINT `fk_Usuarios_Procesos1` FOREIGN KEY (`Proceso`) REFERENCES `procesos` (`idProceso`) ON DELETE NO ACTION ON UPDATE NO A
826 ADD CONSTRAINT `fk_Usuarios_Puestos1` FOREIGN KEY (`Puesto`) REFERENCES `puestos` (`idPuesto`) ON DELETE NO ACTION ON UPDATE NO ACTIO
827
828 --
829 -- Filtros para la tabla `usuarios_has_documentos`
830 --
831 ALTER TABLE `usuarios_has_documentos`
832 ADD CONSTRAINT `fk_Usuarios_has_Documentos_Documentos1` FOREIGN KEY (`Documentos_idDocumentos`) REFERENCES `documentos` (`idDocumento
833 ADD CONSTRAINT `fk_Usuarios_has_Documentos_Usuarios1` FOREIGN KEY (`Usuarios_NoEmpleado`) REFERENCES `usuarios` (`NoEmpleado`) ON DEL
834
835 /*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
836 /*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
837 /*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

Imagen 83. Base de datos 22.

7.7.2. Clases



```

DocumentosMod.php          Usuarios.php
1 <?php
2
3 defined('BASEPATH') OR exit('No direct script access allowed');
4
5 class Usuarios extends CI_Controller {
6
7     public function __construct() {
8
9         parent::__construct();
10
11        $this->load->model('UsuariosMod');
12    }
13
14    public function login() {
15
16        $this->form_validation->set_rules('username', 'No. Empleado', 'trim|required|numeric|integer');
17        $this->form_validation->set_rules('password', 'Contraseña', 'trim|required|min_length[6]');
18
19        if ($this->form_validation->run() == true) {
20
21            $n = $this->input->post('username');
22            $p = $this->input->post('password');
23
24            $log = $this->UsuariosMod->loginMod($n, $p);
25
26            if ($log != FALSE) {
27
28                $user_array = array(
29                    'no' => $n,
30                    'privilegio' => $log->Privilegio,
31                    'nombre' => $log->Nombre,
32                    'sesion' => TRUE
33                );
34
35                $this->session->set_userdata($user_array);
36                redirect('Vistas/inicio');
37            } else {
38
39                $data['titulo'] = 'RPPC';
40                $data['cont'] = 'login';
41            }
42        }
43    }
44}

```

Imagen 84. Clase Usuarios 1.

```

40         $data['titulo'] = 'RPPC';
41         $data['cont'] = 'login';
42         $data['errorlog'] = 'true';
43         $this->load->view('login', $data);
44     }
45 }
46 }
47
48 public function logout() {
49     $user_array = array(
50         'sesion' => FALSE
51     );
52     $this->session->set_userdata($user_array);
53     redirect('Vistas/index');
54 }
55
56 public function datos() {
57
58     $selec = $this->input->post("seg");
59     $datos = $this->UsuariosMod->datosMod($selec);
60     echo json_encode($datos);
61 }
62
63 public function autoDatos() {
64
65     $selec = $_GET['term'];
66     $datos = $this->UsuariosMod->autoDatosMod($selec);
67
68     echo json_encode($datos);
69 }
70
71
72 public function add(){
73     $tabla = $this->input->post('tabla');
74     $col = $this->input->post('col');
75     $this->form_validation->set_rules('nombre', 'Nombre', 'trim|required|is_unique['.$tabla.'. '$col.]');

76     if ($this->form_validation->run() == true) {
77
78         $n = $this->input->post('nombre');
79
80     }

```

Imagen 85. Clase Usuarios.

```

157         $data['titulo'] = 'Formulario Usuarios';
158         $data['cont'] = 'frmUsuarios';
159         $this->load->view('vistas', $data);
160     }
161 }
162
163 public function delusuario(){
164
165     $no = $this->uri->segment(3);
166     $s = $this->uri->segment(4);
167
168     $data = array(
169         'status' => $s,
170         'FC' => date('Y-m-d')
171     );
172
173     $this->UsuariosMod->editar("usuarios", "NoEmpleado", $no, $data);
174
175     $data['lst'] = $this->UsuariosMod->allusuarios(false);
176     $data['titulo'] = 'Lista de Usuarios';
177     $data['cont'] = 'lstusuarios';
178     $this->load->view('vistas', $data);
179 }
180
181
182 public function ediusuario(){
183
184     $no = $this->uri->segment(3);
185
186     $editar = $this->UsuariosMod->editarUsuario($no);
187     $data['proceso'] = $this->UsuariosMod->todo("procesos");
188     $data['area'] = $this->UsuariosMod->todo("areas");
189     $data['puesto'] = $this->UsuariosMod->todo("puestos");
190
191     $data['editar'] = $editar;
192     $data['titulo'] = 'Editar Usuario';
193     $data['cont'] = 'frmUsuarios';
194     $this->load->view('vistas', $data);
195 }
196

```

Imagen 86. Clase Usuarios 2.

```

198 ✓ public function saveUsuario(){
199
200     $this->form_validation->set_rules('nombre', 'Nombre', 'trim|required');
201     $this->form_validation->set_rules('proceso', 'Proceso', 'trim|required');
202     $this->form_validation->set_rules('area', 'Area', 'trim|required');
203     $this->form_validation->set_rules('puesto', 'Puesto', 'trim|required');
204     $this->form_validation->set_rules('contraseña', 'Contraseña', 'trim|required');
205     $this->form_validation->set_rules('verificar', 'Verificar Contraseña', 'trim|required|matches[contraseña]');
206     $this->form_validation->set_rules('status', 'Privilegios', 'trim|required');
207     $this->form_validation->set_rules('Email', 'Email', 'trim|required');
208     $this->form_validation->set_rules('Tel', 'Tel', 'trim|required');
209     $this->form_validation->set_rules('Ext', 'Ext', 'trim|required');
210     $this->form_validation->set_rules('Marcacion', 'Marcacion', 'trim|required');
211
212     if ($this->form_validation->run() == true) {
213
214         $no = $this->input->post('noEmpleado');
215         $nom = $this->input->post('nombre');
216         $pro = $this->input->post('proceso');
217         $area = $this->input->post('area');
218         $puest = $this->input->post('puesto');
219         $pass = $this->input->post('contraseña');
220         $status = $this->input->post('status');
221         $Email = $this->input->post('Email');
222         $Tel = $this->input->post('Tel');
223         $Ext = $this->input->post('Ext');
224         $Marcacion = $this->input->post('Marcacion');
225
226         $edi = array(
227             'Nombre' => $nom,
228             'Proceso' => $pro,
229             'Area' => $area,
230             'Puesto' => $puest,
231             'Clave' => $pass,
232             'Privilegio' => $status,
233             'FC' => date('Y-m-d'),
234             'Email' => $Email,
235             'Tel' => $Tel,
236             'Ext' => $Ext,
237             'Marcacion' => $Marcacion
238         );
239
240         $this->UsuariosMod->editar("usuarios","NoEmpleado",$no,$edi);
241         $data['lst'] = $this->UsuariosMod->allusuarios(false);
242         $data['titulo'] = "Lista de Usuarios";
243         $data['cont'] = "listusuarios";
244         $this->load->view('vistas', $data);
245     } else {
246         $data['proceso'] = $this->UsuariosMod->todo("procesos");
247         $data['area'] = $this->UsuariosMod->todo("areas");
248         $data['puesto'] = $this->UsuariosMod->todo("puestos");
249         $data['titulo'] = "Formulario Usuarios";
250         $data['cont'] = "frmusuarios";
251         $this->load->view('vistas', $data);
252     }
253 }
254 public function buscar() {
255     $this->form_validation->set_rules('buscar', 'Busqueda', 'trim|required');
256     if ($this->form_validation->run() == true) {
257         $bus = $this->input->post('buscar');
258         $data['lst'] = $this->UsuariosMod->buscar($bus);
259         $data['titulo'] = "Lista de Usuarios";
260         $data['cont'] = "listusuarios";
261         $this->load->view('vistas', $data);
262     } else {
263         redirect('Vistas/listusuarios');
264     }
265 }
266 }
267
268 }

```

Imagen 87. Clase Usuarios 4.

```

229         'Area' => $area,
230         'Puesto' => $puest,
231         'Clave' => $pass,
232         'Privilegio' => $status,
233         'FC' => date('Y-m-d'),
234         'Email' => $Email,
235         'Tel' => $Tel,
236         'Ext' => $Ext,
237         'Marcacion' => $Marcacion
238     );
239
240     $this->UsuariosMod->editar("usuarios","NoEmpleado",$no,$edi);
241     $data['lst'] = $this->UsuariosMod->allusuarios(false);
242     $data['titulo'] = "Lista de Usuarios";
243     $data['cont'] = "listusuarios";
244     $this->load->view('vistas', $data);
245 } else {
246     $data['proceso'] = $this->UsuariosMod->todo("procesos");
247     $data['area'] = $this->UsuariosMod->todo("areas");
248     $data['puesto'] = $this->UsuariosMod->todo("puestos");
249     $data['titulo'] = "Formulario Usuarios";
250     $data['cont'] = "frmusuarios";
251     $this->load->view('vistas', $data);
252 }
253
254 public function buscar() {
255     $this->form_validation->set_rules('buscar', 'Busqueda', 'trim|required');
256     if ($this->form_validation->run() == true) {
257         $bus = $this->input->post('buscar');
258         $data['lst'] = $this->UsuariosMod->buscar($bus);
259         $data['titulo'] = "Lista de Usuarios";
260         $data['cont'] = "listusuarios";
261         $this->load->view('vistas', $data);
262     } else {
263         redirect('Vistas/listusuarios');
264     }
265 }
266
267
268 }

```

CRLF UTF-8 PHP master

Imagen 88. Clase Usuarios 5.

7.8. IMPLEMENTACIÓN

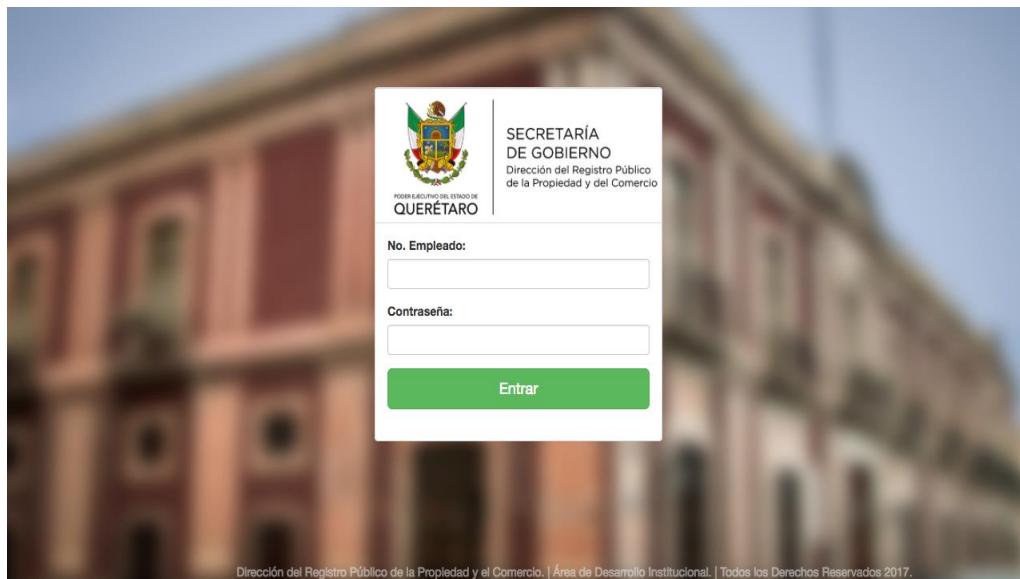


Imagen 89. Login.



Imagen 90. Inicio.



Imagen 91. Inicio 2.



Imagen 92. Inicio 3.

El Registro Público de la Propiedad y del Comercio del Estado de Querétaro opera bajo estándares internacionales de calidad y seguridad:

Política de Calidad

Brindar en la inscripción, publicidad y certificación de los actos y hechos jurídicos, un servicio honesto, oportuno y veraz que satisfaga las necesidades y expectativas de nuestros usuarios, bajo un sistema de calidad y mejora continua.

Política de Seguridad de la Información

Brindar en la inscripción, publicidad y certificación de los actos y hechos jurídicos, un servicio honesto, oportuno y veraz que satisfaga las necesidades y expectativas de nuestros usuarios, bajo un sistema de calidad y mejora continua.

Dirección del Registro Público de la Propiedad y el Comercio. | Área de Desarrollo Institucional. | © Todos los Derechos Reservados 2017.

Imagen 93. Inicio 4.

SECRETARIA DE GOBIERNO | Dirección del Registro Público de la Propiedad y el Comercio. No. 1 Aldo Omar Guajardo Chavez

Nuevo Usuario | Esciba el No. de Empleado o Nombre |

#	Nombre	Proceso	Área	Puesto	Teléfono	Extención	Marcación	Abreviada	Correo Electrónico
1	Aldo Omar Guajardo Chavez	AGN	PRUEBA	PRUEBA	171023281	44	123	aldoogc@gmail.com	
2	PRUEBA	AGN	PRUEBA	PRUEBA	1234567891	444	123	PRUEBA@GMAIL.COM	

Inicio | Empleados | Buscar | Nuevo Empleado | Ajustes | Generador de Acciones | Ambiente Laboral | Documentos | Análisis

Imagen 94. Usuarios.

SECRETARIA DE GOBIERNO | Dirección del Registro Público de la Propiedad y el Comercio.

No.1 Aldo Omar Guajardo Chavez



Nuevo Empleado

• No. Empleado No. de Empleado	• Nombre Nombre Completo	
→ Proceso Selecciona...	→ Área Selecciona...	→ Puesto Selecciona...
● Contraseña *****	● Repetir Contraseña *****	● Privilegios Usuario
● Teléfono Teléfono	● Extensión Extensión	● Marcación Abreviada Marcación
● Correo Electrónico Correo Electrónico		

Agregar

Inicio
Empleados
Buscar
Nuevo Empleado
Ajustes
Generador de Acciones
Ambiente Laboral
Documentos
Análisis

Imagen 95. Formulario Usuarios.

SECRETARIA DE GOBIERNO | Dirección del Registro Público de la Propiedad y el Comercio.

No.1 Miguel Angel Mejia Santiago



Agregar Nuevo

Lista de procesos

#	Nombres	Editar	Eliminar
1	aa	Editar	Eliminar

Inicio
Empleados
Buscar
Nuevo Empleado
Ajustes
Proceso
Área
Puestos
Generador de Acciones

Imagen 96. Procesos.

SECRETARIA DE GOBIERNO | Dirección del Registro Público de la Propiedad y el Comercio.

No.1 Miguel Angel Mejia Santiago



Agregar Nuevo

Lista de areas

#	Nombres	Editar	Eliminar
1	Aa		

- [Inicio](#)
- [Empleados](#)
- [Generador de Acciones](#)
- [Ambiente Laboral](#)
- [Documentos](#)
- [Buzón](#)

127.0.0.1/RPPC/index.php>Show/ajustes

Imagen 97. Áreas.

SECRETARIA DE GOBIERNO | Dirección del Registro Público de la Propiedad y el Comercio.

No.1 Miguel Angel Mejia Santiago



Agregar Nuevo

Lista de puestos

#	Nombres	Editar	Eliminar
1	aaaa		

- [Inicio](#)
- [Empleados](#)
- [Generador de Acciones](#)
- [Ambiente Laboral](#)
- [Documentos](#)
- [Buzón](#)

Imagen 98. Puesto.

SECRETARIA DE GOBIERNO | Dirección del Registro Público de la Propiedad y el Comercio. No.1 Aldo Omar Guajardo Chavez



Agregar Nuevo Documento **Vista General**

Lista de Documentos

#	Código	Nombre	Fecha	Estado	Formato	Tipo		
①	E122A01	UNO	2017-04-07 00:23:00	LIBRE		(A) ANEXOS AL MANUAL DE CALIDAD		
②	E122A03	TRES	2017-04-07 03:31:00	LIBRE		(A) ANEXOS AL MANUAL DE CALIDAD		
③	E122A04	CUATRO	2017-04-07 01:35:00	LIBRE		(A) ANEXOS AL MANUAL DE CALIDAD		
④	E122A05	CINCO	2017-04-07 01:36:00	LIBRE		(A) ANEXOS AL MANUAL DE CALIDAD		

Inicio **Empleados** < **Generador de Acciones** < **Ambiente Laboral** < **Documentos** < **Buscar** **Nuevo Documento** **Análisis** < **Subir** <

Imagen 99. Lista de documentos.

SECRETARIA DE GOBIERNO | Dirección del Registro Público de la Propiedad y el Comercio. No.1 Aldo Omar Guajardo Chavez



Editar Documento

Editar datos del documento

Nombre

Proceso Responsable

Área de Aplicación

<input type="checkbox"/> Avisos	<input type="checkbox"/> Inmobiliario
<input type="checkbox"/> Comercio	<input type="checkbox"/> Inscripción y Certificación
<input type="checkbox"/> Compras	<input type="checkbox"/> Incidentes
<input type="checkbox"/> Consulta Electrónica	<input type="checkbox"/> Integración de Folios
<input type="checkbox"/> Desarrollo Institucional	<input type="checkbox"/> Integración Histórica
<input type="checkbox"/> Digitalización Correcciones	<input type="checkbox"/> Jurídico
<input type="checkbox"/> Digitalización Día a Día	<input type="checkbox"/> Jurídico de Certificados
<input type="checkbox"/> Digitalización Histórica	<input type="checkbox"/> Módulo de Atención y Servicio a Usuarios
<input type="checkbox"/> Dirección	<input type="checkbox"/> PEM
	<input type="checkbox"/> Recepción de Dirección
	<input type="checkbox"/> Registral Foráneo

Buzón

Imagen 100. Formulario editar documento.

127.0.0.1/RPPC/index.php/Vistas/Inicio

<input type="checkbox"/> Digitalización Correcciones	<input type="checkbox"/> Jurídico SJR
<input type="checkbox"/> Digitalización Día a Día	<input type="checkbox"/> Módulo de Atención y Servicio a Usuarios
<input type="checkbox"/> Digitalización Histórica	<input type="checkbox"/> PEM
<input type="checkbox"/> Dirección	<input type="checkbox"/> Recepción de Dirección
<input type="checkbox"/> Documentos de Apoyo	<input type="checkbox"/> Registral Foráneo
<input type="checkbox"/> Encuadernación	<input type="checkbox"/> Registro Inmobiliario
<input type="checkbox"/> Fraccionamientos y Condominios	<input type="checkbox"/> Subdirecciones Foráneas
<input type="checkbox"/> Impresión y Sellado	<input type="checkbox"/> Tecnologías de la Información y Comunicaciones
	<input type="checkbox"/> Todo el RPPC
	<input type="checkbox"/> Unidad de Apoyo Administrativo

Revisión Anterior: 00 Revisión Vigente: 01 Estado: LIBRE

Observaciones:

Actualizar

Imagen 101. Formulario editar documento 2.

SECRETARIA DE GOBIERNO | Dirección del Registro Público de la Propiedad y el Comercio.

No.1 Aldo Omar Guajardo Chavez

Nuevo Documento

Ingresá el tipo de documento para generar código

Tipo de Documento

Selecioná el tipo de documento

P_P Número Tipo Código

SIGUIENTE >

Datos del documento

Imagen 102. Formulario documento.

SECRETARIA DE GOBIERNO | Dirección del Registro Público de la Propiedad y el Comercio.

No.1 Aldo Omar Guajardo Chavez.

Nuevo Documento

Datos del documento

Nombre
Nombre Completo

Proceso Responsable
Selección el tipo de documento

Fecha Registro
dd/mm/aaaa

Área de Aplicación

<input type="checkbox"/> Avisos	<input type="checkbox"/> Inmobiliario
<input type="checkbox"/> Comercio	<input type="checkbox"/> Inscripción y Certificación
<input type="checkbox"/> Compras	<input type="checkbox"/> Incidentes
<input type="checkbox"/> Consulta Electrónica	<input type="checkbox"/> Integración de Folios
<input type="checkbox"/> Desarrollo Institucional	<input type="checkbox"/> Integración Histórica
<input type="checkbox"/> Digitalización Correcciones	<input type="checkbox"/> Jurídico
<input type="checkbox"/> Digitalización Día a Día	<input type="checkbox"/> Jurídico de Certificados
<input type="checkbox"/> Digitalización Día a Día	<input type="checkbox"/> Jurídico SJR
<input type="checkbox"/> Digitalización	<input type="checkbox"/> Módulo de Atención y Servicio a Usuarios
<input type="checkbox"/> Digitalización	<input type="checkbox"/> PEM

Imagen 103. Formulario documento.

127.0.0.1/RPPC/index.php/Documentos/frmDocumentos

Imagen 104. Formulario documento 2.

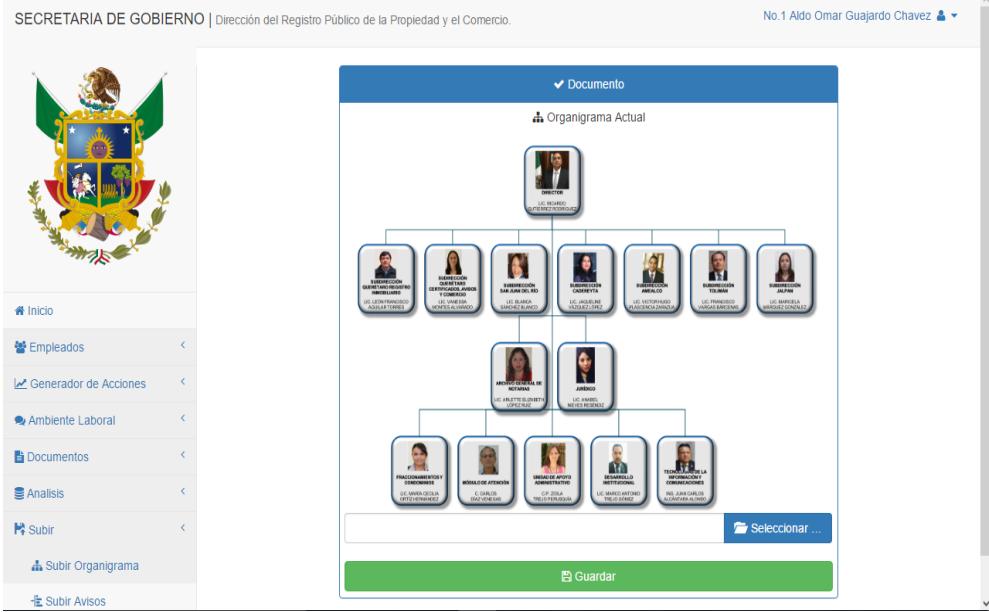


Imagen 105. Organigrama.

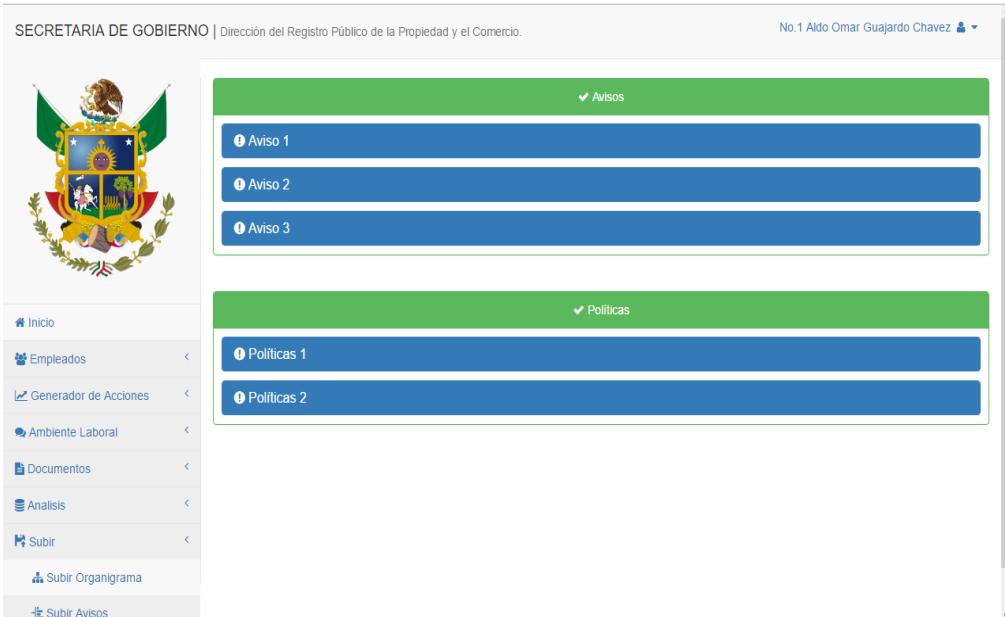


Imagen 106. Avisos.

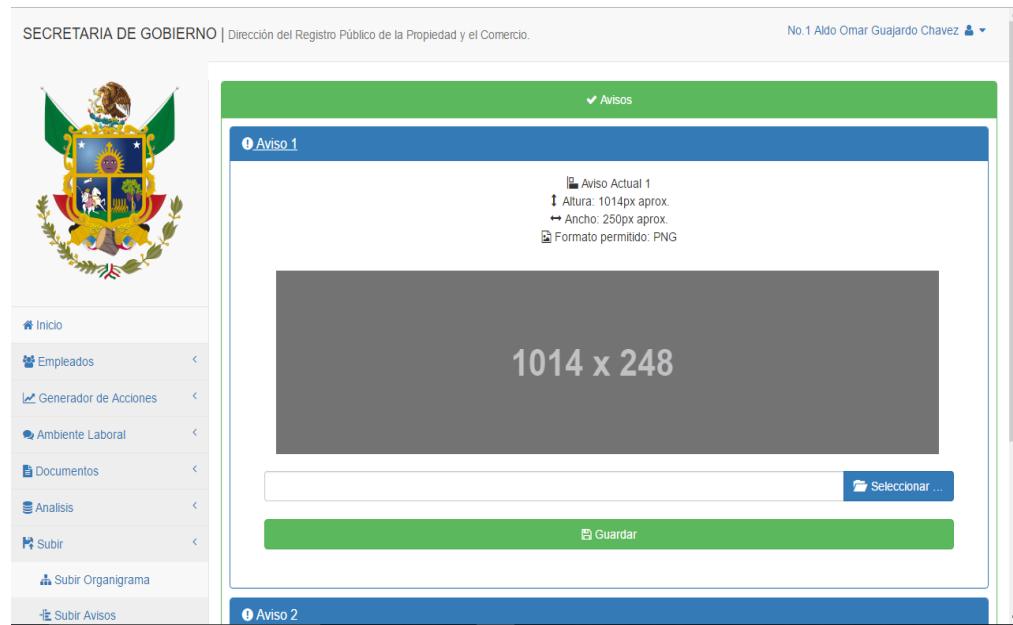


Imagen 107. Aviso 1.

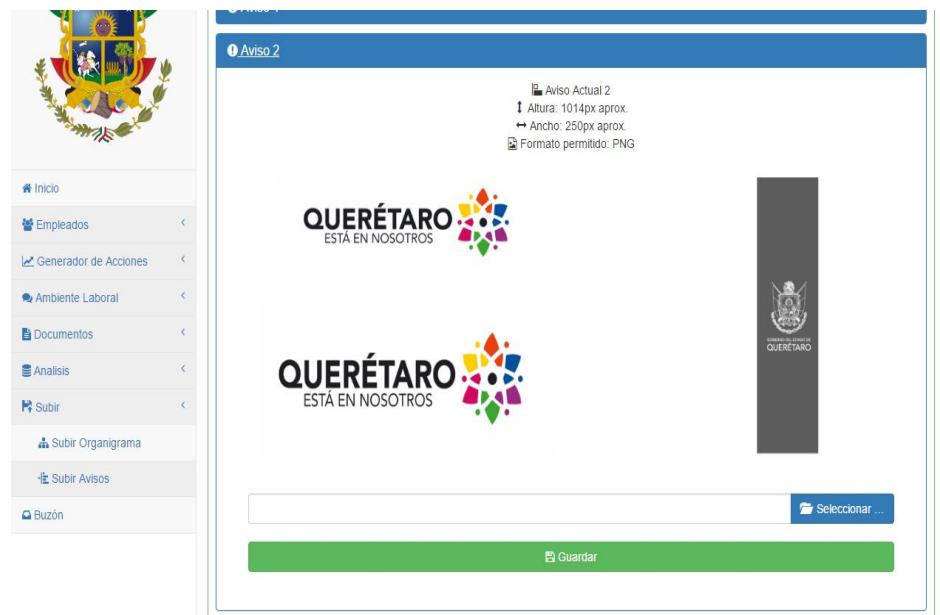


Imagen 108. Aviso 2.

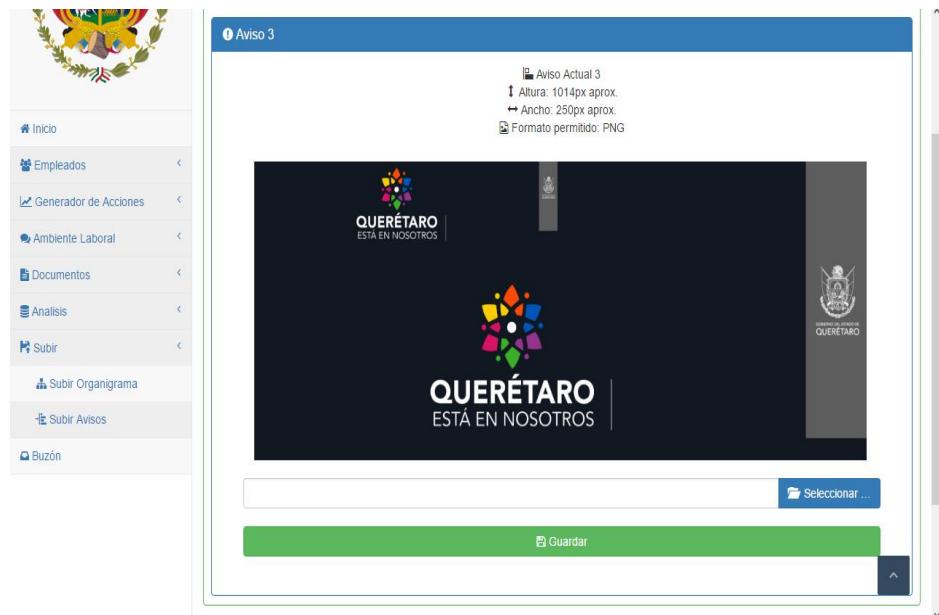


Imagen 109. Aviso 2.

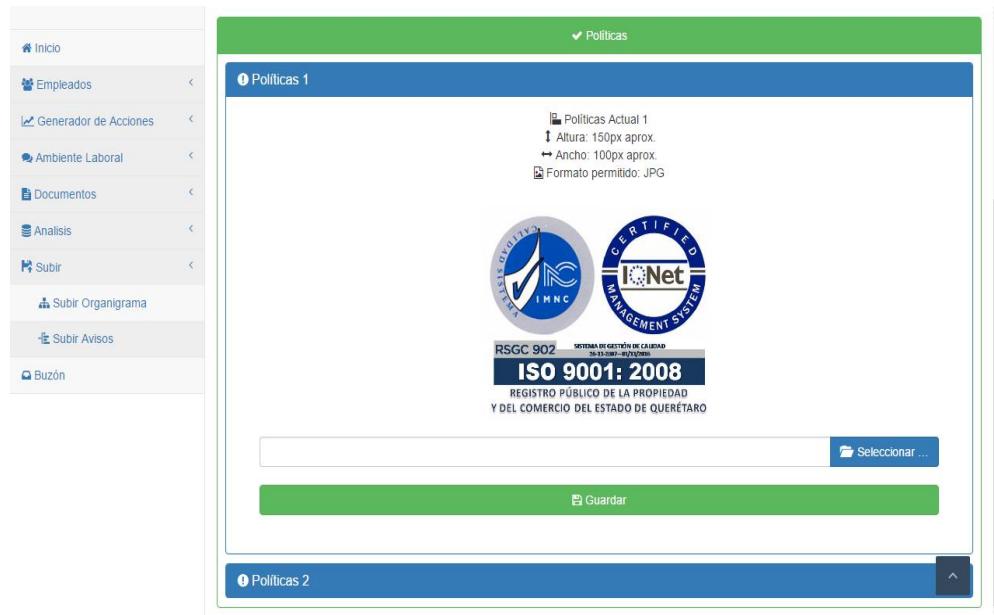


Imagen 110. Políticas 1.

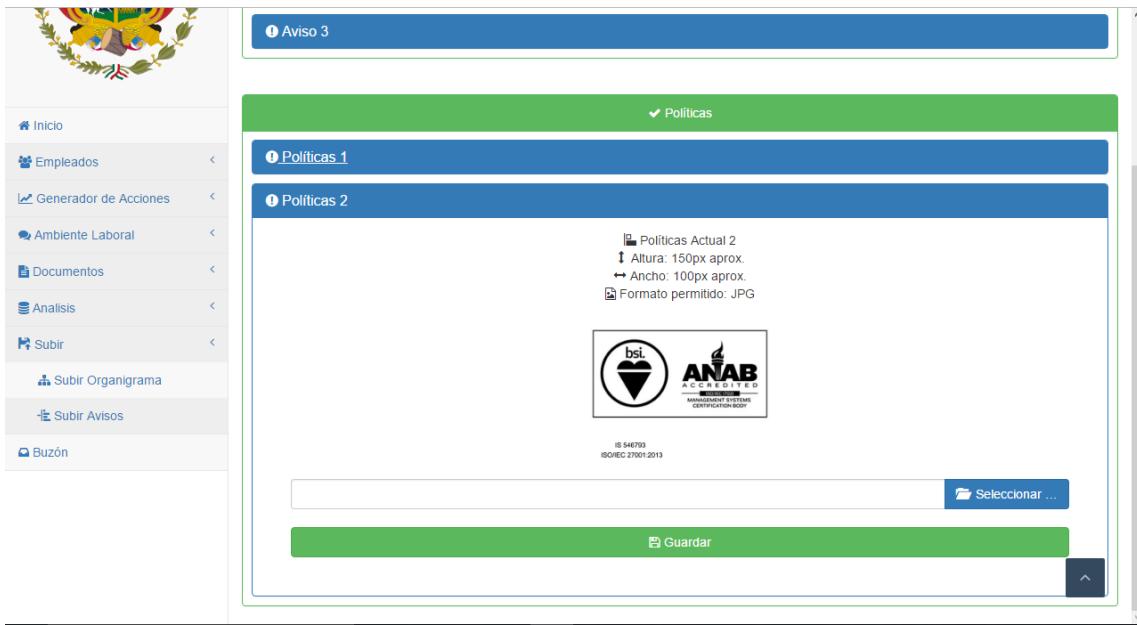


Imagen 111. Políticas 2.

7.9. VERIFICACIÓN

En este apartado se muestran la información en la cual, durante el transcurso de la elaboración del sitio web, se presentaron algunos problemas e inconvenientes, que afectaban el funcionamiento adecuado de dicho aplicación web.



Imagen 112. Slider.

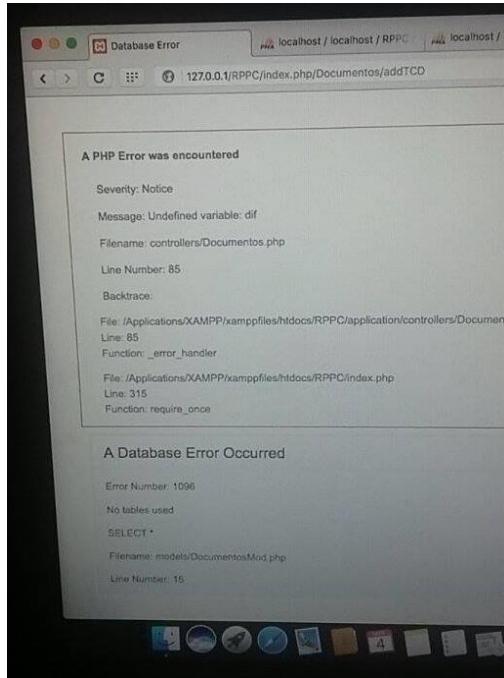


Imagen 113. Variable Indefinida.

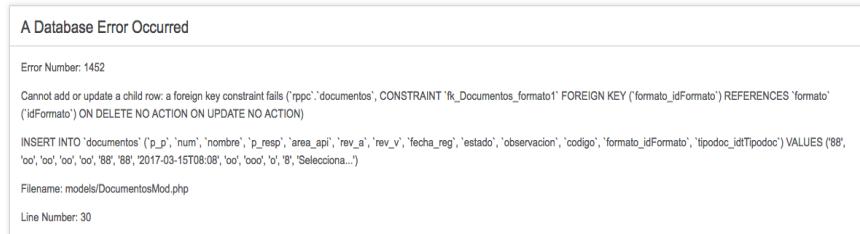


Imagen 114. Llave foránea.

CONCLUSIONES

Al concluir el desarrollo del proyecto de estadía se determinó que la aplicación web cumplió con el objetivo de agilizar el proceso gestionar documentos y ser capaz de darle al usuario un control sobre la gestión de los documentos con los que cuneta la Dirección del Registro Público de la Propiedad y el Comercio. Así también se concluye que el sitio cumplió con los alcances del proyecto establecidos desde el inicio, los cuales son: mostrar de forma intuitiva al usuario como agregar, borrar y actualizar los documentos almacenados. Permitiendo que la aplicación web agilice el proceso de capturar los documentos, siendo para el usuario el fácil su implementación.

El proceso de estadía es una gran oportunidad para los alumnos Ingeniería, ya que de esta manera puede poner en práctica todas las habilidades, conocimientos y destrezas dentro del campo laboral, aprendidas en el transcurso de toda la carrera, además de que puede conocer el ambiente real que se vive en el trabajo dentro de una empresa u organización. La experiencia que se adquiere en esta etapa de la formación académica del alumno es de gran importancia y relevancia ya que se aprende a integrarse al campo laboral de una manera positiva y efectiva.

REFERENCIAS

- Arias, F. G. (1999). *El proyecto de investigación 3ra edición.* Caracas: Episteme.
- Arias, F. G. (2012). *El proyecto de la investigación 6ta edición.* Caracas: EPISTEME, C.A.
- AULANEO. (2012). *Instrumentos de recolección de datos (IRD).* Recuperado el 12 de 11 de 2016, de AULANEO: <https://aulaneo.wordpress.com/mi/instrumentos-de-recoleccion-de-datos-ird/>
- Cabero Almenara, J. (., Romero Tena, R. (., Román Graván, P., Llorente Cejudo, M. d., & Castaño Garrido, C. (207). *Diseño y producción de TIC para la formación : nuevas tecnologías de la información y la comunicación.* (U. O. Catalunya, Ed.) Barcelona, España: Editorial UOE.
- Cabot Sagrera, J. (28 de Marzo de 2016). *Biblioteca Digital ECEST.* Obtenido de E-brary - Ingeniería del software: <http://site.ebrary.com.ezproxy.bibliotecaecest.mx/lib/bidigecestsp/reader.action?docID=10853350>
- Campderrich Falgueras, B. (2013). *Ingeniería del software.* (U. O. Catalunya, Ed.) Barcelona, España: UDC.
- Chamoun, Y. (2002). *Administración profesional de proyectos : la guía.* México, D.F.: McGraw-Hill.
- Chiavenato, Idalberto. (1992). *Introducción a la teoría de la administración.* México, DF.: Mc Graw Hill.
- Corporation, S. M. (03 de 10 de 2016). <https://netbeans.org/features/index.html>. Recuperado el 08 de 12 de 2016, de NetBeans IDE: <https://netbeans.org/>
- Crowder, D. (2005). *La biblia de Creación y Diseño Web.* Madrid: ANAYA.
- Dessler, G. (1979). *Organización y Administración.* Editorial Prentice-Hall.
- Ferron, A. M. (11 de Noviembre de 2016). *Modelos de bases de datos.* Obtenido de Bloq de Antonio M. Velez Ferron: <https://antoniovf.wordpress.com/2012/04/23/modelos-de-bases-de-datos/>

Foundation, E. (22 de 06 de 2016). <https://eclipse.org/ide/>. Recuperado el 08 de 12 de 2016, de Eclipse: <https://eclipse.org/>

Hernández, R., Fernández, C., & Baptista, P. (2006). *Metodología de la investigación* (4ta ed.). México, D.F.: Mc Graw Hill.

Hernández, R., Fernández, C., & Baptista, P. (2010). *Metodología de la investigación* (5ta ed.). México, D.F.: Mc Graw Hill.

Ibañez, J. (11 de Noviembre de 2016). *Manual de Administracion de Proyectos*. Obtenido de Lider de Proyecto: http://www.liderdeproyecto.com/manual/los_requerimientos.html

Institute, P. M. (2013). *Guía de los fundamentos de gestión de proyectos* (5^a edición ed.). Pensilvania : Project Management Institute, Inc.

JetBrains. (s.f.). <https://www.jetbrains.com/idea/>. Recuperado el 08 de 12 de 2016, de IntelliJ IDEA: <https://www.jetbrains.com>

Negocios, C. (06 de 11 de 2013). *Técnicas de investigación de mercados*. Recuperado el 12 de 11 de 2016, de Crece Negocios: <http://www.crecenegocios.com/tecnicas-de-investigacion-de-mercados>

Pressman, R. S. (2016). *Ingeniería del software: Un enfoque práctico* (Séptima edición ed.). D.F., México: Mc Graw Hill Educación.

Project, G. (07 de Febrero de 2016). *Gantt Project*. Obtenido de Gantt chart: <http://www.ganttproject.biz>

Quijado, J. L. (2007). *Domine JavaScript*. Madrid: Alfaomega.

Quijado, J. L. (2007). *Domine PHP y MySQL Programación Dinámica del lado del Servidor*. Madrid, España: RA-MA Editoriales.

Sierra, A. J. (2007). *AJAX en J2EE*. Madrid: Alfaomega.

Source Forge. (15 de 05 de 2013). *Source Forge OpenProj*. Obtenido de Source Forge: <https://sourceforge.net/projects/openproj>

SYSTEMS, S. (11 de Noviembre de 2016). *Diagrama de Actividades UML 2*. Obtenido de SPARX SYSTEMS: http://www.sparxsystems.com.ar/resources/tutorial/uml2_activitydiagram.html

Teniente López, E. C. (11 de Noviembre de 2016). <http://www.bibliotecaecest.mx/>. Obtenido de E-libro Software en UML: <http://site.ebrary.com.ezproxy.bibliotecaecest.mx/lib/bidigecestsp/reader.action?docID=11046224&ppg=40>

Toro López, F. J. (2014). *Ciencias Administrativas*. (E. Ediciones, Ed.) Bogotá , Colombia: Ecoe Ediciones.

Universidad Nacional Autónoma de México. (04 de 11 de 2016). *Industriales Historia Historia Gantt*. Obtenido de Universidad Nacional Autónoma de México:http://www.ingenieria.unam.mx/industriales/historia/carrera_historia_gantt.html

Urbano, C. A., & Yuni, J. A. (2006). *Técnicas para investigar y formular proyectos de investigación* vol. 1. Buenos Aires, Argentina: Editorial Brujas.

v, U. (08 de 12 de 2016). <https://www.uxpin.com/>. Recuperado el 08 de 12 de 2016, de UXPin : <https://www.uxpin.com/>

Werther, William. (9 de febrero de 2017). Biblioteca Digital ECEST. Obtenido de McGraw-Hill – Administración de recursos (7^a.Ed.):<http://site.ebrary.com.ezproxy.bibliotecaecest.mx/lib/bidigecestmhe/reader.action?docID=10820926>

GLOSARIO

Actor: Especifica un rol jugado por un usuario o cualquier otro sistema que interactúa con un sujeto.

Ciclo: Serie de fases o estados por las que pasa un acontecimiento o fenómeno y que se suceden en el mismo orden hasta llegar a una fase o estado a partir de los cuales vuelven a repetirse en el mismo orden.

Codificación: La transformación de la formulación de un mensaje a través de las reglas o normas de un código o lenguaje predeterminado. Cualquier operación que implique la asignación de un valor de símbolos o caracteres a un determinado mensaje verbal o no verbal con el propósito de transmitirlo a otros individuos o entidades que comparten el código.

Código (código fuente): Es un conjunto de líneas de texto que son las instrucciones que debe seguir la computadora para ejecutar dicho programa. Por tanto, en el código fuente de un programa está escrito por completo su funcionamiento.

Diagrama de Gantt: Herramienta gráfica cuyo objetivo es mostrar el tiempo de dedicación previsto para diferentes tareas o actividades a lo largo de un tiempo total determinado.

Diagrama E-R: Es un modelo de datos que describe las asociaciones que existe entre las diferentes categorías de datos dentro de un sistema de empresa o de información.

Diagrama Relacional: Una estructura relacional de datos consiste en una o varias tablas bidimensionales, las cuales se denominan relacionales. Las filas de la tabla representan un registro y las columnas contienen atributos.

EDT: (Estructura de Desglose del Trabajo) es una descomposición jerárquica, orientada al producto entregable del trabajo que será ejecutado por el equipo del proyecto, para lograr los objetivos del proyecto y crear los productos entregables requeridos

Formulario: Documento electrónico que contiene preguntas y espacios en blanco para llenar en la realización de ciertos trámites. Permite al usuario introducir datos los cuales son enviados a un servidor para ser procesados.

Framework: Es un entorno o ambiente de trabajo para desarrollo; dependiendo del lenguaje normalmente integra componentes que facilitan el desarrollo de aplicaciones como el soporte de programa, bibliotecas, plantillas y más.

Hosting: Es el servicio que provee a los usuarios de Internet un sistema para poder almacenar información, imágenes, vídeo, o cualquier contenido accesible vía web

IDE: Entorno de Desarrollo Integrado, llamado también IDE por sus siglas en inglés, es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI).

IEEE: El Instituto de Ingeniería Eléctrica y Electrónica en inglés Institute of Electrical and Electronics Engineers es una asociación mundial de ingenieros dedicada a la estandarización y el desarrollo en áreas técnicas. Con cerca de

425 000 miembros y voluntarios en 160 países, es la mayor asociación internacional sin ánimo de lucro formada por profesionales de las nuevas tecnologías, como ingenieros eléctricos, ingenieros en electrónica, científicos de la computación, ingenieros en computación, matemáticos aplicados, ingenieros en biomedicina, ingenieros en telecomunicación, ingenieros en mechatrónica, ingenieros en telemática etc.

Interfaz gráfica: Programa informático que actúa de interfaz de usuario, utilizando un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles en la interfaz.

Interfaz: Dispositivo capaz de transformar las señales generadas por un aparato en señales comprensibles por otro. Zona de comunicación o acción de un sistema sobre otro.

Internet: Red informática de nivel mundial que utiliza la línea telefónica para transmitir la información.

Módulo: Elemento con función propia concebido para poder ser agrupado de distintas maneras con otros elementos constituyendo una unidad mayor.

Navegador: Programa que permite navegar por internet u otra red informática de comunicaciones.

PDO: Data Objects es una extensión que provee una capa de abstracción de acceso a datos para PHP 5, con lo cual se consigue hacer uso de las mismas funciones para hacer consultas y obtener datos de distintos manejadores de bases de datos.

Plugin: Un complemento es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica.

POO: Es un paradigma de programación que usa objetos y sus interacciones para diseñar aplicaciones y programas de computadora. Está basado en varias técnicas, incluyendo herencia, modularidad, polimorfismo, y encapsulamiento.

Prototipo: Primer ejemplar que se fabrica de una figura, un invento u otra cosa, y que sirve de modelo para fabricar otras iguales, o molde original con el que se fabrica.

Requerimientos: La característica que requiere cumplir un producto o entregable asociado a una función en un proceso o servicio automatizado, o por automatizar.

Sesión: la duración de una conexión a un determinado sistema o red. Suele incluir el intercambio de paquetes de información entre un usuario y un servidor.

Sistema: permite almacenar y procesar información es el conjunto de hardware, software y recursos humanos.

Software: Conjunto de programas y rutinas que permiten a la computadora realizar determinadas tareas.

UML: El Lenguaje Unificado de Modelo (UML) es un lenguaje para especificar, visualizar, construir y documentar los artefactos de los sistemas software, así como para el modelado del negocio y otros sistemas no software

Variable: Está formada por un espacio en el sistema de almacenaje y un nombre simbólico que está asociado a dicho espacio. Ese espacio contiene una cantidad o información conocida o desconocida, es decir un valor.

Verificación: Comprobación del correcto funcionamiento de un aparato, instalación, etc.

Web: Conjunto de información que se encuentra en una dirección determinada de internet. Consiste en un documento electrónico que contiene información, cuyo formato se adapta para estar insertado en la World Wide Web, de manera que los usuarios a nivel mundial puedan entrar a la misma por medio del uso de un navegador, visualizándola con un dispositivo móvil.

ANEXOS

ANEXO 1. ENTREVISTA RECOLECCION DE DATOS

Entrevista

Entrevistador: Aldo Omar Guajardo Chávez **Entrevistado:** Marco Antonio Trejo Gómez

Cargo: Desarrollo Institucional

Empresa: Registró Público de la Propiedad y del Comercio.

Lugar: Querétaro, Querétaro. **Fecha:** 07/02/2017

El objetivo de esta entrevista, es la recolección de información que servirá para el diseño del sistema.

1. ¿Cuál es el objetivo de la aplicación web?

El objetivo de la aplicación es la administración de la dependencia a través de una plataforma.

2. ¿Qué tipo de información se manejará en la aplicación?

Documentos, como archivos de Word, Excel, Pdf, etc.

3. ¿Todas las personas acceden al mismo tipo de información?

No.

4. ¿Piensa quienes pueden llegar a hacer uso del sistema?

Solo los trabajadores de la dependencia tendrán acceso a la descarga de estos documentos.

5. ¿Cómo se pretende administrar la aplicación?

A través de diferentes usuarios que tenga la capacidad y privilegios que se encuentren registrados en el sistema.

6. ¿Qué se le piden al administrador?

Usuario, nombre y contraseña.

7. ¿Qué datos se le piden al usuario?

Usuario, nombre y contraseña.

8. ¿Qué datos se le piden al analista?

Usuario, nombre y contraseña.

ANEXO 2. LOGO REGISTRO PUBLICO DE LA PROPIEDAD Y DEL COMERCIO



SECRETARÍA
DE GOBIERNO
Dirección del Registro Público
de la Propiedad y del Comercio

ANEXO 3. ISO 9001: 2008



ANEXO 4. ISO7/IEC



IS 546793
ISO/IEC 27001:2013

Imagen 3. ISO/IEC